The Computers Have a Thousand Eyes: Towards a Practical and Ethical Video Analytics System for Person Tracking

Andrew Tzer-Yeu Chen

A thesis submitted in fulfilment of the requirements for the degree of
Doctor of Philosophy in Computer Systems Engineering,
The University of Auckland, 2019.

Abstract

With the rise of computer vision, video analytics systems have become more prevalent in the real-world, which automatically process video camera footage to produce information for human users. However, the majority of the existing research has looked at small parts of the overall puzzle in isolation, often focusing on accuracy while ignoring other practical requirements such as computation time and protecting privacy. The research presented in this thesis focuses on the development of a person tracking video analytics system that moves towards what is needed for real-world implementation with embedded systems. This is contextualised in a motivating scenario based on commercial market research, a modern application of computer vision moving away from traditional state-controlled surveillance. The aim is to balance the trade-off between accuracy and speed, with a modular pipeline to allow for better control and transparency into the algorithms being used. The developed video analytics system consists of a novel superpixel-based background estimation algorithm, person detection using off-the-shelf methods, an unsupervised person re-identification approach for classifying identities across multiple camera views, a Kalman Filter-based spatio-temporal model for tracking people by their positions, and a model fusion module that combines the appearance-based re-identification and spatio-temporal models together to improve the classification accuracy. In addition to the algorithms, this thesis also investigates the privacy loss encountered by these types of video analytics systems, firstly through a survey into public perceptions of privacy around surveillance cameras, and secondly through the proposal of a system architecture that uses computer vision and embedded systems to help protect privacy by default. This is then implemented with the use of smart cameras, and the impacts on accuracy, speed, and networking constraints are discussed. Lastly, further techniques for accelerating computer vision tasks in embedded system contexts are presented, with a case study demonstrating the use of Hardware/Software Co-design. The combination of all of these different factors brings a holistic view to the development of practical and ethical video analytics systems for person tracking, making progress towards overcoming the challenges faced by system designers and developers in real-world implementation.

the night has a thousand eyes
and a thousand eyes can't help but see...
the night has a thousand eyes
and a thousand eyes will see me...

Acknowledgements

Thank you to:

My supervisors, Morteza Biglari-Abhari and Kevin I-Kai Wang, for their continued guidance, for proof-reading many papers, for helping me navigate the bureaucratic processes of the university, for making time to let me drop in and chat at weird times during the day (and night). Without their support, the thesis journey would have much significantly more difficult, and I appreciate them for doing their best to help me along the way.

A number of academics throughout the university who have helped me along my journey in some way, including Zoran Salcic, Mark Andrews, Kevin Sowerby, Partha Roop, Karl Stol, Peter Richards, Thomas Lumley, Suzanne Woodward, Ethan Plaut, Jon Pearce, Nicola Gaston, Dion O'Neale, and many others. A significant thank you should be reserved for John Cater, who taught me image processing as part of ENGSCI712 (Computational Methods for Signal Processing), which sparked my interest in computer vision and provided me with an introduction into this space. He also forced me to learn LaTeX, which turned out to be very helpful in the end. Thank you also to Salim and Steve from the University of Wollongong for their contributions to the first part of my research work (which unfortunately hasn't made it into this thesis).

The undergraduate students who assisted with research work as part of summer research scholarships and Part IV projects, as well as the students who ~~suffered through~~ enjoyed my teaching. Thank you to the department for having confidence in me and giving me the opportunity to help shape the next generation of engineers and work hard to improve their educational experiences in a small way.

The participants who contributed towards my person tracking dataset, and those that (anonymously) responded to my privacy perception survey. Without your time and well-considered thoughts, validating my algorithms would have been a lot harder, and the privacy chapter of this thesis would probably not exist. Thank you also to subject matter experts who had conversations with me about my research, and helped to give me new ideas, facts, and perspectives that I could incorporate into the thesis.

The groups of people who helped make my doctoral education more rounded by giving me opportunities outside the scope of my main degree, including those at the Center for Innovation and Entrepreneurship, UniServices, ReturnOnScience, Momentum, the School of Graduate Studies, the Postgraduate Students Association, the Alumni Relations and Development Office, Bridget Williams Books, the Prime Minister's Chief Science Advisor's Office, the Office of the

Privacy Commissioner, the United Nations Youth Association of New Zealand, the Auckland University Robotics Association, and Te Ara Poutama at Auckland University of Technology.

All the friends who let me debate ethics and morality with them, particularly around privacy, for helping me better understand my thoughts and test my hypotheses. Additional thanks go to all of the people that I met through the tight-knit social media networks in New Zealand, who not only helped promote my work but also taught me so much more outside of my doctoral studies (check me out on Twitter, I'm @andrewtychen for now).

The other postgraduate students, technicians, and staff members in the Electrical and Computer Engineering Department for doing our best to create a positive community together under heavy work pressure, for always being willing to lend an extra hand wherever possible, for always having the right monitor cable when we needed one.

The members of the (unofficial) Embedded Systems Research Group for all of the support as we went on this doctoral journey together - Ben, Hammond, Nathan, Krystine, Ameer, Michael, Ryan, and Zac. We enjoyed many lunches, interactions with Slackbot, and teaching experiences (for better or for worse) together over the years. Having a group of close colleagues and friends is critical for surviving the doctoral experience, and I appreciate them for being here, there, and everywhere whenever we needed each other. A special thank you to Benjamin Tan, for being my Part IV Project partner in 2014 and entering the doctoral journey one year ahead of me, so that I could ask him a million questions about the process, and for showing that finishing a doctoral thesis is achievable.

My parents and family - Mum, Dad, Wendy, and Doug - for making sure I was fed, for keeping me healthy, for driving me across the city when I needed transportation. They put up with me when I was tired and cranky, and reduced my burdens in so many unseen and underappreciated ways.

Charlotte, for making me feel worthy of continuing on this journey, for making me feel better when times were tough, for making me feel loved. Four and a half years ago neither of us knew what we were in for, but now as we head towards the completion of my doctoral degree we have the next phase of our lives ahead of us, finally with both of us living in the same city. I'm really looking forward to it.

Contents

List of Publications

The following peer-reviewed and published papers have been adpated for use in parts of this thesis:

- A. T.-Y. Chen, M. Biglari-Abhari, K. I.-K. Wang, A. Bouzerdoum, and F. H. C. Tivive, "Hardware/Software Co-design for a Gender Recognition Embedded System", *International Conference on Industrial, Engineering, and Other Applications of Applied Intelligent Systems (IEA/AIE)*, pp 541-552, 2016.
- A. T.-Y. Chen, M. Biglari-Abhari, and K. I.-K. Wang, "Trusting the Computer in Computer Vision: A Privacy-Affirming Framework", *Conference on Computer Vision and Pattern Recognition (CVPR), Workshop on The Bright and Dark Sides of Computer Vision: Challenges and Opportunities for Privacy and Security (CV-COPS)*, pp 56-63, 2017.
- A. T.-Y. Chen, J. Fan, M. Biglari-Abhari, and K. I.-K. Wang, "A Computationally Efficient Pipeline for Camera-based Indoor Person Tracking", *IEEE Image and Vision Computing New Zealand (IVCNZ)*, 2017.
- A. T.-Y. Chen, M. Biglari-Abhari, K. I.-K. Wang, A. Bouzerdoum, and F. H. C. Tivive, "Convolutional Neural Network Acceleration with Hardware/Software Co-design", *Applied Intelligence*, 48(5), pp 1288-1301, 2018.
- A. T.-Y. Chen, M. Biglari-Abhari, and K. I.-K. Wang, "SuperBE: Computationally-Light Background Estimation with Superpixels", *Journal of Real-Time Image Processing (JRTIP)*, 2018.
- A. T.-Y. Chen, S. Woodward, M Biglari-Abhari, and K. I.-K. Wang, "Public Perceptions of Privacy in Surveillance Camera Networks", *Journalism, Media and Democracy Conference (JMAD)*, 2018.
- A. T.-Y. Chen, M. Biglari-Abhari, and K. I.-K. Wang, "Fast One-Shot Learning for Identity Classification in Person Re-identification", *IEEE International Conference on Control, Automation, Robotics, and Vision (ICARCV)*, pp 1197-1203, 2018.
- A. T.-Y. Chen, M. Biglari-Abhari, and K. I.-K. Wang, "Context is King: Privacy Perceptions of Camera-based Surveillance", *IEEE International Conference on Advanced Video and Signal-based Surveillance (AVSS)*, 2018.
- A. T.-Y. Chen, R. Gupta, A. Borzenko, K. I.-K. Wang, and M. Biglari-Abhari, "Accelerating SuperBE with Hardware/Software Co-Design", *Journal of Imaging*, 4(10), 122, 2018.
- A. T.-Y. Chen, M. Biglari-Abhari, and K. I.-K. Wang, "Investigating Fast Re-identification for Multi-camera Indoor Person Tracking", *Computers and Electrical Engineering*, 77C, pp 273-288, 2019.

Chapter 1

Introduction

"The advancements in computer vision these days are creating tremendous new opportunities in analyzing images, that are exponentially impacting every business vertical, from automotive to advertising to augmented reality" - Evan Nisselson, Venture Capitalist[1]

By some estimates, neurons responsible for vision occupy approximately 30% of the brain [1], and it is estimated that vision accounts for as much as 80-85% of the sensory information entering our brains [2]. It has long been a goal for researchers to develop systems that can perceive the world with as much accuracy and resolution as biological human eyesight. Cameras can be treated as a sensor, just like a light sensor or temperature sensor, which provide data about the external environment to a computer. With robust vision, computer systems such as robots, intelligent environments, and surveillance systems can make more knowledgeable decisions based on the world around them. As Stanford University Professor Fei-Fei Li said, "if we want machines to think, we need to teach them to see" [3].

Computer vision and image processing have been hotly researched and popular areas over the last decade. Video analytics is a relatively new term, referring to the extraction of data and information from video footage. Also known as video content analysis, this is done in an automated way, often using advanced artificial intelligence and machine learning techniques. Where camera feeds once had to be manually watched by a human to interpret and make decisions, computers could potentially automatically process hundreds or thousands of camera feeds at once, generating alerts or statistics as needed. Naturally, this improvement in scalability and reduction in cost has attracted the attention of corporations, broadening the market beyond physical security and law enforcement. An increasing number of video analytics products have become available on the market in recent years, often embedded within other products offered to end consumers. Video analytics can be targeted towards a multitude of applications, from reading number plates to motion detection to asset management.

One of the most important video analytics applications is person tracking. Corporations and governments are interested in knowing who is where and at what times. This information can be used for many purposes, from helping design better customer experiences to keeping people

[1] Ken Weiner, "Why image recognition is about to transform business", *TechCrunch*, May 2016.

safe from security threats. People counting alone is estimated to become a $1+ billion market by 2022[2]. Higher-level insights into how people use physical spaces and being able to provide location-aware services are even bigger markets, and video analytics will play a large role in improving the functionality and capabilities in this space. It is estimated that video analytics broadly will grow to a $8+ billion market by 2023[3]. However, the primary barriers to adoption thus far have been accuracy rates below customer expectations and impractical cost structures. For example, requiring very expensive computing platforms to process large amounts of data quickly for relatively low-value information is not financially viable or sustainable. In other words, there have been practical challenges that constrain the translation of technology from the laboratory into the real-world.

"Our movements and feelings are constantly monitored, because surveillance is the business model of the digital age" - Katharine Viner, Journalist[4]

Video analytics is sometimes seen as business jargon - a euphemistic term that can hide the surveillance systems that are actually being built. Science fiction has no shortage of stories about cameras and surveillance, and we would be foolish to insist that technology is value neutral and that video analytics systems pose no threat to society. In China, a nationwide person tracking system using facial recognition to complement a social credit system is planned for completion by 2020[5]. But advances in video analytics have not only helped governments develop large-scale surveillance systems; they have also enabled a broader range of applications beyond state-driven security and safety. Corporations are increasingly interested in using cameras to gather information that generates additional value and drives up profits. Corporations affect the lives of consumers in different ways to governments, and will wield the power of surveillance systems differently too. There are clear dangers to our privacy, and technologists have an obligation to develop new systems in a way that does not create more harm than good.

The novel *Night Has A Thousand Eyes* by Cornell Woolrich was published in 1945, inspiring a variety of popular artworks including a film, a jazz standard, and a poem. But it is the pop song performed by Bobby Vee that gave the inspiration for the title of this thesis; the song referred to a faceless night that was always watching, with a thousand stars feeling like a thousand eyes. With modern computer vision techniques and hardware, perhaps we can now develop computers with a thousand eyes. Doing so in an ethical way is critical to maintaining our freedoms, and must be taken into account when designing video analytics systems.

1.1 Introductory Definitions

Before progressing any further, a few key terms should be defined. *Person detection* refers to the task of finding people in an image. This is often based on a person model, where the computer

[2]Fior Markets, "Global People Counting System Market Research Report 2018", July 2018.
[3]MarketsandMarkets, "Video Analytics Market Research Report (TC 3523)", May 2018.
[4]Katharine Viner, "A mission for journalism in a time of crisis", *The Guardian*, Nov 2017.
[5]Rachel Botsman, "Big data meets Big Brother as China moves to rate its citizens", *Wired*, Oct 2017

looks for objects/shapes/features in the image that match that model. *Person identification* refers to determining the identity of a detected person, whether that is a global identity (i.e. their real name or ID number that is shared with other systems) or a local identity (i.e. a simple ID number that is not connected to their true identity). *Person re-identification* is similar, but refers to the specific case of determining if a person that has just been detected is the same person as one that has been detected earlier, usually from a different camera and therefore under different angle and lighting conditions. *Person tracking* refers to determining the position of the identified individual, and connecting that up with the previous positions of the person to find the path (or track) that the person has travelled over time.

In this thesis, *person tracking* may be used as an interchangeable proxy for the term *video analytics*. This is not because they are the same thing, but because video analytics refers generally to the task of processing video footage, while person tracking is the specific processing being performed. *Surveillance* may also be used as an interchangeable proxy, again not because it is the same thing as video analytics (and generally there are significant differences in the intention behind the deployment of such systems), but because they share similar underlying technologies to achieve their different purposes. This is particularly prevalent in discussions around privacy, where surveillance is a more commonly used term and video analytics is yet to enter the general consciousness of the literature (especially since there is a field called "surveillance studies").

1.2 A Motivating Scenario

Person tracking and surveillance have traditionally been the domain of law enforcement and public safety, watching for criminal acts and acting as a deterrent. But as the costs of deploying large-scale camera networks continue to fall, and the abilities of artificial intelligence and computer vision continue to rise, more commercial entities will utilise these types of systems to gain insights into how customers use and interact with physical space. This is a form of business intelligence, giving system owners actionable information that can help improve customer experiences and business processes.

For example, let us imagine a typical supermarket. There are a lot of decisions about how that supermarket is set up, such as how aisles are laid out, where the products are placed, and so on, that are known to have strong impacts on consumer purchasing behaviour [4, 5, 6, 7]. Most commonly, these insights have come from stationing human market researchers with a clipboard and a pen, observing shoppers and taking notes manually. It is a boring job, with only partial temporal coverage, and there are observer effects that change how the shoppers behave when they know that they are being watched.

Instead of human researchers, we could set up a camera network that observes the shoppers all the time and automatically processes the video footage. At a basic level, the system would be able to detect people and their positions throughout the store. That would allow the system owner to count the number of people in the shop at any time, determine which aisles are most popular, and identify which paths customers are taking. Higher level insights can be

determined, such as generating alerts when checkout queues are too long, identifying which popular products should be placed closest to the entrance and exits, or scheduling staff to meet periodic consumer demand. Identities of observed shoppers could be linked to loyalty card information, which could provide new, previously unavailable insights such as which items a customer picked up but did not buy on this trip, allowing the system owner to send the customer a targeted special offer for their next visit.

A lot of companies are interested in the abilities of this type of system beyond retail applications. During the development of this thesis, a number of meetings were held with various companies to identify the needs and requirements of industry. A corporate building manager wanted to incorporate this technology into their existing meeting room management system to monitor occupancy. An airline expressed interest in using this technology to better inform staff scheduling decisions for their airport lounges. Shopping mall owners wanted to gather better metrics about the popularity of their client retail outlets. A city council was interested in measuring pedestrian traffic flows in the city centre to provide better evidence for pedestrian-friendly urban planning. These different applications will all have slightly varying requirements, and are mentioned here to show the diversity of potential applications. However, for the purpose of this thesis, the focus will remain on the supermarket scenario and the requirements relevant for that application.

1.3 Problem Statement

Indoor tracking is a well-investigated area with a multitude of potential solutions, ranging from active transmission systems like RFID, Bluetooth, and ultrasonic, to unobtrusive passive systems such as Computer Vision [8]. Camera-based approaches are suitable in scenarios where the subject (i.e. the person being tracked) should not or cannot be an active participant; for example, a Wi-Fi localisation approach that requires the person being tracked to download an app to their smart phone would be inappropriate for a security application that tracks unauthorised individuals in a workplace. Computer vision allows us to passively observe the environment and extract information, such as the current positions of people, for further analysis.

The title of this thesis has two keywords that indicate the requirements of this video analytics system. The first is **practical**, which refers to the need for this system to use multiple cameras to cover large spaces, operate in real-time, and at a reasonable cost. The second is **ethical**, which refers to the need for this system to operate in a way that reduces the privacy loss experienced by the people who are being observed. While functionality and privacy are often treated dichotomously in the literature, the work in this thesis aims to find ways in which these two objectives can be complementary.

In order to achieve this, a camera network is needed with multiple cameras that can generate a continuous track as people walk in and out of view of each camera. There are two main approaches for maintaining tracks between cameras - a camera topology-based approach that identifies the sources and sinks of each camera view and tracks moving objects within

camera views primarily based on motion and kinematics (or uses auxiliary sensors such as accelerometers to help track the objects between the camera views [9]), and a re-identification approach that matches features between people detected across multiple camera views and tracks primarily based on matching identities [10]. In this thesis, both of these models will be combined together to improve the accuracy of the tracking system. A tracking-by-detection [11] approach is used, i.e. detecting all people in each frame and using tracking and re-identification to match the people between frames.

Within the last decade, computer vision has incorporated artificial intelligence/machine learning to achieve various tasks. These two fields owe a lot to each other, and their growth has been largely symbiotic with the research communities from both sides working together to advance common goals. The current trend at computer vision conferences around the world is the use of Deep Learning and Convolutional Neural Networks (CNNs) [12]. These biologically-inspired systems are based on the neurons inside the mammalian visual cortex [13], and have shown to be very successful at a wide range of computer vision tasks, including object recognition [14], scene understanding [15], and person re-identification [16].

However, the main limitation of these approaches is that as the methods become more accurate, they also become more complicated and computationally expensive, pushing them further out of the reach of real-time execution, especially for resource-constrained embedded systems. While some research effort has been made towards accelerating neural network execution on embedded systems [17, 18], the struggle remains that embedded computers today do not have the computational resources available to run deep neural network inference in real-time. The work in this thesis also aims to develop alternative computationally efficient approaches for person tracking, balancing the need for high accuracy with real-time execution.

The focus on computationally-light algorithms suitable for embedded systems is driven by cost[6]. It is not cost-effective to use a full desktop with multiple thousand-dollar Graphics Processing Units (GPUs) for each camera. Instead, the trend is moving towards the use of smart cameras - imaging devices with on-board processing capabilities. Naturally, there is a cost-performance trade-off: in order to reduce the cost of the smart camera, the processing power is also lower. This thesis will also show the impact of deploying the developed algorithms onto embedded computing platforms, and describe how algorithms could be further accelerated using Hardware/Software Co-Design. Combining all of these factors together shows a viable path forward for the design of real-world video analytics systems.

1.4 Existing Person Tracking Systems

The literature has many examples of comprehensive person tracking systems that have been developed. Very simple examples from the early 2000s include [19] from Microsoft, which used stereo cameras and blob detection to determine the position of people in a smart home environment, and [20], which used colour histograms to describe detected individuals. In [21], optical flow is used to track moving people, with a simple kinematics model operating

[6] Joshua Reynolds, Hills Security, Interview on 17 August 2017

in the background to continue tracking when subjects are obscured. M2Tracker [22] was an early example of using multiple cameras viewing the same scene from different viewpoints to separate and identify people in crowded scenes. A team from Carnegie Mellon University [23] identified people using their gait (i.e. walking style), and built a comprehensive tracking system from multi-camera calibration through to data visualisation. There was significant state interest, with many of these projects funded by the American, European, and Japanese governments and militaries [24, 25]. A survey paper [26] from 2005 noted that "automated visual surveillance systems [are] one of the main current application domains in computer vision", but the problems were far from being solved at that point in time.

As image processing techniques continued to advance, the research effort moved in different directions. In [27], the modelling of people became more sophisticated, splitting the detected person up into parts and then describing each part by their colour and shape. A similar approach was taken in [11], except that the focus was on modelling the motion of those body parts and combining them together to track the overall individual. In [28], multiple cameras were calibrated so that a person's track could be placed within the context of a global map so that the relationship between the subject and each camera could be determined. In [29], the focus was on the use of wireless sensor nodes with cameras attached, processing footage directly and sending processed data to a central computer, reducing power consumption and bandwidth. In VideoWeb [30], multi-objective optimisation was used to change camera configurations in order to ensure that cameras are sending the appropriate level of data for what they are seeing, and so that the maximum amount of physical space is being covered by the cameras. In [31], a data fusion approach is used to combine cameras with infrared badge sensors and fingerprint scanners to localise individuals within an office building. ViSE [32] focused on enabling human users of the system to be able to search for individuals based on their clothing characteristics. All of these works showed development in different parts of the overall system, without necessarily building a complete system that could work together overall. A survey paper [10] from 2013 presented this as five main parts and argued that more effort needed to be put into integrating them together:

- Performing multi-camera calibration - producing a single co-ordinate system common for all cameras
- Calculating the topology of the cameras - finding the relationships between the camera positions and camera views
- Object or person re-identification between cameras - appearance-based classification of identities
- Object or person tracking - spatio-temporal-based classification of identities
- Activity analysis - making the extracted data useful for end users and their target applications

However, in most historical cases the accuracy rates were not sufficiently high enough and the processing too slow to justify further development and investment, and many projects disappeared. Perhaps the computer vision techniques were not yet sufficiently advanced, and

the single-core processing hardware was not fast enough. In the early 2010s, deep learning presented a huge leap forwards in artificial intelligence, and combined with convolutional neural networks, achieved significantly higher accuracy rates than before. Improvements in parallel computing, particularly through the use of GPUs, meant that these complicated neural network techniques could be feasibly developed and tested, only requiring days to chew through the training data rather than months or years.

This provided the backdrop for more recent approaches, which do not all use convolutional neural networks, but are many orders of magnitude more complex and sophisticated than those from the previous decade. Modern works focusing on individual tasks will be discussed in each relevant Chapter of the thesis, but it is worth noting some recent examples of complete person tracking systems here to demonstrate how the technology has evolved. In [33], a pilot camera at an information kiosk captures the appearance of an individual, which is then matched to people seen at other query cameras throughout the building using only the lower half of the body based on the colour and texture appearance. In contrast, [34] uses an ensemble of face detectors to extract faces from camera feeds and then perform re-identification - this team from Queensland University of Technology has deployed real-world systems in Australia and South America. In [35], smart cameras perform simple face detection and re-identification to determine if observed people match user-described filters, and send images to a central controller only when there is something interesting or relevant. The main focus was on novel scheduling of distributed node data across a large distributed network. In [36], cameras are connected to Field Programmable Gate Arrays (FPGAs) to perform background modelling and local tracking at the point of image capture, while depth estimation and global object tracking are performed with GPUs at a central server. These computing platforms allow for significant parallelism to combat the computational complexity of the algorithms. In [37], the system moves beyond person tracking and uses a convolutional neural network to achieve human activity recognition. This provides a new source of information, not just telling the user where the people are, but also potentially what they are doing.

As the complexity of these algorithms and systems has increased, the embedded world has been left behind. With limited resources, both in terms of processing/computation and power/energy, the achievements seen in the literature have not generally been seen deployed on embedded devices. In addition, many algorithms use assumptions that may not apply in the real world, such as consistent frame rates, fixed camera positions, and consistent lighting and noise conditions. The performance of these algorithms can be artificially constrained for research purposes, with limited image resolutions and/or limited frame rates, decreasing the usability of these algorithms in the real world. As a result, there are many algorithms, but only a few successful hardware implementations. This is despite the fact that real-time processing doesn't necessarily mean processing 25 frames per second (fps) - even 5-10fps can be sufficient for most applications[7].

It is important to note here that this short literature review does not focus on commercial offerings, even though the market is now hotly competitive - this is mostly because corporations

[7] Joshua Reynolds, Hills Security, Interview on 17 August 2017

tend to keep their methods and approaches secret, so it can be difficult to determine how their products actually work. However, from desk research into a number of different products, they seem to focus on relatively simpler tasks, such as person counting[8], queue detection[9], or heat mapping[10], which are tasks that don't necessarily require re-identification or tracking. There is also a lack of good multi-camera systems in the market, with most video analytics solutions operating on single camera views independently and then potentially aggregating statistics together at a higher level[11]. A few providers offer comprehensive person tracking services, but from interviews with distributors and customers, it appears that high accuracy is often overpromised, and that a number of early adopter customers have been left disillusioned and have turned off the analytics functionality of their systems[12]. It can be hard to separate what is marketing hype and propaganda, and what the true capability of these commercial video analytics systems really is. This is in line with the experiences of researchers who have tried to deploy advanced computer vision techniques into real systems.

These challenges are reflected in a recent paper [38] from a team at Northeastern University and Rensselaer Polytechnic Institute trying to implement a person tracking and surveillance network at an airport in the United States:

> "the deployment of a re-id[entification] algorithm in a real-world environment faces many practical constraints not typically encountered in an academic research laboratory. In contrast to recently purchased, high-quality digital cameras, a legacy surveillance system is likely to contain low quality, perhaps even analog, cameras whose positions and orientations cannot be altered to improve performance. The video collected by cameras in the network is likely to be transmitted to secure servers over limited-bandwidth links, and these servers are likely to have limited storage, since many cameras' data must be compressed and archived. These servers are also likely to be closed off from the Internet so that any algorithm upgrades and testing must be physically done on-site. Since the algorithm must run autonomously, a robust, crash-proof software architecture is required that takes advantage of any possible computational advantage (e.g. parallel or distributed processing) while still guaranteeing low latency. On the front end, the algorithm must run in real time, updating a ranked list of matching candidates as fast as they appear in each potential camera, and the results must be presented to the user in [an] easy-to-use, nontechnical interface."

This passage highlights the poor emulation of real-world environments in academic research, and a need to strongly understand the requirements of end users. The paper also identifies that many academic datasets that are used to test and validate algorithms do not adequately represent the real-world scenarios that these algorithms are expected to be implemented in.

[8] 3VR/Identiv, http://www.3vr.com/products/videoanalytics/peoplecounting
[9] Axis Communications, https://www.axis.com/products/axis-queue-monitor
[10] KiwiSecurity, https://www.kiwisecurity.com/activity-visualizer/
[11] Ibid.
[12] Joshua Reynolds, Hills Security, Interview on 17 August 2017

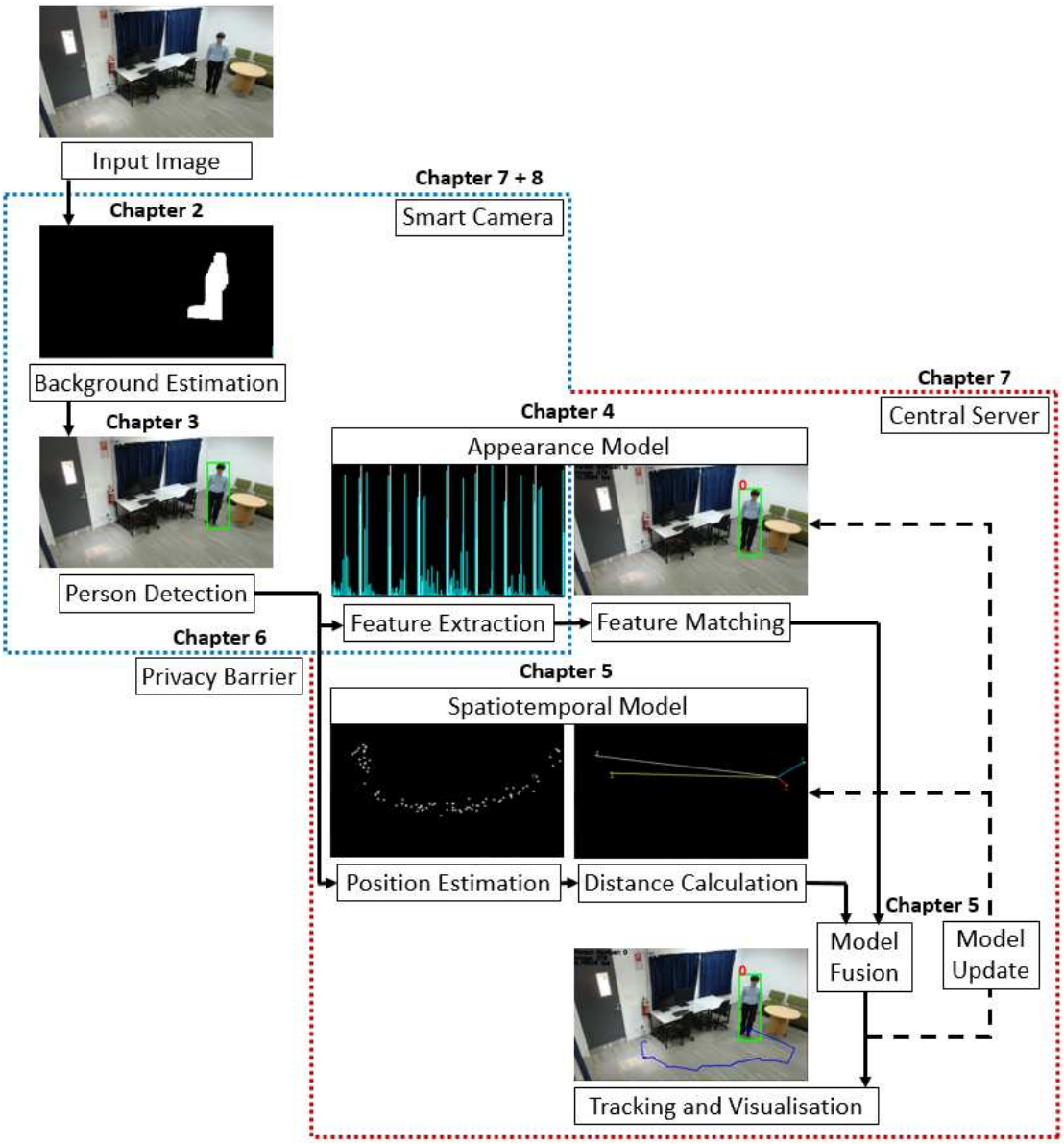

Figure 1.1: The image processing pipeline used in the proposed person tracking approach, with the separation of responsibilities between the smart camera (blue) and the central co-ordinating server (red) indicated

In [39], new metrics for measuring the accuracy of person tracking systems are proposed, reflecting a change in the priorities within the academic community. Large scale datasets have also become more common, including MARS [40] and Airport ALERT [41], taken from real-world environments that provide more data for training complicated models as well as more challenging cases for evaluating the robustness of developed systems.

1.5 The Proposed Image Processing Pipeline

Human vision uses two layers of processing; the image is received by the eye, where initial processing such as colouring, filtering, contrasting, simple edge detection, and some movement/blur detection occurs, and then that information is further processed in the brain to detect and recognise objects, predict motion, and learn. To emulate this, the effort must be focused in two areas: the mathematical algorithms that process image data into meaningful information, and the hardware platforms and architectures that process the large amount of data in real-time and communicate that to other systems as necessary. The integration of these systems requires a multidisciplinary approach in order to build real-world systems.

In this thesis, the main focus is on presenting the different parts of a person tracking system developed as part of the Doctoral degree. In this approach, a number of pipeline modules are used for background estimation, person detection, feature extraction, position estimation, classification, model fusion, model updating, and overall tracking as shown in Figure 1.1, with the corresponding thesis Chapters labelled. Many person tracking systems in the literature use either spatio-temporal/position-based algorithms or appearance-based/person re-identification algorithms, but rarely both together. In more recent works [38, 39], single-view tracking (i.e. within the same camera view) is treated as a spatio-temporal tracking problem to create "tracklets" that have no notion of identity, and then appearance-based person re-identification is used to match identities across multiple cameras (i.e. is the person seen in this camera view the same as previously seen in another camera view) and connect tracklets together. In contrast, the proposed pipeline here brings these two approaches together to help inform each other, applying both a spatio-temporal model and an appearance-based model to each camera as a single-view classification problem, and using a data/model fusion step to improve the accuracy of that classification through re-ranking. This can then be easily extended to a multi-camera context by placing each of the cameras in a global map.

Even though there are potential accuracy losses at each stage, a modular approach allows for better transparency of the system and understanding of what is actually happening in the system rather than treating the system as a black box that opaquely produces correct outputs. This is important for allowing improvements to be made to the system because developers can swap modules as new approaches for each pipeline stage evolve. The modular approach means that it is not dependent on any one solution in a problem domain, and is more transparent than an end-to-end CNN approach. Development on these modules could also be conducted in parallel between multiple developers. Additionally, the modular approach lends itself towards an eventual real-world system that leverages smart cameras for distributed computing, where simpler tasks at the beginning of the pipeline can be completed at the point of image capture and more computationally expensive processing can be done at a central server or in the cloud. The modular approach allows this partitioning to be done more easily between smart cameras that can do some processing at the point of image capture and a more powerful central server; the proposed separation of responsibilities is also shown in Figure 1.1. This reduces the computation load on a central server which may have to match features with a large database,

supporting scalability towards large-scale camera systems.

A significant practical challenge for video analytics is the need for networking infrastructure that can support large-scale camera networks. A huge amount of bandwidth is required to stream data from the camera nodes, and synchronising this data is a non-trivial task. The proposed pipeline, when combined with the privacy-affirming architecture presented in Chapter 6, has the additional advantage of reducing the amount of network bandwidth consumption by only sending feature vectors that describe people to the central server rather than sending entire images, avoiding saturation of the network communication channels. This is a different approach to only sending the images when something interesting is happening [42] or increasing the bandwidth to allow for more data transfer. The privacy-affirming architecture also helps reduce the privacy loss experienced by those being observed by the cameras, which will be explained later in the thesis. This pipeline can also be applied in single-view or multi-camera contexts, with a central controller co-ordinating the camera views and data.

1.6 Research Contributions

The work completed for this thesis brings together methods from multiple fields to create a complete video analytics system for person tracking. There are a number of key research contributions:

- The development of three new algorithms or application of algorithms in new contexts:
  - SuperBE for background estimation (Chapter 2), which is significantly faster than many state-of-the-art methods while achieving comparable accuracy.
  - Sequential k-Means for one-shot person re-identification (Chapter 4), which is competitive against other unsupervised or one-shot methods.
  - A model fusion algorithm (Chapter 5) that combines a linear weighting approach with a decay function and rule-based system to cover corner cases effectively.
- The development of a person tracking and re-identification dataset (and associated annotation tool) called UoA-Indoor that can be used for evaluating the performance of the person tracking system (Chapter 4).
- Experimental analysis that demonstrates that the covariance metric is the most suitable metric learning approach where generalisability is required in a person re-identification context, superior to many more modern and complex choices (Chapter 4).
- Using model fusion to combine spatio-temporal and appearance-based models to improve classification accuracy for person tracking across multiple cameras (Chapter 5).
- Research into public perceptions of privacy through the use of surveillance camera scenarios, which reveal important factors that should be considered during system design (Chapter 6).
- The design of a system architecture that jointly helps protect the privacy of individuals being preserved (Chapter 6) and allows for distributed computing (Chapter 7).

- Full implementation of the proposed video analytics system, including on embedded computing resources with smart cameras (Chapter 7).
- Demonstrating the application of Hardware/Software Co-Design techniques to image processing applications, achieving large speed-ups in embedded systems without significant degradation of accuracy (Chapter 8).

1.7 Thesis Outline

The rest of the thesis is laid out as follows:

- In Chapter 2, the background estimation module is presented. This uses a novel algorithm named SuperBE (Superpixel-based Background Estimation), and presents detailed evaluation of this algorithm in comparison to the state-of-the-art.
- In Chapter 3, person detection algorithms are analysed, with a comprehensive literature review summarising the efforts in this space, and comparing two particular approaches, the Deformable Parts Model (DPM) and Aggregate Channel Features (ACF). Additionally, feature selection and extraction from the detected people are also presented here.
- In Chapter 4, the appearance model is presented, which includes dimensionality reduction and metric learning to improve separability, and an analysis of different classification methods that can be used to determine the identity of an observed individual for re-identification between multiple camera views.
- In Chapter 5, the spatio-temporal model is presented, which includes camera calibration and mapping to convert from image space co-ordinates to real-world co-ordinates, and position estimation using Kalman filters. The model fusion of the appearance and spatio-temporal models is also presented in detail, along with experimental results showing the speed and accuracy of the complete person tracking system.
- In Chapter 6, the topic of ethical video analytics and person tracking is analysed, including through the literature, interviews with subject experts and government officials, and a survey on public perceptions of privacy, leading to the proposal of the privacy-affirming architecture that uses computer vision to help protect privacy by default.
- In Chapter 7, the thesis returns to the person tracking pipeline and implements it with the privacy-affirming architecture using distributed computing approaches. The effect of using embedded smart cameras is discussed, including the impacts on accuracy, speed, control, and networking.
- In Chapter 8, some work looking into the future of embedded vision is presented, using Hardware/Software Co-Design techniques to accelerate computation with constrained resources. This is applied to the SuperBE background estimation algorithm, as a first step towards accelerating this person tracking pipeline further.
- In Chapter 9, conclusions are presented, and avenues for future work to improve the presented video analytics system are discussed.

Chapter 2

Background Estimation

Many computer vision problems are focused on the detection, tracking, and identification of objects in an image or video sequence. In most cases, passing an entire image through a slow algorithm looking for positive cases is wasteful, because the majority of the image yields negatives. Background estimation is a common method for classifying parts of the image as background and foreground, so that the background can be ignored in subsequent processing stages. When implemented appropriately, the additional cost of a background estimation algorithm can be significantly less than the cost of parsing irrelevant parts of an image with more expensive algorithms. This is increasingly important as we move towards processing higher resolution images that have more pixels. Additionally, as devices such as smart cameras become more popular, the viability of executing these complicated algorithms on embedded systems with constrained resources must be considered. In many applications real-time execution is critical, rendering highly accurate but slow and complex algorithms inappropriate.

Most background estimation methods are motion-based, meaning that they rely on the movement of objects between frames of a video sequence in order to identify the foreground. The general assumption is that static or slow-moving objects should be classified as background and masked out of the image. Common challenges include adjusting to slow-changing environmental conditions such as lighting, reducing susceptibility to noise, identifying false negative holes within objects, rejecting periodically-moving objects like tree branches in the wind, and avoiding the formation of "ghosts" when static objects in the background model move and cause the originating region to be erroneously classified as foreground.

The majority of background estimation methods are pixel-based, meaning that they develop background models for every pixel in the image and identify movement at the pixel-level. However, objects of interest are generally represented by more than one pixel. Biological vision does not operate at the pixel-level; the brain segments images into general shapes and then groups those shapes together to form objects. Some more recent approaches [44, 45, 46, 47, 48] have included the use of superpixels: clusters of pixels grouped together based on their perceptual similarity, resulting in groups of pixels with spatial and colour coherency. Operating at the superpixel-level could improve the accuracy of background estimation by classifying

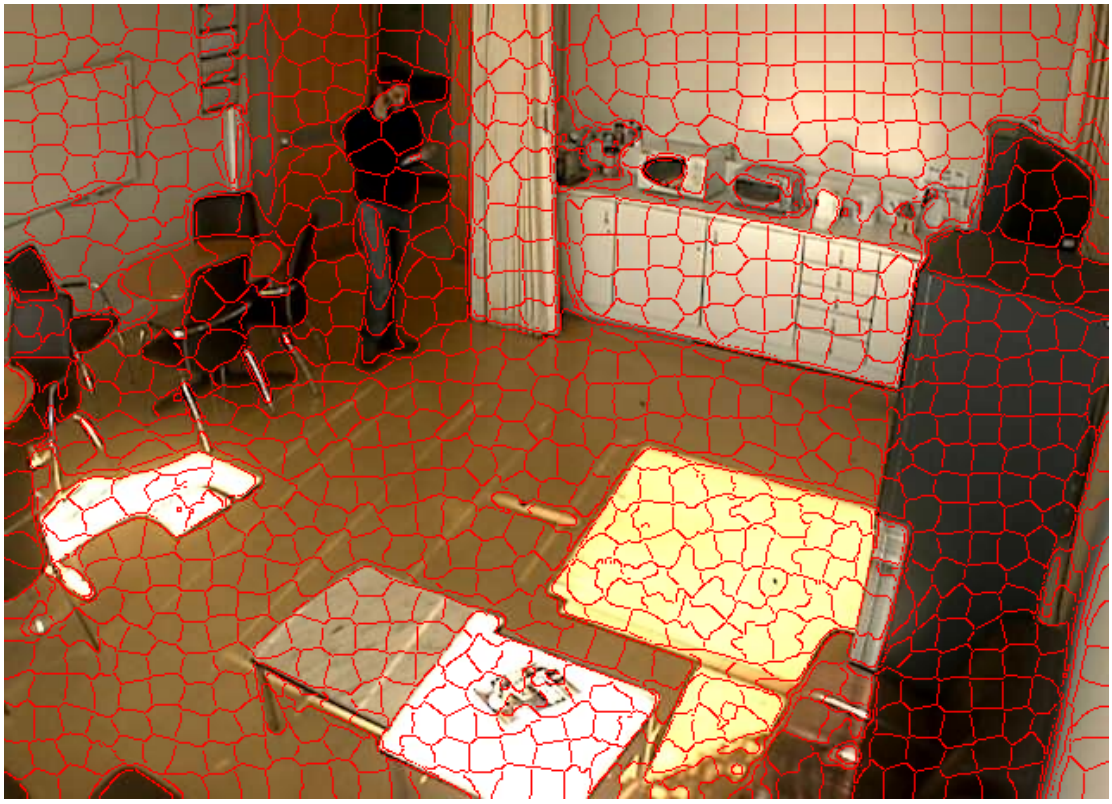

Figure 2.1: A frame from HDA+ [43] segmented with SLIC.

objects rather than just pixels, and can significantly reduce the computational complexity and memory requirements for background estimation by reducing the number of elements in the background model. For example, in a 600x480 image, a pixel-level algorithm must maintain 288,000 elements in the background model (one for each pixel) but in a superpixel-level algorithm the image may be sufficiently represented by 500-2000 elements; two to three orders of magnitude less. These savings in computation and memory costs must be balanced against the added cost of superpixel clustering, which may require multiple iterations to converge.

A lightweight yet accurate background estimation algorithm would be of great utility for embedded applications that operate with constrained resources. In the context of the proposed person tracking pipeline, using some form of background estimation can significantly reduce the search space for the much more expensive person detection algorithm, and thus reduce computation costs. This Chapter presents in detail a novel background estimation algorithm called SuperBE, which aims to balance high accuracy with low computation cost. The primary contribution is the combination of a superpixel-level approach with a simple adaptive background modelling and updating scheme, suitable for video sequences with static camera views. Impacts of the choice between fixed and adaptive superpixels are also discussed. Additionally, related literature and existing methods are explored, specifically relating to superpixel-based background estimation and computationally efficient background estimation.

2.1 Related Works

Background estimation, also known as background subtraction, background modelling, or foreground detection, is a common way of segmenting an image so that objects of interest can be isolated [49]. In a person tracking / surveillance scenario, individuals are identified in images captured from (normally) static cameras. Since the environment is relatively static (i.e. the walls and ground do not move), a background model can be developed and removed from each frame, leaving only foreground objects, such as a person walking through the scene. Background estimation is a popular tool in person detection applications as an initial step in order to reduce overall computation time, as demonstrated in [19, 50, 51].

However, a static background model is generally insufficient; even with indoor settings, lighting conditions and reflections can change throughout the day [52, 53], new features could be introduced to the environment e.g. posters, and background objects such as tables and chairs may be moved. There are many approaches for adapting to these changes, including linear filtering (averaging the last N frames) [19, 20], density estimation (predicting future frames based on previous frames) [22], and machine learning approaches (using a large training set to classify likely frames based on previous frames) [54, 55, 56]. Some of these approaches attempt to solve additional challenges such as people standing still in conversation (which might cause them to become a part of the background) [57]. For readers interested in further details, recent surveys of the field cover a large variety of background modelling/subtraction techniques [49, 58, 59].

One of the state-of-the-art pixel-wise background subtraction algorithms is ViBe [60]. It turned many of the long-held assumptions about background estimation upside down by using a non-deterministic update model that randomly replaces samples in the background model. This allows a smooth decaying lifespan of the samples compatible with any framerate and reduces the number of samples required. Neighbouring pixels are also used to contribute to each pixel's background model, helping ensure spatial consistency and coherency. At the time, ViBe achieved significantly better results than existing algorithms, and motivated the development of a new class of non-deterministic, non-parametric algorithms such as PBAS [61]. The proposed algorithm, SuperBE, uses a similar background and updating model and applies it to superpixels.

A superpixel is a cluster of pixels grouped together for spatial and colour coherency, essentially segmenting the image and forming superpixel borders along object boundaries. They are increasingly popular in computer vision because they significantly reduce the amount of processing required in later processing stages, while also reducing memory requirements. A number of different clustering algorithms have been applied to generating superpixels, such as mean shift and watershed, but more recently, superpixel-specific algorithms such as SLIC [62] and SNIC [63] have proven to be particularly effective, as shown in Figure 2.1. Using superpixels, object shapes can be more accurately identified and boundaries more clearly defined. In [64], colour covariance matrix manifolds are used in addition to colours to isolate foreground objects for image segmentation. In [48], superpixels are tracked between frames in order to segment

objects in videos.

However, many implementations of superpixels in background modelling or object detection are very computationally expensive, optimising for accuracy without consideration for real-time systems or embedded implementations with limited resources. Some papers report computation times in the order of seconds per frame, such as [46], a background subtraction method using superpixels and integrated motion estimation, which states that their method requires approximately 6 seconds to process one frame on a standard laptop. The more computationally expensive algorithms generally involve many dependent stages, such as [65], an object segmentation algorithm that uses superpixels to track objects in unconstrained video, which requires optical flow ($<$1 sec/frame), superpixel formation (1.5 sec/frame), and then object segmentation (0.5 sec/frame) for a total of approximately 2-3 seconds per frame. SuperBE simplifies the use of superpixels greatly in order to significantly reduce the required processing time so that a high fps can be achieved.

Additionally, some papers do not report timing or computational complexity at all [66], which makes it very difficult to compare these approaches on their ability to operate in real-time. Differing methods of reporting computation time can make fair comparison challenging, such as [47], a superpixel-based matrix decomposition method for background estimation, which reports computation times for batch processing 100 frames at only two image resolutions. Variation in the computing platforms can also give misleading impressions of the relative speeds of algorithms; for example, [56], a deep-learning based background subtraction method, reports background subtraction times from a Titan X GPU, which is not going to be a fair comparison against times obtained from a resource-constrained embedded computing platform. In this Chapter, computation time is reported for a wide range of image resolutions in frames per second, and details of the execution platform and dataset are provided for fair future comparison.

2.2 Algorithm Description

This Section discusses the two main stages of the algorithm: clustering superpixels and initialising the background model as shown in Figure 2.2, and classifying the superpixels and updating the background model as shown in Figure 2.3. These diagrams show the simple approach taken by SuperBE, without requiring computationally expensive online learning or search/tree-based approaches that become unsuitable for resource-constrained embedded implementations.

2.2.1 Superpixel Clustering

Images are made up of pixels, which provide a way of discretising the visual space into units that can be described numerically, whether in terms of intensity or colour. It is generally desirable from a consumer perspective to have more pixels in an image where possible, because it provides more information to describe the image and therefore allows for a clearer, sharper image that more closely matches what we might expect from human vision. However, in the computer

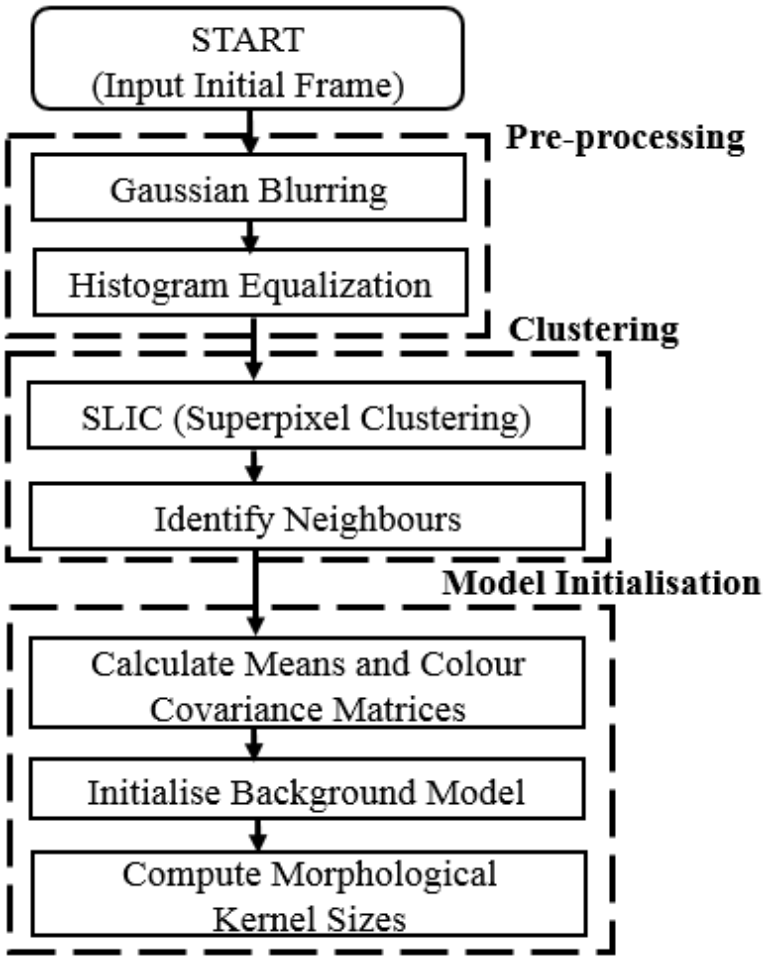

Figure 2.2: Top-level flowchart of SuperBE background model initialisation.

vision context there are often two problems: having more pixels increases the computation cost of processing that image because there is more information, and that additional information may not actually significantly improve the accuracy of algorithms. The majority of historical image processing research has worked on what would now be considered low-resolution images in the marketplace; this is reflected in the image resolutions of many image and video datasets, such as SegTrack, which uses images with resolutions of 640x360 or less [67].

By using superpixel clustering, the amount of information in the image can be reduced in a way that retains the general distribution of the image, reducing information redundancy, reducing the effects of high frequency noise, and reducing the amount of information to be processed by later algorithms. Using a superpixel algorithm leads to a significant improvement in comparison to drawing a static grid over the image because the object boundaries are identified more accurately, improving the accuracy of the background classification between frames. In theory, SuperBE can use any superpixel clustering algorithm, with the assumption that the clustering results in oversegmented images where objects are comprised of multiple superpixels. This is because describing an object with a single superpixel generally oversimplifies the object and reduces the accuracy of the descriptor. Ideally, the superpixel size should be set

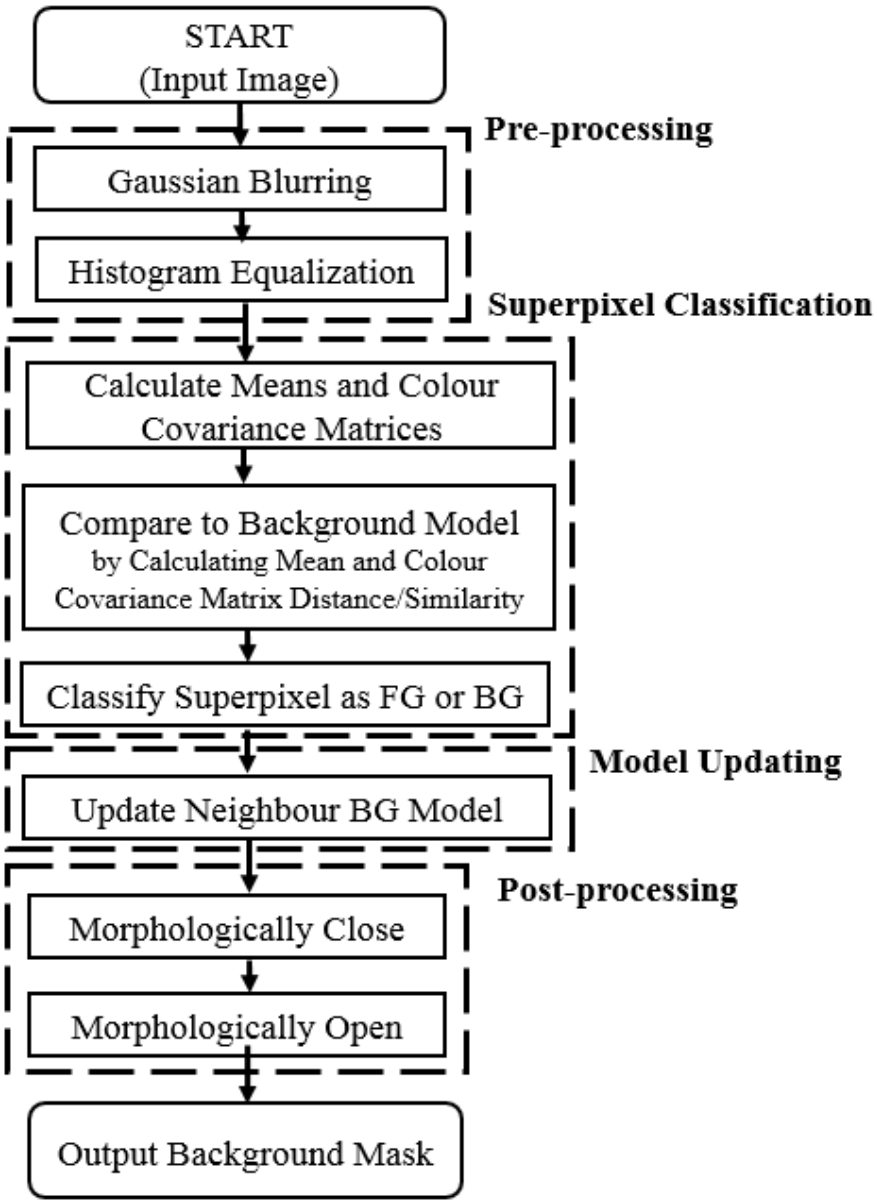

Figure 2.3: Top-level flowchart of the SuperBE algorithm when processing frames.

to oversegment the smallest object to be detected. However, this cannot always be known at configuration-time, and is particularly difficult for camera views which may have perspective distortion, causing objects to appear smaller when they are further away from the camera, such as in long hallways or traffic cameras at low angles.

The SLIC superpixel clustering algorithm [62] was investigated because of its computational efficiency in comparison to other superpixel algorithms, and its ability to set a target number of superpixels in the image. Ultimately, SLICO was selected, a parameter-less version of the algorithm, which determines the number of superpixels automatically, primarily based on the resolution of the image. SLIC and SLICO work well assuming that the objects to be detected are reasonably large in the image and are described by multiple superpixels. Choosing

between SLIC and SLICO depends on whether the size of the objects to be detected is known at configuration-time and if the developer can select an appropriate target number of superpixels for the video sequence; SLICO is more appropriate when this is not possible.

Initially, each frame was passed through SLIC/SLICO, which labelled each pixel with a segment number. However, because each frame had independently clustered superpixels, the number of superpixels could change between frames, making it difficult to track the superpixels in the background model. The superpixels could also spatially "move" between frames, such that a superpixel in one frame corresponding to an object could move with the object in the next frame to a new spatial location, which complicates the process of keeping track of past sample values for each superpixel. Additionally, during preliminary execution profiling, it was found that the SLIC algorithm was responsible for 30% of the total execution time. For these reasons, it was decided that superpixel clustering should only be performed once, during background initialisation, resulting in a significant speed improvement when processing later frames. The superpixel segmentation is then stored and used for all future frames. At a pixel level, this results in more false positives as the bounds of detected objects are not as accurate, but the overall accuracy of the algorithm improved on all test sequences as the superpixels were tracked correctly and foreground objects were correctly classified more often. While the exact bounds of the objects may not be as tight as if superpixels were re-clustered for every frame, relatively good bounds are still achieved at the object-level for determining where the object is. This approach may not be as suitable for fine-grained object segmentation tasks, but satisfies the main objective of identifying the region(s) of interest for the purposes of reducing computation times for subsequent processing steps. Using superpixels was still more accurate than a simple rectangular grid because the background objects were segmented more accurately. In some applications, it could be appropriate to re-initialise the superpixel grid, for example after large-scale lighting changes or on a periodic basis to help reset accrued errors.

2.2.2 Background Model Initialisation

SuperBE uses a similar background model and update procedure to ViBe [60] due to its combination of high accuracy and relatively low computation cost, but has applied it at the superpixel level with some modifications. This modelling scheme has a number of advantages: the background model can be initialised from a single frame, samples in the background model have an exponentially decaying lifespan that is not dependent on the video frame rate, and neighbouring models are used to improve spatial consistency between pixels or superpixels.

In ViBe, when initialising the algorithm, the background model is based on neighbouring pixel values in order to ensure temporal and spatial consistency. However, in SuperBE the spatial consistency is already taken into account when clustering superpixels. The modelling process was adjusted by including the current superpixel itself when considering neighbours, so that the background model for each superpixel includes a self-value, rather than exclusively being filled with neighbouring superpixel values. Selection of the initialisation frame can therefore be quite important; for example, if there is a foreground object in the initialisation

frame, then this can cause a "ghost" [68] to appear once the foreground object starts moving and uncovers the actual background. However, the primary advantage of this method is that it allows the background model to be initialised from a single frame. This means that for a real-world embedded system, the system could be re-initialised when certain conditions are triggered, such as a significant change in illumination conditions when a light is turned on, and can help remove the impact of errors that may accrue over time. This is in contrast to many existing methods that require multiple frames for initialisation such as [69, 70], interrupting the video processing and potentially violating safety-critical continuous operation conditions.

2.2.3 Superpixel Classification

Selecting appropriate features for describing superpixels with sufficient discriminative strength is an active area of research with many proposed options. In the interest of developing a computationally light algorithm, the superpixels in SuperBE are described by using the pixels within each superpixel to calculate the arithmetic mean of each colour channel and a simple 3x3 colour covariance matrix. This is much lighter than related literature such as [71] which uses 11x11 covariance matrices, mapping the image to 11 feature dimensions. The results (in Section 2.3.2) show that three dimensions may be sufficient as the accuracy is high. By describing the mean and variance of the colours inside each superpixel the distributions can be identified, and compared against another superpixel to determine if they are similar. These two features are relatively quick to calculate, and the colour covariance matrix in particular has been shown to have high discriminative strength [66].

In order to determine if a superpixel should be considered part of the background or foreground, SuperBE considers its current similarity to past superpixels stored in the background model. In most pixel-level algorithms, this means that multiple past images have to be stored, consuming a large amount of memory space. However, SuperBE only needs to store the colour means and colour covariance matrices for each superpixel. This not only reduces the amount of memory consumption, but also the number of memory transactions, which helps reduce the computation time of the algorithm. Comparing the colour mean is simple using the Euclidean distance:

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_3 - q_3)^2} \tag{2.1}$$

where p is the vector representing the means of the three colour channels of the current superpixel, q is the vector with the past means of that superpixel, and $d(p,q)$ is the distance between the two vectors.

Comparing the colour covariance matrices is more difficult as the matrices are not in Euclidean space. Since all of the covariance matrices are symmetric and the same size, the similarity calculation is generally analogous to a Euclidean distance metric applied to matrices. Traditionally, the Förstner and Moonen [72] metric is used for covariance matrix similiarity [71, 66]. In contrast, SuperBE uses a more recent measurement called the Jensen-Bregman LogDet Divergence [73]:

$$J(X,Y) = log \mid \frac{(X+Y)}{2} \mid -\frac{1}{2} log \mid XY \mid \tag{2.2}$$

where X and Y are two covariance matrices, $|\cdot|$ denotes the determinant, and $J(X, Y)$ is the matrix divergence, which can be used as a measure of (dis)similarity, where the larger the value the more dissimilar the two covariance matrices are. Additionally, the absolute value of this measure is used since only the magnitude of the distance between the covariance matrices is needed. This measure is chosen primarily for its lower computational complexity, avoiding any linear algebra eigen calculations. If both the Euclidean distance of the colour means and dissimilarity of the colour covariance matrices are below certain thresholds (discussed in Section 2.3.1), then the superpixel is considered to be similar to the past superpixel. In this implementation of the algorithm, floating point numbers are used to increase accuracy - it could be possible to develop an integer-only version of this algorithm with careful consideration of the covariance matrix similarity calculation.

2.2.4 Updating the Background Model

The background model is made up of multiple samples for each superpixel. For each frame, each superpixel is compared to the past superpixels in the background model, based on its colour means and colour covariance matrix. If the current superpixel is similar enough to sufficient background model samples, then the superpixel is classified as part of the background. This operation therefore depends on two parameters: the number of historical samples maintained in the background model for each superpixel (B), and the threshold number of past samples that need to be matched for the superpixel to be classifed as background (numMin). It is important to balance these two parameters. If the ratio of B to numMin is too low, then it can be very easy for false negatives to occur, where background superpixels are incorrectly classified as foreground, because the threshold for background classification is too high. However, if the ratio of B to numMin is too high, then noise can more easily become false positives. Increasing the number of samples in the background model generally improves the accuracy of the algorithm. However, increasing B increases the memory and computation requirements because there are more samples to store and compare, so a balance must be found. In order to reduce the computation time, like ViBe [60], SuperBE iterates through the background model and compares each sample to the current superpixel, and as soon as enough samples have been matched, it breaks out of the loop and does not compare the rest of the background samples.

When a superpixel has been classified as background, the current superpixel's background model is updated by replacing a uniformly random sample from the background model with the new mean and colour covariance matrix values. This non-deterministic update scheme allows for a smooth decaying lifespan, similar to natural processes like radioactive decay. The advantage of this approach is that this decay allows the update mechanism to adapt to different video frame rates. This is also a conservative update policy that never includes foreground samples in the background model in order to ensure the integrity of the background model. However, there may still be erroneous classifications (such as ghosts that permanently become part of the background model). To address this, a random neighbour may then also be selected based on a neighbour subsampling rate (phi), and a random sample of its background model is

Figure 2.4: A frame from the CDW2014 *highway* sequence processed with SuperBE, before and after post-processing steps.

also replaced by the current superpixel's colour mean and covariance matrices, thus "invading" the neighbouring superpixel. SuperBE builds a list of neighbours for each superpixel by looking at the superpixel boundaries and identifying adjacent superpixels. This update procedure helps ensure that the background model remains accurate over time, adaptively responding to low frequency changes in a video such as lighting changes throughout the day. It can also respond to large or sudden step changes (for example, a curtain being opened exposing more light into the room), but does so much more slowly, and reinitialisation of the background model may be more accurate in these cases.

If the current superpixel is sufficiently different to all of the background model samples, then the superpixel is classified as part of the foreground, and the area bounded by the superpixel boundaries is set to 1 (or 255 in greyscale) in a segmentation mask. Optionally, morphological operations can then be used in a post-processing step to fill in any holes (false negatives) between foreground superpixels, and also remove isolated noisy superpixels (false positives) as shown in Figure 2.4. The number of superpixels in relation to the size of the object(s) of interest can make a large difference to the post-processing. Generally, images should be oversegmented, i.e. the object of interest is represented by more than one superpixel, so that changes in lighting or colour across the object are accounted for and a more well-defined object boundary can be identified. When there are more superpixels, then the combined area of the incorrectly classified superpixels decreases, and morphological operations based on a fixed kernel size could cause more pixels to be incorrectly classified. To address this, the morphological operation kernels were parametrised based on the superpixel size. By using SLICO instead of SLIC, the potential for a human user to incorrectly set the target number of superpixels to be too high or too low was also removed. The post-processing step can have a large impact on the overall accuracy of the algorithm, but as shown in Section 2.3.2, even without post-processing SuperBE is comparable to state-of-the-art background estimation algorithms. Post-processing does come at a cost of significantly higher computation times, also reported in the results.

2.3 Experimental Results

To test the performance of the algorithm, the Change Detection (CDnet) 2014 dataset [74] (also referred to as the CDW2014 dataset) was used, which is comprised of 51 video sequences across

11 categories and includes human-annotated ground truth for most sequences. This collection of sequences represents a wide range of scenarios, including low-light conditions, complex scenes with many moving objects, varying frame rates, and different image resolutions. SuperBE was prototyped and tested in Python using OpenCV [75] and associated libraries such as NumPy in order to help reduce development time. Once the algorithm showed acceptable accuracy results, it was ported to C++ (with floating-point numbers) to help improve computation time. Interestingly, this led to a speed-up of at least 5x. The following subsections present discussion on the parameter selection, and show the comparative performance of SuperBE in terms of accuracy and computation time.

2.3.1 Parameter Selection

There are competing interests when performing parameter selection; the parameters that lead to the best accuracy do not necessarily result in the best computation time. Additionally, the Change Detection Workshop Challenge rules require that "only one set of tuning parameters be used for all videos" [74], even though different videos have varying characteristics and would perform better with optimised parameters. In order to balance between these interests, appropriate bounds were set for the parameters, and the parameter space was uniformly subsampled. A supercomputing cluster was used to parallelise the tests; data was collected for over 2000 parameter combinations. This allowed a set of parameter values with the best overall performance across all categories to be found. There are five parameters to set:
B = the number of historical samples in the background model for each superpixel
R = the threshold for the Euclidean distance of the colour means between the current and background model superpixel
DIS = the threshold for the colour covariance matrix dissimilarity between the current and background model superpixel
numMin = the number of similar samples required for the current superpixel to be classified as part of the background
phi(ϕ) = the random subsampling rate for updating neighbouring background models

Other than DIS, these parameter definitions are the same as in ViBe [60] because they relate to the background model and updating scheme. Increasing the number of samples in the background model improves accuracy, but with diminishing returns. Since larger values of B incur larger memory consumption and computational costs, it is desirable to keep the value of B low. While ViBe uses R=20 for their pixel-based approach, the difference in colour mean distance for superpixels needed to be substantially higher, around R=60. This is theoretically consistent with the fact that superpixels are made up of multiple pixels that can include a larger range of values and more noise, so a higher threshold may be needed to distinguish between background and foreground. The DIS parameter is unique to SuperBE since it is used for the colour covariance matrices. The appropriate value for this can depend on how the superpixels are formed and how the dissimilarity is calculated; with the combination of SLICO and the Jensen-Bregman LogDet Divergence, the optimal value was generally around to 16-20. For both

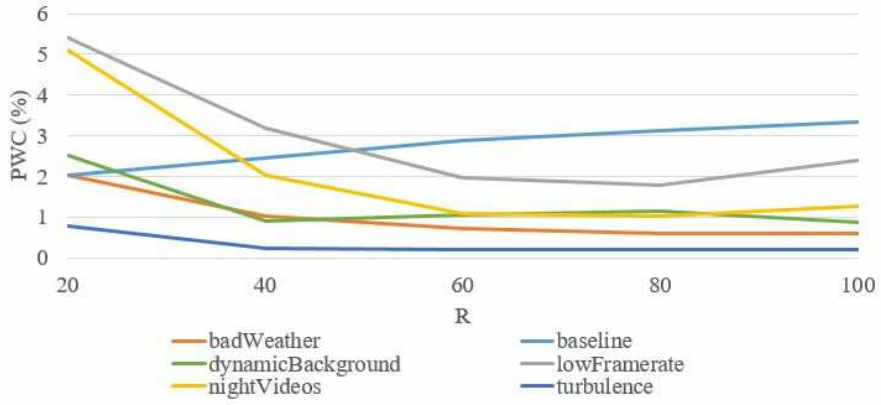

Figure 2.5: PWC (Percentage of Wrong Classifications) for varying values of R across six CDW2014 categories with all other parameters kept constant

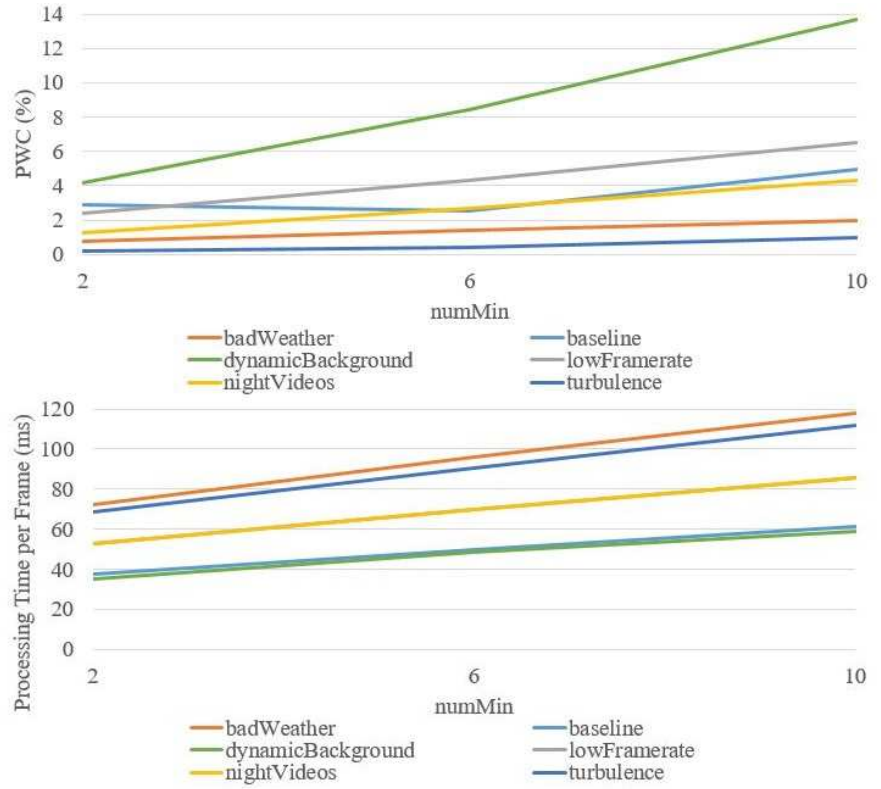

Figure 2.6: PWC and Computation Time per Frame for varying values of numMin across six CDW2014 categories with all other parameters kept constant

R and DIS, the optimal value in terms of accuracy would be strongly dependent on the scene / type of video sequence and the difference in colour between the background and foreground; this is shown in Figure 2.5 where different categories have very different optimal R thresholds.

The largest determinant of computation time is the image size; larger images have more data to process and thus take longer, as shown later in Section 2.3.3. Then numMin has the next largest impact on computation time, as shown in Figure 2.6, because it determines how many comparisons must be done before the algorithm can classify a background superpixel. For

example, with numMin=10, SuperBE would compare the current superpixel to ten background model samples before being able to declare the current superpixel to be part of the background. Increasing numMin therefore has a linear effect on the computation time and should be kept as low as possible. In most cases, a numMin of 2 is sufficient; increasing numMin generally increases the error as well, since it is harder for an erroneous foreground superpixel to become correctly classified as background, which results in more false positives. Lastly, a lower phi value would mean that, in theory, neighbours are updated more often which means that errors can disappear faster at the cost of more memory assignments/transactions. However, it is important to note that a low value of phi at 0 or 1 increases the incidence of "ghosts", because the superpixel values that should be part of the foreground become part of the background model of neighbouring superpixels very easily, replacing legitimate background samples with erroneous foreground samples, especially in slow-moving video sequences. In some sequences a higher phi is desirable to avoid slow-moving objects from becoming incorrectly classified as background, while in sequences with sudden changes in environmental conditions a lower phi is desirable to erode away false positives such as a patch of light. A lower phi would have a higher computation cost, although this is generally negligible.

Importantly, the number of superpixels in the image is not a parameter, as it is left to SLICO to determine an appropriate number of superpixels for the image based on the background objects and the image size. In early testing when using SLIC, the target number of superpixels for a 320x240 image was around 1500. This produces superpixels that are a similar size to [48], which uses fixed 7x7 pixel-sized superpixels. For the same image resolution, SLICO returns around 750 superpixels, leading to a lower computation cost but slightly lower accuracy too. If the developer chooses to use SLIC instead of SLICO, then the selection of the number of target segments is important; the ratio between the size of the image and the size of the object(s) of interest can be used to determine a good value (if the size of the object(s) of interest can be known at configuration-time).

2.3.2 Comparative Accuracy

In this Subsection, a number of statistics are reported for most of the CDW2014 categories. For the purposes of calculating the statistics, a correct background classification is a True Negative (TN), and a correct foreground classification is a True Positive (TP). The seven metrics are defined in Table 2.1. Algorithms aim for high Rec, Spec, FMeas, and Prec, and low FPR, FNR, and PWC. All metrics are out of 1, except for PWC which is out of 100.

This analysis mostly focuses on the PWC (the complement of Percentage of Correct Classifications (PCC)) because it indicates the overall proportion of pixels that are classified incorrectly regardless of whether they should be background or foreground. Five algorithms have been selected from the results reported on the CDW website for comparison purposes. One is a Gaussian Mixture Model (GMM) [76], which is possibly the most popular probabilistic method due to its simplicity and high speed. The other four are more recent approaches from the last three years. FTSG [77] fuses a pixel-level model, a flux tensor model, and a split Gaussian

Table 2.1: Accuracy Metrics

Recall (Rec)	$\frac{TP}{(TP + FN)}$	Foreground classification accuracy
Specificity (Spec)	$\frac{TN}{(TN + FP)}$	Background classification accuracy
False Positive Rate (FPR)	$\frac{FP}{(FP + TN)}$	Foreground error rate
False Negative Rate (FNR)	$\frac{FN}{(FN + TP)}$	Background error rate
Percentage of Wrong Classifications (PWC)	$\frac{100 * (FN + FP)}{(TP + FN + FP + TN)}$	Overall error rate
F-Measure (FMeas)	$\frac{(2 * Prec * Rec)}{(Prec + Rec)}$	Overall accuracy measure
Precision (Prec)	$\frac{TP}{(TP + FP)}$	Overall accuracy measure

model with a rule-based system. SuBSENSE [78] is a pixel-level segmentation method that uses spatio-temporal and colour features as well as a feedback loop to dynamically adjust internal parameters. PAWCS [79] is similar to SuBSENSE, but uses a word-based approach instead of individual pixels. Lastly, DeepBS [56] is a supervised learning approach using a deep convolutional neural network, which should be interpreted with caution as it has the advantage of being able to use dataset frames for each sequence for training, unlike the other algorithms. These four algorithms are all ranked within the top ten in terms of overall accuracy in the CDW2014 dataset. For further comparison against other methods, please refer to the results published on the `changedetection.net` website. The purpose of presenting these results is not necessarily to show that SuperBE has the best accuracy, but that its accuracy is close enough to state-of-the-art approaches, as the focus of this work is to reduce the computation cost. For these results in all categories, B=30, R=60, DIS=16, numMin=2, and phi=16, which were determined via parameter sweep optimising for PWC to lead to good results across all categories, although there are better category-specific parameters that would lead to higher levels of accuracy. For example, during the limited parameter sweep, SuperBE was able to achieve a 1.7925 PWC on the baseline category without post-processing, in comparison to the 3.1053 PWC reported in Table 2.2.

Tables 2.2 and 2.3 show that in most categories, SuperBE outperforms the GMM approach, and is usually within one percentage point of the four state-of-the-art approaches in terms of PWC, if not better. It is important to interpret the results in the context that SuperBE is also computationally lighter than these competing implementations (as shown later in Section 2.3.3), so a small compromise in accuracy should be acceptable as a trade-off. In particular, SuperBE appears to perform extremely well in comparison to all other methods in the *nightVideos*

Table 2.2: Comparative Accuracy on the CDW2014 Dataset

	Category: **badWeather**						
Method	Rec	Spec	FPR	FNR	PWC	FMeas	Prec
SuperBE (no post-processing)	0.8463	0.9994	0.0006	0.1537	0.4727	0.6823	0.7518
SuperBE (post-processing)	0.8640	0.9990	0.0010	0.1360	0.4026	0.8389	0.9360
GMM — Zivkovic	0.6863	0.9978	0.0022	0.3137	0.7707	0.7406	0.8138
FTSG	0.7457	0.9991	0.0009	0.2543	0.5109	0.8228	0.9231
SuBSENSE	0.8213	0.9989	0.0011	0.1787	0.4527	0.8619	0.9091
PAWCS	0.7181	0.9994	0.0006	0.2819	0.5319	0.8152	0.9474
DeepBS	0.7517	0.9996	0.0004	0.2483	0.3784	0.8301	0.9677
	Category: **baseline**						
Method	Rec	Spec	FPR	FNR	PWC	FMeas	Prec
SuperBE (no post-processing)	0.3278	0.9979	0.0021	0.6722	3.1053	0.3690	0.6969
SuperBE (post-processing)	0.3646	0.9970	0.0030	0.6354	2.9444	0.3841	0.8750
GMM — Zivkovic	0.6604	0.9725	0.0275	0.3396	3.9953	0.5566	0.5973
FTSG	0.9513	0.9975	0.0025	0.0487	0.4766	0.9330	0.9170
SuBSENSE	0.9520	0.9982	0.0018	0.0480	0.3574	0.9503	0.9495
PAWCS	0.9408	0.9980	0.0020	0.0592	0.4491	0.9397	0.9394
DeepBS	0.9517	0.9987	0.0013	0.0483	0.2424	0.958	0.966
	Category: **cameraJitter**						
Method	Rec	Spec	FPR	FNR	PWC	FMeas	Prec
SuperBE (no post-processing)	0.4801	0.9967	0.0033	0.5199	2.5247	0.5179	0.8626
SuperBE (post-processing)	0.5688	0.9947	0.0053	0.4312	2.0313	0.5851	0.8855
GMM — Zivkovic	0.6900	0.9665	0.0335	0.3100	4.4057	0.5670	0.4872
FTSG	0.7717	0.9866	0.0134	0.2283	2.0787	0.7513	0.7645
SuBSENSE	0.8243	0.9908	0.0092	0.1757	1.6469	0.8152	0.8115
PAWCS	0.7840	0.9935	0.0065	0.2160	1.4220	0.8137	0.8660
DeepBS	0.8788	0.9957	0.0043	0.1212	0.8994	0.899	0.9313
	Category: **dynamicBackground**						
Method	Rec	Spec	FPR	FNR	PWC	FMeas	Prec
SuperBE (no post-processing)	0.7404	0.9993	0.0007	0.2596	1.0368	0.4845	0.6772
SuperBE (post-processing)	0.7458	0.9995	0.0005	0.2542	0.9028	0.7090	0.9372
GMM — Zivkovic	0.8019	0.9903	0.0097	0.1981	1.1725	0.6328	0.6213
FTSG	0.8691	0.9993	0.0007	0.1309	0.1887	0.8792	0.9129
SuBSENSE	0.7768	0.9994	0.0006	0.2232	0.4042	0.8177	0.8915
PAWCS	0.8868	0.9989	0.0011	0.1132	0.1917	0.8938	0.9038
DeepBS	0.8543	0.9988	0.0012	0.1457	0.2067	0.8761	0.9083

The two best results in each metric are highlighted in grey.

Table 2.3: Comparative Accuracy on the CDW2014 Dataset (continued)

	Category: **lowFramerate**						
Method	Rec	Spec	FPR	FNR	PWC	FMeas	Prec
SuperBE (no post-processing)	0.8143	0.9969	0.0031	0.1857	1.1408	0.7352	0.8031
SuperBE (post-processing)	0.8407	0.9950	0.0050	0.1593	1.0749	0.8359	0.9124
GMM — Zivkovic	0.5300	0.9970	0.0030	0.4700	1.3620	0.5065	0.6686
FTSG	0.7517	0.9963	0.0037	0.2483	1.1823	0.6259	0.6550
SuBSENSE	0.8537	0.9938	0.0062	0.1463	0.9968	0.6445	0.6035
PAWCS	0.7732	0.9963	0.0037	0.2268	0.7258	0.6588	0.6405
DeepBS	0.5924	0.9975	0.0025	0.4076	1.3564	0.6002	0.7018
	Category: **nightVideos**						
Method	Rec	Spec	FPR	FNR	PWC	FMeas	Prec
SuperBE (no post-processing)	0.7138	0.9949	0.0051	0.2862	1.2060	0.6650	0.7167
SuperBE (post-processing)	0.7387	0.9938	0.0062	0.2613	1.2087	0.7128	0.8221
GMM — Zivkovic	0.4797	0.9739	0.0261	0.5203	4.7227	0.3960	0.4231
FTSG	0.6107	0.9759	0.0241	0.3893	4.0052	0.5130	0.4904
SuBSENSE	0.6570	0.9766	0.0234	0.3430	3.7718	0.5599	0.5359
PAWCS	0.3608	0.9932	0.0068	0.6392	3.3386	0.4152	0.6539
DeepBS	0.5315	0.9959	0.0041	0.4685	2.5754	0.5835	0.8366
	Category: **PTZ (Pan, Tilt, Zoom)**						
Method	Rec	Spec	FPR	FNR	PWC	FMeas	Prec
SuperBE (no post-processing)	0.7916	0.9650	0.0350	0.2084	3.7120	0.5691	0.5774
SuperBE (post-processing)	0.8252	0.9398	0.0602	0.1748	6.1040	0.5699	0.6756
GMM — Zivkovic	0.6111	0.8330	0.1670	0.3889	16.9493	0.1046	0.0683
FTSG	0.6730	0.9770	0.0230	0.3270	2.5519	0.3241	0.2861
SuBSENSE	0.8306	0.9629	0.0371	0.1694	3.8159	0.3476	0.2840
PAWCS	0.6976	0.9912	0.0088	0.3024	1.1162	0.4615	0.4725
DeepBS	0.7459	0.9248	0.0752	0.2541	7.7228	0.3133	0.2855
	Category: **turbulence**						
Method	Rec	Spec	FPR	FNR	PWC	FMeas	Prec
SuperBE (no post-processing)	0.7771	0.9998	0.0002	0.2229	0.2066	0.7721	0.8766
SuperBE (post-processing)	0.7542	0.9999	0.0001	0.2458	0.1849	0.7745	0.9828
GMM — Zivkovic	0.7786	0.9886	0.0114	0.2214	1.2460	0.4169	0.3494
FTSG	0.6109	0.9998	0.0002	0.3891	0.1987	0.7127	0.9035
SuBSENSE	0.8050	0.9994	0.0006	0.1950	0.1527	0.7792	0.7814
PAWCS	0.8117	0.9950	0.0050	0.1883	0.6378	0.6450	0.6809
DeepBS	0.7979	0.9998	0.0002	0.2021	0.0838	0.8455	0.9082

The two best results in each metric are highlighted in grey.

Table 2.4: Best Results Obtained by SuperBE on CDW Categories

Method	Rec	Spec	FPR	FNR	PWC	FMeas	Prec
badWeather	0.9026	0.9984	0.0016	0.0974	0.3471	0.8569	0.9099
baseline	0.7219	0.9893	0.0107	0.2781	2.0653	0.6043	0.6350
cameraJitter	0.5540	0.9958	0.0042	0.4460	1.9884	0.5805	0.8996
dynamicBackground	0.8330	0.9980	0.0020	0.1670	0.5938	0.7658	0.8899
lowFramerate	0.8543	0.9945	0.0055	0.1457	1.0662	0.8345	0.8921
nightVideos	0.6496	0.9987	0.0013	0.3504	1.0105	0.6561	0.9293
PTZ	0.7392	0.9927	0.0073	0.2608	1.0809	0.5780	0.6390
turbulence	0.8113	0.9998	0.0002	0.1887	0.1509	0.8197	0.9533

category. Of particular interest is comparing SuperBE to SuBSENSE and PAWCS. One way of distinguishing between these algorithms is that SuBSENSE operates at the pixel-level, PAWCS takes local samples to form words, and SuperBE uses superpixels to cluster similar pixels together. Importantly, both SuBSENSE and PAWCS have feedback loops that adjust internal parameters. It appears that SuBSENSE and PAWCS perform quite well on categories where the background is changing or moving significantly (*cameraJitter*, *dynamicBackground*, and *PTZ*) as well as in *baseline*, but SuperBE is superior in categories with more high frequency noise, such as *badWeather*, *nightVideos*, and *turbulence*. There are likely two factors that contribute to this difference - the dynamically adjusting parameters in SuBSENSE and PAWCS may adapt better to changing backgrounds where the background models need to adjust significantly over time, but the use of superpixels better filters out the effect the high frequency noise because the statistical description of the superpixel through colour means and covariance minimises the contribution of outlier noise. SuperBE could be improved in terms of accuracy in the future by incorporating adaptive internal parameters to make it adjust better to dynamic backgrounds, although in theory this would come at the cost of increased computation time. To demonstrate how SuperBE could perform better with improved parameter values, the best results obtained on each category are listed in Table 2.4. These results are in some cases better than all of the competing algorithms, although it is difficult to know if the results of other algorithms could also potentially be improved with parameter tuning.

Output examples of SuperBE compared to other algorithms are presented in Figure 2.7. SuperBE produces similar masks to comparative background estimation algorithms, although it sometimes misses small objects (where the object has not been oversegmented by superpixels). It can be seen that SuperBE tends to have fewer false positives at the cost of more false negatives. This is shown in the metrics as generally achieving a low FPR but performing more weakly in FNR. In a number of categories, SuperBE has the best Specificity and FPR, indicating highly accurate detection of true negatives. This is most easily apparent in the *dynamicBackground* and *turbulence* categories, where SuperBE can produce a very low FPR that is better than the state-of-the-art algorithms, but with a higher FNR contributing to a higher PWC.

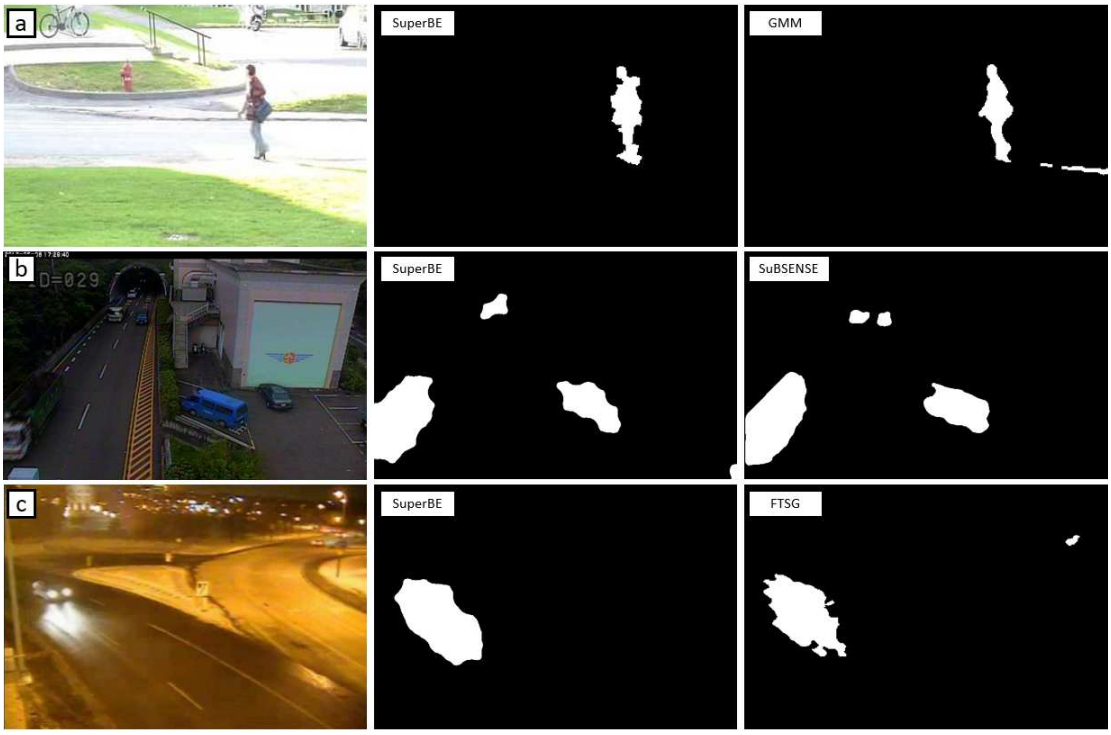

Figure 2.7: (a) A frame from the CDW2014 *pedestrians* sequence, processed with SuperBE, and GMM (Zivkovic) (both without post-processing for fair comparison). (b) A frame from the CDW2014 *tunnelExit_0_35fps* sequence, processed with SuperBE, and SuBSENSE (both with post-processing for fair comparison). (c) A frame from the CDW2014 *winterStreet* sequence, processed with SuperBE, and FTSG (both with post-processing for fair comparison).

Post-processing increases the true positive count (measured through a decreased FNR), generally by closing the holes in any object masks, but can also introduce some false positives (measured through an increased FPR) because the morphological operations mask an area slightly larger than the object(s) of interest. While this is not desirable when measuring accuracy at the pixel-level, from a system point of view a thin false positive border around the object may be appropriate (depending on the application) to provide a small amount of background behind the object(s) of interest, as shown in Figure 2.8. This may be useful when subsequent algorithms use square or rectangular windows for parsing over the image area; an excessive amount of black (indicating background) may cause the windowed data to be misclassified, for example if a machine learning algorithm is trained on examples that do not have black backgrounds. Empirically, this post-processing scheme has produced better results (fewer missed detections (false negatives)) when performing CNN-based person detection as a subsequent step because the input is always rectangular. The FP pixels that are spatially close to the object(s) of interest are generally less problematic for subsequent algorithms processing the same image. None of the existing metrics can sufficiently capture this factor as it is dependent on the target application. However, the performance of the algorithm is reported here with and without post-processing so that the results are more broadly applicable to different applications.

Some reasons have been identified for why SuperBE's accuracy is not better. Firstly, the

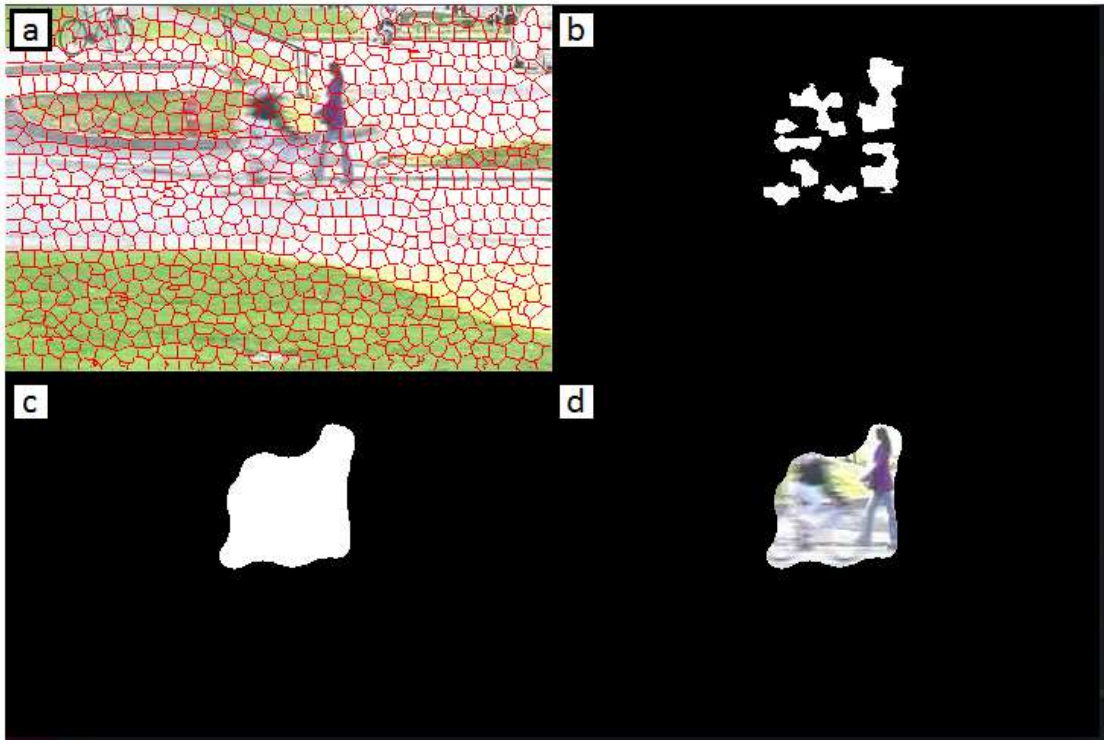

Figure 2.8: SuperBE on the (a) CDW2014 *pedestrians* sequence (b) without and (c) with post-processing, and (d) with the mask applied showing a thin border.

first frame is used in each sequence as the initialisation frame. Some sequences have foreground objects in their initial frames, which become "ghosts" once those objects move in later frames. Since there is no scope for reinitialising the background model during these tests, those ghosts can lead to false positives when computing the statistics. Additionally, SuperBE tends to miss small objects, especially if they are the same size or smaller than one superpixel; an example of this is shown in Figure 2.7c, where the small car in the distance at the top-right is not detected by SuperBE, but would likely not be an object of interest anyway (and is actually outside the region of interest defined for that sequence in the dataset).

Three categories are excluded from these results. In the *thermal* category, the sequences are largely greyscale and there isn't enough difference between the foreground and background for the mean and covariance matrix to sufficiently discriminate between those superpixels. SuperBE does not classify shadows separately to true positives, but shadows have a separate ground truth value in CDW2014, so the *shadow* category was excluded. Also, in most of the sequences in the *intermittentObjectMotion* category, object(s) of interest were in the initialisation frame, creating "ghosts" that led to high PWC. These categories do not represent the target application(s) for SuperBE.

In a number of sequences, such as *office* in the *baseline* category, the ratio between superpixel size and the size of the object(s) of interest is poorly balanced because the objects are very large or very small. This does not cause the algorithm to completely fail, but makes it less accurate. The accuracy could be improved by performing finer object boundary detection,

Table 2.5: SuperBE Computation Time

Image Resolution	Average Computation Time			
	Without Post-processing		With Post-processing	
	ms	fps	ms	fps
320 x 240	7.36	135.94	13.31	75.14
360 x 240	8.54	117.07	15.12	66.14
432 x 288	12.38	80.75	21.38	46.76
540 x 360	19.81	50.49	39.05	25.61
640 x 480	29.37	34.05	50.77	19.70
720 x 480	34.85	28.69	60.50	16.53

either by increasing the number of superpixels (using SLIC) or adding a refinement step after segmenting each frame. However, this would increase the computational cost of the algorithm. Alternatively, superpixel clustering could be applied to every frame with new boundaries each time, similar to [48], but then tracking the superpixels between frames becomes a major challenge (and successfully doing so again increases the computational cost).

Since SuperBE is closely related to ViBe [60], it would be useful to show the improved performance of SuperBE over ViBe. Unfortunately there are no published results for ViBe on the CDW2014 dataset, but there are reported results for ViBe on CDW2012, which is a subset of CDW2014. There are three categories that are both in CDW2012 and the subset of CDW2014 categories that are included in this Chapter: *baseline*, *dynamicBackground*, and *cameraJitter*. Across these three categories, ViBe achieved a PWC of 2.61, while SuperBE achieved a PWC of 2.22 without post-processing and 1.96 with post-processing. This comparison should be interpreted with the knowledge that the parameters for SuperBE were optimised for minimal PWC across all CDW2014 categories, not just for these three, and that SuperBE seems to perform a lot better on some of the other categories as discussed earlier in this section. However, these results indicate that introducing superpixels into the ViBe scheme leads to improvements in accuracy in general.

2.3.3 Comparative Execution Time

The algorithm was developed using OpenCV 3.1.0, and tested using a C++ version of SuperBE compiled with -O3 optimisation. The test computer had an i7 3.4GHz processor (only using one core) running Linux Ubuntu 16.04 with 8GB RAM (but with this process using less than 100MB) and a hard disk drive. Since the processing time is partially dependent on the image resolution, average computation times are provided for a variety of different sizes in Table 2.5. These timings are for the same set of parameter values as presented earlier, and averaged across a few thousand frames. Note that the computation times can vary depending on the parameter values as reported earlier, in particular numMin.

Table 2.6: Comparative Computation Times

	Average Time on CDW2014 (490x330)	
Method	ms	fps
SuperBE (no post-processing)	18.23	54.86
SuperBE (post-processing)	32.29	30.97
GMM — Zivkovic [76]		≈49
SuBSENSE [78]		≈30
FTSG [77]	≈10fps for 320x240	
PAWCS [79]	≈27fps on CDW2012	
DeepBS [56]	10fps for 320x240 (on a Titan X GPU)	
[47] (superpixel-based)	12.3fps for 128x160 on 100 frames	
[48] (superpixel-based)	≈5fps for 320x240 on SegTrack [67]	
[65] (superpixel-based)	≈2fps for 320x240 on SegTrack	
[80] (texton-based)	18.5fps for 320x240 on their Underwater dataset	
[81] (from [47]) (matrix decomposition)	≈0.01fps for 128x160 on 100 frames	

For the purposes of comparing the computation time to other algorithms, the average time across the entire CDW2014 dataset was used, as presented in Table 2.6, which has a weighted average resolution of approximately 490x330 across all sequences. Where possible, this time is compared against other algorithms that also report timing data for CDW2014; however many methods only report accuracy and do not report their timing for this particular dataset. Details about the computation platform that the speed is derived from are also unfortunately often not provided. Comparisons are also shown against other reported background estimation algorithms, particularly recent superpixel-based approaches from the last three years. This comparison is only approximate, as the various algorithms have been tested with different datasets, which may have an effect on the processing speed. All of these factors make this speed comparison less reliable; addressing it requires more attention on speed from the entire computer vision community.

The results show that SuperBE without post-processing is faster than most modern algorithms, with only the simple and less accurate GMM approach achieving a similar processing rate. In particular, SuperBE is significantly faster than other superpixel-based methods for background estimation and object segmentation in video sequences. On 320x240 resolution images, SuperBE without post-processing can achieve 135fps, as shown in Table 2.5. This evidence supports the argument that SuperBE is a computationally efficient algorithm that is suitable for real-time applications, while still achieving comparably high accuracy. It is important to note that adding morphological post-processing does slow down SuperBE's speed, but this is part of the trade-off between accuracy and computational efficiency. There may be alternative post-processing mechanisms that can improve the accuracy more efficiently, such as [44] which uses probabilistic superpixel Markov random fields to clean segmentation maps

Table 2.7: Impact of Background Estimation on Person Detection Time

Number of People Detected	Average Computation Time			
	Background Estimation (SuperBE)		Person Detection (DPM)	
	ms	fps	ms	fps
0	17.3	58.8	2.9	345
1	19.0	52.6	48.9	20.4
2	18.6	53.8	95.7	10.4
3	19.1	52.4	106.1	9.4
4	20.8	48.1	148.9	6.7
Without Background Estimation			172.4	5.8

at roughly 10fps.

2.3.4 Person Tracking Pipeline

In the context of the person tracking system presented in this thesis, using SuperBE significantly reduces the computation time when there are no people in the room, and reduces the search space when people occupy only a portion of the image. Once SuperBE is applied to an image, a dilation operation is performed to fill in any holes and connect contiguous areas together. The contour of the foreground shape is found, and then used to create a bounding rectangle/box around the foreground. A quick check is done to ensure that the window is above a minimum size to avoid small false positive detections, and also to ensure that the windows are compatible with subsequent algorithms that expect inputs of a certain size. Valid windows are then passed to the person detection module - all parts of the image that are outside of this window are discarded. In Table 2.7, the impact of SuperBE on the average speed of the Deformable Parts Model (DPM) algorithm when detecting people in a 640x480 image is shown. The computation time spent by the person detection algorithm is reduced by the background estimation, depending on the number of people in the scene (which influences how much of the image is foreground).

The time taken by SuperBE slightly increases as the number of people increases due to more foreground superpixels needing to be verified for foreground checks. This takes longer than checking if a background superpixel is still considered as part of the background, because SuperBE breaks out of the loop more quickly (based on the B and numMin parameters). When there are fewer people being detected, the person detection time is significantly less than when there are more people (i.e. a larger window of the image has to be checked). The time taken does not vary linearly to the number of people being detected, because it depends on the varying positions of the people in the image space and how close they are together.

2.4 Discussion

2.4.1 Fixed vs. Adaptive Superpixels

As described in Section 2.2.1, SuperBE originally used superpixel clustering on every frame (i.e. *adaptive*), but later was modified to only cluster on the initialisation frame and use that segmentation for all frames (i.e. *fixed*). The fixed scheme has a much lower computation time because SuperBE avoids clustering new superpixels for every frame, but a well-designed flexible scheme would have a better accuracy because it better conforms to the boundaries of the object at a pixel level. This likely explains the large difference in computation time between SuperBE and some other superpixel-based background estimation and segmentation algorithms [46, 65], because clustering and matching superpixels between frames is a computationally non-trivial task. As reported in [48], tracking the superpixels requires a very sophisticated algorithm because the location of the superpixels can change between frames, and the superpixel models may need to "move" between frames as objects move. The choice between a fixed or adaptive superpixel scheme is ultimately application specific - since the object/motion is still detected in the fixed scheme, the lower computation time is more important for the target application, even if there are more false positive pixels. This is a classic case of a trade-off between accuracy and speed, but at some point, additional accuracy may not actually be worth the computation cost. The motivation for developing background estimation is to isolate regions of interest for subsequent processing steps in order to reduce unnecessary processing over true negative areas, so an additional percentage point in accuracy does not have a significant impact on the accuracy of the subsequent steps. The accuracy of SuperBE is at an acceptable level for most applications, while achieving high computation speeds. In applications where pixel-level object segmentation accuracy is important, then adaptive superpixels should be used with the acceptance of higher computation costs.

2.4.2 Future Work

There is a lot of potential for future investigations with SuperBE. Firstly, the algorithm currently uses floating point mathematics and represents the means and colour covariance values as floating point numbers, which may be unsuitable for hardware implementations. It would be worth investigating the effects of casting/rounding in an integer-only version of the algorithm; this would potentially further reduce the computation time at the cost of decreased accuracy. Secondly, a greyscale-only version may be worth investigating, replacing the colour covariance matrix with simple variance. This would similarly reduce the computation time at the cost of decreased accuracy; the greyscale version of ViBe [60] is approximately 15% faster than the RGB version, with less than 1% higher PWC.

SuperBE can also be easily parallelised - for each frame, each superpixel can be classified and even updated independently of the other superpixels, so this could also be explored for multi-core architectures. It may be interesting to target GPUs or FPGAs to leverage that parallelism and further improve the computation speed. Hardware/Software Co-Design

(combining a standard CPU with some hardware fabric such as an FPGA) could be a good fit, so that more complex sequential steps like SLIC clustering can be performed in software while simpler independent tasks such as mean and covariance calculations can be performed directly in hardware. Other computationally efficient discriminative features for classifying superpixels could also be investigated; for example, other measures that incorporate both means and variances, or histogram-based methods. Along with quantisation and greyscale-only optimisations, these hardware acceleration ideas are explored later in the thesis in Chapter 8.

It may be worthwhile to explore the use of superpixel-level algorithms in comparison to pixel-level approaches on higher resolution images. This is a gap in the existing literature, mainly due to the lack of truly high resolution datasets. Recent datasets such as HDA+ [43] have focused more on high-resolution video sequences, but do not necessarily have ground truth for background estimation. Superpixel-level algorithms should see an even greater superiority in computational efficiency in comparison to pixel-level approaches as the number of pixels increases in the images. Higher image resolutions may also help with the accuracy of the algorithm, because the number of superpixels forming the object(s) of interest can also increase.

Further investigation of different methods for comparing covariance matrix similarity may also improve the accuracy of SuperBE. Lastly, it could be worthwhile to develop a version of SuperBE with adaptive parameters, similar to the relationship between PBAS and ViBe. Using some form of learning, the optimal parameter values for each sequence could be determined at runtime, improving the accuracy (although at the cost of significantly increased computation). Removing the parameters from human control is desirable, as it reduces the likelihood of a human selecting sub-optimal values.

2.5 Conclusions

This Chapter presented a novel background estimation algorithm called SuperBE that works at the superpixel-level to improve the segmentation of moving objects while minimising computation costs. Using colour means and colour covariance matrices as discriminative features, it has been demonstrated that the algorithm achieves a high level of accuracy comparable to other state-of-the-art algorithms on a widely used dataset, while also achieving very low computation times, in particular when compared against other superpixel-level background estimation and segmentation algorithms. The impact of SuperBE on the overall person tracking system is also described. The key contribution of this algorithm is the use of superpixels for background estimation in a computationally inexpensive manner, with broad applicability to a wide range of video sequence scenarios. The code for SuperBE has been made publicly available at `https://github.com/andrewchenuoa/superbe`.

In the next Chapter, the next stage of the person tracking pipeline is discussed - person detection. Using background estimation to reduce the search space can significantly reduce the computation time, but the largest factor in the time taken by a person detection algorithm is still the computational complexity of that algorithm itself. An intelligent choice has to be made between different algorithms that provide varying speed vs. accuracy trade-offs.

Chapter 3

Person Detection and Feature Extraction

Person detection, sometimes also called human detection or pedestrian detection, is one of the most important tasks in computer vision. In a world constructed to suit the needs of humans, where humans are the primary users of technology, and where our interactions as humans are primarily with other humans, it makes sense that many useful applications of computer vision require the ability to recognise and detect people. Person detection can be treated as a subset of the more generic task of object detection, where the target object is a person. Generally this requires a "person model", or in other words some idea of what a person looks like or how a person is structured, so that matches can be found within image data. Under controlled conditions, this is relatively trivial, but the main challenge is developing models that can deal with significant amounts of variation in real-world environments, both intrinsic and extrinsic to the target(s). Intrinsic differences include the fact that people look different to each other because they can wear different clothing, they can be tall or short, they can have missing limbs, and so on. Extrinsic differences include varying lighting conditions, different camera view angles, partially obscured views, and more. The person detection methods have to be generic enough and robust enough to detect people in all circumstances, particularly in the context of safety-critical systems like autonomous vehicles or physical security where missed detections are costly.

In the proposed video analytics system, the purpose of person detection is relatively obvious - once the background has been removed and moving objects are segmented, it should be validated that those moving objects are in fact people before trying to track their movement. Additionally, once a person is detected, features that describe the appearance of the person can be extracted to help with appearance-based re-identification later in the pipeline. In this Chapter, a comprehensive literature review of person detection approaches is presented, resulting in the selection of two algorithms: the Deformable Parts Model (DPM) and Aggregate Channel Features (ACF). Feature selection and extraction are also discussed with reference to other methods from the literature.

3.1 Finding People

Significant effort has been poured into person detection over the last two decades. With the rise of autonomous vehicles, most of the interest has been in detecting pedestrians so that the vehicle can avoid the hazard and stop (hence the popularity of the term "pedestrian detection"). However, person detection for video analytics or surveillance systems typically has slightly different requirements. Where autonomous vehicles operate outdoors, video analytics may be both indoors or outdoors, which has implications on lighting conditions and frequency of obstacles appearing. Where autonomous vehicles typically have cameras at the torso or head height, video analytics systems tend to have cameras on the ceiling, high above people looking down at an angle. Other differences include the size of the person relative to the image, the likelihood of crowds of people obscuring each other, and the type of clothing that the people might be wearing. This is reflected in the wide variety of datasets that have been annotated with bounding boxes for people; where Caltech-USA [82] is captured from the perspective of a moving vehicle, HDA+ [43] and DukeMTMC [39] are captured from static cameras around university campuses.

Early person detection relied on the use of blob detection [19, 83, 84]; given that the developers could know what the approximate size of a person would be *a priori*, then any moving object that approximately matched the size (in particular, the ratio of height vs width) would be counted as a person. While conceptually simple and able to be computed without much processing power, the accuracy was very low. People might not be facing the camera, and a side-on view has very different proportions. People might be bent over or leaning, or sitting down, which would be missed by the system. Other objects such as light stands or doorways could be similar in size to a person. A person holding a large object might cause their blob to appear much bigger, and thus cause the system to reject the blob as a person. This approach was also dependent on either background subtraction or image segmentation to identify blobs, and these methods still had a lot of difficulties with varying lighting conditions and low image resolutions at the time. A higher level of sophistication was needed, and researchers started to ask "what makes a person look like a person?"

Perhaps the most well known historic approach to person detection is the Histogram of Oriented Gradients (HOG) and Support Vector Machine (SVM) algorithm. Even though HOG features had been around for almost twenty years, this approach was popularised by the work of Dalal and Triggs in [86], when it was applied towards human detection successfully. The HOG descriptor is essentially a texture feature, which measures the direction of gradients in the intensity of pixels. Typically, this would be combined with patch-based sampling, where an image is sliced up into a regular grid, so that the HOG descriptor was calculated for each patch. It was shown that this approach can match the shape of objects very well, particularly where objects have well defined edges or contours. Images of people were analysed to determine what common person shapes were, and how they might be described using HOG features. SVMs were then used to classify these patches as being good or bad matches to patches in comparison to a previously trained model of valid patches, producing a score that can be used as a measure

Figure 3.1: An example of person parts being extracted using DPM [85]

of confidence that there is part of a person in this patch. This work was further extended into the Deformable Parts Model (DPM) [85], which also used HOG features but introduced a structural relationship between the parts; rather than just accumulating patches that had high scores, DPM requires that those patches have to be in a particular spatial configuration. For example, in the context of person detection, a part that represents the head should be higher than the parts that represent the legs; if this is not the case, then even though the individual parts may have high matching scores, DPM will lower the confidence of the match because it does not match the expected structure of the patches. An example of DPM and the extracted parts is shown in Figure 3.1. As an object detection framework, DPM was shown to have very high levels of accuracy in comparison to the state-of-the-art at the time, but HOG features are very computationally expensive to calculate, and thus adoption was hampered by slow processing times. As an approximate indication, while DPM was able to achieve approximately 26.6% mean accuracy across the twenty categories of the PASCAL 2007 object detection datatset [87], it required multiple seconds of computation time per frame. Efforts to speed this up have largely relied on restricting the search space, such as enforcing geometric constrants using known camera calibration information to estimate bounds for the future positions of a person [88], using boosting to prune bad hypotheses earlier in the computation [89], or using background estimation as described in the video analytics system in this thesis.

In the early 2010s, with the impending arrival of autonomous vehicles and hard real-time safety-critical systems, there was a focus on fast person detection. The work of Dollár developed new descriptors based on the concept of “channel features”, which applies different transformations to an image and then computes sums, histograms, and Haar-like features for each of those transformations. This initially was implemented through the use of Integral

Figure 3.2: An example of ACF pedestrian detection [82] applied to the Oxford Town Centre dataset [90] (from Yuan Gao, Kiel University)

Channel Features (ICF) [91], which took advantage of recent work on integral images that accelerated the computation of HOG features, and applied this to HOG, colour (including grayscale, RGB, HSV, and LUV), and gradient magnitude channels. In a sense, this method was effective because it combined a multitude of different types of information, more than many previous person detection approaches. The focus on speed led to a number of optimisations, firstly through the approximation of features at different scales to reduce redundant computation in The Fastest Pedestrian Detector in the West (FPDW) [92], then through sharing features between neighbouring sampling windows to avoid processing the same pixels multiple times in Crosstalk Cascades [93], and finally through isolating the features that had the largest contribution towards accurate person detection in Aggregate Channel Features (ACF) [87], focusing on gradient magnitude, HOG, and the LUV colour channel. These approaches were popularised by the release of these algorithms as a MATLAB toolbox, which was freely available. While ACF-based approaches were slightly less accurate than DPM (24.5% on PASCAL 2007, about 2% lower than just DPM [87]), the speed was significantly faster, with ACF detecting people at slightly over 30fps on a single core desktop CPU. In [94], a number of person detection algorithms are compared and combined, including DPM and ICF (a precursor to ACF), with results that validate the assumption that ICF/ACF is faster than DPM but less accurate, although the differences reported in this paper are not very large. An example of ACF being applied to pedestrian detection is shown in Figure 3.2, where it can be seen that there are many false positives and false negatives which may need to be pruned or post-processed, but this is offset by a higher computation speed. Work towards accelerating person detection was supported by other researchers, such as Benenson using stixels (rectangular superpixels) to reduce redundant

computation between segments of an image [95], and de Smedt using a warping window approach and exploiting temporal information in videos [96].

Indeed, person detection is considered accurate enough that many large-scale person re-identification datasets use person detection algorithms to help generate the ground truth rather than manually annotating all of the people in the footage [39, 40]. However, a recent paper asked "how far are we from solving pedestrian detection?" [97], which concluded that there is still a $10\times$ gap between a human baseline and the top ensemble method on the Caltech dataset. It is important to note that not all datasets are equal; while the older MIT CBCL pedestrian dataset from 2000 is considered "solved" at this point, the best result on the TUD-Brussels pedestrian dataset [98] still has a 42% miss rate (i.e. error rate) [82]. But this is considered in terms of overall accuracy, which has different meanings for different applications. The normal way to assess the accuracy of a person detection algorithm is to compare all detected bounding boxes to those drawn by human annotators manually for the dataset, with successful matches declared if the Intersection over Union (IoU) (i.e. overlap) is more than 0.5. For a safety-critical system like an autonomous vehicle, where a person that is a hazard to the vehicle needs to be detected as soon as possible, a false negative (i.e. missing a detection) could mean the difference between life and death. For a video analytics system, this requirement is likely weaker, because the consequence of missing a single detection is not significant. Even if a few detections are missed, the position of a person can be easily interpolated if there are valid detections before and after a series of missed detections. Therefore, the type of error has an impact; while (limited) false negatives are acceptable, false positives (i.e. detecting a person when there isn't one) could have a negative impact on accuracy if they are not filtered out.

Thus far, these approaches can be considered relatively traditional - they are well understood with strong formal underpinnings, and generally deterministic and predictable. More recently, the evolution of Convolutional Neural Networks (CNNs) and deep learning have yielded a new generation of object and person detection approaches, which are typically less analysasble and treated more as a black-box. While CNNs have been used in computer vision since the mid 2000s, Girshick (one of the co-authors on the original DPM paper) made significant improvements by combining region proposals (i.e. using a faster but less accurate algorithm to propose areas that potentially have targets in them for further analysis) with CNNs to produce R-CNNs [99], which beat the previous best result on VOC 2012 by more than 30%. This was further accelerated over the coming years, with Fast R-CNN [100] and Faster R-CNN [101]. Even though Faster R-CNN is considered "fast" in comparison to other CNN-based techniques, it runs at 5fps on a very high-end GPU, making it much slower than non-CNN techniques and well out of the reach of embedded computation platforms. Recently, Redmon's YOLO (You Only Look Once) [102] architecture has been very popular, by changing the CNN structure so that the image is only parsed once rather than processing multiple potentially overlapping windows. This popularised the rise of Single Shot Detectors (SSDs) [103], with YOLO falling into this category of CNN methods. Achieving a similar level of accuracy to Faster R-CNN, YOLO is able to reach 30fps on a Titan X GPU, which is very fast but still dependent on high-performance computational hardware. A smaller version of the network, TinyYOLO, has been

popularly proposed as an alternative for mobile/embedded systems, with a lower accuracy rate (about 33% in YOLOv3), but a much higher computation speed (220fps on the same Titan X GPU, falling to about 6-11fps on a high-end Apple smartphone[13,14]). Recent efforts towards accelerating CNNs for mobile applications include MobileNets from Google [104], which in particular highlights the accuracy-speed trade-off by providing a series of model options that have varying levels of accuracies and parameters/weights (which is directly related to the computation time). However, top-end modern smart phones still have more computation power than many embedded systems and smart cameras, and claims of real-time computation are usually assisted by large-scale parallelism enabled by high-end GPUs. Empirically, attempts by our research group to run any CNN-based architecture on embedded systems either ran out of memory or required multiple seconds or minutes to process one frame.

In safety-critical systems, there may be hard real-time requirements (i.e. an image must be processed within a certain time limit), such as in an autonomous vehicle where there has to be enough time for a vehicle to brake and physically come to a stop before it collides with a pedestrian. In a video analytics system, this requirement is again weaker, but it is still important that computation is close to real-time so that results are available in a timely manner and system owners can act on the insights. Work focusing on very fast person detection has yielded some impressive results as described above, but the choice of computing hardware is also critically important and often a hard engineering constraint. A Raspberry Pi, which can be used as a benchmark for the level of computing resources available on a top-end smart embedded camera, has significant difficulties running any modern person detection approach in real-time. The fastest is 3fps achieved using a HOG + SVM method, and even that is reported to only parse part of the image [105]. Recent reports show that using a Intel Movidius Neural Compute Stick can achieve 6-7x speedups by providing additional parallel computation ability (similar to a GPU), but this has a much higher power and monetary cost, and still only achieves 3-4fps with the MobileNet SSD[15].

Taking the two main factors of accuracy and speed into account, two algorithms were selected for the video analytics system. The primary algorithm was DPM; even though it is an older algorithm, it has a relatively high level of accuracy, and recent optimisations made in OpenCV [75] have improved the speed. A few modifications had to be made to the DPM algorithm provided in OpenCV in order to return the parts, rather than just a bounding box for the entire person. This is important for using parts-based sampling of features, as explained in the next Section. The secondary algorithm was ACF, which has a slightly lower accuracy but runs much faster. The strategy was to use DPM in the first instance to ensure that the rest of the pipeline was developed with a reasonable level of detection accuracy, and then to replace the DPM module in the pipeline with ACF to assess the impact on overall accuracy and speed. It is important to note that in both cases, the time taken for person detection is already significantly reduced by the use of background estimation to reduce the search space.

[13] Matthijs Hollemans, "Real-time object detection with YOLO", May 2017.
[14] Maneesh Apte, Simar Mangat, and Priyanka Sekhar, "YOLO Net on iOS", *Stanford University*, 2017.
[15] Adrian Rosebrock, "Real-time object detection on the Raspberry Pi with the Movidius NCS", Feb 2018

3.2 Describing People

Once a person has been detected, that person needs to be represented in some way. While the person is already represented with pixels, it is common to reduce the amount of information by extracting features, forming what is known as a feature vector or signature. The aim is to select features that allow for high inter-class variation (significantly different between multiple people) while maintaining low intra-class variation (similar for the same person). Types of features include colour, texture, and structure, and within each of these types there are many choices. For example, colour can be represented in different colour spaces, such as RGB (Red-Green-Blue), HSV (Hue-Saturation-Value), YCbCr (Luminance and Chroma), and CIELAB (designed to approximate human colour perception). Additionally, multiple features can be extracted and used together, in a balance between representing people with a minimal amount of information while providing enough information to subsequent algorithms to successfully perform classification and re-identification. There is yet to be any consensus on an optimal set of features to use, and it may be that feature selection is application specific and dependent on the other algorithms used in conjunction (e.g. different classification algorithms may prefer features of different types)

In the context of person detection, recent papers have used a variety of different features, summarised in Table 3.1. It is important to note that this table focuses on the feature selection/extraction part of these works, rather than any classification or ranking algorithms even if they are the main focus of the work. Assessing the overall accuracy of these feature choices is challenging, because many are evaluated using different classification algorithms on varying datasets. However, work like [121] has shown that descriptors that include both colour and texture tend to perform better than either one alone, and that some of the more recent descriptors such as GOG and LOMO which operate at multiple scales are particularly effective. However, these methods tend to be much more computationally expensive, and are not very compact (i.e. the descriptors themselves are also quite large, with thousands of dimensions). To balance between the different engineering requirements at play, a 15x16-bin HSV 2D-histogram (across hue and saturation only) and a 16-bin LBP histogram are used in this system to represent colour and texture. HSV was selected over RGB due to its better resilience against lighting variation (by dropping the value channel). These features can be computed very quickly, and have been shown to provide good discriminative strength [108, 109, 122]. This is also similar to the popular LOMO [118] approach, although with a simpler and less verbose sampling scheme. It is important to consider that features are generally not uniform across an entire person. Extracting features across an entire person can cause more dominant features to suppress finer details that may be highly discriminative between different identity classes. For example, it is common for the colour of the upper half of a person to be different to the bottom half, because it is plausible for a person to be wearing clothing of different colours, and to extract features across the entire person would just average out these two distinct characteristics. To help improve the discriminability between people, sampling is used to extract features for different areas of the person.

Table 3.1: Features for Representing People in the Literature (in chronological order)

Feature	Description
ELF [106]	An ensemble of localised features made up of eight colour channels from RGB, YCbCr, and HSV combined with nineteen texture channels from Schmid and Gabor filters
Covariance Regions [71]	Covariance of pixel locations and the values, gradients, and orientations of the RGB colour channels
Haar-like [107]	Trained Haar features extract RGB colour information from segments of the person image
RGB+LBP [108]	Combines RGB colour histograms with LBP texture histograms
Attributes [109]	Uses RGB, HSV, and YCbCr colour histograms with Gabor and Schmid texture features
SDALF [110, 111]	Combines HSV colour histograms with a measure of colour displacement (MSCR) and the presence of repeated structures, leveraging the vertical symmetry in person appearance
Colour Spaces [112]	Combines the use of moments and histograms to describe the HSV and YUV colour spaces
DenseColorSIFT [113, 114]	LAB colour and SIFT texture features are combined, with very dense sampling
gBiCov [115]	Uses Biologically Inspired Features (BIF), based on Gabor filters, and converts them into covariance matrices
CovCov [116]	Takes the covariance of the covariance matrices calculated from gray-level intensity of the pixels in each part/sample
HistLBP [117]	Instead of using the more common Schmid and Gabor filters, histograms of Local Binary Patterns (LBP) describe texture
LOMO [118]	Uses HSV and LBP histograms at multiple scales, transforms the data to maximise the local occurrence of these features
GOG [119]	Uses a Gaussian of Gaussians descriptor to represent the distribution of pixel-level colour and texture features
Semantic Channels [120]	Combines HOG texture and LUV colour descriptors at multiple scales with geometric structures

In *patch-based sampling*, very simple segmentation approaches are used, such as slicing the image into horizontal stripes [38, 106, 107] or using approximate blobs that correspond to the upper and bottom halves of the person [31, 32]. More recently, dense sampling approaches [123] have seen a resurgence [113, 124, 125]. Here, the image is segmented into a regular grid of patches, and then features are extracted for each patch. The sampling is parametrised by the denseness of the segmentation; for example, an image could be segmented into a 4×4 grid with 16 parts (not very dense) or a 20×20 grid with 400 parts (very dense). There becomes a trade-off between isolating finer detail against memory and computation costs. It is also common to combine neighbouring patches together with overlap so that the boundary effects of

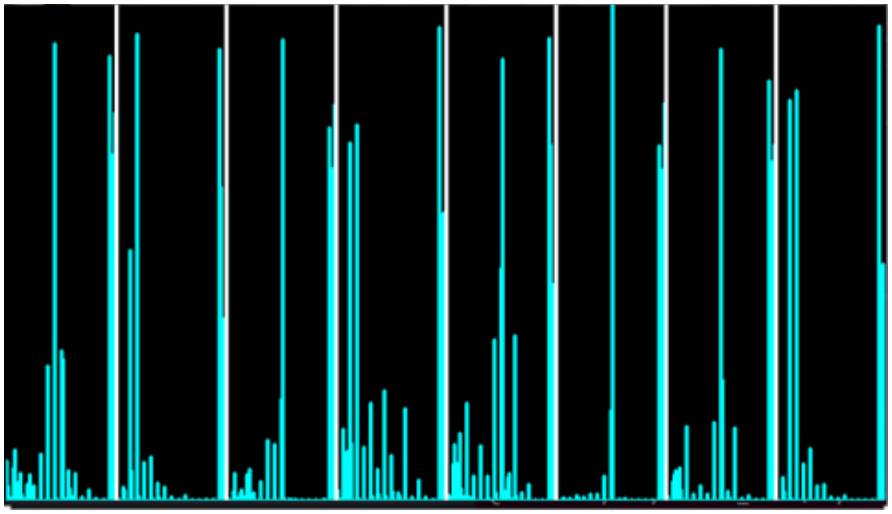

Figure 3.3: A visual representation of an example feature vector made up of HSV colour and LBP texture histograms, with eight parts separated by white lines

arbitrarily selecting a patch density are reduced. This segmentation approach has the advantage of being very computationally efficient, since in most cases it is just taking a slice of the existing memory in a deterministic way. However, these patches generally do not have any semantic meaning, and it is a very rough approximation of the underlying principle that different parts of the body have inter-part variation, significant enough to justify separating them in the first place. Additionally, recent methods encourage segmentation into a larger number of parts, which negates the computational lightness of segmentation by introducing significantly more computation and memory costs for feature extraction and classification. It can also be very challenging to track the patches as people move, because there is a dependency on the accuracy of the bounding box from the person detection algorithm, which may not always position the person in the same place relative to the boundaries of the box, causing the same patch to represent different parts of the body in different frames.

The approach in *parts-based sampling* is to isolate out specific parts of the body as a meaningful descriptor of the body; for example, an image of a person can be segmented into the legs, arms, torso, and head. Features for each part are extracted separately, in order to preserve the details that may differ between these parts. Using body parts provides a better proxy for how humans identify individuals at a distance [27, 71, 110]; for example, a human might identify a person by saying that they have brown hair and are wearing a red jacket, which could be approximated by a computer identifying that the head part of the human has a significant portion of brown and the chest part of the human is predominantly red. This is more complicated than patch-based sampling, because each part needs to be matched or verified to a model with semantic meaning. One way of extracting parts out of a detected person is to use the Deformable Parts Model (DPM) [85] object detection framework, which extracts parts during the process of detecting people, which is more efficient in this case than using a separate algorithm to segment the parts. The person model that is used in this work has eight parts:

head, torso, left shoulder, right shoulder, left hand, right hand, left leg, and right leg. Each part has its own feature vector, which can either be compared to the corresponding part in another person independently of all other parts, or concatenated together to form an overall feature vector representing the person. Figure 3.3 shows a visual representation of an example feature vector (made up of HSV colour and LBP texture histograms) representing an entire person, where there are eight parts separated by white lines. An advantage of using DPM is that a confidence score is provided for each of these parts, which allows the system to determine which parts may be obscured or out of frame (i.e. low confidence). When this happens, those parts can be ignored, reducing the amount of unwanted noise impacting the system [108].

There is also a more recent trend towards segmenting into more specific parts. Methods such as [126] describe detected people in terms of their clothing and accessories, such as "blue short sleeve shirt" or "black backpack", and match those areas across images. This is also common in saliency learning [113], where the system attempts to identify the parts of a person that are significantly different to all other people, which often corresponds to the person wearing a backpack, or carrying a handbag, or wearing a hat, or similar. However, most of these methods use computationally expensive CNNs with large filter banks to test different parts of the image in a wasteful way. While these types of methods have achieved very high accuracy rates, they also require large amounts of densely labelled data, threatening the practicality of testing and developing this approach. This demonstrates that parts-based sampling, while potentially more accurate, is often very expensive to use if the parts have to be detected as a separate stage after person detection. Since DPM already inherently has parts, it is easy to use parts-based sampling with it. Meanwhile, since ACF only returns a bounding box for where the detected people are, it makes sense to use patch-based sampling in this case.

3.3 Conclusions

In this Chapter, person detection and feature extraction are thoroughly discussed, with reviews of methods presented in the existing literature presented and compared. While recent efforts using Convolutional Neural Networks (CNNs) have achieved impressive levels of accuracy, they remain generally impractical for use in embedded systems due to the high computational complexity leading to slow execution times. More traditional methods that use gradients to analyse shape are preferred, as they have a better trade-off between accuracy and speed. Different choices of features are also evaluated, and a combination of HSV for colour and LBP for texture are used to represent patches or parts of the detected people. The primary method that is used in the video analytics pipeline is the Deformable Parts Model (DPM) with parts-based sampling, and the Aggregate Channel Features (ACF) method with patch-based sampling is used as a secondary approach. In the next Chapter, person re-identification is presented, using the feature vectors that have been extracted from detected people. These feature vectors undergo a process of dimensionality reduction and metric transformation, followed by classification into identity classes.

Chapter 4

Person Re-identification

Once people have been detected in a frame, the next step is collecting the detections together over time to help track the individual. The main challenge is introduced when there are multiple cameras in a network. In the single-view case, cameras are limited by their field of view, by the aperture and orientation of the cameras, as well as by obstacles that obscure the object/person of interest. To mitigate this, multiple cameras are used, either overlapping or non-overlapping, at different camera positions and angles to capture a larger proportion of the space. Therefore, it is necessary to identify that one person in one camera view is the same as a person previously seen in another camera view, so that these two instances can be connected to the same identity and tracked throughout the global physical space. There are two ways of approaching this problem: either a spatio-temporal model that uses the topology of the cameras to connect tracks based on when and where objects have entered and exited camera views, or a re-identification approach that matches the visual appearances of individuals across multiple camera views [10]. In this thesis, formulations of both approaches are used and then combined together to improve the accuracy overall; this Chapter presents the appearance-based re-identification approach, while Chapter 5 will present the spatio-temporal tracking approach, as well as the model fusion of these two approaches.

Person re-identification is a commonly studied area, with many papers appearing at recent computer vision and pattern recognition conferences. However, this task is non-trivial; an example of some samples from two identities across four cameras is shown in Figure 4.1, demonstrating some of the challenges. There are significant differences in camera and subject pose, partial obscuration by objects naturally appearing in the environment, motion blur from low quality cameras, and the presence of multiple lighting sources causing significant colour changes and saturation. Importantly, these issues appear not only between cameras but also within camera views, making learning a single transfer function between the different camera views very challenging. The person re-identification task can be divided into several components. Once a person has been detected in a camera view, that person needs some form of data representation. One option is to pass the pixel information of the person directly into a deep convolutional neural network (CNN) for classification [127, 128]. In the case of [129], person detection/search and re-identification are done jointly on raw images with a

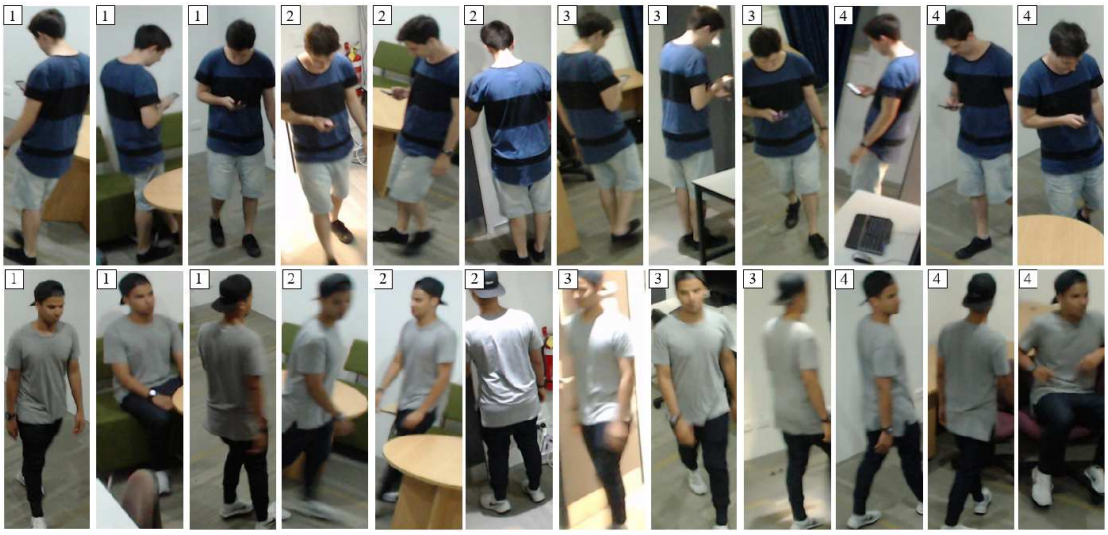

Figure 4.1: Samples from the UoA-Indoor re-identification dataset extracted with DPM for two classes, with the camera view number shown in the top left

single CNN. However, CNN approaches require a significant amount of training data for each identity and are computationally very expensive, and still face many challenges for real-time embedded implementations. A generally more preferable and simple solution is to extract some higher-level features, manipulate that data to make it more separable, and then classify those features into person identity classes [38, 130, 131, 132]. This also better supports one-shot or unsupervised learning methods in scenarios with limited training data, where the system is expected to learn as it processes real data. Additionally, any method needs to be scalable, since in a real-world scenario it might reasonably be expected that a system may have to manage hundreds or thousands of possible identities.

In this Chapter, a complete yet computationally light person re-identification approach is presented, including dimensionality reduction, metric learning, and classification. The steps conducted in this process are shown in Figure 4.2. Experiments are conducted for each of these steps to determine appropriate methods and parameter values. The efficacy of this approach is compared against other methods quantitatively. In particular, the focus for the development of this practical approach is on computationally light re-identification, as well as one-shot or unsupervised learning that can be used in real-world situations where large amounts of training data are not available. For the purposes of this Chapter, re-identification is discussed independently from position-based tracking, which will be combined with re-identification in Chapter 5.

4.1 Dataset

Evaluation of different approaches requires a standardised dataset for fair comparison; a selection of popular datasets is shown in Table 4.1 in approxixmate chronological order. There

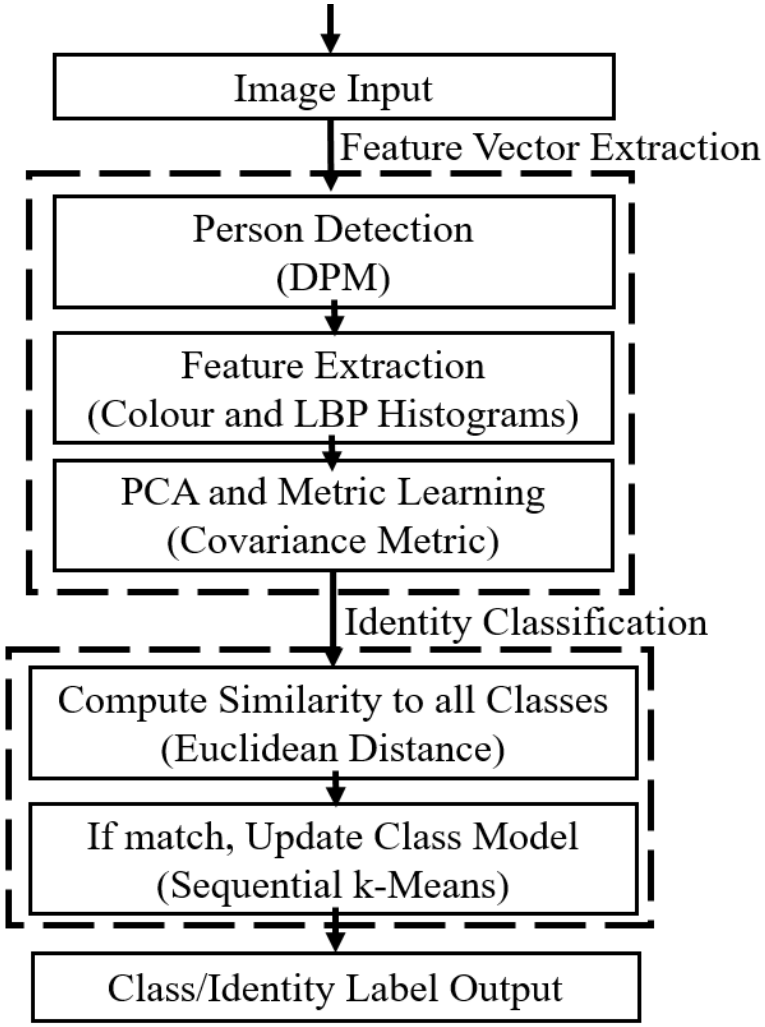

Figure 4.2: The feature extraction and identity classification processes in the appearance model

are many choices available, but generally datasets have become larger with more identities as technology has made it easier to capture, store, and transfer large amounts of data. A standard re-identification formulation is perhaps not as suitable for the motivating target application - rather than re-identifying one individual with only one or a few valid gallery samples, it is necessary to continually re-identify a person over time with an increasing number of gallery images, both within and across different camera views. Instead of using a dataset like ViPER [133], which only has a single source and single target image for each identity, person tracking datasets that can be used to extract people from each frame and build up a gallery over time are preferred, providing the algorithm with more information as time progresses. [39] includes a discussion of existing person tracking datasets, but most of these datasets only contain bounding boxes and do not include location data or ground truth topology information between the cameras. Recent datasets such as HDA+ [43] and DukeMTMC [39] are perhaps the closest to this use case, but in both cases the calibration matrices do not seem to be very accurate and there is limited or no overlap between the camera views. As noted in [43], many

Table 4.1: Person Tracking and Re-identification Datasets

Dataset Name	Context	Identities	Cameras	Samples per Class	Real-world Positions or Calibration Matrices
ViPER [133]	Outdoor	632	2	2	✗
i-LIDS [134]	Outdoor	300	2	2	✗
3DPeS [135]	Outdoor	192	8	>10	✗
EPFL [136]	Mixed	30	4	≈9	✗
CMV100 [137]	Indoor	100	5	>100	✗
HDA+ [43]	Indoor	53	13	>100	✓
CUHK03 [138]	Outdoor	1467	5	>10	✗
Market-1501 [125]	Outdoor	1501	6	≈20	✗
DukeMTMC [39]	Outdoor	>2000	8	>100	✓
MARS [40]	Outdoor	1261	6	>10	✗
PRW [139]	Outdoor	932	6	36	✗
UoA-Indoor	Indoor	19	4	>100	✓
Airport [41]	Indoor	9651	6	>10	✗

of these datasets are only suitable for very narrow experiments, with the ground truth labelling intended for metrics focused on specific tasks (e.g. person re-identification alone) that are less suitable for measuring the performance of complete person tracking systems.

The standard formulation of a re-identification challenge, such as in ViPER [133] and DukeMTMC4ReID [121], involves having a large number of identities and a small number of samples for each identity in a gallery, and then trying to match a new source image of a person to an existing target (or probe) image in the gallery. In contrast, with an indoor retail application it is likely that there will be a smaller number of identities across a given time period, who remain within view for a much longer time period, allowing for more samples of each identity to be collected [38, 40, 140]. In the motivating target application, a global identification system is not required; individuals do not need to be re-identified when they come back to the store on a different day or when they are wearing different clothing, only while they are within the shop during a single visit, so a local identification designation is sufficient. For the purposes of presenting one-shot learning algorithms it is also assumed that with the assistance of an auxiliary sensor, such as an infra-red detector or a Radio Frequency Identification (RFID) authenticator at a turnstile, the first time a person enters a store they can be detected and designated as a new identity class, so that the first frame can be used as an anchor in a one-shot learning setting.

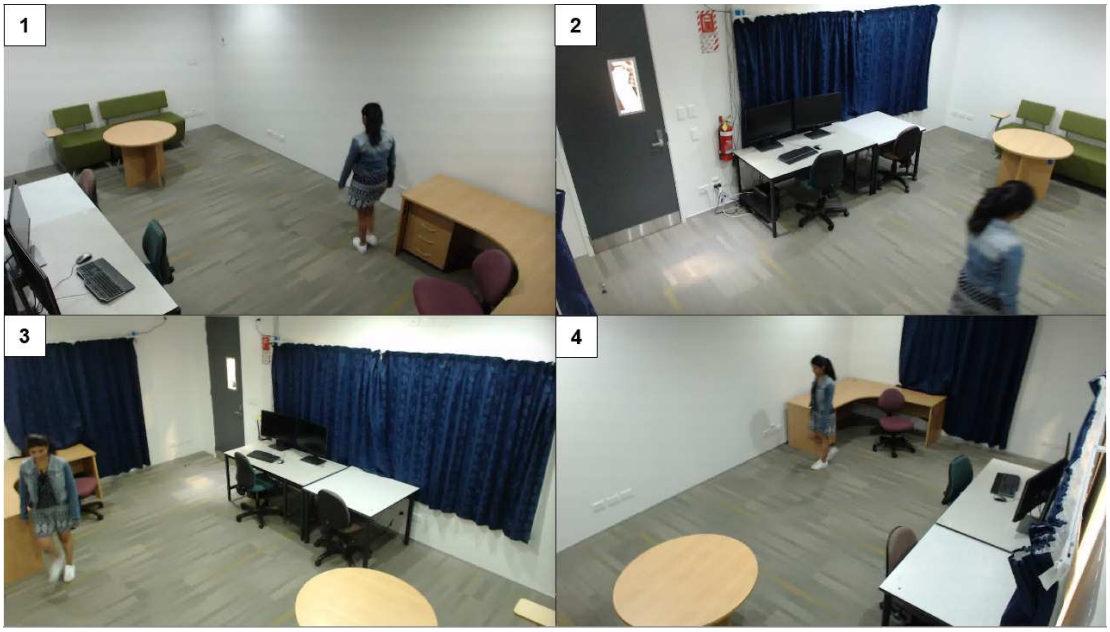

Figure 4.3: Side-by-side views from the four cameras in the dataset

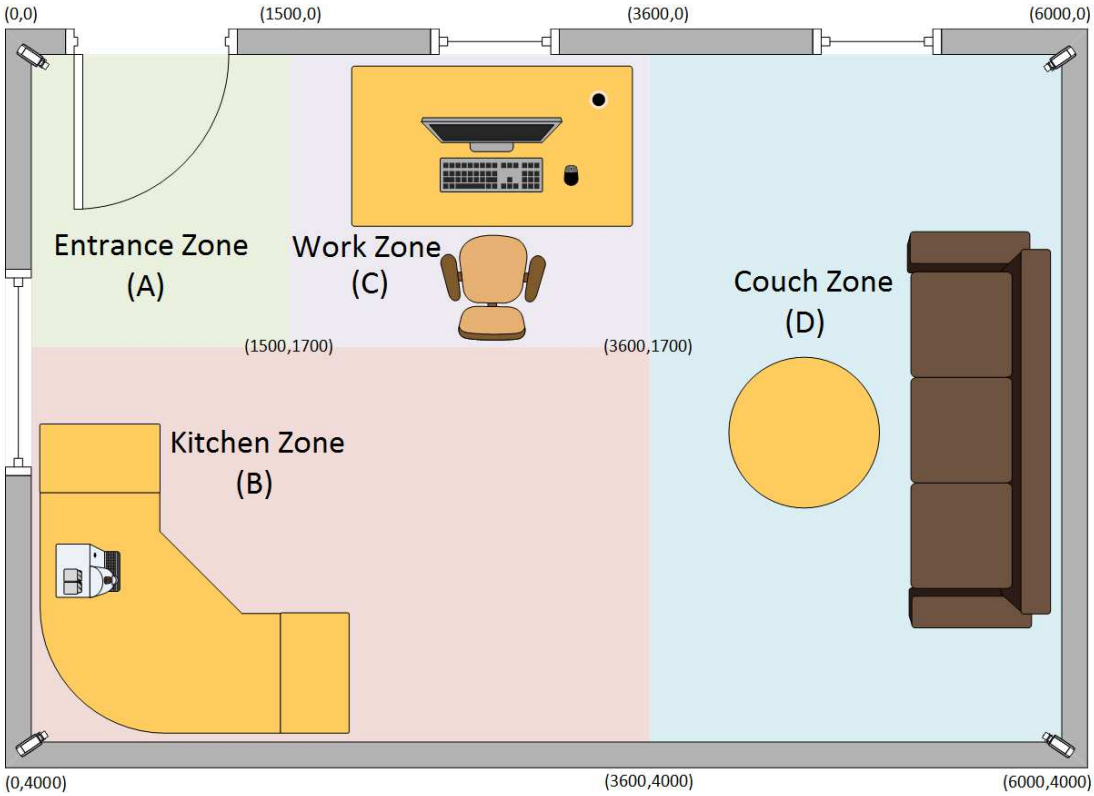

Figure 4.4: A map of the UoA-Indoor dataset test environment with a camera in each corner (co-ordinates given in mm)

4.1.1 UoA-Indoor

As a result, a small dataset was internally created called UoA-Indoor, which used four overlapping camera views as shown in Figure 4.3. The room was set up as shown in Figure 4.4 in order to emulate an indoor office environment. The room was divided up into four zones, each representing the different class of activities that would be expected in each zone. The motivation for doing this is that in real-world applications, knowing that someone is at a specific co-ordinate like (3210, 1820) is not useful information for humans. Rather, people prefer to abstract the co-ordinate away and replace it with a "zone" that helps locate the individual approximately. The advantage of such an approach is that it explicitly loosens the requirements on position precision, while still providing sufficient information for most human-environment interaction applications. This is an important factor that future researchers should consider when defining their datasets and determining what elements to include in annotation, and could give rise to an accuracy metric that is more relevant to non-technical users. In this dataset, the zones are defined based on co-ordinate points and labelled alphabetically {A,B,C,D} representing the entrance, kitchen, work, and couch zones respectively, which are human-readable labels that can be understood outside the context of computer vision research.

Footage was recorded at a resolution of 1920x1080 at 15 frames per second, using a script to initialise and synchronise all four cameras so that they record at the same time. For the purposes of further analysis, the footage was downsampled to 640x480 to reduce computation time. Consumer grade Logitech C920 webcams were used for the collection of the footage. The use of common webcams over more advanced camera equipment was an intentional decision to allow the setup to be more accessible, low-cost, and reproducible, as well as for any developed algorithms to be more robust and more likely to work with low-cost or legacy camera systems. While it has been argued that using stereo cameras or other cameras that provide some measurement of depth (e.g. RGBD, infra-red, etc.) can lead to higher accuracy levels [141, 142], single view cameras are still currently the most common in existing surveillance infrastructure, allowing single view camera-based algorithms to be more easily integrated. Based on the review of similar multi-camera person tracking efforts and their evaluation needs, seven action categories were developed, such as simply walking through the room (*Walk*), walking and sitting on the couch (*Break*), and a group of people talking to each other in the room (*Group*). This dataset currently contains over 165 minutes of footage collected with 19 different identities appearing across over 150,000 frames. All subjects gave their informed consent for inclusion before they participated in the study, and creating the dataset was approved by the University of Auckland Human Participants Ethics Committee (reference number 018182).

4.1.2 Annotation Tool

A number of annotation tools have also been released in recent years for annotating object and person tracking videos. At the most fundamental level, the annotation tool needs to produce a file that describes where a bounding box or bounding polygon for each relevant object or person is in each frame of the video. During the development of UoA-Indoor, it

became apparent that there was a need to create location annotations in both the image space and real world, along with attribute labels for the room zone and indication of whether a person was partially obscured or not. A number of existing tools were reviewed for their suitability for this dataset and the needs of the overall project. ViPER-GT [143] has a highly customisable annotation structure which could support various data types and labels, along with a time-saving feature to copy annotations across frames. However, its interface proved to be cluttered and annotations required too much manual effort compared to other tools as every frame still need to be manually annotated. VATIC [144] uses optical flow, a way of interpolating bounding boxes in the frames between two manually annotated frames, which are termed keyframes. However, the tool intentionally only offers bounding box annotations to simplify the annotation process, making it unsuitable for complex annotation tasks like incorporating calibration matrices or annotating zones. LabelMe Video [145] includes a 3D interpolation technique for tracking moving objects across frames, reducing annotation time while retaining a high degree of accuracy compared to more basic 2D interpolation. However, the annotation output is inflexible, and does not support the addition of location markers or other contextual information such as whether an object is obscured or not.

Since none of these tools specifically fulfilled the need for an annotation scheme that could include real-world co-ordinates and zones, a new annotation tool was developed from scratch, as shown in Figure 4.5. The annotations themselves are saved as a JSON file for each video, along with a global JSON file containing an array of person objects to assign a globally unique ID for each person in the ground truth. Most significantly, camera calibration is built into the tool, so that real-world co-ordinates can be automatically calculated based on image space co-ordinates and included as part of the annotations. Based on the real-world co-ordinates, the tool also automatically suggests a zone for the current person, the boundaries for which can be defined in a configuration file. The tool is targeted at more technical users and exposes the full annotation data to the user for editing, as opposed to only letting the user adjust bounding boxes. The user is also able to annotate the bounding boxes and the location separately (i.e. the user takes two passes through a video), which allows the user to concentrate on one element of the annotation process, and improves efficiency [144].

Reducing the number of annotations required is the most important factor in reducing overall annotation time. In this tool, the current frame annotations are automatically copied over to the next frame when the user progresses through the video in the same way as ViPER-GT [143], reducing the annotation time if few elements have changed. Keyboard shortcuts also allow the user to quickly move between frames without having to move the mouse to click on buttons. Additionally, a simple optical flow process is used to make estimates for bounding box locations between keyframes, similar to VATIC [144]. This was found to be a good compromise between automation complexity and time, as too much automation during ground truthing could favour one person tracking approach over others, which is a common issue with modern large-scale person re-identification datasets. The VATIC research suggests that keyframes should be fixed for optimal speed, but this was deemed to be too limiting as people tend to move in and out of view often in the videos in this dataset, and keyframes need

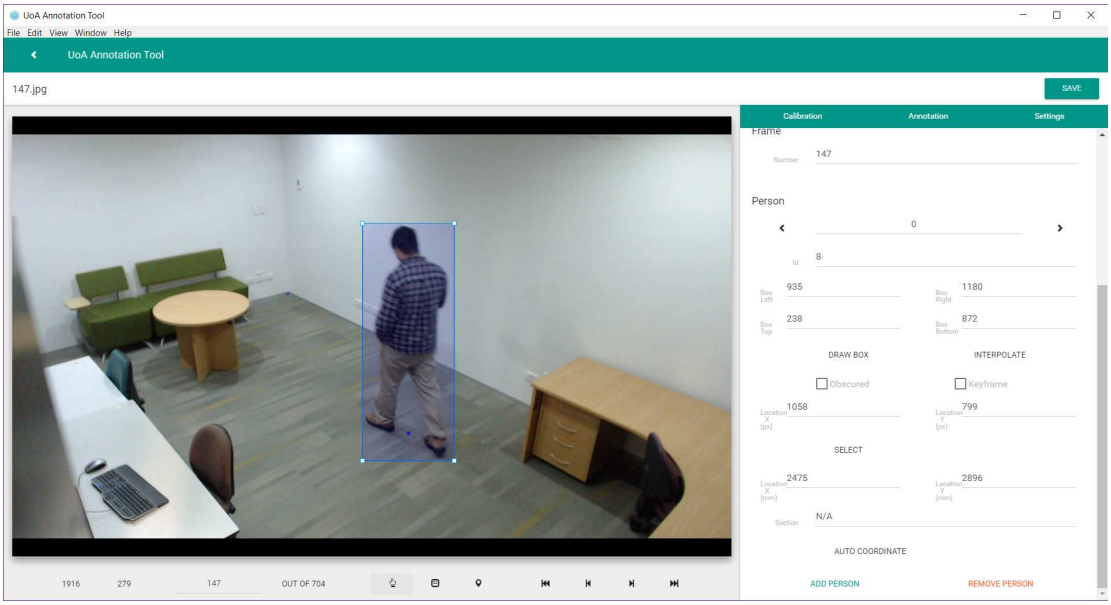

Figure 4.5: A screenshot of the annotation tool

to be marked more carefully if a transition happens. Keyframes are assumed when the user adjusts a bounding box or enables the keyframe checkbox, and then tracked for each person in the frame.

While it is desirable for keyframes to be set far apart in order to reduce annotation time, this is generally only possible in outdoor contexts where small objects move in reasonably linear trajectories (e.g. a car on a road) with relatively constant size. In an indoor context, since the cameras are closer to the target objects/people, the appearance size in image space changes much more frequently. To measure the error introduced by the optical flow, the Intersect Over Union (IOU) score was calculated between human annotated bounding boxes and optical flow generated bounding boxes. The maximum IOU was 0.991, the minimum IOU was 0.717, and the average IOU across approximately 400 frames was 0.942. The difference in annotation time was also measured, with an average 2 hours required per video for full manual annotations, compared to 1.4 hours required per video for annotation with keyframes. Therefore, a 30% reduction in annotation time was achieved in exchange for approximately 5% error, which would be considered an acceptable trade-off. This annotation tool is publicly available at `https://github.com/TheGuardianWolf/Annotation-Tool` for other researchers to annotate their own video data.

After the creation of the dataset, the OpenCV [75] implementation of the Deformable Parts Model (DPM) [85] detector was used to extract all instances of people in the dataset for the purpose of testing re-identification algorithms. This re-identification focused dataset contained approximately 28,000 samples across 19 identities with between 750 and 2640 samples per identity class, as well as approximately 13,000 hard negative samples, across all four camera views. It should be noted here that DPM is only used as a benchmark method for person detection and feature extraction so that metric learning and classification algorithms can be

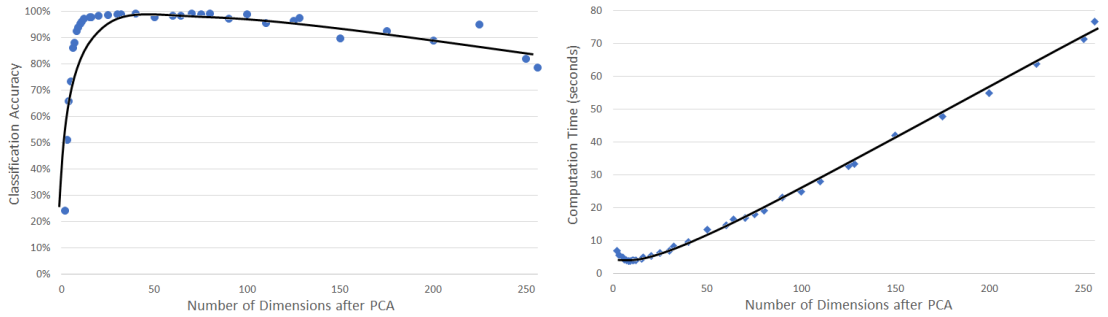

Figure 4.6: Classification accuracy (using 5-fold cross-validation) and computation time (training and testing time combined, per fold) from a linear one-vs-all SVM classifying approx. 28,000 feature vectors across 19 classes with between 2 and 256 dimensions after PCA and a baseline covariance metric transformation

tested fairly, and alternative methods include using another person detection algorithm and then segmenting using dense sampling [124, 146] or spatial transformer networks [147, 148] as an approximation for limb-specific segmentation.

4.2 Dimensionality Reduction

Based on this re-identification dataset, Principal Components Analysis (PCA) was used as an unsupervised method of determining the most important dimensions of the feature vectors in terms of variation - dimensions that have low variation are likely to correspond to background parts of the image that are common between all classes, while high variation dimensions are likely to help with separability between classes. PCA has been shown to be effective in a number of previous person re-identification applications [38, 112, 115]. Reducing the number of dimensions to a relatively small number of most important dimensions lowers the computation time for metric learning and classification as there is less data to process. Additionally, for some algorithms, reducing the number of dimensions can improve accuracy as uniformly weighted noisy and misleading dimensions can be removed, leaving the more important discriminative dimensions. For example, dimensions that correspond to the histogram bins that represent textures mostly only found in the background, such as on a wooden table, would be removed as they are likely to have the same values for two different people standing in front of a wooden table, and therefore would not help with inter-class classification. Once the most important dimensions have been established by processing enough training samples offline, this can be applied to all future feature vectors without any additional training. As the number of resultant dimensions increases exponentially, the increase in classification computation time is also generally exponential (in the best case where the classification algorithm is linearly complex) while the accuracy improves diminishingly or even decreases as more dimensions are introduced, as shown in Figure 4.6. This figure shows the result of an experiment using an offline SVM for classification, where between 32 and 64 dimensions were generally sufficient for good linear separability.

4.3 Metric Learning

After PCA, the detected people are represented by short feature vectors based on the features/variables/dimensions with the highest level of variation. However, these vectors are often not linearly separable, which makes classification more challenging, and no effort is made to keep samples from the same class together. While it is common to use non-linear kernels such as Radial Basis Functions (RBF) to achieve non-linear classification, this can be very computationally expensive, and can still find some data distributions very challenging. Metric learning (also known as distance metric learning) attempts to simplify this by learning a distance metric, generally a Mahalanobis distance metric, that optimises towards establishing high inter-class (between-class) variation and low intra-class (within-class) variation. Metric learning has been shown to be a useful pre-processing step for many machine learning approaches [149], and has been shown to improve the accuracy of person re-identification [38, 150, 151] by transforming the vectors so that they are more linearly separable into identity classes, which has the effect of increasing the accuracy and reducing the subsequent training and testing times required by the classification algorithms. For more detailed information about metric learning methods and their applications, particularly in the context of processing feature vectors, readers can refer to the survey technical report [152].

By including annotated images of detected people across a number of different camera views, the aim is to use the metric learning algorithm to learn how to minimise the impact of the different camera characteristics by minimising intra-class variation, where the class is a single identity that has samples across multiple camera views. This is in contrast to approaches that explicitly learn a transfer function that models the change in parameters between two camera views [153, 154] so that the appearance of a person in one camera view can be modified to be more similar to what might be expected if they had appeared in another camera view. The challenge for real-world re-identification is that a metric has to be learnt that is applicable to classes that may not have seen at training time, as new people enter and exit the monitored environment. Thus far, the literature has not shown there to be a "universal" distance metric that can be applicable in all cases, as it is common for the learned metric to be dependent on camera parameters, external environmental factors, and variations in background [155]. While the optimum metric could be learned if many samples of all classes were available at training time, in reality the metric has to be found based on a small number of training classes that can then be used on other new classes at test time, similar to how the important dimensions are established through PCA. This is further complicated by the fact that there are many, many different metric learning algorithms, and there is no consensus yet on which algorithm is the most suitable for person re-identification.

4.3.1 Algorithms

In order to decide on which metric learning algorithm to use in the proposed video analytics system, a number were experimentally tested on the dataset to find not only their ability to learn a useful metric, but also to learn a generalisable metric based on a small number of classes.

The tested algorithms are selected based on their usage in related applications, showing good results in either person re-identification or similar image/signal processing applications. A brief description of each of the tested algorithms is provided here, but readers should refer to the cited sources for more details.

Large Margin Nearest Neighbour (LMNN) [156] uses a k-Nearest Neighbours (kNN) approach to try and keep the closest points together in the same class, while separating classes with margins that are as large as possible using semidefinite programming. This approach is similar to an SVM in that there are similar goals to maximise linear margins between classes, except it is based on kNN instead of a linear kernel. In these experiments, k is set to be equal to the size of the smallest class. LMNN and its derivatives are some of the most popular metric learning algorithms, and have previously been applied to person re-identification in [157, 158, 159].

Information Theoretic Metric Learning (ITML) [160] views the samples of each class as a Gaussian and attempts to minimise the differential relative entropy, essentially bringing the samples for the same class closer together and inherently creating larger distances between the classes. This approach does not require semidefinite programming or expensive eigenvalue computation, but may require additional linear constraints to work effectively. ITML and its improved derivatives have been used for facial recognition/verification and person re-identification in [150, 161].

Sparse Determinant Metric Learning (SDML) [162] attempts to take advantage of the typically sparse nature of data points in high-dimensional feature spaces, using regularisation approaches to learn a compact distance metric in a computationally efficient way. However, since PCA is performed as a form of dimensionality reduction first, some of the sparsity assumptions about the ratio of samples to dimensions may not hold true.

Least Squares Metric Learning (LSML) [163] is designed to minimise the sum of squared residuals in order to improve clustering accuracy by using a simple gradient-descent approach. It is also designed to work in sparse dimension spaces, and has been shown to work well in scenarios with minimal supervision. LSML is applied to handwritten character recognition in the original paper, but it is likely to be suitable for other image-based matching or re-identification applications since sparsity is a common characteristic of some dimensions in the feature vectors. LSML has been used in diverse image classification applications in [164] and face recognition in [165].

Local Fisher Discriminant Analysis (LFDA) [166] solves a generalised eigenvalue problem in order to perform supervised dimensionality reduction, and thereby produce separable classes. In particular, it is targeted at scenarios with multimodal distributions where the same class is actually distributed across multiple clusters (or in other words, the cluster may be made up of multiple separable sub-clusters), which might be expected when drawing samples from multiple camera views. It has previously been applied in a person re-identification context in [117].

Relevant Components Analysis (RCA) [167] reduces global unwanted variability by assigning larger weights to relevant dimensions and lower weights to irrelevant dimensions

[168]. RCA does this based on "chunklets", defined as a subset of points belonging to the same class, in order to identify intra-class variability and minimise the weighting for those dimensions. RCA has been applied to facial recognition in [169].

Lastly, **Neighbourhood Components Analysis (NCA)** [170] is based on a kNN approach, similar to LMNN, except it maximises an accuracy score based on the training data by iterating over stochastic solutions. However, NCA faces substantial computation and memory challenges with datasets that have high-dimensionality and a large number of samples, and can become trapped by local maxima. Both RCA and NCA have been applied to handwritten character recognition [170], as well as face recognition and character recognition in [171].

The recently popular KISSME [172] algorithm was investigated, but it was not included in these experiments as [117] showed that it produced similar results to LFDA on person re-identification datasets, with LFDA outperforming KISSME on smaller datasets with fewer identity classes. In addition to the metric learning approaches described above, a **covariance metric transformation** was included as a baseline method for comparison, based on the original Mahalanobis work [173]. The covariance metric essentially applies a standard covariance operation over the feature vector:

$$\Sigma_{ij} = \mu[X_i X_j] - \mu_i \mu_j \tag{4.1}$$

where Σ is the output matrix, X is the input feature vector, μ is the mean, and i and j refer to the positions of elements within the vector/matrix. For the rest of this Chapter, "metric learning algorithms" includes the covariance metric in this collective group, even though strictly speaking there is no learning involved in this calculation. It is also important to include a control case in these experiments, where no metric learning is applied at all, in order to show that metric learning does produce better results than doing nothing.

4.3.2 Experimental Results

4.3.2.1 UoA-Indoor Dataset

Two experiments were set up to test the effectiveness of different metric learning algorithms based on UoA-Indoor. In both cases, post-PCA feature vectors were used (32 dimensions). Firstly, in Experiment A (shown in Table 4.2), 5-fold cross-validation (CV) was used, where the metric learning algorithm had access to 80% of the samples across all classes as a training set. It should be noted that these samples come from all four camera views, so there may be significant intra-class variation from the different camera views that needs to be minimised. The learnt distance metric was then applied to all of the samples, and made the transformed training set available to a linear one-vs-all Support Vector Machine (SVM), and tested this with the remaining 20% of the transformed samples as a test set, evaluating the accuracy and timing across five folds. The samples used in the training and test set were selected uniformly randomly, rather than being in sequential time order. In all experiments, the accuracy is essentially the "Rank 1" result, i.e. only the highest ranked result is used for classification with no further

Table 4.2: Experiment A: Metric Learning with 5-fold CV across all classes, evaluated with SVM on UoA-Indoor

Method	Accuracy (%)	Metric Learning Time (s)	SVM Training Time (s)
None	9.44	0	58.53
ITML	9.44	7.84	56.27
SDML	9.44	0.019	56.23
LFDA	90.25	1.03	19.67
RCA	96.92	0.0084	60.52
Covariance	99.65	0.011	5.98
LSML	99.72	3.82	5.99
NCA	N/A	N/A	N/A
LMNN	N/A	N/A	N/A

Table 4.3: Experiment B: Metric Learning with a subset of classes, evaluated on 5-fold CV with SVM on unseen classes on UoA-Indoor

Method	Accuracy (%)	Metric Learning Time (s)	SVM Training Time (s)
None	18.66	0	17.78
LFDA	55.01	0.49	1133.59
RCA	58.56	2.95	82.62
LSML	96.07	3.86	17.58
Covariance	98.82	0.0061	8.38

re-ranking. This is in contrast to many papers in the literature that report accuracy at multiple ranks, which is not actually representative of what is needed for the real-world task.

After determining which metric learning algorithms were most likely to be suitable, in Experiment B, shown in Table 4.3, 5-fold cross-validation was used to provide four random classes (21% of all classes) to the metric learning algorithm, and the learnt distance metric was applied to all samples across all classes. The remaining transformed fifteen classes were then classified with a linear one-vs-all SVM. For each class, 20% of the transformed samples were provided to the SVM for training, and the other 80% for testing. Using 5-fold cross-validation in this way allows us to confirm that the effect of metric learning was not closely tied to the specific classes that are chosen for training and testing. This experiment differs from Experiment A and the work done in [150] in that not all classified classes are made available to the metric learning algorithm, so the test is determining the generalisability of the learnt distance metric on different classes in the dataset. The methods that were considered unsuccessful in Experiment A were not included again in Experiment B.

In these experiments, the best case scenario is to find a metric learning algorithm that leads to high classification accuracy and low computation times, both in terms of metric learning and classification. The metric learning time is the time taken for the metric learning algorithm to return a transformation matrix - the time required to apply that transformation matrix to the data is constant for the same number of samples and negligibly small, regardless of the metric learning algorithm. Only the SVM training time is reported here for the classification process - the time required to evaluate (i.e. classify at run-time) each sample of the testing data is also constant for the same number of samples regardless of the metric learning algorithm, since the SVM just evaluates which side of the hyperplanes the sample lies on. It is important to note that these times are provided for processing the entire dataset, not per frame or per detected person. The metric learning and SVM training times should be interpreted comparatively between the different methods rather than taken as universally applicable computation times, as they will vary significantly depending on the computation platform and dataset. All accuracies and computation times are derived from a desktop computer running Ubuntu 16.04 with an i7-6700 CPU with four 3.40GHz cores and 64GB of RAM. This is a far more powerful computational platform than would normally be expected to be available in low-cost resource-constrained environments, but is used as a preliminary testbench. Even then, a number of the metric learning algorithms crashed when the PC ran out of memory, indicating that they are unlikely to be suitable for the resource-constrained use case. In particular, LMNN and NCA were not able to complete execution on the test computer, and RCA crashed when the number of post-PCA dimensions increased to 64. The code was written in Python with NumPy to improve computation speed, and made use of the *metric-learn*[16] library for some of the algorithm implementations in order to accelerate development and testing. In each experiment, there are 19 identity classes and a hard negative class containing erroneous background samples that are not people.

As shown in the results for Experiment A in Table 4.2, the use of an appropriate metric learning algorithm clearly makes a large impact on the linear separability of the feature vectors, improving the accuracy significantly. In the case where no metric learning is used, the SVM seems to classify all of the test set vectors as being part of the hard negative class, resulting in a baseline accuracy of 9.44%. ITML and SDML show similar results, indicating that they have failed to learn a meaningful metric and have become biased towards the class with the largest number of samples (the hard negative class). When LSML, LFDA, or RCA are used, accuracy rises to over 90% using the same classification method. Calculating the simple covariance metric of the reduced feature vector also leads to a high classification accuracy.

There is also large variation in the metric learning times between the different approaches. This is mostly dictated by how many iterations the approach needs, and whether the error can be reduced sufficiently or if the algorithm reaches the maximum number of iterations before exiting. However, it is also interesting to observe the differences in SVM training time, which can be interpreted as a sign of how successful the metric learning algorithm is at producing linearly separable classes, as it represents how easy or difficult it is for the SVM to optimise

[16] https://github.com/metric-learn/metric-learn

hyperplanes between classes. High training times are a sign that learning a suitable hyperplane has been very difficult for the SVM, and in some cases leads to a very low accuracy when the SVM has failed to improve over time. This is with the exception of RCA, where the end accuracy is relatively high, but it requires a high number of iterations, as shown in the very high SVM training time.

In terms of optimising towards high accuracy and low computation time, it is clear that LSML should be the winner in Experiment A, but the covariance metric performs very similarly with a much lower metric learning time (since there is no iterative learning process involved). These results are reinforced by the results of Experiment B, shown in Table 4.3. In this case, the accuracies generally drop in comparison to Experiment A, which is to be expected since the metric learning algorithm has not seen all possible classes. The accuracy for the case with no metric learning at all increases since there are fewer classes (and samples) in the SVM train/test sets. The SVM training time increases, sometimes very significantly, since the learnt metric is likely to be less suitable for classes that have not been seen by the metric learning algorithm, and therefore the feature vectors are harder to separate. Experiment B shows that the high accuracies achieved by LFDA and RCA in Experiment A are highly dependent on the samples seen at training time, suggesting that these approaches tend to overfit. This may be suitable for some applications where all classes are available at training time, but these are generally not available. LSML is left as the only learning approach that achieves a high accuracy when the distance metric is applied to new data containing new classes, but surprisingly the covariance metric, without any learning, achieves an even higher accuracy.

LSML and the covariance metric transformation produce similar results because in LSML, the covariance metric (i.e. converting to the Mahalanobis Distance) is used as an initial guess, which is then further iterated upon to reduce the error - since the covariance metric is already a good guess, further iterations only lead to marginal gains. However, a 0.07% increase in accuracy shown in Experiment A probably should not be accepted when it comes at the cost of a 360$\times$ increase in metric learning time. In Experiment B, LSML actually achieves a lower accuracy than the covariance metric, because it has optimised towards the classes available at training and not performed as well for the new test classes during classification (essentially overfitting), also leading to a significantly increased SVM training time. Therefore, more computationally expensive metric learning can be avoided by using a simple covariance metric transformation instead to achieve sufficiently good linear separability between identity classes. The use of the covariance metric as a feature descriptor has been established in the literature [71, 174], but not necessarily in terms of colour histograms or as a superior alternative to more sophisticated metric learning. In general, this is a finding ‘supported by [150], which states briefly in their conclusion that “even a standard Mahalanobis metric not using any discriminative information yields quite good classification results”. It is logically consistent that this effect is due to the metric learning algorithms generally overfitting to the training data, and the covariance metric remains relatively generalisable while being far less computationally expensive.

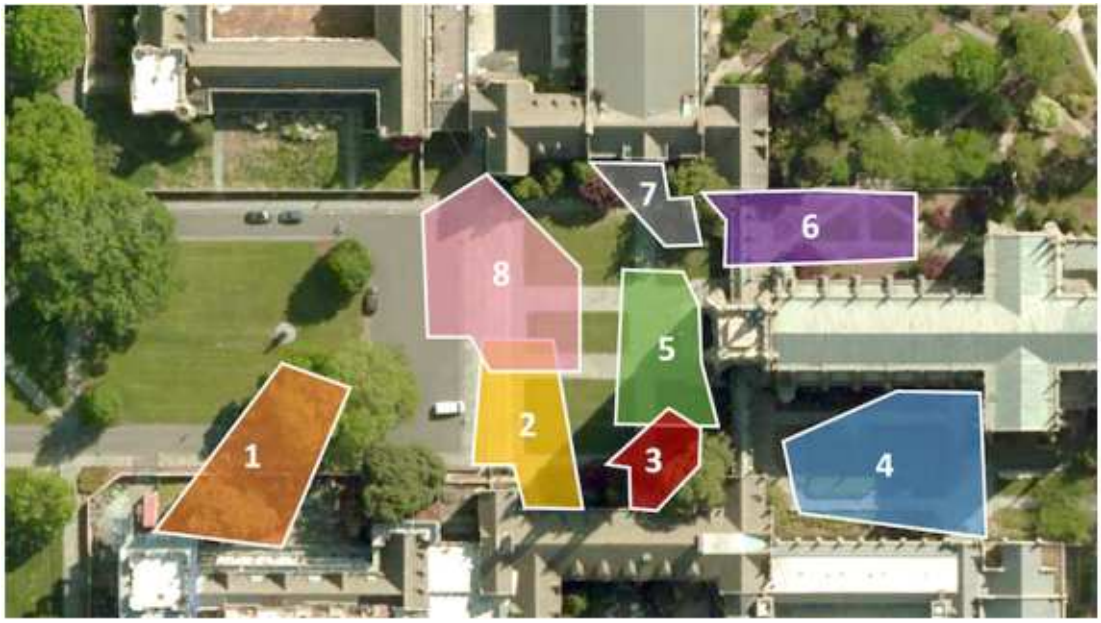

Figure 4.7: A map of the camera layouts in the DukeMTMC dataset (from [121])

4.3.2.2 DukeMTMC4ReID Dataset

To further validate the results, the experiment was repeated on a much more challenging dataset, DukeMTMC4ReID [121], which is based on individuals detected with Doppia [175] from the DukeMTMC dataset [39]. In this dataset, there are 1852 unique individuals (classes), with a total of 46,261 detections at varying bounding box resolutions. This is a more traditional person re-identification dataset in comparison to UoA-Indoor, with a small number of detections per class (as few as 10), making training much more difficult. Additionally, the dataset is captured in an outdoor setting, with eight cameras that are mostly not overlapping (as shown in Figure 4.7), with significant variation in pose and differing environmental conditions. In this dataset, the individuals are already cropped, which makes it harder to use DPM to consistently extract person parts without significant retraining. Instead, patch-based sampling [123] is used to segment the cropped person images by dividing each person image into 16 parts in a regular 4x4 grid. While the image resolutions differ between detections (depending on if the person was closer or further away from the camera when they were detected), leading to differently sized parts, this does not affect this approach since the extracted features are normalised histograms for each part. The accuracy of this approach would likely be improved with more dense sampling [146], but for the purposes of this experiment, comparative measures are more important. The same colour and texture features are extracted as before - since there are now 16 parts per person instead of 8, the resultant feature vector is twice as long. Without dimensionality reduction, this would have significant impacts on the classification algorithm and its ability to deal with noisy data in more dimensions, but PCA is used to reduce the number of dimensions down to 32 to be consistent with the previous experiments.

Typically, a simple one-vs-all SVM would probably not be used in this scenario as there

Table 4.4: Experiment C: Metric Learning with a subset of classes, evaluated on 5-fold CV with SVM classification on unseen classes on DukeMTMC4ReID

Method	Accuracy (%)	Metric Learning Time (s)	SVM Training Time (s)
None	0.0896	0	323.25
LFDA	0.0899	0.048	326.37
LSML	22.28	11.43	365.95
Covariance	22.68	0.0096	375.77
RCA	N/A	N/A	N/A

are too many different classes, leading to state space explosion and a high level of difficulty in separating the classes linearly. This makes following the SVM training time less reliable as an indicator of the impact of the different metric learning algorithms on separability, as the effect of the large number of classes dominates, as shown by the accuracy rates. However, the SVM is used as a benchmark method for evaluating the different metric learning methods so that there is fair comparison. This experiment follows the same procedure as Experiment B, where not all classes are seen by the metric learning algorithm in order to determine the algorithm's ability to identify a generalisable metric for a given environment and emulate the real-world case. In this case, 10% of the total classes are provided to the metric learning algorithm (because there are far more classes in this dataset than UoA-Indoor), with the expectation that the learnt metric should be applicable to the other 90% of classes too. This experiment tested the same metric learning methods as Experiment B, and used 5-fold cross validation to help with establishing more robust results since there is less data per class. It is important to note that the training times reported are aggregated across the entire training set in the dataset, not per detected person, so the computation times are naturally higher than in the case of UoA-Indoor since the dataset is larger.

As shown in Table 4.4, without metric learning the SVM essentially fails to learn. RCA failed with the larger dataset because of resource constraints. The increased number of classes and reduced number of samples per class causes the accuracy rate to generally be very low for all methods. The covariance metric still produces the best distance metric, with the best accuracy and the lowest metric learning time, while producing a similar SVM training time to the other methods. As before, LSML was very close to the covariance metric since it starts with the covariance metric as an initial guess and then converges towards a solution optimal for the given training set. Since the learnt transformation matrix is then applied to other previously unseen classes (in the same environmental settings), this causes the learnt matrix to move away from the more general solution.

The accuracy rate can be compared to the results reported in the original DukeMTMC4ReID [121] paper as rank 1 results. This has to be done carefully, as [121] uses different features and a different evaluation protocol to the previous experiments. Firstly, their work compares seven

Table 4.5: Comparison against DukeMTMC4ReID experiments from [121] (LOMO features)

Method	Accuracy (%)
KISSME [172]	0.49
SVMML [176]	9.61
PCCA [177]	14.16
MFA [178]	14.92
L2-norm	20.6
Covariance (from Exp D)	22.68
RankSVM [134]	23.19
XQDA [118]	27.88
LSML (from Exp D)	28.12
kMFA [117] with exp kernel	38.29
kLFDA [117] with exp kernel	38.56

different types of features, with varying colour and texture choices. LOMO [118], which extracts HSV colour histograms and LBP texture features, is the most similar to the features used in the previous experiments. Secondly, their evaluation protocol selects 50% of the identities as the training set, rather than the 10% used in the previous experiments, so it is natural for the accuracy rates to be much higher since the training data represents a larger proportion of the global distribution. It is important to note that this is because [121] is only focusing on presenting baseline results for their new dataset, whereas in this Chapter the focus in on finding a metric learning algorithm that can learn a generalisable metric transformation with minimal training data, hence the difference in experimental design. To resolve this difference, for the purposes of comparison only, experiment C was re-run with only LSML and the covariance metric (experiment D), with 50% of the samples provided to the metric learning algorithm. Naturally, since there is no actual learning in the covariance metric transformation, the accuracy here did not change, which disadvantages it against the other algorithms presented that have more opportunity to learn a metric that suits the distribution of the dataset. Additionally, rather than mixing samples from all cameras for both testing and training, [121] fixes one camera as the probe, and detections for all other cameras are combined as the gallery. Importantly, no offline SVM is used in their experiments; instead, they simply pick the most similar source image for each probe and classify in that way. In Table 4.5, a rough comparison between some different metric learning algorithms on DukeMTMC4ReID is shown - other than the covariance metric and LSML, the results are obtained directly from [121].

The first interesting result to interpret is the increase in accuracy for LSML - from 22.28% under the previous evaluation protocol to 28.12%. This is an indication of the difference in accuracy that can be expected simply by changing how the accuracy is measured; in this case the most significant impact is caused by the much larger amount of training data used. Secondly, the covariance metric result is still reasonably competitive against other, more complicated

methods. In particular, the SVM-based methods SVMML and RankSVM are likely to require significant training time, and the kernel-based methods kMFA and kLFDA may not produce linearly separable classes that can be interpreted by a linear SVM. Unfortunately, no timing results are reported in [121], so it is not possible to identify the amount of speed that must be given up in exchange for higher accuracy rates. Given the focus on computationally light algorithms in this thesis, that the target application requires a more generalised metric that can be applied without training data that represents a large proportion of the global distribution, and that the covariance metric is not significantly worse than most of the other state-of-the-art methods, the covariance metric is perhaps the best option for making the feature vectors easier to classify in the proposed re-identification scheme. This is an important result, as other researchers have continued to develop and use more sophisticated metric learning approaches for person re-identification.

In this Section, it is comprehensively established that the covariance metric transformation is sufficient for fast person re-identification across multiple camera views; this could be further established on other datasets to determine the wider generalisability of this claim. The primary reason for the covariance metric transformation's superior performance is due to the more sophisticated metric learning algorithms generally overfitting to the training data, failing to remain generalisable enough for new unseen test data. In particular, this claim may be more applicable to real-world re-identification scenarios, where there are a large number of samples per class as they are observed over time, and a smaller number of classes visible over any reasonable time period, rather than the standard dataset formulation of a large number of classes with a very small number of gallery images per class. However, the covariance metric is still competitive against more sophisticated metric learning approaches on a large standard person re-identification dataset.

In terms of future work, there are other metric learning approaches that could be investigated. Newer methods such as Marginal Fisher Analysis (MFA) [178], Pairwise Constrained Component Analysis (PCCA) [177], and Cross-view Quadratic Discriminant Analysis (XQDA) [118] may perform better, but there is no indication that they are able to overcome the overfitting problem suffered by the existing metric learning algorithms; [179] alludes to MFA requiring all subject identities during training in order to yield a good metric, and the data from DukeMTMC4ReID in Table 4.5 would indicate poorer performance even with 50% of the identities made available during training. Deep transfer metric learning [155, 180] allows for a partially trained model to incorporate new distributions from new datasets quickly to learn a more specific metric for a new environment. X^2-LMNN metric learning [181] has been shown to be particularly suitable for histogram-based data. However, both methods are all likely to be more computationally expensive than the covariance metric while still overfitting, and the covariance metric already yields sufficiently good results.

4.4 Classification

Once a detected person has been passed through feature extraction, dimensionality reduction, and metric learning, the resultant feature vectors can then be classified. The difficulty lies in the fact that not only is there significant variation in the appearance of a person between camera views, but even within the same camera view, environmental factors such as lighting can have an impact. Within-class variation makes classification challenging, especially if there are overlaps between different classes. An example of this is when people are standing directly underneath a bright light - depending on the angle of the camera, this can result in the appearance of the person being saturated (i.e. a lot of the pixels are at their maximum values, representing white), making it difficult to visually distinguish between two different people. The aim of the classification step is to attempt to overcome these issues and correctly identify the individual based on a feature vector representing their appearance.

In this problem formulation, each class represents a single identity, and the aim is to classify the transformed feature vectors into classes. This is in contrast to approaches that attempt to learn a transfer function between the different camera views in an effort to model the environmental and parameter changes [10, 153, 154]. There are a few characteristics of the end use case that affect the choice of classification algorithm. Firstly, while there is a dataset available for algorithm development, in the real world, training data can not be obtained for each possible individual that might enter an unconstrained public camera view. This means that most supervised methods will be unsuitable, especially those that require large amounts of training data such as deep convolutional neural networks. Instead, the preference is for methods that either can utilise transfer learning based on the dataset, or unsupervised methods that do not require labelled data for each individual [182]. Unfortunately, purely unsupervised methods are unlikely to be suitable for processing footage in real-time, as stream processing (i.e. processing the footage as it arrives) means that the system will not have all of the samples at classification time, so finding patterns in the overall distribution is not possible. As a compromise, there are algorithms that require a minimal number of training samples. Algorithms that only use a single sample during training are commonly known as one-shot learning methods [183]. Secondly, in low-cost environments there is generally limited computation power and memory, so classification methods that require expensive GPUs or cloud computation services in order to achieve real-time speeds are also unsuitable. It is important to keep in mind that any algorithm needs to be scalable as the number of identity classes increases (up to a point), so methods that suffer from state space explosion such as decision tree-based algorithms would also be unsuitable.

4.4.1 Algorithms

This Subsection investigates four potentially suitable one-shot learning approaches, as well as two supervised approaches and one semi-supervised approach as benchmark comparisons to show that a) the feature vectors can theoretically be classified accurately if there is sufficient training data and b) the difference in computation time between fully supervised and one-shot

learning approaches. This is important as it shows that any errors for the one-shot learning methods are attributable to the algorithm, rather than to the data.

4.4.1.1 Benchmark Algorithms

The first method was a linear **Support Vector Machine (SVM)**, which is commonly used in machine learning as a supervised method for classifying data in two classes. This is done by optimising a hyperplane in multi-dimensional space between the classes so that the margin on each side of the hyperplane is as wide as possible. In the multi-class case where there are many possible output classes, a number of different approaches have been suggested. Variations of RankSVM [106, 134, 146] have previously been used in the re-identification context, where the kernel trick is used to learn a ranking function in a high dimensional space, which would also absolve the need for metric learning. However, simpler multi-class (pairwise or one-vs-all) SVMs have been shown to produce reasonably good results already, at much lower computation costs during execution [184]. A standard SVM formulation is also more suitable in sparse cases, i.e. where the number of dimensions is low relative to the number of samples, as is the case when dimensionality reduction has already been performed on the data. In these experiments, a one-vs-all scheme is used, where one model is trained for each class and the decision functions are used to determine the most positive model for any given input sample.

A simple **Perceptron** was used as another supervised method to provide a comparison to the SVM. A single perceptron with many output classes can suffer from separability issues, especially if there is significant noise or variation at the input, so a one-vs-all scheme is used to match the SVM model, where there is a single perceptron trained for each class. A multi-layer perceptron (MLP) network was also tested, but the relatively small number of input dimensions made selection of appropriate parameter values challenging, with a very low accuracy as a result. Deep neural networks were rejected for similar reasons to the MLP, on top of the high computation costs that might be unsuitable in a resource-constrained environment and the assumption that the additional computation costs were unlikely to yield sufficient accuracy improvements on top of the high accuracy rate already achieved by the single perceptron. Normally, the deep neural network would work on raw images and expect much more data than is being made available with the transformed feature vectors.

A semi-supervised method was included in order to provide another comparison point in terms of accuracy and computation time when less training data is made available. In **Label Spreading**, a similarity graph is constructed across all of the samples given in the training set, with regularisation so that the resultant model is more robust against noise. Importantly, only a small number of the training samples need to be labelled; the other samples are marked as unlabelled, which would normally be useless for a supervised learning algorithm, but in a semi-supervised algorithm allows the model to still develop some understanding of the underlying data distribution and lead to more accurate classification than unsupervised algorithms. For these experiments, the *scikit-learn* [185] implementations of SVM, Perceptron, and Label Spreading were used to accelerate development.

With the supervised and semi-supervised methods, there is no model update step, so the model does not improve over time and new classes cannot be introduced at runtime. However, as indicated earlier, the target application likely requires unsupervised methods. One-shot learning is very similar to unsupervised learning, in that one sample is provided per class to help start the unsupervised learning process. Firstly, a one-shot formulation of the SVM and perceptron can be used by providing an anchor sample for each class to initialise the model, and then using online learning to both classify and continue to train the model over time as samples arrive, with early stopping to avoid overfitting. However, as with most unsupervised methods, there is a risk in that misclassification of samples has a significant impact on the accuracy of future classification, since incorrect and correct classifications contribute equally to the learnt model. In the case of the SVM and perceptron, a single misclassification has a potentially large impact due to the repeated training epochs performed with each sample. Instead, the following subsections present two novel methods that attempt to minimise the impact of misclassification in one-shot/unsupervised learning. Both of these methods follow the flowchart shown in Figure 4.2, where identity classification is comprised of a similarity calculation with the existing model descriptors, and then a model update step to improve the accuracy of matching over time.

4.4.1.2 Gallery

The ViBe classification method was originally developed for the purposes of background estimation [60], to determine whether a pixel is part of the background or foreground based on comparison with samples in a background model. Importantly, the classification is reasonably robust against noise, with noisy samples ageing out over time. This was adapted to work in a person re-identification context, removing the irrelevant parts such as sample propagation for spatial consistency. This is called the **Gallery** approach because it is primarily based on maintaining a gallery of feature vectors for each identity class. This algorithm has two main parts - a classification step and a model update step.

When a new person is seen for the first time, a new class is created and the extracted feature vector is used as an anchor. A gallery of N samples (*targets*) is established, where all of the samples are initially identical. During the classification step, the new sample (the *probe*) is compared to each target sample in the gallery for each class. Since metric learning has already been applied, theoretically the data points now lie in Euclidean space, and a simple Euclidean distance can be used to determine the similarity between two feature vectors, where lower distances mean that the vectors are more similar. A maximum threshold for the distance (DIS) can be found using a parameter sweep, so that if the distance is below that threshold then the two feature vectors are considered to be a match. For each class, if the number of matches is above a minimum number ($numMin$), then the probe feature vector is classified as part of that identity class. Formally, this classification scheme can be described by saying that a probe feature vector $\mathbf{x}$ is part of class c if $C(\mathbf{s}) >= numMin$, where $C(\mathbf{s})$ is the number of target feature vectors $s_0, s_1, ..., s_N$ that satisfy the condition $||\mathbf{s} - \mathbf{x}|| < DIS$, where $||\mathbf{s} - \mathbf{x}||$ denotes

the Euclidean distance between $\mathbf{s}$ and $\mathbf{x}$.

In the model update step, a random target sample in the class gallery is replaced with the probe sample. Each target sample has a uniformly equal chance of being replaced. This has the effect of creating a smooth decaying lifespan for the gallery samples, providing a mixture of recent and older samples to compare against. The constraint on the size of the gallery to N samples ensures that the method remains scalable and does not grow excessively over time. The impact of a single misclassification leading to replacement in the class gallery is therefore minimised, as multiple matches (up to $numMin$) need to be made for a future probe vector to be classified. However, this behaviour is non-deterministic due to the random replacement, leading to slight differences in performance between trials. To determine the appropriate parameter values, a simple parameter sweep was conducted, and for the UoA-Indoor dataset the best values were N = 40, DIS = 40, and $numMin$ = 5. In other words, for a probe sample to be classified, it needs to match with at least 5 out of the 40 gallery samples, where the two distance vectors need to have a distance less than 40 in order to be declared a match.

4.4.1.3 Sequential k-Means

In this approach, a modified form of k-means clustering that supports online learning to classify feature vectors is used, where each class/cluster is a new person. This is based on Sequential k-Means Clustering[17], but there are significant differences to allow this algorithm to work in a re-identification context. Originally, in the development of this system, this algorithm was designed to work with the DPM output without dimensionality reduction or metric learning, determining the distance between pairs of body parts and then creating an overall similarity score. However, this required a heuristic model update step that was similar to Relative Components Analysis (RCA), but was not very effective because the streaming nature of the video footage meant that the learnt model would become heavily biased towards earlier classes. That heuristic is no longer needed because the Mahalanobis distance (covariance metric transformation) is calculated before classification. An additional benefit of using the metric learning transformation is that instead of having to compute similarity using correlation distances (since originally the distances were based on histograms), a linear Euclidean distance can be used, which is computationally lighter.

In Sequential k-Means, each class is represented only as a cluster mean instead of retaining all of the data points, significantly reducing the memory and computation requirements in comparison to most other classification methods. When a new person appears for the first time, the first sample is used to initialise a new cluster center with that feature vector. For each new probe feature vector, it is compared to the cluster mean for each existing class. The vector is classified into class c with the lowest Euclidean distance, defined as $||\mathbf{m}_c - \mathbf{x}||$, where $\mathbf{m}_c$ is the cluster mean for class c. It should be noted here that this has the potentially negative effect of creating spherical clusters and decision boundaries, in comparison to most of the other methods which have linear boundaries.

[17] Richard Duda, "Sequential k-Means Clustering", Princeton University, 2008.

The selected cluster mean is then updated using $\mathbf{m}_c = \beta\mathbf{x} + (1-\beta)\mathbf{m}_c$, where $\mathbf{x}$ is the new probe feature vector and β is a constant weighting factor. This essentially becomes a form of linear filtering, where each new sample contributes to the new mean by a small amount while still allowing past samples to have most of the influence. In a parameter sweep with the UoA-Indoor dataset, β = 0.1 was found as a parameter value that led to good accuracy results with images from multiple camera views, although this is likely dependent on the rate of change in visual appearance, which is difficult to model and dependent on the dataset or real-world scenario. The optimal value of β is ultimately a trade-off between the need to update the model so that it is more similar to future samples in the short-term while remaining robust against noise and erroneous classifications in the long-term.

4.4.2 Experimental Results and Discussion

4.4.2.1 Experimental Settings

To compare the different classification approaches, each method was run on the transformed feature vectors from UoA-Indoor after PCA (32-dimensions) and metric learning (covariance metric transformation) with the following procedures and parameters:

- For the supervised methods, a modified version of 5-fold cross-validation was used to determine the accuracy and computation time per fold, with the smaller number of samples in the training set and the larger number of samples in the test set to make the problem setting slightly more challenging. The offline SVM and Perceptron were both linear. In each fold, there were approximately 5,000 samples in the training set and 23,000 samples in the test set across 19 identity classes.

- For the semi-supervised Label Spreading method, 10% of the samples were used to build a training set, and every 50th sample was labelled while all others were marked as unlabelled. The other 90% of the samples were then used as the test set. Accuracy and computation times were derived as an average across five trials. In each trial, there were approximately 2,800 samples in the training set with only 56 labelled samples, and approximately 25,000 samples in the test set across 19 identity classes. This is a reasonably challenging setup for label spreading.

- For the one-shot methods, feature vectors are streamed in order of appearance in the recorded footage, mixed between the four camera views. The first sample of each class is labelled and given to the algorithm as an anchor, so that a new class is created for the new identity, and then all subsequent vectors were labelled by the algorithm itself and used for further training (i.e. model update). The online SVM and online perceptron have the same underlying implementation as the offline versions, except that they are only partially fit with each sample, using early stopping to avoid overfitting. Accuracy and computation times were again averaged across five trials, with a single sample per class across 19 identity classes as the "training set" and over 28,000 unlabelled samples as the test set. Importantly, for the online SVM and online perceptron methods, the number of output classes need to be defined at initialisation as part of model creation, limiting the flexibility and scalability of the classification approach. In the case of Gallery and Sequential k-Means, the number of classes is only defined for the

Table 4.6: Classification Accuracy and Computation Time on UoA-Indoor

Method	Accuracy (%)	Classification Time (s)
Supervised Learning Methods		
Offline SVM	98.82	8.377
Offline Perceptron	91.35	0.391
Semi-Supervised Learning Methods		
Label Spreading	72.92	1.772
One-shot Learning Methods		
Sequential k-Means	67.22	0.282
Gallery	48.41	4.720
Online Perceptron	34.44	29.797
Online SVM	29.61	28.891

purposes of memory allocation, which could be defined as an upper bound limit on the number of simultaneously detectable classes.

4.4.2.2 Results

An ideal classification algorithm is one that has high classification accuracy and low computation time. All accuracies and computation times are derived from a desktop computer running Ubuntu 16.04 with an i7-6700 CPU with four 3.40GHz cores and 64GB of RAM, with the algorithms written in Python augmented with NumPy and Scikit-learn. The computation times should be compared to each other based on this machine, rather than taken as evidence for real-time execution generally. They are also times taken to classify the entire dataset, not per image.

As shown in the results in Table 4.6, the supervised learning results show that the dataset is largely classifiable in an offline training context, so any further decreases in accuracy are likely due to the method rather than inherent characteristics of the dataset. As the number of training samples is reduced, the accuracy rate falls, as shown in the semi-supervised and one-shot methods. Including both the supervised and one-shot variations of the perceptron and SVM show that without many training samples, it becomes very difficult to create accurate models and many more training epochs will be attempted with large errors, leading to much higher classification times and lower accuracy rates. The Gallery method still has a relatively high classification time due to the need to iterate through many target samples in the gallery for each possible class, but potentially does not replace samples in the gallery fast enough to keep up to date with recent samples, leading to erroneous classifications. It is clear that out of the one-shot methods, the Sequential k-Means is the best method, with a substantially higher accuracy rate and a substantially lower classification time. However, losing 30% accuracy between the supervised and one-shot learning methods needs to be further mitigated in order

for these algorithms to be viable for real-world applications. These one-shot learning methods may not yet be able to sufficiently overcome inter-camera variation when mixing the camera views together.

4.4.2.3 Comparative Analysis

When other person tracking and re-identification methods are published, the results can be reported in many different ways. A CMC/ROC curve is very common, but in a real system the method ultimately needs to make a decision about which identity to assign to a person, so only the r=1 (top rank) accuracy should be used for comparison against these methods. State-of-the-art re-identification approaches achieve around 50-80% accuracy for rank one[18], although this appears to be strongly dependent on the dataset and therefore must be interpreted cautiously. The CUHK01 dataset is perhaps the most similar to UoA-Indoor and the test scheme described here, using a single-shot evaluation protocol similar to the experiments presented here, but with an outdoor context, a much larger number of identities, and only two camera views. The top performing method on this dataset [186] achieves 66.6% accuracy at rank 1 using a CNN (no computation time is reported). In [187], a deep colour and texture-based one-shot learning approach, learning local colour metrics for transfer functions between camera views achieves 51.2% on iLIDS and 45.6% on CUHK01, but requires a powerful GPU to achieve reasonable computation times. A comparative one-shot learning algorithm achieves 43.9% accuracy at rank 1 for walking people in the BIWI RGBD-ID dataset, although this result has the advantage of using depth information as well [188].

More recently, a series of identity-based measures have been proposed [39] in a person tracking context, utilisting the more traditional measures of Precision, Recall, and F-Score. The accuracy results from the previous experiments can be interpreted as being largely analogous to IDP (ID-Precision), as they both measure the number of correctly classified identities out of the total number of detected individuals. In a multi-camera context, the state-of-the-art achieves approximately 58% IDP [189] on the DukeMTMC dataset [39], which also has an outdoor context but with eight non-overlapping cameras. Most of these datasets have a much larger number of identity classes than UoA-Indoor, so the scalability of Sequential k-Means needs to be further validated for fair comparison to other methods. The challenge for fair comparison with other works in the published literature is that methods like Sequential k-Means that iteratively improve their models, taking into account small and gradual changes over time, are inherently disadvantaged by most re-identification datasets, which only have a small number of samples for each identity class. These samples in many datasets are also generally discontinuous, meaning that they are not from sequential frames and there are often large time periods between the captured samples. Sequential k-Means achieves relatively high accuracy on the UoA-Indoor dataset in part because there are thousands of samples available for each class, and the samples can be fed into the algorithm sequentially in time order of capture.

[18] William Robson Schwartz, "Person Re-identification Results", http://www.ssig.dcc.ufmg.br/reid-results/.

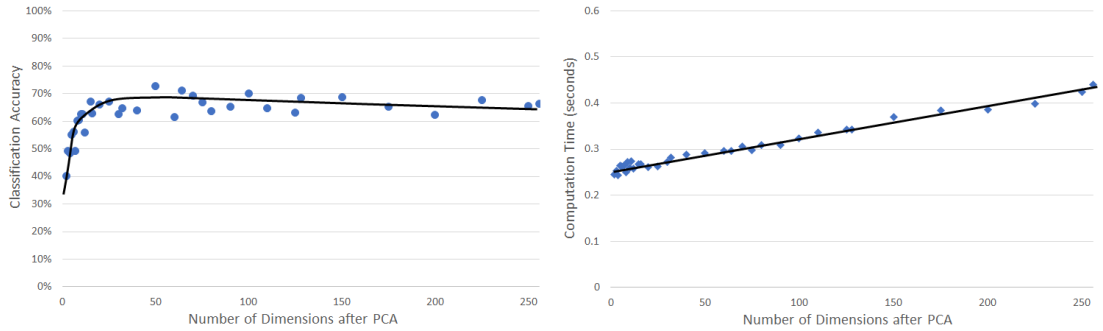

Figure 4.8: Classification accuracy (averaged over five tests) and computation time (per test) from a Sequential k-Means method classifying approx. 28,000 feature vectors across 19 classes with between 2 and 256 dimensions after PCA and a covariance metric transformation.

4.4.3 Tuning Sequential k-Means

For the purposes of fair comparison, the same data was used for all four one-shot learning algorithms, but the parameters used in previous steps can be tuned to improve the overall accuracy of the system. Firstly, the number of dimensions to include in the feature vectors after PCA was experimentally validated again, since the original number of 32-64 dimensions was determined based on an offline SVM. Figure 4.8 shows a similar curve to Figure 4.6 from the offline SVM experiment, where the accuracy initially increases and then drops off, slowly decreasing as an increasing number of less discriminative dimensions are kept. However, the computation time for Sequential K-means increases linearly (rather than exponentially for the SVM), since the model size linearly increases as the feature vectors become longer. Selecting 50 dimensions for the PCA reduction improved the accuracy of Sequential K-means from 67.22% to 72.65%, a substantial improvement of over 5% just from increasing the number of dimensions from 32 to 50.

Secondly, the update rate β was retuned with different lengths of feature vectors after PCA. It appears that these two parameters are largely independent, i.e. changing the number of dimensions has negligible effect on the impact of changing β on accuracy. This can be seen in Figure 4.9, where the main shape of the curve is roughly the same regardless of the model update rate, and as the model update rate increases the accuracy decreases. This implies that the two parameters likely have independently additive effects. The highest accuracy is achieved when β is between 0.05 and 0.1, around 50-dimensions post-PCA. Variations in β have no impact on computation time since it is just a constant used in a multiplication operation.

4.5 Conclusions

In this Chapter, fast person re-identification is presented using an approach with multiple parts: dimensionality reduction, metric learning, and classification. Firstly, it is shown that dimensionality reduction can have positive effects for both accuracy and computation time when reducing the feature vectors to the most easily separable dimensions. Secondly, a number

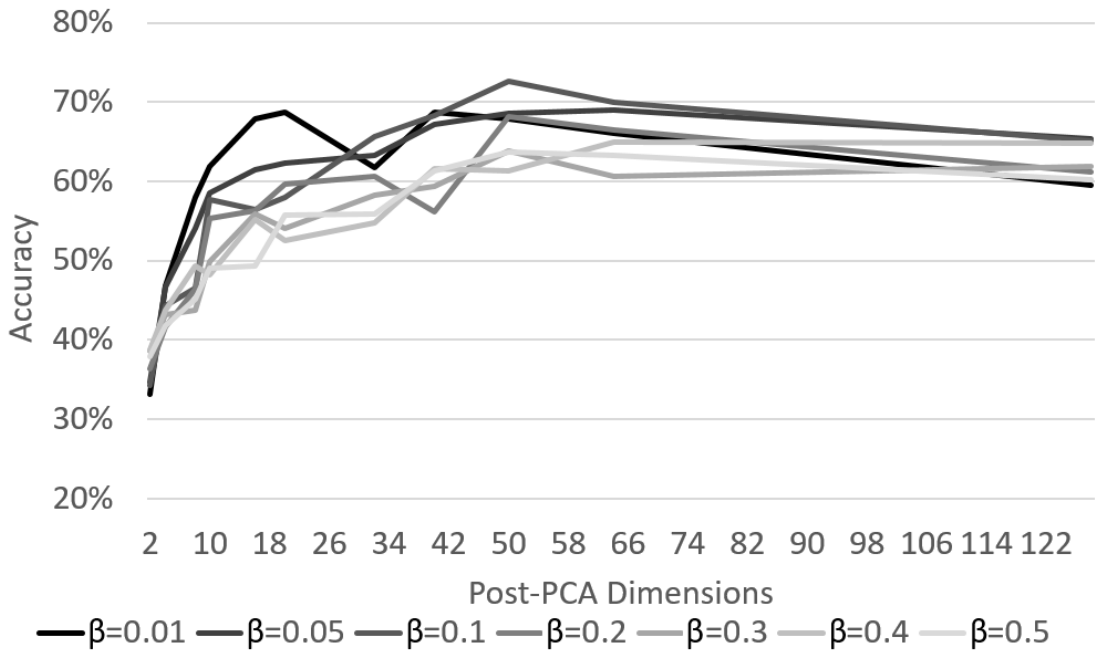

Figure 4.9: Classification accuracy (vertical axis) from Sequential k-Means depending on the number of dimensions after PCA (x-axis) for different model update rates β. Lighter colours represent higher model update rates.

of metric learning algorithms are tested for improving the separability of reduced feature vectors, and it is discovered that the covariance metric transformation has the best accuracy-speed characteristics, which has superior generalisibility for previously unseen classes at run-time. Thirdly, a number of one-shot learning classification algorithms are compared, both against offline supervised methods and against each other. The Sequential k-Means method is shown to have the best accuracy amongst one-shot learning methods. Importantly, this relies on the assumption that the model can continue to update and develop over time, in contrast to supervised learning approaches developed on most existing re-identification datasets. Finally, parameter selection for the Sequential k-Means method is discussed. Future work includes establishing longer-term understandings of appearance, such as in [190], which establishes a wardrobe model for each identity to help with long-term re-identification over many days. In the next Chapter, the spatio-temporal tracking model is presented, along with the model fusion process of combining both the appearance and spatio-temporal models together to produce reliable identity classification.

Chapter 5

Person Tracking

While person re-identification using Sequential k-Means offers some promise for classifying identities across multiple cameras in an unsupervised manner, the results in the previous Chapter show that it does not classify with perfect accuracy. Appearance only uses a subset of the information that is available, and by combining a different type or source of information, the accuracy could be improved. This is particularly relevant if other types of information are strong or reliable where appearance is weak or prone to error. Examples include combining camera information with infrared sensors [191], badge sensors or swipe cards for physical localisation [31], and triangulated wireless Bluetooth signals [192], but these all require additional infrastructure to be installed. A spatio-temporal based tracking model presents an alternative type of information based on the positions of detected people and how they move over time and can be extracted from the camera footage. These two types of information can be used independently to achieve person tracking, but could also be used in a complementary way to improve the overall accuracy of the system. This Chapter works towards this goal, using a Kalman Filter-based spatio-temporal model, and combining it with appearance-based person re-identification to jointly perform classification and tracking to achieve higher levels of accuracy while still maintaining low computation times.

5.1 Spatio-temporal Based Tracking

Within a single-camera view a person's movements throughout a space can be tracked by estimating the positions of individuals in successive frames, and then connecting points together based on proximity and time to form tracks or "tracklets". By tracking the person in space and time, a spatio-temporal model of how the person moves is formed. This can be as simple as taking the points in one frame, and connecting each one to the nearest point in the previous frame based on the Euclidean distance. However, the simplicity of this scheme also has many weaknesses; errors can occur if people are walking closely together, or if they walk past each other, or if a person walks behind an object that obstructs that camera's view of the person momentarily (creating a discontinuity). The most common solution is to add some kinematics physics modelling, essentially estimating the velocity of the person in order to constrain

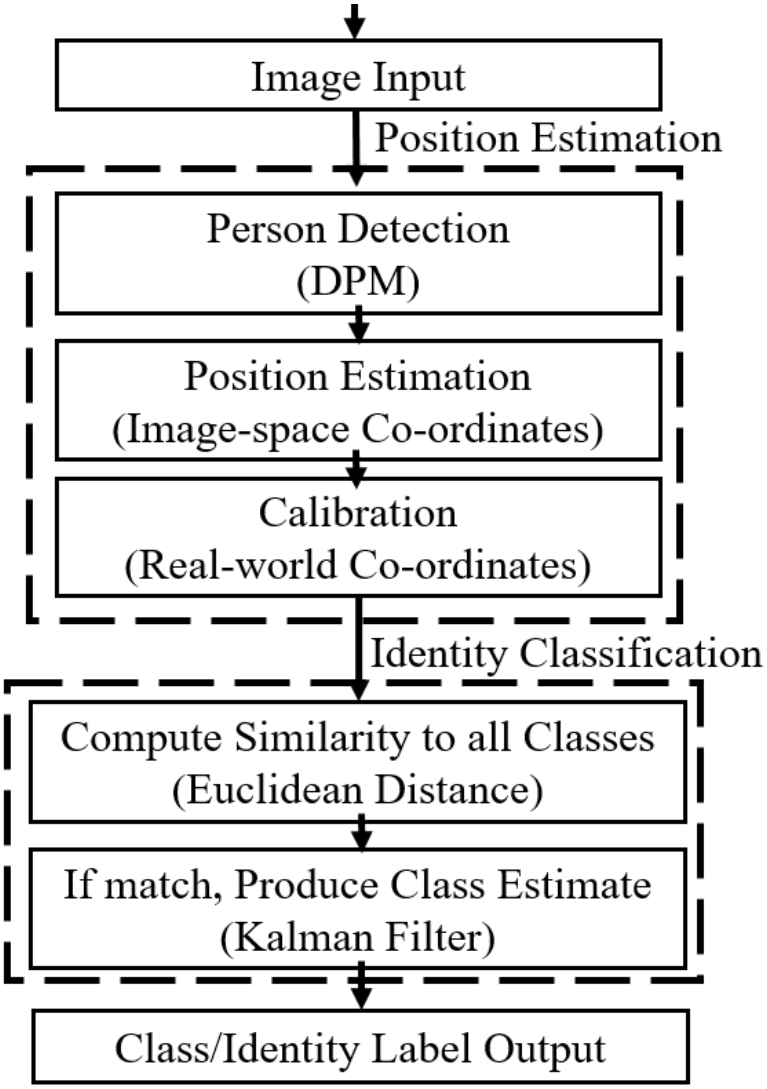

Figure 5.1: The position estimation and identity classification process in the spatio-temporal model

the directionality of the track and bridge discontinuities. Using kinematics is common in tracking moving objects generally, but there are still challenges, such as second-order effects (i.e. acceleration), sudden or abrupt changes in direction, the impacts of noise, and real-time processing requirements [193, 194].

The desired output of a tracking system is usually a set of lines or points that represent the path of the object. In the video analytics system proposed in this thesis, the tracking problem is formulated slightly differently. Instead of constructing identity-agnostic tracks that just show how people move, the intention is to use person tracking techniques for classification, i.e. giving each track a label corresponding to an identity. This is achieved by using tracking techniques to produce position predictions for each class, and then classifying the input position by finding the closest predicted postion. This is so that the output of the tracking system can be combined with the output of the re-identification system to help determine the identity

of the person. Once the position of the person has been estimated in the image/camera view, the spatio-temporal classification is achieved through three main steps shown in Figure 5.1. These use off-the-shelf algorithms, so they will only be described briefly in this thesis; readers interested in further details may investigate the references given. Firstly, camera calibration matrices are used to convert the image-space pixel co-ordinate for the person to a real-world co-ordinate on a map [195, 196]. Secondly, the position of each person detected in frame N (the current frame) is classified based on their proximity to each of the predictions in frame N-1 (the previous frame). Lastly, Kalman Filters are used to predict the next position of each track in a way that takes the kinematics of the person into account, with robustness against noise [193, 197, 198].

5.1.1 Calibration and Mapping

Humans are used to seeing maps of physical spaces from a bird's eye or overhead view (i.e. above the scene, with the field of view towards the floor). However, mounted cameras in ceilings are rarely at that angle - in fact, this would be undesirable in many cases, as the camera would not be able to see the body of the person and would likely only see the top of their head [199, 200]. Instead, cameras are often mounted at approximately 45 degrees, halfway between the wall and the ceiling. Calibration is the process of finding a transformation matrix (also known as a projection matrix or homography) that converts the image-world space pixel co-ordinates to real-world space position co-ordinates by using a number of fixed known points. This can also be described as a process of reconstructing 3D points from the 2D image by determining the camera parameters that transform the real-world space into the image-world space, and then finding the inverse of that transformation. With multiple cameras, this has the added dimension of positioning the camera views relative to each other, usually based on overlaps that allow for topologies to be determined (i.e. where the cameras are, relative to each other). The key to this is assuming a common ground plane (sometimes also called a homography) that is shared between the different camera views. Only the position of the person is needed in this system, so while a 3D-point can be constructed, generally the height is not relevant and an assumption is made that all movement is on a flat ground plane, such that the height co-ordinate can be held constant. This allows for a small number of points with known physical co-ordinates to be used to find a homography between the camera and the ground.

Calibration is a relatively well understood area of computer vision, as it is largely based on single-view geometry and matrix operations, allowing for deterministic operations that are simple enough to be solved by hand. Essentially, the desired output of the calibration operation is a 3x3 matrix, which represents the translation and rotation operations that should be applied to each point of the image in order to convert it to the real-world co-ordinate. There are actually two parts to the derivation of this matrix - the camera has both intrinsic and extrinsic parameters; intrinsic parameters are those that are dependent on the design of the camera, which affects the focal length, the image sensor itself, and other scaling factors, while the extrinsic parameters are those that are based on the placement of the camera relative

Figure 5.2: An example of the effects of camera calibration on camera view 1 from UoA-Indoor, with the transformation matrices from intrinsic (left) and extrinsic (right) calibration applied.

to the environment it is measuring/observing. Intrinsic calibration is often performed using a checkerboard, while extrinsic calibration requires some knowledge of the real-world co-ordinates. The idea behind extrinsic calibration is to determine some mapping operation that allows image space or "virtual" co-ordinates to be converted into positions in another set of axes, often from a bird's eye view. As is shown in Figure 5.2, the effect of intrinsic calibration on the image co-ordinates is relatively limited, while the effect of extrinsic calibration is much more significant. Hence, the focus is largely on getting good extrinsic calibration.

In the UoA-Indoor dataset, this is achieved by marking four points on the floor that are visible to all of the cameras, taking a frame from each of the camera views, marking the points in the image space, and then combining that with the real-world co-ordinates to find the transformation matrix. A small application was written to allow a frame to be shown to the user, with the user clicking on each of the marked points, and then storing the pixel co-ordinates. The `findHomography()` function in OpenCV [75] then optimises towards a homography with minimal reprojection error (i.e. applying the derived matrix in the reverse direction to check the accuracy), using the hard-coded real-world co-ordinate points. This is performed for each camera view, producing four transformation matrices. The result of this is shown in Figure 5.3, where the four camera views have been superimposed upon each other to show the overlapping and non-overlapping parts of the camera views. The corners of the yellow rectangle in the middle of the Figure are the four points used for extrinsic calibration; they were positioned so that all four cameras could see the same points. The main weakness of this approach is evident in the Figure - while the calibration is quite good for the center of the field of view, as the pixels get further from the center the image gets increasingly stretched and distorted. However, for the purposes of measuring the position of a detected human, this may be acceptable; the stretching largely occurs on the walls, which will be ignored when calculating the co-ordinate on the ground plane. There are still some issues on the edges of the camera view (as evidenced by the small circular table on the left side of the image appearing multiple times); potentially these could be resolved by increasing the number of points used for calibration, and relaxing the requirement that those points be visible to all four cameras. In this scheme, each camera is actually independently calibrated to the ground plane, so calibrating

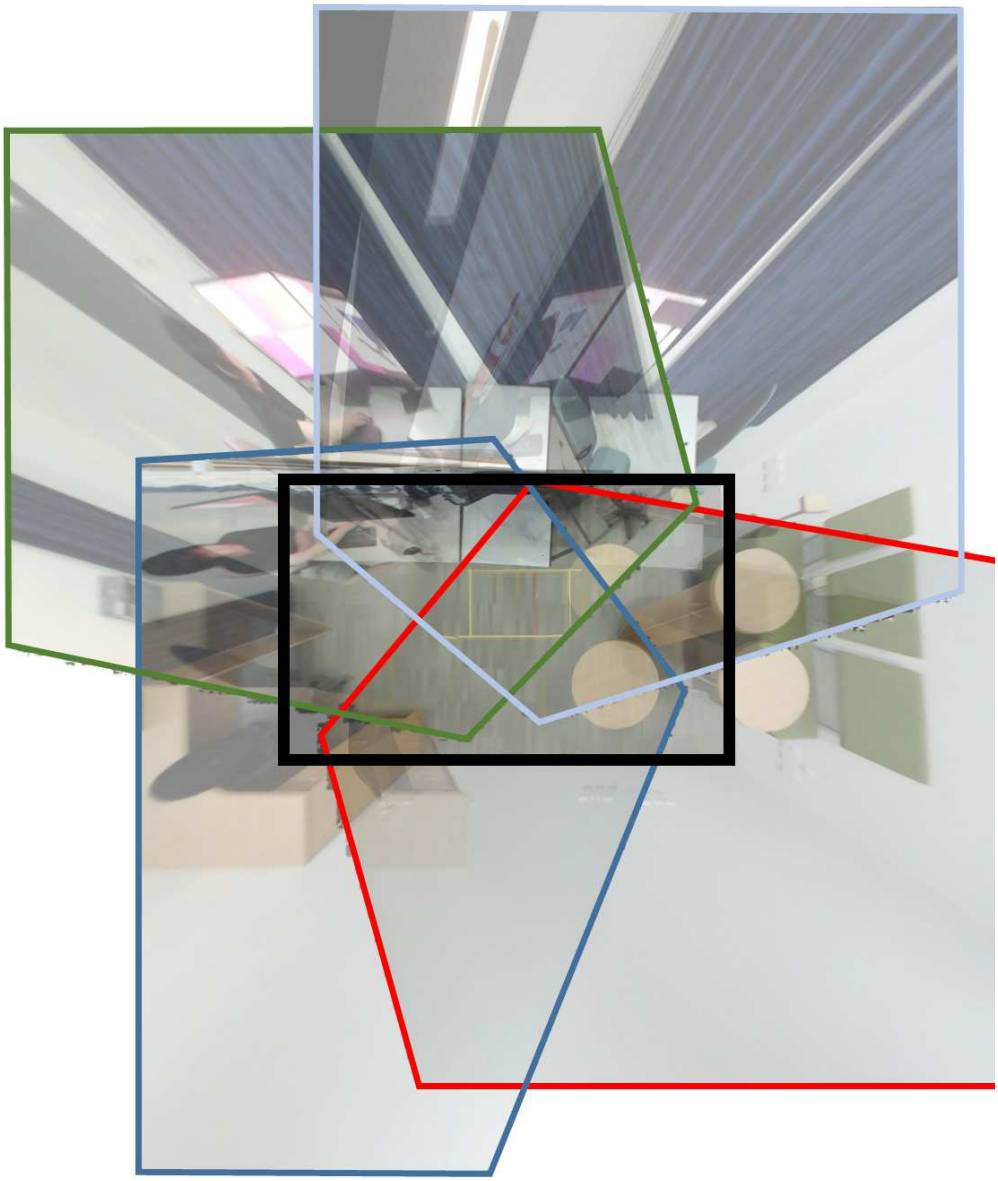

Figure 5.3: The four camera views, transformed and superimposed upon each other with 50% transparency, with boundaries of each camera view marked with red (view 1), light blue (view 2), green (view 3), and dark blue (view 4) lines, and the floor space marked with a thick black rectangle.

points to the edges of the floor in each camera view would likely extend the central area that has been well calibrated. Another idea is to automatically extract common points between the camera views using the SURF or RANSAC algorithms, calibrating one camera with the ground plane, and then calibrating the other cameras relative to this first one. The advantage of this approach is that it would significantly reduce the human errors associated with placing marks in the physical space and measuring the physical co-ordinates, and there should be many more points available to help improve the calibration. An idea related to this has been explored in [201], and could be further investigated as part of future work.

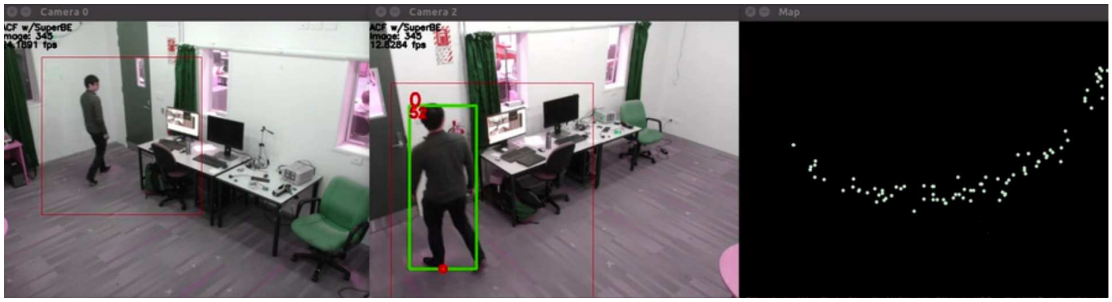

Figure 5.4: An example of the positions of a detected person being mapped over time. The top edge of the map corresponds to the wall with the door shown in the two camera views.

As people are detected by the cameras, their positions can now be measured using the homographies. First, the position in image-space is measured by taking either the midpoint between the two foot parts in DPM, or the middle of the bottom edge of the bounding box in other person detection algorithms such as ACF, as a reasonable representation of where on the floor the person is currently standing. Then, that pixel point is multiplied by the transformation matrix, which provides a real-world co-ordinate, relative to Figure 4.4. This co-ordinate can also then be plotted on a map, showing the position of the person as they move throughout the room. An example is shown in Figure 5.4, where the points from all four camera views are used to show the path, which enables full coverage since not all cameras will detect the person or have visibility of the person in every frame. As the map shows, there is significant variation/error/noise - even though a human can easily discern a path that goes through these plotted points, this can present challenges once there are multiple people who might be close to each other, and the wide variation can cause their paths to mix and intersect. This is largely due to the instability of the person detection algorithms; even if a person is standing still, the bounding box is not usually very tight or exact, and has some freedom to move around the person. Since the position is measured based on the bounding boxes, this causes variation in the position measurement, and even a one pixel shift can translate to a shift in the real-world co-ordinate of several centimetres. So while the calibration algorithm itself is relatively consistent and accurate, and produces reasonably good homographies for the shared ground plane between the cameras, the real-world co-ordinates should be considered a noisy and imperfect data source. This can be addressed using a filtering technique that deals with noisy measurements.

5.1.2 Kalman Filters

The use of an algorithm to process the position data in some way has two main purposes: to reduce the level of noise in the position measurements, and to incorporate some kinematics modelling to improve future position estimates. It is important to make a syntactic note here: a measurement refers to the calculated point derived from the sensor (in this case a camera), whereas an estimate refers to the output of further processing that produces a synthetic point

that does not exist in the raw data, either from multiple sensor sources or from one sensor source over time. There are a number of algorithm choices available, but a balance needs to be struck between accuracy and speed. The simplest method is averaging of the measured points from different camera views, but this requires synchronisation of the cameras, and the person is not necessarily detected in all four camera views at any given point in time, introducing the need for more complex co-ordination logic between the cameras. It would also likely fail to model the kinematics of the moving person when the velocity is not exactly constant. Particle filters [202] are another common option, which simulate many possible estimates in the environment (modelled as particles), and iteratively update the probabilities or likelihood of each particle being the true measurement until convergence. However, this is a very computationally expensive method because of its iterative nature depending on the number of particles used, although [203] applied particle filters in a person tracking context and claimed real-time computation of video feeds on a standard desktop PC without parallel processing. Optical flow methods are also common for tracking moving objects, such as the Lucas-Kanade method [204], although this class of algorithm can also be relatively computationally expensive [38]. Perhaps the most popular choice for tracking applications today is the Kalman Filter; it is well understood and used in a wide variety of contexts, from analysing time series signals to navigation and control of robots, vehicles, and aircraft [205, 206]. It is a statistical approach using joint probability distributions to combine multiple measurements over time, producing estimates that tend to be more accurate than the original unprocessed measurements. The Kalman Filter is also popular because of its computational efficiency, producing reasonably good results quickly. Kalman Filters have been used in person tracking applications in the past [51, 200], and seem to be a good candidate for overlapping multi-camera environments where the position measurements from each camera view may disagree and include some error.

The Kalman Filter maintains a model of past information (or state), and combines that with recent measurements to produce output estimates. This is achieved iteratively as new measurements arrive, giving the model more confidence about the true state of the system as more information is provided. The Kalman Filter begins with Gaussian distributions, which model the potential states centered around the first few measurements. As more measurements are introduced, the shape of that distribution is transformed based on the mean and uncertainty of the combined measurements. This is achieved by forming a prediction matrix, and then correcting that prediction with the new measurements to minimise the error between the prediction and the actual measurement(s) using the joint probability distribution, which can be visualised as overlapping Gaussians. OpenCV [75] provides a built-in Kalman Filter implementation, based on the standard formulation given in [197]. In this prototype system, a Kalman Filter is instantiated for each potential identity class, along with an array to store the previously measured points for the purposes of creating tracks. The Kalman Filter is initialised to keep track of four variables: the x position, the y position, the x velocity, and the y velocity (or in other words, two dimensions for position and their first derivatives with respect to time). This can be represented in vector form as [x, y, v_x, v_y]. Since the Kalman Filter used in this spatio-temporal model keeps track of both position and velocity, the probability distribution

for the position should be influenced by the velocity estimates as well, although the standard Kalman Filter enforces linear assumptions on these variables.

In this formulation of the Kalman Filter, there are four matrices to be initialised: the measurement matrix H (sometimes called the observation model), the transition matrix A (sometimes called the state-transition model), the measurement noise covariance matrix R, and the process noise covariance matrix Q. It should be noted that a wide variety of symbols/variables are used throughout the literature, so the symbols used here may be different to those found in other reference material. The measurement matrix indicates which variables will be provided in any given measurement; since only the x and y positions can be inferred from the cameras, the x and y velocities have to be modelled internally within the filter. The transition matrix represents how the variables influence each other; in this case, the x position is related to the x velocity, and the y position is related to the y velocity. These first two matrices govern how the filter should process measurements to produce estimates, but a filter with only these two matrices set would assume a perfect world with zero noise. Providing some assumptions around the noise levels can help the Kalman Filter better understand the level of uncertainty associated with the measurements, and adjust the predictions accordingly. The measurement noise represents some guess at the level of uncertainty in the measurements, in this case the uncertainty that a real-world position calculated from the image-world position of a detected person is accurate. The process noise represents the level of uncertainty on the consistency of the physical process, in this case the movement of the person. Where the motion is relatively steady (i.e. a close-to-constant velocity), then the values in this matrix can be set low, while if the motion is sporadic or changes in magnitude and direction significantly, then the values should be set high.

The values for H and A were set by design, while R and Q were selected through empirical experimentation, since they are largely dependent on the scale and range of the measurements. The magnitude and type of noise encountered are not well understood at design time, so small values are used to minimise overcorrection and avoid excessive uncertainty. For the purposes of replication, the four control matrices were initialised to:

$$
\begin{aligned}
H &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} & R &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} * 0.1 \\
A &= \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & Q &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * 0.005
\end{aligned}
\tag{5.1}
$$

In this formulation, the Kalman Filter is governed by two sets of equations: prediction/estimation and update. As more measurements are provided, the Filter updates itself to better model the motion of the person and reduce the level of uncertainty on the predictions.

Keeping the mathematical equations simple, the predictions for the state x and system noise P are formed using:

$$x = A\hat{x} \tag{5.2}$$

$$P = A\hat{P}A^T + Q \tag{5.3}$$

which can be described as using the internal representations of state $\hat{x}$ and system noise $\hat{P}$ and progressing them one step forwards based on the transition (A) and process noise (Q) matrices. When a new measurement is provided to the Filter, these internal representations are then updated to allow the next prediction to be more accurate, using the equations:

$$\hat{x} = x + Ky \tag{5.4}$$

$$\hat{P} = (I - KH)P \tag{5.5}$$

where y represents the error between the new measurement and the current prediction (also called the "innovation"), K represents the Kalman gain (which is calculated from a ratio between the process noise R and measurement noise Q), and I is the identity matrix. Essentially, these two equations modify or "correct" the internal representations of state and noise based on the new measurement (through y) and both sources of noise (through K). For more detail about the standard Kalman Filter formulation and how y and K are determined, please refer to the OpenCV documentation and [207].

One issue with this formulation is that the Kalman Filter strictly speaking does not have a notion of time, since it assumes that the measurements are uniformly sampled in time by progressing in logical steps, which may not always be the case in this system. This makes modelling the velocity somewhat challenging, especially since at any given point in time the number of cameras contributing measurements to a particular Kalman Filter is not known *a priori*. While the effect of this is non-zero, since there are only four cameras in the system, and most of the time only one or two cameras can see a person, the likelihood of multiple cameras contributing points at the same physical time that are being interpreted as being input at successive logical times is relatively low. However, the opposite issue is also challenging to deal with, where there may be indeterminate gaps in time between detections, which can be hard to model using the Kalman Filter because it is difficult to know how many logical steps the Filter should be progressed to compensate for a loss of measurements.

The spatio-temporal model for classifying person identities is used in a two-step process: classification and updating. At frame N, when a person is detected, the measurement of the position is compared to all of the predictions from frame N-1, and the person is classified as being part of the class with the most physically proximate prediction based on the Euclidean distance. Then, the measurement at frame N is passed to the relevant Kalman Filter, which produces a new estimate that serves as a prediction to be used in frame N+1. There is a slight caveat in that the Kalman Filter normally needs to "warm up", because when there are not very many measurements the initial probability distributions can be very large since there has not

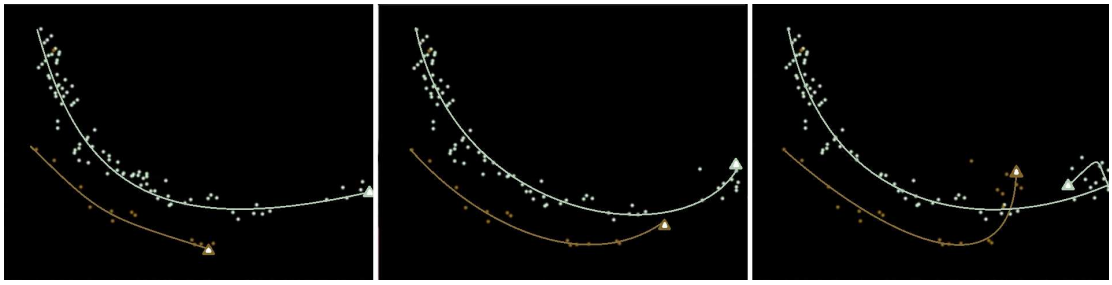

Figure 5.5: Tracks (drawn by a human) for two people being estimated over time. Full circles indicate the measured points from the images, and the triangles at the end of each track indicate the next Kalman Filter estimated position, which is treated as the prediction for that identity class.

been enough input data to produce good estimates. This means that for the first 10 detections in each class, the measurements are input to the Kalman Filter, but that measurement is also used as the prediction instead of the estimate from the Kalman Filter. Using the Kalman Filter approach described here, reasonably good predictions can be obtained, as shown in Figure 5.5, where the tracks and predictions for two people are shown. Person 0 (white) shows a good example of how the Kalman Filter can deal with noisy readings from multiple cameras - after each frame, up to four points could be added (one for each camera view), and the Kalman Filter combines them all (with the past points) to produce a reasonable estimate of the actual position of the person. Person 1 (brown) shows an example of the prediction being very responsive as newer measurements are added, and while the previous measurements have some impact, the more recent points have the largest influence. This spatio-temporal Kalman Filter-based model appears to be reasonably effective for tracking the position of people and helping to classify detected people into their identity classes; evaluation of the accuracy and computation time is presented in Section 5.3.

There are a few weaknesses with this approach to highlight here. The most significant is that the model is dependent on knowing how many people there are in the room, i.e. how many people it should be tracking at any point in time, so that the right number of Kalman Filters can be instantiated. This is achievable in the one-shot learning case where an auxiliary sensor is able to determine when a person has entered and exited the region of interest, but in a purely unsupervised case it becomes very difficult for this model to determine when a new identity class should be created (i.e. a new person has been seen). This can be somewhat alleviated by setting an upper bound during the classification step, where if the closest prediction is still further away than this upper bound, then a new identity class should be created. However, this then exposes a weakness in the entrance/exit areas, where the beginnings and ends of tracks tend to accumulate. Without a way to know how many people have entered or exited the area, it can become easy for a new person entering the space to be assigned a track that belongs to a person who had previously exited the space through the same area, as the prediction from the previous person will be the closest to the current position of the new person. Furthermore, the model faces significant difficulties differentiating between people who are walking closely

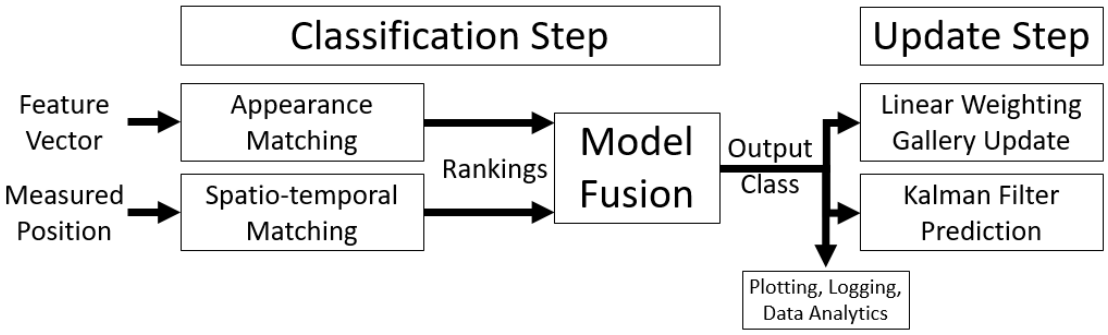

Figure 5.6: A diagram showing the dataflow architecture for the classification (including model fusion) and model update steps of the person tracking pipeline.

together - especially because the measurements from the camera images contain significant noise and error. ID switches can become very common, which may not matter while the people are still walking close together, but if their paths diverge then it may be important to ensure that those tracks are assigned to the correct identities from earlier in time. These weaknesses are different to those experienced by the appearance-based re-identification model, so combining these two models together in a novel and complementary way may reduce the negative impacts of these issues.

5.2 Model Fusion

Now that there are two sources of information, they can be combined or "fused" together. Figures 5.6 and 5.7 show the architecture that is described in this Section, split into the classification and model update steps. Both the appearance and spatio-temporal models are designed to return a ranked list of identity classes, based on the similarity between the probe and the samples stored in the gallery. The aim of the model fusion module is to combine these two lists, taking into account information such as the relative confidence or likelihood that each class in the list is the correct one, to produce a single output class that is declared the identity number for the detected person. This output is primarily used for subsequent processing such as plotting on a map or keeping records in a log, which can be used for further data analytics to answer some of the higher level questions posed in the motivating scenario in Section 1.2. But importantly, that output class is then also used to update the relevant gallery samples, with separate processes for the appearance and spatio-temporal models, before the next person is detected and classification has to be performed again. It is therefore important that the output of the model fusion step is of high quality in terms of accuracy, as erroneous fusion can lead to the models learning incorrectly and increase the likelihood of future misclassification.

Combining person tracking and person re-identification is an idea that holds a lot of promise, yet the literature exploring it is limited. There is some interest, as shown by the 1st Workshop on Target Re-identification and Multi-Target Multi-Camera Tracking[19], organised

[19]https://reid-mct.github.io/

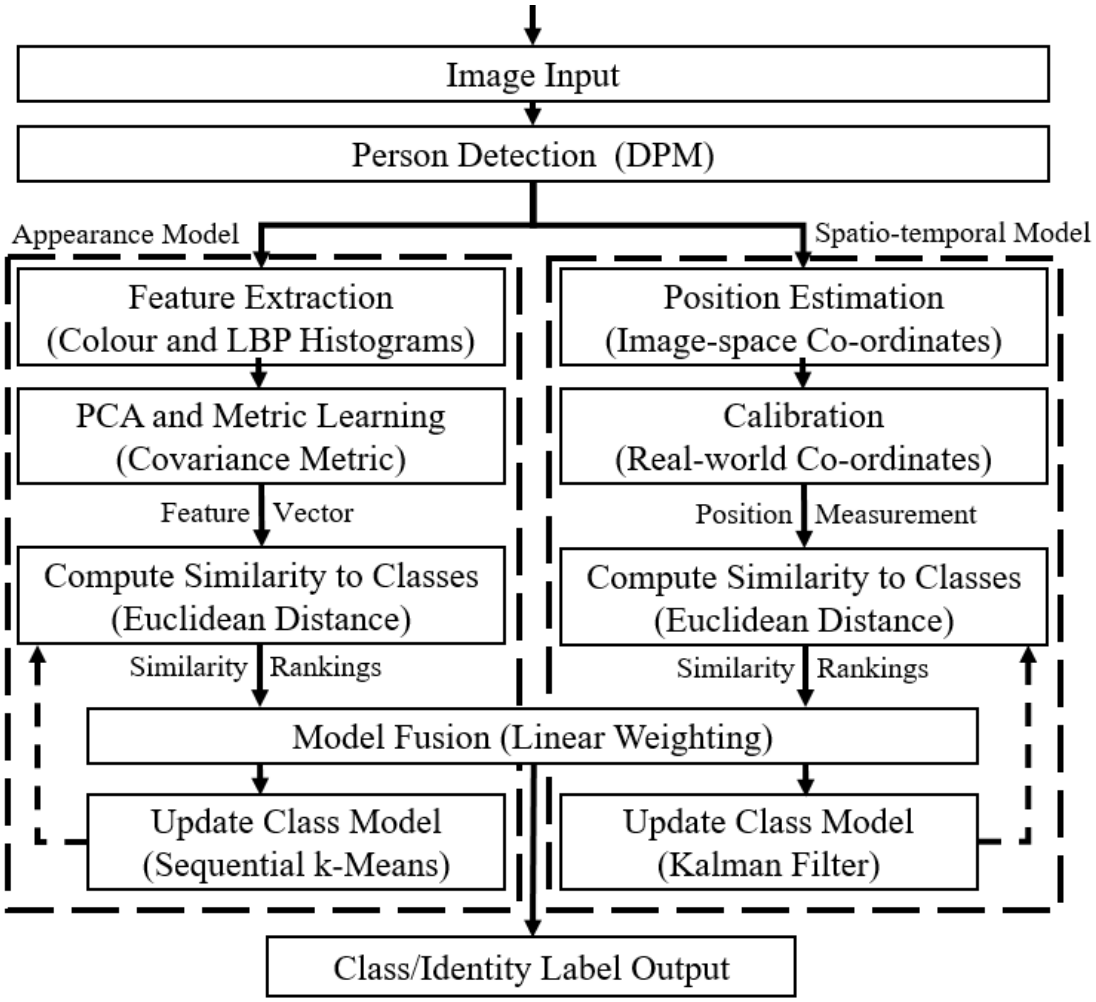

Figure 5.7: A flowchart of the combined appearance and spatio-temporal models in parallel, leading to the model fusion step, with the class updates feeding back into the classification step for the next detection.

at the Conference of Computer Vision and Pattern Recognition (CVPR) in 2017, but most of the papers at this Workshop still focused on either tracking or re-identification but not both. Commonly, spatio-temporal based tracking is used within single camera views, and person re-identification is used to connect tracklets between camera views together. In contrast, the method presented in this Chapter combines both spatio-temporal tracking and appearance-based re-identification to jointly identify and track individuals as they move throughout a space, making it applicable to both single-view and multi-camera contexts. Previous work has focused on using person re-identification in crowded scenes to help differentiate people within a single view and therefore allow for separate tracks to be assigned to each individual, avoiding ID switch and fragmentation errors. In [208], the PIRMPT system uses colour (RGB), texture (HOG), and shape as appearance features to train a boosted affinity model based on offline training data, and combines this with tracklets to connect them together and produce longer, more stable tracks. A similar idea is used in [209], where they form the task as a

Lifted Multicut Problem that combines a probabilistic multicut method to select and connect acceptable tracks from the recorded positions with a supervised CNN-based re-identification model. In these systems, the spatio-temporal tracking usually still takes primacy, with the appearance/re-identification side usually just helping to refine the concatenation of tracklets. The work in [210] presents one of the first systems that treats spatio-temporal tracking and re-identification as two valid sources of information that can be fused together probabilistically, using the Optimal Bayes Filter (based on Bayes Theorem) to combine measures of confidence and dynamically change the level of contribution from each model. In the system described in this thesis, the weaknesses of both the spatio-temporal and appearance models are explicitly taken into account using a re-ranking process and a rule-based system to determine weights that control the contribution of each model to the final output.

5.2.1 Re-ranking with Linear Weighting

In the context of person re-identification, the process of re-ranking refers to taking an ordered list of potentially matching identities and then refining that list to increase the likelihood that the correct identity is near the top of the list. For example, POP [211] takes the ordered list of candidates, asks a human to select some definite negative results, and then re-ranks the list by deprioritising similar choices. More recent works have focused on identifying the group of neighbouring/similar classes that should be close together (contextual similarity), which can be used to retrieve classes towards the bottom of the list that should be closer to the top [212, 213]. For most person re-identification works, the aim is to produce a good Cumulative Match Curve (CMC), which is based on the correct identity being in the top r entries of that list [41]. However, in a comprehensive system like the one described in this thesis, having the correct identity in the top r entries is not particularly helpful unless $r = 1$; ultimately, the system needs to select one class or identity for labelling so that the appropriate models can be updated. Furthermore, many re-ranking algorithms operate in a supervised manner, using large amounts of data available within a dataset to improve the rankings for that particular dataset; the general applicability of these approaches is questionable, and cannot function in an unsupervised way where training data is not available [213].

The main model fusion task in this person tracking pipeline can be framed as a re-ranking problem where two ordered lists, one from the spatio-temporal model and one from the appearance model, are combined to produce one refined list that incorporates the results of both models. There are a family of Ranking Aggregation methods [214], particularly those that relate to optimisation and voting [215], that could be used to achieve this; for example, a Borda Count would allocate points to each position in the list, with the most points allocated to the top (or most likely) classes in each list, and then simply sum the points for each class and re-sort to produce a new ordered list. However, approaches that solely rely on the ordering of the lists suffer from two major problems. Firstly, there is an inherent assumption made that the choices are an equal distance apart from each other (or in other words, the ranking is linear); the third rank is as far away from the second rank as the second rank is away from the first rank. This

misrepresents how the list was derived - the classes are ordered based on their distances to the probe (i.e. the person detected in the current frame), and those distances are almost never uniformly distributed. Secondly, since there are only two lists, the likelihood of encountering a Condorcet Paradox is relatively high [216]; the top two options on one list might be classes 1 and 2, and the top two options on the other list might be classes 2 and 1, leading to classes 1 and 2 being equally ranked in the final list with no deterministic/non-random way to separate them.

Two ideas can resolve these two issues. Firstly, the input lists should be accompanied by additional information that gives some sense of confidence about the rankings, for example by including the probe-gallery distances derived for each model, so that the ranking is non-linear. Secondly, not all votes need to be equal, and non-uniform weightings can be introduced to bias the fusion of the two lists towards one or the other, thus avoiding the Condorcet Paradox by ensuring that in the event of a perfect contradiction between the two lists, one list will dominate and win. It makes sense to apply these ideas in this system because the appearance and spatio-temporal models have different weaknesses at different points in time, which should be compensated for by relying more heavily on the other complementary model. Fusing the two lists into one list can be achieved by calculating the similarity score S for each class c using a linear weighting equation of the form:

$$S_c = \beta_c ST_c + (1 - \beta_c) AP_c \tag{5.6}$$

where ST and AP represent some measure of confidence that class c is the correct match for the probe identity from the spatio-temporal and appearance models respectively, and β is a dynamic weighting factor between 0 and 1 that biases the similarity score towards one of the two models. The variation in this approach comes from how ST and AP are determined, along with how the weighting factor β changes during system operation. More complicated statistical methods such as using the Central Limit Theorem, Bayes Theorem / Bayesian Networks, and Dempster-Shafer theory [217, 218] were also investigated, but they did not seem to yield significantly more accurate results, so they were abandoned since the linear weighting solution was the most computationally efficient due to its simplicity. In the prototype video analytics system, ST and AP are based on the distance between each probe-gallery pair. It should be noted here that smaller distances are better because it indicates that the probe and gallery sample are closer together and therefore more similar, which can be a little counter-intuitive since the resultant "similarity" score is actually a *dis*-similarity score. In principle, the result is a score for each class that incorporates both models, which can then be sorted to produce a new re-ranked list, from which the system selects the top ranked identity as the output class.

A problem is that the ST and AP distances can be at very different scales; for example, since the position difference is measured in mm, the distance is in the order of 10^3, while since the appearance model uses histograms that have already been normalised to be between 0 and 255, the distance is usually in the order of 10^0 to 10^1. Therefore, the distances need to be normalised to a common scale based on different scaling factors, otherwise one model will

always dominate. It is ideal to normalise the distances to a scale between 0 and 1, but this is challenging because the upper bound on the distances is so large that scaling that to be equal to 1 would force all of the calculated distances to be very small and thus compromise the sensitivity of the model. Instead, the aim can be to set the scaling factors such that they act as approximate thresholds for when the distance for a probe-gallery pair is large enough that it should be considered unlikely to be a match. The effect is that a score of 0 is a perfect match, whereas a score of 1 or larger is very unlikely to be a match. In a purely unsupervised setting, if a probe person has a score of 1 or larger with all of the gallery classes, then it would make sense to create a new identity class, although this is not a perfect measure or condition.

The similarity calculation is also strongly dependent on how β is set. In its simplest form, this can be a parameter set at system initialisation and kept constant throughout system operation. For example, a β of 0.5 would mean that both models contribute equally to the similarity score, while a β of 0.6 would recognise that the spatio-temporal model tends to be more accurate than the appearance model and thus rely on those results more heavily. However, a static weight is too inflexible and fails to properly address the weaknesses of each model. In the general case, the spatio-temporal model is very reliable and produces good classification outputs on its own. But as soon as more than one person is within the uncertainty range of the position measurements, the spatio-temporal model can easily confuse the identities and switch them or allocate the same identity multiple times within the same frame. In this case, it makes sense to rely on the appearance model more to try and help classify the identities based on their (hopefully) different clothing and general visual appearance. Similarly, if there are multiple people wearing uniforms that make them look visually similar, then the spatio-temporal model could be used to help separate the identities based on their (hopefully) different positions. If a group of people are close together and also have similar appearance, then the system should be able to return an output that indicates it is uncertain, and thus avoid an erroneous classification. In order to cover the different reasons for why a model may be weak and need more support from the complementary model, β is influenced by two functions: a decay function and a rule-based system.

5.2.2 Decay Function

In the general case, the spatio-temporal model should be relied upon more than the appearance model because kinematic modelling is likely to be more accurate than re-identification within a single camera view, especially as the number of classes becomes larger. However, the most common alternative case is where the person being tracked becomes undetectable for some time, for example by sitting down, or being temporarily obscured by an object, or leaving the camera views and then returning, or because of failures in the background estimation or person detection modules of the pipeline. The longer that a person is not detectable, the more likely that the person has travelled a further distance, and so the more inaccurate their position prediction for the next detection becomes, which is naturally reflected by an increasing level of uncertainty on that prediction. Therefore, it is logical to reduce the reliance on the

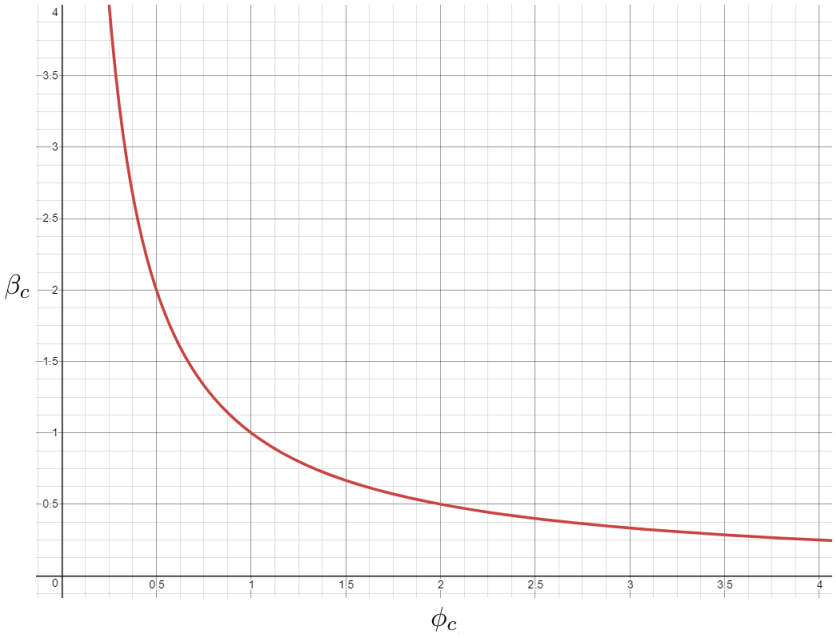

Figure 5.8: An example curve showing how β_c decreases inversely exponentially as the time since the class was last seen (ϕ_c) increases, where TF is set to 2 seconds.

spatio-temporal model over time if the system has not seen a particular identity for a while. This is achieved by using a decay function, which models an inverse exponential curve for β as time progresses since the last detection. In Equation 5.6, β is shown to be actually class-specific, rather than a common weighting that is shared between all classes. This allows for β to be based on ϕ, a variable that keeps track of the time elapsed since the person in the gallery was last detected or seen:

$$\beta_c = (2\phi_c/TF)^{-1} \tag{5.7}$$

where TF is a timeout factor that is set at system initialisation based on the expected average velocity of the people in the scene. An example of a decaying curve is shown in Figure 5.8. TF is set such that β would be 0.5 when TF seconds have passed since the last time the system saw the person. The expectation is that the system should have seen the person again within TF seconds in order for the spatio-temporal model to be reliable (or more reliable than the appearance model). This variability is only needed because in simulation (or offline processing), while ϕ is based on the physical amount of time that has passed, this doesn't take into account the processing speed of the system. If stored footage is being processed and the computer is slow or the previous algorithms in the pipeline are complex, then the equation needs to penalise the impact of ϕ appropriately so that β does not decay too quickly relative to the observed speed of the people. On live footage, where the system only samples frames when it has finished processing the previous frame, then the velocity of the person will be the same regardless of how slow the processing speed is. Mathematically, this equation would allow β to be much larger than 1 when ϕ is very small, so it is appropriate to clip β as well:

$$\beta_c = \begin{cases} \beta_{max}, \text{if } \beta_c \geq \beta_{max} \\ \beta_c, \text{if } \beta_{max} > \beta_c > \beta_{min} \\ \beta_{min}, \text{if } \beta_c \leq \beta_{min} \end{cases} \quad \text{where } \beta_{max} \geq \beta_{min} \tag{5.8}$$

where β_{max} and β_{min} are two weight constraint thresholds that are ostensibly 1 and 0. However, the reason to generalise this equation and allow for other values of β_{max} and β_{min} is that if they are equal to 1 and 0, then that allows for one of the two models to potentially be completely suppressed. It is unlikely that the confidence in either model would ever be perfect, as there is always the potential for errors to be made. Therefore, slightly more conservative values like $\beta_{max} = 0.8$ and $\beta_{min} = 0.15$ may be more appropriate to ensure that both models always have a chance to contribute to the end result. Appropriate values were experimentally determined through optimisation techniques, aiming to improve the overall system accuracy; β_{max} was between 0.7 and 0.9 and βmin was between 0.1 and 0.3.

5.2.3 Rule-based System

While the decay function covers the most common deviation from the general case, there are still many other corner cases that could be covered to help improve the accuracy of the system. Since there are a number of categorical corner cases, a rule-based system was formulated to further modify β under certain conditions, as shown in Algorithm 1. These corner cases cover initial conditions, dealing with scaling effects, and adding additional constraints when multiple classes are similar to the probe detection. Importantly, the notion of an uncertain result is also introduced, which can be used to block the model update step in order to reduce the likelihood of misclassifications being incorporated into the appearance and spatio-temporal models.

5.2.3.1 Initialising Gallery Models by Camera

At this point, it is important to note that a major change has been made to the re-identification system in comparison to what was described in the previous Chapter; rather than maintaining one global model for each identity across all camera views, a separate model is maintained for each camera view. When a person has been detected for the first time, the model for each camera view is initialised with the same sample, but after that, each camera is allowed to modify its own appearance model for that person. This improves classification accuracy for the appearance-based model, as inter-camera effects such as changes in illumination and viewpoint that contribute to variation between cameras are reduced. This is a reasonably important departure from the standard re-identification problem formulation, which usually aims to re-identify across multiple camera views. However, a robust re-identification approach is still needed, as even within the same camera view there can be significant variation in the person appearance, as previously shown in Figure 4.1. Since position information and predictions of future positions are also available in this system, the confidence for the appearance model can be reduced if the source camera view has not seen the classified person yet. If the top ranked

Algorithm 1 Computation and use of β, including the rule-based system to cover corner cases

Set a β value for each class c

1: **for** each class c in the gallery C **do**
2: **if** this camera is seeing the ST model's top ranked identity for the first time **then**
3: $\beta_c = \beta_{max}$
4: **else if** the scaled appearance distance is very close to zero **then**
5: $\beta_c = \beta_{min}$
6: **else**
7: $\beta_c = (2/TF * \phi_c)^{-1}$ {the decay function}
8: Clamp β_c between β_{min} and β_{max}
9: **end if**
10: **end for**

Cover similarity-based corner cases, overwriting previous β values if necessary

11: Calculate ψ_{AP} and ψ_{ST} {the ratio matrices}
12: Initialise w, w_{AP}, and w_{ST} to False {assume good certainty initially}
13: **if** $1.25 > \psi_{AP_{1,2}} > 0.8$ {the first and second classes appear similar} **then**
14: Form a restricted set R_{AP} of classes c where $1.25 > \psi_{AP_{1,c}} > 0.8$
15: **for** each class $c \epsilon R$ **do**
16: $\beta_c = \beta_{max}$
17: w_{AP} = True {appearance model has low certainty}
18: **end for**
19: **end if**
20: **if** $1.25 > \psi_{ST_{1,2}} > 0.8$ {the first and second classes are proximate to each other} **then**
21: Form a restricted set R_{ST} of classes c where $1.25 > \psi_{ST_{1,c}} > 0.8$
22: **for** each class $c \epsilon R$ **do**
23: $\beta_c = \beta_{min}$
24: w_{ST} = True {spatio-temporal model has low certainty}
25: **end for**
26: **end if**
27: **if** w_{AP} and w_{ST} {both the appearance and spatio-temporal models are uncertain} **then**
28: w = True {declare that the final output will have low certainty}
29: **end if**

Calculate the class c with the most similarity to the input person

30: **for** each class $c \epsilon C$ **do**
31: $S_c = \beta_c ST_c + (1 - \beta_c) AP_c$
32: **end for**

33: $R = R_{AP} \cup R_{ST}$ {Combine the two restricted sets together}
34: **Return** the highest ranking class c where $c \epsilon R$ if $R \neq \emptyset$ else $c \epsilon C$, with S_c to represent the similarity score and w to indicate the certainty on the result

identity from the spatio-temporal model has not been seen by the current camera, then β is set to β_{max} in order to emphasise the spatio-temporal model and rely less on the appearance model (Algorithm 1, Lines 2-3).

5.2.3.2 Scaling Effects

Another change is rebalancing the distances being produced by the appearance model. Since the appearance of a person is very susceptible to noise, the histograms that comprise the feature vector are not very stable. This means that while the threshold for normalisation may not have to be particularly large, the distance values themselves can fluctuate below this threshold. Empirically, it was found that the distance values for the appearance model tended to be in the top 50-60% of the dynamic range. For example, if the normalisation threshold is set to 10, then the within-class distances tend to be between 4 and 10. The 0-4 range of the distance space is essentially unused and wasted. If the distance between a probe image and the gallery image is 5, then this would be normalised to 0.5, even though to a human the images may appear to be quite visually similar. This can become an issue when fusing the appearance and spatio-temporal models, because since the distances are normalised and directly used as a measure of confidence in the rankings, then when the spatio-temporal model uses the lower end of its dynamic range it can easily but potentially erroneously win over the appearance model stuck at the upper half of its range. This can be somewhat alleviated by subtracting a minimum value from each of the appearance model distances and the threshold, thus improving the precision of the remaining range. For example, a floor could be set at 4, and simply subtracted from each of the distance values and the normalisation threshold so that a distance of 5 becomes a distance of 1, with a normalisation threshold of 6, corresponding to a 0.167 normalised score. While this helps the appearance model compete against the spatio-temporal model, any normalised distance (after rebalancing) that is very close to 0 is also extremely likely to be a match. It would be undesirable for an erroneous spatio-temporal distance measurement to cause that to be misclassified, so in this case it is best to set β to β_{min} in order to rely mostly on the appearance model and only allow the spatio-temporal model to override the appearance model rankings in very, very extreme circumstances where the spatio-temporal model is very confident in its classification (Algorithm 1, Lines 4-5).

5.2.3.3 Similar Similarity

There are two conditions that cover perhaps the most challenging cases for a person tracking and re-identification system; if people look very similar, or are physically proximate, or both, then the chances of the person tracking system producing incorrect labels increases substantially. This is represented numerically as there being multiple classes that produce similar similarity scores between the probe detection and the target model in the gallery. To check for these conditions, this system uses ratios to standarise the comparison of how similar distances are to each other. This is inspired by the use of likelihood ratios in medical sciences for validating diagnostic tests, based on Bayes Theorem [219, 220]. Ratios are popular because hard thresholds for determining similarity are inflexible and fail in situations where the scale of the values can change. For example, a 0.1 threshold for declaring two measurements as being similar means very different things if those measurements are in the 10^{-1} order of magnitude or the 10^{2} order of magnitude. Instead, a ratio (i.e. dividing one measurement by another) reduces

those measurements down to a common scale, which can then be compared to a threshold that is applicable across all scales of the original measurements. A ratio matrix ψ can be derived by dividing each probe-gallery distance by each other probe-gallery distance, as shown in this brief example:

$$\begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix} \rightarrow \begin{bmatrix} 1/1 & 1/2 & 1/3 & 1/4 \\ 2/1 & 2/2 & 2/3 & 2/4 \\ 3/1 & 3/2 & 3/3 & 3/4 \\ 4/1 & 4/2 & 4/3 & 4/4 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0.5 & 0.33 & 0.25 \\ 2 & 1 & 0.67 & 0.5 \\ 3 & 1.5 & 1 & 0.75 \\ 4 & 2 & 1.33 & 1 \end{bmatrix} \tag{5.9}$$

where the vector on the left represents four distances, the middle matrix shows how each value in the ratio matrix is derived, and the matrix on the right is the ratio matrix for that vector. The similarity between any two distances can then be determined by how close the ratio is to 1. Since each model produces a ranked list of classes, ψ_{AP} for the appearance model and ψ_{ST} for the spatio-temporal model can be calculated, and then the person tracking system can retrieve the ratio for the first and second ranked classes for each model. If the ratio is close to 1 (approx. 0.8-1.25), then both the first and second rank classes should be considered viable, and sufficiently similar solutions to the classification problem. In fact, there may be more than two classes that are very similar - if the first two classes are similar, then the other ratios for that first ranked class should also be checked. All classes c that have distance measurements that are sufficiently similar to the top ranked class can then be placed into a restricted set R. For each class c in R, its β weighting value can then be set to prioritise the appropriate model; if people appear very similar, then prioritise the spatio-temporal model, and if people are physically proximate, then prioritise the appearance model. Importantly, the final re-sorting operation is then limited to only those classes that are in R. This logically makes sense; for example, if the spatio-temporal model has identified that the probe image belongs to one of three people standing in a circle talking to each other, then it should use the appearance model scores for those three people only to determine which one it is, rather than allowing another identity to be erroneously selected. The final corner case is where the model fusion module has found both that the first and second rank people are very similar in both appearance and position. In this case, rather than risk misclassification, the output can be declared uncertain (using w), so that the person tracking system does not update the gallery models, and can record that as required in any plotting or logging operations. This allows the system to fully resolve the Condorcet Paradox. The logic described in this Subsubsection is presented in Algorithm 1, Lines 11-29 and Algorithm 2, Lines 2-4. Figure 5.9 shows an example of this rule-based system in action; a group of three people standing closely together are able to be separated based on their appearance, and the system is able to indicate when it is uncertain about the identity of a person.

Using a linear weighting approach along with the weight decay function and rule-based system, this model fusion process is able to comprehensively cover a wide variety of scenarios, and take advantage of the strengths of the appearance and spatio-temporal models when

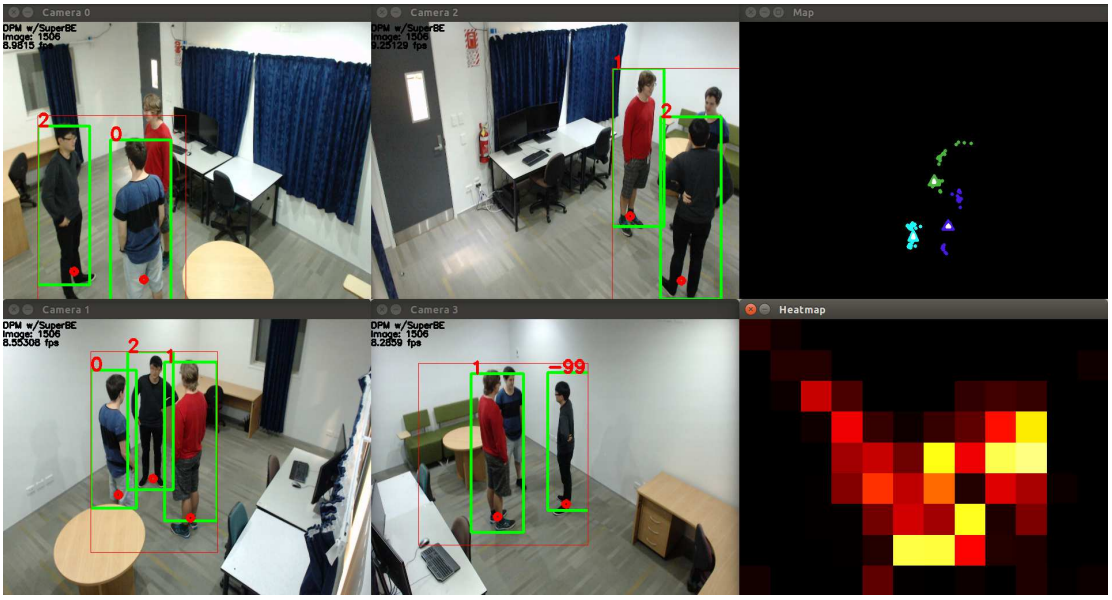

Figure 5.9: An example of a group of three people being tracked and identified correctly. In each camera view, the thin red box indicates the area that is isolated during background estimation for further processing, the green boxes represent the detected people, and the red circles represent the estimated position for each of those people. They have consistent ID numbers between the cameras (shown in red above each person), where -99 indicates low certainty (and is ignored). The maps show the positions of the individuals, with the top of the map matching the wall with curtains shown in Camera 0 and 2. The top-right map shows the last fifty detections per class as circles and the estimated future position from the Kalman Filters as triangles. The bottom-right heatmap shows the most occupied areas of the room since system initialisation, quantised to 40cm×40cm patches, where lighter/brighter colours indicate higher occupancy.

appropriate. However, it is important to keep in mind that the required output is a single identity class that matches the identity of the person in the probe image. While the two models may disagree on their rankings, a sophisticated model fusion process is only needed when they disagree on the top ranked class. A speed optimisation to the model fusion system can therefore be made by declaring an output class immediately if both the spatio-temporal and appearance models agree on the top ranked class, skipping the re-ranking process entirely since it is very unlikely that the re-ranking process would produce a different result. This shortcut further reduces the computation time of the model fusion process without compromising the accuracy. The output class can then be used to update the appropriate class in the spatio-temporal and appearance models, using the Kalman Filter and Linear Weighting (as part of Sequential k-Means) respectively.

5.2.4 One-shot vs Unsupervised Learning

As far as the model fusion module itself is concerned, there is no learning as the parameters are not being adjusted during runtime. However, the resultant similarity scores S from the

Algorithm 2 Model update and class creation with unsupervised learning

1: Receive the Similarity Score S_c, the class label c, and the certainty w from model fusion
2: **if** w is True {if there is low certainty on the result} **then**
3: Overwrite the class label with -99 to indicate low certainty
4: Do not update the models, do not create a new class
5: **else if** $S_c >$ New Class Threshold {if most similar class is above the threshold} **then**
6: Increment the total number of classes by 1
7: Create a new class ID
8: Initialise the spatio-temporal model (Kalman Filter)
9: Initialise the appearance model (Sequential k-Means)
10: **else**
11: Update the spatio-temporal and appearance models for class c
12: **end if**

fusion module can have a large impact on the learning parts of the system. This is because the main difference between one-shot and unsupervised learning is class creation; in one-shot learning, when to create a new class is given to the system (e.g. when an RFID access tag is triggered to indicate that a new person has entered the scene), whereas in unsupervised learning, the system has to determine itself when to create the new class. In this implementation, when using unsupervised learning, a new class is created if the best similarity score between the probe and the gallery classes is above a set threshold (Algorithm 2, Lines 5-9). This is not particularly flexible, and is strongly dependent on the consistency of the model fusion algorithm, but is probably the simplest way to determine when new classes should be created. Algorithm 2 shows how the system deals with both the model update (which is essentially the learning part of the system) and class creation when using unsupervised learning, following on from the results returned by model fusion in Algorithm 1. Any errors encountered by the one-shot learning approach should only be where the same identity class is given out to multiple people, whereas the unsupervised learning approach would have to contend with that problem as well as the potential of creating extra identity classes and fragmenting a single person across multiple identities. Therefore, it should be natural to expect that unsupervised learning should lead to a lower accuracy rate than one-shot learning, but in some applications unsupervised learning may be the only option available to system designers.

5.3 Experimental Results

To test the performance of the overall video analytics system, experiments were conducted on the UoA-Indoor dataset on *Walk* (where there is only one person in the room at a time) and *Group* sequences (up to four people in the room at the same time, interacting with each other). The pipeline modules tested are summarised in Table 5.1, along with their average computation times on a desktop computer running Ubuntu 16.04 with an i7-6700 CPU with four 3.40GHz cores and 16GB of RAM. It is important to remember that not all pipeline modules are used on every frame; the pipeline can exit early if there is nothing of interest. For most frames, the background estimation module is the only one being used, then person detection is run

Table 5.1: Pipeline Stages/Modules used during the main experiments on UoA-Indoor

#	Module	Method	Computation Time per frame (ms)
1	Background Estimation	SuperBE	18.5
2	Person Detection	DPM	48.2
3	Feature Extraction & Position Estimation	HSV + LBP Histograms & Calibration	0.5
4a	Spatio-temporal Classification (Distance Calculation)	Euclidean Distance	0.7
4b	Appearance Classification (Feature Matching)	PCA & Covariance & Euclidean Distance	5.7
5	Model Fusion	Re-ranking	0.28
6a	Spatio-temporal Update	Kalman Filter	0.8
6b	Appearance Update	Sequential k-Means	0.1
7	Mapping/Tracking		0.0

on any image windows that contain foreground movement, and modules 3-7 only execute if a person is found, with module 6 only run if the spatio-temporal and appearance models disagree on the classification. These stages largely match Figure 1.1, shown in the introductory Chapter. Even though both 4a and 4b use Euclidean distances, the appearance classification requires much more time because it includes the PCA and covariance metric transformations, and eventually has to compare feature vectors that are 64 elements long, whereas the spatio-temporal classification only uses two dimensions (the x and y co-ordinates). Conversely, the spatio-temporal update in 6a uses a Kalman Filter, which requires a number of matrix operations, while the appearance update in 6b just updates using a linear weighting operation between the probe feature vector and the gallery, which is much less complex and therefore needs less computation time. The mapping/tracking module is reported as requiring 0.0ms here because the time spent rounds down to 0.0 at one decimal place since the mapping/tracking operation is just maintaining a list of points, although if more sophisticated analytics are being performed (such as forming tracklets) then naturally the computation time for this module will go up. The bottleneck in this system is still the person detection stage, so further time optimisations should be made here; in the last Subsection of the results in this Chapter, the effect of swapping DPM with ACF is explored.

5.3.1 Model Fusion Performance

To test the accuracy of the classification system(s) and determine if model fusion is helpful, the *Walk* sequences from UoA-Indoor were concatenated back to back and passed through the video analytics system (19 identities). The effect is that while this experiment starts off relatively easy, by the time the last person is moving through the space, there are many identities in

Table 5.2: System Accuracy with DPM for Person Detection

Classification Model	Accuracy (%)	
	One-Shot Learning	Unsupervised Learning
Spatio-temporal Only	33.3	31.7
Appearance Only	51.9	48.8
Both Fused Together	**65.7**	**61.6**

the gallery that the system could misclassify with. Some identities also appear multiple times in the dataset (i.e. they may have more than one sequence, with other identities inbetween), which means that the system has to be able to recognise the people after they leave the camera views and re-enter later. Two sets of experiments were run; one with one-shot learning, where the presence of an auxiliary sensor such as an RFID tag or infrared sensor indicating when people are entering the space for the first time is assumed, allowing the system to know when to initialise a new class with a single labelled sample, and the other with unsupervised learning, where the system has to determine when to initialise a new class based on similarity score thresholds. There are a number of parameter values that need to be determined, including thresholds for the appearance and spatio-temporal models for normalisation, score thresholds for when to create a new identity class, and ceiling/floor weights for the β weight values in the model fusion module. These experiments were repeated thousands of times in a parameter sweep, trying uniformly random combinations of the different parameter values in the first pass to try and get a general understanding of the parameter space, and then optimising towards higher accuracy levels in a second pass. Accuracy is measured as the number of detections that are correctly classified; missed detections (false negatives) are ignored for the purposes of this metric. The best accuracy results are shown in Table 5.2, which clearly show that fusing the spatio-temporal and appearance models together improves the accuracy of the video analytics system overall. The unsupervised learning results are only about 4% points lower than the one-shot learning approach, so taking other engineering requirements into account such as the cost of installing an auxiliary sensor or difficulties in synchronising sensor signals may lead to the conclusion that the unsupervised learning approach is acceptable or even superior for certain applications.

It is important to understand the failure modes in this system to account for the accuracy rate. Analysing the results from the unsupervised learning approach has shown that it does tend to create more identity classes than the true number; in the best results, there are 37 classes in the gallery, even though there are only 19 identities in the dataset. In the accuracy measurements, the system is penalised for these extra identity classes because they are all essentially counted as errors or misclassifications. These extra classes tend to be created when a person appears to "teleport", i.e. suddenly move very far from where they currently are. This mainly happens due to the person detection algorithm not providing a stable position as the bounding box shifts, and if the box shifts then this can lead to a large change in the

estimated position. This should be treated as noise by the spatio-temporal model, so this suggests that the spatio-temporal model is still not robust enough to deal with all types of noise, and could be improved by using a more sophisticated tracking approach since the spatio-temporal classification and model update are not the speed bottleneck in this prototype. This issue does not exist in the one-shot learning approach, since the number of classes is controlled and only increases as each new person enters the camera views. However, for both types of learning, there appear to still be some difficulty in separating appearances when they are very similar, for example, when there are two identities wearing black shirts and black pants. Since the results in Chapter 4 showed that offline SVMs in a one-vs-all configuration were able to separate the classes, it appears that differences in how class boundaries are constructed (linear in the SVM case, spherical in the k-means case) may have an adverse impact on the ability of the appearance model to perform reliable classification. However, the most significant issue is probably that the spatio-temporal model gets confused by the conglomeration of identities at the entrance/exit areas of the camera views. Even though the model fusion algorithm allows for an output to be uncertain if there are multiple gallery appearances and positions that are similar to the probe sample, this is problematic if a new identity is similar but it has not yet been initialised in the gallery, especially while the Kalman Filter is still warming up. In this case, the video analytics system is likely to misclassify the new identity as one of the existing gallery classes, reducing the accuracy of the system. It should also be noted that only a subset of the relevant parameter space was explored due to time and resource constraints, yet the parameters can have a critical impact on the performance of the system. For example, with model fusion and unsupervised learning, the best result was 63.5% but the worst result obtained was only 3.6%. With further experimentation, better parameters could be found, but this can be challenging as the parameter space appears to be multi-modal (i.e. there are many local minima and maxima), and it is likely that these would still need to be modified for each new operating environment.

Comparing these accuracies to the state-of-the-art results is very difficult; the proposed video analytics system has only been tested on relatively small numbers of people and on a dataset that has not been tested with other methods. If the accuracy results are directly compared, then this pipeline may not appear to be that impressive. However, there are some key strengths that make this a promising avenue for further exploration. The vast majority of modern person tracking systems use supervised methods to artificially achieve high accuracy rates that could never be achieved outside of the laboratory. Even when transfer learning is used (i.e. training on a similar context and then performing fine-tuning training within the real environment), significant amounts of training data from the real environment are still needed, which is often simply not practically available. In addition to this, the computation time for this video analytics pipeline is almost real-time; with DPM, this pipeline runs at about 10 frames per second on a standard PC without GPU acceleration, which is significantly faster than any modern deep learning or CNN-based system. This speed has to be taken into account when interpreting these accuracy results; systems that can achieve a similar level of speed tend to have much lower accuracy rates, such as [38], which has a re-identification rate of

Table 5.3: Performance Metrics with Unsupervised Learning for DPM and ACF

Person Detection Method	System Accuracy (%)	Computation Time (ms)	Computation Time (fps)
DPM	**63.5**	102	9.8
ACF	56.7	**44.8**	**22.3**

approximately 20-40% for rank-1 (which is only shown in CMC curves; their results tables start at rank-5) with between 57 and 94 identities, and [43], which reports a rank-1 accuracy of 18.7% on their clean dataset with no false positives or occlusions across 462 bounding boxes / person detections.

5.3.2 Comparing Person Detection Algorithms

The video analytics pipeline operates at just under 10fps using DPM when there is a person being detected in the camera views. This is an average time, and can vary significantly; when there is no one in the room, then the background estimation module alone runs at approximately 50fps, while if there are multiple people in the room at different positions, then the system can slow down to 3-4 fps. While 10fps may be considered reasonably fast, this needs to be interpreted in the context that there may be multiple cameras running concurrently (in the case of the UoA-Indoor dataset, four cameras), and without parallelism (i.e. multi-core processors or multiple computers), each camera view would have to be processed one after another. This is also on a normal desktop PC; while it is not as powerful as using cloud resources or a computer accelerated with high-end GPUs, it still has more computational resources than an embedded computer. Since the end goal is to implement this system on smart cameras, more should be done to accelerate the system. As shown in Table 5.1, the person detection module is the most time-consuming part of the pipeline. Therefore, DPM can be replaced with ACF to move to the faster end of the speed-accuracy trade-off, as discussed in Chapter 3. The ACF implementation used in this research is based on the work of [87] and [94], with modification of the code to allow for non-maximum suppression and extraction of patches.

A few changes to the pipeline are needed when switching person detection from DPM to ACF. Most significantly, parts-based sampling is no longer available, as extra computation would be required to segment the detected person into semantically meaningful body parts, which would defeat the purpose of using a faster person detection algorithm. Instead, patch-based sampling is used, with 32 patches (four columns and eight rows, with 50% overlap in each direction) to segment the bounding box. This results in a much longer feature vector, but about 10,000 examples are used to perform PCA and establish a transformation matrix that can reduce the feature vectors to 64 elements long, so that it is the same length as the vectors used with DPM. New parameter values had to be found as the appearance features may vary numerically. The same experiments run with DPM are also run with ACF (i.e. random sampling of the parameter space with both one-shot and unsupervised learning). The results

Table 5.4: System Accuracy with ACF for Person Detection

Classification Model	Accuracy (%)	
	One-Shot Learning	Unsupervised Learning
Spatio-temporal Only	34.3	30.6
Appearance Only	53.8	47.1
Both Fused Together	**69.4**	**56.7**

for unsupervised learning are shown in Table 5.3; while using ACF is about 7% points lower in accuracy than DPM, it is more than two times faster. Whether the new speed-accuracy trade-off offered by ACF is acceptable will depend on the end application. Since the target for this video analytics pipeline is implementation on smart cameras, speed is potentially more important, so ACF may be more appropriate than DPM in this case.

Practically, the accuracy drop between DPM and ACF is caused mostly by there being far fewer detections/observations in ACF; for the unsupervised learning case with model fusion, DPM classified approximately 12,700 detections without high uncertainty, while ACF classified approximately 4,800 detections. This is because the ACF detection threshold has been set at a relatively high level (60 out of 100), which causes the algorithm to miss some detections (false negatives), but also avoids the worse problem of there being extra detections (false positives). False positives have to be avoided as otherwise they can cause significant changes in the gallery models; the system is robust to the occasional, rare false positive, but as the false positive rate increases the legitimate detections would become swamped. This could potentially be avoided in the future by introducing a separate "false positive class", as suggested by [43]. When there are fewer detections, the spatio-temporal model tends to suffer more, as the movement of the person appears more discontinuous and more difficult to model using a Kalman Filter. This makes it more likely for the system to erroneously create new classes, leading to more misclassifications and a drop in accuracy. Interestingly, when that challenge is removed by using one-shot learning, the accuracy rate for the ACF-based system is actually higher than the DPM-based system, as shown in Table 5.4; in this case, there being fewer detections appears to be positive since it reduces the number of opportunities for misclassification and errors. However, it is also important to note that along with more missed detections, ACF also has less accurate bounding boxes that are less stable (i.e. they are more likely to move frame-to-frame even if the person is standing still) [87], which can also negatively impact the spatio-temporal model.

Only qualitative results are available for the *Group* sequences since there is no labelled ground truth available yet. The screenshot in Figure 5.10 shows an example of the unsupervised video analytics pipeline using the DPM person detector on one of these sequences, with a different group of people in a different position to Figure 5.9. While the detection and tracking looks reasonably good, there is one misclassification in Camera 1 where the person in the grey shirt should be person 1, not person 2. Looking at the map on the top-right, it can be seen that

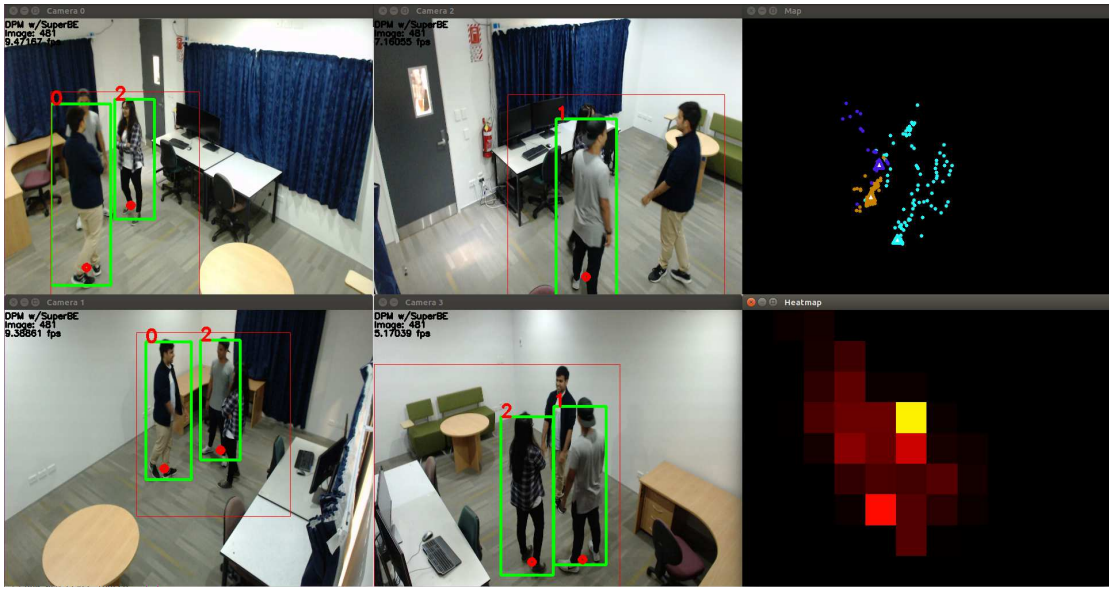

Figure 5.10: A group of people being tracked, with an erroneous classification due to the people standing closely together and the model fusion algorithm failing to separate the individuals. This figure follows the same annotations and mapping structure as Figure 5.9.

the position estimates and predictions are very close together, making classification difficult for the model fusion algorithm. When the ACF person detector is used instead, the accuracy does fall; since there are fewer detections, the uncertainty and errors associated with the position estimates become larger, negatively affecting the spatio-temporal model and causing more classes to be created than needed. Since there are also more people in the room at the same time, this causes ID switches to occur more frequently. Once an ID switch has occurred, it can be difficult for the models to recover since they are updating with the incorrect data, so other methods of improving robustness should be investigated. Figure 5.11 shows two additional examples of conditions where the footage is difficult to process; one where the camera has gone out of focus, and a second where the person detection algorithm has drawn bounding boxes that are not very accurate (for person 0, the ground plane position estimation is wrong, and for persons 1 and 2 a single bounding box has been drawn that contains both people).

5.4 Conclusions

In this Chapter, two modules of the video analytics pipeline were presented. Firstly, a spatio-temporal model was described, using calibration and mapping to convert pixel points to real-world co-ordinates, and then using Kalman Filters to help reduce noise and provide position predictions for classification. Secondly, the model fusion approach was presented, using a linear weighting approach to combine the spatio-temporal and appearance models together to improve the overall accuracy of the system. This was combined with a decay function and rule-based system to help change the weights dynamically for different scenarios. The

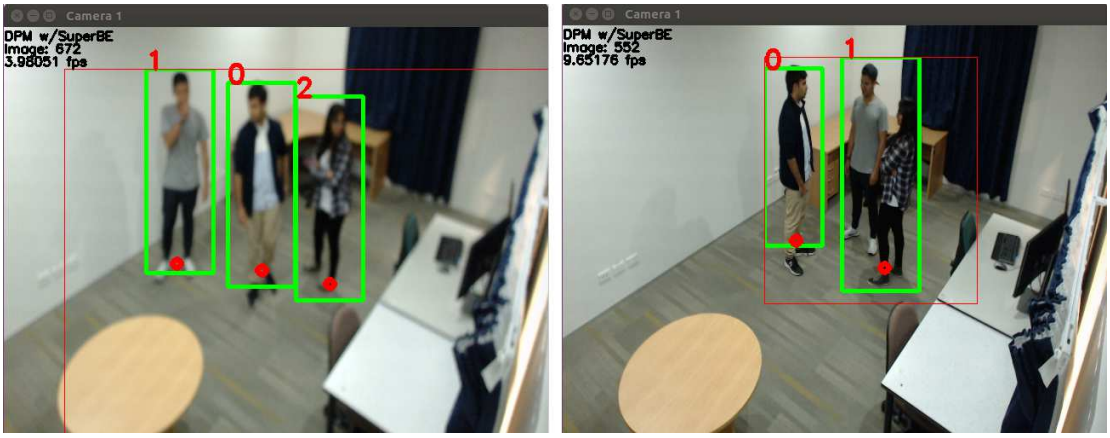

Figure 5.11: Two examples of difficult frames; on the left, the camera has gone out of focus, blurring the texture descriptors, while on the right, the person detection algorithm has drawn a single bounding box that actually contains two people, introducing erroneous/noisy data into the classification and model update process.

accuracy and computation time for these modules, along with the appearance-based person re-identification module, were found through experiments on the UoA-Indoor dataset, and show that the model fusion approach significantly improves the accuracy of the system. Lastly, both DPM and ACF person detection algorithms were compared against each other in the context of the overall video analytics system, showing two different options on the speed-accuracy trade-off curve. This also demonstrates one of the advantages of the modular pipeline approach - that modules can be swapped out to investigate different options in order to allow for the best combination of performance metrics for a particular application.

In terms of future work, perhaps the most significant is to further investigate the use of camera topology as contextual information during model fusion. At the moment, each detection of a person is processed independently; when a camera detects a person, it is classified without taking into account what the other cameras are seeing at that particular point in time, or even what else is being seen by the same camera. It could be possible to constrain identity classification with additional rules, such as "the same ID cannot be allocated twice in the same frame for the same camera view" or "at least two cameras must agree on a classification". Some preliminary ideas are explored in [221], where multiple ranked lists of people are collected for people seen in the same frame by the same camera, and then classes are penalised down the list if they are also near the top of other lists since the same identity should not appear at the top of multiple lists. It could also be possible to modify the confidence levels and weightings during the model fusion process based on the characteristics of the camera or the environment of a particular camera view; for example, it may be valid to declare that a camera view is particularly saturated by the presence of external lighting (e.g. sun coming through the windows) that makes colour appearance less reliable, so the appearance model should be penalised and the spatio-temporal model more heavily relied upon. Last of all, more options could be considered and implemented for the different pipeline stages; perhaps a particle filter is a better option for

the spatio-temporal tracking model, or a small neural network might be more suitable than Sequential k-Means for the appearance model, to improve the accuracy overall.

Now that there is a complete video analytics pipeline, from image input to output identity class, the thesis moves onto other practical implementation questions beyond algorithm design. The next three Chapters focus on privacy, distributed processing architectures, and Hardware/Software Co-Design for system acceleration. These are important topics for considering real-world implementation of the proposed person tracking pipeline in a practical and ethical sense. It is not sufficient to just propose and test the algorithm; issues like co-operation from the people being observed by video analytics systems, networking communication limits, and hardware processing platforms should also be considered as part of the system design.

Chapter 6

Privacy-Affirming Architectures

In a digital age where personal information is often traded without a second thought [222], a rising awareness of privacy and surveillance has led to resistance against corporations and governments that gain power by ignoring or dismantling privacy rights. Individuals have become increasingly conscious that their freedom to have autonomy and individuality is lost when privacy rights are eroded away [223]. A traditional definition of surveillance (beyond camera-based or CCTV surveillance) is "the collection and analysis of information about populations in order to govern their activities" [224]. This broad definition has been shown to be applicable in new contexts over the last decade; Foucauldian ideas of power and control around surveillance are not restricted to relationships with the state - corporate forms of surveillance have become more prevalent as data and information about consumers have become a valuable commodity [225]. Overreaching predictive analytics and marketing based on data about consumers, not necessarily with genuinely informed consent, has revealed how corporations are seeking much more control over their consumers. Even without the state, corporations are pushing society towards a surveillance state, where "all people are being watched", innocent or not, customer or not[20]. The discussion around privacy therefore has to evolve, as our understanding of what "a reasonable standard of privacy" constitutes continues to shift[21] and the implications of surveillance in this new context present themselves.

The use of surveillance camera networks has always been a controversial topic in the realm of privacy, as one of the most invasive forms of surveillance that challenges the trade-off between privacy loss and system utility [226]. The use of cameras is an increasingly attractive option for collecting information about people, from identifying dangerous individuals in public spaces to measuring how shoppers move through shopping malls in market research studies. Traditionally, camera footage has to be processed manually, i.e. by a person watching either recorded or live video, interpreting that footage, and then conducting some form of data entry (e.g. counting the number of people) or making some decision (e.g. dispatching security guards to investigate an area). As the number of cameras and camera networks increases, the amount of data being collected will exceed our human capacity to process it, and therefore we will

[20] Thomas Beagle, New Zealand Council for Civil Liberties, Interview on 1 Nov 2017

[21] Nicole Moreham, Faculty of Law, Victoria University of Wellington, Interview on 06 September 2017

increasingly rely on computer vision algorithms to process video footage [227]. While this shift in control over the data has implications on how surveillance cameras impact the privacy of individuals, computer vision can also offer new opportunities for reducing unnecessary breaches of privacy.

In this Chapter, the role of privacy in video analytics and surveillance camera systems is investigated. A mixed-methods survey is presented, which analyses what factors make people feel more or less comfortable about surveillance camera systems. Scenarios are used to elucidate meaningful responses by considering applications in the relevant context with multiple underlying factors. It is posited that humans need to design computer vision systems to only provide the information that is genuinely needed for a particular application. As the accuracy and speed of computer vision and image processing techniques improve, privacy control should be taken out of the hands of humans, and computers should be instructed to extract only the information necessary for human use. This is achieved through the "privacy-affirming" architecture (as opposed to "privacy-aware"), which works towards this goal and targets some of the factors uncovered in the survey by creating a hard privacy barrier between smart cameras and a central co-ordinating server.

6.1 Privacy and Computer Vision

The philosopher William Parent defined privacy as "the condition of not having undocumented personal information known or possess by others" [228]. He was not talking about a legal definition of privacy, but a notion that most people who value personal rights and freedom would be able to agree with [229] as a common, public, and collective right [230]. It can be argued that surveillance does not intrude on privacy if no undocumented information is gained; it therefore follows that if undocumented information has to be collected and made documented, then privacy breaches are unavoidable. However, this should not be considered in a binary nature; collecting a small amount of undocumented information is only a small privacy breach, and that may be preferable to a large privacy breach. The specific information becoming documented is also important; most people would not mind low sensitivity information being documented, such as the colour of their clothing or the language(s) that they speak, but they might care more about high sensitivity information, such as their identity in conjunction with their recent purchases or their credit card details [231].

This Chapter does not argue against the existence of surveillance systems generally. There may be good uses of surveillance cameras that can be combined with good system owners with good intentions. Rather, the focus is on the extraneous decisions that can be made following unintentional information capture. For example, a surveillance camera in a city centre that is intended to protect people from criminal activities may also collect footage that shows a person walking into an Alcoholic's Anonymous (AA) meeting. The privacy fear is that another human may watch the video footage and identify the individual, which may influence their perception and decisions in relation to that target individual, thereby breaching the target's privacy rights if they had wanted to keep that information private.

The rise of computer-automated processing of video footage presents an opportunity for reducing the risk of unintended privacy breaches. In the previous scenario, if a computer is programmed to run activity recognition algorithms that detect if suspicious or criminal activity is occurring, then the computer will only arrive at conclusions related to that objective. Humans have control over the machine, and can program it so that it is oblivious to the fact that a person has walked into an AA meeting. In contrast, with a human operator, even if instructions are given to only focus on suspicious or criminal activity, the human brain will inadvertently collect far more information. Most information is a non-exclusive/non-rivalrous good, meaning that it can be shared and used multiple times without degrading the value or quality of that information[22]. Even with no ill intention or malice, the operator may see an individual that they recognise walk into an AA meeting, and then the operator's opinions about that person may change, leading to changes in the operator's decision-making in relation to that person. This is clearly a breach of privacy, caused purely by the human brain's tendency to process sensory information regardless of our intentions. Some would argue that even without processing, the act of collecting excess data that is not necessary for achieving the primary objective is in itself a breach of privacy.

The majority of privacy-aware frameworks or privacy-enhacing technologies (PETs) utilise some form of censorship to reduce the amount of information available to a human operator, such that a human user cannot see and inadvertently identify any individual in captured camera footage. This includes using access controls to ensure only the right people have access to sensitive footage [232, 233], scrambling parts of video feeds to make privacy breaches less likely [234, 235, 236], and privacy-aware systems that rely on face or person detection algorithms to then censor potentially sensitive parts of an image or video feed [237, 238, 239, 240]. However, there has been significant opposition to these techniques on the grounds that they can provide a false sense of privacy; anonymised data can become de-anonymised, and information from different sources can be combined or fused together. A human operator watching partially censored footage may still receive too much information; they could piece together other information like the clothing of the person, the time and place the person was observed at, and other contextual information that lead to identification, whether that is accidental or intentional. Using post-processing to reduce the amount of information in the image can be computationally expensive, yet can still unintentionally reveal too much, especially when there are false negatives (missed detections) that cause videos to be momentarily uncensored [232, 241, 242].

There has recently been more recognition that individuals can have different views and perceptions of privacy, which is not reflected by one-size-fits-all PETs. Approaches such as visual privacy markers [243] and customised privacy advisors [244] allow for the observed to ensure that systems respect the privacy that people want on an individual level. However, these systems focus on the type of information, rather than considering the context in which that information exists. For technologists to develop appropriate protections, they need to understand the perceptions of those being observed, and understand the drivers for accep-

[22]Lindy Siegert, Principal Advisor to the Government Chief Privacy Officer, Interview on 3 November 2017

tance or concern surrounding privacy loss and utility. Otherwise, it is easy for the continued development of PETs to diverge from the desires of the general population.

For regulators, the privacy of observed individuals needs to be protected from system owners who seek to take advantage of the power that comes with a surveillance camera network and try to maximise utility and/or profit at any cost to those being observed. Conversely, there is often a commercial or political imperative for these systems to be used in a way that is palatable to the general populace, in order for system owners to receive co-operation from customers or voters. In order to understand when and how to protect privacy in these surveillance camera contexts, we need to ascertain how observed individuals perceive privacy, what factors drive their perceptions, and what changes can be made to help people feel more (or less) comfortable about the presence of surveillance cameras.

6.2 Understanding Public Perceptions of Privacy

There are many factors that can influence how someone perceives a surveillance camera system and its impacts on their privacy. We can take a simple example - there may be a surveillance camera above the entrance of a fast food restaurant for security and safety purposes. If you are just a regular customer and not a criminal, then that may not feel like an intrusion of privacy. However, if you are supposed to be on a diet, and you know that a friend works at the company who monitors the security cameras and might see you, then you might feel more wary about being seen. The human element can be a weakness in the system; human operators of surveillance systems may use information that they gather for personal gain, discriminatory targeting, and voyeurism [245]. If video analytics are applied to the footage and the data is then sold to health insurance companies who can raise your premiums if you eat too much fast food, then you might have a different reaction. The act of data collection itself may not be a primary concern; rather, it is the fact that the documented data could be used to inform or influence decisions about the recorded individual, potentially by unauthorised people, that is of paramount concern [246, 247, 248]. The underlying concern is that collected information could be used for nefarious, embarrassing, or otherwise unfair means. Even though this is physically the same camera, how it is being used, who is controlling it, and our own personal circumstances can be contextual factors that change how we perceive privacy and privacy loss.

In order to understand how people perceive surveillance cameras and privacy, a survey was conducted to consult the public and collect their opinions. Existing literature is often based on surveys that ask questions in a very direct way – on a scale of 1 to 7, rate which of these factors would make you feel more comfortable about surveillance cameras: if the camera is in a public place, if there is a big sign under the camera, if the police have access to the camera feed, and so on. The problem with this approach is that it presents each factor in isolation, when in reality they likely have interactive effects with each other. This style of questioning can also be too leading and restrictive, pushing respondents towards responding in limited ways. There is a risk that the survey designers would have to determine a list of all possible factors beforehand, which could lead to some factors being left out, and eliminate the potential for

#	Scenario Title	System Owner	Recorded Footage	Human Observer
1	Workplace Biometrics	Corporation	Unclear	No
2	Disaster Recovery	Government	Yes	Yes
3	Traffic Analysis	Government	No	No
4	Advertising Analytics	Corporation	No	No
5	Sports Tracking	Corporation	Yes	Yes
6	Pedestrian Traffic Analysis	Government	Yes	No
7	Supermarket Motion Tracking	Corporation	Unclear	No
8	Shopping Mall Trespass Enforcement	Corporation	Unclear	Unclear
9	Department Store Shoplifting Detection	Corporation	Yes	Yes
10	Intelligence Agency Person Tracking	Government	Yes	Yes

Table 6.1: A summary of the ten surveillance camera system scenarios

unexpected but important results. This survey design also requires the respondents to be asked about each factor in a similar and boringly repetitive fashion, leading to low engagement and potentially low survey completion rates.

Instead, this work used scenarios, inspired by [249], which are short paragraphs describing a surveillance camera system and how it is being used. Scenarios have been used as a research methodology to challenge the existing assumptions of the researchers and reveal novel lines of inquiry [250, 251]. This allows for a much richer survey experience that provides more contextual information, as well as allowing the respondents to infer their own details relevant to their personal life experiences. These scenarios were designed to reflect technically feasible systems that either already exist or could exist in the very near future. For each scenario, a quantitative Likert scale between 1 and 7 was used to measure the level of comfort each respondent has towards the given scenario, as well as an open-ended qualitative question that allowed respondents to (optionally) justify their selection. This led to a much wider range of responses that covered influencing factors that the survey designers did not imagine, and gave more insight into the complexities of each scenario.

For each scenario, respondents were asked about how "comfortable" they felt, rather than asking specifically about privacy implications, in order to understand the broader emotional feelings that these scenarios evoked within the respondents. Privacy often seems like a rational word, making people think about privacy rights and legislation and what is right or wrong. Privacy legislation also tends to focus on harm, on what would tangibly happen if privacy was lost[23], putting aside less tangible impacts that are harder to identify. Instead, the survey was designed to evoke deeper emotional feelings and subjective opinions, beyond the perception of what might be legal or not. Comfort is also considered an important part of understanding social license, or in other words, how easily these surveillance camera systems might be accepted by the people in a society [252].

[23]Lindy Siegert, Principal Advisor to the Government Chief Privacy Officer, Interview on 3 November 2017

One of the problems with the use of scenarios rather than asking questions about specific individual factors is the potential for confounding factors that can make analysis more challenging. For example, a scenario may contain five or six different factors, each of which could be an influencing factor that changes the respondent's perception of privacy in that scenario. Ten scenarios were developed, shown in Table 6.1, that cover a wide range of potential use cases and combinations of factors that were hypothesised to have an impact. These factors were whether the system owner was a corporation (private) or a government (public), whether the footage would be recorded or not, whether a human would be observing the footage or an algorithm would process the footage automatically, and what sort of benefit could be provided to the people being observed (i.e. what is being traded in exchange for privacy loss).

While these scenarios were described in terms of a limited number of factors, this did not exclude the respondents from being influenced by other factors, and it helped to ensure a diversity of scenarios. For example, although almost all of the scenarios had some potential for benefit to the people being observed, every scenario also had some potential for harm and misuse in different ways, which was not explicitly stated so that the respondents could draw their own conclusions. Using the responses from all of the scenarios and looking for the commonalities in the quantitative and qualitative responses, the scenarios that were more similar can be identified, and the underlying drivers/factors can be determined.

The text of the ten scenarios is listed here for completeness, and to help the reader appreciate the differences between the scenarios and how they may evoke different reactions amongst the respondents:

Scenario 1 (Workplace Biometrics): Your workplace is upgrading their access systems from the current swipe cards to biometric scanners. When you want to get into the building, there would now be two parts: firstly, as you approach the door, a camera will be watching you and analysing how you walk, as one way of identifying you. Secondly, when you get to the door, there is a fingerprint scanner that also helps check who you are. The system combines these two parts to verify your identity and provide access.

Scenario 2 (Disaster Recovery): A major weather event has caused significant damage to a number of houses in your area. Emergency services want to use drones with cameras to survey the damage quickly and identify houses that need urgent repairs, but this will include capturing footage of houses and people that are safe.

Scenario 3 (Traffic Analysis): The local traffic authority wants to be able to track cars and trucks on major city streets and highways in order to learn about the traffic patterns. They propose to do this by placing surveillance cameras on top of every traffic light and at certain points of highways, and running an automated algorithm that can count the vehicles automatically. The footage would not be recorded as the algorithm just produces a report with the number of vehicles on each road at certain times.

Scenario 4 (Advertising Analytics): An advertising company wants to be able to test how effective different advertisements in their bus shelter billboards are, and plans to use surveillance cameras to gather information about how many people are looking at an ad, for how long, and demographic information about them such as their gender, height, and ethnicity. However,

the raw footage is not stored, no human ever sees the footage, and the information cannot be linked back to identities.

Scenario 5 (Sports Tracking): The organisers of a city marathon want to use cameras to track the runners in order to prevent cheating and assist with health and safety. Trained referees will be watching the footage in real-time, and the runners are identifiable. The footage is also recorded in case it needs to be replayed during a dispute. The footage may capture the spectators as well, although there will be no identifying information for non-competitors.

Scenario 6 (Pedestrian Traffic Analysis): The city council wants to be able to measure pedestrian traffic flows around the CBD to help inform urban planning. They plan to do this by deploying person counting cameras along certain pathways. This data could also be used for other purposes within the council, such as in marketing campaigns to show the popularity of those pathways. The raw footage has to be recorded so that the data can be verified later, and may be checked by a human.

Scenario 7 (Supermarket Motion Tracking): A supermarket wants to conduct some market research so that they have a better idea of where their customers are in the store, and what areas they spend the most time in. This will allow them to design better shopping experiences for their customers. They also plan to link this with loyalty card information at the checkout so that they can provide more targeted advertising (for example, special deals for products that people were interested in and looked at, but ultimately did not buy on a previous visit).

Scenario 8 (Shopping Mall Trespass Enforcement): A shopping mall has previously issued trespass notices against individuals who have caused significant disruption to other shoppers or been caught stealing items. However, they find that security guards are ineffective at enforcing those trespass notices because people can still enter undetected too easily, and the guards cannot remember all of the people that have been trespassed. They want to use surveillance cameras to check each person as they enter the mall against their database of trespass notices automatically using an algorithm, and send an alert to security guards if someone enters the shopping mall when they shouldn't be allowed to so that they can be stopped and asked to leave.

Scenario 9 (Department Store Shoplifting Detection): An average department store can suffer from 20-30 incidences of theft a day. They currently use surveillance cameras with security officers watching the footage, but these officers often have to watch many cameras at once and are not specifically trained in detecting shoplifters. One store thinks that computers might be able to catch more people. A computer vision algorithm is proposed that would be able to detect shoplifters, and generate alerts for security guards so that they can prevent the person from leaving the store. The system would be designed to augment the existing security officers so that they can still see the footage in real-time, and humans might be able to catch the cases where the algorithm fails to detect the shoplifting. The footage would be stored for evidential purposes to be used in court proceedings where necessary.

Scenario 10 (Intelligence Agency Person Tracking): A government intelligence agency wants to be able to support police operations by tracking suspects in public spaces such as train stations and airports, in real-time. They want to deploy a nationwide camera network

Q3 Care About Privacy		Q28 More or Fewer Cameras	
Not at all	1	More cameras	55
A little	23	About the same	83
Moderately	81	Fewer cameras	73
A lot	76		
A great deal	48		

Table 6.2: Counts for the two calibration questions

that has the capability to record and store footage, as well as identify individuals against a national database using an automated algorithm, and provide locations for any person at any time. However, the intelligence agency will need to get a court warrant before being able to request the records for a particular person or track them in real-time, and only authorised individuals with the appropriate security clearance will have access to this system.

These scenarios were presented to the respondents in approximate order from less invasive scenarios to more invasive; this is primarily to avoid question order biases that could cause stronger psychological responses to more invasive scenarios transferring over to subsequent less invasive scenarios. This intention was not communicated to the respondents to avoid the opposite question order bias of respondents thinking that they should gradually feel less comfortable as they progressed through the survey. In addition to the ten scenarios, open-ended text entry questions were included that asked respondents about whether they felt that their perceptions were influenced by the owners of the surveillance network, whether recording footage had an impact, and whether having a human-in-the-loop was a factor or not. These questions were asked after the scenarios, so that respondents could reflect on whether these particular factors affected their earlier choices.

As a way of calibrating the responses, before the scenarios were presented, respondents were asked whether privacy was an important issue to them (on a 5-point Likert scale), and after the scenarios, whether they felt that there should be more, fewer, or approximately the same number of surveillance cameras in their life. These two questions were intended to be used to identify self-selection biases, as it was expected that people who care a lot about privacy might be more likely to participate in the online survey than those who do not care about privacy. The survey concluded with demographic information; most of these were free-entry, allowing respondents to answer in whichever way they were most comfortable, or to refuse to answer.

6.2.1 Calibration

The survey was distributed online over a one month period; the distribution was global, so there were a wide range of participants, although there is a geographic bias towards New Zealand in particular. A larger study could be conducted to more comprehensively understand perceptions of people around the world. The sample had a reasonable diversity of gender, age,

	C1 (N=44)	C2 (N=55)	C3 (N=83)	C4 (N=47)
Q3 Care About Privacy				
Not at all	1	0	0	0
A little	10	3	10	0
Moderately	22	21	30	8
A lot	6	23	33	14
A great deal	5	8	10	25
Q28 More or Fewer Cameras				
More cameras	31	14	10	0
About the same	9	34	36	4
Fewer cameras	1	4	29	39

Table 6.3: Cross-tab counts for the two calibration questions against the four clusters

and ethnicity, although 95% of respondents had either completed or were in the process of completing a tertiary qualification, well above the population proportion. All respondents gave their informed consent for inclusion before they participated in the study, and the study was approved by the University of Auckland Human Participants Ethics Committee (reference number 020597). After data cleaning, there were 229 valid responses that were further analysed. Firstly, the two calibration questions were used to determine how biased the dataset might be overall, shown in Table 6.2. Although the means for both questions are relatively close to the center (3.64 on a 1-5 scale for Q3 and 2.09 on a 1-3 scale for Q28), this should not be misinterpreted as a neutral result because there are significant numbers of respondents on either side of center. There is a significant bias towards people who care about privacy more, which was the expected self-selection bias. This is an important caveat that should be kept in mind when interpreting the results of the survey. It should also be noted that there are fewer respondents for Q28 than Q3, because of some respondents dropping out of the survey after answering the last scenario question (which was Q23).

Using k-means clustering, four main groups of respondents were identified based on the scenario response scores alone, labelled C1-C4. Four clusters were chosen in order to balance within-group and between-group variation, and to minimise overlap of the cluster groups. These clusters were then validated by the strong correlation with the calibration questions, as shown in Table 6.3. Even though C2 and C3 look to have similar proportions in Q3, there is a clear difference in Q28, which is supported by some significant comfort score differences in a number of the scenarios (shown in Table 6.4).

6.2.2 Demographics

One of the initial hypotheses was that privacy perceptions may differ depending on the respondent's demographic characteristics. For example, it has been argued in the literature

Question	Overall	C1 (N=44)	C2 (N=55)	C3 (N=83)	C4 (N=47)
#1 Workplace Biometrics	4.07	6.07	4.73	3.77	1.98
#2 Disaster Recovery	5.84	6.59	5.87	6.02	4.77
#3 Traffic Analysis	5.62	6.82	6.22	5.67	3.89
#4 Advertising Analytics	3.21	5.16	4.13	2.41	1.70
#5 Sports Tracking	5.08	6.50	5.36	5.02	3.51
#6 Pedestrian Traffic Analysis	4.24	6.23	4.82	3.82	2.47
#7 Supermarket Motion Tracking	3.03	5.41	3.33	2.39	1.62
#8 Shopping Mall Trespass Enforcement	3.79	6.32	4.42	3.23	1.68
#9 Department Store Shoplifting Detection	4.19	6.45	4.42	4.01	2.13
#10 Intelligence Agency Person Tracking	3.09	5.18	4.45	2.16	1.19
(1 to 7 scale, from “not comfortable” to “very comfortable”)					
Q3 Care About Privacy (1 to 5 scale, from “not at all” to “a great deal”)	3.64	3.09	3.65	3.52	4.36
Q28 More or Fewer Cameras (1 to 3 scale, from “more” to “fewer”)	2.09	1.27	1.81	2.26	2.91

Table 6.4: Mean scores for each scenario and calibration question, overall and by cluster.

that women feel safer if there is camera surveillance to deter criminals, although it has also been argued that women are wary of camera surveillance being used for voyeuristic purposes [253, 254]. In this survey data, no statistically significant correlations (at the 5% level) were found between the clusters and any of the demographic variables. This included gender, age, level of education, ethnicity, country of origin, country of current residence, occupation, and work experience with surveillance cameras. Respondents in each demographic category were distributed between the four clusters; not necessarily uniformly, but close enough to not be considered a correlative relationship. This would support an argument against the hypothesis; that in fact, no demographic group uniformly perceives privacy and surveillance camera networks in the same way, and that the factors that influence our perceptions of privacy are more nuanced and non-demographic. Perhaps this is an obvious conclusion to draw, that what you are does not necessarily define what you believe, but that has not stopped the state from justifying the installation of surveillance cameras by saying that women prefer to have surveillance cameras in their neighbourhoods for safety reasons. The only exception was those who identified as gender diverse (i.e. non-binary male/female), who were more likely to oppose surveillance cameras, although the number of gender diverse respondents was relatively small, so further work could be done to validate this.

6.2.3 Factor Analysis

Since there are a sufficiently large number of respondents in each cluster group (more than 40), the impact of the self-selection bias can be reduced by analysing these groups independently

without assuming what proportion of the population they represent. Table 6.4 shows the means for each of the scenarios and calibration questions, both in terms of the overall survey sample and for each cluster. From this data, it can be seen that overall there is some variation in means between the scenarios, and that they are not in exact order from least comfortable to most comfortable. However, this becomes more complicated when the respondents are separated out into the cluster groups. Then, the qualitative part of the survey was analysed based on the optional comments/feedback for each scenario inviting respondents to justify their choice(s). It is important to note that there may be some participation bias since not all respondents wrote comments, although if they wrote a comment for one scenario then generally they wrote comments for all ten, so factors can be compared within the same observation.

Cluster 1 has relatively low variation between the different scenarios, generally finding most of the scenarios comfortable with few concerns. When analysing the qualitative comments given for why these respondents felt comfortable, there were two main factors: that the benefit or positive value is strong enough to justify the existence of the surveillance system, and/or that these systems already exist and insufficient harm has accrued to these individuals. Some respondents saw commercial uses of extracted data as being a natural extension from existing public safety camera surveillance networks. Many respondents in this cluster mentioned that they do not have an expectation of privacy while in public, and that if they are not doing anything wrong then they have nothing to worry about from the state. In general, there is high trust of both corporations and governments to be able to run these surveillance camera systems responsibly and effectively. One particular comment summed up many of the responses: "this is a common sense use of camera tech for everyone's benefit!"

Cluster 2 was slightly more cautious than Cluster 1, but still relatively consistent across most of the scenarios. While many of the comments were still supportive, there were more concerns about potential abuse or misuse of camera systems, such as the potential for discriminatory profiling of individuals during shoplifting detection. One of the main outliers was the traffic analysis scenario, which said that information about individual vehicles would not be stored and humans would only see a "report with the number of vehicles on each road at certain times" at an aggregate level. Many respondents highlighted that they felt more comfortable in this case because the data is not personally identifiable, and saw potential for large benefits to society without much cost to their personal privacy. The main outlier scenario in the opposite direction was the supermarket motion tracking; respondents in this cluster noted that the benefits seemed to only be for the supermarket, and felt that making the tracking data personally identifiable for advertising purposes was intrusive and could lead to behaviour manipulation. These two outliers show that Cluster 2 was somewhat influenced by whether they could be identified in any camera surveillance network, and were generally okay with these systems if data was sufficiently anonymised.

Cluster 3 had some scenario means that were close to Cluster 2, but also a few where the respondents were significantly less comfortable. The commonality in scenarios 4 and 7 was that the corporations that owned the camera networks would be deriving significant commercial value out of the data extracted from these networks, while providing little to no direct benefit

to the individuals being observed. Concerns were raised about the collected data being on-sold, as well as the lack of explicit consent being given by the observed for analysis to occur about them. Underlying this was a deep sense of mistrust about the motives of corporations, believing that promises such as "the footage will not be stored" were merely facades, and that data would be used for other, more nefarious purposes. A few noted that this type of surveillance already happens online, but that in these cases there is no incentive for the observed to give up their privacy. The other scenario that caused concern among the respondents in this cluster was the Intelligence Agency Person Tracking system; with many references to Orwell's 1984 and Big Brother, respondents in this cluster said that intelligence agencies regularly break the law, and that such a system would be very susceptible to abuse/misuse. Court warrants were seen as being an ineffective safeguard because they were granted too easily, whereas Clusters 1 and 2 had more faith in the judiciary. A small number of respondents were worried that such a system could be hacked, with a wide range of potential alternative uses if in the wrong hands.

Cluster 4 were generally uncomfortable with most uses of surveillance cameras. The main theme was about trust; respondents did not trust surveillance camera operators (human observers) to follow processes and behave professionally, they did not trust camera network owners to be honest about the stated purposes or characteristics of their networks, they did not trust data to be stored securely, they did not trust safeguards or checks and balances to operate successfully, they did not trust algorithms to be functional or non-discriminatory, they did not trust governments to be accountable or competent, they did not trust corporations to not sell their data, and they did not trust anonymised data to remain anonymised. It was surprising to find a number of respondents who did not believe the scenario as written, who assumed that the system owners would be lying about their intentions. A number of the scenarios were described as "intrusive", "exploitable", "unnecessary", "manipulative", "a civil rights issue", and "deeply uncomfortable". Many commented on the apparent power disparity between camera network owners and those being observed, including that in a number of the scenarios, individuals have no choice but to be surveilled if they want to function normally in society, essentially negating the concept of consent. A few respondents were concerned about "side effects" or secondary effects of camera surveillance, with our sensitivity to pervasive surveillance dulled over time. Some claimed that the camera networks were not necessary to achieve the stated intended purposes, suggesting that lower fidelity sensors or humans could collect less data. The one exception was the disaster recovery scenario, where drones were used to survey occupied housing areas for damage after a natural disaster. Similar to the other clusters, most respondents were willing to lose some privacy in an emergency situation. The potential to save lives or provide help was deemed a sufficiently high enough benefit, even if it didn't benefit the observed individual directly. The fact that these cameras would only be used temporarily rather than becoming fixed infrastructure was also seen as a positive. The traffic analysis scenario was also viewed slightly more positively than the other scenarios, mainly because identities are irrelevant and hidden when only reporting vehicle counts.

This analysis shows that simply looking at a population mean hides the diversity of perceptions about privacy, and that there are clear ideological themes that can influence how a person

perceives and thinks about surveillance cameras. However, the exceptions in almost every cluster also show that the context is very significant in influencing perceptions, and that they can drive a person away from their baseline ideology and dominate their thinking. Even the most cynical respondents found some merits in using cameras in a disaster recovery scenario. Even the most pro-camera respondents had some concerns about tracking for commercial purposes or pervasive intelligence agency person tracking.

In summarising the qualitative evidence, five significant factors were found between the scenarios that most significantly influenced the respondents:

- Access: Who has access to the video feed or footage, including secondary data derived from the cameras? Perceptions changed if only three trusted government officials can view the footage, vs any one of ten thousand employees in a large corporation.
- Human Influence: Is there a person-in-the-loop? Will the footage be recorded and watched by humans? Generally in a public safety context, people felt better if a human is watching or the footage is recorded so that evidence was recorded, but in a commercial or corporate setting, people felt better if a computer processed the footage and no human ever saw it.
- Anonymity: Are the observed people in the footage personally identifiable or anonymous? Will there be personally targeted actions as a result? In most scenarios, respondents felt uncomfortable if they were not anonymous.
- Data Use: How will the data be used? Is the purpose in the public good or providing benefit to the observed? Are there secret secondary uses of the data? The least popular scenario was not the intelligence agency person tracking system – it was the one where the supermarket tracked consumers and tried to sell them more stuff, which is perhaps a surprising result that is contrary to the portrayal of surveillance in the media.
- Trust: Do we trust the owner of the surveillance camera network? Are they competent? Trust was the hardest factor to understand, because respondents trusted different entities in different ways; there was a wide diversity in the levels of trust for private and public entities, with contradictory reasons for trust between the respondents. But the underlying message was the same – if there is a trust deficit, where people simply do not believe what the system owner is telling them, or people do not believe that they have good intentions, then people feel uncomfortable.

While these identified factors are largely consistent with the existing privacy literature [232, 252, 255, 256], there are a few important differences. For example, being able to see and correct any personal data was not raised as a particularly important factor, and very few respondents mentioned anything about consent. This may be due to the nature of the scenarios being presented in a way that did not make it seem like respondents had a choice to interact with these systems or not, although a few respondents mentioned that they would try to avoid some of these camera systems. This is an important factor to consider, because in some cases the observed genuinely have little to no choice - for example, if they need to buy groceries but all of the nearby supermarkets have camera surveillance networks, then

there is no real choice about participation. People may even knowingly accept the privacy trade-off in exchange for a service and more convenience or for safety, but they often have no bargaining power and have to accept the terms of whatever is offered with no room for negotiation or proper assessment of value[24]. The issues around consent are particularly thorny from a legal enforcement perspective [257], and regulation is far behind the technology in terms of establishing meaningful consent. An alternative explanation for why these factors were less important may be due to changing and shifting expectations around privacy causing older philosophical principles around access to data and consent to become outdated. It may be that people have already lost their expectations of consent in the digital age.

It is also important to understand the data in the context that many people simply do not understand the downside risks of privacy loss. Often the standard that people use is "would you be embarrassed by some information about you becoming public", when perhaps a better question is "what can a company or the government do with that information, and how would it affect you?"[25]. The fact that anonymity is an important factor for many people partly speaks to this - people do not want personally targeted actions, but it may not always be clear how being observed by a camera leads to those targeted actions if they are not communicated to those being observed. Under the status quo, many people are simply unaware of these implications, and being further educated about these issues may change their perceptions of privacy.

Lastly it is necessary to acknowledge the privacy paradox [258, 259, 260, 261] that may limit the utility of this research. The paradox is that while people say that privacy is very important to them and that they are concerned, they behave in a way that does not reflect this, and are generally not motivated to pay or otherwise inconvenience themselves to protect their privacy. This often stems from a lack of understanding about the consequences, but also because privacy is relatively low in the hierarchy of rights. How survey participants respond to hypothetical scenarios can differ significantly to their behaviour in real-life. This is the justification that is sometimes given for why privacy has not been prioritised by existing system owners, because the privacy paradox clashes with the commercial imperatives that require corporations and governments to reduce costs and avoid unnecessary spending [26]. At the same time, the paradox may not matter because democratic responses to system owners are often based on the perceptions of individuals rather than their actual behaviour. In order to convince the public that a surveillance camera system is safe, they may need to appeal to the more principled and philosophical parts of the public psyche, rather than the pragmatic and practical elements.

The challenge for protecting privacy through regulation of camera surveillance networks is twofold: the population or electorate do not hold universal views towards these surveillance networks, and perceptions differ depending on the specific application and use case. Context is King, meaning that context overrides personal factors like demographics or ideology when shaping these privacy perceptions. Ultimately, these factors need to be considered carefully

[24] Blair Stewart, Assistant Commissioner, Office of the Privacy Commissioner, Interview on 19 October 2017
[25] Thomas Beagle, New Zealand Council for Civil Liberties, Interview on 1 November 2017
[26] Mohan Kankanhalli, National University of Singapore, Interview on 24 November 2018

by system designers, network owners, and regulators, in order to design camera surveillance networks that are accepted by the surveilled. Establishing trust is critical for co-operation and participation, requiring transparency about how these systems are used in order to provide enough context for the observed. Only then can we build camera systems that deliver beneficial value for all parties.

6.3 Protecting Privacy

There are two main approaches to protecting privacy. The first is the regulatory or legislative pathway, which generally restricts the use of surveillance and how data can be used. Governments can pass laws that require system owners to play by rules such as restricting access to camera systems to only necessary staff, anonymising footage by default, banning unconsented secondary uses of data, requiring footage to be deleted within a set timeframe if unused, requiring opt-in rather than opt-out approaches to consent, requiring transparency or reliability tests for algorithmic processing of footage, and so on. The difficult part is actually enforcing these laws, regularly auditing these surveillance systems to ensure that they do what they say they do, checking that they are compliant, and punishing those that turn out to be infringing upon the privacy rights of individuals.

But governments are slow, and they simply cannot respond to the pace of technological development that creates these threats and dangers. Legislators often are not expert enough in these areas, and rely on outside information that is amplified by money, which means that the information that they get is more likely to be in the interests of malicious system owners than in the interests of the general population. Additionally, public opinion surveys over the last decade have shown that camera-based surveillance is less of a concern than other forms of privacy infringement, such as internet surveillance or social media monitoring, which has caused camera-based surveillance to become less of a priority for regulators[27]. However, recent privacy breaches have shown that system owners can prove to be malicious or incompetent, and once privacy loss has occurred, it is generally very difficult to return documented information back to an undocumented state. Businesses may not have the technical capacity to select good technical choices that allow them to protect privacy, tending to use off-the-shelf products where possible, even though they are actually liable for any harm caused by these systems[28].

The second approach is the technical pathway, where the people who develop the systems ensure that privacy is "baked in". This is the vision of Privacy-by-Design [262], where legislation is complemented by ethical system design that proactively ensures that privacy is protected. As discussed earlier, this has mostly manifested in camera-based systems using privacy-aware frameworks that censor parts of an image so that observed people are not identifiable. But this approach seems to miss the point of privacy. When discussing privacy in computer vision the discussion should go beyond whether an individual is identifiable in any camera footage. What information is being received about a person in any footage? What information genuinely

[27] Blair Stewart, Assistant Commissioner, Office of the Privacy Commissioner, Interview on 19 October 2017
[28] Ibid.

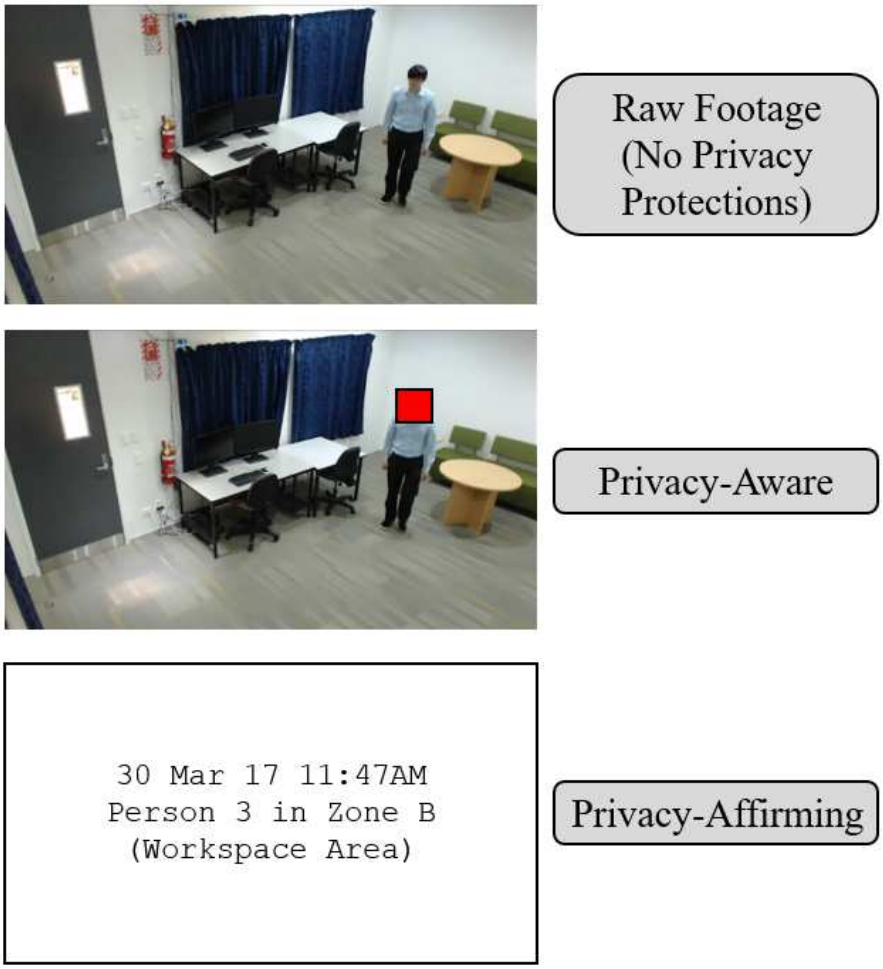

Figure 6.1: An example of the different views provided to human operators under different privacy-oriented frameworks

needs to be collected about that person? What other undocumented information might be inadvertently recorded?

In contrast to most PETs, rather than taking all of the image data and using computer vision to remove a small amount of it before giving it to a human operator, perhaps the opposite privacy-affirming proposition is better: taking all of the information and using computer vision to extract only the necessary data and giving that to the human operator, as shown in Figure 6.1. Privacy-aware and privacy-affirming are somewhat analogous to black-listing and white-listing; rather than censoring some unwanted information, explicitly only delivering the desired or necessary information may be a better approach. Under the status quo, the privacy chain of accountability ends with a human, but perhaps machines can help absolve humans of that responsibility.

It should be acknowledged that this may not be possible for all computer vision applications with current-day technology, but as computer vision continues to improve, system designers should consider the merits of an abstractive approach to protecting privacy.

6.3.1 Hiding Information

Abstraction is one of the key concepts in computing; by moving to higher and higher levels of abstraction, developers can save time (and therefore money) by hiding unnecessary information and focusing on high-level processes. This concept could be applied to video analytics too. Rather than requiring human operators to interact in some way with the raw data (i.e. the video footage), that information can be hidden with higher-level information and statistics presented instead. Hiding unnecessary information from humans follows naturally from common privacy principles; in New Zealand, the Privacy Act lists twelve privacy principles, the first of which requires that personal information not be collected unless "the collection of the information is necessary for that purpose" [263]. Similar legislative principles or requirements exist in other jurisdictions [264]. Data minimisation has been proposed as a fundamental principle for protecting privacy, and can also be pitched as a risk minimisation strategy by reducing the amount of information an organisation has to store and manage [265]. Under the status quo, by collecting camera footage and presenting that directly to human users, far more information is recorded than is necessary for the relatively narrow specific purposes used to justify the existence of these systems.

Images are incredibly rich in information in comparison to other forms of sensory data like audio or text. When humans look at an image, our brains abstract many of the details away from ourselves (also known as encoding), allowing us to focus on high-level features such as where objects of interest are, classifying those objects, identifying specific objects or people, and determining the context of the image [266]. Memory is also important, and our brains do not normally make an exact record of the past footage that has entered our eyes. We may be able to store images in short-term memory, we may remember high-level details for longer, and we may be able to reconstruct what the scene looked like with our imagination based on those details, but long-term memory does not store the vast majority of what we see [267]. It would be simply inefficient to do so, when the majority of the information that enters our eyes has little meaning or bearing on our future decisions, so our brains have learnt to discard that information. This presents a justification for how we can ask machines to abstract away information in images through the use of computer vision.

Naturally, system designers may feel uneasy at the idea of throwing away image information when it may be useful for debugging or as a backup in case of system failure. However, there are two types of machines that we already trust to discard unnecessary sensory information. Firstly, we can consider machines that are monitoring the environment, but discard data until they are activated. For example, personal assistant devices such as Amazon's Alexa utilise microphones to pick up human speech. They are actually constantly listening because they need to hear the activation/command word that enables the system. Developers trust that the system will hear that word correctly and that all other information that may be inadvertently collected while the device is listening can be discarded. At the same time, human users of these devices trust that audio not relevant to the operation of the device is discarded.

Secondly, we can consider machines where delivering raw data would be unhelpful to

human users. For example, complex algorithms are used in driver-assistance systems to process LIDAR and camera data to identify hazards, but these systems do not show raw outputs to the human user (i.e. the driver). Instead, humans trust the car to process the sensory information and draw usable conclusions, such as indicating to the driver that there is a hazard that requires the driver to slow down or stop the car. System designers would not expect the human driver to be able to interpret raw data while also driving, so the car abstracts the low-level sensor data away from the human and only presents high-level summaries (in this case, an alert). Clearly, abstraction is already an acceptable concept when it comes to dealing with machines which discard unnecessary information and hide other information from human users.

Combining these two examples presents ideas suitable for use in surveillance camera systems. For example, surveillance cameras could be always watching but do not record the footage until trigger conditions occur, such as a target individual being identified or a violent action being detected. Those cameras could also hide the raw footage away from humans, and simply present a report. For example, if police are trying to track a suspect across a camera network, the report could just list the times and positions where that particular person was detected, or show this on a map. This prevents the human operator from accessing more information than strictly necessary for the system objective to be achieved, fulfilling the promise of the principle of least privilege [268]. This can be considered to be the least intrusive form of camera surveillance, assuming that surveillance has to occur.

6.3.2 Creating a Privacy Barrier

In traditional surveillance systems, cameras are simply sensor devices; they capture images at a fixed location and transmit that footage to a central location for processing and storage. However, the centralised approach presents several challenges: as the number of cameras increases in the network, the amount of computation required by any central computer or human operator increases, and the network can become saturated as the amount of data exceeds the network capacity. Additionally, there are security concerns as a central computer is a single point of failure, so by compromising that single computer (or operator), an attacker has control and access to the entire network's data. As hardware capabilities continue to improve, and computer vision algorithms become further optimised for computational efficiency, smart cameras have become more popular; essentially cameras with some on-board processing capacity [269, 270]. These smart cameras can be used in a distributed manner, alleviating processing load on a central computer and improving security by eliminating the single point of failure. Camera surveillance research has already started to incorporate different types of smart cameras, balancing computational capacity with power and energy limitations [35, 36]. These advantages can be complementary to the objective of preserving the privacy of observed individuals.

Figure 6.2 shows a potential distributed computing layout, where smart cameras do some of the processing at the point of image capture. Smart cameras generally have constrained computational resources, so it may be inappropriate to execute a complete algorithm there,

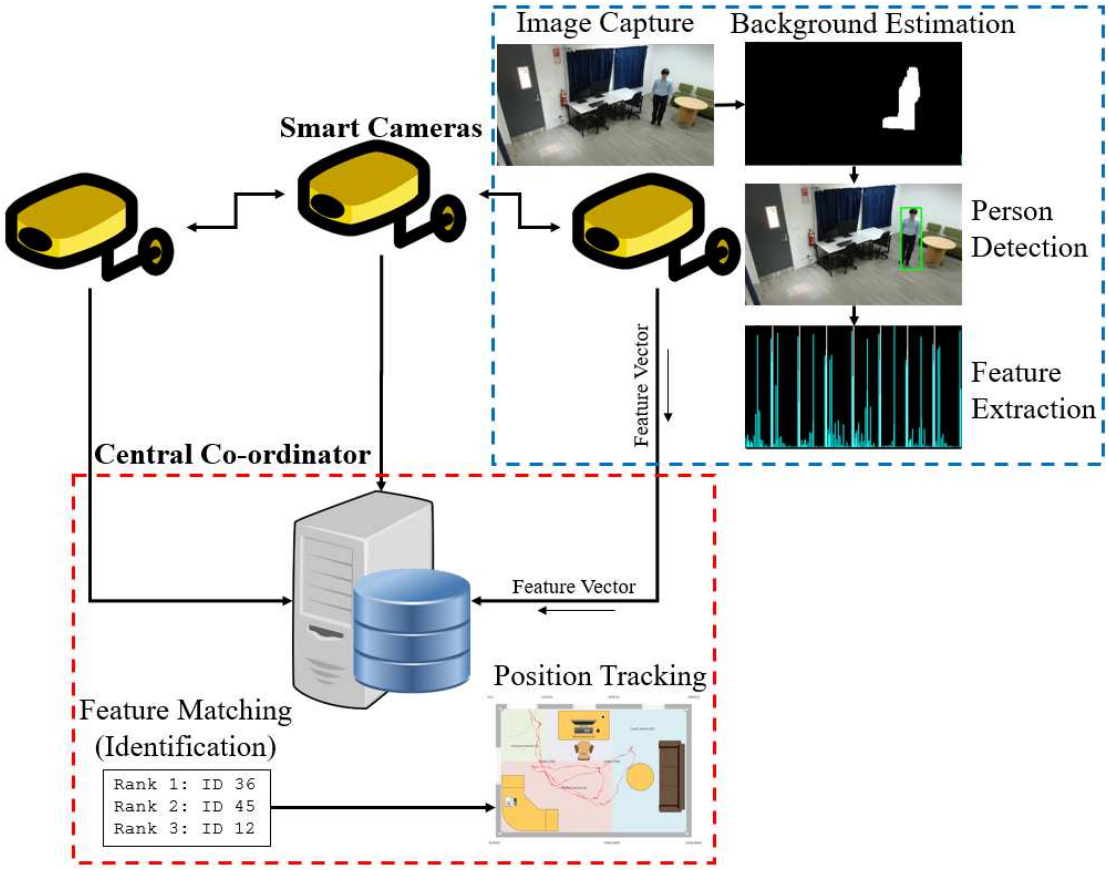

Figure 6.2: A distributed system architecture, with tasks divided between the smart cameras (blue) and the central co-ordinator (red)

and some central processing capability may be helpful for reducing the computational load on the smart cameras. Additionally, the presence of some central co-ordinator can simplify the communication protocols significantly, avoiding complex data sharing procedures. The role of the smart camera is therefore to process the image initially, extract some top-level features, and then pass those to the central co-ordinator for further processing. Importantly, the human operator only interacts with the central co-ordinator and never directly with any of the smart cameras, therefore preventing them from ever seeing the raw captured image. Additionally, the cameras have no notion of identity at the point of image capture, so if an individual camera is compromised, there is no assigned identity information available. This creates a *privacy barrier*, a term used to indicate a separation of access that means that private information cannot cross that boundary, in this case from the smart camera to the central controller.

Let us take the motivating scenario for this thesis as an example, where a large supermarket wants to track individuals as they move through a large store. The person tracking system presented thus far can be used to achieve this task, but at this stage it is all centralised. With the use of smart cameras, one potential system architecture (as shown in Figure 6.2) is for the background subtraction, person detection, and feature extraction modules to be executed on the camera itself. Feature vectors (including the person location) are then sent to the

central co-ordinator for more computationally expensive matching against a large database, and rectification between multiple camera views to centrally determine the person track. With this architecture, the smart cameras never send the raw image to the central co-ordinator, only feature vectors, which are the only information necessary from the raw image for completing the person re-identification and tracking objective of the system. The image therefore only exists in the memory of the smart camera, and can be immediately deleted after processing while still retaining the utility of the overall system. Additional security precautions such as hardware-enforced memory access control [271] can be used to restrict access to the image by foreign processes, and images are never transmitted across any networking interface, making it harder for hackers to steal those images before deletion. This has the added benefit of significantly reducing network bandwidth requirements, which is particularly important in wireless systems [35, 272]. The distributed nature of the system architecture allows for a natural separation of responsibilities in a way that also eliminates the unintentional collection of undocumented information.

This is not to say that distributed computing is a necessity for privacy-affirming frameworks. While distributed architectures may more naturally fit with privacy-affirming principles, centralised systems can still be designed to reduce the likelihood of leaking the raw image to a human user. This may be important in less modularised computer vision methods, such as state-of-the-art Convolutional Neural Networks (CNN) that may require the entire raw image as an input. If significant processing resources are required, it may not be possible to split a complicated CNN into separate modules for distributed processing without expensive co-ordination. In these types of scenarios, the system can be designed so that the raw image is processed and then discarded, with only processed outputs/statistics shown to the user, rather than overlaying that on top of the raw image as is currently often the case. However, recent advancements have allowed the separation of deep learning layers, such that some preliminary layers can be computed at the edge, with the rest of the layers completed centrally at the host [273]. When doing this, ensuring that the original image cannot be reconstructed from the partially processed features being transmitted between the edge and the host is critically important.

Jana et al [236] describe a system that intercepts video frames before they are processed (via image library APIs such as OpenCV) called Darkly, which vastly reduces the amount of information made available for processing. This is based on user-selected levels of privacy, and *privacy transforms* which use various pre-processing algorithms, such as thresholding and clustering, depending on the API call(s). Darkly is perhaps a seminal example of a system that moves towards privacy-affirming principles, abstracting away data and not only hiding it from the human user, but also potentially untrustworthy computers, in a similar way to our proposed distributed computing architecture. The Darkly paper reports that for the relatively simple applications tested, these principles can be applied successfully without significant loss of accuracy in the overall application. The challenge comes in applying these principles to more complex tasks with unspecified or broad application needs as computer vision techniques continue to evolve. Further steps such as deleting the raw image data immediately after privacy

transforms have been applied, or further abstracting away results for display to human users, could also improve Darkly's ability to affirm privacy.

The privacy-affirming architecture proposed here does not comprehensively address all privacy concerns, but it does go some way towards protecting the private information of those being observed. This can be tested using the factors identified from the survey presented earlier in Section 6.2.3. As can be seen below, how comfortable people feel about the presence of the camera system is still strongly dependent on the decisions and actions of the system owners.

- Access: Since the footage is deleted upon initial processing, no human has access to it. Access to the derived data is dependent on the system owners.
- Human Influence: In a commercial setting, people tend to feel more comfortable if the footage is processed automatically by computers, and never seen by a human.
- Anonymity: Depending on the specific video analytics application, anonymity could be enforced, although global identification can also be applied by computers.
- Data Use: This architecture is agnostic to the data use, so whether this factor is considered depends on the choices of the system owners.
- Trust: While using a privacy-affirming architecture can help improve trust, utilising this and communicating the value to those being observed is up to the system owners.

6.4 A Test for Privacy-Oriented Systems

In 2005, a team from IBM proposed a framework that used smart cameras to extract information at the point of image capture, and produce outputs at different levels of abstraction [232]. The primary limitation of their framework is that it promotes the use of permissions, providing access to data at different levels of the abstraction hierarchy. Other papers have also supported the use of access controls or usage controls to limit access to raw data to only trustworthy individuals [274, 275]. Unfortunately, this still allows for potential abuse, either by the human operators themselves who may find ways to circumvent access controls, or by external hackers who can use social engineering or network-based exploits to gain access to user accounts. Instead, the privacy-affirming architecture removes permissions entirely (whenever it is appropriate to do so) by making it so that no human, regardless of their role or rank, can have access to potentially privacy-infringing material.

But clearly there are situations when a privacy-affirming architecture is not suitable, because there are applications where it is not appropriate to abstract away information in this manner. A simple test is proposed that system designers can use when developing surveillance camera systems, in order to help decide whether a privacy-affirming approach is appropriate for a given application. There are two main factors to consider: whether the footage is required for visual evidence, and whether the footage requires human processing as part of achieving the main objective of the surveillance system. These factors are shown in Table 6.5.

Archive-critical scenarios are ones where the raw video footage is actually a requirement for meeting the principal objective or purpose of the surveillance system. For example, a law

	Archive-Critical	Archive-Free
Human Processing	Access Control or Permissions	Privacy-Aware
Machine Processing	Privacy-Aware *and* Permissions	Privacy-Affirming

Table 6.5: A rubric for making system design decisions for privacy in surveillance camera systems

enforcement camera system may require the raw footage for the purposes of evidence in court trials. Even though we already trust other types of machines to produce processed results for evidential purposes (e.g. DNA tests, GPS-based electronic monitoring of offenders, phone call metadata), the high fidelity of visual information means that it is considered to be extremely valuable, and any intentional degradation of that fidelity could be considered destruction of evidence. This is a reflection of our reliance on visual information, and in a sense people expect raw footage for court trials because that is what has been available in the past. As computer vision technologies improve, it is possible that new privacy legislation could be introduced that forces a shift in these expectations. It is often said that privacy legislation is "contagious", in that once one jurisdiction introduces new privacy legislations it is often copied by others after a few years[29]. However, under the status quo, a permissions-based system may be the only way to mitigate unintentional privacy breaches for archive-critical scenarios, especially if a human is required for processing the data. If the majority of the processing can be done by a machine (i.e. computer), then privacy-aware techniques can be used to produce anonymised footage for general use [240], while keeping the raw footage accessible via permissions-based systems for use in archive-critical applications [232].

Archive-free scenarios are ones where the raw video footage can be discarded, and the main objective is to extract high-level processed data and statistics. For example, in an urban planning application where a city council wants to measure the flow of people through a public space, retaining access to recorded footage is not necessary as long as accurate statistics can be calculated. If some human processing is required (because computer vision is not accurate or fast enough), then privacy-aware post-processing should be applied before the footage is provided for human interpretation. If the computer vision approach is accurate and fast enough, then the proposed privacy-affirming approach should be used, eliminating any exposure of the raw footage to humans. Even under existing legislation, such as the "Right to Privacy" in the European Union [276], it could be argued that the privacy-affirming approach is required wherever possible to avoid collecting information superfluous to the primary objective.

Currently, the vast majority of surveillance camera applications still require some human processing, and most large-scale networks are implemented with public safety or law enforcement objectives, so they are archive-critical. It may be that privacy-affirming approaches are not suitable for most current-day scenarios. However, two trends may tip this balance in

[29]Nicole Moreham, Faculty of Law, Victoria University of Wellington, Interview on 06 September 2017

favour of privacy-affirming principles. Firstly, the number of archive-free applications will continue to grow as the cost of camera systems fall and more businesses become aware of new opportunities enabled by smart cameras, such as market research. Secondly, our expectations of what evidence is genuinely required for archive-critical applications may change, especially if privacy-conscious activists and governments promote higher standards of protection. In the past, privacy-affirming frameworks were not technologically possible, but that barrier is quickly eroding, and since they can lead to better privacy outcomes for individuals and for society as a whole, they should be carefully considered for adoption as a baseline standard.

For privacy-affirming methods to become more popular, trust must be established with the computer in computer vision. This trust can only be earned over time, through repeated demonstrations of the validity and accuracy of computer vision applications, as has been seen with other technologies such as assembly line automation and aeroplanes. Currently, machine processing may not be accurate or fast enough for many applications, and because of that, humans may not be able to trust machines with their camera footage. High accuracy will need to be proven beyond minimum thresholds, and a certification and auditing regime may be necessary to build public confidence in these systems[30]. Computer vision systems will need to develop sufficient robustness against adversaries that seek to generate false negatives or false positives during processing.

Human operators will need to become used to the idea of making decisions informed only by processed algorithm outputs rather than seeing the raw footage themselves. Public education and marketing campaigns will be needed to convince the public that certain camera systems are privacy-affirming, and that these may protect their privacy better than privacy-aware systems (or systems with no privacy protections at all). There are (justified) concerns that integrating computer vision with surveillance may enable more information to be extracted than possible by humans [277], so privacy-conscious surveillance camera network owners and operators will need to become much more transparent about how their systems work, what data is being collected, how that data is being used, and what data is not being collected. Only then will we be able to empirically measure the impacts of a privacy-affirming framework, and whether it reduces unintentional privacy breaches in reality. Ultimately, whether privacy is respected by surveillance camera systems is dependent on the humans involved in creating that system - privacy-aware and privacy-affirming frameworks can never comprehensively protect against individuals, corporations, or states with malicious intentions.

In the late 1990s, futurist David Brin suggested two choices for managing privacy in surveillance camera systems: trusting human authorities to act with the interest of the citizenry at heart, or to democratise access by allowing the human public to “watch the watchers” in order for everyone to trust everyone else [278]. Perhaps there is a third option - to trust machines to process and extract exactly the right amount of information needed to serve human objectives, reducing the power imbalance between the observers and the observed, and thus preserving privacy from unintended and unnecessary vulnerability.

[30] Ibid.

6.5 Conclusions

In this Chapter, the notion of privacy in video analytics systems is investigated. Rather than defining a philosophical or psychological ideal of privacy, a mixed-methods survey is used to identify significant factors that influence public perceptions of privacy. It is concluded that contextual factors dominate these perceptions, more strongly than ideology or demographics. Five main factors were determined that affect how people feel about surveillance cameras: access, human influence, anonymity, data use, and trust.

Then, the privacy-affirming architecture is presented, which abstracts away unnecessary information and allows only the required information to pass through a privacy barrier for further processing and human interaction. This protects privacy much more completely and consistently than censorship-based privacy-aware approaches. This particularly targets the human influence and anonymity factors that were identified in the survey analysis. While this architecture does not solve all of the privacy problems, it can go a long way towards reducing privacy loss among people who are observed by video analytics systems. Privacy-affirming approaches can be combined with other privacy-conscious steps, such as encryption of transmitted data, event-based triggering to avoid unnecessary recording, deleting unneeded data by default after a certain time period, and using relative IDs where possible to preserve anonymity, as well as other general security precautions such as those presented in [279]. The limitations of when the privacy-affirming architecture can be used are discussed in this Chapter as well. Even though privacy-affirmed data is safer than privacy-aware data, it should still be treated as privileged and not immune from exploitation.

The person tracking system that has been described in previous Chapters is compatible with the privacy-affirming architecture proposed here. In the next Chapter, distributed computing will be used to implement parts of the pipeline on smart cameras, such that a privacy barrier blocks raw images from being seen by any human.

Chapter 7

Distributed Image Processing

One of the distinct advantages of using a modular pipeline approach over an end-to-end Convolutional Neural Network to solve the entire video analytics task is that having modules better supports meaningful distributed processing, where separate computers are used to take on some of the computation load in parallel. In the case of the video analytics system presented here, the idea is to use smart cameras that can do some of the processing at the point of image capture, with each camera sending extracted feature vectors through to a central co-ordinating server. As discussed in the previous Chapter, this can also help support ethical aims of protecting the privacy of people by helping reduce secondary privacy loss. The work in this Chapter shows how the prototype video analytics system can be implemented using low-cost smart cameras, and reports on performance statistics collected from experiments.

7.1 Smart Cameras

There are a variety of smart camera products that are now available on the market. However, many of the devices that are marketed as "smart cameras" are not actually that smart; they simply capture the image, potentially perform some basic pre-processing like distortion correction or noise filtering, and then send the footage to the cloud for further processing. They may be "smart" in other ways, such as being connected to other internet-of-things devices to automatically unlock doors for example, but that uses the term with a different type of intention. The smart cameras used in this prototype system aim to perform significant amounts of processing at the camera, reducing the load on any central computation platform and supporting the aims of high scalability, better parallelism, and faster decision making. Smart cameras have been available since the 1980s, but they still sit in the resource-constrained embedded systems space, usually due to limited form factors and the need to minimise energy consumption and cost [280]. It is common for modern smart cameras to use a simple ARM processor or similar computational device, although there has recently been more focus on hardware acceleration, such as through the use of FPGAs [281, 282]; this is discussed in Chapter 8. These computational constraints have meant that the types of applications for smart cameras have been relatively simple, such as detecting motion, reading vehicle number plates, or tracking lights [283, 284].

Figure 7.1: The ODROID-C2 board in a plastic case with a USB Wi-Fi adapter.

Smart cameras have become increasingly popular in both surveillance and video analytics systems as the need to support larger and larger networks puts more strain on central computing resources, with a push towards smart cameras becoming pervasive and ubiquitous [285, 286]. Implementing this in a scalable way requires treating the camera network as a distributed system [287], which presents different challenges around synchronisation and networking. A thorough discussion of the issues facing distributed smart camera design is available in [288].

In the prototype video analytics system presented in this thesis, a low-cost Single Board Computer (SBC) and a consumer-grade webcam are combined to form a smart camera, shown in Figures 7.1 and 7.2. This is comprised of an ODROID-C2 SBC (similar to a Raspberry Pi 3), which has a quad-core 1.5GHz ARM Cortex-A53 (ARMv8) processor on board along with 2GB of RAM, and a Logitech C920 webcam. Along with a memory card, a case for the SBC, and a 3D-printed mount, the total cost of the smart camera is approximately USD$150. The intention was to keep this cost low so that the results may be applicable in a wider variety of contexts; it is easier to buy faster and more expensive hardware to accelerate computation than to develop within tight resource constraints, as long as the higher cost is acceptable to the system owner. The SBCs offer a full Ubuntu operating system, which allows the previously developed code to be easily ported and recompiled for the simpler processor architecture. The desktop PC used for the central server in this implementation was running Ubuntu 16.04 with an i7-6700 CPU, which has four 3.40GHz cores that can be utilised concurrently (as allocated by the scheduler) and 16GB of RAM.

The next step is to separate the responsibilities between the smart cameras and the central server. As shown in Figures 1.1 and 6.2, the proposed architecture makes the smart camera extract feature vectors that correspond to each detected person, and then perform matching

Figure 7.2: A photo of the prototype smart camera used in the video analytics system. The cables leading out of the figure only provide power.

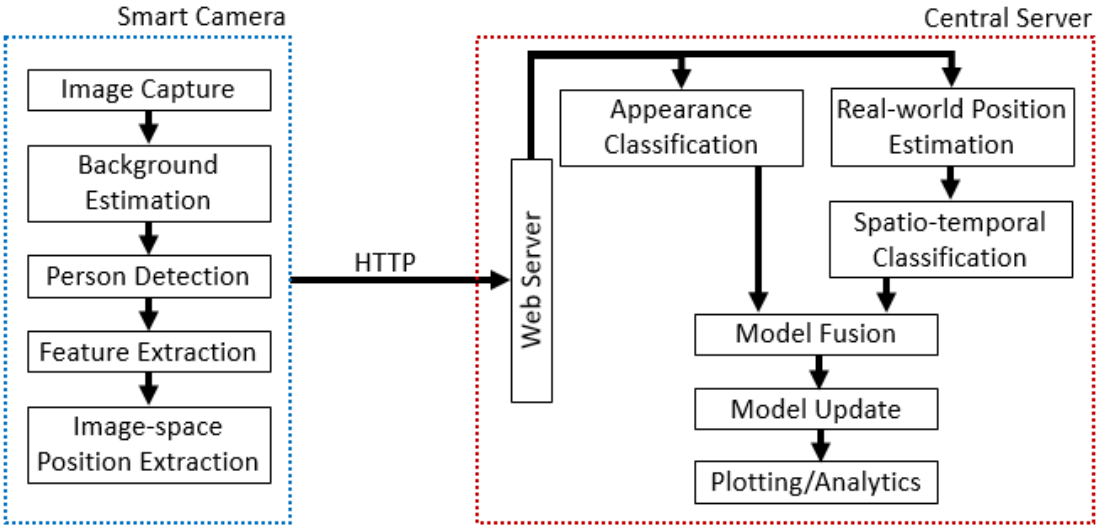

Figure 7.3: A flowchart of the separation of responsibilities between the smart camera (blue) and the central server (red)

and classification at the central server, as well as any further data analytics. This is shown in a flowchart in terms of the order of sequential execution (rather than the flow of data) in Figure 7.3. The communication is assisted with the use of a web framework at the central server, which allows the smart cameras to send the feature vectors across HTTP communication links.

While using a lower layer of the communication stack could be more efficient (such as TCP sockets), the feature vectors can be quite long, and the risk of packet drop over a wireless connection is quite high, so more robust error checking is needed. Allowing for the use of arguments/parameters also offers more flexibility; for example, the same web server can deal with smart cameras running DPM or ACF for person detection, depending on an argument value being reported alongside the feature vector. The code for the smart camera is written in C++ with the help of OpenCV to reduce the computation time as much as possible, while the code on the central server is developed in Python to allow for more developer-friendly prototyping and library availability for data analytics. JSON encoding is used to structure the payload, which is sent using the `curlpp` and `json-c` libraries on the smart camera side, and received using `CherryPy` on the central server side. Each smart camera is connected to the central server through a Wi-Fi connection, allowing for a local connection to be created without needing to expose the system to the external internet. A Python script was written to start the server first, and then start each of the cameras automatically and initialise the system.

An advantage of using a mature and established web framework is that it can handle requests from multiple sources and execute them in multiple threads, which helps avoid connections timing out and losing data; it is important to remember in Figure 7.3 that there are actually multiple smart cameras sending data through to the central server in parallel. However, in the prototype system developed, feature vectors are reported to the central server and then queued for processing, with a mutex to prevent race conditions from multiple feature vectors being processed concurrently (since the model update is part of the processing of each feature vector). The central server is also used for some rudimentary synchronisation of the cameras. This is particularly important in the case of processing dataset footage, since if one camera is many frames ahead of another camera, then the position of the person will vary significantly and thus affect the accuracy of the spatio-temporal model. Synchronisation is achieved by each smart camera sending a heartbeat to the central server for each frame processed, and the central server waiting for all cameras to report in before releasing the cameras to process the next frame. A timeout is used to cover the case where a camera becomes non-responsive or crashes, so that the system does not wait forever. While this may slow down the processing speed overall, it is important to maintain synchronicity for the spatio-temporal model. Due to a combination of a slower clock speed and simpler processor architecture, the smart camera is likely to be many times slower than the Desktop PC used for previous experiments.

7.2 Experimental Results

The proposed system was implemented with four smart cameras replacing the four cameras in the UoA-Indoor environment, and experiments run to obtain performance metrics. The cameras captured footage at an image resolution of 1920x1080, although the image processing pipeline downsample to reduce processing load. The results in Table 7.1 show that the smart camera system is about 15x slower than everything running on the Desktop PC. This is a somewhat expected but still disappointing result, as the speed is well below the requirement for real-time

Table 7.1: System Frame Rates (fps)

Computation Platform	DPM			ACF			SuperBE Only
	Worst	Average	Best	Worst	Average	Best	
Desktop PC Only	3	9.6	14	10	22.5	30	55
Smart Camera and Central Server	0.4	0.7	0.9	0.9	1.3	1.5	1.9

processing. Interestingly, the SuperBE computation time has decreased significantly, indicating that there were some compiler and processor optimisations that were available on the Desktop CPU that are not available on the embedded ARM CPU. Since the computation time of the entire pipeline is limited by the background estimation module, this has a significant impact on the overall computation time. The worst computation times presented in Table 7.1 generally represent cases where the entire image is designated foreground after background estimation, and thus person detection has to process the most amount of data under that condition. In UoA-Indoor this case is relatively rare, but could be more common in crowded scenes where there are many people.

One potential way to improve the speed of the smart camera is to take advantage of the multiple processor cores and parallelise. However, since the processing steps are largely dependent and sequential (for example, background estimation has to be completed before person detection, which has to be completed before feature extraction), manually partitioning the code and finding parallelism targets can be difficult and costly. A simpler approach is to pipeline the images being captured, such that up to N images are being processed at the same time but at different stages of processing, where N is the number of cores. Since there are four cores in the ARM Cortex-A53 CPU, up to four images could be processed in parallel, with each image delayed by 1/4 of the processing time for each frame. While a 4x speed-up should not be expected due to co-ordination costs and other background tasks needing computational resources, it would be likely that a 3-3.5x speed-up in the frame rate could be achieved, even though the actual time to process one frame would remain the same.

Since the smart cameras were deployed in a live setting, there is no accuracy data as there is no ground truth, but the accuracy can be observed empirically by storing the footage (annotated with bounding boxes and identity labels) on the smart camera and reviewing it offline. While theoretically the accuracy should not be affected since the algorithm and code is exactly the same as before, since the frame rate is so much lower, it can negatively impact the spatio-temporal model and reduce the accuracy. People may be able to move large distances between frames, which can be accounted for using parameter values for the spatio-temporal model, but these parameters can be difficult to determine without a dataset.

In terms of scalability, since the matching and classification processes are not too computationally expensive (as shown in the results of Chapter 5), the central server can potentially support a large number of cameras. However, the wireless communication bandwidth is likely

to saturate well before the central server hits a processing capacity limit. The communication time between the smart camera and the central server was measured to be 60ms on average (with a payload of approximately 32kB), although in the worst case this extended to 300ms even though it is a local network. This adds a signficant amount of latency to the system as well, although since only the feature vectors are sent this should be better than if the entire image was transmitted. The latency could potentially be reduced by using a shorter feature vector or applying PCA on the smart camera to reduce the number of dimensions, although the majority of the timing cost is due to the routing of packets between the camera and the server and is unavoidable. There have also traditionally been challenges around commercial webcam devices introducing latency through slow pre-processing steps, such as automatically correcting lens distortion or providing contrast balancing that cannot be switched off. Recently, stacked image sensor systems have been proposed as an idea to help combat these latency issues while maintaining low power consumption [289].

7.3 Conclusions

Once the algorithms for the video analytics system have been selected, a distributed architecture can help to improve scalability, reduce costs, and protect privacy. This Chapter presents the real-world implementation of this system, including the development of a prototype smart camera and networking infrastructure to demonstrate the viability of the architecture on live data. It demonstrates that it is feasible to organise a video analytics system in this way, particularly in terms of incorporating the privacy-affirming architecture. However, the experimental results show that there is a significant loss of speed when using the smart camera with fewer computing resources, well below what would be acceptable in a real-world application. In the next Chapter, acceleration while remaining in a resource-constrained embedded context is discussed. The use of Hardware/Software Co-Design allows the introduction of hardware that can further parallelise parts of the system, improving the execution speed.

Chapter 8

Hardware/Software Co-Design

The abilities of embedded computers are limited by their relatively low computational power. The most obvious target for improvement is the processor - these are often low-power, low-cost chips that have a small number of cores running at a low clock frequency. However, computer vision algorithms are often well suited for parallelism - dividing the computation up into independent threads and processing them on dedicated hardware at the same time. This is because images are made up of discrete pixels, which can be treated as separate units of information and in some cases processed independently. Unfortunately, general purpose CPUs running software are not well suited for large-scale parallelism especially at the data-level, often with a small number of cores. Instead, designers can take advantage of hardware, where specialised circuits can be designed to maximise speed.

In general, having a limited range of parallel computation options is better suited for hardware execution, while a broad range of sequential computation options is better suited for software execution. Hardware/Software Co-Design (HW/SW Co-Design) offers a balance between speed and flexibility by combining traditional software execution on CPUs or GPUs with more customised hardware accelerators on FPGAs, DSPs, or ASICs, as shown in Figure 8.1. HW/SW Co-Design has been applied in system design for embedded electronics in automobiles, avionics, and industrial automation [290], yet it is underutilised in embedded computer vision applications. There is the potential to accelerate the smart cameras in the person tracking video analytics system with HW/SW Co-Design techniques. In this Chapter, a case study for accelerating embedded vision is presented, based on the SuperBE algorithm presented in Chapter 2. Optimisations in both software and hardware are pursued systematically, with the speed and accuracy trade-off carefully considered.

8.1 Accelerating SuperBE

Many computer vision applications rely on scanning an image by applying a sliding window across the image, whether it is a simple filter operation or a more complex object detection and recognition challenge. The use of multi-scale image pyramids may mean that the entire image is effectively scanned multiple times. This is often wasteful because the regions or

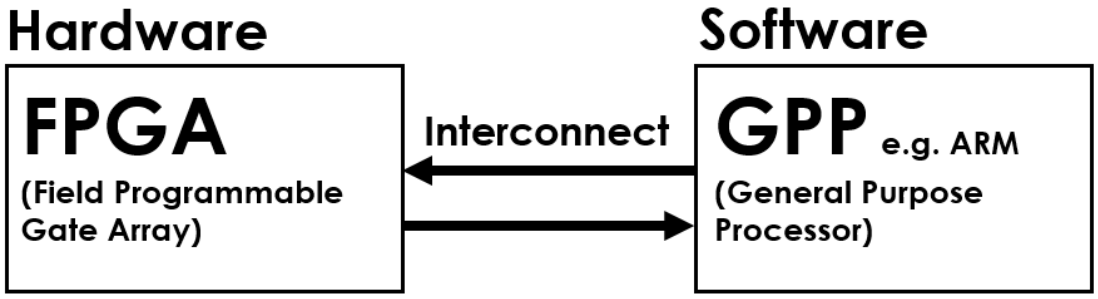

Figure 8.1: General Hardware/Software Co-Design Architecture

objects of interest only occupy some of the image space, while the majority of the camera view yields negative results. Background estimation (also known as background subtraction, background modelling, or foreground detection) is a popular method of segmenting images in order to isolate foreground regions of interest. This allows for further analysis with subsequent algorithms, saving computation time by processing a smaller image with less data and therefore iterating over fewer window positions.

However, using background estimation has two limitations. Firstly, sequential frames in a video are usually required in order to compare frames to each other and classify similar parts of the images as background, and there is a general assumption that objects of interest are moving between frames. This eliminates the applicability of background estimation in some offline image processing applications where the images contain no notion of time or may be entirely independent, but for most real-world applications there is some continuous monitoring where multiple frames of the same view are captured. Secondly, background estimation is not free; it still requires some computation time, and that computation time must be sufficiently low to justify introducing background estimation before running more complex algorithms.

While basic background estimation algorithms such as frame differences, running averages, and median filters are very fast, they tend to suffer from an inability to deal with high-frequency salt and pepper noise as well as low-frequency environmental changes such as lighting variations over time [291]. First popularised by Stauffer and Grimson [292] in 1999 and improved upon by others including [76, 293, 294], the Gaussian Mixture Model (GMM) remains one of the most popular background estimation algorithms today, primarily because of its simplicity and wide availability as one of the default algorithms available in most image processing libraries. In many applications, a GMM approach is "good enough" for background estimation, even though it still produces a substantial amount of noise from false positives and negatives. For applications where false positives or negatives are costly, such as safety-critical systems, acceptable error rates will depend on the requirements of the specific application and may need to be further reduced. This can somewhat be alleviated through the use of post-processing filters, but this adds further computation time. On the challenging CDW2014 [74] dataset, the Zivkovic GMM [76] misclassifies just under 4% of all pixels across the 11 categories and 53 video sequences. Figure 8.2 shows the difference in segmentation quality between different levels of PWC (Percentage of Wrong Classifications) on different video sequences. The figure

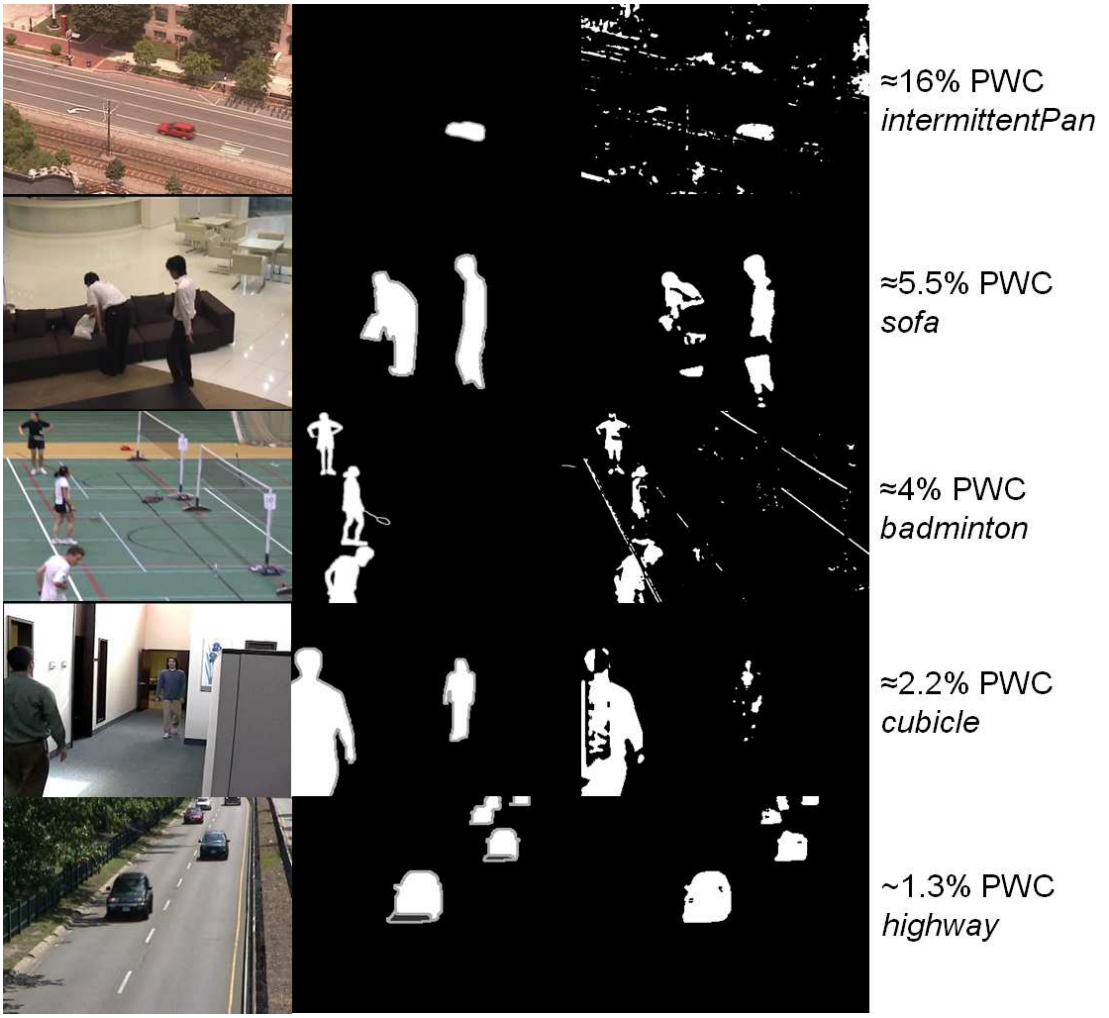

Figure 8.2: Examples of background estimation with different levels of PWC (Percentage of Wrong Classifications). In each row, the leftmost image is the raw image, the middle image is the ground truth, and the right image is the output of the Zivkovic GMM algorithm [76], with the approximate PWC and name of the sequence from the CDW2014 [74] dataset on the right.

shows that even a 4% PWC can appear to be poorly segmented, while a 1.3% PWC is perhaps sufficiently accurate.

Even though computation time is a critical factor for justifying background estimation, most of the literature focuses on incrementally improving accuracy with new algorithms at any cost. While popular pixel-level models have been around for decades, recent approaches have included applying adaptive weights and parameters [61, 295], deep convolutional neural networks [56, 296, 297], and ensemble models with stochastic model optimisation [298], all of which significantly increase computation time while only marginally improving accuracy, failing to address the challenges of real-world implementation.

Instead of blindly pursuing gains in accuracy, once a sufficient level of accuracy has been reached we should focus on accelerating those approaches, in order to minimise the impact of background estimation in more complex image processing pipelines with multiple stages. In

Chapter 2, superpixels were applied to ViBE [60], a popular background estimation algorithm, and some further optimisations of computation time were incorporated to develop an algorithm called SuperBE. The algorithm was able to reach 135fps on 320x240 images on an i7 CPU using only one core, while achieving an error rate of between 0.4 and 3% depending on the type of video. This is fast enough for real-time processing, while maintaining comparable accuracy to other state-of-the-art algorithms.

To further improve the speed while targeting embedded devices, this Chapter explores further acceleration in two directions. Firstly, since the original SuperBE algorithm used RGB images with floating-point mathematics, software acceleration can be targeted by investigating the effect of reducing the amount of information being processed, using greyscale and integer-only versions of SuperBE. Secondly, hardware acceleration can be targeted by developing an embedded system implementation of the algorithm with constrained computational resources for a real-world use case, using Hardware/Software Co-Design techniques to partition the algorithm on a System-on-Chip (SoC) with an ARM processor and connected FPGA fabric. In both cases, quantitative results justifying our acceleration strategies are presented, while also detailing the effects on accuracy. The SuperBE algorithm is described in detail in Chapter 2, so it is not repeated here.

8.2 Literature Review

8.2.1 Hardware/Software Co-design

HW/SW Co-Design is a growing field of interest because it offers a path forward for system implementations that have to be both fast and correct, while satisfying the design constraints and requirements [290, 299]. Pure software solutions tend to be more flexible, but the sequential bottlenecks may prevent real-time execution and lead to inefficient energy use. Pure hardware solutions can be very fast, but this comes at the cost of less flexibility, as well as a need to balance the accuracy (especially for floating-point arithmetic) and implementation cost. In addition, it may require significant development costs due to the high cost of hardware fabrication and the shortage of sufficiently skilled hardware engineers. By combining the software and hardware worlds, we can strike a balance between execution speed and execution flexibility in order to achieve real-time performance with sufficient accuracy.

A number of existing papers have revealed how Hardware/Software Co-Design can accelerate algorithm execution in an embedded vision context, particularly taking advantage of the parallelism often found in image processing scenarios. For example, a vehicle detection algorithm based on the detection of taillights and finding pairs of lights was accelerated using an FPGA and PowerPC HW/SW Co-Design system, achieving a 12-20x speedup in comparison to the PowerPC alone [300]. Harris Corner Detection (HCD) and Histogram of Oriented Gradient (HOG) feature calculations were implemented on a Xilinx Zynq system that included an ARM processor and FPGA fabric, achieving a 15x speedup in comparison to an octo-core i7 CPU [301]. A recent pedestrian detection implementation used an ARM processor for control and

reconfigured an FPGA on the fly, achieving a 120x speedup over a PC-only implementation [302]. While not all HW/SW Co-Design approaches lead to speedups as impressive as the above cases, the examples serve to show that in the quest for real-time image processing, HW/SW Co-Design is a valuable tool in the developers design options.

In a previous case study completed early in the Doctoral research program, Hardware/Software Co-design approaches were applied to a convolutional neural network (CNN) for a gender recognition application. In that case, a 2.9x speedup was achieved using these techniques on a three-layer network, with most of the effort put towards accelerating the Gabor filter operation in hardware. While SuperBE does not use a CNN, a number of lessons were transferrable to this context. This includes following four key steps: using execution profiling to identify bottlenecks in the algorithm for partitioning between hardware and software, using data quantisation techniques to remove floating-point operations, alleviating communication bottlenecks using data packing and smart windowing, and leveraging system parallelism where possible. In particular, being aware of the communication requirements between the hardware and software worlds allowed for better partitioning decisions to be made earlier in the development process, and more effort was put towards leveraging shared memory. For more information about the gender recognition case study, readers are referred to [303].

8.2.2 Background Estimation

Some literature does exist for describing systems that implement various background estimation algorithms on hardware platforms with the goal of achieving real-time speeds. Pham et al. [304] implement a GMM algorithm on a high-end Graphics Processing Unit (GPU) device, achieving speeds of over 50fps for high-definition video. Carr [305] reported that their GPU implementation of GMM has a 5x speed-up over CPU implementations, reaching 58.1fps for 352x288 resolution images. A competing FPGA implementation of GMM reported 20fps at a 1920x1080 resolution [306], while an FPGA implemenation of ViBe achieved 60fps on 640x480 resolution images [307]. Alternatively, instead of taking an existing background estimation algorithm and merely porting it to a hardware device, algorithm designers could take the hardware architecture into account to leverage memory structures and parallelism. An algorithm that is highly optimised for hardware using a codebook implementation on an FPGA achieved 50fps on 768x576 resolution images [308]. Using a simple convolutional filter as the main processing step in their algorithm, [309] reports 60fps on 800x480 images, although the simplicity of their approach is likely to lead to low accuracy on large images, which was unreported.

Unfortunately, the largest challenge with hardware design has generally been the high level of skill needed, and the associated high development time and cost required for well-optimised designs. While pure hardware designs can be very fast, this continuing challenge impedes adoption of these hardware systems. This can partly be addressed through the use of Hardware/Software Co-Design. In these systems, the algorithm is still predominantly software-based and controlled on a standard CPU, but parts of the processing are offloaded to specialised

hardware accelerator components that can decrease the computation time significantly. In [310], shared memory resources are leveraged to compute multi-modal background masks based on a GMM approach on an FPGA, achieving 38fps on 1024x1024 resolution images. In [311], a kernel based tracking system is implemented on an FPGA with a soft-core processor, reporting hundreds of frames per second based on a window size of 64x64 pixels, using pipelining to process multiple frames at the same time. In [312], the Mixture of Gaussians (MoG) algorithm was implemented using many pipeline stages in hardware alongside an ARM processor to achieve real-time background estimation on Full-HD images. Nevertheless, there are relatively few HW/SW Co-Design systems for background estimation published in the literature, and most of these use either the Gaussian Mixture Model (GMM) or Mixture of Gaussians (MoG), instead of more modern and sophisticated background estimation techniques.

8.3 Software Acceleration

The main strategy for reducing computation time in software is to reduce the amount of information that needs to be processed while maintaining a sufficiently high accuracy. Three new versions of SuperBE were produced: greyscale-only, integer-only, and a combined greyscale + integer version. In [60], it is reported that a greyscale variant of ViBE is approximately 15% faster than the RGB version, with a less than one percentage point increase in the error rate. In the case of SuperBE, each superpixel is described by its colour means and colour covariance matrices, and since there are three colour channels, this results in three mean values and a 3x3 matrix for each superpixel. In a greyscale version, the superpixel still needs to be described in terms of its mean and variance, but this becomes much simpler as there is only one channel. While reducing the colour means from three to one would not have a large impact on time, replacing the colour covariance matrix calculation and the covariance matrix similarity calculation with a simple single-variable variance leads to a much lower computational complexity. Removing the covariance matrix similarity calculation also means that a number of logarithm operations can be removed, which could lead to a positive effect on the accuracy because they can produce undesirable results at extreme values. In addition to this, histogram equalisation tends to make smaller changes on greyscale images than colour images, so this step was removed from greyscale versions of SuperBE to further reduce computation time.

Since the eventual target is a hardware device, it is also worth considering the effect of casting/rounding all numbers in an integer-only version of the algorithm, otherwise known as quantisation. For a standard desktop CPU, floating-point mathematics is well optimised, so there may not be a large speed improvement. On many smaller processors used in embedded devices, simpler processor architectures may not include specialised hardware for floating-point operations, causing these operations to be extremely costly. This was seen in the results presented in Chapter 7, where the slow-down between the desktop CPU and the embedded CPU was more than just the difference in clock frequency. Implementing floating-point mathematics on hardware is also much more resource-intensive than fixed-point mathematics, restricting

Table 8.1: Software Acceleration Results

Version	PWC (%)	Speed (FPS)	Relative Speed
Without Post-processing			
Reference	1.75	53.00	1.00
Integer-only	1.23	81.25	1.53
Grayscale	1.88	184.65	3.48
Grayscale + Integer	2.33	232.39	4.38
With Post-processing			
Reference	1.66	28.99	1.00
Integer-only	1.08	37.35	1.08
Grayscale	2.42	53.12	1.83
Grayscale + Integer	2.51	52.12	1.80

most designs to fixed-point or integer-based arithmetic [313]. It should be expected that there will be some increase in error as a result of losing precision [314], but it is likely to be small since background estimation is generally looking for relatively large changes in features. However, casting or rounding of the numbers is not the only effect of moving towards an integer-only version of the algorithm; operators such as log and square root also need to be approximated with integers, which could potentially lead to larger errors in output.

8.3.1 Software Evaluation

The effect of these optimisations was tested on the CDW2014 dataset [74], excluding the three categories PTZ, intermittent object motion, and thermal, which were also omitted in Chapter 2 as SuperBE is unsuitable for these video types without regular re-initialisation of the background model. The tested video sequences included low frame rates, shaky cameras, poor image quality, and a variety of image resolutions ranging from 320x240 to 720x576. As shown in Table 8.1, experiments were conducted both with and without post-processing, where the "reference algorithm" is the unmodified algorithm from Chapter 2. The main metric that used for accuracy was the Percentage of Wrong Classifications (PWC), which is equivalent to the error rate, calculated by dividing the number of incorrectly classified pixels by the total number of pixels and converting to a percentage. The speeds given in FPS are normalised for 320x240 resolution images; since the dataset is comprised of multiple video sequences of varying resolutions, it is important to normalise the results to a common resolution for fair comparison. The relative speed for each version is given relative to the reference version. All experiments were conducted on the same laptop computer, with a 2.4GHz i7-4700HQ CPU, 16GB of RAM, running Linux Kubuntu 17.04. The algorithms are implemented in C++, compiled with -O3.

As expected, the software optimisations significantly increased the speed of the algorithm.

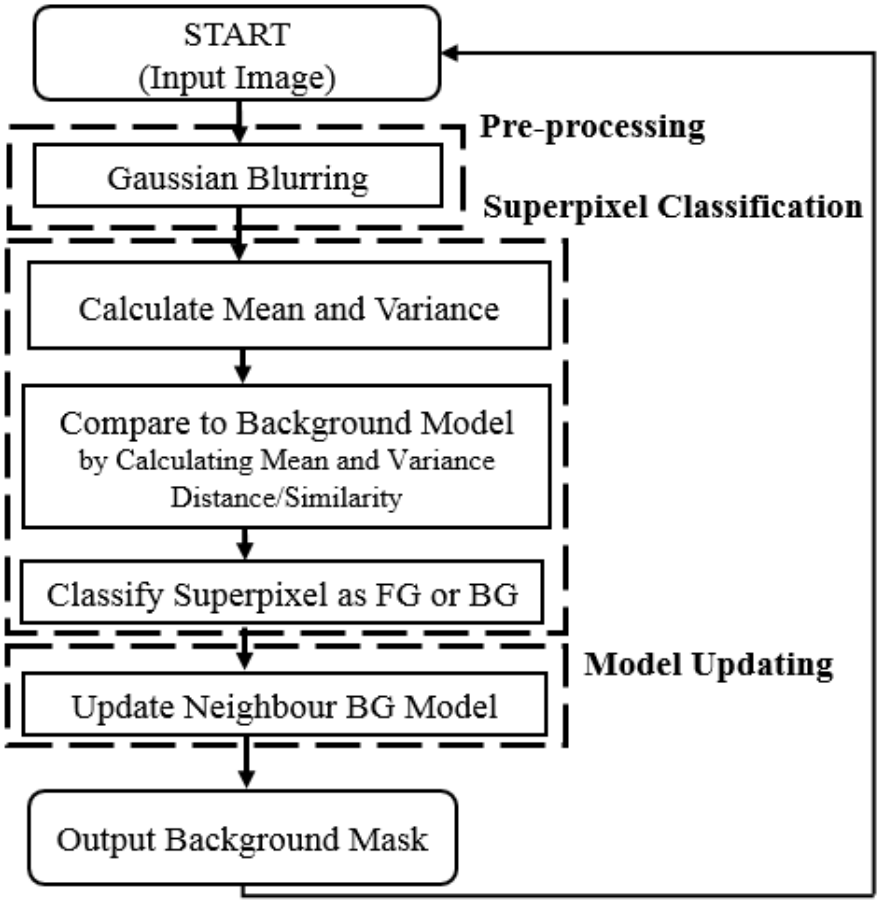

Figure 8.3: Flowchart showing the simplified Greyscale + Integer algorithm used for embedded implementation

In the integer-only case, there is an unexpected improvement to the accuracy as well - this was identified to be due to the approximated log function in the integer version, which was clamped to not return negative values, whereas the reference version included a log function that would sometimes give very negative values that could cause misclassification of superpixels. The grayscale version does increase the error rate, but this is still low enough to be suitable for many applications. It appears that the grayscale optimisation has a much larger impact on speed than the integer optimisation, although this is probably because the CPU in the laptop has specialised floating-point operation hardware, which may not be available in a simpler processor.

It should be noted that with post-processing, the grayscale and grayscale + integer cases have very similar error rates and speeds, performing far worse than without post-processing. This would suggest that the current post-processing scheme of morphologically closing and then opening the background mask may not be as suitable when applied to an output derived from grayscale data. It appears that this post-processing method also does not justify itself in terms of the computation time, as in the reference and integer-only cases it only reduces the error rate by about 0.1 to 0.2% but slows down the algorithm by 45 to 55%.

Table 8.2: Runtime Analysis of SuperBE

Task	**Percentage Runtime (%)**
Gaussian Blurring	4.17
Superpixel Classification	
- Mean/Variance Calculation	50.58
- Similarity Calculation	8.35
- Superpixel Classification	0.12
Model Updating	0.17

For a standard case where SuperBE is being used on a desktop PC or in the cloud, the grayscale version without post-processing is likely to be sufficient in terms of accuracy while delivering very fast speeds. If it is desirable to add the integer optimisation on top for hardware implementation, then the effect on accuracy is relatively limited and likely to be acceptable in exchange for the further improvement in speed. Based on these results, the grayscale + integer version without post-processing was selected for embedded implementation and hardware acceleration. A flowchart of the resultant simplified algorithm is shown in Figure 8.3.

8.4 Hardware Acceleration

In order to execute SuperBE in real-time in an embedded context, hardware can be used to accelerate execution. One option is to implement the entire algorithm using hardware only, such as on an FPGA or in an ASIC. However, this presents the immediate problem of a high cost barrier, both in terms of development time and hardware cost. Both of these cost factors can be significantly reduced by utilising Hardware/Software Co-Design principles. For this case study, an FPGA chip with an embedded hard core processor is used to demonstrate how the algorithm can be partitioned between hardware and software; these techniques can be applied to other applications and Hardware/Software combinations to also achieve speed-ups in an embedded context. The FPGA acts as reconfigurable hardware fabric, which can be programmed to implement digital circuits.

8.4.1 Algorithm Partitioning

Given that there are both hardware resources in the form of FPGA fabric and software resources in the form of a typical processor, the main question to be addressed is how to partition the overall algorithm between hardware and software. A management decision has to be made about the amount of time and other resources available to the developer(s). As part of this decision, designers should understand where the most efficient performance gains are, i.e. which parts of the algorithm can be implemented in hardware the most easily for the most speed improvement. Execution profiling allows developers to identify bottlenecks in the algorithm.

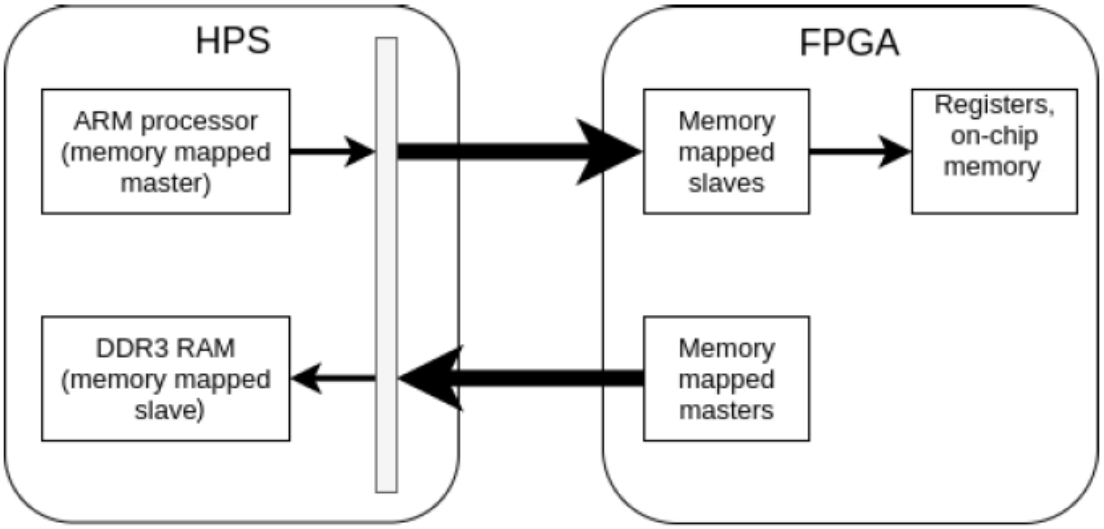

Figure 8.4: Block diagram showing the software (HPS) and hardware (FPGA) partitions of the Cyclone V SOC device

This requires an initial implementation of the algorithm in software; in this case, the algorithm from the previous Section was passed through `Valgrind` with `Callgrind` to perform execution profiling across a few thousand frames. The results are shown in Table 8.2. Note that the values in this table do not sum to 100% because the steps that cannot be accelerated in this system have not been included, such as reading the image from memory, or initialising matrices and vectors.

In addition to a timing analysis, the communication requirements need to be taken into consideration. While the FPGA may be able to process data in parallel, it must receive that data in, and output it afterwards. The communication link(s) between the HPS and FPGA can become a bottleneck when large amounts of data need to be transmitted, as is common in image processing applications [315]. In this case study, the target execution platform is the DE1-SoC development board, which has an Altera Cyclone V 5CSEMA5F31C6 System-on-Chip (SOC) device. This device includes a dual-core ARM Cortex A9 (which is referred to as the hard processor system or HPS) and FPGA logic cells, DSP blocks, and memory resources. A conceptual diagram of the HPS-FPGA system is shown in Figure 8.4, where the AMBA AXI bridges between the HPS and FPGA are shown in bold arrows. These bridges allow two-way communication, so that the master side can send an instruction to request data and have the result returned on the same bridge. Therefore, communication between the HPS and FPGA is not single-cycle, creating an overhead for each transaction. For example, a write transaction requires at least six clock cycles, which add up quickly when there are millions of pixels in each frame. If the link between the HW and SW is poorly designed, then the communication channels can easily become the bottleneck that prevent faster speeds from being achieved. If multiple non-sequential tasks are partitioned onto the hardware, then data needs to be passed between the HW and SW units multiple times. Therefore, it is desirable, where possible, to complete a contiguous block of the algorithm together on the HW accelerator, and then pass

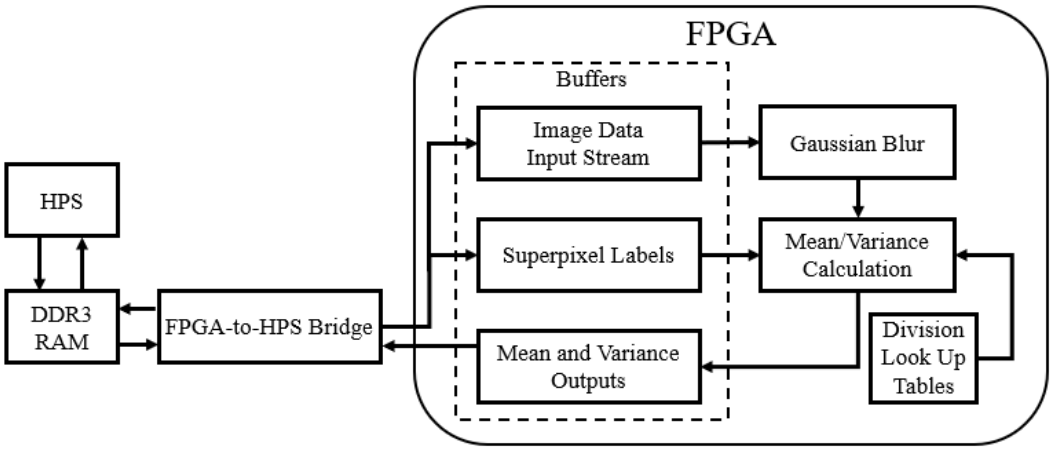

Figure 8.5: Top-level architectural diagram of the hardware partition, with arrows representing data flow

the data to the SW processor for completion.

Taking the computation and communication times into account, the decision was made to accelerate the two earliest stages of the algorithm: Gaussian blurring and the mean/variance calculation. It makes sense that these are the stages that may require the most computation time because they process the largest amount of data - after the mean/variance calculation, the algorithm represents each superpixel with two numbers, rather than all of the pixel values within the superpixel, essentially reducing the amount of data that needs to be processed in subsequent steps. The similarity calculation step was not accelerated because there would be significant communication and memory overheads, as the background model values would need to be either transferred between the HPS and FPGA regularly or duplicated and updated on the FPGA side as well as the HPS side. A high-level block diagram of the hardware partition is shown in Figure 8.5, showing the data flow between the HPS and FPGA as well as the different hardware components. The hardware components are described in VHDL, and then synthesised onto the FPGA fabric. The buffers shown are modular Scatter Gather DMA (mSGDMA) IP blocks from Intel (Altera) that provide interfacing between the HPS and FPGA, allowing the memory-mapped interface of the HPS to feed into a streaming FIFO buffer on the FPGA. There are some savings in communication time by interfacing the FPGA with the RAM module so that image data can be read directly [301, 316], rather than transferring the data from the RAM to the HPS and then through the HPS-to-FPGA bridge. Control signals are omitted, since the only control signal comes from the HPS to the FPGA to tell the components to reset and start again when a new image is being transferred across.

The Gaussian blur component was implemented using a sliding window approach [317], shown in Figure 8.6. A shifting (or sliding) window significantly reduces the amount of data that needs to be provided to the FPGA, at the cost of a few hardware resources for control. When windows from an image are passed from the HPS to FPGA, it makes sense to iterate from the top-left corner to the bottom-right corner. As the window moves across a row of pixels, the windows are overlapping; out of 25 pixels, only 5 are actually new. This can be

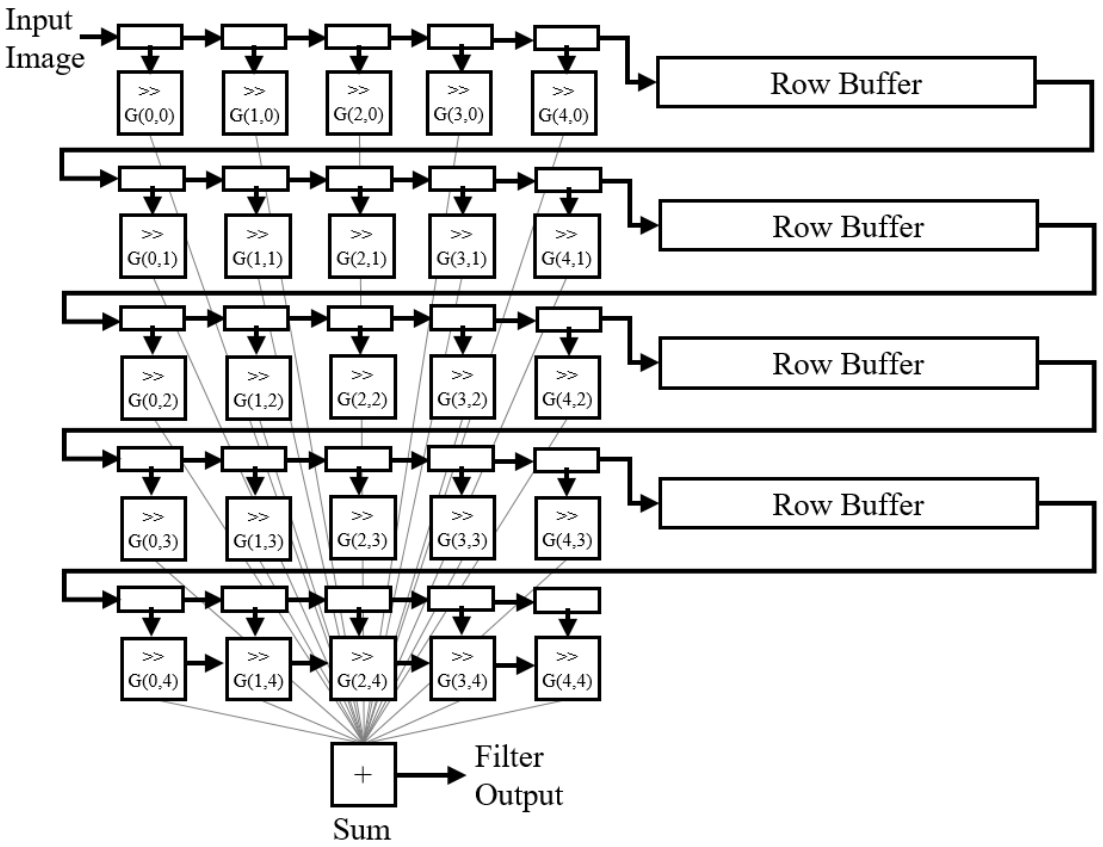

Figure 8.6: A block diagram showing a 5x5 Gaussian blur operator in hardware, where >> indicates a right shift and G is a matrix representing the Gaussian kernel

exploited by adding shift registers into the filter on the FPGA so that redundant transactions can be avoided. An additional transaction is required in order to control the FPGA and indicate that a shift should occur, but overall this reduces the number of required memory transactions significantly. To further reduce hardware resource consumption, instead of implementing floating-point multipliers (since a standard Gaussian kernel has floating-point values), the kernel was approximated with powers of two so that the appropriate right shifts could be applied to the binary values instead of using combinational logic. While this is not a perfect Gaussian blur, it can sufficiently filter out high frequency noise, while remaining a single-cycle operation with a small hardware footprint.

The mean and variance calculation was more challenging to implement in hardware, as it needs to iterate through all of the pixel values within each superpixel. Traditionally, a two-pass method is used, where the first pass calculates the mean, and then the second pass uses the previously computed mean to determine the variance. This has a critical drawback in that either all of the pixels have to be transferred between the SW and HW subsystems twice (once for each pass) to stream the data through, or a substantial amount of memory is needed on the HW side to store an entire image worth of pixels. As an approximation, the modified Welford algorithm [318, 319], shown in Algorithm 3, can compute mean and variance in a single pass. However, this comes at the cost of introducing some error as it is only an approximation. With

uniformly random pixel values between 0 and 255, empirical experiments showed that the Modified Welford algorithm has an error of approximately 20% from the true mean and 5% from the true variance for each superpixel. This could be problematic since the errors are relatively large, except that the variance on the errors is very small, meaning that they are usually in the same direction and have a similar magnitude (i.e. consistently wrong). Therefore, for the purposes of classification of the superpixel, as long as the errors are relatively consistent, then the overall classification accuracy may not be affected too strongly.

Algorithm 3 Modified Welford algorithm for calculating mean and variance in a single pass

```
1: n = 0, mean = 0, var = 0
2: for pixel in superpixel do
3:     n += 1
4:     delta1 = pixel - mean
5:     mean += delta1 / n
6:     delta2 = pixel - mean
7:     var += delta1 * delta2
8: end for
9: var /= n-1
```

This method does require the use of division, which normally requires multiple clock cycles and is a relatively computationally expensive operation in comparison to addition or multiplication operations. To simplify the division operator, a multiply-shift approach was used with a Look Up Table (LUT) for all possible division values, since it is known *a priori* that the operation is limited to integer values between 1 and 255. Empirically, it was found that for the image resolutions in the CDW2014 dataset, the largest superpixel contained 165 pixels, so a safe upper bound of 255 was set for the denominator, making it easy to store a finite number of multiply-shift parameters. Using the LUT, any division operation can be approximated by multiplying the numbers together and then shifting right, which is the same as dividing by a power of two. While this method does introduce some error since it is an integer approximation, it can be completed in a single pass of all the pixels in a superpixel and is much faster than a standard division operation.

To summarise the communication requirements between the FPGA and HPS shown in Figure 8.5, for each image being processed, the image data input stream receives one value per pixel (the greyscale intensity of that pixel), the superpixel labels register receives one value per pixel (representing the superpixel number for that pixel), and the mean and variance outputs register returns two numbers per superpixel (a mean and a variance). Since the pixels are provided in image order (from the top-left to the bottom right in rows), the superpixel labels are generally not in order (since the superpixels span across multiple rows of the image). The mean and variance registers hold values for all of the superpixels and are iteratively updated as pixels are fed into the HPS, and wait for the entire image to be processed before sending the computed values back to the HPS. In order to make full use of the communication buses between the FPGA fabric and HPS, the full-size 128-bit bridge is used. It would be wasteful to use this wide bridge to send single integers, so data packing [320] is used to concatenate as

Table 8.3: Hardware Acceleration Results

Version	**PWC (%)**	**Speed (FPS)**
Original Reference, HPS-only	1.49	4.18
Grayscale + Integer, HPS-only	1.74	18.31
Gaussian Blurring on FPGA	1.55	18.56
Mean/Variance on FPGA	3.22	29.16
Gaussian Blurring + Mean/Variance on FPGA	2.88	37.91

much data together before transmission in order to minimise the number of transactions and therefore the communication overheads. The FPGA was clocked at 50MHz during testing, with interconnect logic clocked at 150MHz, although the system could potentially be run at a higher clock frequency depending on the device, which would further improve the computation speed.

8.4.2 Hardware Evaluation

As shown in Table 8.3, running SuperBE in software alone (using the HPS only) is much slower than on a Laptop PC (Table 8.1). This is predominantly caused by the fact that the embedded processor is much slower, running at 800MHz with a simpler architecture in comparison to the 2.4GHz+ CPU on a laptop or desktop. The HPS-only version is used as the reference embedded benchmark against which hardware accelerated versions should be compared. Firstly, parallelising the Gaussian blur operator has a negligible effect on speed, as the speed gain through parallelisation in hardware barely covers the added communication overheads. It is theoretically possible to increase the throughput of the Gaussian blur component further by instantiating multiple copies of the component and dividing the image into blocks for processing in parallel [321], but this further increases the hardware cost and is likely to still be constrained by the communication bandwidth between the HPS and FPGA.

The hardware versions of the mean and variance operations add speed to the system, although this is at the cost of also introducing some error, which should be expected since there are multiple approximations in that calculation. This is an approximate computing trade-off, where there could be a higher level of accuracy, but this would require more hardware resources and likely reduce the speed. The final HW/SW Co-Design version with both Gaussian blurring and the mean/variance calculation accelerated on FPGA in one contiguous block yields a speed of 37.91 fps (normalised for 320x240 resolution images), a 2x increase from the HPS-only Grayscale + Integer version. This comes at the cost of approximately 1% extra error introduced into the system, which is likely to be acceptable for most purposes. More importantly, this can be compared to the final version to the original colour SuperBE algorithm being run on the HPS to find the overall improvement from both the software and hardware optimisations on the same test platform. The overall 9x speed improvement more than justifies the 1.4% higher

Table 8.4: Hardware Resource Consumption on a Cyclone V 5CSEMA5F31C6 FPGA device

Logic Utilization (in ALMs)	10,316 / 32,070 (32%)
Block Memory Bits	750,840 / 4,065,280 (18%)
DSP Blocks	7 / 87 (8%)
Clock Frequency	50MHz

Table 8.5: Average Computation Times on Selected Sequences from CDW2014

Image Resolution and Sequence	ms	fps
320 x 240 on *backdoor*	15.3	65.2
352 x 240 on *peopleInShade*	16.9	59.3
360 x 240 on *bungalows*	17.3	58.0
380 x 244 on *copyMachine*	18.5	54.0
432 x 288 on *fountain02*	30.8	32.5
540 x 360 on *skating*	74.6	13.4
645 x 315 on *turbulence2*	75.7	13.2
720 x 480 on *cubicle*	69.0	14.5
720 x 576 on *PETS2006*	89.9	11.1
720 x 480 on *blizzard*	132.5	7.5
720 x 540 on *wetsnow*	149.1	6.7
320 x 240 on *turnpike*	57.0	17.6
480 x 295 on *tramStation*	97.7	10.2
595 x 245 on *streetCornerAtNight*	100.6	9.9
640 x 350 on *tramCrossroad*	166.2	6.0
700 x 450 on *fluidHighway*	217.4	4.6

error. Future designers can choose different positions in the accuracy-speed trade-off, with higher accuracy possible if a slower speed is acceptable.

The hardware resource consumption for the final HW/SW Co-Design system is shown in Table 8.4. While there is still some space for further hardware components, the bottleneck in communication between the HW and SW systems needs to be addressed first before adding further hardware parallelism.

In all previous results in this Chapter, the computation times have been normalised for a 320x240 image size. In the first part of Table 8.5, it is shown how that time varies as the image resolution becomes larger, reaching 720p. As the image becomes larger, the computation time will increase, which is due to the fact that there are more pixels to process when calculating the mean and variance of each superpixel. As the modified Welford algorithm allows this to be done in one pass, the increase in computation time is linear, or in other words, this part of the algorithm is O(n) complex. Since this algorithm is superpixel-based, the classification and model

update steps do not increase based on the image resolution, since the number of superpixels remains relatively similar. However, the resolution is not the only factor that influences the computation time; the more dominating factor is how much of the image is foreground, since SuperBE has to spend more time comparing superpixel values to past model values to confirm that the superpixel is foreground. This is reflected in the second part of Table 8.5, where the sequences with a grey background are from the *lowFramerate* and *nightVideos* categories, and the processing time per frame is considerably slower for the same image resolutions in the first part of the Table. This is the primary reason that the normalised 320x240 speed is so much lower than the computation time for the *backdoor* sequence even though it is also 320x240 resolution - the more computationally expensive sequences pull the average computation time up (and therefore push the fps down).

While this work focuses on SuperBE, it would be generally useful to compare the performance of this system against other hardware-accelerated background estimation systems. However, this is surprisingly difficult for three reasons. Firstly, different papers use different datasets to perform their experiments, which can have implications on variation between the ways that results are reported. For example, SuperBE requires more computation time to verify that superpixels are foreground than it does to classify superpixels as background, so it naturally takes longer on datasets with more moving objects in the video sequences. Secondly, different computation platforms can have significant effects on the speed of the system; as shown in Section 8.2.2, some may use GPUs, others use FPGAs, and every implementation can be clocked differently, which has to be taken into account when trying to compare the effectiveness and quality of the underlying engineering work. Lastly, while many background estimation algorithm papers report accuracy, many do not report computation time at all. Papers focusing on hardware implementations of background estimation report fast computation times, but not accuracy. Some works, such as [312], only report accuracy in comparison to the accuracy in "simulation" i.e. a desktop CPU, not based on the ground truth. In some cases, accuracy evaluation is only through visual inspection, for example in [311]. The lack of accuracy results is likely because actually running the system against a large test dataset and evaluating the accuracy can be quite challenging for embedded implementations and require long test times. This makes it impossible to determine where those works sit on the accuracy-speed trade-off, and do not tell the full story when impressive speed or accuracy results are reported alone.

8.5 Conclusions

This Chapter presents the acceleration of a background estimation algorithm, SuperBE, in both the software and hardware worlds, through a systematic approach towards improving speed while maintaining acceptable levels of accuracy. In software, the main optimisations focused on reducing the amount of data to be processed by converting the algorithm into greyscale and integer-only versions, yielding a 4.38x speed improvement over the original algorithm (without post-processing) at the cost of a 0.6% higher error rate. In hardware, the main optimisations focused on accelerating the Gaussian blur and mean/variance calculation

steps, parallelising these steps and adding more specialised computation units. This resulted in a further 2x speed improvement within the embedded implementation. When combined, there is a 9x speed improvement over the original SuperBE algorithm when executed on an embedded processor. This work shows that Hardware/Software Co-Design is a valid approach for improving the performance of algorithms, without needing to invest significant resources to develop a pure hardware design. This work also provides evidence that SuperBE can be accelerated sufficiently to be used in embedded real-time processing contexts, especially where background estimation is used as a first step in an image processing pipeline to reduce the workload of subsequent algorithms. The case study in this Chapter demonstrates the potential of using Hardware/Software Co-Design techniques to accelerate video analytics algorithms without significantly compromising accuracy levels. Some general principles are shown:

- Intelligently partition algorithms between the software and hardware subsystems using execution profiling
- Reduce the amount of information that needs to be processed through techniques like quantisation and reducing the number of dimensions used to describe the data
- Use integer or fixed-point mathematics to keep the hardware system simple
- Reduce communications overheads with data packing and shared memory
- Understand the accuracy vs. speed trade-offs that are acceptable for the target application

When computer vision algorithms are first placed on embedded computers, they may seem hopelessly slow in comparison to standard desktop computers or servers with powerful computational resources. The extreme option is to spend a lot of time and effort implementing the algorithm in pure hardware. Hardware/Software Co-Design offers a compromise - significant speedups can be achieved by intelligently accelerating only the parts of the algorithm that will have the largest impact. This is an underutilised solution, perhaps because the level of knowledge required to combine both software and hardware worlds is perceived to be too much of a barrier for most programmers and developers. There is a lot more research to be done in this space, particularly in the areas of high-level synthesis and in the development of open-source IP blocks, before these practices can become mainstream. In the meantime, this case study aims to demonstrate that applying these techniques and principles to even a small part of an algorithm can have large impacts on the computation time, mitigating the fear of spending large amounts of developer resources on pursuing small gains in speed.

In future work, there is opportunity for improvements to be made to both speed and accuracy if needed. Further reducing the usage of the HPS-FPGA bridges would decrease the communication time, which could be done by directly loading images onto the FPGA and completing preliminary processing there, and then only sending the mean and variance values for each superpixel back across to the CPU for model comparison and updating. This may be challenging, as the first frame still needs to be provided to the CPU for model initialisation, as it would be very difficult to implement superpixel segmentation in hardware. Additionally, the value of doing so would be limited since it is only executed once, during initialisation. There is also potential for further parallelisation - the SOC CPU has more than one core, and multiple

copies of the hardware components could be made to allow independent superpixels to be processed simultaneously if more hardware resourcing was avxailable on a larger device. Line buffers could also be used to allow for multiple pixels in the same window to be processed in parallel, although the hardware is not the bottleneck in the system at this stage. This partitioning process could also be further automated [322], depending on the library of hardware units available for parametrisation and automatic instantiation on the FPGA. Lastly, accuracy could be improved by further investigating the effect of different data widths in the hardware components; increasing the bit widths of the mean/variance component would likely make the results more accurate, but would also consume more hardware resources. Investigating more suitable post-processing schemes that clean up the output background masks, particularly in hardware, would also improve accuracy but introduce additional computational complexity. Judging whether the additional effort and cost required is worth the accuracy gain is a design and management decision, dependent on the end application. In the context of the person tracking system in this thesis, now that SuperBE has been accelerated using HW/SW Co-Design techniques, the person detection module is the next obvious target for acceleration since it consumes the most computation time on the smart camera portion of the pipeline. While the techniques in this Chapter have only been applied to SuperBE, the positive results provide some promise that these techniques could be applied to the rest of the pipeline in order to achieve good speeds on embedded platforms.

Chapter 9

Conclusions and Future Work

9.1 Summary of Contributions

A lot of research into computer vision and video analytics focuses myopically on accuracy, losing sight of the real-world issues that developers face when implementing these systems. The research presented in this thesis explored the engineering development of a video analytics system for person tracking that is practical, in the sense that it is able to run in real-time on embedded computers with constrained resources, and is ethical, in the sense that the system is designed, by default, to help protect the privacy of people being observed. The combination of both practicality and ethics is what makes this system more suitable for real world use. Achieving this required a multi-disciplinary approach that brought together methods and ideas from a wide variety of fields, including computer vision, artificial intelligence and machine learning, embedded system design, optimisation, computer networking, statistics, social sciences, and philosophy.

The work presented in this thesis started with a motivating scenario and short literature review of person tracking systems, before introducing a proposed modular pipeline for a novel person tracking system. Making this entire pipeline a reality involved the development of each module, and forming new ways of solving each task when existing solutions were insufficient. A new background estimation algorithm based on superpixels called SuperBE was developed, with better speed and comparable accuracy to other state-of-the-art approaches. Several person detection algorithms were discussed and compared, leading to the selection of the Deformable Parts Model (DPM) and Aggregate Channel Features (ACF) methods for use in this system. Feature selection was also discussed, along with dimensionality reduction and metric learning as methods of helping make extracted feature vectors easier to classify. A new dataset called UoA-Indoor was introduced, which contained individuals walking around an office environment, captured by four overlapping cameras. Appearance-based re-identification was discussed for classifying the identities of people across multiple camera views. This included the finding that the use of a covariance metric transformation was more generalisable than more sophisticated metric learning algorithms that tend to overfit to the training data. A number of re-identification approaches were compared, particularly focusing on one-shot

learning approaches, resulting in the novel formulation of Sequential k-Means for one-shot and unsupervised classification. A spatio-temporal model was developed to further aid the classification of detected individuals into identity classes, using Kalman Filters to predict the future positions of people. The appearance and spatio-temporal models were then combined using a model fusion approach that incorporated linear weighting with a decay function and rule-based system, leading to demonstrably higher accuracy rates with the fused models.

This concluded the main development of the algorithms used in the person tracking system from a software perspective. Further practical and ethical considerations were then taken into account for the implementation of the system. Firstly, this involved identifying the types of privacy risks that can exist in a person tracking system, and a survey was conducted to better understand public perceptions of privacy in different camera-based surveillance contexts. This led to the development of the privacy-affirming architecture, which uses computer vision technology to help protect the privacy of individuals being observed by cameras. This architecture took advantage of the increasing popularity of smart cameras, which allow for some processing to be conducted at the point of image capture. The architecture was implemented, with the use of single board computers to help emulate an embedded smart camera, and the impacts of using the much slower computing hardware were presented and discussed. Since the implemented system was significantly lower in speed and insufficient for real-time purposes, further methods of acceleration were explored. Hardware/Software Co-design was applied to SuperBE to demonstrate how hardware acceleration using FPGAs alongside software optimisations can help improve the speed of the system within an embedded context, without significantly affecting the accuracy.

However, even with all these efforts, the system developed still falls short of what is necessary for real-world implementation. While 60-70% accuracy may be considered to be quite good for person re-identification and tracking in an academic context, real-world customers have different expectations. They tend to assume that computers always work, and therefore demand accuracy rates closer to 95%[31], which are yet to be achieved by any academic or commercial systems in uncontrolled environments. The speed of the developed system also needs to be improved, as the use of distributed smart cameras significantly constrains the available computing resources. There are therefore many opportunities for future work.

9.2 Future Work

The bulk of the algorithm selection and development has focused on the accuracy-speed trade-off that is critical for practical implementation. Keeping in mind that the eventual target was to deploy parts of the algorithm on embedded hardware, improving speed by reducing computational complexity was prioritised. However, the timing analysis of the different modules in Chapter 5, combined with the final separation of responsibilities between the smart cameras and the central co-ordinating server, revealed that this could be rebalanced. The person detection algorithm remains the bottleneck in the system, and since it is performed on the smart

[31] Everett Berry, Perceive Inc, Interview on 12 December 2018

camera, more effort is needed to reduce the computation cost involved. With an understanding of the types of errors that are more acceptable or less acceptable (i.e. false negatives are more acceptable than false positives), much simpler techniques such as blob detection [19] and intelligent object segmentation [65] could be used to reduce the computation time while maintaining acceptable accuracy rates. Alternatively, a suitable deep learning candidate could be the recently released MobileNets [104], which are streamlined deep convolutional neural networks that have been shown to be very fast while achieving relatively good accuracy rates. Utilising Hardware/Software Co-Design techniques to further accelerate person detection could also be helpful. At the same time, the appearance-based re-identification and spatio-temporal modules were also designed to be computationally efficient, even though ultimately they were executed on the central server where the computational constraints are much less of a concern. Since these two modules have the largest impact on the final accuracy, it therefore makes sense that more complex methods could be explored here as long as the computation time remains acceptable. For example, transfer learning approaches can be combined with neural networks or component analysis to improve re-identification [153], while methods such as particle filters [203], local binary tracklets [323], and kernalised correlation filters [38] could be explored for the spatio-temporal module. Using more specific parts for feature extraction [142] or using soft biometric descriptors [190, 324, 325] may also help improve the accuracy of re-identification.

There is also potential for further improvements to be made to the other modules as well. SuperBE could be further improved by better understanding alternative feature choices that may provide better accuracy or may be simpler to calculate in comparison to the colour means and covariance matrices [326]. Recent advances in the formation of superpixels such as SNIC [63] could also be introduced instead of SLICO. Calibration and mapping are important parts of installing a multi-camera video analytics system in a real-world environment and would currently be very time-consuming and costly to perform for each new environment. There is the potential to better automate this process, using overlapping camera views to computationally determine the topology of the cameras by finding common feature points, and then forming a homography that can serve as a ground plane map. This could have the additional advantage of improving the accuracy of person localisation after person detection. The model fusion algorithm is relatively limited in the current implementation, as it is difficult to obtain proper probabilities for statistical inference; more effort towards quantifying the errors on the measurements would be helpful for obtaining a more statistically robust result [327]. Alternatively, a loss function could be implemented to help learn the best way to fuse the spatio-temporal position and appearance features [328], although this would need to be imbued with some notion of time. Improvements could also be made to the data analytics module at the last stage of the system. In the system presented in this thesis, the visualisations were restricted to maps showing the positions of individual people, and a heatmap of the areas where individuals have spent the most time. End users may want different types of visualisations that can help create alternative insights, such as saliency maps, time-series trend graphs, and analyses of inter-person interactions. This has to be done in a way that avoids information overload for the end user as well as excessive computation time, but would be

necessary for the development of a commercial product. Lastly, there is still a lot of potential work to be done in better understanding privacy, and how the relationships between the observed and system owners change when the contexts change. Technological advancements in the digital age have meant that researchers in surveillance studies has shifted the focus from governments to corporations, and new forms of rights may need to be developed to provide appropriate protection for consumers. Developing frameworks that help both system designers and regulators understand the impacts of video analytics systems are necessary for the ethical and responsible use of this new technology.

Overall, this thesis has shown how different technologies and methods can be brought together to work towards video analytics systems for person tracking that aim to be both practical and ethical. It is important for engineers and developers to not lose sight of the bigger picture, focusing too narrowly in one area without considering the impacts and effects more broadly. Computer vision technologies will rapidly advance in the coming years, and real-world video analytics systems are already emerging. Holistic system design is required for the development of functional systems that meet the requirements of system owners while minimising the harm accrued to those being observed. It is a difficult balance, defined by the fact that the parties have differing and potentially contradictory incentives and needs. The aim must be to find solutions that can be positive-sum; solutions where all parties can benefit, to bring this technological change through in a way that is net positive for our communities and societies.

"We need the best and the brightest thinkers, strategists, coders, surveillance experts, tech geeks, and disruptors to utilise all of the tools we have available to us to build the world that we want to see." - Alicia Garza, Civil Rights Activist

Human Participants Ethics

The work presented in this thesis includes research that involved human participants. Two approvals were given by the University of Auckland Human Participants Ethics Committee:

- 018182: "Embedded Vision-based Person Detection, Tracking, and Re-identification" This allowed us to form the UoA-Indoor dataset, which was used for experimental validation of various algorithms and the overall video analytics pipeline described in Chapters 1-5. Participants were recruited from the University of Auckland Newmarket Campus, and the dataset is securely held in local storage at the University of Auckland. Footage has not been publicly released, beyond screenshots being included in publications and this thesis.
- 020597: "Understanding Public Perceptions of Surveillance Camera Systems" This allowed us to conduct a survey into privacy perceptions and preferences with the general public, which was described in Chapter 6. Participants were recruited largely from social media networks in New Zealand, and all participants were anonymous. Data has mostly been reported in aggregate form, with the exception of a small number of quotes from the qualitative answers used in presentations and publications.

Any concerns about the ethics of including human participants in this research, or how the research was conducted, should be directed to the author in the first instance, the Head of Department (Electrical, Computer, and Software Engineering) in the second instance, and/or the University of Auckland Human Participants Ethics Committee.

Copyright Notices

The following copyright notices cover material that has been adapted from published papers, according to the standards required by the publishers:

- ©2016 Springer Nature. Reprinted by permission from Springer Nature: Chen A.TY., Biglari-Abhari M., Wang K.IK., Bouzerdoum A., Tivive F.H.C. Hardware/Software Co-design for a Gender Recognition Embedded System. In: Fujita H., Ali M., Selamat A., Sasaki J., Kurematsu M. (eds) Trends in Applied Knowledge-Based Systems and Data Science. (2016) *Lecture Notes in Computer Science*, vol 9799. Springer, Cham.
- ©2017 IEEE. Reprinted, with permission, from A. T.-Y. Chen, M. Biglari-Abhari, and K. I.-K. Wang, "Trusting the Computer in Computer Vision: A Privacy-Affirming Framework", *Conference on Computer Vision and Pattern Recognition (CVPR), Workshop on The Bright and Dark Sides of Computer Vision: Challenges and Opportunities for Privacy and Security (CV-COPS)*, pp 56-63, 2017.
- ©2017 IEEE. Reprinted, with permission, from A. T.-Y. Chen, J. Fan, M. Biglari-Abhari, and K. I.-K. Wang, "A Computationally Efficient Pipeline for Camera-based Indoor Person Tracking", *IEEE Image and Vision Computing New Zealand (IVCNZ)*, 2017.
- ©2018 Springer Nature. Reprinted by permission from Springer Nature: Chen A.TY., Biglari-Abhari M., Wang K.IK., Bouzerdoum A., Tivive F.H.C. Convolutional Neural Network Acceleration with Hardware/Software Co-design. *Appl Intell* (2018) 48: 1288.
- ©2018 Springer Nature. Reprinted by permission from Springer Nature: Chen, A.TY., Biglari-Abhari, M., Wang, K.IK. SuperBE: Computationally-Light Background Estimation with Superpixels. *J Real-Time Image Proc* (2018).
- ©2018 IEEE. Reprinted, with permission, from A. T.-Y. Chen, M. Biglari-Abhari, and K. I.-K. Wang, "Fast One-Shot Learning for Identity Classification in Person Re-identification", *IEEE International Conference on Control, Automation, Robotics, and Vision (ICARCV)*, 2018.
- ©2018 IEEE. Reprinted, with permission, from A. T.-Y. Chen, M. Biglari-Abhari, and K. I.-K. Wang, "Context is King: Privacy Perceptions of Camera-based Surveillance", *IEEE International Conference on Advanced Video and Signal-based Surveillance (AVSS)*, 2018.
- ©2018 The Authors. Distributed under a Creative Commons Attribution License (CC BY 4.0): Chen, A.T.-Y.; Gupta, R.; Borzenko, A.; Wang, K.I.-K.; Biglari-Abhari, M. Accelerating SuperBE with Hardware/Software Co-Design, *J. Imaging*, 2018.

In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of The University of Auckland's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please visit the IEEE website to learn how to obtain a License from RightsLink. If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

Bibliography

[1] D. Grady, "The vision thing: Mainly in the brain," *Discover*, vol. 14, p. 56, 6 1993.

[2] L. D. Rosenblum, *See What I'm Saying: The Extraordinary Powers of Our Five Senses*. W. W. Norton & Company, 2010.

[3] M. McNeal, "Fei-Fei Li: If we want machines to think, we need to teach them to see," *Wired*, 2015.

[4] C. W. Park, E. S. Iyer, and D. C. Smith, "The effects of situational factors on in-store grocery shopping behavior: The role of store environment and time available for shopping," *Journal of Consumer Research*, vol. 15, no. 4, pp. 422–433, 1989.

[5] D. R. Lichtenstein, N. M. Ridgway, and R. G. Netemeyer, "Price perceptions and consumer shopping behavior: A field study," *Journal of Marketing Research*, vol. 30, no. 2, pp. 234–245, 1993.

[6] L. W. Turley and R. E. Milliman, "Atmospheric effects on shopping behavior: A review of the experimental evidence," *Journal of Business Research*, vol. 49, no. 2, pp. 193–211, 2000.

[7] P. Chandon, J. W. Hutchinson, E. T. Bradlow, and S. H. Young, "Does in-store marketing work? Effects of the number and position of shelf facings on brand attention and evaluation at the point of purchase," *Journal of Marketing*, vol. 73, no. 6, pp. 1–17, 2009.

[8] G. Deak, K. Curran, and J. Condell, "A survey of active and passive indoor localisation systems," *Computer Communications*, vol. 35, no. 16, pp. 1939–1954, 2012.

[9] A. C. Nazare, F. de O. Costa, and W. R. Schwartz, "Content-based multi-camera video alignment using accelerometer data," in *International Conference on Advanced Video and Signal-based Surveillance (AVSS)*, 2018.

[10] X. Wang, "Intelligent multi-camera video surveillance: A review," *Pattern Recognition Letters*, vol. 34, no. 1, pp. 3–19, 2013.

[11] M. Andriluka, S. Roth, and B. Schiele, "People-tracking-by-detection and people-detection-by-tracking," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008, pp. 1–8.

[12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, pp. 2278–2324, 11 1998.

[13] G. Cristóbal, L. Perrinet, and M. S. Keil, *Biologically Inspired Computer Vision: Fundamentals and Applications*. Wiley, 2015.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2012, pp. 1097–1105.

[15] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 487–495.

[16] L. Wu, Y. Wang, Z. Ge, Q. Hue, and X. Li, "Structured deep hashing with convolutional neural networks for fast person re-identification," *Computer Vision and Image Understanding (CVIU)*, vol. 167, pp. 63–73, 2018.

[17] C. Farabet, B. Martini, P. Akselrod, S. Talay, Y. LeCun, and E. Culurciello, "Hardware accelerated convolutional neural networks for synthetic vision systems," in *International Symposium on Circuits and Systems (ISCAS)*, 2010, pp. 257–260.

[18] C. Alippi, S. Disabato, and M. Roveri, "Moving convolutional neural networks to embedded systems: The Alexnet and VGG-16 case," in *International Conference on Information Processing in Sensor Networks (IPSN)*, 2018, pp. 212–223.

[19] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer, "Multi-camera multi-person tracking for EasyLiving," in *International Workshop on Visual Surveillance*, 2000, pp. 3–10.

[20] W. Lu and Y.-P. Tan, "A color histogram based people tracking system," in *International Symposium on Circuits and Systems (ISCAS)*, vol. 2, 2001, pp. 137–140.

[21] H. Tsutsui, J. Miura, and Y. Shirai, "Optical flow-based person tracking by multiple cameras," in *International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2001, pp. 91–96.

[22] A. Mittal and L. S. Davis, "M2Tracker: A multi-view approach to segmenting and tracking people in a cluttered scene," *International Journal of Computer Vision (IJCV)*, vol. 51, no. 3, pp. 189–203, 2003.

[23] R. T. Collins, A. J. Lipton, H. Fujiyoshi, and T. Kanade, "Algorithms for cooperative multisensor surveillance," *Proceedings of the IEEE*, vol. 89, no. 10, pp. 1456–1477, 2001.

[24] M. Bogaert, N. Chelq, P. Cornez, C. S. Regazzoni, A. Teschioni, and M. Thonnat, "The PASSWORDS project," in *International Conference on Image Processing (ICIP)*, vol. 3, 1996, pp. 675–678.

[25] T. Matsuyama, "Cooperative distributed vision," in *DARPA Image Understanding Workshop*, 1998, pp. 365–384.

[26] M. Valera and S. A. Velastin, "Intelligent distributed surveillance systems: A review," *Vision, Image, and Signal Processing*, vol. 152, 2 2005.

[27] X. Wang, G. Doretto, T. Sebastian, J. Rittscher, and P. Tu, "Shape and appearance context modeling," in *International Conference on Computer Vision (ICCV)*, 2007, pp. 1–8.

[28] G. A. Jones, J. R. Renno, and P. Remagnino, "Auto-calibration in multiple-camera surveillance environments," in *International Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*, 2002, pp. 40–47.

[29] W.-T. Chen, P.-Y. Chen, W.-S. Lee, and C.-F. Huang, "Design and implementation of a real time video surveillance system with wireless sensor networks," in *Vehicular Technology Conference (VTC)*, 2008, pp. 218–222.

[30] H. Nguyen, B. Bhanu, A. Patel, and R. Diaz, "VideoWeb: Design of a wireless camera network for real-time monitoring of activities," in *International Conference on Distributed Smart Cameras (ICDSC)*, 2009, pp. 1–8.

[31] V. A. Petrushin, G. Wei, O. Shakil, D. Roqueiro, and V. Gershman, "Multiple-sensor indoor surveillance system," in *Canadian Conference on Computer and Robot Vision (CRV)*, 2006, pp. 40–47.

[32] U. Park, A. K. Jain, I. Kitahara, K. Kogure, and N. Hagita, "ViSE: Visual search engine using multiple networked cameras," in *International Conference on Pattern Recognition (ICPR)*, 2006, pp. 1204–1207.

[33] A. Narang, S. P. Joglekar, K. B. Dhanapal, and A. A. Somasundara, "A novel way of tracking people in an indoor area," in *International Conference on Advanced Computing, Networking and Security*, 2012, pp. 85–94.

[34] S. Yang, A. Wiliem, and B. C. Lovell, "It takes two to tango: Cascading off-the-shelf face detectors," in *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 535–543.

[35] T. Zhang, A. Chowdhery, P. Bahl, K. Jamieson, and S. Banerjee, "The design and implementation of a wireless video surveillance system," in *Annual International Conference on Mobile Computing and Networking*, 2015, pp. 426–438.

[36] S. C. Chan, S. Zhang, J.-F. Wu, H.-J. Tan, J. Q. Ni, and Y. S. Hung, "On the hardware/software design and implementation of a high definition multiview video surveillance system," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 3, no. 2, pp. 248–262, 2013.

[37] J. B. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2015, pp. 3995–4001.

[38] O. Camps, M. Gou, T. Hebble, S. Karanam, O. Lehmann, Y. Li, R. J. Radke, Z. Wu, and F. Xiong, "From the lab to the real world: Re-identification in an airport camera network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, pp. 540–553, 3 2017.

[39] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *European Conference on Computer Vision (ECCV) Workshop on Benchmarking Multi-Target Tracking (BMTT)*, 2016, pp. 17–35.

[40] L. Zheng, Z. Bie, Y. Sun, J. Wang, C. Su, S. Wang, and Q. Tian, "MARS: A video benchmark for large-scale person re-identification," in *European Conference on Computer Vision (ECCV)*, 2016, pp. 868–884.

[41] S. Karanam, M. Gou, Z. Wu, A. Rates-Borras, O. Camps, and R. J. Radke, "A systematic evaluation and benchmark for person re-identification: Features, metrics, and datasets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[42] H. Wang, X. Bao, R. R. Choudhury, and S. Nelakuditi, "InSight: Recognizing humans without face recognition," in *Workshop on Mobile Computing Systems and Applications*, 2013, pp. 7.1–7.6.

[43] D. Figueira, M. Taiana, A. Nambiar, J. Nascimento, and A. Bernardino, "The HDA+ data set for research on fully automated re-identification systems," in *European Conference on Computer Vision Workshop (ECCV)*, 2014, pp. 241–255.

[44] A. Schick, M. Bäuml, and R. Stiefelhagen, "Improving foreground segmentations with probabilistic superpixel markov random fields," in *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2012, pp. 27–31.

[45] S. K. Devi, N. Malmurugan, and S. Poornima, "Improving the efficiency of background subtraction using superpixel extraction and midpoint for centroid," *International Journal of Computer Applications*, vol. 43, no. 10, pp. 1–5, 2012.

[46] J. Lim and B. Han, "Generalized background subtraction using superpixels with label integrated motion estimation," in *European Conference on Computer Vision (ECCV)*, 2014, pp. 173–187.

[47] S. Javed, S. H. Oh, A. Sobral, T. Bouwmans, and S. K. Jung, "Background subtraction via superpixel-based online matrix decomposition with structured foreground constraints," in *International Conference on Computer Vision Workshop (ICCVW)*, 2015, pp. 930–938.

[48] D. Giordano, F. Murabito, S. Palazzo, and C. Spampinato, "Superpixel-based video object segmentation using perceptual organization and location prior," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4814–4822.

[49] S. Brutzer, B. Höferlin, and G. Heidemann, "Evaluation of background subtraction techniques for video surveillance," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 1937–1944.

[50] D. Tsai and S. Lai, "Independent component analysis-based background subtraction for indoor surveillance," *IEEE Transactions on Image Processing*, vol. 18, no. 1, pp. 158–167, 2009.

[51] M. C. Liem and D. M. Gavrila, "Joint multi-person detection and tracking from overlapping cameras," *Computer Vision and Image Understanding (CVIU)*, vol. 128, pp. 36–50, 2014.

[52] M. Piccardi, "Background subtraction techniques: A review," in *International Conference on Systems, Man, and Cybernetics (SMC)*, 2004, pp. 3099–3104.

[53] Y. Ivanov, A. Bobick, and J. Liu, "Fast lighting independent background subtraction," in *Workshop on Visual Surveillance*, 1998.

[54] R. M. Luque, E. Domínguez, E. J. Palomo, and J. Muñoz, "A neural network approach for video object segmentation in traffic surveillance," in *International Conference on Image Analysis and Recognition (ICIAR)*, 2008, pp. 151–158.

[55] F. El Baf, T. Bouwmans, and B. Vachon, "Type-2 fuzzy mixture of gaussians model: Application to background modeling," in *International Symposium on Visual Computing (ISVC)*, 2008, pp. 772–781.

[56] M. Babaee, D. T. Dinh, and G. Rigoll, "A deep convolutional neural network for video sequence background subtraction," *Pattern Recognition*, vol. 76, pp. 635–649, 2018.

[57] M. C. Liem and D. M. Gavrila, "A comparative study on multi-person tracking using overlapping cameras," in *International Conference on Computer Vision Systems (ICVS)*, 2013, pp. 203–212.

[58] T. Bouwmans, "Traditional and recent approaches in background modeling for foreground detection: An overview," *Computer Science Review*, vol. 11, no. 12, pp. 31–66, 2014.

[59] A. Sobral and A. Vacavant, "A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos," *Computer Vision and Image Understanding (CVIU)*, vol. 122, pp. 4–21, 2014.

[60] O. Barnich and M. van Droogenbroeck, "ViBe: A universal background subtraction algorithm for video sequences," *IEEE Transactions on Image Processing*, vol. 20, no. 6, pp. 1709–1724, 2011.

[61] M. Hofmann, P. Tiefenbacher, and G. Rigoll, "Background segmentation with feedback: The pixel-based adaptive segmenter," in *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2012, pp. 38–43.

[62] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[63] R. Achanta and S. Süsstrunk, "Superpixels and polygons using simple non-iterative clustering," *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4895–4904, 2017.

[64] X. Gu, J. D. Deng, and M. K. Purvis, "Improving superpixel-based image segmentation by incorporating color covariance matrix manifolds," in *International Conference on Image Processing (ICIP)*, 2014, pp. 4403–4406.

[65] A. Papazoglou and V. Ferrari, "Fast object segmentation in unconstrained video," in *International Conference on Computer Vision (ICCV)*, 2013, pp. 1777–1784.

[66] G. Shu, A. Dehghan, and M. Shah, "Improving an object detector and extracting regions using superpixels," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 3721–3727.

[67] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg, "Video segmentation by tracking many figure-ground segments," in *International Conference on Computer Vision (ICCV)*, 2013, pp. 2192–2199.

[68] B. Shoushtarian and H. E. Bez, "A practical adaptive approach for dynamic background subtraction using an invariant colour model and object tracking," *Pattern Recognition Letters*, vol. 26, no. 1, pp. 5–26, 2005.

[69] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. S. Davis, "Real-time foreground-background segmentation using codebook model," *Real-Time Imaging*, vol. 11, no. 3, pp. 172–185, 2005.

[70] H. Wang and D. Suter, "Background subtraction based on a robust consensus method," in *International Conference on Pattern Recognition (ICPR)*, 2006, pp. 223–226.

[71] S. Bak, E. Corvée, F. Brémond, and M. Thonnat, "Person re-identification using spatial covariance regions of human body parts," in *International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2010, pp. 435–440.

[72] W. Förstner and B. Moonen, "A metric for covariance matrices," *Geodesy-The Challenge of the 3rd Millennium*, pp. 299–309, 2003.

[73] A. Cherian, S. Sra, A. Banerjee, and N. Papanikolopoulos, "Jensen-Bregman LogDet divergence with application to efficient similarity search for covariance matrices," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 9, pp. 2161–2174, 2013.

[74] Y. Wang, P. Jodoin, F. M. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "CDnet 2014: An expanded change detection benchmark dataset," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 393–400.

[75] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[76] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *International Conference on Pattern Recognition (ICPR)*, vol. 2, 2004, pp. 28–31.

[77] R. Wang, F. Bunyak, G. Seetharaman, and K. Palaniappan, "Static and moving object detection using flux tensor with split gaussian models," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 420–424.

[78] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "SuBSENSE: A universal change detection method with local adaptive sensitivity," *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 359–373, 2015.

[79] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "A self-adjusting approach to change detection based on background word consensus," in *Winter Conference on Applications of Computer Vision (WACV)*, 2015, pp. 990–997.

[80] C. Spampinato, S. Palazzo, and I. Kavasidis, "A texton-based kernel density estimation approach for background modeling under extreme conditions," *Computer Vision and Image Understanding (CVIU)*, vol. 122, pp. 74–83, 2014.

[81] B. Xin, Y. Tian, Y. Wang, and W. Gao, "Background subtraction via generalized fused lasso foreground modeling," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4676–4684.

[82] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 743–761, 2012.

[83] M. Isard and J. MacCormick, "BraMBLe: A bayesian multiple-blob tracker," in *International Conference on Computer Vision (ICCV)*, vol. 2, 2001, pp. 34–41.

[84] M. Everingham and A. Zisserman, "Automated person identification in video," in *International Conference on Image and Video Retrieval (CIVR)*, 2006, pp. 289–298.

[85] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[86] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2005, pp. 886–893.

[87] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 8, pp. 1532–1545, 2014.

[88] H. Cho, P. E. Rybski, A. Bar-Hillel, and W. Zhang, "Real-time pedestrian detection with deformable part models," in *Intelligent Vehicles Symposium (IVS)*, 2012, pp. 1035–1042.

[89] J. Yan, Z. Lei, L. Wen, and S. Z. Li, "The fastest deformable part model for object detection," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 2497–2504.

[90] B. Benfold and I. Reid, "Stable multi-target tracking in real-time surveillance video," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 3457–3464.

[91] P. Dollár, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," in *British Machine Vision Conference (BMVC)*, 2009, pp. 91.1–91.11.

[92] P. Dollár, S. Belongie, and P. Perona, "The fastest pedestrian detector in the west," in *British Machine Vision Conference (BMVC)*, 2010, pp. 68.1–68.11.

[93] P. Dollár, R. Appel, and W. Kienzle, "Crosstalk cascades for frame-rate pedestrian detection," in *European Conference on Computer Vision (ECCV)*, 2012, pp. 645–659.

[94] F. De Smedt and T. Goedemé, "Open framework for combined pedestrian detection," in *International Conference on Computer Vision Theory and Applications (VISIGRAPP)*, 2015, pp. 551–558.

[95] R. Benenson, M. Mathias, R. Timofte, and L. V. Gool, "Pedestrian detection at 100 frames per second," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 2903–2910.

[96] F. D. Smedt, K. V. Beeck, T. Tuytelaars, and T. Goedemé, "Pedestrian detection at warp speed: Exceeding 500 detections per second," in *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2013, pp. 622–628.

[97] S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele, "How far are we from solving pedstrian detection?" In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1259–1267.

[98] C. Wojek, S. Walk, and B. Schiele, "Multi-cue onboard pedestrian detection," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 794–801.

[99] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[100] R. Girshick, "Fast R-CNN," in *International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448.

[101] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[102] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, real-time object detection," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[103] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *European Conference on Computer Vision (ECCV)*, 2016, pp. 21–37.

[104] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, vol. abs/1704.04861, 2017.

[105] M. Noman, M. H. Yousaf, and S. A. Velastin, "An optimized and fast scheme for real-time human detection using Raspberry Pi," in *Digital Image Computing: Techniques and Applications (DICTA)*, 2016.

[106] D. Gray and H. Tao, "Viewpoint invariant pedestrian recognition with an ensemble of localized features," in *European Conference on Computer Vision (ECCV)*, 2008, pp. 262–275.

[107] M. Hirzer, C. Beleznai, P. M. Roth, and H. Bischof, "Person re-identification by descriptive and discriminative classification," in *Scandinavian Conference on Image Analysis (SCIA)*, 2011, pp. 91–102.

[108] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah, "Part-based multiple-person tracking with partial occlusion handling," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 1815–1821.

[109] C. Liu, S. Gong, C. C. Loy, and X. Lin, "Person re-identification: What features are important?" In *European Conference on Computer Vision Workshops (ECCVW)*, 2012, pp. 391–401.

[110] M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani, "Person re-identification by symmetry-driven accumulation of local features," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 2360–2367.

[111] L. Bazzani, M. Cristani, and V. Murino, "Symmetry-driven accumulation of local features for human characterization and re-identification," *Computer Vision and Image Understanding*, vol. 117, no. 2, pp. 130–144, 2013.

[112] S. Pedagadi, J. Orwell, S. Velastin, and B. Boghossian, "Local fisher discriminant analysis for pedestrian reidentification," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 3318–3325.

[113] R. Zhao, W. Ouyang, and X. Wang, "Unsupervised salience learning for person re-identification," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 3586–3593.

[114] R. Zhao, W. Ouyang, and X. Wang, "Learning mid-level filters for person re-identification," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 144–151.

[115] B. Ma, Y. Su, and F. Jurie, "Covariance descriptor based on bio-inspired features for person re-identification and face verification," *Image and Vision Computing*, vol. 32, no. 6-7, pp. 379–390, 2014.

[116] G. Serra, C. Grana, M. Manfredi, and R. Cucchiara, "Covariance of covariance features for image classification," in *International Conference on Multimedia Retrieval (ICMR)*, 2014, pp. 411–414.

[117] F. Xiong, M. Gou, O. Camps, and M. Sznaier, "Person re-identification using kernel-based metric learning methods," in *European Conference on Computer Vision (ECCV)*, 2014, pp. 1–16.

[118] S. Liao, Y. Hu, X. Zhu, and S. Z. Li, "Person re-identification by local maximal occurrence representation and metric learning," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 2197–2206.

[119] T. Matsukawa, T. Okabe, E. Suzuki, and Y. Sato, "Hierarchical gaussian descriptor for person re-identification," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1363–1372.

[120] A. Daniel Costea and S. Nedevschi, "Semantic channels for fast pedestrian detection," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2360–2368.

[121] M. Gou, S. Karanam, W. Liu, O. Camps, and R. J. Radke, "DukeMTMC4ReID: A large-scale multi-camera person re-identification dataset," in *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 10–19.

[122] I. Kviatkovsky, A. Adam, and E. Rivlin, "Color invariants for person reidentification," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 35, no. 7, pp. 1622–1634, 2013.

[123] L. Liang, C. Liu, Y.-Q. Xu, B. Guo, and H.-Y. Shum, "Real-time texture synthesis by patch-based sampling," *ACM Transactions on Graphics (TOG)*, vol. 20, no. 3, pp. 127–150, 2001.

[124] M. Hirzer, P. M. Roth, M. Köstinger, and H. Bischof, "Relaxed pairwise learned metric for person re-identification," in *European Conference on Computer Vision (ECCV)*, 2012, pp. 780–793.

[125] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *International Conference on Computer Vision (ICCV)*, 2015, pp. 1116–1124.

[126] Y. Lin, L. Zheng, Z. Zheng, Y. Wu, and Y. Yang, "Improving person re-identification by attribute and identity learning," *CoRR*, vol. abs/1703.07220, 2017.

[127] E. Ahmed, M. Jones, and T. K. Marks, "An improved deep learning architecture for person re-identification," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3908–3916.

[128] W. Li and Q. Du, "Gabor-filtering-based nearest regularized subspace for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 4, pp. 1012–1022, 2014.

[129] T. Xiao, S. Li, B. Wang, L. Lin, and X. Wang, "Joint detection and identification feature learning for person search," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3415–3424.

[130] I. Filković, Z. Kalafatić, and T. Hrkać, "Deep metric learning for person re-identification and de-identification," in *International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2016, pp. 1360–1364.

[131] F. M. Khan and F. Brémond, "Person re-identification for real-world surveillance systems," *CoRR*, vol. abs/1607.05975, 2016.

[132] S. Paisitkriangkrai, C. Shen, and A. van den Hengel, "Learning to rank in person re-identification with metric ensembles," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1846–1855.

[133] D. Gray, S. Brennan, and H. Tao, "Evaluating appearance models for recognition, reacquisition, and tracking," in *International Workshop on Performance Evaluation for Tracking and Surveillance (PETS)*, 2007.

[134] B. Prosser, W.-S. Zheng, S. Gong, and T. Xiang, "Person re-identification by support vector ranking," in *British Machine Vision Conference (BMVC)*, 2010, pp. 21.1–21.11.

[135] D. Baltieri, R. Vezzani, and R. Cucchiara, "3DPeS: 3D people dataset for surveillance and forensics," in *Joint ACM Workshop on Human Gesture and Behavior Understanding*, 2011, pp. 59–64.

[136] J. Berclaz, F. Fleuret, E. Türetken, and P. Fua, "Multiple object tracking using k-shortest paths optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1806–1819, 2011.

[137] V. Takala and M. Pietikäinen, "CMV100: A dataset for people tracking and re-identification in sparse camera networks," in *International Conference on Pattern Recognition (ICPR)*, 2012, pp. 1387–1390.

[138] W. Li, R. Zhao, T. Xiao, and X. Wang, "Deepreid: Deep filter pairing neural network for person re-identification," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 152–159.

[139] L. Zheng, H. Zhang, S. Sun, M. Chandraker, Y. Yang, and Q. Tian, "Person re-identification in the wild," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1367–1376.

[140] A. Bedagkar-Gala and S. K. Shah, "A survey of approaches and trends in person re-identification," *Image and Vision Computing*, vol. 32, no. 4, pp. 270–286, 2014.

[141] A. O. Balan, L. Sigal, and M. J. Black, "A quantitative evaluation of video-based 3D person tracking," in *International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (PETS)*, 2005, pp. 349–356.

[142] E. Insafutdinov, M. Andriluka, L. Pishchulin, S. Tang, E. Levinkov, B. Andres, and B. Schiele, "Arttrack: Articulated multi-person tracking in the wild," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6457–6465.

[143] D. Doermann and D. Mihalcik, "Tools and techniques for video performance evaluation," in *International Conference on Pattern Recognition (ICPR)*, vol. 4, 2000, pp. 167–170.

[144] C. Vondrick, D. Patterson, and D. Ramanan, "Efficiently scaling up crowdsourced video annotation," *International Journal of Computer Vision*, vol. 101, no. 1, pp. 184–204, 2013.

[145] J. Yuen, B. Russell, C. Liu, and A. Torralba, "Labelme video: Building a video database with human annotations," in *International Conference on Computer Vision (ICCV)*, 2009, pp. 1451–1458.

[146] R. Zhao, W. Ouyang, and X. Wang, "Person re-identification by salience matching," in *International Conference on Computer Vision (ICCV)*, 2013, pp. 2528–2535.

[147] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Conference on Neural Information Processing Systems (NIPS)*, 2015, pp. 2017–2025.

[148] D. Li, X. Chen, Z. Zhang, and K. Huang, "Learning deep context-aware features over body and latent parts for person re-identification," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7398–7407.

[149] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell, "Distance metric learning, with application to clustering with side-information," in *International Conference on Neural Information Processing Systems (NIPS)*, 2002, pp. 521–528.

[150] P. M. Roth, M. Hirzer, M. Köstinger, C. Beleznai, and H. Bischof, "Mahalanobis distance learning for person re-identification," in *Person Re-Identification*. Springer London, 2014, pp. 247–267.

[151] J. Zheng and B. Lu, "A support vector machine classifier with automatic confidence," *Neurocomputing*, vol. 74, no. 11, pp. 1926–1935, 2011.

[152] A. Bellet, A. Habrard, and M. Sebban, "A survey on metric learning for feature vectors and structured data," *CoRR*, vol. abs/1306.6709, 2013.

[153] T. Avraham, I. Gurvich, M. Lindenbaum, and S. Markovitch, "Learning implicit transfer for person re-identification," in *European Conference on Computer Vision Workshops (ECCVW)*, 2012, pp. 381–390.

[154] O. Javed, K. Shafique, and M. Shah, "Appearance modeling for tracking in multiple non-overlapping cameras," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2005, pp. 26–33.

[155] J. Hu, J. Lu, Y. P. Tan, and J. Zhou, "Deep transfer metric learning," *IEEE Transactions on Image Processing (TIP)*, vol. 25, no. 12, pp. 5576–5588, 2016.

[156] K. Q. Weinberger, J. Blitzer, and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," in *Advances in Neural Information Processing Systems (NIPS)*, 2006, pp. 1473–1480.

[157] M. Hirzer, P. M. Roth, and H. Bischof, "Person re-identification by efficient impostor-based metric learning," in *International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, 2012, pp. 203–208.

[158] Y. Wang, R. Hu, C. Liang, C. Zhang, and Q. Leng, "Person re-identification by smooth metric learning," in *Pacific-Rim Conference on Advances in Multimedia Information Processing*, 2013, pp. 561–573.

[159] B. Yao, Z. Zhao, and K. Liu, "Metric learning with trace-norm regularization for person re-identification," in *International Conference on Image Processing (ICIP)*, 2014, pp. 2442–2446.

[160] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *International Conference on Machine Learning (ICML)*, 2007, pp. 209–216.

[161] X. Xu, W. Li, and D. Xu, "Distance metric learning using privileged information for face verification and person re-identification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 12, pp. 3150–3162, 2015.

[162] G.-J. Qi, J. Tang, Z.-J. Zha, T.-S. Chua, and H.-J. Zhang, "An efficient sparse metric learning in high-dimensional space via l1-penalized log-determinant regularization," in *International Conference on Machine Learning (ICML)*, 2009, pp. 841–848.

[163] E. Y. Liu, Z. Guo, X. Zhang, V. Jojic, and W. Wang, "Metric learning from relative comparisons by minimizing squared residual," in *International Conference on Data Mining (ICDM)*, 2012, pp. 978–983.

[164] E. Amid, A. Gionis, and A. Ukkonen, "Semi-supervised kernel metric learning using relative comparisons," *CoRR*, vol. abs/1612.00086, 2016.

[165] A. Thyagharajan and A. Routray, "An ensemble metric learning scheme for face recognition," in *International Conference on Multimedia and Expo (ICME)*, 2017, pp. 115–120.

[166] M. Sugiyama, "Local fisher discriminant analysis for supervised dimensionality reduction," in *International Conference on Machine Learning (ICML)*, 2006, pp. 905–912.

[167] A. Bar-hillel, T. Hertz, N. Shental, and D. Weinshall, "Learning distance functions using equivalence relations," in *International Conference on Machine Learning (ICML)*, 2003, pp. 11–18.

[168] J. B. Tenenbaum and W. T. Freeman, "Separating style and content with bilinear models," *Neural Computation*, vol. 12, no. 6, pp. 1247–1283, 2000.

[169] N. Shental, T. Hertz, D. Weinshall, and M. Pavel, "Adjustment learning and relevant component analysis," in *European Conference on Computer Vision (ECCV)*, 2002, pp. 776–790.

[170] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighbourhood components analysis," in *Advances in Neural Information Processing Systems (NIPS)*, 2004, pp. 513–520.

[171] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *Journal of Machine Learning Research (JMLR)*, vol. 10, pp. 207–244, 2009.

[172] M. Köstinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof, "Large scale metric learning from equivalence constraints," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 2288–2295.

[173] P. C. Mahalanobis, "On the generalized distance in statistics," *Proceedings of the National Institute of Sciences (India)*, vol. 2, pp. 49–55, 1936.

[174] J. Yao and J.-M. Odobez, "Fast human detection from joint appearance and foreground feature subset covariances," *Computer Vision and Image Understanding*, vol. 115, no. 10, pp. 1414–1426, 2011.

[175] R. Benenson, M. Omran, J. Hosang, and B. Schiele, "Ten years of pedestrian detection, what have we learned?" In *European Conference on Computer Vision Workshop (ECCVW)*, 2014, pp. 613–627.

[176] Z. Li, S. Chang, F. Liang, T. S. Huang, L. Cao, and J. R. Smith, "Learning locally-adaptive decision functions for person verification," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 3610–3617.

[177] A. Mignon and F. Jurie, "PCCA: A new approach for distance learning from sparse pairwise constraints," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 2666–2672.

[178] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 29, no. 1, pp. 40–51, 2007.

[179] M. Faraki, M. T. Harandi, and F. Porikli, "No fuss metric learning, a hilbert space scenario," *Pattern Recognition Letters*, vol. 98, pp. 83–89, 2017.

[180] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4004–4012.

[181] D. Kedem, S. Tyree, F. Sha, G. R. Lanckriet, and K. Q. Weinberger, "Non-linear metric learning," in *Advances in Neural Information Processing Systems (NIPS)*, 2012, pp. 2573–2581.

[182] W. S. Zheng, S. Gong, and T. Xiang, "Towards open-world person re-identification by one-shot group-based verification," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 38, no. 3, pp. 591–606, 2016.

[183] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 28, no. 4, pp. 594–611, 2006.

[184] C. Nakajima, M. Pontil, B. Heisele, and T. Poggio, "Full-body person recognition system," *Pattern Recognition*, vol. 36, no. 9, pp. 1997–2006, 2003.

[185] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[186] T. Xiao, H. Li, W. Ouyang, and X. Wang, "Learning deep feature representations with domain guided dropout for person re-identification," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[187] S. Bak and P. Carr, "One-shot metric learning for person re-identification," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2990–2999.

[188] M. Munaro, A. Fossati, A. Basso, E. Menegatti, and L. Van Gool, "One-shot person re-identification with a consumer depth camera," in *Person Re-Identification*. Springer London, 2014, pp. 161–181.

[189] Y. Liang and Y. Zhou, "Multi-camera tracking exploiting person re-id technique," in *International Conference on Neural Information Processing (ICONIP)*, 2017, pp. 397–404.

[190] K. W. Lee, N. Sankaran, S. Setlur, N. Napp, and V. Govindaraju, "Wardrobe model for long term re-identification and appearance prediction," in *International Conference on Advanced Video and Signal-based Surveillance (AVSS)*, 2018.

[191] S. Kumar, T. K. Marks, and M. Jones, "Improving person tracking using an inexpensive thermal infrared sensor," in *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2014, pp. 217–224.

[192] J.-H. Huh and K. Seo, "An indoor location-based control system using bluetooth beacons for IoT systems," *Sensors (MDPI)*, vol. 17, no. 12, p. 2917, 2017.

[193] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Computing Surveys*, vol. 38, no. 4, 2006.

[194] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 2411–2418.

[195] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[196] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, *An Invitation to 3-D Vision: From Images to Geometric Models*. SpringerVerlag, 2003.

[197] G. Bishop and G. Welch, "An introduction to the Kalman filter," *Special Interest Group on Computer Graphics and Interactive Techniques (SIGGRAPH)*, vol. 8, pp. 1–46, 2001.

[198] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1442–1468, 2014.

[199] W. Yan, C. Weber, and S. Wermter, "A hybrid probabilistic neural model for person tracking based on a ceiling-mounted camera," *Journal of Ambient Intelligence and Smart Environments*, vol. 3, no. 3, pp. 237–252, 2011.

[200] J. García, A. Gardel, I. Bravo, J. L. Lázaro, M. Martínez, and D. Rodríguez, "Directional people counter based on head tracking," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 9, pp. 3991–4000, 2013.

[201] B. Li, L. Heng, K. Koser, and M. Pollefeys, "A multiple-camera system calibration toolbox using a feature descriptor-based calibration pattern," in *International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 1301–1307.

[202] F. Gustafsson, "Particle filter theory and practice with positioning applications," *IEEE Aerospace and Electronic Systems Magazine*, vol. 25, pp. 53–82, 2010.

[203] J.-W. Choi, D. Moon, and J.-H. Yoo, "Robust multi-person tracking for real-time intelligent video surveillance," *ETRI Journal (Electronics and Telecommunications Research Institute)*, vol. 37, no. 3, pp. 551–561, 2015.

[204] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *International Joint Conference on Artificial Intelligence*, vol. 2, 1981, pp. 674–679.

[205] E. W. Kamen and J. K. Su, *Introduction to Optimal Estimation*. Springer London, 1999.

[206] P. Zarchan and H. Musoff, *Fundamentals of Kalman Filtering: A Practical Approach*. American Institute of Aeronautics and Astronautics, 2000.

[207] R. Faragher, "Understanding the basis of the kalman filter via a simple and intuitive derivation," *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 128–132, 2012.

[208] C. H. Kuo and R. Nevatia, "How does person identity recognition help multi-person tracking?" In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 1217–1224.

[209] S. Tang, M. Andriluka, B. Andres, and B. Schiele, "Multiple people tracking by lifted multicut and person re-identification," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3539–3548.

[210] L. Beyer, S. Breuers, V. Kurin, and B. Leibe, "Towards a principled integration of multi-camera re-identification and tracking through optimal bayes filters," in *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 29–38.

[211] C. Liu, C. Change Loy, S. Gong, and G. Wang, "POP: Person re-identification post-rank optimisation," in *International Conference on Computer Vision (ICCV)*, 2013, pp. 441–448.

[212] Q. Leng, R. Hu, C. Liang, Y. Wang, and J. Chen, "Person re-identification with content and context re-ranking," *Multimedia Tools and Applications*, vol. 74, no. 17, pp. 6989–7014, 2015.

[213] Z. Zhong, L. Zheng, D. Cao, and S. Li, "Re-ranking person re-identification with k-reciprocal encoding," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3652–3661.

[214] S. Lin, "Rank aggregation methods," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, pp. 555–570, 5 2010.

[215] E. Pacuit, "Voting methods," in *The Stanford Encyclopedia of Philosophy*, E. N. Zalta, Ed., Stanford University, 2017.

[216] W. V. Gehrlein, "Condorcet's paradox," *Theory and Decision*, vol. 15, no. 2, pp. 161–197, 1983.

[217] R. A. Hummel and M. S. Landy, "A statistical viewpoint on the theory of evidence," *IEEE Transactions on Patern Analysis and Machine Intelligence*, vol. 10, no. 2, pp. 235–247, 1988.

[218] S. Challa and D. Koks, "Bayesian and Dempster-Shafer fusion," *Sādhanā*, vol. 29, no. 2, pp. 145–174, 2004.

[219] T. W. Anderson, *An Introduction to Multivariate Statistical Analysis*. Wiley New York, 1958.

[220] H. J. van der Helm and E. A. Hische, "Application of Bayes's theorem to results of quantitative clinical chemical determinations," *Clinical Chemistry*, vol. 25, pp. 985–988, 6 1979.

[221] V. Nguyen, T. D. Ngo, K. M. T. T. Nguyen, D. A. Duong, K. Nguyen, and D. Le, "Re-ranking for person re-identification," in *International Conference on Soft Computing and Pattern Recognition (SoCPaR)*, 2013, pp. 304–308.

[222] J. H. Moor, "Towards a theory of privacy in the information age," *ACM Computers and Society*, vol. 27, no. 3, pp. 27–32, 1997.

[223] E. Barendt, Ed., *Privacy*. Routledge, 2001.

[224] K. D. Haggerty and R. V. Ericson, *The New Politics of Surveillance and Visibility*. University of Toronto Press, 2006.

[225] M. Andrejevic, "Surveillance in the big data era," in *Emerging Pervasive Information and Communication Technologies (PICT): Ethical Challenges, Opportunities and Safeguards*, K. D. Pimple, Ed. Springer Netherlands, 2014, pp. 55–69.

[226] M. Saini, P. K. Atrey, S. Mehrotra, and M. Kankanhalli, "Privacy aware publication of surveillance video," *International Journal of Trust Management in Computing and Communications*, vol. 1, no. 1, pp. 23–51, 2012.

[227] M. Andrejevic, *Infoglut: How Too Much Information Is Changing the Way We Think and Know*. Routledge, 2013.

[228] W. A. Parent, "Privacy, morality, and the law," *Philosophy & Public Affairs*, vol. 12, no. 4, pp. 269–288, 1983.

[229] J. DeCew, "Privacy," in *The Stanford Encyclopedia of Philosophy*, E. N. Zalta, Ed., Metaphysics Research Lab, Stanford University, 2015.

[230] P. M. Regan, *Legislating Privacy: Technology, Social Values, and Public Policy*. University of North Carolina Press, 1995.

[231] L. Lee, J. Lee, S. Egelman, and D. Wagner, "Information disclosure concerns in the age of wearable computing," in *NDSS Workshop on Usable Security (USEC)*, 2016.

[232] A. Senior, S. Pankanti, A. Hampapur, L. Brown, Y.-L. Tian, A. Ekin, J. Connell, C. F. Shu, and M. Lu, "Enabling video privacy through computer vision," *IEEE Security & Privacy*, vol. 3, pp. 50–57, 2005.

[233] F. Roesner, D. Molnar, A. Moshchuk, T. Kohno, and H. J. Wang, "World-driven access control for continuous sensing," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2014, pp. 1169–1181.

[234] E. M. Newton, L. Sweeney, and B. Malin, "Preserving privacy by de-identifying face images," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 2, pp. 232–243, 2005.

[235] F. Dufaux and T. Ebrahimi, "Scrambling for privacy protection in video surveillance systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, pp. 1168–1174, 8 2008.

[236] S. Jana, A. Narayanan, and V. Shmatikov, "A scanner darkly: Protecting user privacy from perceptual applications," in *IEEE Symposium on Security & Privacy*, 2013, pp. 349–363.

[237] J. Schiff, M. Meingast, D. K. Mulligan, S. Sastry, and K. Goldberg, "Respectful cameras: Detecting visual markers in real-time to address privacy concerns," in *International Conference on Intelligent Robots and Systems (IROS)*, 2007, pp. 971–978.

[238] M. S. Barhm, N. Qwasmi, F. Z. Qureshi, and K. el-Khatib, "Negotiating privacy preferences in video surveillance systems," in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (IEA/AIE)*, 2011, pp. 511–521.

[239] M. Blazevic, K. Brkic, and T. Hrkac, "Towards reversible de-identification in video sequences using 3D avatars and steganography," in *Croatian Computer Vision Workshop (CCVW)*, 2015, pp. 9–14.

[240] H. A. Rashwan, A. Solanas, D. Puig, and A. Martínez-Ballesté, "Understanding trust in privacy-aware video surveillance systems," *International Journal of Information Security*, vol. 15, no. 3, pp. 225–234, 2016.

[241] S. Moncrieff, S. Venkatesh, and G. A. W. West, "Dynamic privacy in public surveillance," *Computer*, vol. 42, no. 9, pp. 22–28, 2009.

[242] M. Saini, P. K. Atrey, S. Mehrotra, and M. Kankanhalli, "W3-privacy: Understanding what, when, and where inference channels in multi-camera surveillance video," *Multimedia Tools and Applications*, vol. 68, no. 1, pp. 135–158, 2014.

[243] N. Raval, L. P. Cox, A. Srivastava, A. Machanavajjhala, and K. Lebeck, "MarkIt: Privacy markers for protecting visual secrets," in *International Conference on Ubiquitous Computing (UbiComp)*, 2014, pp. 1289–1295.

[244] T. Orekondy, B. Schiele, and M. Fritz, "Towards a visual privacy advisor: Understanding and predicting privacy risks in images," in *International Conference on Computer Vision (ICCV)*, 2017, pp. 3706–3715.

[245] C. Norris and G. Armstrong, *The Maximum Surveillance Society: The Rise of CCTV*. Berg, 1999.

[246] D. H. Nguyen, A. Kobsa, and G. R. Hayes, "An empirical investigation of concerns of everyday tracking and recording technologies," in *International Conference on Ubiquitous Computing (UbiComp)*, 2008, pp. 182–191.

[247] D. H. Nguyen, A. Bedford, A. G. Bretana, and G. R. Hayes, "Situating the concern for information privacy through an empirical study of responses to video recording," in *SIGCHI Conference on Human Factors in Computing Systems (CHI)*, 2011, pp. 3207–3216.

[248] H. J. Smith and S. J. Milberg, "Information privacy: Measuring individuals' concerns about organizational practices," *MIS Quarterly*, vol. 20, no. 2, pp. 167–196, 1996.

[249] M. Skirpan and T. Yeh, "Designing a moral compass for the future of computer vision using speculative analysis," in *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 64–73.

[250] W. Odom, J. Zimmerman, S. Davidoff, J. Forlizzi, A. K. Dey, and M. K. Lee, "A fieldwork of the future with user enactments," in *Designing Interactive Systems Conference*, 2012, pp. 338–347.

[251] R. Ramirez, M. Mukherjee, S. Vezzoli, and A. M. Kramer, "Scenarios as a scholarly methodology to produce interesting research," *Futures*, vol. 71, pp. 70–87, 2015.

[252] Data Futures Partnership (New Zealand), "A path to social license: Guidelines for trusted data use," 2017.

[253] H. Koskela, "Video surveillance, gender, and the safety of public urban space: "peeping tom" goes high tech?" *Urban Geography*, vol. 23, no. 3, pp. 257–278, 2002.

[254] B. von Silva-Tarouca Larsen, *Setting the Watch: Privacy and the Ethics of CCTV Surveillance*. Bloomsbury, 2011.

[255] R. Beckwith, "Designing for ubiquity: The perception of privacy," *IEEE Pervasive Computing*, vol. 2, no. 2, pp. 40–46, 2003.

[256] J. Mills, *Privacy: The Lost Right*. Oxford University Press, 2008.

[257] Office of the Privacy Commissioner (New Zealand), "Privacy and CCTV: A guide to the Privacy Act for businesses, agencies, and organisations," 2009.

[258] P. A. Norberg, D. R. Horne, and D. A. Horne, "The privacy paradox: Personal information disclosure intentions versus behaviors," *The Journal of Consumer Affairs*, vol. 41, no. 1, pp. 100–126, 2007.

[259] M. Williams, J. R. C. Nurse, and S. Creese, "Privacy is the boring bit: User perceptions and behaviour in the internet-of-things," in *International Conference on Privacy, Security, and Trust (PST)*, 2017, pp. 181–189.

[260] S. Barth and M. D. T. de Jong, "The privacy paradox – investigating discrepancies between expressed privacy concerns and actual online behavior – a systematic literature review," *Telematics and Informatics*, vol. 34, no. 7, pp. 1038–1058, 2017.

[261] S. Kokolakis, "Privacy attitudes and privacy behaviour: A review of current research on the privacy paradox phenomenon," *Computers & Security*, vol. 64, pp. 122–134, 2017.

[262] H. van Rossum, H. Gardeniers, J. Borking, A. Cavoukian, J. Brans, N. Muttupulle, and N. Magistrale, "Privacy-Enhancing Technologies: The Path to Anonymity," 1995.

[263] New Zealand Government, *Privacy Act 1993*, 2013.

[264] Q. M. Rajpoot and C. D. Jensen, "Video surveillance: Privacy issues and legal compliance," in *Promoting Social Change and Democracy through Information Technology*, V. Kumar and J. Svensson, Eds. IGI Global, 2015.

[265] S. Gürses, C. Troncoso, and C. Diaz, "Engineering privacy by design," Tech. Rep., 2011.

[266] B. A. Olshausen and D. J. Field, "Vision and the coding of natural images: The human brain may hold the secrets to the best image-compression algorithms," *American Scientist*, vol. 88, no. 3, pp. 238–245, 2000.

[267] J. J. Todd and R. Marois, "Capacity limit of visual short-term memory in human posterior parietal cortex," *Nature*, vol. 428, pp. 751–754, 2004.

[268] S. Jana, D. Molnar, A. Moshchuk, A. Dunn, B. Livshits, H. J. Wang, and E. Ofek, "Enabling fine-grained permissions for augmented reality applications with recognizers," in *USENIX Conference on Security (SEC)*, 2013, pp. 415–430.

[269] A. Hornberg, *Handbook of Machine Vision*. Wiley, 2006.

[270] H. Kavalionak, C. Gennaro, G. Amato, C. Vairo, C. Perciante, C. Meghini, and F. Falchi, "Distributed video surveillance using smart cameras," *Journal of Grid Computing*, vol. 17, no. 1, pp. 59–77, 2019.

[271] B. Tan, M. Biglari-Abhari, and Z. Salcic, "A system-level security approach for heterogeneous MPSoCs," in *Conference on Design and Architectures for Signal and Image Processing (DASIP)*, 2016, pp. 74–81.

[272] W. Hu, B. Amos, Z. Chen, K. Ha, W. Richter, P. Pillai, B. Gilbert, J. Harkes, and M. Satyanarayanan, "The case for offload shaping," in *International Workshop on Mobile Computing Systems and Applications (HotMobile)*, 2015, pp. 51–56.

[273] J. H. Ko, T. Na, M. Amir, and S. Mukhopadhyay, "Edge-host partitioning of deep neural networks with feature space encoding for resource-constrained internet-of-things platforms," in *International Conference on Advanced Video and Signal-based Surveillance (AVSS)*, 2018.

[274] P. Birnstill and A. Pretschner, "Enforcing privacy through usage-controlled video surveillance," *International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 318–323, 2013.

[275] Q. M. Rajpoot and C. D. Jensen, "Security and privacy in video surveillance: Requirements and challenges," in *IFIP International Information Security Conference*, 2014, pp. 169–184.

[276] F. E. Bignami, "Privacy and law enforcement in the European Union: The data retention directive," *Chicago Journal of International Law*, vol. 8, no. 1, pp. 233–256, 2007.

[277] H. Koch, T. Matzner, and J. Krumm, "Privacy enhancing of smart CCTV and its ethical and legal problems," *European Journal of Law and Technology*, vol. 4, no. 2, 2013.

[278] D. Brin, *The Transparent Society: Will Technology Force Us To Choose Between Privacy And Freedom?* Basic Books, 1999.

[279] T. Winkler and B. Rinner, "Security and privacy protection in visual sensor networks: A survey," *ACM Computing Surveys*, vol. 47, no. 1, 2:1–2:42, 2014.

[280] W. Wolf, B. Ozer, and T. Lv, "Smart cameras as embedded systems," *Computer*, vol. 35, no. 9, pp. 48–53, 2002.

[281] J. Sérot, F. Berry, and C. Bourrasset, "High-level dataflow programming for real-time image processing on smart cameras," *Journal of Real-Time Image Processing*, vol. 12, no. 4, pp. 635–647, 2016.

[282] P.-J. Lapray, B. Heyrman, and D. Ginhac, "HDR-ARtiSt: An adaptive real-time smart camera for high dynamic range imaging," *Journal of Real-Time Image Processing*, vol. 12, no. 4, pp. 747–762, 2016.

[283] Y. Shi and F. D. Real, "Smart cameras: Fundamentals and classification," in *Smart Cameras*, A. N. Belbachir, Ed. Springer US, 2010, pp. 19–34.

[284] A. Almagambetov, S. Velipasalar, and M. Casares, "Robust and computationally lightweight autonomous tracking of vehicle taillights and signal detection by embedded smart cameras," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3732–3741, 2015.

[285] M. Bramberger, A. Doblander, A. Maier, B. Rinner, and H. Schwabach, "Distributed embedded smart cameras for surveillance applications," *Computer*, vol. 39, no. 2, pp. 68–75, 2006.

[286] B. Rinner, T. Winkler, W. Schriebl, M. Quaritsch, and W. Wolf, "The evolution from single to pervasive smart cameras," in *International Conference on Distributed Smart Cameras (ICDSC)*, 2008, pp. 1–10.

[287] D. Eigenraam and L. J. M. Rothkrantz, "A smart surveillance system of distributed smart multi cameras modelled as agents," in *Smart Cities Symposium Prague (SCSP)*, 2016, pp. 1–6.

[288] M. Wolf, "Platforms and architectures for distributed smart cameras," in *Distributed Embedded Smart Cameras: Architectures, Design and Applications*, C. Bobda and S. Velipasalar, Eds. Springer New York, 2014, pp. 3–23.

[289] S. Mukhopodhyay, M. Wolf, M. F. Amir, E. Gebahrdt, J. H. Ko, J. H. Kung, and B. A. Musassar, "The CAMEL approach to stacked sensor smart cameras," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, 2018, pp. 1299–1303.

[290] J. Teich, "Hardware/software codesign: The past, the present, and predicting the future," *Proceedings of the IEEE*, vol. 100, pp. 1411–1430, 2012.

[291] T. Horprasert, D. Harwood, and L. S. Davis, "A statistical approach for real-time robust background subtraction and shadow detection," in *International Conference on Computer Vision (ICCV)*, 1999, pp. 1–19.

[292] C. Stauffer and E. Grimson, "Adaptive background mixture models for real-time tracking," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 1999, pp. 246–252.

[293] P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," *Video-based Surveillance Systems*, vol. 1, pp. 135–144, 2002.

[294] Y.-L. Tian, M. Lu, and A. Hampapur, "Robust and efficient foreground analysis for real-time video surveillance," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2005, pp. 1182–1187.

[295] S. Jiang and X. Lu, "WeSamBE: A weight-sample-based method for background subtraction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 9, pp. 2105–2115, 2017.

[296] Y. Wang, Z. Luo, and P.-M. Jodoin, "Interactive deep learning method for segmenting moving objects," *Pattern Recognition Letters*, vol. 96, pp. 66–75, 2017.

[297] L. A. Lim and H. Y. Keles, "Foreground segmentation using convolutional neural networks for multiscale feature encoding," *Pattern Recognition Letters*, vol. 112, pp. 256–262, 2018.

[298] S. Bianco, G. Ciocca, and R. Schettini, "Combination of video change detection algorithms by genetic programming," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 6, pp. 914–928, 2017.

[299] G. de Michell and R. K. Gupta, "Hardware/software co-design," *Proceedings of the IEEE*, vol. 85, pp. 349–365, 3 1997.

[300] N. Alt, C. Clause, and W. Stechele, "Hardware/software architecture of an algorithm for vision-based real-time vehicle detection in dark environments," in *Design, Automation, and Test in Europe (DATE)*, 2008, pp. 176–181.

[301] G. van der Wal, D. Zhang, I. Kandaswamy, J. Marakowitz, K. Kaighn, J. Zhang, and S. Chai, "FPGA acceleration for feature based processing applications," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 42–47.

[302] D. Tasson, A. Montagnini, R. Marzotto, and M. Farenzena, "FPGA-based pedestrian detection under strong distortions," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 65–70.

[303] A. T.-Y. Chen, M. Biglari-Abhari, K. I.-K. Wang, A. Bouzerdoum, and F. H. C. Tivive, "Convolutional neural network acceleration with hardware/software co-design," *Applied Intelligence*, vol. 48, no. 5, pp. 1288–1301, 2018.

[304] V. Pham, P. Vo, and V. T. Hung, "GPU implementation of extended gaussian mixture model for background subtraction," in *International Conference on Computing Communication Technologies, Research, Innovation, and Vision for the Future (RIVF)*, 2010, pp. 1–4.

[305] P. Carr, "GPU accelerated multimodal background subtraction," in *Digital Image Computing: Techniques and Applications (DICTA)*, 2008, pp. 279–286.

[306] M. Genovese and E. Napoli, "FPGA-based architecture for real time segmentation and denoising of HD video," *Journal of Real-Time Image Processing*, vol. 8, no. 4, pp. 389–401, 2013.

[307] T. Kryjak and M. Gorgon, "Real-time implementation of the ViBe foreground object segmentation algorithm," in *Federated Conference on Computer Science and Information Systems*, 2013, pp. 591–596.

[308] R. Rodriguez-Gomez, E. J. Fernandez-Sanchez, J. Diaz, and E. Ros, "Codebook hardware implementation on FPGA for background subtraction," *Journal of Real-Time Image Processing*, vol. 10, no. 1, pp. 43–57, 2015.

[309] C. Sánchez-Ferreira, J. Y. Mori, and C. H. Llanos, "Background subtraction algorithm for moving object detection in FPGA," in *Southern Conference on Programmable Logic*, 2012, pp. 1–6.

[310] H. Jiang, H. Ardo, and V. Owall, "Hardware accelerator design for video segmentation with multi-modal background modelling," in *International Symposium on Circuits and Systems (ISCAS)*, vol. 2, 2005, pp. 1142–1145.

[311] U. Ali and M. B. Malik, "Hardware/software co-design of a real-time kernel based tracking system," *Journal of Systems Architecture*, vol. 56, no. 8, pp. 317–326, 2010.

[312] H. Tabkhi, M. Sabbagh, and G. Schirner, "An efficient architecture solution for low-power real-time background subtraction," in *International Conference on Application-specific Systems, Architectures and Processors*, 2015, pp. 218–225.

[313] W. J. MacLean, "An evaluation of the suitability of FPGAs for embedded vision systems," in *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2005, pp. 131–137.

[314] S. Han, H. Mao, and W. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," in *International Conference on Learning Representations (ICLR)*, 2016.

[315] Y. Chen, W. Xu, R. Zhao, and X. Chen, "Design and evaluation of a hardware/software FPGA-based system for fast image processing," *Photonic Sensors*, vol. 4, no. 3, pp. 274–280, 2014.

[316] E. Gudis, P. Lu, D. Berends, K. Kaighn, G. van der Wal, G. Buchanan, S. Chai, and M. Piacentino, "An embedded vision services framework for heterogeneous accelerators," in *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2013, pp. 598–603.

[317] D. G. Bailey, *Design for Embedded Image Processing on FPGAs*. Wiley, 2011.

[318] R. F. Ling, "Comparison of several algorithms for computing sample means and variances," *Journal of the American Statistical Association*, vol. 69, no. 348, pp. 859–866, 1974.

[319] B. P. Welford, "Note on a method for calculating corrected sums of squares and products," *Technometrics*, vol. 4, no. 3, pp. 419–420, 1962.

[320] M. Sankaradas, V. Jakkula, S. Cadambi, S. Chakradhar, I. Durdanovic, E. Cosatto, and H. Graf, "A massively parallel coprocessor for convolutional neural networks," in *International Conference on Application-specific Systems, Architectures, and Processors (ASAP)*, 2009, pp. 53–60.

[321] B. A. Draper, J. R. Beveridge, A. P. W. Bohm, C. Ross, and M. Chawathe, "Accelerated image processing on FPGAs," *IEEE Transactions on Image Processing*, vol. 12, no. 12, pp. 1543–1551, 2003.

[322] I. Sahin and N. Saritekin, "A data path design tool for automatically mapping artificial neural networks on to FPGA-based systems," *Journal of Electrical Engineering and Technology*, vol. 11, no. 5, pp. 1466–1474, 2016.

[323] M. Ravanbakhsh, H. Mousavi, M. Nabi, L. Marcenaro, and C. Regazzoni, "Fast but not deep: Efficient crowd abnormality detection with local binary tracklets," in *International Conference on Advanced Video and Signal-based Surveillance (AVSS)*, 2018.

[324] D. Reid, S. Samangooei, C. Chen, M. Nixon, and A. Ross, "Soft biometrics for surveillance: An overview," in *Handbook of Statistics*, C. R. Rao and V. Govindaraju, Eds., vol. 31, 2013, pp. 327–352.

[325] P. Tome, J. Fierrez, R. Vera-Rodriguez, and M. S. Nixon, "Soft biometrics and their application in person recognition at a distance," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 3, pp. 464–475, 2014.

[326] Y.-Q. Chen, Z.-L. Sun, N. Wang, and X.-Y. Bao, "Background subtraction with superpixel and k-means," in *International Conference on Intelligent Computing (ICIC)*, 2018, pp. 108–112.

[327] N. Narayan, N. Sankaran, D. Arpit, K. Dantu, S. Setlur, and V. Govindaraju, "Person re-identification for improved multi-person multi-camera tracking by continuous entity association," in *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 566–572.

[328] C. Thomas and N. Balakrishnan, "Improvement in intrusion detection with advances in sensor fusion," *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 3, pp. 542–551, 2009.