On the Use of Deep Learning for Open World Person Re-Identification in Thermal Imagery

T.A. Robson, supervised by T.P. Breckon

Department of Computer Science, Durham University

***Abstract*—Thermal imagery provides significant advantages for round-the-clock surveillance, providing visibility of targets under all environmental conditions. However, a key limitation of thermal imagery is the lack of colour information available. Person re-identification across multiple cameras is a key research problem within the domain of visual surveillance and an important challenge for the future deployment of thermal sensing as an autonomous sensor technology. Many current approaches to the problem rely on colour features [8]. We have previously attempted to use a set of comparable features in thermal space to solve the re-identification problem with little success [19], and are now exploring alternatives.**

The aim of this research is to build a real time system that can detect a object within a thermal image, distinguish a person from other objects and track a person moving through a scene. This is done using a combination of a Mixture of Gaussians (MoG) background subtractor, a Histogram of Oriented Gradients (HOG) person detector and a Track-Learn-Detect (TLD) tracker. We then attempt to re-identify people using a siamese Convolutional Neural Network (CNN), which determines whether a pair of images contains the same person or two different people. Therefore, this research tackles the open world re-identification problem, rather than simply training and testing in a closed world on the same set of people, as many previous approaches have done [26]. This enables us to deploy our solution on any camera system and with any human targets.

Our CNN achieved 99% accuracy on our test dataset and our full re-identification system works well, achieving an accuracy of 96%. We conclude that it is possible to use deep learning as an effective re-identification system in thermal imagery and we show the strong performance of this on a varied dataset.

Keywords: Deep Learning, Siamese Network, Open World, Computer Vision, Person Re-Identification, Re-ID, Person Tracking, Track-Learn-Detect, Thermal Imagery, Thermal Video

I. Introduction

A fundamental task for a distributed multi-camera surveillance system is to associate people across different camera views at different locations and times [8]. This is referred to as the person re-identification problem [8] and is an interesting and important research area within the field of computer vision. From the previous work discussed in Section II, we can see that a substantial amount of research has been done on person re-identification, mainly revolving around the use of features or attributes of a person [11]. However, much of this operates in the visible spectrum, with attributes of the form "red shirt" [11]. In the modern world, thermal imagery is often used for round-the-clock surveillance when varying environmental conditions are present. It is important, therefore, that an effective re-identification system is developed to utilise the data available from thermal surveillance, as this problem has yet to be effectively solved.

There are many potential applications for this technology, but the most important in the modern world would be to support intelligence organisations. The surveillance data that a system like this can provide is critical for crime-prevention, forensic analysis, and counter-terrorism activities in both civilian and governmental agencies alike. While this surveillance data is widely used by human operators, these operators have to be trained, which offsets the utility of this approach with training and staffing costs. The implementation of an automated re-identification system is therefore of great interest, as it would be very useful in supporting these human operators and enabling them to achieve improved results more efficiently. Figure 1 shows five different views of the same person that our system is required to re-identify as being the same person.

Whilst thermal imagery has many advantages, it is not able to identify colour, making features that rely on colour useless. Therefore, alternative features are required to facilitate re-identification. In our previous work in [19], we attempted to use features that a human would think were distinctive enough to facilitate re-identification, but this achieved limited success. Since our work in [19], the attention of researchers in this area has shifted to deep learning based solutions, as is discussed in more detail in Section II. These deep learning approaches can be split into two categories: closed world, where the system is trained and tested on the same set of people from a pre-defined dataset; and open world, where the system is trained to learn what makes people different, so that it can be applied to any dataset of people.

To improve the performance and runtime of our new approach, we are replacing the Kalman filter used in [19] with an implementation of the Track-Learn-Detect (TLD) algorithm, originally proposed by the authors of [10]. The aim is to ensure that we do not have to pass every frame to the neural network, but instead, when a person has been identified, they can then be tracked for as long as they remain unobscured in the frame,

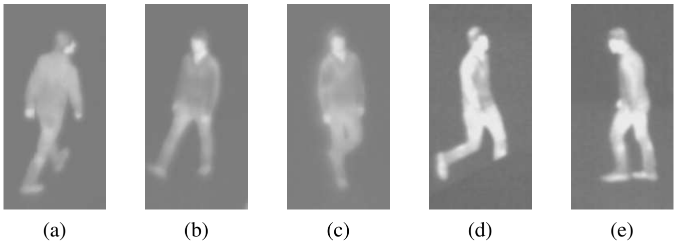


Figure 1: Sample thermal images of the same person

and continue to be labelled as the same person.

A. Aims of the Project

This project has three main aims: to extend and improve the real time person detection system from our previous work in [19] to include a TLD tracker; to develop, train and test a deep siamese CNN to determine whether a pair of images show the same person or different people; and to combine these into a fully functional re-identification system tested on both single cameras and a network of multiple cameras. This enables us to solve the open world thermal re-identification problem and set a new standard for the state of the art.

B. Achievements of the Project

We have achieved our aims. The person detection system functions far better than in our previous work due to the inclusion of TLD and we have achieved 99% accuracy on a siamese CNN. These components have been combined into a re-identification system and work well on our dataset, achieving an accuracy of 96%. We have set the state of the art for thermal re-identification, and shown it can be done using deep learning.

II. Related Work

We will now examine the previous work that has been done relating to the person re-identification problem, most of which concerns the visible spectrum.

A. Feature-Based Attempts at Re-Identification

Re-identification in colour is a well researched area, particularly using distinctive features for this purpose. Many of methods for utilising these features are discussed in [8], and have been widely researched and understood, so research has moved on to explore more complex methods. An important part of the current state of the art in this area is camera network layout and topology, as explored in [15]. In this work, the authors employ the technique of distance vector routing to understand the relative locations of the cameras and prioritise people seen most recently by the closest camera, as these are most likely to correlate with the new image. This understanding is achieved by first analysing the overlap between cameras, and then computing distance vectors and probabilities of a person going from one camera to another, thus reducing the time complexity of the re-identification process in most cases.

The work in [5] is on a similar theme to [15], but assumes a non-overlapping camera system. Each camera has entry and exit zones from its field of view and if a person can get from one camera field of view to another then the cameras are considered to be directly connected. The system can then create what is referred to as a camera link model, using a temporal, spatial and appearance relation between the entry and exit zones of the cameras. These paths are obtained from training data, but the system itself learns how to recognise people by attributes, and uses the training data to estimate where they are most likely to reappear after leaving a given camera field of view.

The authors of [23] propose a different method for feature-based identification, using a feature projection matrix to project image features of one camera to the feature space of another, to effectively eliminate the difference of feature distributions between the two cameras. This feature projection matrix is obtained through supervised learning. The method emplyed by the authors aims to use a simple gradient descent algorithm to accelerate and optimise the re-identification process by compensating for the inconsistency of feature distributions captured by different cameras.

The work in [24] emphasises the importance of making good use of all images and video frames captured of a person. The system proposed creates a gallery of images of known individuals, with more images increasing the accuracy of the system. When a gallery exists for a person, this is referred to as multi-shot re-identification, and single-shot re-identification when only one image is available in the gallery. For multi-shot re-identification, the authors propose to use geometric distance by collaboratively approximating the query sets using all available images in the gallery, a method known as collaborative sparse approximation.

Another approach, taken by the authors of [13], relies not only on extracting a set of features to compare people, but also on determining which of these features is the most discriminative for each person. This is achieved through the use of a random forest classifier and functions well in an open world setting. However, since the publication of this work, the state of the art in this area has moved on to deep learning.

Our own previous attempt at thermal re-identification in [19] attempted to use features that a human would consider useful to re-identify people, such as an approximation of the shape of the person, an analysis of their gait and a measure of where an individual's the thermal hotspots are located. This performed reasonably well for some features, but its inaccuracy in many situations led us to explore an alternative approach, using deep learning, based on the current state of the art in the area.

B. Use of Deep Learning for Re-Identification

The use of deep learning to facilitate re-identification has been the driving force in the research community in this area in recent times [16], [26], [25], [22], [2], [6], [18], [1]. As our ability to train deeper and deeper networks on larger and larger datasets has increased, largely thanks to modern improvements in graphics hardware, these approaches become increasingly relevant and powerful. As before, much of this research is focused on the colour spectrum. The work of [16] shows us how a combination of deep learning and human recognisable features can be used for re-identification. This works by training a neural network to recognise certain features that the authors consider to be discriminative. The result from the network is an ordered vector for the presence or absence of these features. However, as our previous work showed that we were unable to choose suitably discriminative features for thermal re-identification, we have chosen not to follow this approach.

Much of the rest of the previous deep learning work is split into open world and closed world problems. The work of [26] presents a useful comparison between these two, as shown in Figure 2. They refer to the closed world problem as Identification and the open world problem as Verification, with the justification that in closed world, the aim is to identify which of a given set of people the subject is, where as in open world, the aim is to verify whether two people are the same. These two approaches differ greatly in terms of input method, network architecture, feature extraction and the loss function used. The authors of [26] conclude, as we have, that the open world problem is more relevant to real world applications. Networks intended to solve the open world re-identification problem are referred to as siamese networks, meaning that the overall architecture contains two identical sister networks. Each of these sister networks takes one of the images being compared as its input, and outputs a feature vector describing it. These vectors are then compared and a decision is made as to whether the people in the two images are the same.

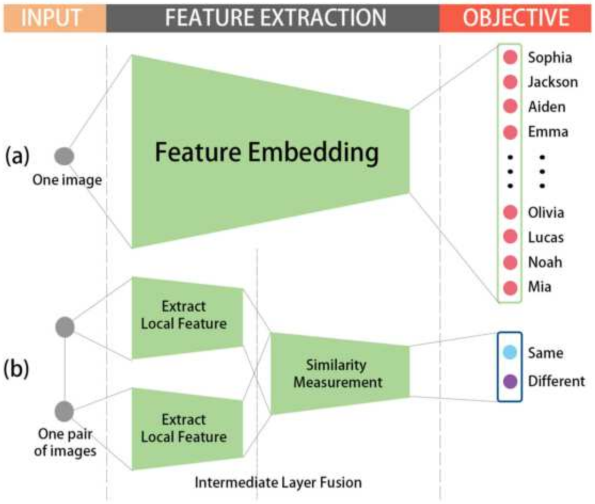


Figure 2: The difference between closed world (a) and open world (b) re-identification systems, from [26]

A solution to the open world problem is attempted in [25]. The authors aim to combine feature learning and re-identification into one framework, which is a deep siamese CNN with an Support Vector Machine (SVM) placed after the last layer of the CNN to attempt open world re-identification. The results of this are promising for the potential use of such an architecture, but fall short of the contemporary state of the art. The apparent reason for this is overfitting to the dataset used to train the architecture, and the authors hypothesise that if they could use a dataset with a wider variation of people, improved results could be achieved. As this work is several years old, we can hypothesise that with modern graphics hardware, this model could realistically be trained on a much larger and more varied dataset, which should improve the results. This hypothesis is reinforced by the authors of [1], who have implemented a similar architecture and trained on a larger dataset which consistently outperforms the contemporary state of the art.

The work in [22] proposes a similar deep siamese CNN architecture, but also retains the single image representation. The purpose of this is to reduce the amount of computational work that must be done on both images together, as well to jointly optimise the single image representation and cross image representation for improved accuracy at a lower computational cost. The authors claim to outperform the majority of contemporary state of the art methods. The work presented in [6] uses a pair of siamese CNNs, one to learn spatial information and the other to learn temporal information. These features are then weighted, as the authors claim that spatial features are more discriminative than temporal features. Depending on the dataset, this method outperforms or shows comparable results to the existing best performing methods. The work in [18] is another siamese CNN with the aim of improving performance for people represented at different scales. This would be an interesting addition to our work, but is beyond the scope of our project.

The authors of [2] do not propose a novel architecture as such, but instead propose a set of good practices that should be followed for effective re-identification, and implement these in their siamese CNN. The recommended good practices that we intend to follow include using data augmentation, as this enables us to grow our dataset substantially whilst also introducing difficult examples that try to ensure that the network is not overfitting to a specific image format. We also ensure that our network has a state of the art base architecture, taking inspiration from the architectures used in [26], [25], [2], [22], [6] to help us to design the most effective re-identification network. We also endeavour to follow the advice of using sufficiently large image resolution by using a size large enough so that the network has plenty of information to learn from. The work of [2] also compares several state of the art approaches on multiple complex datasets. The results of these are quoted in terms of maximum average precision, and the best of these is 91.2%.

A notable observation we have taken from many of these papers is that they train and test their networks on predefined datasets of images, but do not test it as part of a full re-identification system on multiple video files. This is done by the authors of [12], who argue that obtaining the maximum correct matches across a distributed camera network is as important as determining whether two images of pedestrians are the same person or not. They argue that existing pairwise re-identification methods cannot obtain globally optimal re-identification results for a whole camera network, as people may have vastly differing profiles across different cameras. To try and achieve this, the authors use a gradient descent algorithm to obtain a globally optimal matching by maximizing the sum of similarity values for all camera pairs. After obtaining this, they then propagate the difference to adjust their CNN. They report good performance varying between 80% and 99% from different datasets, differing in complexity and number of cameras.

C. TLD Tracker

The Track-Learn-Detect (TLD) algorithm was first proposed by Zdenek Kalal in [10]. The aim of this tracker is to break down the person-tracking task into tracking, learning and detection. The tracking component predicts the location of the object in the next frame. The detection component corrects the tracker if necessary based on true observations. The learning component estimates the errors made by the detector and updates it to avoid these errors in the future. However, the main purpose of TLD is to be able to consistently follow the same object through a changing background, for example tracking a car as it travels along a road using footage from a camera mounted on a helicopter. The authors have said that it performs sub optimally on varying targets such as people. During our early experimentation, we found that is was capable of tracking a person passing through a single camera , but could not re-identify people across multiple cameras, or even reappearing from behind an obstruction.

D. Relation to this Project

In this research we use elements from many of the papers discussed in this section. The TLD tracker is used for tracking a person from frame to frame for as long as they remain visible in the camera view. We disable some of the re-identification based capability of TLD, as this was leading to erroneous results. Whether this person has been previously observed by the system or not is determined by an open world siamese CNN, with an architecture similar to the related work. We take inspiration from [22] and extract the features of each individual image before comparison between them. We also incorporate the recommendations for effective re-identification from [2].

However, many of these papers that propose such a network only show results when it is trained and tested on a specific dataset of still images. They do not integrate it with a tracking system to see how it can perform in the real world, as we propose to do. Even the work presented in [12] does not employ their network on a full system on real time video, although they do take the importance of different camera views into account.

Furthermore, this previous work has all been done on colour imagery. Very little work has been done to try and solve the re-identification problem in thermal imagery, save for our own previous work [19]. Therefore, the major research aim of our project, and where we are advancing the state of the art, is to determine if such a network architecture can successfully be applied to thermal imagery, and whether it can perform effectively as a part of a full re-identification system in real time.

III. Solution

Having established the problem that we want to solve, we now descibe the most important elements of our solution, giving an overview of the structure of our implementation and a description of the elements used.

A. Implementation Structure and Tools Used

The structure of our implementation is shown in Figure 3. The implementation begins by opening each video stream and concurrently running the real time person detection code on each stream. Each time this code identifies a person, it checks whether it has a TLD tracker object associated with them. This is decision point [A] in Figure 3. If not, a new TLD tracker object is created and the siamese CNN is used to compare this person with the other people that have been seen previously. If this person is deemed to be sufficiently similar to one of these people, then they are re-identified as the same person, otherwise the system creates a new person object. This is decision point [B] in Figure 3. Each of these person objects has an associated TLD tracker and set of previous observations, and these are used to facilitate the comparison between people, and are updated each time the person is successfully identified.

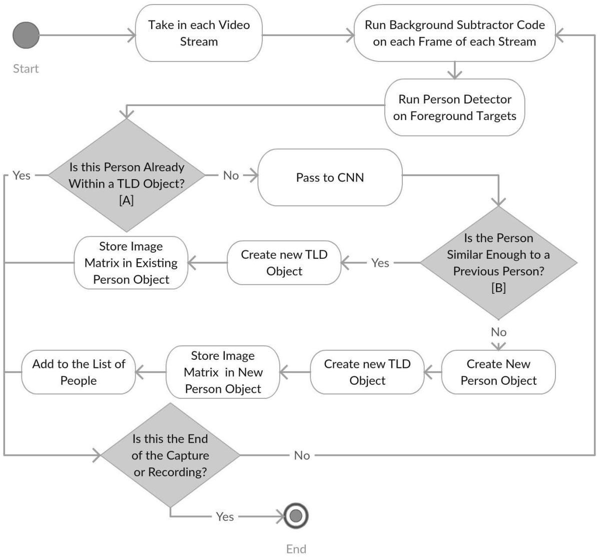

Start
Take in each Video Stream
Run Background Subtractor Code on each Frame of each Stream
Run Person Detector on Foreground Targets
Yes
Is this Person Already Within a TLD Object? [A]
No
Pass to CNN
Store Image Matrix in Existing Person Object
Create new TLD Object
Yes
Is the Person Similar Enough to a Previous Person? [B]
No
Add to the List of People
Store Image Matrix in New Person Object
Create new TLD Object
Create New Person Object
Is this the End of the Capture or Recording?
No
Yes
End

Figure 3: Logic of our re-identification system

Many of the computer vision techniques that we use are complex to implement. Therefore, the OpenCV library [3] has been used to allow us to use stable, well tested code. However, the implementation of TLD code in OpenCV is suboptimal, so we have used the multi-object implementation of TLD from [14]. The neural network side of the project is written in Keras [4], using the Tensorflow backend.

B. Real Time Person Detection

Before we can start concerning ourselves with deep learning based person re-identification, we must be able to identify whether a person is present in the image at all. This will give us a region of interest to pass to the CNN for re-identification, and enable us to use TLD to track targets through single camera views.

1) Background Subtraction: The first stage of the person detection process is background subtraction from the static camera viewpoint. We use the Mixture of Gaussians (MoG) technique to facilitate this, taking inspiration from [27], [28].

This technique works by building up a background model over multiple camera frames, modelling each of the pixels using multiple Gaussians. Using a Gaussian over the last N frames, where N is given by a parameter specifying the rate at which the background model is updated, is far more memory efficient than storing all the pixel values across the entire video capture. This update rate is determined by a trade-off between being fast enough to absorb objects that have become stationary into the background and being slow enough to allow the detection of slow moving objects.

As the program runs through the video, during each new frame, a Gaussian for each pixel is evaluated using a simple heuristic to determine which is most likely to correspond to the background model. Pixels that do not match closely enough with these background Gaussians are classified as foreground elements and added to a new image in the code. Once these foreground pixels are identified and built into a foreground mask, erosion and dilation image operators [20] are used to clean up these results. From here, we use the contour detection to find the connected components and draw bounding boxes around these contours.

2) Person Identification: Having identified the foreground objects, we now determine whether they are people. The implementation uses Histogram of Oriented Gradients (HOG), discussed in [7]. This method works by performing edge gradient calculation on the bounding box identified by the background subtractor. From here cell histograms are computed, with each of the histogram entries filled by gradient magnitudes. These histograms are then used to create overlapping block histograms of the adjacent cells. These block histograms are then concatenated to give a a high dimensional vector know as a HOG descriptor. This HOG descriptor is then passed to a pre-trained machine learning algorithm, in this case a Support Vector Machine (SVM). If this returns a positive identification, then it is classified as a person and will be given an associated TLD tracker object. In our previous work in [19], we selected between HOG and one of its alternatives, Haar cascades, depending on whether runtime or accuracy was more important. However, as we are using a TLD tracker in this research, there will be fewer calls to the person detector so we can use the superior performance of HOG when compared to Haar Cascades without fear of the slower runtime making an impact on the overall performance.

3) Person Tracking using Track-Learn-Detect (TLD): Once we know where the people are in the camera view, we can track their movement. We use the Track-Learn-Detect algorithm, originally proposed in [10]. This algorithm was intended to facilitate the long term tracking of an unknown object in a video stream. The algorithm works by breaking down the person tracking task into tracking, learning and detection. The tracking component predicts the location of the object in the next frame. The detection component corrects the tracker if necessary based on true observations. The learning component estimates the errors made by the detector and updates it to avoid these errors in the future. The relationship between these components is shown in Figure 4.

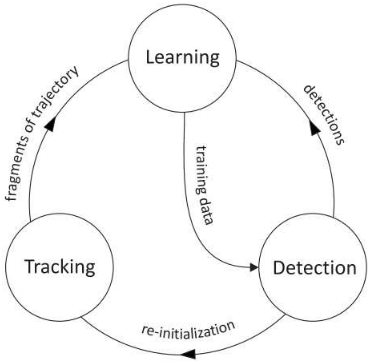


Figure 4: Block diagram of the TLD framework, from [10]

In our early experimentation, we found that the learning part of this algorithm was attempting to perform re-identification when people left and re-entered a video stream, or appeared in a different one. As the majority of these re-identifications were erroneous, we cut some of the functionality out of TLD for this implementation, so that each object was deleted when it left the scene or became obstructed. This eliminated the misclassifications that the limited re-identification capabilities of this algorithm were causing, enabling us to effectively track people across the scene, while using our CNN for the re-identification.

Consequently, in our implementation each person currently present in any of the camera views has an associated TLD tracker object, which is created on the first HOG identification and classified by the CNN. The tracker then follows this person through the frame until either the person leaves the frame or HOG is no longer getting a positive identification within the tracker. If either of these conditions occurs, the TLD object is deleted. In the case of no HOG identification being present, this indicates that the tracker has incorrectly predicted the position, or rate of movement, of the person and has lost them. A new tracker will then be created to replace it when HOG detects this person again. Figure 5 shows the TLD system in action following a human target through a camera view, as well as displaying the positive and negative matches used to inform the learning part of TLD. The dotted boxes represent other possible areas that the TLD system thinks the target may be in, while the blue tracking points show the features on the person that the system is looking for in each frame. Some are not found, so the region containing the best of these is chosen.

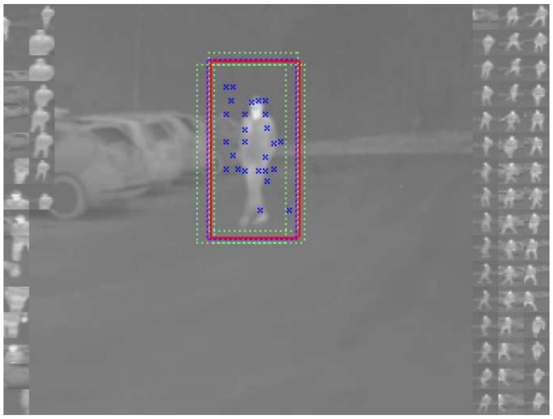


Figure 5: An example of TLD tracking a person through a scene. The images on the left are negative matches, and the images on the right are positive matches

C. Deep Learning

Taking inspiration from the state of the art literature in the area of open world person re-identification, our network architecture is a deep siamese CNN. This means that the network is trained on pairs of images, and attempts to determine whether these images show the same person or a different person. The size of these images was fixed at 256x128, to ensure that the size of the file had no effect on the training. These images were then formed into a balanced dataset of positive and negative pairs. The positive pairs contain two images of the same person, labelled 1, and negative pairs contain different people, labelled 0. We then applied some data augmentation to the images, following the advice of [2], with the aim of introducing more variation into our dataset to enable it to cope better with difficult real world situations. The possible augmentations include small rotations, translations and horizontal reflection. This data is split 60:20:20 into training, validation and testing sets respectively. The purpose of the training set is to train the model. The validation set is the sample of data used to provide an evaluation of the model during training, whereas the test set is made up of previously unseen data that is used to provide an unbiased evaluation of the final model.

The two sister networks which make up the siamese network have exactly the same architecture and weights, as they are trained together. Our CNN architecture consists of convolutional layers, pooling layers and fully connected layers, as seen in Figure 6.

The convolutional layers operate directly on the image to reduce the complexity of the input in a manner that is both meaningful and structured. This enables the network to have fewer neurons than a traditional fully connected feedforward network would have if operating on an image. This enables the convolutional neural network to be far deeper with fewer parameters. This resolves the vanishing or exploding gradients problem that would otherwise occur, as a greater number of training parameters in the early layers of the network would make it unstable, and prone to the weights becoming much too large (exploding gradients) or much too small (vanishing gradients).

Pooling layers then downsample their input, converting a neighbourhood of pixels into a single value. In the case of max pooling that we have used here, this is the maximum value. The aim of this is to provide robustness to the changes in the spatial location of features across the dataset and to reduce input dimensionality. This is also an effective tool to control overfitting, as it reduces the number of training parameters and ensures that it is the general regions where features are present that are learned, not the specific pixels of the images in the training set in which they are present.

Finally, we flatten the output of our last pooling layer to a single dimension and then pass it to a fully connected layer. The high dimensional vector output of this layer will be the representation of this image, and we will use the Euclidean distances between these vectors to determine whether the inputs show the same person or different people.

At various stages of the network, we have some dropout layers. The purpose of dropout is to eliminate a proportion of the training data at each epoch to help to prevent overfitting, as the network is being trained on a different dataset each epoch. The goal is therefore to force the network to learn more robust features that are useful in conjunction with random subsets of the other neurons so that good performance is still achieved when some neurons are removed.

We now consider the loss function used for our CNN. A loss function is a measure used to quantify the cost of a mistake in classification. The loss function that we have chosen to use for this network was proposed by the authors of [9], and is given by Equation (1) . This paper defines a contrastive loss function, which maps high dimensional inputs to lower dimensional outputs, given distances between samples in its input space. These distances between samples are supplied by the Euclidean distance. The formula for this function is given by Equation (1) , where Y is the label (0 or 1), m is the margin that determines the maximum Euclidean distance between two points that will influence the loss function. The purpose of this is to ensure that vastly different images do not have a disproportional effect on the learning outcome. D is the Euclidean distance, which is defined here by Equation (2), where X_1 and X_2 are the input images and F represents the feature vectors output by the siamese pair of identical CNNs.

$$L = (1-Y)\frac{1}{2}(D)^2 + Y\frac{1}{2}\{max(0, m-D)\}^2 \tag{1}$$

$$D = \sqrt{\{F(X_1) - F(X_2)\}^2} \tag{2}$$

The architecture that we chose to use is made up of two convolutional layers, a max pooling layer, a dropout layer, two convolutional layers, a max pooling layer, a dropout layer, a flatten layer and a dense fully connected layer. The parameters for these layers were all determined by an exhaustive grid search using the Hyperas library [17]. For the convolutional layers the parameters we were experimenting with were the dimensionality of the output space (i.e. the number of output filters in the convolution) and the kernel size, specifying the width and height of the convolution window. For the pooling layers we experimented with the pool size, which gave the

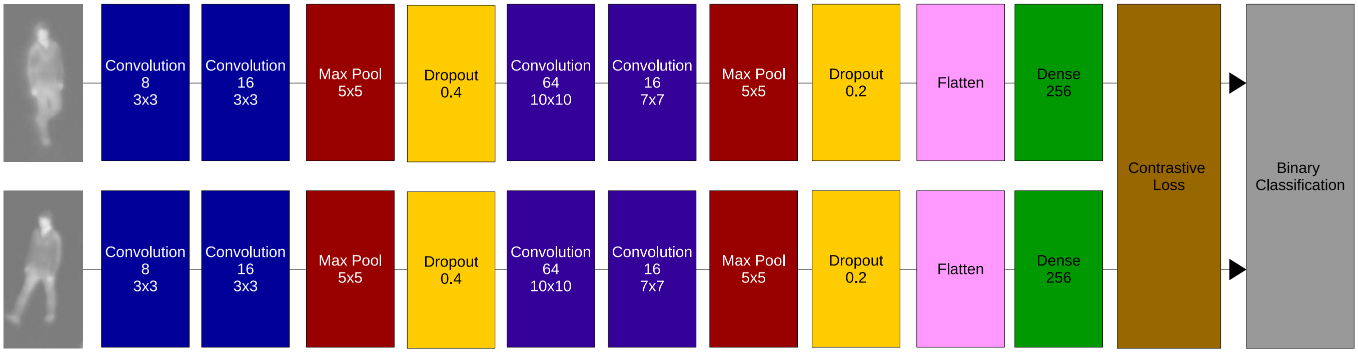


Figure 6: The architecture of our CNN. Diagram and colours inspired by GoogLeNet [21]

factors by which to downscale the input. For the dropout layers we experimented with how much of the dataset should be removed at each point, and for the dense layer we experimented with the dimensionality of the input space. The chosen parameters are shown in Figure 6. We also experimented with various activation functions for the convolutional layers and the dense layers, before settling on the rectified linear unit (ReLU) activation function.

ReLU was chosen as our activation function as it is a good approximator, and can be used to approximate any function by stacking layers, as we have in this architecture. Also, the fact that ReLU only activates on positive weights leads to sparse activation and a lighter network. This, combined with the fact that ReLu is less computationally expensive than other activation functions such as tanh and sigmoid because it involves simpler mathematical operations, makes the network quicker and easier to train and query. Also the fact that ReLU is a good general classifier that can approximate any function is important for this purpose, as we do not know what features we are trying to learn for effective re-identification in thermal imagery as a consequence of our previous work in [19].

D. Implementation Issues

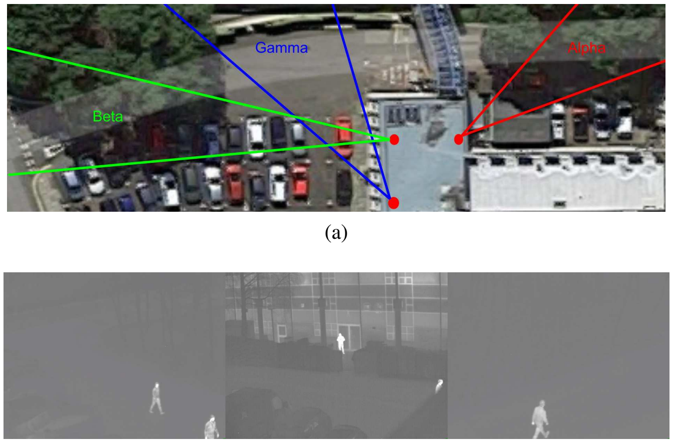
In the implementation of this system we have used a modular approach, implementing each major component separately. The multi-object implementation of TLD from [14] was originally written in C as a header-file based library, which was incompatible with the rest of our C++ codebase. We therefore had to spend considerable time rewriting this library so it was compatible with our system, as well as refactoring and adding new functions to suppress the erroneous attempts at re-identification and to enable us to delete TLD objects from the system. Also, as Keras uses Tensorflow as its backend, and OpenCV has the ability to read and use Tensorflow networks, we assumed OpenCV would be able to read the network directly. However, this turned out to be incompatible, which required us to spend much time trying to import it correctly, before deciding that the superior method was to use both C++ and Python for the separate components and connect them at system level.

IV. Results

We now present the performance that our system has been able to acheive, both as a standalone CNN and as part of a full re-identification system and evaluate the strengths, weaknesses and limitations of the research that has been presented here.

A. The Dataset

The data used to evaluate the re-identification system was gathered with three cameras at Durham University. The cameras and their fields of view are arranged as in Figure 7 (a), with the cameras labelled as camera α, β and γ respectively, and their fields of view shown by the matching coloured lines. The images seen by the cameras are shown in Figure 7(b). The dataset contains five people. These videos make up the data we use to test the full re-identification system, and the person images extracted from them make up the dataset for the CNN.



(a)

(b)

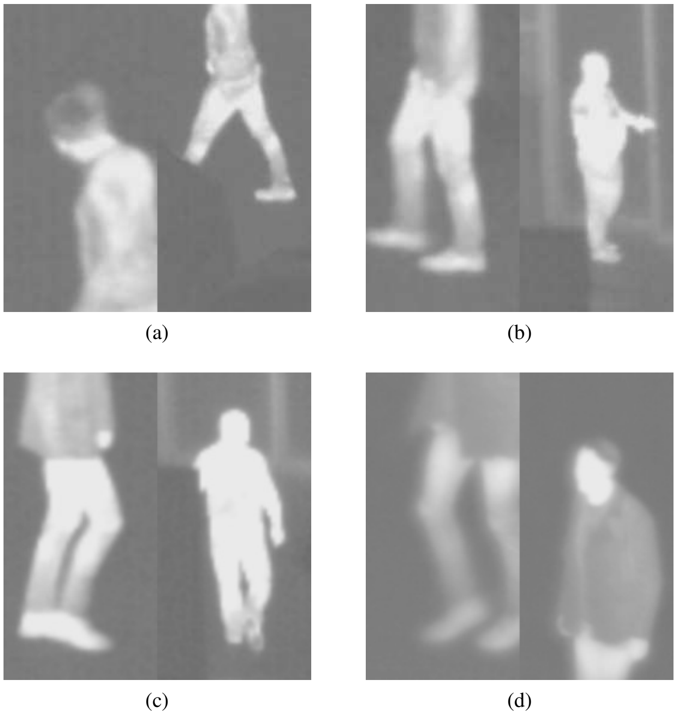
Figure 7: Position (a) and field of view (b) of the cameras used to record our this dataset

B. Real Time Person Detection System

The inclusion of the TLD tracker has greatly improved the performance of our person tracking system in comparison to our previous work in [19]. We are getting fewer misclassifications from HOG as it is being called less frquently due to the TLD tracker. Table I shows the number of images from each camera view that contain a person and the number of these images that were passed to the CNN for re-identification. The remainder of the images were handled by the TLD tracker as the people were within one of the TLD objects, and did not need to be re-identified. Without TLD, all of the frames would have to be passed to the CNN, which would increase runtime and the amount of computational resources used, as well as lead to the possibility of more misclassifications. As the results in Table I show, the person detection system performs particularly well on cameras α and β, but less well on camera γ. This seems to be due to the fact that HOG is worse at person identification from a side profile than from a front profile. This inferior performance is also due to the fact that many of the people present in camera γ are partially obstructed, leading to the TLD objects being deleted. We also encounter a few errors caused by HOG classifying a small part of a person as a full person, leading to errors from the CNN. The effect of these errors is discussed in Sections IV-D and IV-E, and are shown in Figure 8.

Camera	Frames with person present	Calls to CNN
Alpha	1259	65
Beta	1863	114
Gamma	2454	339

Table I: Performance gains from TLD


(a)

(b)

(c)

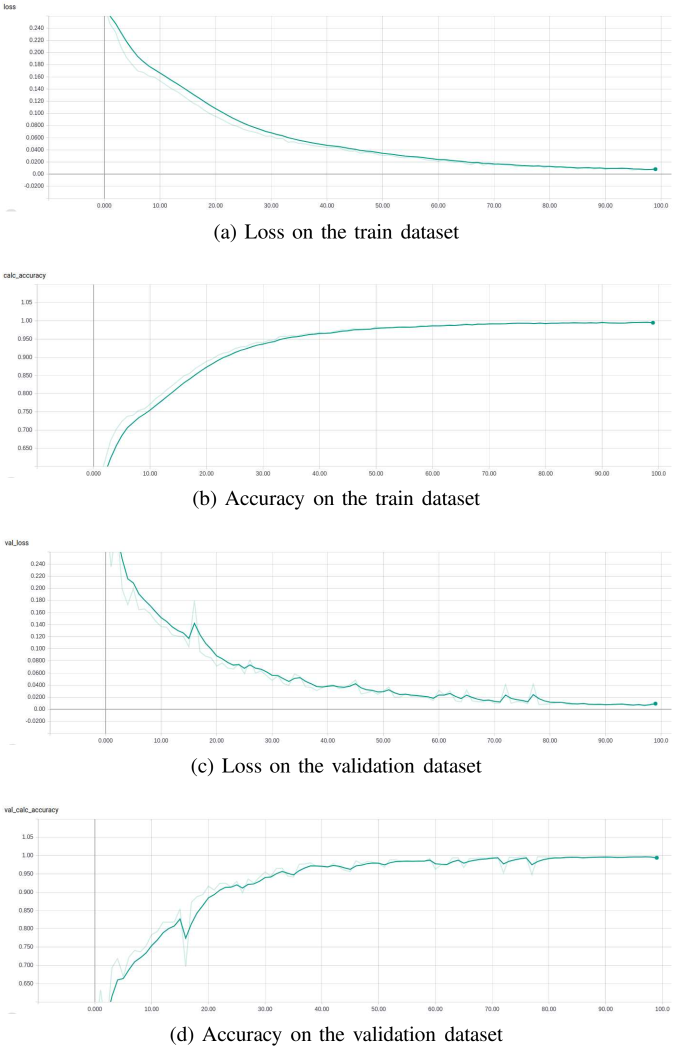
(d)

Figure 8: Hog Errors

C. The Deep Learning System

When designing our network we did many training runs to determine the optimal architecture, and then performed extensive grid searches to optimise the parameters. While training the network, the metrics with which we were primarily concerned were the loss, from the contrastive loss function defined in Section III-C, and the accuracy, meaning the percentage of the image pairs that were classified correctly. We calculate the accuracy based on a re-identification threshold of 0.5. This means that a pair of images with a Euclidean distance of less than 0.5 would be classified as the same person, whereas a pair with a Euclidean distance of greater than 0.5 would be classified as a different person. Figure 9 shows how the loss and accuracy varied on our training and validation sets as the network was trained over 100 epochs.

Figure 9 shows us that the training and validation patterns of the network matched very closely, with the loss steadily decreasing and the accuracy steadily increasing before both settled towards the end of training. The use of 100 epochs gave both the loss and the accuracy time to stabilise, and ensured that the final values of the network weights were optimal.



(a) Loss on the train dataset

(b) Accuracy on the train dataset

(c) Loss on the validation dataset

(d) Accuracy on the validation dataset

Figure 9: Loss and Accuracy results during training of the network over 100 epochs

Table II shows the confusion matrices of the network when predicting the training, validation and testing data partitions. These compare the output of the network against our ground truth labels and report the number of true positives (TP), where a pair is correctly classified as the same person, false positives (FP), where a pair is wrongly classified as the same person, false negatives (FN), where a pair is wrongly classified as different people and true negatives (TN), where a pair is correctly classified as different people. Table III shows the classification reports for these data partitions, reporting the Accuracy, which is the percentage of the image pairs that were correctly classified; Precision, which is the percentage of the pairs that we labelled the same person actually were the same person; Recall, which is the percentage of the pairs that were the same person that we correctly labelled; and F1-Score, which is the average of precision and recall, taking into account both false positives and false negatives. The equations of accuracy, precision, recall and F1-score are given in Equations (3), (4), (5) and (6).

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{3}$$

$$Precision = \frac{TP}{TP+FP} \tag{4}$$

$$Recall = \frac{TP}{TP+FN} \tag{5}$$

$$F1-Score = 2\times\frac{Precision\times Recall}{Precision+Recall} \tag{6}$$

The results from these are encouraging, as all are high percentages across all data partitions, giving us confidence that our system is working well and has learnt useful features for re-identification. Table II shows that the number of false negatives is greater than the number of false positives, which is advantageous for our re-identification system, as misclassifying different people as the same person (FP) is more destructive to our re-identification system than misclassifying the same person as different people (FN), as false positives can lead to corruption of the data assotiated with a particular person. False negatives would create unnecessary new targets, but this is easy to deal with, as we would be able to clearly determine that one person in the video stream was represented by two person objects in our system.

		Predicted Label	
		Positive	Negative
Ground Truth	Positive	TP = 5924	FN = 11
	Negative	FP = 2	TN = 5933

(a) Confusion matrix for training set

		Predicted Label	
		Positive	Negative
Ground Truth	Positive	TP = 1675	FN = 20
	Negative	FP = 4	TN = 1691

(b) Confusion matrix for validation set

		Predicted Label	
		Positive	Negative
Ground Truth	Positive	TP = 2097	FN = 18
	Negative	FP = 9	TN = 2106

(c) Confusion matrix for test set

Table II: Confusion Matrices

Metric	Accuracy	Precision	Recall	F1 Score
Value	99.89%	99.81%	99.97%	99.89%

(a) Classification report for training set

Metric	Accuracy	Precision	Recall	F1 Score
Value	99.29%	99.76%	98.82%	99.29%

(b) Classification report for validation set

Metric	Accuracy	Precision	Recall	F1 Score
Value	99.36%	99.15%	99.57%	99.36%

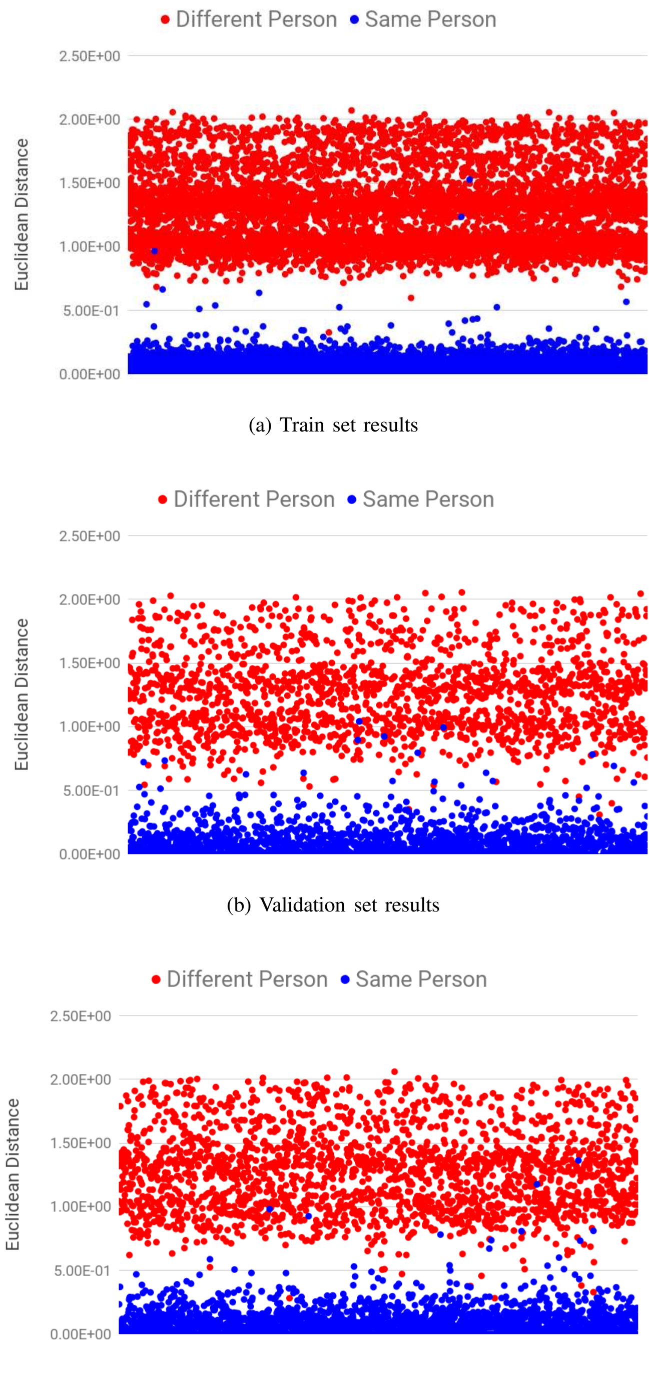
(c) Classification report for test set

Table III: Classification Reports

Figure 10 is a graphical representation of the Euclidean distances reported by the network on the train, validation and test data splits. These graphs show a clear separation between the pairs of images of the same person and pairs of images of different people. This separation is a good result because it shows that the Euclidean distances of the positive image pairs were much lower in almost all cases to the Euclidean distances of the negative pairs. The separation for the training set is more clear than the split for the validation and testing set. This improved performance on the training set is expected, as these are the exact images the network was trained on, so some level of overfitting to the specific training data was expected.

It is interesting at this point to note which of the image pairs were the misclassifications, and also to establish which were close and which were far apart. Figure 11 shows examples of these images, with their Euclidean distances in the captions, as well as whether they were a true positive, false positive, false negative, or true negative. As before, those with a Euclidean distance less than 0.5 were classified as the same person, while those with a Euclidean distance greater than 0.5 were classified as different people. Subfigures (a) and (b) of Figure 11 show true positive classifications with very small Euclidean distances. Subfigures (c) and (d) show false negatives, where (c) is a difficult classification as the two images are very similar, and (d) has a Euclidean distance on the boundary of a positive or negative classification. Subfigure (e) is the other side of this boundary, showing a close false negative. Subfigure (f) is another false negative, which seems to be a difficult classification due to a different viewpoint and resolution. Figures (g) and (h) then show comfortable true negative classifications.

From these results, we can hypothesise that the small number of mistakes that our system makes on this dataset are due to low resolution images that are difficult to distinguish between as in Subfigure (c), or due to differences in viewing angle and resolution as in Subfigures (d), (e) and (f). These are due to the differing viewing angles between cameras α and β, which are quite similar, and camera γ, which is different. As the Euclidean distances show, particularly in the cases of Subfigures (d) and (e), these lie on the borderline of positive and negative classifications, showing almost identical comparisons between images taken from different viewing angles at different resolutions.



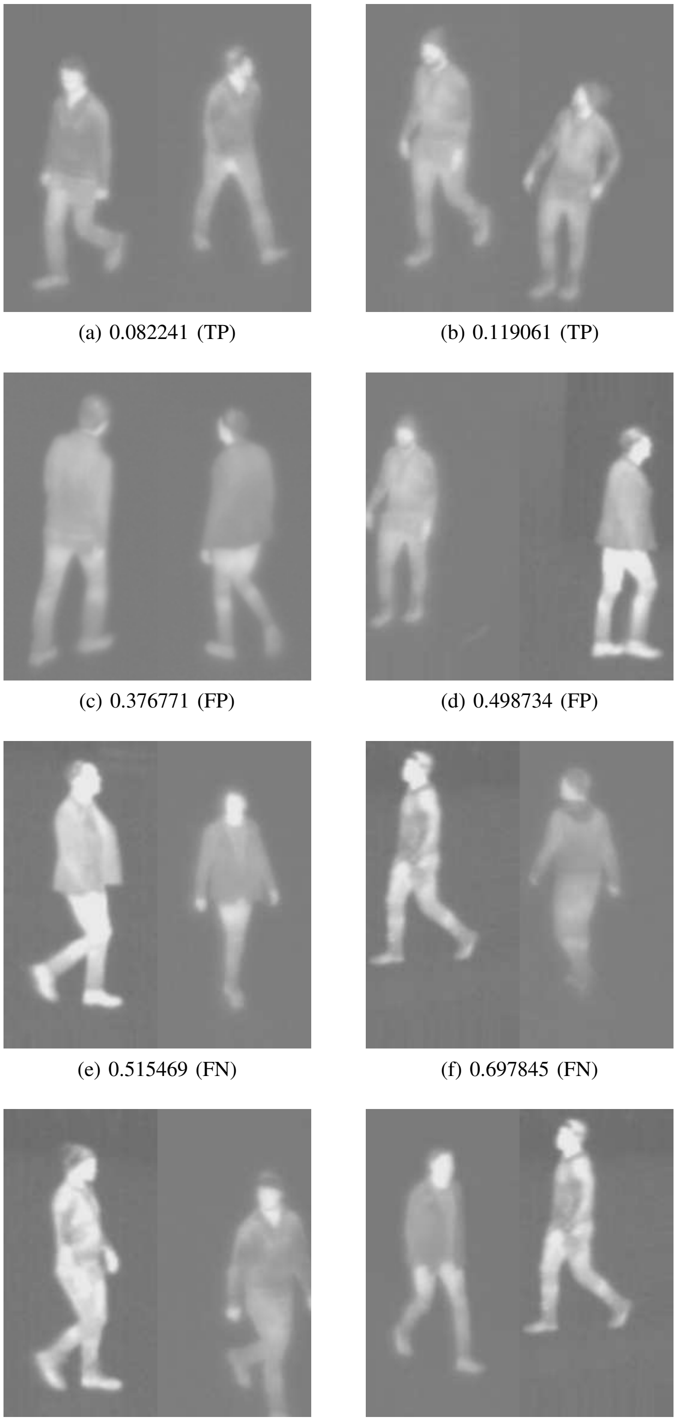
(a) Train set results

(b) Validation set results

(c) Test set results

Figure 10: Euclidean distance results of the CNN on the Train, Validation and Test Data Partitions

We also tested this system on an alternative dataset, the one used for our work in [19]. Henceforth, we shall refer to this dataset as the low level dataset, as the cameras were located at ground level, compared to our main dataset, which we shall refer to as the high level datsaset, as the cameras were situated on the roof. As cameras for the low level dataset are placed in different locations to the high level dataset, the profiles of the people are different in both resolution and viewing angle. The results that the network gave for the low level dataset are shown in Table IV , Table V and Figure 12.



(a) 0.082241 (TP)

(b) 0.119061 (TP)

(c) 0.376771 (FP)

(d) 0.498734 (FP)

(e) 0.515469 (FN)

(f) 0.697845 (FN)

(g) 0.734032 (TN)

(h) 0.960652 (TN)

Figure 11: Image pairs and their Euclidean distances

		Predicted Label	
		Positive	Negative
Ground Truth	Positive	TP = 5738	FN = 1318
	Negative	FP = 3286	TN = 3815

Table IV: Confusion Matrix

Metric	Accuracy	Precision	Recall	F1 Score
Value	67.58%	81.44%	63.77%	71.53%

Table V: Classification Report

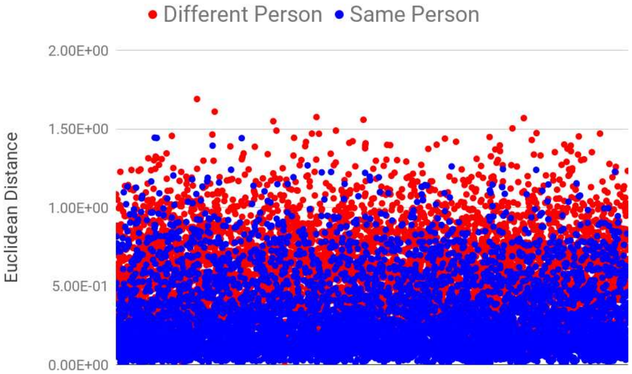
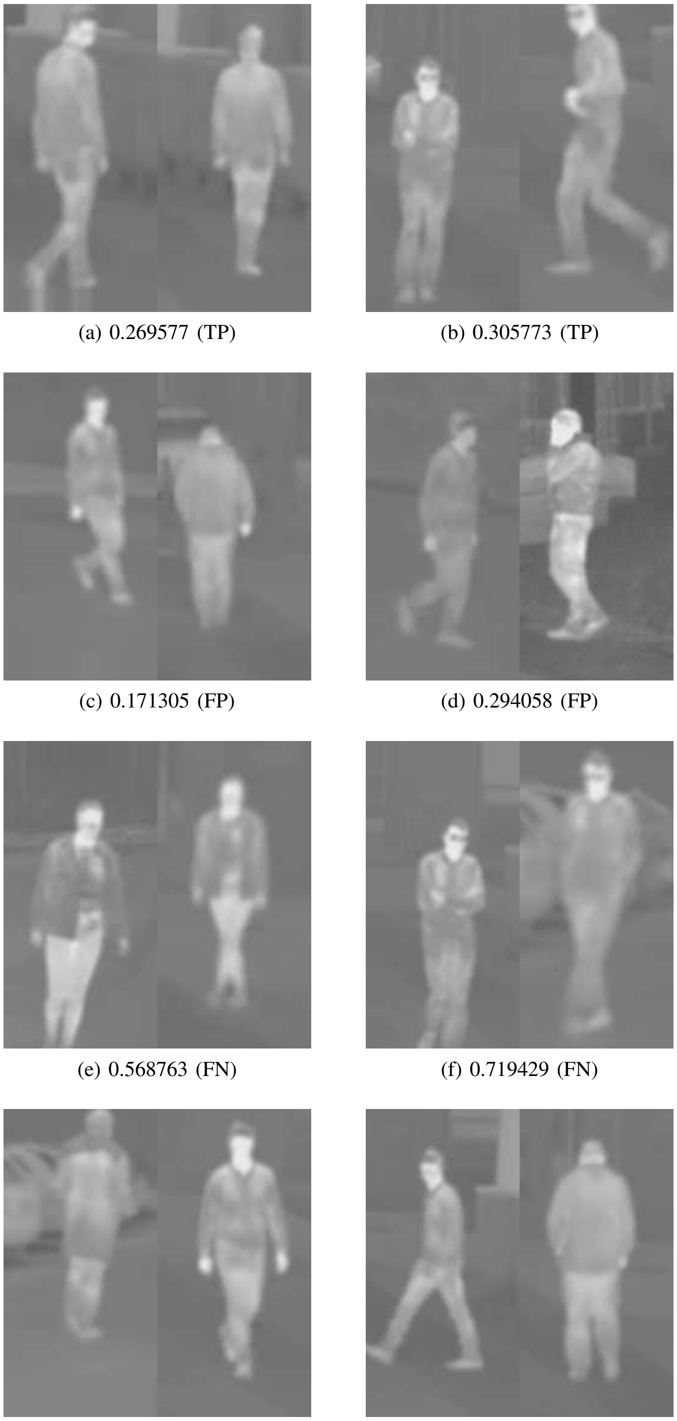


Figure 12: Euclidean distance results

As we can see from these results, the performance of the system on the low level dataset is substantially worse than the high level dataset, which the CNN was trained on. We hypothesise that this is due to the different locations of the cameras. The effect of this can be seen more clearly when the classified image pairs of Figure 11 are contrasted with those in Figure 13. Subfigures of (a) and (b) of Figure 13 are true positive classifications, but have larger euclidean distances than the true positive examples from the high level dataset. Subfigures (c) and (d) again show false positives, but these seem to be fairly easily identifiable as different people, as opposed to the false positives from Figure 11, which appear to be some of the more difficult classifications. The same is true for the false negatives, as those shown in Figure 13 appear to be fairly easy to classify as the same person, whereas those shown in Figure 11 are more difficult classifications. Subfigures (g) and (h) are true negatives, but seem to be of comparable difficulty to the false positives.

As we can clearly see, the images in Figure 13 are higher resolution and are taken from a different angle, and our network seems to have trouble classifying them correctly. Unlike our evaluation of the erroneous classifications from Figure 11, the results from Figure 13 do not show an obvious trend to explain why the misclassifications have occurred, as they seem relatively simple classifications to a human observer. We therefore conclude that the network does not perform well on this dataset.



(a) 0.269577 (TP)

(b) 0.305773 (TP)

(c) 0.171305 (FP)

(d) 0.294058 (FP)

(e) 0.568763 (FN)

(f) 0.719429 (FN)

(g) 0.848649 (TN)

(h) 0.997519 (TN)

Figure 13: Image pairs and their Euclidean distances

To identify whether the cause of these inferior results was that the low level dataset is more difficult or that the network does not work well as a general classifier, we trained the network on the low level dataset and then tested it on the high level dataset. The results are shown in Table VI, presented as a comparison of the accuracies possible on the test splits of the two datasets depending which has been used for training. This shows that although the low level dataset achieved 2% lower accuracy, and is therefore slightly more difficult, the real issue is that the ability of this CNN to function as a general re-identification system on any location or camera layout is limited. Solving this is the natural extension to the work that we have presented here, as we have shown that a siamese CNN works well for re-identification, and now needs to improve as a generic classifier. Due to the poor results achieved here on the low level dataset, we will only use the high level dataset to test our full re-identification system, as this will highlight any more problems that may arise and their effects, rather than reinforcing the fact that the network is not a very good generic classifier.

Training Dataset	High Level Accuracy	Low Level Accuracy
High Level	99.36%	67.58%
Low Level	69.83%	96.89%

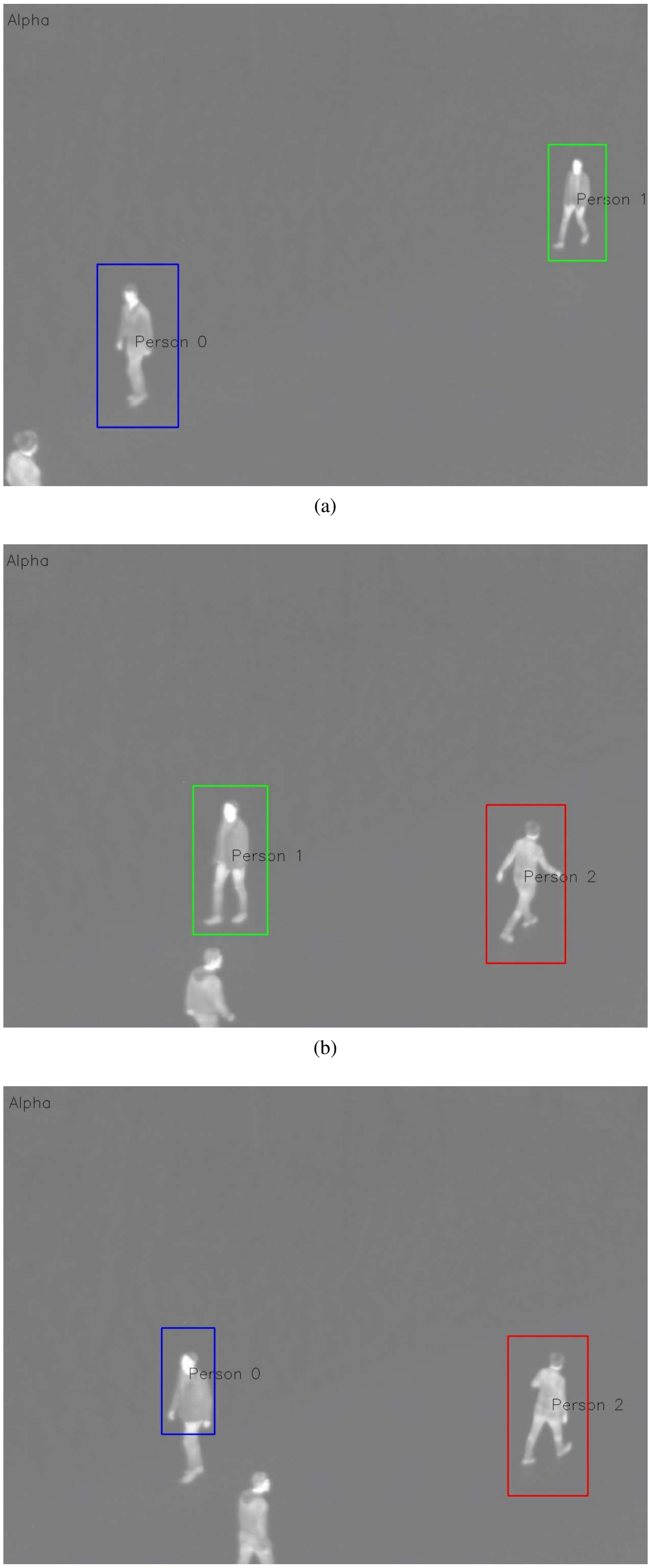
Table VI: Comparison of accuracies from training on the two different datasets

D. Single Camera Re-Identification System

Having trained and evaluated the CNN, we then integrate this with our person detection system to form a full re-identification system, and run this on our high level dataset. Each image that needs classification is paired up with each of the images already in the system. It is re-identified as the person whose data gives the best match of these, as long as the Euclidean distance is less than 0.5. If it is greater 0.5, then a new person object is created. Each person object has two previous images associated with them, which update as the person moves though the cameras. We found this number of previous images to be the best compromise between runtime and accuracy.

First, we consider the performance of the system when run on a single camera view. This enables us to simply and effectively test and evaluate our re-identification system on one camera viewpoint to remove the potential misclassifications that could result from examining all of the camera views at once. Examples of such misclassifications from this dataset can be found in Subfigures (d), (e) and (f) of Figure 11.

We ran the system on all three camera views from the high level dataset, and selected images of the performance are shown in Figures 14, 15 and 16. The output appears to be accurate, as each person is constantly followed by their own associated bounding box, both with colour and number, for as long as they are in the frame. To a human observer watching these videos in real time, it would appear to be almost 100% accurate. However, there are a few small errors where some of the people get missclassified as another person or get a new target associated with them for a single frame, before they revert to their correct target. This is caused by errors from the HOG person classifier, where it has identified a small part of a person to be a complete person, as previously dicussed in Section IV-B and shown in Figure 8. Examples of where this has occurred are shown in Subfigure (c) of Figures 14, 15 and 16, while Subfigures (a) and (b) show the system performing well. Footage of the system running on these cameras individually can be found in a YouTube playlist at https://tinyurl.com/ycmsfsce. The accuracy results were 95.38% for camera α, 99.12% for camera β and 96.17% for camera γ.

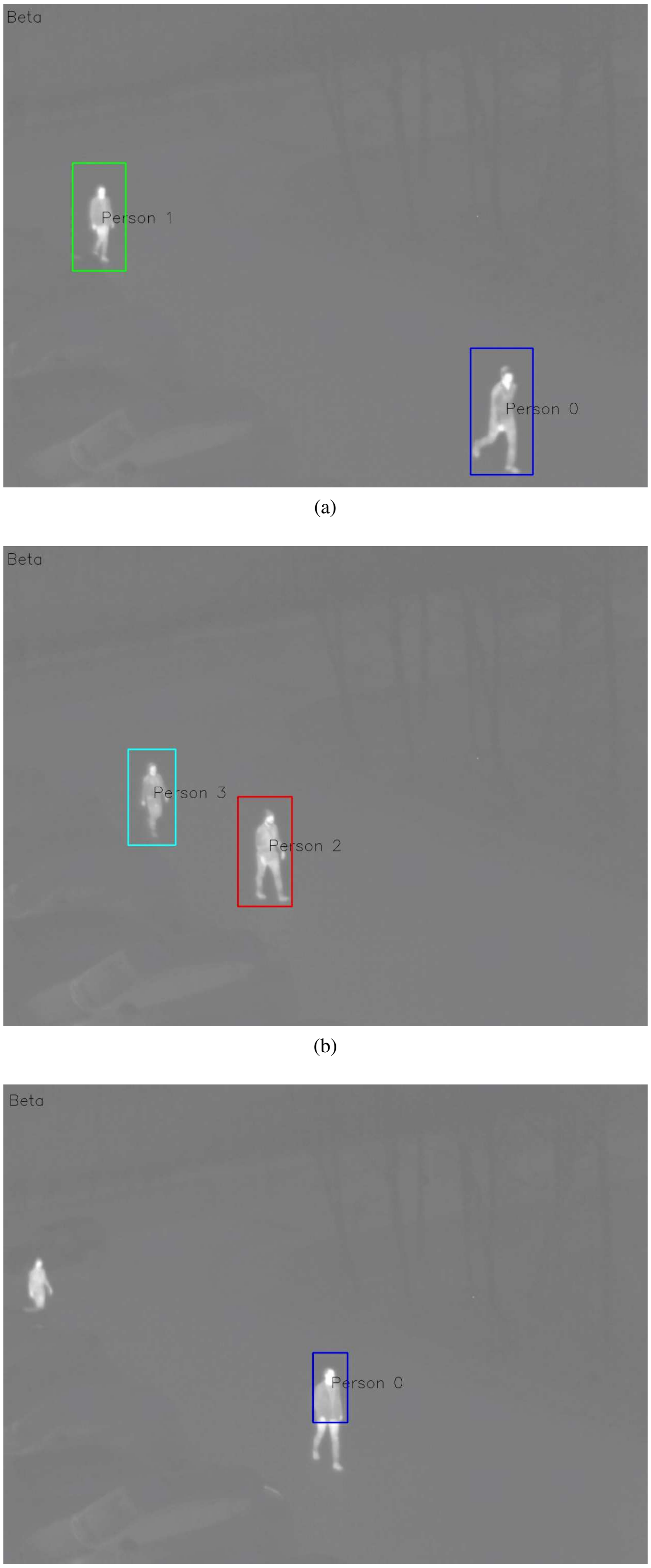


(a)

(b)

(c)

Figure 14: Camera α

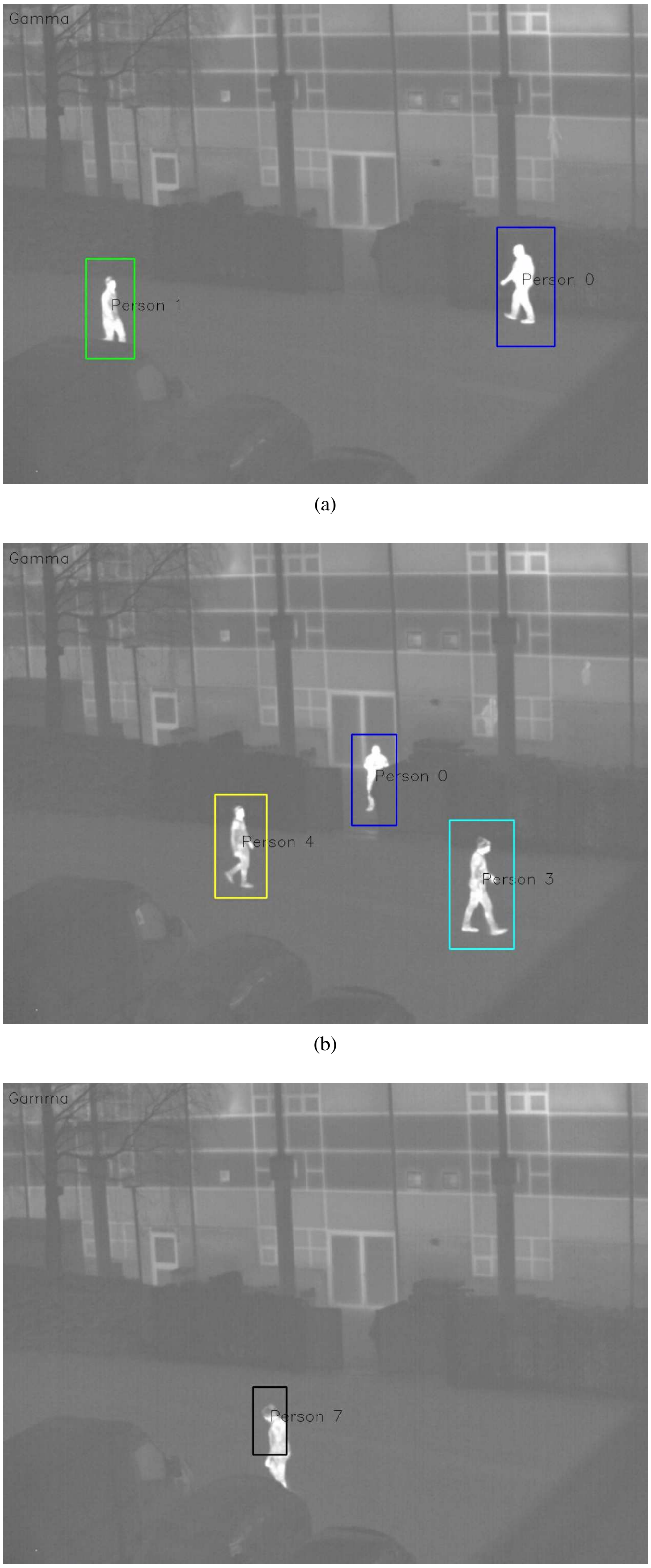


(a)

(b)

(c)

Figure 15: Camera β



(a)

(b)

(c)

Figure 16: Camera γ

E. Multi-Camera Re-Identification System

We now run the system on all of the camera views at once. This introduces the different viewpoints that we feared may cause some errors. Some images of the results of this are shown in Figure 17, and a video can be found in the YouTube playlist at https://tinyurl.com/ycmsfsce. The performance was good once again, with only a few errors, and these do not have a particularly damaging impact on the performance of the the full system. The system made gave us an accuracy of 96.32%.

As previously discussed for the single camera system, there were a few frames that caused misclassifications due to errors with HOG. There were also some false negative classifications where additional targets were created for some of the people present in the dataset. This is fairly easy for a human operator to deal with, as these extra targets still only ever identify a single person, so we can clearly determine that one person in the video stream was represented by two person objects in our system. Examples of this are shown in Subfigure (c) of Figure 17, where person 0 has become person 5, and in Subfigure (d) of Figure 17, where person 1 has become person 8. The majority of these new targets were created as the people moved through camera γ, as this camera is at a different angle to the people and causes them to have a different thermal profile.

V. Conclusion

In this paper, we have developed a fully functional thermal re-identification system using a track-learn-detect (TLD) tracker and a deep siamese CNN that performs well on a varied dataset. The figures shown in this paper and the videos athttps://tinyurl.com/ycmsfsce verify this claim of performance, with an accuracy of 96.32% on the multi-camera system. As there has been very little previous work on solving the re-identification problem in thermal, we are defining the state of the art with this work, and have shown that it is possible when using deep learning to extract features that would not necessarily occur to a human observer when trying to re-identify people. Possible extensions to this work would be to make the network a better generic classifier across multiple datasets and camera arrangements, as we have shown that our network functions very well on our camera arrangement, with 99% accuracy, but far worse on an alternative camera arrangement, with only 67% accuracy. Inspiration for this could be drawn from the previous work in [15] and [18], where this problem is addressed for colour re-identification.

References

[1] Ejaz Ahmed, Michael Jones, and Tim K Marks. An Improved Deep Learning Architecture for Person Re-Identification. *Computer Vision and Pattern Recognition (CVPR)*, pages 3908–3916, 2015.

[2] Jon Almazan, Bojana Gajic, Naila Murray, and Diane Larlus. Re-ID done right: towards good practices for person re-identification. *CoRR*, 1801.0, 2018.

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

[4] François Chollet. Keras, 2015.

[5] Chun-Te Chu and Jenq-Neng Hwang. Fully Unsupervised Learning of Camera Link Models for Tracking Humans Across Nonoverlapping Cameras. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(6):979–994, jun 2014.

[6] Dahjung Chung, Khalid Tahboub, and Edward J Delp. A Two Stream Siamese Convolutional Neural Network for Person Re-identification. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1992–2000, 2017.

[7] Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005.*, 2005.

[8] Shaogang Gong, Marco Cristani, Chen Change Loy, and Timothy M Hospedales. The Re-Identification Challenge. In *Person Re-Identification*, pages 1–21. 2014.

[9] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1735–1742, 2006.

[10] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-Learning-Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(1), 2010.

[11] Ryan Layne, Timothy M Hospedales, and Shaogang Gong. Attributes-based Re-Identification. In *Person Re-Identification*, pages 93–121. 2014.

[12] Ji Lin, Liangliang Ren, Jiwen Lu, Jianjiang Feng, and Jie Zhou. Consistent-Aware Deep Learning for Person Re-identification in a Camera Network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3396–3405, 2017.

[13] Chunxiao Liu, Shaogang Gong, and Chen Change Loy. On-the-fly feature importance mining for person re-identification. *Pattern Recognition*, 47(4), 2014.

[14] Christian Lutz and Thorsten Engesser. MOTLD, 2015.

[15] Niki Martinel, Gian Luca Foresti, and Christian Micheloni. Person Reidentification in a Distributed Camera Network Framework. In *IEEE Transactions on Cybernetics*, pages 1–12. IEEE, 2016.

[16] Tetsu Matsukawa and Einoshin Suzuki. Person re-identification using CNN features learned from combination of attributes. In *Proceedings - International Conference on Pattern Recognition*, pages 2428–2433, 2017.

[17] Max Pumperla. Hyperas, 2017.

[18] Xuelin Qian, Yanwei Fu, Yu Gang Jiang, Tao Xiang, and Xiangyang Xue. Multi-scale Deep Learning Architectures for Person Re-identification. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2017-Octob, pages 5409–5418, 2017.

[19] Thomas A. Robson and Toby P. Breckon. *Camera-to-Camera Tracking for Person Re-identification within Thermal Imagery*. Bachelors thesis, Durham University, 2017.

[20] C J Solomon and T P Breckon. *Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab*. Wiley-Blackwell, 2010.

[21] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going Deeper with Convolutions. *CVPR2015*, 2014.

[22] Faqiang Wang, Wangmeng Zuo, Liang Lin, David Zhang, and Lei Zhang. Joint Learning of Single-Image and Cross-Image Representations for Person Re-identification. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1288–1296, 2016.

[23] Yimin Wang, Ruimin Hu, Chao Liang, Chunjie Zhang, and Qingming Leng. Camera compensation using a feature projection matrix for person reidentification. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(8):1350–1361, aug 2014.

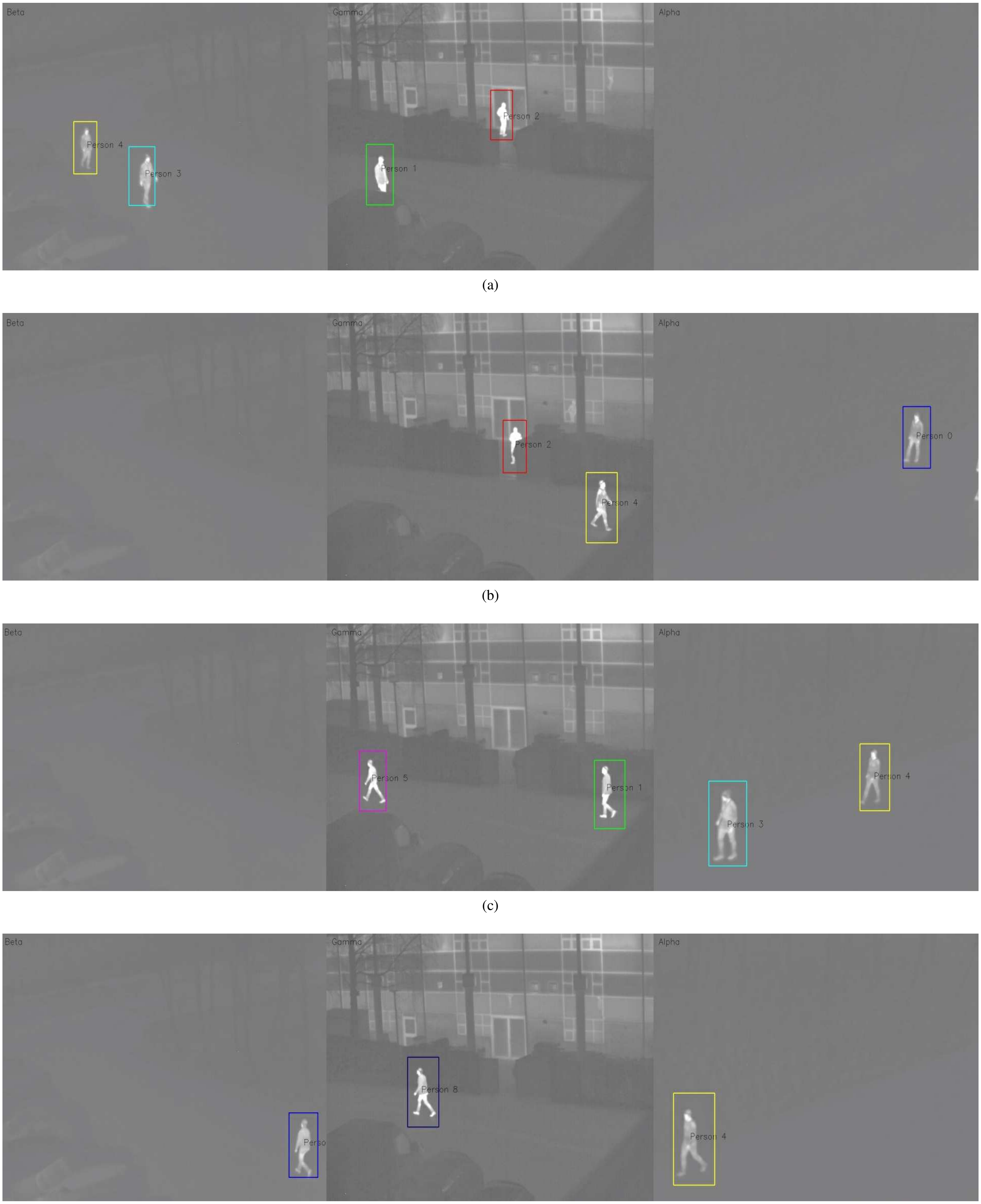
[24] Yang Wu, Michihiko Minoh, Masayuki Mukunoki, Wei Li, and Shihong Lao. Collaborative sparse approximation for multiple-shot across-camera person re-identification. In *Proceedings - International Conference on Advanced Video and Signal-Based Surveillance*, pages 209–214. IEEE, sep 2012.

[25] Guanwen Zhang, Jien Kato, Yu Wang, and Kenji Mase. People re-identification using deep convolutional neural network. *VISAPP 2014 - Proceedings of the 9th International Conference on Computer Vision Theory and Applications*, 3:216–223, 2014.

[26] Zhedong Zheng, Liang Zheng, and Yi Yang. A Discriminatively Learned CNN Embedding for Person Re-identification. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 14(1), 2016.

[27] Zoran Zivkovic. Improved Adaptive Gaussian Mixture Model for Background Subtraction. *Proceedings of the 17th International Conference on Pattern Recognition*, 2004.

[28] Zoran Zivkovic and Ferdinand Van Der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, 27(7):773–780, 2005.



(a)

(b)

(c)

(d)

Figure 17: Multi-camera re-identification results