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

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

***Abstract*—Although use of thermal imagery currently poses significant advantages for 24/7 surveillance in terms of the visibility of targets under all environmental conditions, a key limitation is the lack of colour information. Person re-identification across multiple cameras is a key research problem within the domain of visual surveillance and a 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 similar features to solve the thermal re-identification problem with little success [19], and are now exploring alternatives.**

The aim of this project is to build a system that can detect a person within thermal imagery in real time, distinguish a person from other objects and track a person moving through a scene in real time. 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 must then be able to re-identify people based on those our system has already seen, which is done using a siamese Convolutional Neural Network (CNN) to determine if a pair of images contains the same person or two different people.. It will be trained on a set dataset, and then tested on a set of different people. This means that we are aiming to solve the open world re-identification problem, rather than simply to train and test in a closed world on the same set of people, as many previous approaches have [25]. This enables us to deploy our solution on any camera system with any human targets.

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 problem within the field of computer vision. From the previous work, we can see that a substantial amount of research that has been done on person re-identification, mainly revolving around the use of features or attributes of a person [11]. However, much of this relies on the visible spectrum, with attributes of the form “red shirt” [11]. However, in the modern world, thermal imagery is often used for 24/7 surveillance when varying environmental conditions are present. Therefore, it is important that an effective re-identification system is developed to utilise this area of 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 human intelligence organisations. The surveillance data that a system like this can provide would be critical for crime-prevention, forensic analysis, and counter-terrorism activities in both civilian and governmental agencies alike. While this surveillance data is currently 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 better results more efficiently. Figure 1 shows five different views of the same person that our features must be able 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. This had only limited success. Since our work in [19], the attention of the researchers of re-identification has shifted to deep learning based solutions, as shall be discussed in more detail in section 2. 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 it can be applied to any dataset of people. Figure 2 shows the difference between these two possible approaches to deep learning enabled re-identification.

In order 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) tracker, originally proposed by the authors of [10]. The aim here 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, and continue to be labelled as the same person.

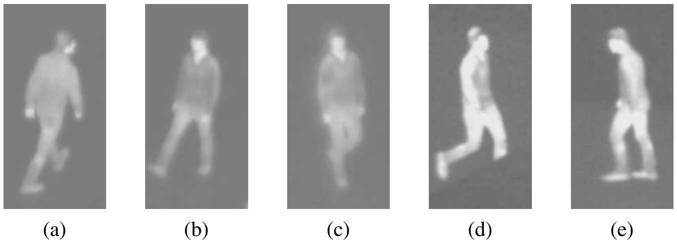


Figure 1: Several images showing the same person that our system must be able to re-identify as the same person

A. Aims of the Project

The intent of this project is to develop a system that can utilise an open world deep neural network, paired with a real time person detection system and TLD tracker, to solve the thermal re-identification problem, and improve on the performance of state of the art solutions, including our own previous work [19].

B. Achievements of the Project

II. Related Work

We will now examine the previous work that has been done relating to the person re-identification problem.

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 these methods 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]. Here, the technique of distance vector routing is employed to get an idea of the relative locations of the cameras, enabling the system to prioritise the people seen most recently by the closest camera, as these are most likely to be correct. This is done by first analysing the overlap between cameras, and then computing distance vectors and probabilities of going from one camera to another, reducing the time complexity of the re-identification process in the majority of 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 they are 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 have gone after leaving a given camera field of view.

The authors of [22] 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 proposed method 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 [23] emphasises the importance of making good use of all images and video frames captured of a target. The system proposed here creates a gallery of images of known individuals, with more images increasing the accuracy of the system. When a gallery exists for a target, this is referred to as multi-shot re-identification, and single-shot re-identification when only one image is available in both the query and the gallery. For multi-shot re-identification, the authors propose to use geometric distance in another way by collaboratively approximating the query sets using all galleries, 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 use to compare people, but also on determining which of these features is the most discriminative for each person individually on the fly. This is achieved through the use of a random forest classifier, and functions well in an open world setting. However, since the time of publication of this work, the state of the art of 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 target, an analysis of their gait and a measure of where the thermal hotspots of each person were. This performed resonably 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], [25], [24], [21], [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 more and more 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 will not be able to follow this approach.

Much of the rest of the previous deep learning work is split into open world and closed world. The work of [25] presents a useful comparison between these two problems, 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 target is, where as in open world, the aim is to verify whether two targets are the same or not. These two approaches differ greatly in terms of input method, feature extraction and the loss function used. The authors of [25] conclude, as we have, that the open world problem is more relevant to real world applications.

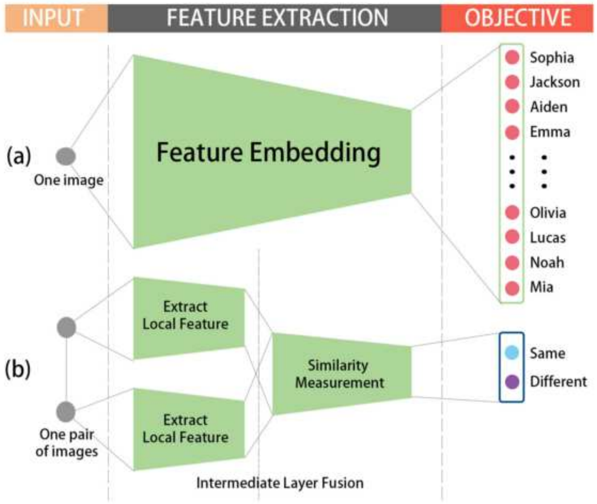


Figure 2: The difference between closed world (a) and open world (b) re-identification systems, from [25]

A solution to the open world problem is attempted in [24]. The authors aim to combine feature learning and re-identification into one framework, which is a deep siamese CNN with an SVM placed on top of the network, after the last layer, 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 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 backed up by the authors of [1], who have implemented architecture and trained on a larger dataset which consistently outperforms the contemporary state of the art.

The work in [21] 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 joint 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. This method outperforms or shows comparable results to the existing best performing methods. The work in [18] is another siamese CNN with the particular aim of making it work for people represented at different scales. This would be an interesting addition to our work, but is currently out of scope for us.

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 will ensure that our network has a state of the art base architecture, taking inspiration from the architectures used in [25], [24], [2], [21], [6] to help us to design the most effective re-identification network possible. We also endeavour to follow the advice of using sufficiently large image resolution, but using a size large enough so that the network has plenty of information to learn from, but small enough so that each person is shown with sufficient detail and clarity. 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 maiximum 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 important on top of the ability to determine whether two images of pedestrians are the same person or not. They argue that 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% form different datasets, differing but complexity and number of cameras.

C. TLD Tracker

The Track-Learn-Detect (TLD) tracker was first proposed by Zdenek Kalal in [10]. The idea of this tracker is to break down the person-tracking task into tracking, learning and detection. The tracker follows the object between frames. The detector corrects the tracker if necessary based on previous observations. The learning 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 as changing background, for example a car driving along a road being recorded from 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 sufficiently capable of tracking a person through a single camera frame, but could not re-identify people across multiple cameras, or even returning from behind an obstruction.

D. Relation to this Project

In this project we use elements from many of the papers discussed in this section. The TLD tracker will be used simply for tracking a person from frame to frame for as long as they remain visible in the same area one camera view. We will disable some of the re-dentification based capability of TLD Whether this person has been previously observed by the system or not will be determined by an open world siamese CNN, with an architecture similar to the related work. We take inspiration from [21] and will extract the features of each individual image before comparison between them. We will 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 see 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 will now break down 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 will begin by opening each video file, or live camera feed, and concurrently running the real time target detection code on these. Each time this code identifies a person, it checks whether it has a TLD tracker object associated with them or not. This is decision point [A] in Figure 3. If not, a new TLD tracker object is created and this person is compared to the other people that have been seen previously using the siamese CNN, and if they are deemed to be sufficiently similar to one of these people, then they are re-identified as the same person, else 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 targets, and are updated each time the target is successfully identified.

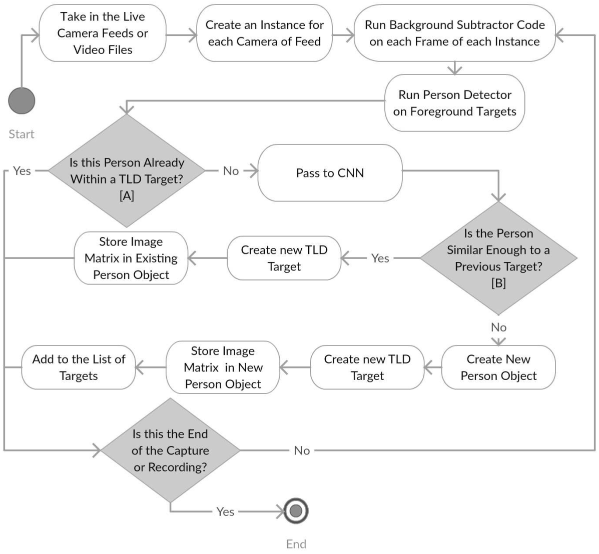

Take in the Live Camera Feeds or Video Files
Create an Instance for each Camera of Feed
Run Background Subtractor Code on each Frame of each Instance
Run Person Detector on Foreground Targets
Start
Is this Person Already Within a TLD Target? [A]
Yes
No
Pass to CNN
Is the Person Similar Enough to a Previous Target? [B]
Yes
Create new TLD Target
Store Image Matrix in Existing Person Object
No
Create New Person Object
Create new TLD Target
Store Image Matrix in New Person Object
Add to the List of Targets
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 neural network for re-identification, and so we can use TLD to track targets through single camera views.

1) Background Subtraction: The first stage of this process is background subtraction from the static camera viewpoint. We use the Mixture of Gaussians (MoG) technique to facilitate this, taking inspiration from [26], [27]. 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 the 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 must 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 HOG descriptor, a high dimensional vector. This HOG descriptor is then passed to a pre-trained machine learning algorithm, in this case a Support Vector Machine (SVM). If this comes up with 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 had a tradeoff between HOG and one of its alternatives, Haar cascades, due to HOG being slower. However, as we are using a TLD tracker this year, there will be far 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 a large impact on the overall performance.

3) Person Tracking using Track-Learn-Detect : 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 originally 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 tracker follows the object between frames. The detector corrects the tracker if necessary based on previous observations. The learning estimates detector's errors and updates it to avoid these errors in the future. The relationship between these components is shown in Figure 4.

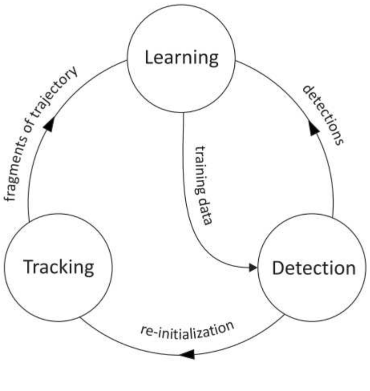


Figure 4: Block diagram of the TLD framework, from [10]

In our early experimentation, we found that the learning part of this algorithm meant that it was attempting to perform re-identification when people left and re-entered a video stream, or appeared in a different one. Therefore, we had to 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 missclassifications that the limited re-identification capabilities of this algorithm were causing, enabling us to effectively track people across the scene, whilst using our neural network for the re-identification.

Consequently, in our this 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 in this neighbourhood, and classified by the neural network. 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 indicated 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 in the next frame when HOG detects this person again. Figure 5 shows the TLD system in action following a human target through a frame, as well as displaying the positive and negative patches 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.

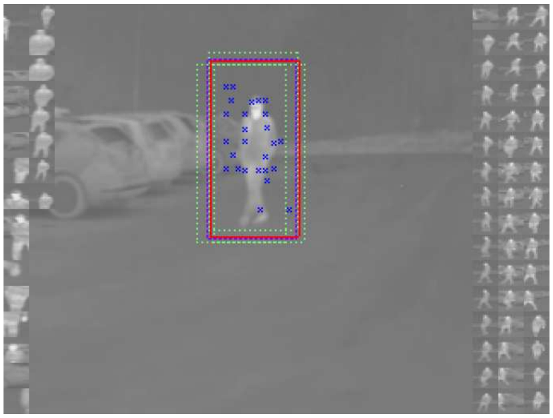


Figure 5: An example of TLD tracking a person through a scene. The images on the left are negative patches, and the images on the right are positive patches

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 258x128, 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 and enable it to cope better with difficult real world situations. This data is split 60:20:20 into training, validation and testing sets respectively. The purpose of the training set is to be the actual data that we use 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 a 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 down-sample their input, looking at a neighbourhood of pixels and, in the case of max pooling that we have used here, output the maximum. 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.

The loss function to be used for this network was proposed by the authors of [9], and is given by Equation (1) below. 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) below, where Y is the label (0 or 1), m is the margin that determines the 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) below, 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 2D convolution window. For the poolling layers we experimented with the pool size, which gave the factors to downscale the input by. 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.

The reason that ReLU was chosen as our activation function is that it is a good approximator, and can be used to approxmate 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 exactly what function we are trying to learn for effective re-identification in thermal imagery as a consequence of our previous work in [19].

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 they are in figure 7(a), with the cameras labelled as camera α, β and γ respectively, and their fields of view shown by the matching

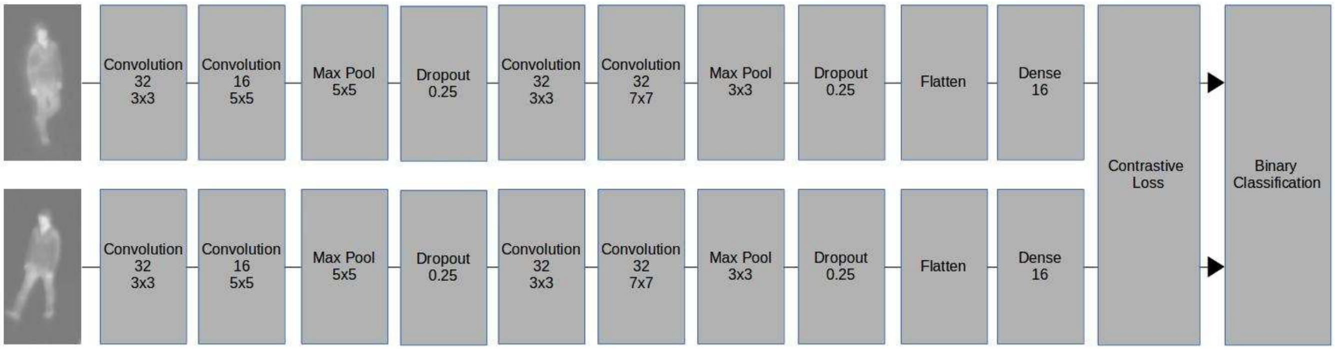
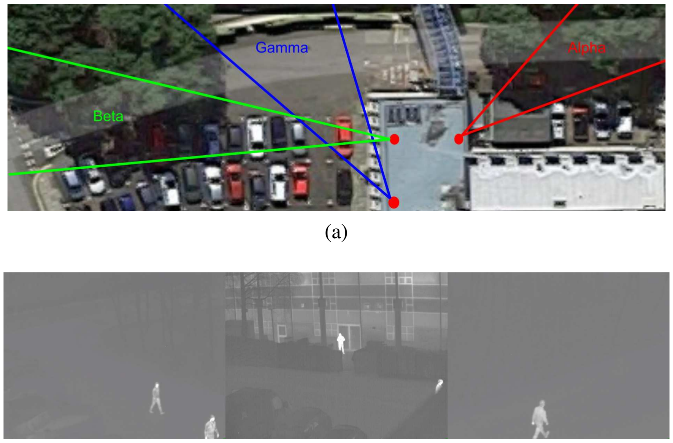


Figure 6: The architecture of our CNN

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 will 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 far fewer missclassifications from HOG as it is being called far fewer times 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 missclassifications. 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 the side than from the front. This inferior performance is also due to the fact that many of the people present in camera γ are partially obstructed by other objects, leading to the TLD objects being deleted.

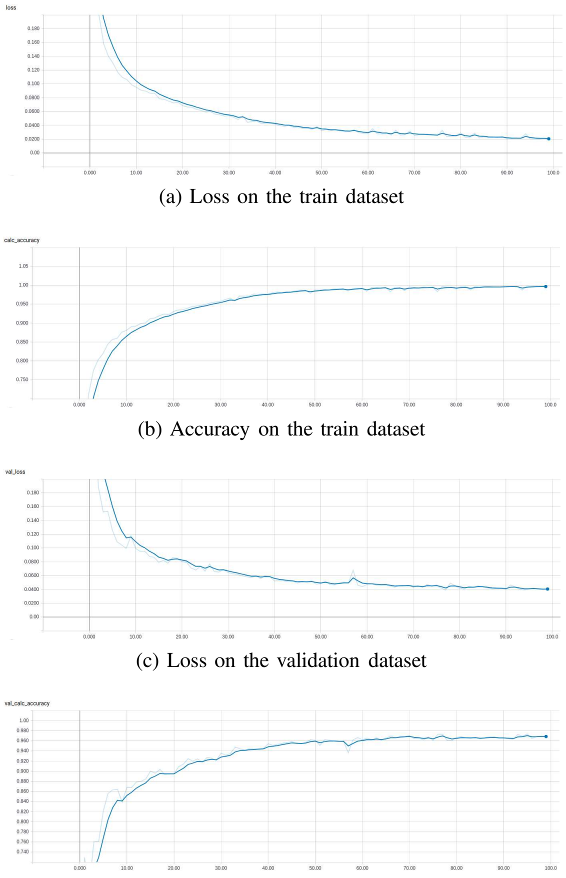
Camera	Frames with person present	Calls to CNN
Alpha	1259	65
Beta	1863	114
Gamma	2454	339

Table I: Performance gains from TLD

C. The Deep Learning System

When designing our network we did many training runs to determine the best architecture, and then performed extensive grid searches to determine the best parameters. While training the network, the metrics that we were mainly concerned about were the loss, from the contrastive loss function defined in Section 3, 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 8 shows how the loss and accuracy varied on our training and validation sets as the network was trained over 100 epochs.

Figure 8 shows use that the training and validation patterns of the network matched very closely, with the loss steadily decreasing and the accuracy steadily increasing before both flattened out towards the end of training. The use of 100 epochs gave both the loss and the accuracy time to stabilise, and ensure that the results we got for these, and the trained network as a whole were the best that could be attained.



(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 8: Loss and Accuracy results during training of the network over 100 epochs

Table II shows the confusion matrices of the network when predicting all three datasets. 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 datasets, reporting the accuracy, which is the percentage of the dataset that was 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 results from these are encouraging, as all are high percentages across all datasets, giving us confidence that our system is working well and has learnt useful features for re-identification. The fact that precision and recall are close together means that the numbers of false positives and false negatives are fairly close, as shown by Table II. Both missclassifying different people as the same person and missclassifying the same person as different people are destructive to our full re-identification system, as this can lead to corruption of a persons data in the first case, or create unnecessary new targets in the second case. Therefore, the fact that all of these metrics get good results means that we have not sacrificed proficiency in one of these cases for the other.

		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%

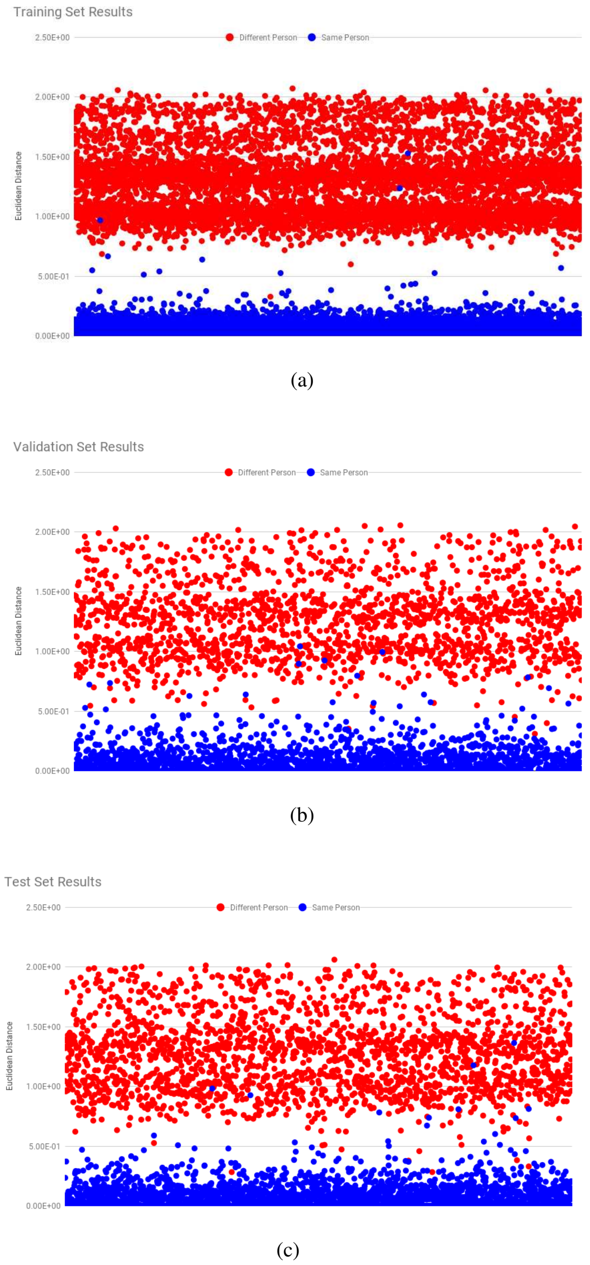
(b) Classification report for validation set

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 9 is a graphical representation of the Euclidean distances reported by the network on the train, validation and test datasets. These graphs show a clear split between the pairs of images of the same person and pairs of images of different people. The split 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.

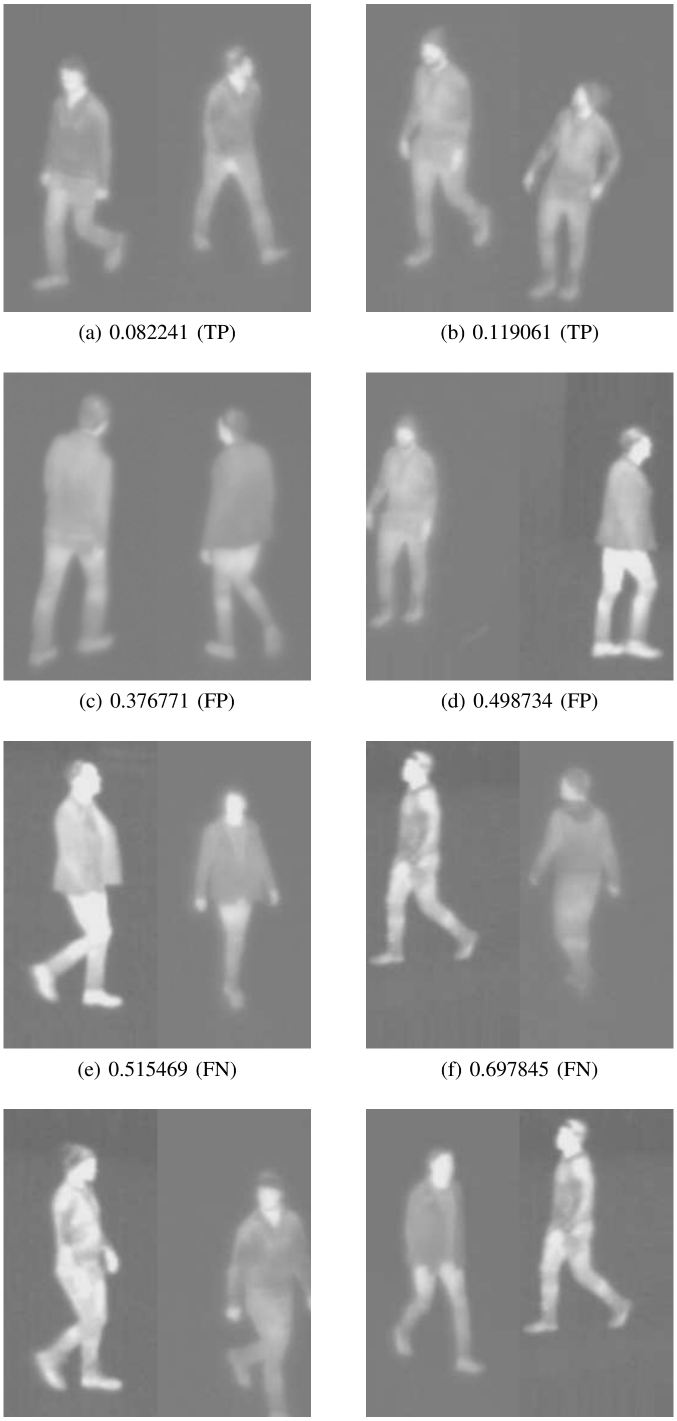


(a)

(b)

(c)

Figure 9: Euclidean distance results of the CNN on the Train, Validation and Test Datasets



(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 10: Image pairs and their Euclidean distances

It is interesting at this point to note which of the image pairs were the missclassifications, and also to establish which were very close and which were very far apart. Figure 10 shows a few examples of these images, with their Euclidean distances in the captions. As before, those with a distance less than 0.5 were classified as the same person, while those with a distance greater that 0.5 were classified as different people.

We also ran this on an alternative dataset, the dataset used for our work in [19]. The cameras for this dataset are placed in different locations to our main dataset, and the profiles of the people are different. The results that the network gave for this dataset are shown below in Table IV , Table V and Figure 11. Figure 12 shows a few examples of these images, with their Euclidean distances in the captions.

		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

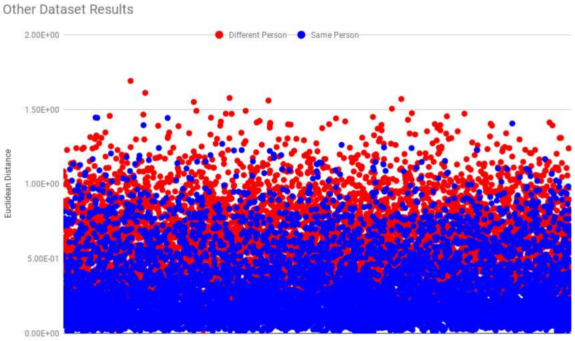
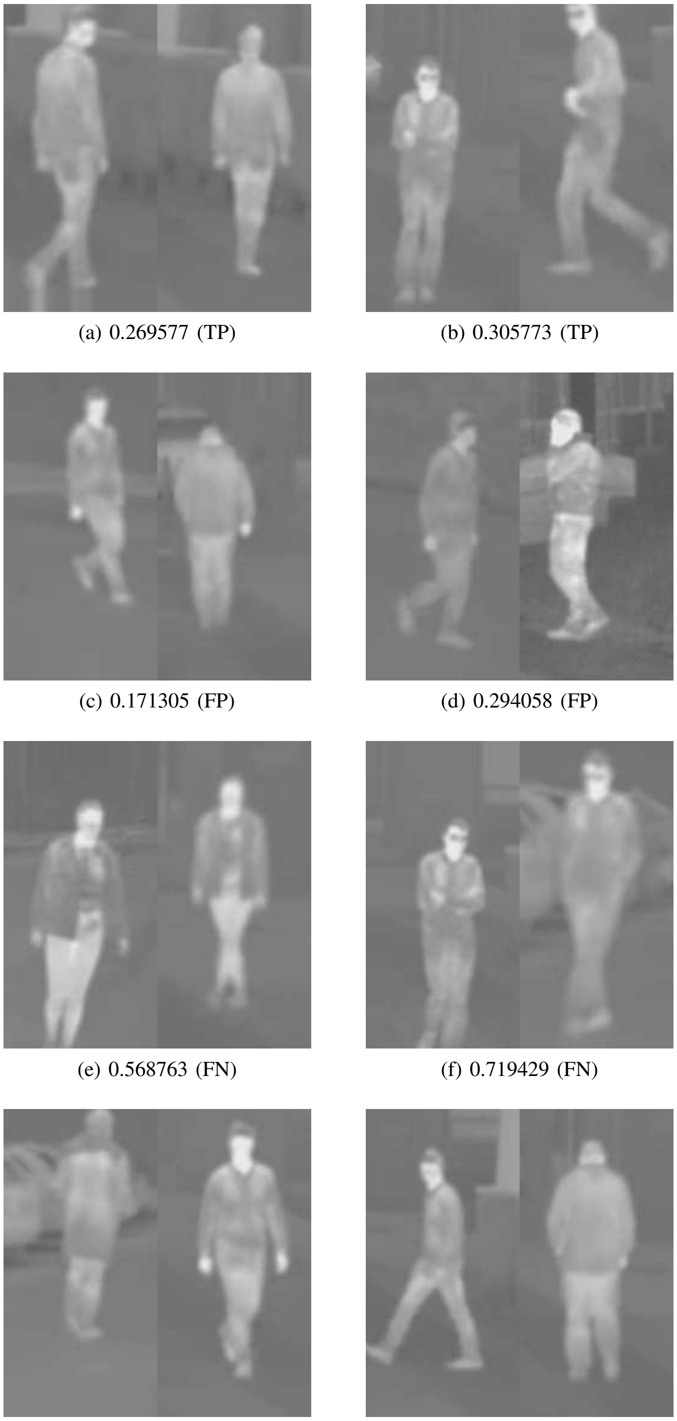


Figure 11: Euclidean distance results



(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 12: Image pairs and their Euclidean distances

As we can see from these results, the performance of the system on this dataset is substantially worse. As we can clearly see, the images in Figure 12 are higher resolution. This tells us that while our network performs very well on the camera views that it was trained on, its ability as a general re-identification system on any camera network is limited. We hypothesise that this could be solved by using an optical zoom to ensure that all video streams give us images of person targets with similar resolutions.

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 dataset. First, we consider the performance of the system when run on a single camera view. This enables use to simply and effectively test and evaluate our re-identification system on one camera viewpoint to remove the potential missclassifications that could result from examining all of the camera views at once.

E. Multi-Camera Re-Identification System

We now move on to running the system on all of the camera views at once.

V. Conclusion

A. Appraisal of Project Organisation

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. Despite this issue, and our aims were accomplished, enabling us to contribute valuable information to the state of the art.

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] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

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

[27] 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.