Person Re-Identification using Deep Convnets with Multi-task Learning

McLaughlin, N., Martinez del Rincon, J., & Miller, P. (2016). Person Re-Identification using Deep Convnets with Multi-task Learning. IEEE Transactions on Circuits and Systems for Video Technology. DOI: 10.1109/TCSVT.2016.2619498

Published in:
IEEE Transactions on Circuits and Systems for Video Technology

Document Version:
Peer reviewed version

Person Re-Identification using Deep Convnets with Multi-task Learning

Niall McLaughlin, Jesus Martinez del Rincon, and Paul Miller

***Abstract*—Person re-identification involves recognizing a person across non-overlapping camera views, with different pose, illumination, and camera characteristics. We propose to tackle this problem by training a deep convolutional network to represent a person's appearance as a low-dimensional feature vector that is invariant to common appearance variations encountered in the re-identification problem. Specifically, a Siamese-network architecture is used to train a feature extraction network using pairs of similar and dissimilar images. We show that use of a novel multi-task learning objective is crucial for regularizing the network parameters in order to prevent over-fitting due to the small size the training dataset. We complement the verification task, which is at the heart of re-identification, by training the network to jointly perform verification, identification, and to recognise attributes related to the clothing and pose of the person in each image. Additionally, we show that our proposed approach performs well even in the challenging cross-dataset scenario, which may better reflect real-world expected performance.**

***Index Terms*—person re-identification, deep learning, neural networks, feature embedding**

I. Introduction

Person re-identification models the problem of tracking a person as they move through a non-overlapping camera network. Reliable person re-identification is vital for multiple camera tracking in realistic conditions, where there is little control over the image acquisition process, where camera coverage may be sparse, and where the subject may be non-cooperative. The core of this problem is to decide whether pedestrian detections from several non-overlapping cameras, acquired at different times, were all caused by the same person. In the most general case, the individual cameras will have different hardware, will have non-overlapping fields of view, will capture the person from different angles, with different pose, and with differing illumination. Due to the large number of uncontrolled sources of variation, as well as generally poor image quality, this task remains very challenging.

Current approaches to the re-identification problem are based on either extraction of features that are invariant to the expected sources of variation [30], [29], or use of a supervised learning algorithm to discover the most relevant information for matching images [57], [67]. Each of these approaches has its own drawbacks: Invariant feature extraction algorithms may be designed based on intuition [15] or using a physics based model [30] of the predicted variation. As both physically based models and hand-crafted feature extraction algorithms rely on human judgement, they may be unable to capture some of the subtle, but potentially important, aspects of the data. Similarly with learning based approaches that rely on pre-established features, a large number of parameters may have to be learned, leading to over-fitting, due to the small size of the training sets available for this problem [17], [65]. Their performance may also be strongly related to the chosen pre-established features and not be reproducible with other feature types and/or datasets, reducing their generality. In this paper we will address the person re-identification problem using a deep-learning based approach. In contrast to most existing approaches, we propose to combine invariant feature extraction and supervised learning into a single unified framework based on deep convolutional neural networks, trained to specifically address the person re-identification problem. We make use of several techniques, including multi-task learning, data-augmentation, and dropout to reduce the risk of over-fitting.

Our proposed system uses the Siamese network architecture [19], [6], to train a deep convolutional network to extract features useful for person re-identification. This architecture trains a neural network to produce a low-dimensional feature representation of the input images, where diverse images of the same person are mapped onto similar locations in the feature space, while images of different people are mapped onto different locations in the feature space. The network is trained using a diverse set of images, helping it to learn the subtle cues that indicate whether images depict the same person or not, which would be virtually impossible to discover using hand-crafted features.

This paper differs from other re-identification methods based on deep networks [61], [37] by inclusion of multi-task learning to improve re-identification performance and prevent over-fitting to the training dataset. In particular, the multi-task network will be trained to perform verification, identification, clothing attribute labelling, and pose labelling. This aims to encourage the learned feature representation to better generalize to unseen data such as other re-identification datasets. We show that by using multi-task learning, the re-identification performance of a simple convolutional network can match that of the more complex network, such as [61], given the same training and testing datasets. Our approach may therefore have computational efficiency advantages at run-time, due to the smaller number of parameters. In addition, our multi-task learning approach is general enough that it could be applied to the training of more complex networks to further improve their performance. Finally, as a side effect of the multi-task training procedure we obtain accurate full-body classifiers for gender, pose, and clothing attributes, which have useful applications independent of person re-identification.

II. Related Work

Person re-identification is typically modelled as a verification problem that involves deciding whether images from disjoint cameras depict the same person or not. Traditional approaches to re-identification typically involve extracting invariant features [53], [56], [29], [8], [15], optionally followed by metric learning [22], [21], [57], [27]. In practice it can be challenging to design a complete re-identification system that fulfils all the desirable, and somewhat conflicting criteria, because of the difficulty involved with separating factors related to identity from those related to other causes.

A. Invariant feature extraction

Ideally, the features used for re-identification should be invariant under common image transformations, while having a high degree of inter-person variation, and a low degree of intra-personal variation. Colour is commonly used as it exhibits a degree of pose invariance [53]. However, colour features, especially RGB or HSV, tend not to be illumination invariant [56] or invariant to different camera set-ups. Attempts have therefore been made to use physical illumination models, such as the Retinex model [30], to understand how colour features are affected by illumination to improve their invariance. Brightness transfer functions (BTF) can be used to transform colour features as a person moves between a pair of cameras [46], [26], however the transfer function may need to be relearned for each camera pair if the illumination changes significantly, limiting their value for real world applications. The intra-distributional structure of colour features, which may remain invariant even after illumination change has been exploited by [29]. Texture and shape features can also be used, such as in the SDALF method [15], which uses prior knowledge of the bi-lateral symmetry of human appearance to extract robust colour and texture features, and omit extraneous variation. Pictorial structures, which isolate specific body parts, have been used to segment the person from the background, in order to more reliably measure colour and texture features [8].

Hand-designed features may not take full advantage of the information contained in the training images and are labour intensive to develop. Therefore supervised learning approaches have been developed that distinguish between the features and variations likely to be related to identity, and those likely caused by unrelated factors. Examples of supervised learning systems that have been used for person re-identification include the Ensemble of Localized Features (ELF) approach [18], which uses an Adaboost classifier combining many simple classifiers, to select the features that most discriminate between different people. A related approach learns to represent different body regions using different features [3]. The features from all the different body regions are then combined together to provide strong discrimination among people. An approach based on learning invariant colour features in small image patches, which makes used of sparse coding and an auto-encoder neural network is presented in [56] Dictionary learning has been used by [35] to learn patch representations that are constant across differing views in a supervised learning setting.

B. Metric Learning

Metric learning encompasses a family of supervised learning methods that use a Mahalanobis distance metric to compare features while emphasising inter-personal differences and de-emphasising intra-personal differences. The simplest such approach is Linear Discriminant Analysis (LDA), however enforcing different constraints can give better performance [22], such as transferring the optimisation problem into the information-theoretic setting [11]. Metric learning can be applied in a single or multiple shot setting [21], depending on the number of example images of each person. Methods such as Relaxed Pairwise Learning (RPLM) [22], which uses similarity and dissimilarity constraints, have demonstrated that high re-identification accuracy can be achieved using only simple colour and texture features, in conjunction with a suitable metric learning algorithm. Other related metric learning approaches include Large Margin Nearest-Neighbour (LMNN) [57], which has been adapted to the re-identification problem by inclusion of rejections [13]. Equivalence constraints, based on similarity and dissimilarity, are used in [27] which makes efficient use of sparse labels. The relative distance between image pairs is used by [67] to produce a distance metric, while attempting to avoid over-fitting. An explicit polynomial kernel feature map is used in [7] to compare the similarity of all patches in two images, which produces a feature used as input to a mixture of linear similarity functions. Metric learning and deep learning are combined by [24], which uses hand-crafted features as input to a deep-network that learns a non-linear local metric to compare images. Prior knowledge of the re-identification problem is used by [39] to cope with illumination changes and to extract low-level features, before using the features with a subspace metric learning method. It is also possible to learn verification decision function together with a distance metric to improve performance compared with a fixed verification threshold [38]. Another method that can be used to learn a distance metric is Canonical Correlation Analysis (CCA) [40]. CCA is used in conjunction with reference descriptors in [2], to achieve highly accurate re-identification given only simple features. Person re-identification can also be cast as a ranking problem, where the ranking function is learned using a maximum margin framework [47]. In contrast to the above supervised methods, side information, which can be collected in an unsupervised manner and indicates that certain examples belong to the same class, can be used by the relevance component analysis (RCA) method to learn a Mahalanobis metric [5]. Additional information, such as depth information, can be used with a modified version of information theoretic metric learning [11] to improve re-identification performance [59]. The fact that people move through camera networks has been used to learn multiple related Mahalanobis distance metrics between camera pairs [42], however, this approach required knowledge of the camera network layout and different training for each camera pair. The main drawback of the above metric learning approaches is their tendency to over-fit, due to the small size and high variability of the available re-identification datasets compared with the large number of parameters that must be learned.

C. Deep Convolutional Neural Networks

There has been renewed interest in using neural networks for computer vision, sparked by the significant performance improvements over previously *state of the art* methods recently achieved using deep convolutional networks (CNNs) [28]. An application of neural networks that is particularly suited to person re-identification is that of learning *embeddings*, which involves mapping images into a low dimensional feature space, while preserving semantic relationships between the images [19]. For example, the 'Siamese network' [19], [6] can learn to map visually different images of the same person to similar locations in feature space, and map images of different people to distant locations in the feature space. This requires the network to learn to discriminate between the identifying information and unimportant background variation.

Several other deep learning methods have been tried for person re-identification, such as using triplets rather than pairs of images to enforce the similarity and dissimilarity constraints between images [14], or using a new type of neural network layer to directly compare the appearance similarity between different image regions [1].

The standard Siamese network architecture has previously been used for person re-identification [61], however as we will show (see Section V-A), over-fitting can be an issue when this approach is used alone. Multi-task learning, where a network is trained to complete several related tasks in addition to the main problem of interest, has previously been used to address over-fitting. Multi-task learning may encourage the network to learn a more robust internal feature representation [10]. It has been used to improve the performance of networks for tasks such as face key-point recognition, face alignment [63], and facial verification [51]. This will be combined with the notion that, given accurate recognition of attributes, such as clothing type or sex, which do not vary under changing illumination, person re-identification could be performed without the need for low-level features, such as texture and colour, which are likely to vary with environmental conditions [31].

There are also several re-identification methods that make use of attributes. Such as [32] which directly uses predicted attributes as a feature to perform re-identification. However this method does not take advantage of the fact that multiple tasks are learned simultaneously. The correlations between attributes are taken into account by [50], which also combines attribute labels with low-level features for re-identification. Although this approach achieves good performance, it uses hand-designed image features and several independent classification components. This makes the approach difficult to fully optimise as the different components are trained independently.

In this work we propose to combine the Siamese network architecture, used in [61], with multi-task learning as the basis of our approach to tackling the person re-identification problem. Although verification and identification tasks were combined in a Siamese network architecture for face recognition [51], as far as we know, this work is the first to apply this method to full-body person re-identification. Our novel multi-task learning framework will learn the embedding function making use of similar and dissimilar image pairs, while simultaneously training the network to perform identification of each person in the training image pairs. Furthermore, we extend the multi-task learning approach of [51] by training the network to perform a diverse set of attribute labelling tasks, based on pose, sex, and clothing. Use of a diverse set of related tasks, unlike the single repeated task of [42], aims to improve the generalisation performance of re-identification networks and help to prevent over-fitting to a particular training set or camera layout.

III. Person Re-Identification Architecture

A. Person Re-Identification using Neural Networks

The conventional Siamese network architecture (See Fig. 1) consists of two identical copies of a sub-network G i.e., each sub-network has identical weights. During training, the network is presented with image pairs from either the same or differing classes. Given image pair (x_1, x_2), and label $y \in 0, 1$, indicating whether the images in the pair are from the same or different classes, a forward pass of each image through sub-network G, with network parameters w (for notational simplicity we will use w throughout to represent all the network parameters i.e., the weights and biases), produces vectors $G(x_1; w)$ and $G(x_2; w)$, which are low-dimensional feature representations of the respective input images. The Euclidean distance between the feature representations can then be computed as

$$D(x_1, x_2; w) = \|G(x_1; w) - G(x_2; w)\|_2 \tag{1}$$

The Euclidean distance between the feature representation of each image is used for training the network to perform verification. For a training image pair (x_1, x_2), the cost function, $\mathcal{V}$, is dependent on whether the images are from the same, or different people. We will first introduce the cost functions for both cases, then we will show how these cost functions can be combined. When x_1 and x_2 are images of the same person, the cost function can be written as follows:

$$\mathcal{V}_S(x_1, x_2; w) = \frac{1}{2} D(x_1, x_2; w)^2 \tag{2}$$

Therefore, in the same person case, the cost increases as the Euclidean distance between the feature representations increases, and when the feature representations are identical, the cost is zero. In the alternative case, when x_1 and x_2 are images of different persons, the cost function can be written as follows:

$$\mathcal{V}_D(x_1, x_2; w) = \frac{1}{2} (\max(0, m - D(x_1, x_2; w)))^2 \tag{3}$$

In this case the cost increases as the Euclidean distance between the feature representations decreases. In Eq. 3 the variable m is known as the margin. When the distance between the feature representations of both samples is greater than the margin, the cost is set to zero. The margin therefore encourages the network to concentrate on difficult cases where

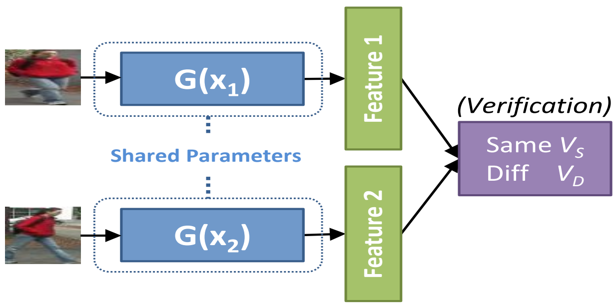

Fig. 1. Standard Siamese network architecture that can be trained to perform verification, when presented with image pairs (x_1,x_2) from the same/different class, using the cost function Eq. 4.

images from different classes produce similar features, thus helping the network to better discriminate between similar and non-trivially dissimilar images.

The above defined cost functions, $\mathcal{V}_S$ and $\mathcal{V}_D$, can be combined into a single cost function, $\mathcal{V}$, capable of handling both cases. Given a label $y \in 0, 1$, where $y = 1$ indicates that the two images belong to the same class and $y = 0$ indicates the images belong to different classes, the cost function can be written as follows

$$\mathcal{V}(x_1, x_2|y; w) = (1-y)\mathcal{V}_S(x_1, x_2; w)+y\mathcal{V}_D(x_1, x_2; w) \quad (4)$$

Given Eq. 4, which is the cost function for a single image pair, and given the full training set X, consisting of I image pairs $(x_1^i, x_2^i) \in X$, where each image pair has a corresponding verification label, $y^i \in \{0, 1\}$, the cost function over the whole training set can be written as

$$\mathcal{C}(X) = \sum_{(x_1^i,x_2^i),y^i \in X} \mathcal{V}(x_1^i, x_2^i|y^i; w) \quad (5)$$

Cost function Eq. 5 can be minimized using stochastic gradient descent, by finding $\frac{\delta C(X)}{\delta w}$, the gradient of the cost function with respect to the parameters w for each of the training pairs. The implementation details of sub-network G will be discussed in Section IV.

B. Multi-task Learning

Given the small size of data-sets available for person re-identification, over-fitting is a serious concern. Over-fitting is a problem encountered when training large neural networks (or any learning algorithm with a large number of parameters) on limited training data, and is characterised by a network that performs well on the training examples (or well only on a particular dataset), but performs poorly when presented with novel examples not seen during training. To address the risk of over-fitting, and hence to improve the generalisation performance of the re-identification network, we propose to modify the network architecture described in the previous section to include multi-task learning, which will act as a regularization method. Multi-task learning involves training a network to complete several auxiliary tasks in addition to the main problem of interest, and has been shown to improve performance, if the axillary tasks have been chosen to complement the main learning problem [10].

Person re-identification can be considered a verification problem where the goal is to decide if two images depict the same or different persons. To perform verification the two images are compared, and if their similarity score is greater than a pre-specified threshold the images are considered to depict the same person. The cost function used for training the network described in Section III-A, was designed for verification, i.e. to produce a high similarity score for images of the same person, and a low similarity score for images of different people. To add multi-task learning to this network, we assume that in addition to the verification task, we now have K additional auxiliary tasks, where each auxiliary task has an associated cost function $\mathcal{T}_k$. As before, we are given training set X, consisting of I image pairs, $(x_1^i, x_2^i) \in X$, where each image pair has a corresponding verification label, $y^i \in \{0, 1\}$, indicating whether the images depict the same or different persons. For multi-task learning, the individual images in each pair are now also associated with task specific labels, $l_1^{i,k} \in L_k$ and $l_2^{i,k} \in L_k$, where each $l^{i,k}$ is the ground-truth label for a particular auxiliary task, and where L_k is the set of all labels available for task $k \in K$. The cost function $\mathcal{C}$ defined in Eq. 5 can now be modified to include multi-task learning as follows

$$\begin{aligned}\mathcal{C}_m(X) = &\sum_{(x_1^i,x_2^i) \in X} \mathcal{V}(x_1^i, x_2^i|y^i; w) + \\ &\sum_k \alpha_k \mathcal{T}_k(G(x_1^i)|l_1^{i,k}; w) + \sum_k \alpha_k \mathcal{T}_k(G(x_2^i)|l_2^{i,k}; w)\end{aligned} \quad (6)$$

For a given training image pair, multi-task learning is performed on both images. Each task has corresponding weight α_k allowing different tasks to be assigned different importances. Note that all the auxiliary tasks operate on the abstract feature representation, $G(x)$, of the input image, rather than on the raw input image itself. This is because the goal of using multi-task learning is to encourage the network to learn an abstract feature representation, $G(x)$, with better generalisation properties than one learned using only the verification error signal. The above cost function, $\mathcal{C}_m$, can be minimized using stochastic gradient descent, by calculating the gradient of $\mathcal{C}_m(X)$ with respect to the network parameters w.

Depending on the problem of interest a wide variety of auxiliary tasks can be used to complement the main learning problem. In our case, all the auxiliary tasks involve assigning one of several mutually exclusive labels to each training image. Therefore the softmax regression cost function can be used, where the number of labels is customised to each task. Softmax regression is a multi-class linear classifier that calculates the probability of its input belonging to each class i.e., its probability of having a specific label. For task k, with associated label set, L_k, which contains the ground truth label, $l^k \in L_k$, the softmax regression cost function can be defined as follows

$$\mathcal{T}_k(z|l^k; w) = \sum_{j \in L_k} 1\{l^k = j\} \log \frac{e^{w_j^T z}}{\sum_{q \in L_k} e^{w_q^T z}} \quad (7)$$

Where z is the input to the softmax function, which in our case is $G(x_1^i)$ or $G(x_2^i)$ i.e., the feature representation of the input image, and where $1\{l^k = j\}$ is an indicator function that takes the value one when the prediction, $j \in L_k$, is equal to the ground truth label l^k, and takes the value zero otherwise. The size of the label set, L_k, is customised for each specific task, and each label is associated with an individual decision boundary, w_q, learned during training. The denominator on the right hand side of Eq. 7 normalises the distribution over all possible labels $q \in L_k$. The final predicted label is the maximum likelihood class output by the softmax classifier. This general framework allows incorporation of any task into our multi-task network, in the following section we will describe the individual auxiliary tasks in more detail.

1) Identification: The identification task is used to complement the main verification task. To perform identification, the network must predict the identity of the person in each training image. A closed set of persons is used for training, and the identity associated with each training image is known. By inclusion of an identification task, in addition to verification, the network is encouraged to learn an abstract feature representation capable of encoding the appearance information specific to each person, while also fulfilling the verification objectives. In order to perform the identification task, the network predicts a label for each image using a softmax classifier, which takes the abstract feature representation, $z = G(x)$, of each image as input as follows

$$\mathcal{T}_I(G(x)|p;w) = \sum_{j \in P} 1\{p = j\} \log \frac{e^{w_j^T G(x)}}{\sum_{q \in P} e^{w_q^T G(x)}}, \forall x \in (x_1 \cup x_2) \quad (8)$$

where the true identity, $p = j$, is known, and where the number of labels in the softmax function is equal to the number of people in the training set, $P = I$.

2) Attributes: Each training image depicts a person wearing several known articles of clothing. For the attribute labelling task, the network must predict the presence or absence of each article of clothing included in the attribute set, in addition to predicting the sex of the person. A complete list of the attributes to be predicted is shown in Table I. Two different methods of predicting attributes were tested: firstly equally weighting each attribute, and secondly using different weights for each attribute, with weights either learned from data or based on prior information.

Each attribute labelling task was performed independently by a separate softmax classifier. The feature representation, $G(x)$, of each image, was used as input to every softmax classifier. All the softmax classifiers are defined in the same way as Eq. 7, where the label set L_k is of size two, indicating the presence or absence of each attribute. Where weighted attribute classification tasks were used, the weights were scaled so that the sum of all weights was one. The option of predicting a vector of all attribute labels jointly using regression loss was also tested for comparison. The effect of the different attribute weighting and prediction methods on re-identification accuracy will be discussed in Section V-E1.

3) Pose: The unconstrained nature of person re-identification means the subject may be facing at an unknown angle with respect to the camera. The pose identification task asks the network to predict the direction, θ, a person is facing with respect to the camera. A softmax classifier $\mathcal{T}_{\theta k}(G(x)|\theta;w)$ which takes $G(x)$ as input is used for this task. Due to the ambiguity in identifying the precise angle a person is facing, pose is discretized into several bins with centre angles of $\theta = [0, 45, 90, 180, 270]$ degrees with respect to the camera. We note that although pose, by itself, cannot be used for identification purposes, training the network to perform this task may help it to discover features which are useful for modelling human appearance, and which may therefore be indirectly helpful for re-identification.

Network Architecture for Multi-task Learning: We propose a network architecture using a separate softmax classifier for each auxiliary task: identification, pose labelling, and each attribute labelling task. In this architecture, the identification and pose labelling tasks have weight, $\alpha_I = \alpha_\theta = 1$. When equal weighting is used, each attribute labelling task has weight, $\alpha_a = 1/A$, where A is the number of attributes. When independent weights are used, each task has weight, $\alpha_a = w_a / \sum_j w_{a_j}$, where w_a is the weight associated with a specific task. This weighting is intended to prevent the attribute labelling tasks from dominating the cost function, which could reduce re-identification performance. The network architecture after inclusion of multi-task learning is shown in Fig. 2.

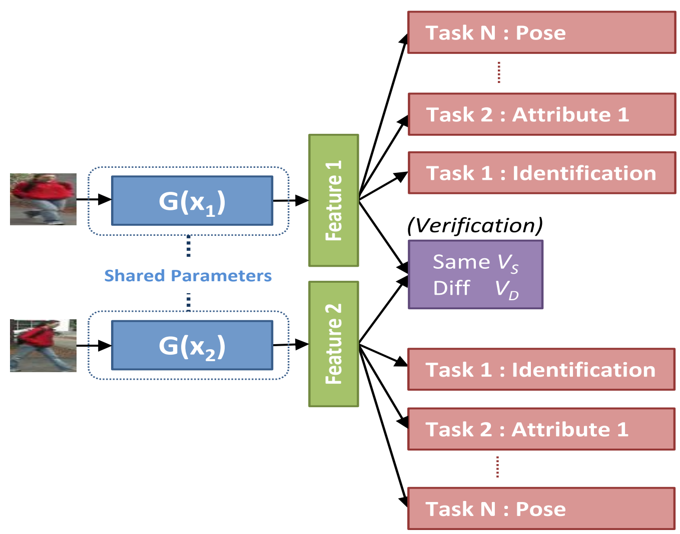

Fig. 2. The architecture of the Siamese network including multi-task learning. The network performs verification using Eq. 4, and auxiliary tasks 1 to N, including identification, performed using softmax classifiers (See Eq. 8), and attribute classification and pose labelling tasks using individual softmax classifiers for each classification task (see Section III-B2).

IV. Re-identification Framework Implementation

A. Individual Convolutional Networks

The parallel copies of sub-network, G, used for learning the mapping from images to the feature space, are implemented as convolutional networks [33]. The convolutional network architecture takes advantage of the stationarity property of natural images i.e., for a large set of natural images, the statistics for the set of image patches at any given location are invariant [44], [33]. This property allows sharing of network weights between image areas, significantly reducing the total number of parameters that must be learned. In practice, each layer of a convolutional network learns several small filters, which are convolved with the layer's input i.e., the previous layer's activation maps, to produce a new set of activation maps. Note that the filters in the first convolutional layer are connected to the colour channels of the input image. The activation maps are typically passed through a non-linear activation function, such as hyperbolic tangent, before further processing. Finally, a pooling operation, such as max-pooling [25], which takes the maximum response within a small window, is applied to the activation maps to reduce their dimensionality and to provide a small degree of translation invariance. Note that, while the network weights can be learned using back-propagation, the hyperparameters such as the number of convolutional layers, the size of the convolutional filters in each layer, and the layer widths i.e., the number of convolutional filters per-layer, are usually set by selecting the values that maximise the network's accuracy on a set of held-out validation data.

The overall architecture of the convolutional network G used for re-identification in our approach is shown in Fig. 3. This network is composed of repeated convolutional and pooling layers, followed by a final fully connected layer that acts as the output. The hyperbolic tangent activation function was used between each convolutional layer, while a linear layer was used between the final convolutional layer and the fully connected layer. The activation of neurons in the fully connected layer gives the feature representation of the input image. Dropout regularization [20] was used between the final convolutional layer and the fully connected layer. Note that no pre-processing was applied to the input image other than converting to the YUV colour space and normalizing each colour channel to have zero mean and unit variance.

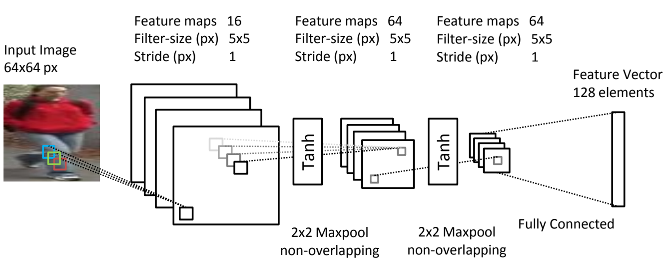

Fig. 3. Overview of the architecture of the convolutional network $G(x)$.

B. Data augmentation

Many current re-identification datasets are small, in terms of number of persons, and number of examples per-person. For example, VIPeR [17] contains 632 persons, with two images per person; each image captured by a different camera. Following the standard testing protocol, each dataset is split into 50% of persons used for training and 50% for testing. Due to the small size of many available datasets, in comparison to the number of network parameters to be learned (around 780,000), over-fitting is a serious concern. Therefore data-augmentation [28] was used during training to artificially increase the size and diversity of the training set.

Input images were resized to 64x64 pixels, as our convolutional network implementation is optimized for square images. During training each image was presented to the network multiple times. At each presentation the image was subject to random horizontal flipping with 50% probability, to help the network learn invariance to the direction a person is facing. In addition, rather than passing the whole 64x64 image to the network during each training step, a random 56x56 pixel crop was selected with uniform probability from the set of all 64 possible crops, for use as input. This helps the network to learn invariance to small translations, which could occur in a realistic scenario due to pedestrian detector inaccuracy. By combining random horizontal flipping with random cropping, the number of image pairs available for training the network is increased by 128 times. Although these additional images tend to be highly correlated, it is known that this type of data augmentation can improve the performance of convolutional networks [28], especially when training data is very limited. We will show in Section V-C3 that use of data augmentation is critical for obtaining good performance from our system.

C. Re-identification Testing Architecture

During testing the multi-task learning blocks were deactivated as their main functionality is to regularize the training process. Sub-network G was used alone to produce the feature representation for each image, consisting of the activations of the neurons in the final layer of G i.e. a low-dimensional vector of real numbers.

To improve re-identification performance, data augmentation was used during testing, as in [23]. To calculate the similarity-score between a given testing image and a given gallery image, features were extracted from 10 samples (the four corner crops, the centre crop, and their horizontal flips) of the testing image and gallery image. The similarity-score was then calculated as the mean Euclidean distance between the features of all one hundred test/gallery image pairs, where a smaller distance indicates two images are more similar.

The overall re-identification procedure was as follows: Given each testing image from camera A, and a gallery of 316 images from camera B, the camera A testing images were compared with all the gallery images using Euclidean distance, as described above. The gallery images were then ranked according to similarity and a CMC curve [15] was produced for evaluation purposes.

V. Experiments

Several re-identification datasets were used to evaluate the proposed system: VIPeR [17], iLIDS [65], 3DPeS [4], CAVIAR [8], and PETA [12]. The VIPeR and PETA datasets were used in the experiments focused on multi-task learning, due to the availability of attribute labels. Attribute and pose labels for VIPeR are provided by [32]. PETA, is composed of several common re-identification datasets, of which we use a subset consisting of VIPeR, iLIDS, 3DPeS, and CAVIAR. PETA provides a different set of attribute labels including the type and colour of clothing. As per the standard testing protocol for re-identification, 10-fold cross-validation was carried out [15]. Within each cross validation fold, the datasets were randomly partitioned into 50% of persons for training, and 50% for testing.

Details of the network architecture including the number of convolutional layers, layer widths, and the convolutional filter sizes are shown in Fig. 3. The network architecture, and hyper-parameters such as the learning rate and number of training epochs, were set using an initial random partitioning of the VIPeR dataset into 50% training and 50% testing data. This initial partition was not included during testing, and no further attempt was made to fit the hyperparameters to the other individual datasets. While this procedure could be seen as over-fitting the hyperparameters to the VIPeR dataset, we show in Section V-C that the network's performance is not overly sensitive to the values of these hyperparameters. Furthermore, testing was also performed using additional datasets, not seen during hyperparameter optimisation, to provide a better indication of the system's true performance (see Section V-F).

The number of training epochs was fixed at 600. During each epoch the network was presented with all matching image pairs once, and an equal number of randomly selected mismatching image pairs. Training was carried out using stochastic gradient descent, with a batch size of one, i.e. the parameters were updated after showing the network each image pair. The weights were initialised using the default Torch initialisation, described in [34]. During training the learning rate was linearly decreased from 1e-3 to 1e-5. Evaluation was carried out using an Nvidia Tesla M2070, with training taking around 10 seconds per epoch. Testing can be carried out quickly as each image need only be passed through the network once to produce a feature, which is saved for reuse. Similarity scores between the gallery and an image can then be computed quickly using a matrix vector product.

A. Training Analysis

During training, all network parameters were recorded every ten epochs. This allows the network's performance to be compared between training epochs by using the saved weights to calculate a CMC curve, then plotting the rank 1 accuracy over time. In Fig. 4 we compare the rank 1 CMC for the network trained using multi-task learning, and a version of the network trained using only verification, for both the training set and testing sets. We can see that when multi-task learning is used, the training-set rank 1 CMC consistently increases, and the testing rank 1 CMC asymptotically approaches ~33%. However, when the network is trained using verification only i.e., a standard Siamese architecture, the training-set rank 1 CMC consistently increases, while the testing-set rank 1 CMC does not significantly change. These trends suggest that this version of the network is over-fitting to the training-set i.e., the learned weights do not generalise well, as they do not improve performance on the testing-set. This experiment provides evidence that the use of multi-task learning (i.e., a combination of verification, identification, and attribute labelling), helps to prevent over-fitting and to improve the network's ability to perform re-identification, compared to using verification error alone.

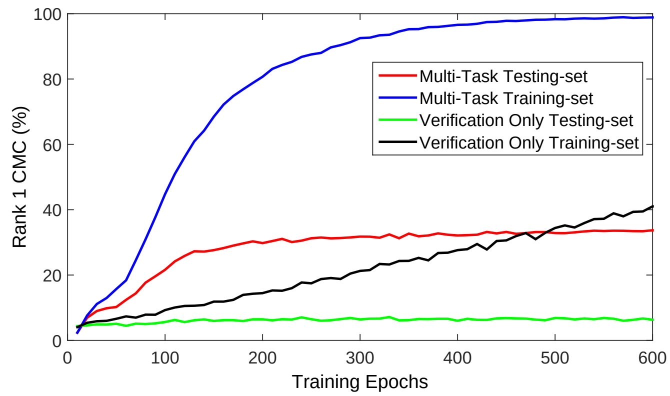

Fig. 4. Comparing Rank-1 CMC accuracy on the VIPeR dataset as a function of training epochs, for the network trained with Multi-Task Learning, and for the network trained using verification only.

B. Multi-task Learning

This experiment explores the effect of multi-task learning on re-identification performance for VIPeR and PETA. Pose information was supplied with VIPeR, however this was not available for PETA. In this experiment the network was trained using the same test / train split and the same weight initialisation, but with different sets of tasks. A CMC curve for each set of tasks, on each dataset, is shown in Fig. 5.

The results in Fig. 5, which are consistent across both datasets, indicate that using the identification task, in addition to verification, may be key to achieving good performance. The large boost in performance could be attributed to the fact that each person has a unique identity, meaning the identification task involves predicting from a diverse set of labels, forcing the network to learn the subtle differences between the appearances of individuals, rather than simply performing verification by comparing image pairs. A further improvement, achieving the best performance, occurs when the verification, identification, and attribute labelling tasks are used together. Use of attribute labelling together with verification produces a large improvement in performance compared with verification used alone. However, the relatively smaller improvement when attribute labelling is used with verification and identification, may be due to the success already achieved by identification and the relative lack of diversity of the attribute labels. The relative utility of the attribute classification and identification tasks can be seen by comparing the curve for verification and attributes with that of verification and identification.

While the identification task requires a highly diverse set of labels to be predicted, in the attribute labelling task, the labels for the VIPeR dataset show a lack of diversity, with positive examples of certain attributes, such as *Headphones* or *Sandals*, observed in only a small number of persons. Only the *Sex* and *Jeans* attributes have an approximately balanced number of positive and negative examples, which is reflected in the observed attribute classification accuracies (See Section V-B1). We hypothesise that if a more diverse and balanced set of auxiliary labelling tasks were to be used, larger performance gains may be observed.

It is interesting to note that pose classification task causes a drop in performance on VIPeR. As explained in Sec-

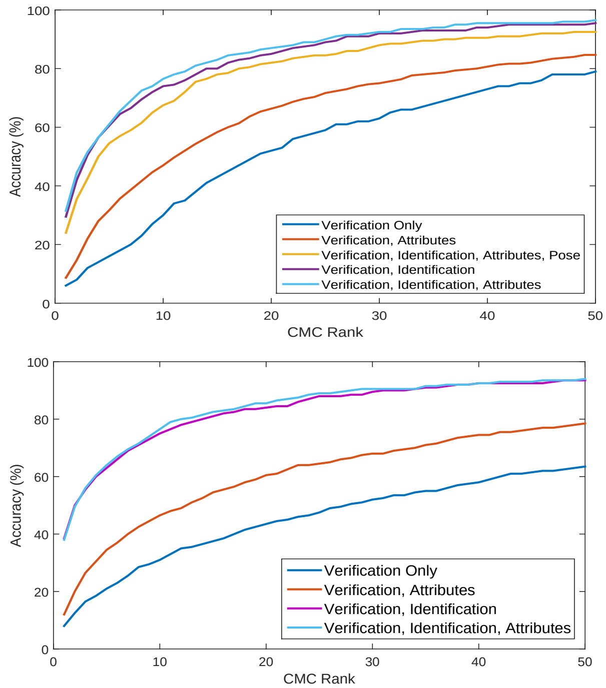

Fig. 5. CMC curve after multi-task training on VIPeR (top) and PETA (bottom) with different sets of tasks.

tion III-B3, this is perhaps to be expected as re-identification features should be invariant to pose. While other attributes such as sex, clothing or identity help to discriminate between individuals, pose may be different for the same individual in different images. In light of these considerations, we hypothesise that re-identification accuracy may be improved by explicitly enforcing pose invariance in the learned feature representation. This objective could be achieved using the gradient reversal method proposed by [16], but we leave this to future work.

1) Attribute and Pose Classification: While the primary aim of using the auxiliary attribute classification tasks is to help improve re-identification performance, it is interesting to observe the network's classification accuracy for these tasks, as they may have applications independent of person re-identification. In this experiment the network was trained using a random 50% of the VIPeR dataset, and used to perform attribute and pose classification on the remaining 50%. Due to the very unbalanced number of positive and negative examples, results are reported in Table I in terms of classification accuracy, precision, and recall.

The best performance, in terms of precision and recall, occurs for the *Sex,* and *Jeans* attributes. This is perhaps unsurprising as these attributes have an almost balanced number of positive and negative examples in the training-set, which makes training an accurate classifier easier. The situation is reversed with the *headphones*, *v-neck*, *stripes*, and *sandals* attributes, where the classifier accuracy, albeit high, is not statistically significant due to the very unbalanced distribution of positive and negative training examples. For these attributes the vast majority of instances are negative, meaning a high classification accuracy, with very low precision and recall, is achieved by a classifier that always predicts the negative label regardless of its input. The classification accuracy for the pose labelling task is 67%, which is well above the level of chance (~20%). Overall, these results show that the best classification accuracy occurs when the network is given a balanced number of positive and negative examples for training. To obtain better attribute classification performance with the given label set, the training procedure would need to be modified to take the data imbalance into account. Our current attribute classification results compare favourably with [32], which obtained an average classification accuracy of 59%, whereas our system obtained 80%. For gender classification, our method compares favourably with [9], which obtained a classification accuracy of 80.62% on the frontal VIPeR images only, whereas we test and train on all images regardless of pose.

TABLE I
ATTRIBUTE CLASSIFICATION ON THE VIPeR DATASET, IN TERMS OF CLASSIFICATION ACCURACY (%), PRECISION (PRE.), AND RECALL (REC.). WE ALSO INCLUDE THE ENTROPY OF THE LABEL DISTRIBUTION (ENT.).

Attribute	Acc.	Pre.	Rec.	Ent.	Attribute	Acc.	Pre.	Rec.	Ent.
Shorts	90	0.47	0.45	0.49	Stripes	91	0.25	0.04	0.40
Sandals	93	0.00	0.00	0.34	Sunglasses	75	0.35	0.21	0.68
Backpacks	65	0.52	0.39	0.94	H.phones	97	0.00	0.00	0.16
Jeans	81	0.75	0.71	0.93	S. Hair	65	0.65	0.66	0.99
Carrying	68	0.31	0.23	0.85	L. Hair	75	0.63	0.51	0.91
Logo	78	0.36	0.21	0.71	Sex	75	0.76	0.73	0.99
V-Neck	91	0.00	0.00	0.41	Skirt	75	0.50	0.29	0.27
OpenOuter	78	0.47	0.25	0.76	Pose	67	0.60	0.60	2.20

2) Attribute Utility : The attribute classification results in the previous section suggest that some attributes may be potentially more useful than others for improving performance. Therefore, two experiments were performed to firstly, evaluate the contribution of individual attributes, and secondly, to explore a better way to combine attributes classification tasks based on their reliability.

First, it is informative to examine the testing-set area under the CMC curve for the network trained to jointly perform re-identification and classification of a single attribute. Comparing the area under the curve (AUC) with the system trained without any additional attribute classification tasks shows how each attribute affects re-identification accuracy. The results in Fig. 6 show that the attributes which give the largest increase in performance, compared with the system trained only to perform re-identification, are again jeans and sex.

3) Weighted Attributes: Following the above idea (see Section V-B2), we hypothesise that re-identification performance could be improved by weighting the attributes in the overall cost function i.e. by using a different α_k value for each attribute. Several weighting methods were compared: zero-weighting i.e. no attribute classification tasks used, equal-weighting of all attribute classification tasks i.e. $\alpha = 1/A$, and weighting each attribute classification task w_a proportional to the Shannon entropy of its label distribution, giving a higher weight to attributes with a near-equal number of positive and negative training examples, which may contribute more useful discriminative information than attributes with a highly skewed label distribution. In all cases the weights were normalised to sum to one. For each weighting approach, the network

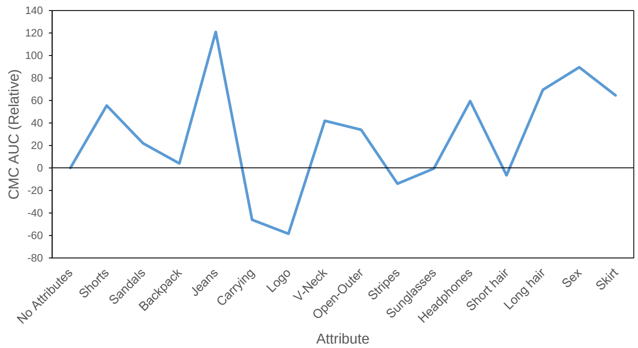

Fig. 6. The area under the CMC curve (CMC AUC) for the system trained to perform re-identification and a single attribute classification task. The AUC values are relative to the AUC for the system trained without additional attribute classification tasks.

TABLE II
THE CMC CURVES ON THE VIPER AND PETA DATASETS USING DIFFERENT METHODS OF WEIGHTING THE ATTRIBUTE LABELLING TASKS. THE VARIANCE AT EACH CMC IS SHOWN IN BRACKETS.

		CMC							
Dataset	Weighting	1	2	3	4	5	10	20	50
VIPeR	Entropy	**29.8**	**42.2**	**49.8**	55.4	59.6	**73.0**	84.6	**96.4**
		(4.2)	(5.2)	(9.1)	(3.0)	(1.6)	(5.0)	(2.3)	(0.7)
	Equal	28.2	41.0	48.4	53.8	58.4	72.2	**84.8**	96.2
		(1.4)	(7.0)	(4.3)	(2.6)	(2.4)	(2.3)	(1.0)	(0.7)
	No Attr.	27.8	41.6	49.2	**56.6**	**60.4**	72.6	84.2	96.0
		(4.3)	(5.3)	(1.3)	(0.3)	(0.8)	(1.3)	(2.7)	(0.5)
PETA	Entropy	**50.1**	61.6	68.2	**71.9**	**75.0**	83.2	90.8	**97.0**
		(4.8)	(6.4)	(2.6)	(2.8)	(3.8)	(1.9)	(2.7)	(1.9)
	Equal	50.1	61.6	**68.3**	71.6	74.5	82.9	90.3	96.8
		(4.6)	(3.9)	(1.9)	(2.8)	(1.6)	(2.6)	(3.1)	(2.2)
	No Attr.	49.9	**61.9**	68.0	71.8	74.7	**83.3**	**90.8**	96.8
		(1.9)	(3.5)	(6.7)	(3.9)	(5.2)	(1.8)	(4.0)	(1.4)

was trained using the whole training-set, with the attribute classification tasks appropriately weighted. Re-identification was then performed on the testing-set. The CMC accuracy for each weighting method is given in Table II for the VIPeR and PETA datasets.

Overall it can be seen that use of attributes during training generally gives higher performance than not using attributes, and using attributes generally reduces the variance in performance. The best testing performance in low ranks, including rank 1, is given by entropy weighting. Entropy weighting does not require learning of weights as the entropy is calculated based on the distribution of attribute labels in the training samples. This means that all training samples are available for training the network, rather than learning the weights, with the corresponding potential performance improvement that this entails. Therefore, we can conclude that entropy weighting is a convenient attribute weighting choice as no prior information about the attributes is required for training.

C. Parameter Sensitivity

This experiment will investigate the importance of the main network hyperparameters, such as the number of convolutional layers, number of filters per convolutional layer (i.e., layer width), and the length of the feature representation (number of neurons in the final layer). The hyperparameters were systematically varied, while keeping the test / train split and weight initialisation constant, allowing their impact on performance to be measured and compared.

1) Layer Width: In this experiment the layer width (i.e., the number of convolutional filters per-layer), and number of neurons in the final network layer were varied. In order to reduce the number of networks to be trained, the number of first layer filters was fixed at 16, while the number of filters in deeper layers was varied together between 32,64 and 128 filters, and the number of final layer neurons was varied between 32, 64, and 128 neurons. Results are shown in Table III for CMC 1, 5 and 10, for both simplicity, and as these are arguably the most important CMC results.

We can see that increasing the layer width generally improves performance. However, performance improves slowly even when the number of filters is increased significantly. Similarly, increasing the number of final layer neurons, while holding the other parameters constant, also improves performance, but again this approach gives diminishing returns. The final row of Table III show performance in the limit, as the layer width, and feature length, is increased to 256, giving only a small improvement in performance while significantly increasing the computational demands.

TABLE III
CMC ACCURACY WHEN VARYING THE NUMBER OF CONVOLUTIONAL FILTERS PER LAYER (1ST, 2ND, AND 3RD LAYERS), AND THE NUMBER OF NEURONS IN THE FULLY-CONNECTED LAYER (FINAL).

Network Layer Parameters				CMC		
1st	2nd	3rd	Final	1	5	10
16	32	32	32	23	50	63
16	64	64	32	26	54	71
16	128	128	32	27	55	65
16	32	32	64	25	55	66
16	64	64	64	32	56	69
16	128	128	64	29	59	71
16	32	32	128	27	53	66
16	64	64	128	31	59	71
16	128	128	128	33	59	73
16	256	256	256	33	60	73

2) Number of Layers: An important hyperparameter of any deep network is the number of layers used, as deeper networks can give better performance than shallower networks [54], [49]. In this experiment the number of layers was systematically varied. Note that, because the initial input image was 64x64 pixels, and each convolutional layer is followed by a max-pooling step which halves the size of the activation map, a maximum of 4 layers were used. For all the networks tested, the first layer always had 16, 5x5 pixel convolutional filters, and all subsequent layers had 64, 5x5 convolutional filters. Re-identification performance, for CMC Rank 1 to 50, as a function of the number of network layers is shown in Fig. 7. We can see that as the number of layers is increased, re-identification performance initially improves, and then saturates after 3 layers i.e., the performance of a network with 4 layers is almost identical to that of a network with 3 layers. These results may indicate that the amount of training data available, or the training methods used, are insufficient for training a very deep network.

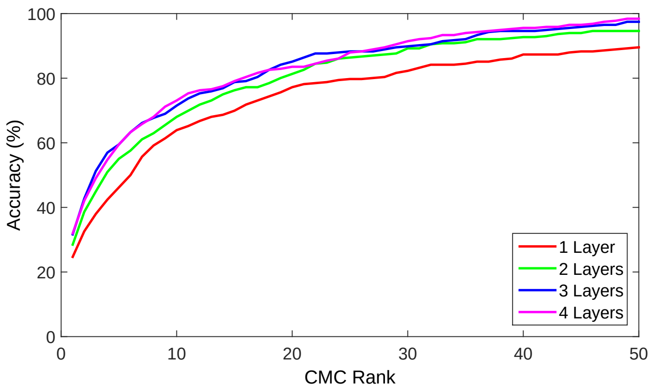

Fig. 7. The network's CMC curve when different numbers of layers are included in the convolutional network, $G(x)$.

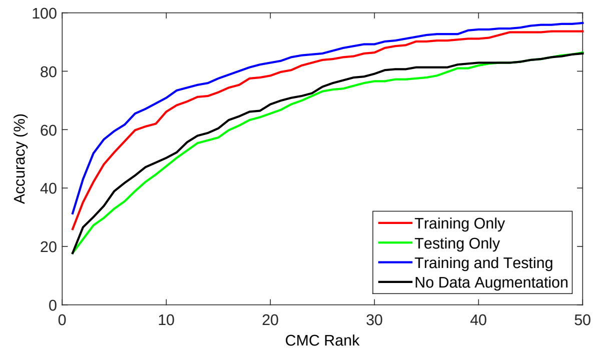

Fig. 8. Network's CMC curve with different data augmentation (DA) methods.

3) Data Augmentation: In this experiment the network was trained with and without data augmentation, while keeping all other parameters constant. Note that while the standard testing procedure for finding the similarity between images involves taking ten crops of 56x56 pixels from each image, then finding the mean Euclidean distance between the features of the crops, with data-augmentation disabled, only the centre crop was used.

The results in Fig. 8 show that using data augmentation during both training and testing gives the best performance, and that using data augmentation during training alone produces slightly poorer results. However, when data augmentation is used during testing only, performance drops significantly, in fact performance drops to below that observed when no data-augmentation is used. This performance drop could be attributed to a mismatch between the training and testing conditions. These results confirm the importance of data-augmentation for re-identification, where due to the small number of examples per-person data augmentation significantly increases the diversity of the training data, thus improving performance.

D. Analysis of Feature Representation

An issue sometimes raised regarding neural networks is the difficulty of understanding what the network is doing [62]. We therefore try to understand the image characteristics used by the network when performing re-identification. Firstly, by using artificially corrupted testing images to understand which image regions the network considers important. And secondly, by investigating the link between the activation of neurons in the final network layer and specific image characteristics.

1) Image Region Importance: In this experiment, the importance of each image region was visualised using a method similar to [62]. A pre-trained network was presented with an uncorrupted set of testing images, and the CMC curve recorded for use as a baseline. One thousand testing iterations were then carried out using the same images, but with each image corrupted by a region filled with zeros. At the start of each testing iteration, a new corruption region was randomly generated, and held constant across all the testing images during that iteration. At the end of each iteration, the decrease in rank 1 CMC accuracy, compared with the baseline, as well as the size and position of the corruption region, were recorded. An example corrupted image is shown in Fig. 9.

The premise of this experiment is that the decrease in overall CMC accuracy, when a region is corrupted, can be used to infer the region's importance for re-identification. To visualise region importance we generate an image where each pixel p has brightness proportional to $\sum_{r \in R} 1 - (c_r/b)$, where R is the subset of all corruption regions that include pixel p, where c_r is the rank 1 CMC accuracy given corruption region r, and where b is the baseline rank 1 CMC accuracy given no corruption. Region importance is visualised in Fig. 9, showing that the network has learned to assign higher importance to the centre of the images, which are more likely to contain discriminative information related to identity, than the edges, which are more likely to contain background information.

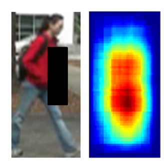

Fig. 9. Left image shows an illustration of the corruption added to testing images. Right image shows the heat-map of image region importance for re-identification, where red areas are the most important, and blue the least.

2) Relating Neuron Activations to Image Features: To gain an insight of the image characteristics learned by the network for re-identification, we investigate the relationship between input images and the activation of individual neurons in the final layer. For example, we would like to know whether individual neurons activate in response to specific items of clothing, or in response to features such as colour.

To visualise the image characteristics that activate individual neurons, we present a trained network with a large set of person images I. For each image, $i \in I$, the activation, $x_{n,i}$, of final layer neuron n is recorded. We then calculate an activation image, a_n, for each final-layer neuron, equal to the weighted average of all the presented images i.e., $a_n \propto \sum_{i \in I} w_{n,i} i$. Where the weight, $w_{n,i}$, associated with each image is $w_{n,i} = e^{x_{n,i}} / \sum_{i \in I} e^{x_{n,i}}$ i.e., images are weighted in proportion to how strongly they activate neuron n, with greatest emphasis on images that cause the strongest activation. Finally, the (unweighted) mean of all the images is subtracted from each a_n to produce the final activation image.

This method can also be used to visualise the image characteristics that cause the most negative activation of each final-layer neuron by using inverse weightings i.e. $\bar{w}_{n,i} = 1 - w_{n,i}$. Some example activation images are shown in Fig. 10.

By comparing the image characteristics that most and least excite each neuron it seems that in many cases the network has learned to detect colour contrasts, such as a difference in colour between the top and bottom of a person's clothing. This type of feature is likely to be well preserved across viewpoint and illumination changes. In other cases it seems that certain neurons are responding to pose variation, for example, compare the position of the legs in the second image, top row, of Fig. 10, with the other images.

Fig. 10. Synthetic image-pairs for a random selection of final-layer neurons, showing the weighted average of the testing images giving the most positive, and most negative responses. For each pair, the positive response image is on the left, and negative response image is on the right.

E. Person Re-Identification Accuracy

1) VIPeR: The CMC results for the VIPeR dataset, calculated using 10-fold cross-validation, are shown in Table IV and are compared with published results from the literature. We also show the results for a multi-label variant of our system that predicts a vector of all attributes simultaneously (Ours Multi-Label). The rank 1 performance of our system is approximately 33%, which compares favourably with recent results such as [2], [36], [64] and significantly outperforms [56]. The standard deviation of the rank 1 CMC accuracy across the cross-validation runs was 3.2%, showing that performance is consistent across different data splits and different network initialisations. The performance of both variants of our system is similar, however the softmax based architecture performs marginally better for CMC ranks above 10. Therefore this architecture is used in all following experiments as the high CMC ranks could be considered more relevant in practical applications. Our network's performance is similar to that of [61], which also makes use of a Siamese network architecture, but was trained using verification only. The network in [61] is significantly larger than ours, with around 14 million parameters, and uses a significantly different convolutional network architecture. Our results are also comparable with [1] which uses a new network layer to compare local patches, and has around 2 million parameters. Our results show that a much simpler network with only around $800,000$ parameters can be trained using multi-task learning to produce near identical performance to a network with a more complex design. Therefore, our training approach has the advantage of producing a network with improved computational efficiency. Additionally, multi-task learning could be used during the training of a network with a more complex architecture design to further improve its performance. Finally, the output of the network, within an ensemble of classifiers, could be fused with other approaches to give higher overall performance.

Although other methods have been proposed that can achieve higher performance on VIPeR, such as [55], [60], [24], [40], our proposed multi-task learning approach is complementary to these methods and could therefore contribute to further performance improvements. Additionally, several of the above mentioned methods use ensembles of re-identification systems [55], [60] to achieve high performance on VIPeR, and use hand-designed features [24], [40]. We suggest that while hand-designed features may be appropriate for small datasets, we expect that as larger datasets become available, automatically learned features will start to perform better, as has been the case in other image recognition applications [28]. Finally, we note that none of the above mentioned methods report their re-identification accuracy in the cross-dataset scenario, which we argue gives a better indication of the system's real world performance. This could imply that over-fitting to VIPeR is occurring, for example in [40] the within-dataset results on VIPeR are very high, while those for PRID are much lower.

TABLE IV
CMC ACCURACY OF OUR SYSTEM ON THE VIPER DATASET, WITH 316 TESTING PERSONS, COMPARED WITH RESULTS FROM THE LITERATURE.

CMC Rank	1	5	10	20	30	40	50	100
Ours	**33.6**	**62.9**	76.5	87.6	92.4	94.8	96.5	99.1
Ours Multi-Label	33.1	62.8	**77.3**	**88.5**	**93.8**	**96.3**	**97.4**	**99.4**
NLML [24]	**42.30**	70.99	**85.23**	**94.25**	-	-	-	-
KCCA [40]	37	-	85	93	-	-	**98**	**100**
IDLA [1]	34.8	64	75	-	-	-	-	-
DML [61]	34.4	62.15	75.89	87.22	92.28	-	96.52	-
SM [64]	30.16	52.3	-	-	-	-	-	-
LAFT [36]	29.6	-	69.31	-	-	-	-	-
RB [2]	30	-	75	87	-	-	96	99
JLCF [56]	26.27	51.90	67.09	-	-	-	-	-
HIF [55]	39.3	**73.0**	84.6	92.5	-	-	-	-
SCN [60]	37.8	68.5	81.2	90.4	-	-	-	-

2) Additional Datasets: Although VIPeR is widely used for assessing re-identification performance, there are many other datasets available. These experiments may provide a better indication of the network's performance, as VIPeR was used for setting the network's hyperparameters, while the other datasets act as unseen testing sets. In this scenario a new network was trained and tested on each re-identification dataset individually. Due to the small size of the datasets, this scenario tests the re-identification performance of the network given extremely limited training data. The network hyperparameters and training procedure remained identical to those described above for the VIPeR dataset (See Section IV). Within each cross-validation fold, the dataset was randomly partitioned into 50% of persons for training and 50% of persons for testing. For each testing-set person, a single randomly selected image was used as their gallery image, and all other images are used for testing. The CMC results, from 10-fold cross-validation, for the iLIDS (p=59), CAVIAR (p=35), and 3DPeS (p=96) datasets, where p is the number of testing people, are reported in Table V.

The results in Table V show that the network can successfully learn to perform re-identification even with a very limited amount of training data, however performance can be further improved by better weight initialisation, which will be discussed in the next experiment (See Section V-E3).

3) Pre-Training and Fine-Tuning: The final scenario of this section investigates the effect of pre-training the network using a larger dataset (VIPeR), then using the learned weights to initialise training on a different dataset. This differs from the standard method of using randomly initialised weights. It has previously been shown that using a good weight initialisation can significantly improve performance [52]. To use this technique, a network is first trained using a dataset for which there is sufficient data e.g. VIPeR, the learned weights are then used to initialise training of a new network on a second, potentially much smaller, dataset. This procedure may take advantage of the generalisation ability of neural networks i.e. the features learned by one network can adapted for solving a variety of related problems [48].

As in previous experiments, 50% of the VIPeR dataset was used to train a re-identification network. The learned weights were then used to initialise training on another re-identification dataset. All the hyperparameters, such as the learning rate and number of epochs, remained identical to those used in previous experiments. This pre-training and testing procedure was repeated 10-times for each dataset, where each dataset was randomly partitioned into 50% training/testing parts. The CMC results for the, iLIDS (p=59), CAVIAR (p=35), and 3DPeS (p=96) datasets, where p is the number of testing people, are shown in Table V.

TABLE V
CMC ACCURACY ON THE ILIDS, CAVIAR AND 3DPES DATASETS, WITH AND WITHOUT INITIALIZING WEIGHTS BY PRE-TRAINING ON VIPER.

				CMC			
Pre-Train	Dataset	#Train	#Test	1	5	10	20
VIPeR	iLIDS	59	60	**37.3**	**66.2**	**76.2**	**87.1**
N/A	iLIDS	59	60	32.8	59.3	72.2	85.7
VIPeR	CAVIAR	36	36	**38.2**	**66.7**	**81.2**	**95.2**
N/A	CAVIAR	36	36	29.1	58.6	74.7	93.6
VIPeR	3DPeS	96	96	**38.2**	**63.7**	**75.4**	**86.6**
N/A	3DPeS	96	96	32.3	56.3	69.1	81.2

The results show that pre-training can significantly improve re-identification performance compared to using random weight initialisation, giving approximately a 6% improvement in rank1 CMC accuracy. Pre-training is especially important for re-identification due to the very small number of training examples for most datasets, compared to VIPeR. These results can be compared with [58], which follows the same experimental procedure, of using a 50% test/train split, however their best reported results were produced by several variants of their algorithm, whereas we maintain constant parameter settings in all experiments. Their highest rank 1 CMC accuracy for iLIDS is 38%, for CAVIAR 40.2%, and for VIPeR 32.3%, which are comparable with our results using pre-training. On the CAVIAR dataset, our method out-performs [45] which obtains rank 1 CMC of 36.2, and which has the advantage of using 5 gallery images per-person. On the 3DPeS dataset, the accuracy of our method, 38%, using 1 gallery image per-person, exceeds that of [43], which obtained a rank 1 CMC of 35.4% using 3 gallery images per-person, and 27.8% using 1 gallery image per-person.

TABLE VI
THE CROSS-DATASET AVERAGE AND INDIVIDUAL CMC CURVES OF THE VIPER, ILIDS, 3DPES, AND CAVIAR DATASETS USING DIFFERENT METHODS OF WEIGHTING THE ATTRIBUTE CLASSIFICATION TASKS. THE HIGHEST VALUE FOR EACH CMC RANK HAS BEEN HIGHLIGHTED.

			CMC				
Train	Test	Weighting	1	5	10	20	50
Average	Average	Entropy	**43.4**	**61.4**	**68.8**	**77.5**	79.8
		Equal	43.1	60.9	68.7	77.3	79.8
		No Attr.	42.7	60.8	68.7	77.4	**79.9**
VIPeR	iLIDS	Entropy	50.4	75.2	85.0	92.6	97.6
VIPeR	iLIDS	No Attr.	49.2	74.6	86.2	92.0	97.6
VIPeR	CAVIAR	Entropy	81.6	92.2	94.6	98.2	-
VIPeR	CAVIAR	No Attr.	80.6	91.0	95.8	98.2	-
VIPeR	3DPeS	Entropy	36.6	55.2	67.0	76.0	91.6
VIPeR	3DPeS	No Attr.	34.4	55.0	66.2	77.4	93.4
iLIDS	VIPeR	Entropy	8.4	20.4	27.8	38.0	56.2
iLIDS	VIPeR	No Attr.	8.2	19.0	26.6	37.4	56
CAVIAR	VIPeR	Entropy	8.8	19.4	27.0	38.8	58.6
CAVIAR	VIPeR	No Attr.	8.8	20.0	27.8	38.8	57.6
3DPeS	VIPeR	Entropy	11.0	25.6	34.4	43.8	61.6
3DPeS	VIPeR	No Attr.	10.6	24.6	33.4	43.6	60.8

F. Cross Data-set Testing

In realistic scenarios a re-identification system will be trained offline with a specific dataset then used on unseen real-world data. This requires the network to generalise from its training dataset. We simulate this scenario by training on one dataset and testing on several different datasets.

1) Weighted Attributes: We first investigate the effect of training using attributes on cross-dataset re-identification performance. A subset of PETA, consisting of VIPeR, iLIDS, 3DPeS, and CAVIAR was used. For each attribute weighting method - no attributes, equally weighted, and entropy weighted - each dataset was used to train a re-identification network, which was then used to preform re-identification on the remaining three datasets. Training used 100% of persons in the training dataset, and testing used 50% of persons in each testing dataset. This process was repeated for all combinations of datasets used for either training or testing, and the average re-identification accuracy was recorded for each attribute weighting method. Results are reported in Table VI, which also shows the results for a subset of the individual combinations of training and testing datasets.

The results in Table VI, agree with those in Table II that using entropy weighting of attributes improves performance compared to the system trained without the use of attributes. These results are evidence that using attributes, specifically with entropy weighting, marginally improves the robustness and generalisation properties of the features learned by the network.

2) Comparison with Literature: This experiment tests the generalisation performance of the re-identification network by training and testing on several different combinations of datasets. For each training dataset, 100% of the persons were used for training, and for each testing dataset a randomly selected 50% of persons were used for testing in each cross-validation split. This follows the cross-dataset testing protocol

TABLE VII
CROSS-DATASET RE-IDENTIFICATION ACCURACY USING A NETWORK THAT TRAINED USING ONE DATASET, THEN PERFORMING RE-IDENTIFICATION ON THE iLIDS, CAVIAR, 3DPES, CUHK, AND PRID2011 DATASETS.

			CMC							
Train Dataset	Test Dataset	Test Persons	1	5	10	20	30	40	50
VIPeR	iLIDS	119	29.7	49.2	60.5	73.1	79.9	84.9	88.6
VIPeR	iLIDS	80	32.7	54.6	67.6	79.3	86.2	91.3	94.9
VIPeR	CAVIAR	72	23.7	40.8	51.0	64.4	74.4	83.3	90.7
VIPeR	3DPeS	96	26.0	47.0	59.4	72.4	79.3	85.5	89.7
VIPeR	CUHK	908	10.3	21.8	28.3	37.3	43.5	48.8	52.0
VIPeR	PRID	100	**9.8**	**19.5**	**25.8**	**33.3**	**41.0**	**47.8**	**52.3**
VIPeR (DTRSVM [41])	PRID	100	4.6	-	17.25	22.9	28.1	-	-
CUHK	PRID	100	**11**	**23.3**	**31.3**	**40.0**	**45.7**	**50.3**	**56.3**
CUHK (IDML [61])	PRID	100	7.6	-	23.4	30.9	36.1	-	-
CUHK	VIPeR	316	16	36.3	45.7	57.3	66.3	74.0	78.3
CUHK (IDML [61])	VIPeR	316	**16.27**	-	**46.27**	**59.94**	**70.13**	-	-
CUHK (DML [61])	VIPeR	316	16.17	-	45.82	57.56	64.24	-	-
iLIDS	VIPeR	316	9.7	19.0	27.7	38.0	47.0	53.3	59.0
iLIDS (IDML [61])	VIPeR	316	**11.61**	-	**34.43**	44.08	52.69	-	-
iLIDS (DTRSVM [41])	VIPeR	316	8.26	-	31.39	**44.83**	**55.88**	-	-
iLIDS	PRID	100	6	17.3	23.3	31.7	38.7	46.0	49.7
iLIDS (IDML [61])	PRID	100	**8**	-	**25.5**	**38.9**	**45.6**	-	-
iLIDS (DTRSVM [41])	PRID	100	3.95	-	18.85	26.6	33.2	-	-
PRID	VIPeR	316	9.7	22.3	**30.3**	**40.0**	**48.7**	**55.7**	**61.7**
PRID (DTRSVM [41])	VIPeR	316	**10.9**	-	28.2	37.69	44.87	-	-

of [61], thus allowing for comparison with their published results. For each training dataset an independent network was trained, and then used to perform re-identification using the images from several different re-identification datasets. Training was performed with the VIPeR, CUHK, PRID-2011 and iLIDS datasets. The results of this experiment for the VIPeR (p=316), iLIDS (p=119 and p=80), CAVIAR (p=72), 3DPeS (p=96) and CUHK (p=908) datasets, where p is the number of people used in testing, are shown in Table VII.

The results in Table VII show that our cross-dataset rank 1 CMC performance is comparable to that of recent within-dataset results (matched datasets in training and testing) obtained by other approaches. Thus our rank 1 CMC of 32.7 on the iLIDS dataset with 80 testing persons is comparable to [66], which achieves a rank 1 CMC of 32.6. Similarly, our results for the CAVIAR and 3DPeS datasets are only slightly poorer than [45], which achieves a rank 1 CMC of 33.4 on 3DPeS, and 36.2 on CAVIAR. Compared with [61] our cross-dataset performance is similar when tested on VIPeR, and better when tested on PRID2011, but with the computational and practical advantages of our system using a much smaller network (around one third the size). These results demonstrate that our network trained on one dataset can learn a feature representation that generalises well to different datasets.

VI. CONCLUSIONS

In this paper we have demonstrated a novel method for person re-identification using a deep convolutional network. We show that by using multi-task learning, consisting of verification, identification, and attribute labelling; a convolutional network with a simple architecture can be trained to perform person re-identification at state-of-the-art levels. Our experiments show that use of identification in conjunction with verification is crucial to achieving high re-identification accuracy. As a side effect of the training process, our network is capable of accurately classifying attributes related to clothing, pose, and sex, from a single full body image, which could have applications independent of person re-identification. We go on to show that the features learned by the network generalise well to unseen datasets by performing cross-dataset testing. Finally, we note that our new multi-task training algorithm is general enough that it could be applied to the training of other re-identification networks to improve their performance regardless of the underlying architecture.

REFERENCES

[1] E. Ahmed, M. Jones, and T. K. Marks. An improved deep learning architecture for person re-identification. In *Proceedings of the IEEE Conference on CVPR*, pages 3908–3916, 2015.

[2] L. An, M. Kafai, S. Yang, and B. Bhanu. Reference-based person re-identification. In *Advanced Video and Signal Based Surveillance*, pages 244–249, Aug 2013.

[3] S. Bak, G. Charpiat, E. Corvee, F. Bremond, and M. Thonnat. Learning to match appearances by correlations in a covariance metric space. In *ECCV*, pages 806–820. 2012.

[4] D. Baltieri, R. Vezzani, and R. Cucchiara. 3dpes: 3d people dataset for surveillance and forensics. In *International ACM Workshop on Multimedia access to 3D Human Objects*, pages 59–64, Scottsdale, Arizona, USA, Nov. 2011.

[5] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning a mahalanobis metric from equivalence constraints. *JMLR*, 6(6):937–965, 2005.

[6] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, and R. Shah. Signature verification using a siamese time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(04):669–688, 1993.

[7] D. Chen, Z. Yuan, G. Hua, N. Zheng, and J. Wang. Similarity learning on an explicit polynomial kernel feature map for person re-identification. In *2015 IEEE Conference on CVPR (CVPR)*, pages 1565–1573, June 2015.

[8] D. S. Cheng, M. Cristani, M. Stoppa, L. Bazzani, and V. Murino. Custom pictorial structures for re-identification. In *BMVC*, volume 2, page 6, 2011.

[9] M. Collins, J. Zhang, P. Miller, and H. Wang. Full body image feature representations for gender profiling. In *ICCV Workshops*, pages 1235–1242. IEEE, 2009.

[10] G. E. Dahl, N. Jaitly, and R. Salakhutdinov. Multi-task neural networks for qsar predictions. *arXiv preprint arXiv:1406.1231*, 2014.

[11] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216. ACM, 2007.

[12] Y. Deng, P. Luo, C. C. Loy, and X. Tang. Pedestrian attribute recognition at far distance. In *Proceedings of the ACM International Conference on Multimedia*, pages 789–792. ACM, 2014.

[13] M. Dikmen, E. Akbas, T. Huang, and N. Ahuja. Pedestrian recognition with a learned metric. In *Asian Conference on Computer Vision*, volume 6495, pages 501–512. 2011.

[14] S. Ding, L. Lin, G. Wang, and H. Chao. Deep feature learning with relative distance comparison for person re-identification. *Pattern Recognition*, 48(10):2993–3003, 2015.

[15] M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani. Person re-identification by symmetry-driven accumulation of local features. In *CVPR*, pages 2360–2367, June 2010.

[16] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016.

[17] D. Gray, S. Brennan, and H. Tao. Evaluating appearance models for recognition, reacquisition, and tracking. In *International Workshop on Performance Evaluation for Tracking and Surveillance*, 2007.

[18] D. Gray and H. Tao. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In *ECCV*, volume 5302, pages 262–275. 2008.

[19] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, volume 2, pages 1735–1742. IEEE, 2006.

[20] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[21] M. Hirzer, C. Beleznai, P. Roth, and H. Bischof. Person re-identification by descriptive and discriminative classification. In *Image Analysis*, volume 6688, pages 91–102. 2011.

[22] M. Hirzer, P. Roth, M. Kstinger, and H. Bischof. Relaxed pairwise learned metric for person re-identification. In *ECCV*, volume 7577, pages 780–793. 2012.

[23] A. G. Howard. Some improvements on deep convolutional neural network based image classification. *arXiv preprint arXiv:1312.5402*, 2013.

[24] S. Huang, J. Lu, J. Zhou, and A. K. Jain. Nonlinear local metric learning for person re-identification. *arXiv preprint arXiv:1511.05169*, 2015.

[25] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In *ICCV*, pages 2146–2153. IEEE, 2009.

[26] O. Javed, K. Shafique, and M. Shah. Appearance modeling for tracking in multiple non-overlapping cameras. In *CVPR*, volume 2, pages 26–33. IEEE, 2005.

[27] M. Kostinger, M. Hirzer, P. Wohlhart, P. Roth, and H. Bischof. Large scale metric learning from equivalence constraints. In *CVPR*, pages 2288–2295, June 2012.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[29] I. Kviatkovsky, A. Adam, and E. Rivlin. Color invariants for person reidentification. *Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1622–1634, July 2013.

[30] E. H. Land and J. McCann. Lightness and retinex theory. *JOSA*, 61(1):1–11, 1971.

[31] R. Layne, T. M. Hospedales, and S. Gong. Attributes-based re-identification. In *Person Re-Identification*, pages 93–117. 2014.

[32] R. Layne, T. M. Hospedales, S. Gong, et al. Person re-identification by attributes. In *BMVC*, volume 2, page 3, 2012.

[33] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[34] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. 2012.

[35] S. Li, M. Shao, and Y. Fu. Cross-view projective dictionary learning for person re-identification. In *Proceedings of the 24th International Conference on Artificial Intelligence, AAAI Press*, pages 2155–2161, 2015.

[36] W. Li and X. Wang. Locally aligned feature transforms across views. In *CVPR*, pages 3594–3601. IEEE, 2013.

[37] W. Li, R. Zhao, T. Xiao, and X. Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *CVPR*, pages 152–159. IEEE, 2014.

[38] Z. Li, S. Chang, F. Liang, T. Huang, L. Cao, and J. Smith. Learning locally-adaptive decision functions for person verification. In *Proceedings of the IEEE Conference on CVPR*, pages 3610–3617, 2013.

[39] S. Liao, Y. Hu, X. Zhu, and S. Z. Li. Person re-identification by local maximal occurrence representation and metric learning. In *Proceedings of the IEEE Conference on CVPR*, pages 2197–2206, 2015.

[40] G. Lisanti, I. Masi, and A. Del Bimbo. Matching people across camera views using kernel canonical correlation analysis. In *Proceedings of the International Conference on Distributed Smart Cameras*, page 10. ACM, 2014.

[41] A. Ma, P. Yuen, and J. Li. Domain transfer support vector ranking for person re-identification without target camera label information. In *Proceedings of the IEEE ICCV*, pages 3567–3574, 2013.

[42] L. Ma, X. Yang, and D. Tao. Person re-identification over camera networks using multi-task distance metric learning. *Image Processing, IEEE Transactions on*, 23(8):3656–3670, 2014.

[43] N. Martinel and C. Micheloni. Classification of local eigen-dissimilarities for person re-identification. *Signal Processing Letters*, 22(4):455–459, April 2015.

[44] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997.

[45] S. Pedagadi, J. Orwell, S. Velastin, and B. Boghossian. Local fisher discriminant analysis for pedestrian re-identification. In *CVPR*, pages 3318–3325. IEEE, 2013.

[46] F. Porikli. Inter-camera color calibration by correlation model function. In *International Conference on Image Processing*, volume 2, pages II–133. IEEE, 2003.

[47] B. Prosser, W.-S. Zheng, S. Gong, and T. Xiang. Person re-identification by support vector ranking. In *BMVC*, pages 21.1–21.11, 2010.

[48] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. *arXiv preprint arXiv:1403.6382*, 2014.

[49] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[50] C. Su, F. Yang, S. Zhang, Q. Tian, L. S. Davis, and W. Gao. Multi-task learning with low rank attribute embedding for person re-identification. In *Proceedings of the IEEE ICCV*, pages 3739–3747, 2015.

[51] Y. Sun, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. *arXiv preprint arXiv:1406.4773*, 2014.

[52] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *International Conference on Machine Learning*, pages 1139–1147, 2013.

[53] M. J. Swain and D. H. Ballard. Indexing via color histograms. In *Active Perception and Robot Vision*, pages 261–273. 1992.

[54] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

[55] R. R. Varior and G. Wang. Hierarchical invariant feature learning with marginalization for person re-identification. *arXiv preprint arXiv:1511.09150*, 2015.

[56] R. R. Varior, G. Wang, and J. Lu. Learning invariant color features for person re-identification. *arXiv preprint arXiv:1410.1035*, 2014.

[57] K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, pages 1473–1480, 2005.

[58] F. Xiong, M. Gou, O. Camps, and M. Sznaier. Person re-identification using kernel-based metric learning methods. In *ECCV*, pages 1–16. 2014.

[59] X. Xu, W. Li, and D. Xu. Distance metric learning using privileged information for face verification and person re-identification. *Neural Networks and Learning Systems, IEEE Transactions on*, 26(12):3150–3162, 2015.

[60] Y. Yang, J. Yang, J. Yan, S. Liao, D. Yi, and S. Z. Li. Salient color names for person re-identification. In *ECCV*, pages 536–551. 2014.

[61] D. Yi, Z. Lei, and S. Z. Li. Deep metric learning for practical person re-identification. *arXiv preprint arXiv:1407.4979*, 2014.

[62] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833. 2014.

[63] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Learning and transferring multi-task deep representation for face alignment. *arXiv preprint arXiv:1408.3967*, 2014.

[64] R. Zhao, W. Ouyang, and X. Wang. Person re-identification by salience matching. In *ICCV*, pages 2528–2535. IEEE, 2013.

[65] W.-S. Zheng, S. Gong, and T. Xiang. Associating groups of people. In *BMVC*, pages 23.1–23.11, 2009. doi:10.5244/C.23.23.

[66] W.-S. Zheng, S. Gong, and T. Xiang. Person re-identification by probabilistic relative distance comparison. In *CVPR*, pages 649–656. IEEE, 2011.

[67] W.-S. Zheng, S. Gong, and T. Xiang. Reidentification by relative distance comparison. *Transactions on Pattern Analysis and Machine Intelligence*, 35(3):653–668, March 2013.