Pattern Recognition 65 (2017) 238–250

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Deep linear discriminant analysis on fisher networks: A hybrid architecture for person re-identification

Lin Wu, Chunhua Shen*, Anton van den Hengel

School of Computer Science, The University of Adelaide, Adelaide 5005, Australia

ARTICLE INFO

Keywords:
Linear discriminant analysis
Deep Fisher networks
Person re-identification

ABSTRACT

Person re-identification is to seek a correct match for a person of interest across different camera views among a large number of impostors. It typically involves two procedures of non-linear feature extractions against dramatic appearance changes, and subsequent discriminative analysis in order to reduce intra-personal variations while enlarging inter-personal differences. In this paper, we introduce a hybrid deep architecture which combines Fisher vectors and deep neural networks to learn non-linear transformations of pedestrian images to a deep space where data can be linearly separable. The proposed method starts from Fisher vector encoding which computes a sequence of local feature extraction, aggregation, and encoding. The resulting Fisher vector output are fed into stacked supervised layer to seek non-linear transformation into a deep space. On top of the deep neural network, Linear Discriminant Analysis (LDA) is reinforced such that linearly separable latent representations can be learned in an end-to-end fashion. By optimizing an objective function modified from LDA, the network is enforced to produce feature distributions which have a low variance within the same class and high variance between classes. The objective is essentially derived from the general LDA eigenvalue problem and allows to train the network with Stochastic Gradient Descent and back-propagate LDA gradients to compute Gaussian Mixture Model (GMM) gradients in Fisher vector encoding. For empirical evaluations, we test our approach on four benchmark data sets in person re-identification (VIPeR [1], CUHK03 [2], CUHK01 [3], and Market 1501 [4]). Extensive experiments on these benchmarks show that our method can achieve state-of-the-art results.

1. Introduction

The problem of person re-identification (*re*-id) which refers to associate people across camera views at different locations and time, is a fundamental task for a distributed multi-camera surveillance system [5,6]. In surveillance applications, an individual disappearing in one view needs to be matched in other views at different physical locations over time, and be differentiated from numerous visually similar but different candidates in those views. The problem is extremely challenging because it is difficult to match the visual features of individuals captured in different camera views due to the dramatic variations of lightings, poses, viewpoints, and cluttered backgrounds. Some image samples are shown in Fig. 1.

Current research efforts for solving this problem mainly focus on two streams: Developing feature representations that are discriminative for identities yet robust against poses, illumination, viewpoint, and background changes [7–12]; Developing metric learning methods to optimize distance metrics for matching people across views with some attempts to learn a class of effective features from training data [13–24]. Commonly, low-level visual features such as such as color, HOG, Local Binary Pattern (LBP) or a combination of them are used in this matching purpose. However, these hand-crafted features generated from people's visual appearance are either insufficiently discriminative for cross-view matching or unreliably robust to viewing condition changes. To extract features against complex yet large visual variations in person re-id, some deep models based on Convolution Neural Networks (CNNs) [25,26,2,27,28] are developed to compute robust, non-linear, data-dependent feature representations, which are able to describe identities faithfully under extreme circumstances. Despite the witnessed improvement achieved by deep models, the learning of parameters requires a large amount of training data in the form of matching pairs whereas person re-id is facing the small sample size problem [29,30]. Specifically, only hundreds of training samples are available due to the difficulties of collecting matched training images. Data augmentations [31] (e.g., flipping, cropping) provide a way to increase training data. Nonetheless, such trivial augmentations cannot

* Corresponding author.
E-mail addresses: lin.wu@adelaide.edu.au (L. Wu), chunhua.shen@adelaide.edu.au (Chunhua Shen), Anton.vandenhengel@adelaide.edu.au (A.v.d. Hengel).

http://dx.doi.org/10.1016/j.patcog.2016.12.022
Received 25 July 2016; Received in revised form 16 November 2016; Accepted 21 December 2016
Available online 24 December 2016

Fig. 1. Samples of pedestrian images observed in different camera views in CUHK03 data set [2]. Images in columns display the same identity.

enrich data source, but lead to slow convergence rate in training [26]. On the other hand, deep models in person re-id have limitation caused by the loss function they are using, which is defined only at single image level. A popular choice is the cross-entropy loss which is to map each pedestrian image to class scores and then quantifies the agreement between predicted scores and the ground truth labels [2,26,25]. However, the gradients computed from the class-membership probabilities may enlarge the inter-personal differences while unable to reduce the intra-personal variations. Mover, this classification-mode training pipeline essentially maps the inputs to a single, scalar prediction. As a result, intra- and inter-class variations can be destroyed which should be maintained in the learned features.

1.1. Motivation

To address the identified limitations in deep CNNs for person re-id, in this paper we are motivated to develop a hybrid deep architecture that can be comparable to deep convolutions but more suitable for person re-identification with moderate data samples. The proposed method starts from Fisher vector encoding, where the encoding/aggregation step typically involves gradient filter, pooling, and normalization. This can be interpreted as a series of layers that alternate linear and non-linear operations, and hence can be paralleled with the convolutional layers of CNNs [32]. Fisher vectors are empirically demonstrated to be competitive with representations learned by convolutional layers of CNNs [33,34,32]. For instance, Sydorov et al. [32] transfer the idea of end-to-end training and discriminatively learned feature representations in neural networks to SVMs with Fisher kernels. As a result, Fisher kernel SVMs can be trained in a deep way by which classifier parameters and Gaussian mixture model (GMM) parameters are learned jointly from training data. While complex CNN models require vast number of training data to find good representations and task-dependent parameters, learning GMM parameters in Fisher kernels only demands moderate amounts of training data. Thus, the proposed deep network based on Fisher vectors can learn effective features to match persons across views, but with training computational cost reduced substantially and overfitting issue mitigated simultaneously.

The whole Fisher vector encoding process which sequentially performs local feature extraction, generative model of GMM, and dimensionality reduction is an unsupervised fashion. To make the learned features highly discriminative in differentiating individuals, we combine Fisher vectors with supervised layers to seek the discriminant analysis on training data of matching image pairs across camera views. Inspired by recent studied related to Linear Discriminant Analysis (LDA) in combination with deep neural networks [35,36], we introduce deep linear discriminant analysis on Fisher networks to produce feature representations which could be easily separable by a linear model in its deep space such that visual variation within the same identity is minimized while visual differences between identities are maximized. In this way, the discriminative capability of Fisher vectors can be improved and helpful in preserving class separability if LDA gradients are back propagated to compute GMM gradients. Also, in the deep linearly separable feature space, we are able to exploit the beneficial properties of LDA such as low intra-class variance, high inter-class variance, and optimal decision boundaries, to have some plausible improvements on person re-identification.

1.2. Our method

In this paper, we present a hybrid deep architecture for person re-identification, which is comprised of Fisher vectors and multiple supervised layers. The network is trained with an Linear Discriminant Analysis (LDA) criterion because LDA approximates inter- and intra-class variations by using two scatter matrices and finds the projections to maximize the ratio between them [37]. As a result, the computed deeply non-linear features become linearly separable in the resulting latent space. The overview of architecture is shown in Fig. 2.

The proposed hybrid architecture is a feed-forward and back-propagate network. The network feed-forwardly encodes the SIFT features through a parametric generative model i.e., Gaussian Mixture Model (GMM), a set of fully connected supervised layers, followed by a modified LDA objective function. Then, the parameters are optimized by back-propagating the error of an LDA-based objective through the entire network such that GMM parameters are updated to have task-dependent values. In this way, the established non-linearities of Fisher networks provide a natural way to extract optimal discriminative features from arbitrarily distributed raw measurements. The

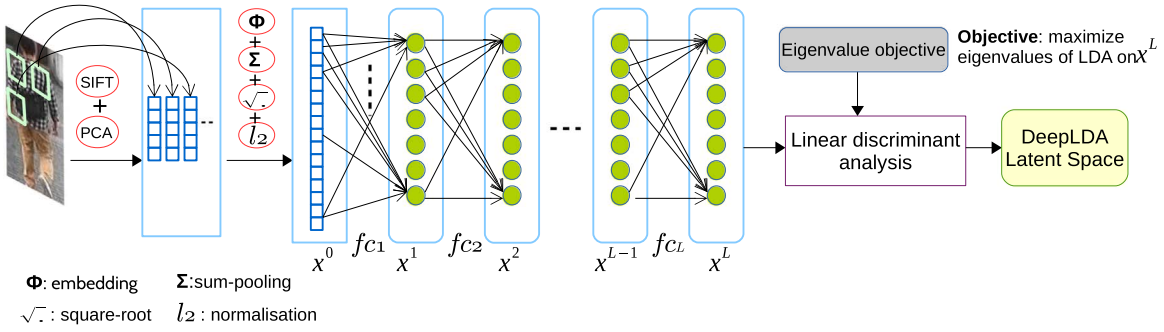

Fig. 2. Overview of our architecture. Patches from a person image are extracted and described by PCA-projected SIFT descriptors. These local descriptors are further embedded using Fisher vector encoding and aggregated to form its image-level representation, followed by square-rooting and ℓ_2-normalization (x^0). Supervised layers $fc_1, \ldots, fc_L$ that involve a linear projection and an reLU can transform resulting Fisher vectors into a latent space into which an LDA objective function is directly imposed to preserve class separability.

subsequent supervised layers serve to learn non-linear transformations from Fisher vector outputs to mixture of Gaussian distributions. Thus, for LDA, since discriminative features generated from non-linear transformations of mixture Gaussian distribution, it is capable of finding linear combinations of the input features which allows for optimal linear decision boundaries. To validate the proposed model, we apply it to four benchmarks VIPeR [1], CUHK03 [27], CUHK01 [3], and Market-1501 [4], which are moderately sized data sets. From this perspective, the learning of our approach can keep the number of parameters reasonable so as to enable the training on medium sized data sets while achieving competitive results.

Our major contributions can be summarized below:

- A hybrid deep network comprised of Fisher vectors, stacked of fully connected layers, and an LDA objective function is presented; The modified LDA objective allows to train the network in an end-to-end manner where LDA derived gradients are back-propagated to update GMM parameters.
- We introduce the optimization of LDA which operates on the properties of parameter distributions of the hidden representation produced by the deep neural networks. This results in linearly separable representations in deep feature space.
- Extensive experiments on benchmark data sets are conducted to show state-of-the-art performance achieved by our method.

The remainder of the paper is structured as follows. Section 2 reviews related works. In Section 3, we introduce linear discriminant analysis on deep Fisher kernels, a hybrid system that learns linearly separable latent representations in an end-to-end fashion. Section 4 explains how the proposed architecture relates to deep Fisher kernel learning and CNNs. In Section 5, we experimentally evaluate the proposed method on benchmark data sets. Section 6 concludes this paper, followed by an appendix in Section 7.

2. Related work

2.1. Person re-identification

Recent studies on person re-id have primarily focused on two aspects: Developing new feature descriptors which are discriminative to identities, yet invariant to view angel and lighting; Developing metric learning algorithms to learn optimal distance metrics across views.

Many person re-id approaches aim to seek robust and discriminative features that can describe the appearance of the same individual across different camera views under various changes and conditions [7–12], For instance, Gray and Tao introduced an ensemble of local features which combines three color channels with 19 texture channels [11]. Farenzena et al. [8] proposed the Symmetry-Driven Accumulation of Local Features (SDALF) that exploited the symmetry property of a person where positions of head, torso, and legs were used to handle view transitions. Zhao et al. combined dcolorSIFT with unsupervised salience learning to improve its discriminative power in person re-id (known as eSDC) [12], and further integrated both salience matching and patch matching into an unified RankSVM framework (SalMatch [38]). They also proposed mid-level filters for person re-id by exploring the partial area under the ROC curve (pAUC) score [10]. However, descriptors of visual appearance are highly susceptible to cross-view variations due to the inherent visual ambiguities and disparities caused by different view orientations, occlusions, illumination, background clutter, etc. Thus, it is difficult to strike a balance between discriminative power and robustness. In addition, some of these methods heavily rely on foreground segmentations, for instance, the method of SDALF et al. [8] requires high-quality silhouette masks for symmetry-based partitions.

Metric learning approaches to person re-id work by extracting features for each image first, and then learning a metric with which the training data have strong inter-class differences and intra-class similarities. In particular, Prosser et al. [23] developed an ensemble RankSVM to learn a subspace where the potential true match is given the highest ranking. Koestinger et al. [16] proposed the large-scale metric learning from equivalence constraint (KISSME) which considers a log likelihood ratio test of two Gaussian distributions. In [14], a Locally-Adaptive Decision Function is proposed to jointly learn distance metric and a locally adaptive thresholding rule. An alternative approach is to use a logistic function to approximate the hinge loss so that the global optimum still can be achieved by iteratively gradient search along the projection matrix as in PCCA [19], RDC [20], and Cross-view Quadratic Discriminant Analysis (XQDA) [22]. Pedagadi et al. [39] introduced the Local Fisher Discriminant Analysis (LFDA) to learn a discriminant subspace after reducing the dimensionality of the extracted features. A further kernelized extension (kLFDA) is presented in [21]. However, these metric learning methods share a main drawback that they need to work with a reduced dimensionality, typically achieved by PCA. This is because the sample size of person re-id data sets is much smaller than feature dimensionality, and thus metric learning methods require dimensionality reduction or regularization to prevent matrix singularity in within-class scatter matrix [22,39,21]. Zhang et al. [30] addressed the small sample size problem by matching people in a discriminative null space in which images of the same person are collapsed into a single point. Nonetheless, their performance is largely limited by the representation power of hand-crafted features, i.e., color, HOG, and LBP.

More recently, deep models based on CNNs that extract features in hierarchy have shown great potential in person re-id. Some recent representatives include FPNN [2], PersonNet [26], DeepRanking [28], and Joint-Reid [25]. FPNN is to jointly learn the representation of a pair of person image and their corresponding metric. An improved architecture was presented in [25] where they introduced a layer that computes cross-input neighborhood difference features immediately after two layers of convolution and max pooling. Wu et al. [26] improved the architecture of Joint-Reid [25] by using more deep layers and smaller convolutionary filter size. However, these deep models learn a network with a binary classification, which tends to predict most input pairs as negatives due to the great imbalance of training data. To avoid data imbalance, Chen et al. [28] proposed a unified deep learning-to-rank framework (DeepRanking) to jointly learn representation and similarities of image pairs.

2.2. Deep fisher networks

One widely used approach in large-scale image classification is to extract local features such as SIFT [40] densely from each image, aggregate and encode them as high-dimensional vectors, and feed the later to a classifier, e.g., an SVM. Among a variety of encodings including BoW encoding [41], sparse coding [42], the Fisher vector encoding [43] outperforms other encodings and achieves impressive performance on various image recognition benchmarks [44]. More recent proposed encodings [45,46] based on Fisher vectors are retaining weak spatial information, and thus yield an improved classification performance. This can incorporated by dividing the image into regions, encoding each of them individually, and stacking the result in a composite higher-dimensional code.

Another vast family of image classification techniques is based on deep convolutional neural networks (CNNs), which can be trained on a large scale with excellent performance [31]. There have been attempts to bridge these two families, exploring the trade-offs network depth and width, as well as the complexity of the layers. For instance, Simonyan et al. [34] proposed a deep Fisher network by stacking

Fisher vector image encoding into multiple layers. This hybrid architecture can significantly improve on standard Fisher vectors, and obtain competitive results with deep convolutional networks at a smaller computational cost. A deep Fisher kernel learning method is presented by Sydorov et al. [32] where they improve on Fisher vectors by jointly learning the SVM classifier and the GMM visual vocabulary. The idea is to derive SVM gradients which can be back-propagated to compute the GMM gradients. Finally, Perronnin [33] introduce a hybrid system that combines Fisher vectors with neural networks to produce representations for image classifiers. However, their network is not trained in an end-to-end manner.

Our method is fundamentally different from aforementioned approaches in deep Fisher networks in terms of a few aspects. Firstly, our architecture is a composition of Fisher vector encoding, stacked supervised layers, and objective with linear discriminant analysis where it allows an end-to-end training to learn data-dependent feature representations. Second, we directly model the parameter distributions instead of penalizing individual training samples in terms of negative/positive pairs. This leads to consistent training and faster convergence rate.

3. Our architecture

3.1. Fisher vector encoding

The Fisher vector encoding Φ of a set of features is based on fitting a parametric generative model such as the Gaussian Mixture Model (GMM) to the features, and then encoding the derivatives of the log-likelihood of the model with respect to its parameters [47]. It has been shown to be a state-of-the-art local patch encoding technique [43,48–50]. Our base representation of an image is a set of local descriptors, e.g., densely computed SIFT features [40]. The descriptors are first PCA-projected to reduce their dimensionality and decorrelate their coefficients to be amenable to the FV description based on diagonal-covariance GMM.

Given a GMM with K Gaussians, parameterized by $\{w_k, \mu_k, \sigma_k, k = 1\ldots, K\}$, the Fisher vector encoding leads to the representation which captures the average first and second order differences between the features and each of the GMM centres [34,51].For a local descriptor $x \in \mathbb{R}^D$, we define a vector $\Phi(x) = [\varphi_1(x), \ldots, \varphi_K(x), \psi_1(x), \ldots, \psi_K(x)] \in \mathbb{R}^{2KD}$. The subvectors are defined as

$$\varphi_k(x) = \frac{1}{\sqrt{w_k}}\gamma_k(x)\left(\frac{x-\mu_k}{\sigma_k}\right) \in \mathbb{R}^D,\ \psi_k(x) = \frac{1}{\sqrt{2w_k}}\gamma_k(x)\left(\frac{(x-\mu_k)^2}{\sigma_k^2} - 1\right) \in \mathbb{R}^D, \tag{1}$$

where $\{w_k, \mu_k, \sigma_k\}_k$ are the mixture weights, means, and diagonal covariance of the GMM, which are computed on the training set. We have:

$$\mu_k(x) = \frac{1}{(2\pi)^{D/2}|\Sigma_k|^{1/2}}\exp\left(-\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)\right), \tag{2}$$

where we assume diagonal covariance matrices which is a standard assumption i.e., $\Sigma_k = diag(\sigma_k)$, $\sigma_k \in \mathbb{R}^D$, and we require $\forall_k : w_k \geq 0$, $\sum_k^K w_k = 1$. $\gamma_k(x)$ is the soft assignment weight of the feature x to the k-th Gaussian:

$$\gamma_k(x) = \frac{w_k\mu_k(x)}{\sum_{j=1}^K w_j\mu_j(x)}. \tag{3}$$

The GMM parameters can be estimated using the Expectation Maximum (EM) algorithm to optimize a Maximum Likelihood (ML) criterion. For more details about the GMM implementation, please refer to the Appendix in [43].

To represent an image $X = \{x_1, \ldots, x_M\}$, one averages/sum-pool the vector representations of all descriptors, that is, $\Phi(X) = \frac{1}{M}\sum_{i=1}^M \Phi(x_i)$. $\Phi(X)$ refers to Fisher vector of the image since for any two image X and X', the inner product $\Phi(X)^T\Phi(X')$ approximates to the Fisher kernel, $k(X, X')$, which is induced by the GMM. The Fisher vector is further processed by performing signed square root and ℓ_2 normalization on each dimension $d = 1, \ldots, 2KD$:

$$\overline{\Phi}_d(X) = (sign\Phi_d(X))\sqrt{|\Phi_d(X)|}/\sqrt{\|\Phi(X)\|_{\ell_1}} \tag{4}$$

For simplicity, we refer to the resulting vectors from Eq. (4) as Fisher vectors and to their inner product as Fisher kernels.

3.2. Supervised layers

In our network, the PCA-reduced Fisher vector outputs are fed as inputs to a set of L fully connected supervised layers $fc_1, \ldots, fc_L$. Each layer involves a linear projection followed by a non-linear activation. Let x^{l-1} be the output of layer fc_{l-1}, and σ denotes the non-linearity, then we have the output $x^l = \sigma(F^l(x^{l-1}))$ where $F^l(x) = W^l x + b^l$, and W^l and b^l are parameters to be learned. We use a rectified Linear Unit (reLU) non-linearity function $\sigma(x) = \max(0, x)$, which showed improved convergence compared with sigmoid non-linearities [31]. As for the output of the last layer fc_L, existing deep networks for person re-id commonly use a softmax layer which can derive a probabilistic-like output vector against each class. Instead of maximizing the likelihood of the target label for each sample, we use deep network to learn non-linear transforms of training samples to a space in which these samples can be linearly separable. In fact, deep networks are propertied to have the ability to concisely represent a hierarchy of features for modeling real-world data distributions, which could be very useful in a setting where the output space is more complex than a single label.

3.3. Deep linear discriminant analysis on fisher networks

Given a set of N samples $\boldsymbol{X} = \{\boldsymbol{x}_1^L, \ldots, \boldsymbol{x}_N^L\} \in \mathbb{R}^{N\times d}$ belonging to different classes $\omega_i, 1 \leq i \leq C$, where $\boldsymbol{x}_i^L$ denotes outputted features of a sample from the supervised layer fc_L, linear discriminant analysis (LDA) [37] seeks a linear transform $\mathbf{A}: \mathbb{R}^d \to \mathbb{R}^l$, improving the discrimination between features $z_i = \boldsymbol{x}_i^L\mathbf{A}^T$ lying in a lower l-dimensional subspace $L \in \mathbb{R}^l$ where $l = C - 1$. Please note that the input representation $\boldsymbol{X}$ can also be referred as hidden representation. Assuming discriminative features z_i are identically and independently drawn from Gaussian class-conditional distributions, the LDA objective to find the projection matrix $\mathbf{A}$ is formulated as:

$$\underset{\mathbf{A}}{\text{argmax}}\frac{|\mathbf{A}\boldsymbol{S}_b\mathbf{A}^T|}{|\mathbf{A}\boldsymbol{S}_w\mathbf{A}^T|}, \tag{5}$$

where $\boldsymbol{S}_b$ is the between scatter matrix, defined as $\boldsymbol{S}_b = \boldsymbol{S}_t - \mathbf{S}_w$. $\boldsymbol{S}_t$ and $\boldsymbol{S}_w$ denote total scatter matrix and within scatter matrix, which can be defined as

$$\boldsymbol{S}_w = \frac{1}{C}\sum_c \boldsymbol{S}_c,\ \boldsymbol{S}_c = \frac{1}{N_c - 1}\overline{\boldsymbol{X}}_c^T\overline{\boldsymbol{X}}_c; \boldsymbol{S}_t = \frac{1}{N-1}\overline{\boldsymbol{X}}^T\overline{\boldsymbol{X}}; \tag{6}$$

where $\overline{\boldsymbol{X}}_c = \boldsymbol{X}_c - \mathbf{m}_c$ are the mean-centered observations of class ω_c with per-class mean vector $\mathbf{m}_c$. $\overline{\boldsymbol{X}}$ is defined analogously for the entire population $\boldsymbol{X}$.

In [52], the authors have shown that nonlinear transformations from deep neural networks can transform arbitrary data distributions to Gaussian distributions such that the optimal discriminant function is linear. Specifically, they designed a deep neural network constituted of

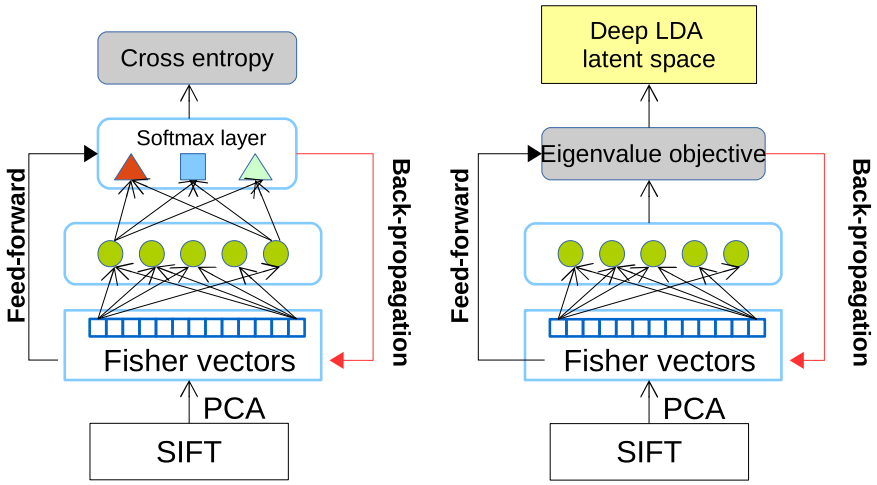

Fig. 3. Sketch of a general DNN and our deep hybrid LDA network. In a general DNN, the outputs of network are normalized by a softmax layer to obtain probabilities and the objective with cross-entropy loss is defined on each training sample. In our network, an LDA objective is imposed on the hidden representations, and the optimization target is to directly maximize the separation between classes.

unsupervised pre-optimization based on stacked Restricted Boltzman machines (RBMs) [53] and subsequent supervised fine-tuning. Similarly, the Fisher vector encoding part of our network is a generative Gaussian mixture mode, which can not only handle real-value inputs to the network but also model them continuously and amenable to linear classifiers. In some extent, one can view the first layers of our network as an unsupervised stage but these layers need to be retrained to learn data-dependent features.

Assume discriminant features $z = \mathbf{A}\boldsymbol{x}$, which are identically and independently drawn from Gaussian class-conditional distribution, a maximum likelihood estimation of $\mathbf{A}$ can be equivalently computed by a maximization of a discriminant criterion: Q_z: $\mathcal{F} \rightarrow \mathbb{R}$, $\mathcal{F} = \{\mathbf{A}: \mathbf{A} \in \mathbb{R}^{l\times d}\}$ with $Q_z(\mathbf{A}) = \max(trace\{\boldsymbol{S}_t^{-1}\boldsymbol{S}_b\})$. We use the following theorem from the statistical properties of LDA features to obtain a guarantee on the maximization of the discriminant criterion Q_z with the learned hidden representation $\boldsymbol{x}$.

Lemma 1. *On the top of the Fisher vector encoding, the supervised layers learn input-output associations leading to an indirectly maximized Q_z. In [52,54], it has shown that multi-layer artificial neural networks with linear output layer*

$$z_n = \mathbf{W}\boldsymbol{x}_n + \mathbf{b} \tag{7}$$

maximize asymptotically a specific discriminant criterion evaluated in the l-dimensional space spanned by the last hidden outputs $z \in \mathbb{R}^l$ if the mean squared error (MSE) is minimized between z_n and associated targets $\mathbf{t}_n$. In particular, for a finite sample $\{\boldsymbol{x}_n\}_{n=1}^N$, and MSE minimization regarding the target coding is

$$t_n^i = \begin{cases} \sqrt{N/N_i}, & \omega(\boldsymbol{x}_n) = \omega_i \\ 0, & \text{otherwise} \end{cases}$$

where N_i is the number of examples of class ω_i, and $t_n{}^i$ is the i-th component of a target vector $\mathbf{t}_n \in \mathbb{R}^C$, approximates the maximum of the discriminant criterion Q_h.

Maximizing the objective function in Eq. (5) is essentially to maximize the ratio of the between and within-class scatter, also known as separation. The linear combinations that maximize Eq. (5) leads to the low variance in projected observations of the same class, whereas high variance in those of different classes in the resulting low-dimensional space L. To find the optimum solution for Eq. (5), one has to solve the general eigenvalue problem of $\boldsymbol{S}_b\boldsymbol{e} = \mathbf{v}\boldsymbol{S}_w\boldsymbol{e}$, where $\boldsymbol{e}$ and $\mathbf{v}$ are eigenvectors and corresponding eigenvalues. The projection matrix $\mathbf{A}$ is a set of eigenvectors $\boldsymbol{e}$. In the following section, we will formulate LDA as an objective function for our hybrid architecture, and present back-propagation in supervised layers and Fisher vectors.

3.4. Optimization

Existing deep neural networks in person re-identification [25,2,26] are optimized using sample-wise optimization with cross-entropy loss on the predicted class probabilities (see Section below). In contrast, we put an LDA-layer on top of the neural networks by which we aim to produce features with low intra-class and high inter-class variability rather than penalizing the misclassification of individual samples. In this way, optimization with LDA objective operates on the properties of the parameter distributions of the hidden representation generated by the hybrid net. In the following, we first briefly revisit deep neural networks with cross-entropy loss, then we present the optimization problem of LDA objective on the deep hybrid architecture.

3.4.1. Deep neural networks with cross-entropy loss

It is very common to see many deep models (with no exception on person re-identification) are built on top of a deep neural network (DNN) with training used for classification problem [32,33,31,55]. Let X denote a set of N training samples $X_1, \ldots, X_N$ with corresponding class labels $y_1, \ldots, y_n \in \{1, \ldots, C\}$. A neural network with P hidden layers can be represented as a non-linear function $f(\Theta)$ with model parameters $\Theta = \{\Theta_1, \ldots, \Theta_P\}$. Assume the network output $\mathbf{p}_i = [p_{i,1}, \ldots, p_{i,C}] = f(X_i, \Theta)$ is normalized by the softmax function to obtain the class probabilities, then the network is optimized by using Stochastic Gradient Descent (SGD) to seek optimal parameter Θ with respect to some loss function $\mathcal{L}_i(\Theta) = \mathcal{L}(f(X_i, \Theta), y_i)$:

$$\Theta = \underset{\Theta}{\operatorname{argmin}} \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_i(\Theta). \tag{8}$$

For a multi-class classification setting, cross-entropy loss is a commonly used optimization target which is defined as

$$\mathcal{L}_i = -\sum_{j=1}^{C} y_{i,j}\log(p_{i,j}), \tag{9}$$

where $y_{i,j}$ is 1 if sample X_i belongs to class y_i i.e., $(j = y_i)$, and 0

otherwise. In essence, the cross-entropy loss tries to maximize the likelihood of the target class y_i for each of the individual sample X_i with the model parameter Θ. Fig. 3 shows the snapshot of a general DNN with such loss function. It has been emphasized in [56] that objectives such as cross-entropy loss do not impose direct constraints such as linear separability on the latent space representation. This motivates the LDA based loss function such that hidden representations are optimized under the constraint of maximizing discriminative variance.

3.4.2. Optimization objective with linear discriminant analysis

A problem in solving the general eigenvalue problem of $\boldsymbol{S}_b\boldsymbol{e} = \mathbf{v}\boldsymbol{S}_w\boldsymbol{e}$ is the estimation of $\boldsymbol{S}_w$ which overemphasizes high eigenvalues whereas small ones are underestimated too much. To alleviate this effect, a regularization term of identity matrix is added into the within scatter matrix: $\boldsymbol{S}_w + \lambda\mathbf{I}$ [57]. Adding this identify matrix can stabilize small eigenvalues. Thus, the resulting eigenvalue problem can be rewritten as

$$\boldsymbol{S}_b\boldsymbol{e}_i = v_i(\boldsymbol{S}_w + \lambda\mathbf{I})\boldsymbol{e}_i, \tag{10}$$

where $\boldsymbol{e} = [\boldsymbol{e}_1, \ldots, \boldsymbol{e}_{C-1}]$ are the resulting eigenvectors, and $\mathbf{v} = [v_1, \ldots, v_{C-1}]$ the corresponding eigenvalues. In particular, each eigenvalue v_i quantifies the degree of separation in the direction of eigenvector $\boldsymbol{e}_i$. In [56], Dorfer et al. adapted LDA as an objective into a deep neural network by maximizing the individual eigenvalues. The rationale is that the maximization of individual eigenvalues which reflect the separation of respective eigenvector directions leads to the maximization of the discriminative capability of the neural net. Therefore, the objective to be optimized is formulated as:

$$\underset{\Theta,G}{\operatorname{argmax}} \frac{1}{C-1}\sum_{i=1}^{C-1} v_i. \tag{11}$$

However, directly solving the problem in Eq. (11) can yield trivial solutions such as only maximizing the largest eigenvalues since this will produce the highest reward. In perspective of classification, it is well known that the discriminant function overemphasizes the large distance between already separated classes, and thus causes a large overlapping between neighboring classes [52]. To combat this matter, a simple but effective solution is to focus on the optimization on the smallest (up to m) of the $C-1$ eigenvalues [56]. This can be implemented by concentrating only m eigenvalues that do not exceed a predefined threshold for discriminative variance maximization:

$$\mathcal{L}(\Theta, G) = \underset{\Theta,G}{\operatorname{argmax}} \frac{1}{m}\sum_{i=1}^{m} v_i, \text{ with } \{v_1, \ldots, v_m\} = \{v_i | v_i < \min\{v_1, \ldots, v_{C-1}\} + \epsilon\}. \tag{12}$$

The intuition of formulation (12) is the learned parameters Θ and G are encouraged to embed the most discriminative capability into each of the $C-1$ feature dimensions. This objective function allows to train the proposed hybrid architecture in an end-to-end fashion. In what follows, we present the derivatives of the optimization in (12) with respect to the hidden representation of the last layer, which enables back-propagation to update G via chain rule.

Gradients of LDA loss. In this section, we provide the partial derivatives of the optimization function (12) with respect to the outputted hidden representation $\boldsymbol{X} \in \mathbb{R}^{N\times d}$ where N is the number of training samples in a mini-batch and d is the feature dimension outputted from the last layer of the network. We start from the general LDA eigenvalue problem of $\boldsymbol{S}_b\boldsymbol{e}_i = v_i\boldsymbol{S}_w\boldsymbol{e}_i$, and the derivative of eigenvalue v_i w.r.t. $\boldsymbol{X}$ is defined as [58]:

$$\frac{\partial v_i}{\partial \boldsymbol{X}} = \boldsymbol{e}_i^T\left(\frac{\partial \boldsymbol{S}_b}{\partial \boldsymbol{X}} - v_i\frac{\partial \boldsymbol{S}_w}{\partial \boldsymbol{X}}\right)\boldsymbol{e}_i. \tag{13}$$

Recalling the definitions of $\boldsymbol{S}_t$ and following [52,35,56], we can first obtain the partial derivative of the total scatter matrix $\boldsymbol{S}_t$ on $\boldsymbol{X}$ as in Eq. (12). Here

$$\frac{\partial \boldsymbol{S}_t[a,b]}{\partial \boldsymbol{X}[i,j]} = \begin{cases} \frac{2}{N-1}(\boldsymbol{X}[i,j] - \frac{1}{N}\sum_n \boldsymbol{X}[n,j]), & \text{if } a=j,\ b=j \\ \frac{1}{N-1}(\boldsymbol{X}[i,b] - \frac{1}{N}\sum_n \boldsymbol{X}[n,b]), & \text{if } a=j,\ b\neq j \\ \frac{1}{N-1}(\boldsymbol{X}[i,a] - \frac{1}{N}\sum_n \boldsymbol{X}[n,a]), & \text{if } a\neq j,\ b=j \\ 0, & \text{if } a\neq j,\ b\neq j \end{cases} \tag{12}$$

$\boldsymbol{S}_t[a,b]$ indicates the element in row a and column b in matrix $\boldsymbol{S}_t$, and likewise for $\boldsymbol{X}[i,j]$. Then, we can obtain the partial derivatives of $\boldsymbol{S}_w$ and $\boldsymbol{S}_b$ w.r.t. $\boldsymbol{X}$:

$$\frac{\partial \boldsymbol{S}_w[a,b]}{\partial \boldsymbol{X}[i,j]} = \frac{1}{C}\sum_c \frac{\partial \boldsymbol{S}_c[a,b]}{\partial \boldsymbol{X}[i,j]}; \frac{\partial \boldsymbol{S}_b[a,b]}{\partial \boldsymbol{X}[i,j]} = \frac{1}{C}\sum_c \frac{\partial \boldsymbol{S}_t[a,b]}{\partial \boldsymbol{X}[i,j]} - \frac{\partial \boldsymbol{S}_w[a,b]}{\partial \boldsymbol{X}[i,j]}. \tag{13}$$

Finally, the partial derivative of the loss function formulated in (12) w.r.t. hidden state $\boldsymbol{X}$ is defined as:

$$\frac{\partial}{\partial \boldsymbol{X}}\frac{1}{m}\sum_{i=1}^{m} v_i = \frac{1}{m}\sum_{i=1}^{m}\frac{\partial v_i}{\partial \boldsymbol{X}} = \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{e}_i^T\left(\frac{\partial \boldsymbol{S}_b}{\partial \boldsymbol{X}} - \frac{\partial \boldsymbol{S}_w}{\partial \boldsymbol{X}}\right)\boldsymbol{e}_i. \tag{14}$$

Back-propagation in Fisher vectors. In this section, we introduce the procedure for the deep learning of LDA with Fisher vector. Algorithm 1 shows the steps of updating parameters in the hybrid architecture. In fact, the hybrid network consisting of Fisher vectors and supervised layers updates its parameters by firstly initializing GMMs (line 5) and then computing Θ (line 6–7). Thereafter, GMM parameters G can be updated by gradient decent (line 8–12). Specifically, Line 8 computes the gradient of the loss function w.r.t. the GMM parameters w, μ and σ. Their influence on the loss is indirect through the computed Fisher vector, as in the case of deep networks, and one has to make use of the chain rule. Analytic expression of the gradients are provided in Appendix.

Evaluating the gradients numerically is computationally expensive due to the non-trivial couplings between parameters. A straightforward implementation gives rise to complexity of $O(D^2P)$ where D is the dimension of the Fisher vectors, and P is the combined number of SIFT descriptors in all training images. Therefore, updating the GMM parameters could be orders of magnitude slower than computing Fisher vectors, which has computational cost of $O(DP)$. Fortunately, [32] provides some schemes to compute efficient approximate gradients as alternatives.

It is apparent that the optimization w.r.t. G is non-convex, and Algorithm 1 can only find local optimum. This also means that the quality of solution depends on the initialization. In terms of initialization, we can have a GMM by unsupervised Expectation Maximization, which is typically used for computing Fisher vectors. As a matter of fact, it is a common to see that in deep networks layer-wise unsupervised pre-training is widely used for seeking good initialization [59]. To update gradient, we employ a batch setting with a line search (line 10) to find the most effective step size in each iteration. In optimization, the mixture weights w and Gaussian variance Σ should be ensured to remain positive. As suggested by [32], this can be achieved by internally parameterizing and updating the logarithms of their values, from which the original parameters can be computed simply by exponentiation. A valid GMM parameterization also requires the mixture weights sum up to 1. Instead of directly enforcing this constraint that would require a projection step after each update, we avoid this by deriving gradients for unnormalized weights $\widetilde{w}_k$, and then normalize them as $w_k = \widetilde{w}_k / \sum_j \widetilde{w}_j$.

Algorithm 1. Deep Fisher learning with linear discriminant analysis.

Input: training samples $X_1, \ldots, X_n$, labels $y_1, \ldots, y_n$
Output: GMM G, parameters Θ of supervised layers
Initialize: initial GMM, $G = (\log w, \mu, \log\Sigma)$ by using unsupervised expectation maximization[43,51].
repeat
  compute Fisher vector with respect to G: $\Phi_i^G = \Phi(x_i; G)$, $i = 1, \ldots, n$.
  solve LDA for training set $\{(\Phi_i^G, y_i), i = 1, \ldots, n\}$:
  $\Theta \leftarrow \mathcal{L}(\Theta, G) = \text{argmax}_\Theta \frac{1}{m}\sum_{i=1}^{m} v_i$ with $\{v_1, \ldots, v_m\} = \{v_i$
    $|v_i < \min(v_1, \ldots, v_{C-1}) + \epsilon\}$.
  compute gradients with respect to the GMM parameters:
  $\delta_{\log w} = \nabla_{\log w}\mathcal{L}(\Theta, G)$, $\delta_\mu = \nabla_\mu\mathcal{L}(\Theta, G)$, $\delta_{\log\Sigma} = \nabla_{\log\Sigma}\mathcal{L}(\Theta, G)$.
  find the best step size η^* by line search:
  $\eta^* = \text{argmin}_\eta \mathcal{L}(\Theta, G)$ with $G_\eta = (\log w - \eta\delta_{\log w}, \mu - \eta\delta_\mu, \log\Sigma - \eta\delta_{\log\Sigma})$.
  update GMM parameters: $G \leftarrow G_{\eta *}$.
until *stopping criterion fulfilled*
Return GMM parameters G and Θ.

3.5. Implementation details

Local feature extraction. Our architecture is independent of a particular choice of local descriptors. Following [33,43], we combine SIFT with color histogram extracted from the LAB color space. Person images are first rescaled to a resolution of 48×128 pixels. SIFT and color features are extracted over a set of 14 dense overlapping 32×32-pixels regions with a step stride of 16 pixels in both directions. For each patch we extract two types of local descriptors: SIFT features and color histograms. For SIFT patterns, we divide each patch into 4×4 cells and set the number of orientation bins to 8, and obtain $4 \times 4 \times 8 = 128$ dimensional SIFT features. SIFT features are extracted in L, A, B color channels and thus produce a 128×3 feature vector for each patch. For color patterns, we extract color features using 32-bin histograms at 3 different scales (0.5, 0.75 and 1) over L, A, and B channels. Finally, the dimensionality of SIFT features and color histograms is reduced with PCA respectively to 77-dim and 45-dim. The employment of PCA is to ensure the feature vectors from local patches better fit the diagonal covariance matrix assumption in (2). More importantly, PCA dimensionality reduction is the key to make the Fisher vector work and accuracy in image classification is not sensitive to the exact number of PCA components [43].

Considering the patches of an image are not independent especially patches overlap, we take into account the joint distribution of low-level descriptors across patches. Spatial pyramids [45] provide a solution to this purpose by subdividing an image into a set of regions and pooling descriptor-level statistics over these regions. However, this results in an increase in dimensionality, that is, from original $(2D + 1)K$ -dim to $(2D + 1)KR$-dim where R is the number of regions. In this process, following [46], we use a simple solution that competes favorably in performance with spatial pyramids. Let $m = [m_x, m_y]^T$ be the 2D-coordinates of an image patch associated to a PCA reduced local descriptor $x \in \mathbb{R}^{\overline{D}}$, and δ the patch scale, H and W denote the image height and width, then we have the augmented feature vector $\overline{x} \in \mathbb{R}^{\overline{D}+3}$: $\overline{x} = [x, m_x/W - 0.5, m_y/H - 0.5, \log\delta - \log\sqrt{WH}]^T$. Then, we concatenate these descriptors with coordinates as well as the scale of the patches, thus, yielding 80-dim and 48-dim descriptors. In our experiments, we use the publicly available SIFT/LAB implementation from [12]. In Fisher vector encoding, we use a GMM with K=256 Gaussians, which is an empirically value from [43]. The per-patch Fisher vectors are aggregated with sum-pooling, square-rooted and ℓ_2-normalized. One Fisher vector is computed on the SIFT and another one on the color descriptors. The two Fisher vectors are concatenated into a 256 K-dim representation.

4. Comparison with CNNs [31] and deep fisher networks [32]

Comparison with CNNs [31]. The architecture composed of Fisher vector encoding has some conceptual similarities with that of CNNs. Like a CNN, the Fisher vector extraction can also be interpreted as a series of layers that alternate linear and non-linear operations, and can be paralleled with the convolutional layers of CNNs. For example, when training a convolutional network on natural images, the first stages always learn the essentially the same functionality, that is, computing local image gradient orientations, and pool them over small spatial regions. This is, however, the same procedure how a SIFT descriptor is computed. It has been shown in [33] that Fisher vectors are competitive with representations learned by the convolutional layers of the CNN, and the accuracy comes close to that of the standard CNN model [31]. The main difference between our architecture and CNNs lies in lower layers of feature extractions. In CNNs, basic components (convolution, pooling, and activation functions) operate on local input regions, and locations in higher layers correspond to the locations in the image they are path-connected to (a.k.a receptive field), whilst they are very demanding in computational resources and the amount of training data necessary to achieve good generalization. In our architecture, as alternative, Fisher vector representation of an image effectively encodes each local feature (e.g., SIFT) into a high-dimensional representation, and then aggregates these encodings into a single vector by global sum-pooling over the over image, and followed by normalization. This serves very similar purpose to CNNs, but trains the network at a smaller computational learning cost.

Comparison with deep Fisher kernel learning [32]. With respect to the deep Fisher kernel end-to-end learning, our architecture exhibits two major differences. The first difference can be seen from the incorporation of stacked supervised fully connected layers in our model, which can capture high-order relationships between features in each dimension of Fisher vector. The fully connected layers project the input through non-linear transformations[1] into a space where it becomes linearly separable. The second difference is the training objective function. In [32], they introduce a SVM classifier with Fisher kernel that jointly learns the classifier weights and image representation. However, this classification training can only classify an input image into a number of identities. It cannot be applied to an out-of-domain case where new testing identifies are unseen in training data. Specifically, this classification supervisory signals pull apart the

[1] For example, if x_{n-1} is the input, then we have the output $x_n = \sigma(W_n x_{n-1} + b_n)$ where $\sigma(x) = \max(0, x)$ is a rectified Linear Unit (reLU) re-linearity.

Table 1
A summary on related deep models. √ and - indicate the availability and non-applicable of the corresponding property, respectively.

Method	end-to-end	training efficiency	generalization	feature distribution
CNNs	√	–	√	–
Deep Fisher kernel[32]	√	√	–	–
Hybrid	√	√	√	√

features of different identities since they have to be classified into different classes whereas the classification signal has relatively week constraint on features extracted from the same identity. This leads to the problem of generalization to new identities in test. By contrast, we reformulate LDA objective into deep neural network to learn linearly separable representations. In this way, favorable properties of LDA (low (high) intra- (inter-) personal variations and optimal decision boundaries) can be embedded into neural networks. A summary of the advantage and disadvantage between our method and related deep models are given in Table 1.

5. Experiments

5.1. Experimental setting

Data sets. We perform experiments on four benchmarks: VIPeR [1], CUHK03 [2], CUHK01 [3] and the Market-1501 data set [4].

- The **VIPeR** data set [1] contains 632 individuals taken from two cameras with arbitrary viewpoints and varying illumination conditions. The 632 person's images are randomly divided into two equal halves, one for training and the other for testing.
- The **CUHK03** data set [2] includes 13,164 images of 1360 pedestrians. The whole dataset is captured with six surveillance camera. Each identity is observed by two disjoint camera views, yielding an average 4.8 images in each view. This dataset provides both manually labeled pedestrian bounding boxes and bounding boxes automatically obtained by running a pedestrian detector [60]. In our experiment, we report results on labeled data set.
- The **CUHK01** data set [3] has 971 identities with 2 images per person in each view. We report results on the setting where 100 identities are used for testing, and the remaining 871 identities used for training, in accordance with FPNN [2].
- The **Market-1501** data set [4] contains 32,643 fully annotated boxes of 1501 pedestrians, making it the largest person re-id dataset to date. Each identity is captured by at most six cameras and boxes of person are obtained by running a state-of-the-art detector, the Deformable Part Model (DPM) [61]. The dataset is randomly divided into training and testing sets, containing 750 and 751 identities, respectively.

Evaluation protocol. We adopt the widely used single-shot modality in our experiment to allow extensive comparison. Each probe image is matched against the gallery set, and the rank of the true match is obtained. The rank @ k recognition rate is the expectation of the matches at rank k, and the cumulative values of the recognition rate at all ranks are recorded as the one-trial Cumulative Matching Characteristic (CMC) results [24]. This evaluation is performed ten times, and the average CMC results are reported.

Competitors. We compare our model with the following state-of-the-art approaches: SDALF [8], ELF [11], LMNN [62], ITML [13], LDM [63], eSDC [12], Generic Metric [3], Mid-Level Filter (MLF) [10], eBiCov [64], SalMatch [38], PCCA [19], LADF [14], kLFDA [21], RDC [20], RankSVM [23], Metric Ensembles (Ensembles) [24], KISSME [16], JointRe-id [25], FPNN [2], DeepRanking [28], NullReid [30] and XQDA [22]. For a fair comparison, the method of Metric Ensembles takes the structured ensembling of two types of features,[2] SIFT and LAB patterns, where kLFDA is used as the base metric in its top-k ranking optimization.

Table 2
Rank @1 accuracy and number of parameters in training for various deep models on the CUHK03 data set.

Method	r@1	♯ of parameters
DeepFV+CEL	59.52	5,255,268
VGG+CEL	58.83	22,605,924
VGG+LDA	63.67	22,605,824
DeepFV+LDA	63.23	5,255,168

Table 3
Rank @1, @5, @10, @20 accuracy and number of parameters in training for baselines on the CUHK03 data set.

Method	r @1	r @5	r @10	r @20
Deep fisher networks[34]	53.59	77.43	82.50	87.25
Deep fisher kernels[32]	57.47	81.85	87.05	90.40
Spatial+fisher vectors[45]	48.62	72.55	79.14	81.50
Hybrid	63.23	89.95	92.73	97.55

Training. For each image, we form a 6,5536-dimensional Fisher vector based on a 256-component GMM that was obtained prior to learning by expectation maximization. The network has 3 hidden supervised layers with 4096, 1024, and 1024 hidden units and a drop-out rate of 0.2. As we are estimating the distribution of data, we apply batch normalization [65] after the last supervised layer. Batch normalization can help to increase the convergence speed and has a positive effect on LDA based optimization. In the training stage, a batch size of 128 is used for all data sets. The network is trained using SGD with Nesterov momentum. The initial learning rate is set to 0.05 and the momentum is fixed at 0.9. The learning rate is then halved every 50 epochs for CUHK03, Market-1501 data sets, and every 20 epochs for VIPeR and CUHK01 data sets. For further regularization, we add weight decay with a weighting of 10^{-4} on all parameters of the model. The between-class covariance matrix regularization weight λ (in Eq. (10)) is set to 10^{-3} and the ϵ-offset (in Eq. (12)) for LDA optimization is 1.

5.2. Experimental results

5.2.1. Architecture analysis

In this experiment, we provide detailed analysis of the proposed architecture on CUHK03 data set. To have a fair comparison, we train a modified model by replacing LDA loss with a cross-entropy loss, denoted as DeepFV+CEL. To validate the parallel performance of Fisher vectors to CNNs, we follow the VGG model [55] with sequences of 3×3 convolution, and fine tune its supervised layers to CUHK03 training samples. Two modified VGG models are optimized on LDA loss, and cross-entropy loss, and thus, referred to VGG+LDA, and VGG +CEL, respectively. In training stage of VGG variant models, we perform data pre-processing and augmentation, as suggested in [2,26].

Table 2 summarizes the rank @1 recognition rates and number of training parameters in deep models on the CUHK03 data set. It can be seen that high-level connections like convolution and fully connected layers result in a large number of parameters to be learned (VGG+CEL and VGG+LDA). By contrast, hybrid systems with Fisher vectors

[2] In each patch, SIFT features and color histograms are ℓ_2 normalized to form a discriminative descriptor vector with length 128×3, and $32 \times 3 \times 3$, respectively, yielding 5376-dim SIFT and 4032-dim color feature per image. PCA is applied on the two features to reduce the dimensionality to be 100-dim, respectively.

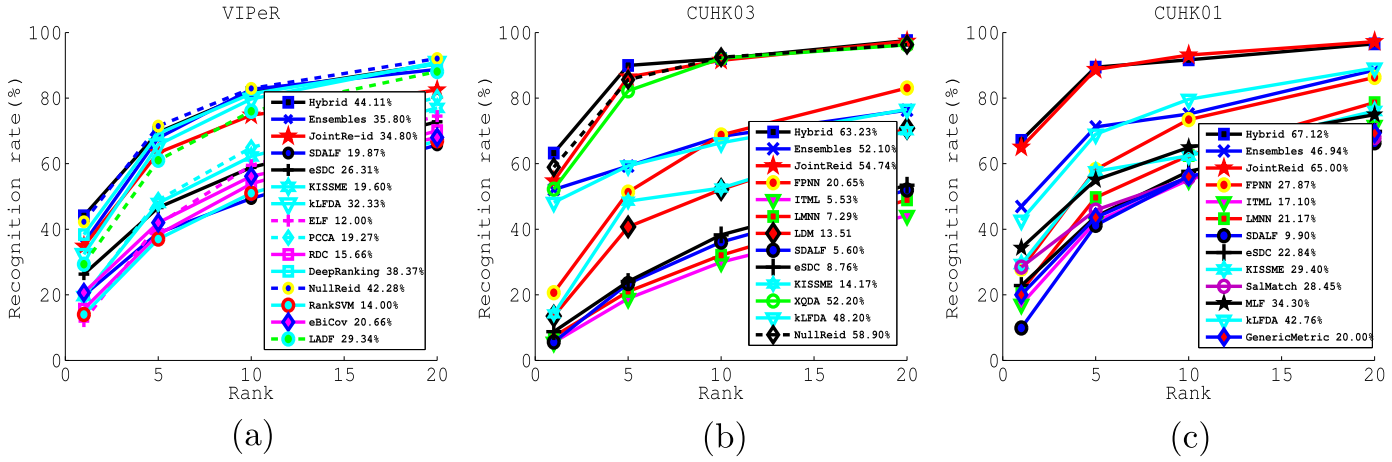

Fig. 4. Performance comparison with state-of-the-art approaches using CMC curves on VIPeR, CUHK03 and CUHK01 data sets.

(DeepFV+CEL, and DeepFV+LDA) can reduce the number of parameters substantially, and thus alleviate the over-fitting to some degree. The performance of DeepFV+LDA is very close to VGG+LDA but with much less parameters in training. Also, deep models with LDA objective outperform those with cross-entropy loss.

5.2.2. Comparison with strong baselines

Since our approach integrates Fisher vector embedding, we perform comparison with recent representative studies in Fisher vectors including deep Fisher networks [34], deep Fisher kernel learning [32], and modeling spatial relationship with Fisher vectors [45]. We implement these baselines using their optimal parameters under their own experimental settings.

- Deep Fisher networks [34]: This method extends standard Fisher vector to a layer architecture suitable for stacking.
- Deep Fisher kernels [32]: This method is to train SVMs with Fisher kernels that jointly learns the classifier weight vector and a suitable image representation.
- Spatial + Fisher vectors [45]: This method is to combine the representation of spatial layout with the use of Fisher kernels to encode the appearance of local features.

Comparison results are presented in Table 3. It can be observed that our approach outperforms these strong baselines in a notable margin. Deep Fisher models i.e., Deep Fisher networks and Deep Fisher kernels show better performance than conventional Fisher vector method i.e., Spatial+Fisher vectors. In particular, deep Fisher kernels has impressive matching rate mainly because it allows end-to-end training to jointly learn parameters and data-dependent representations.

Table 4
Rank @1, @5, @10, @20 recognition rate of various methods on the VIPeR data set. The method of Ensembles* takes two types of features as input: SIFT and LAB patterns.

Method	r @1	r @5	r @10	r @20
Ensembles*[24]	35.80	67.82	82.67	88.74
JointRe-id[25]	34.80	63.32	74.79	82.45
LADF[14]	29.34	61.04	75.98	88.10
SDALF[8]	19.87	38.89	49.37	65.73
eSDC[12]	26.31	46.61	58.86	72.77
KISSME[16]	19.60	48.00	62.20	77.00
kLFDA[21]	32.33	65.78	79.72	90.95
eBiCov[64]	20.66	42.00	56.18	68.00
ELF[11]	12.00	41.50	59.50	74.50
PCCA[19]	19.27	48.89	64.91	80.28
RDC[20]	15.66	38.42	53.86	70.09
RankSVM[23]	14.00	37.00	51.00	67.00
DeepRanking[28]	38.37	69.22	81.33	90.43
NullReid[30]	42.28	71.46	**82.94**	**92.06**
Hybrid	**44.11**	**72.59**	81.66	91.47

Table 5
Rank @1, @5, @10, @20 recognition rate of various methods on the CUHK03 data set.

Method	r @1	r @5	r @10	r @20
Ensembles*[24]	52.10	59.27	68.32	76.30
JointRe-id[25]	54.74	86.42	91.50	97.31
FPNN[2]	20.65	51.32	68.74	83.06
NullReid[30]	58.90	85.60	92.45	96.30
ITML[13]	5.53	18.89	29.96	44.20
LMNN[62]	7.29	21.00	32.06	48.94
LDM[63]	13.51	40.73	52.13	70.81
SDALF[8]	5.60	23.45	36.09	51.96
eSDC[12]	8.76	24.07	38.28	53.44
KISSME[16]	14.17	48.54	52.57	70.03
kLFDA[21]	48.20	59.34	66.38	76.59
XQDA[22]	52.20	82.23	92.14	96.25
Hybrid	**63.23**	**89.95**	**92.73**	**97.55**

5.3. Comparison with state-of-the-art approaches

5.3.1. Evaluation on the VIPeR data set

Fig. 4 (a) shows the CMC curves up to rank @20 recognition rate comparing our method with state-of-the-art approaches. It is obvious that our method delivers the best result in rank @1 recognition. To clearly present the quantized comparison results, we summarize the comparison results on several top ranks in Table 4. Specifically, our method achieves 44.11% rank @1 matching rate, outperforming the

Table 6
Rank @1, @5, @10, @20 recognition rate of various methods on the CUHK01 data set.

Method	r @1	r @5	r @10	r @20
Ensembles*[24]	46.94	71.22	75.15	88.52
JointRe-id[25]	65.00	88.70	**93.12**	**97.20**
SDALF[8]	9.90	41.21	56.00	66.37
MLF[10]	34.30	55.06	64.96	74.94
FPNN[2]	27.87	58.20	73.46	86.31
LMNN[62]	21.17	49.67	62.47	78.62
ITML[13]	17.10	42.31	55.07	71.65
eSDC[12]	22.84	43.89	57.67	69.84
KISSME[16]	29.40	57.67	62.43	76.07
kLFDA[21]	42.76	69.01	79.63	89.18
Generic Metric[3]	20.00	43.58	56.04	69.27
SalMatch[38]	28.45	45.85	55.67	67.95
Hybrid	**67.12**	**89.45**	91.68	96.54

Table 7
Rank @1 and mAP of various methods on the Market-1501 data set.

Method	*r* @1	mAP
SDALF[8]	20.53	8.20
eSDC[12]	33.54	13.54
KISSME[16]	39.35	19.12
kLFDA[21]	44.37	23.14
XQDA[22]	43.79	22.22
Zheng et al.[4]	34.40	14.09
Hybrid	**48.15**	**29.94**

previous best results 42.28% of NullReid [30]. Overall, our method performs best over rank @1, and 5, whereas NullReid is the best at rank @15 and 20. We suspect that training samples in VIPeR are much less to learn parameters of moderate size in our deep hybrid model.

5.3.2. Evaluation on the CUHK03 data set

The CUHK03 data set is larger in scale than VIPeR. We compare our model with state-of-the-art approaches including deep learning methods (FPNN [2], JointRe-id [25]) and metric learning algorithms (Ensembles [24], ITML [13], LMNN [62], LDM [63], KISSME [16], kLFDA [21], XQDA [22]). As shown in Fig. 4 (b) and Table 5, our methods outperforms all competitors against all ranks. In particular, the hybrid architecture achieves a rank @1 rate of 63.23%, outperforming the previous best result reported by JointRe-id [25] by a noticeable margin. This marginal improvement can be attributed to the availability of more training data that are fed to the deep network to learn a data-dependent solution for this specific scenario. Also, CUHK03 provides more shots for each person, which is beneficial to learn feature representations with discriminative distribution parameters (within and between class scatter).

5.3.3. Evaluation on the CUHK01 data set

In this experiment, 100 identifies are used for testing, with the remaining 871 identities used for training. As suggested by FPNN [2] and JointRe-id [25], this setting is suited for deep learning because it uses 90% of the data for training. Fig. 4 (c) compares the performance of our network with previous methods. Our method outperforms the state-of-the-art in this setting with 67.12% in rank @1 matching rate (vs. 65% by the next best method) Table 6.

5.3.4. Evaluation on the market-1501 data set

It is noteworthy that VIPeR, CUHK03, and CUHK01 data sets use regular hand-cropped boxes in the sense that pedestrians are rigidly enclosed under fixed-size bounding boxes, while Market-1501 employs the detector of Deformable Part Model (DPM) [60] such that images undergo extensive pose variation and misalignment. Thus, we conduct experiments on this data set to verify the effectiveness of the proposed method in more realistic situation. Table 7 presents the results, showing that our method outperforms all competitors in terms of rank @1 matching rate and mean average precision (mAP).

5.4. Discussions

5.4.1. Compare with kernel methods

In the standard Fisher vector classification pipeline, supervised learning relies on kernel classifiers [43] where a kernel/linear classifier can be learned in the non-linear embedding space. We remark that our architecture with a single hidden layer has a close link to kernel classifiers in which the first layer can be interpreted as a non-linear mapping Ψ followed by a linear classifier learned in the embedding space. For instance, Ψ is the feature map of arc-cosine kernel of order one if Ψ instantiates random Gaussian projections followed by an reLU non-linearity [66]. This shows a close connection between our model (with a single hidden layer and an reLU non-linearity), and arc-cosine kernel classifiers. Therefore, we compare two alternatives with arc-cosine kernel and RBF kernel. Following [33], we train the non-linear kernel classifiers in the primal by leveraging explicit feature maps. For arc-cosine kernel, we use the feature map involving a random Normal projection followed by an reLU non-linearity. For the RBF kernel, we employ the explicit feature map involving a random Normal projection, a random offset and a cosine non-linearity. For our model, we extract feature outputs of the first and the third hidden layer for which a SVM classifier is trained, respectively. Results are shown in Fig. 5 as a function of rank @ *r* recognition rate. The arc-cosine and RBF kernels perform similarly and worse than our model with a single hidden layer. The SVM classifier trained on the features of three hidden layers consistently achieves superior performance because this latent space is the most linearly separable.

5.4.2. Eigenvalue structure of LDA representations

Recall that the eigenvalues derived from the hidden representations can account for the extent to which the discriminative variance in the direction of the corresponding eigenvectors. In this experiment, we investigate the eigenvalue structure of the general LDA eigenvalue problem. Fig. 6 show the evolution of training and test rank-1 recognition rate of three data sets along with the mean value of eigenvalues in the training epoch. It can be seen that the discriminative potential of the resulting representation is correlated to the magnitude of eigenvalues on the three data sets. This emphasizes the essence of LDA objective function which allows to embed discrimination into the lower-dimensional eigen-space.

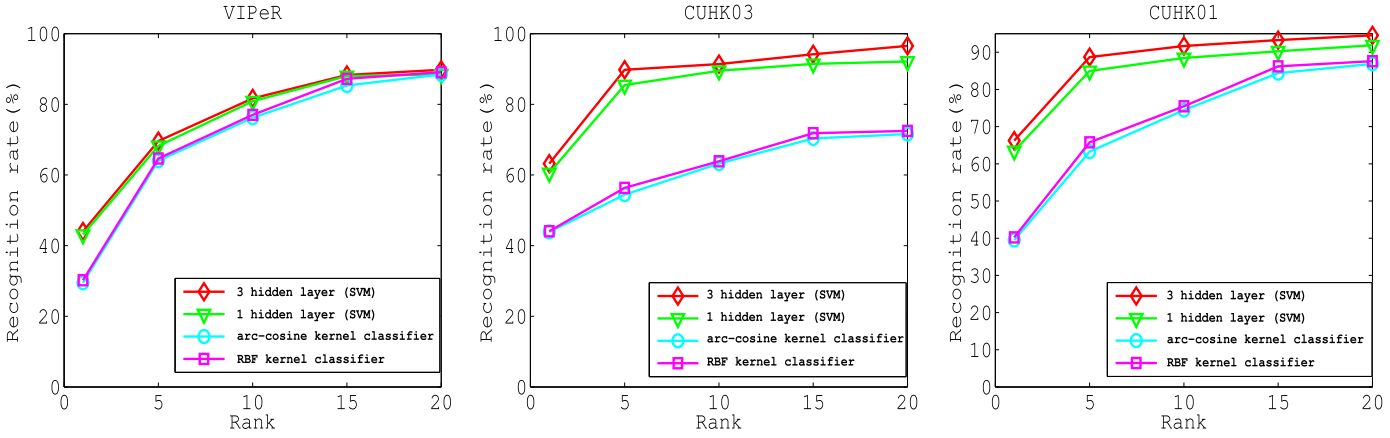

Fig. 5. Compare with kernel methods using CMC curves on VIPeR, CUHK03 and CUHK01 data sets.

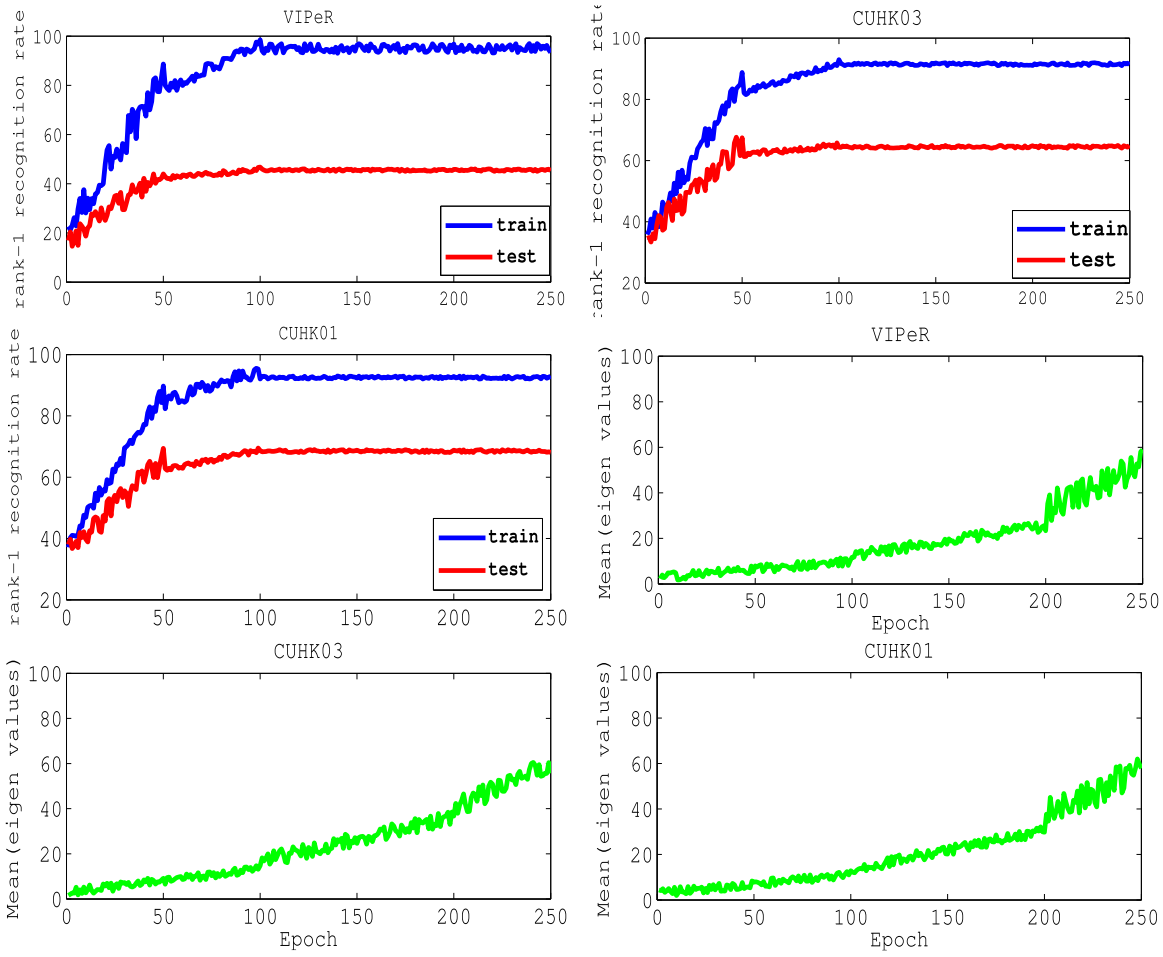

Fig. 6. Study on eigenvalue structure of LDA representations. Each column corresponding to a benchmark shows the evolution of rank @1 recognition rate along with magnitude of discriminative separation in the latent representations.

6. Conclusion

In this paper, we presented a novel deep hybrid architecture to person re-identification. The proposed network composed of Fisher vector and deep neural networks is designed to produce linearly separable hidden representations for pedestrian samples with extensive appearance variations. The network is trained in an end-to-end fashion by optimizing a Linear Discriminant Analysis (LDA) eigen-valued based objective function where LDA based gradients can be back-propagated to update parameters in Fisher vector encoding. We demonstrate the superiority of our method by comparing with state of the arts on four benchmarks in person re-identification.

In future work, we would spend efforts in the reconstruction of real-world complex motion sequences regarding pedestrians based on the success of a re-identification system which can identify the same individual across a distributed camera network. A successful trajectory reconstruction system on pedestrians can benefit a lot to surveillance applications [67,68]. Also, one interesting direction is to generalize the proposed method to be applicable to 3D person re-identification [69].

7. Appendix

This appendix provides explicit expression for the gradients of loss function $\mathcal{L}$ w.r.t. the GMM components.

Let $k, k' = 1, \ldots, K$ be all GMM component indicies, and $d, d' = 1, \ldots, D$ be vector component indicies, the derivatives of the Fisher vectors in Eq. (1) can be derived as

$$\begin{aligned}
\frac{\partial \varphi_{d'}^{k'}}{\partial w^k} &= \frac{\gamma_{k'}\alpha_{d'}^{k'}}{2w^k\sqrt{w^{k'}}}(w^k + \delta_{kk'} - 2\gamma_k)\frac{\partial \varphi_{d'}^{k'}}{\partial \mu_d^k} \\
&= \frac{\gamma_{k'}}{\sigma_d^k\sqrt{w^{k'}}}(\alpha_{d'}^{k'}\alpha_d^k(\delta_{kk'} - \gamma_k) - \delta_{dd'}^{kk'})\frac{\partial \varphi_{d'}^{k'}}{\partial \sigma^k} \\
&= \frac{\gamma_{k'}\alpha_{d'}^{k'}}{\sigma_d^k\sqrt{w^{k'}}}(((\alpha_d^k)^2 - 1)(\delta_{kk'} - \gamma_k) - \delta^{kk'} - dd')\frac{\partial \psi_{d'}^{k'}}{\partial w^k} \\
&= \frac{\gamma_{k'}((\alpha_{d'}^{k'})^2 - 1)}{2w^k\sqrt{2w^{k'}}}(w^k + \delta_{kk'} - 2\gamma_k)\frac{\partial \psi_{d'}^{k'}}{\partial \mu_d^k} \\
&= \frac{\gamma_{k'}\alpha_d^k}{\sigma_d^k\sqrt{2w^{k'}}}(((\alpha_{d'}^{k'})^2 - 1)(\delta_{kk'} - \gamma_k) - 2\delta_{dd'}^{kk'})\frac{\partial \psi_{d'}^{k'}}{\partial \sigma_d^k} \\
&= \frac{\gamma_{k'}}{\sigma_d^k\sqrt{2w^{k'}}}(((\alpha_{d'}^{k'})^2 - 1)((\alpha_d^k)^2 - 1)(\delta_{kk'} - \gamma_k) - 2\delta_{dd'}^{kk'}(\alpha_d^k)^2)
\end{aligned} \tag{15}$$

where $\alpha^k := \frac{x - \mu^k}{\sigma^k}$, and $\delta_{ab} = 1$ if $a{=}b$, and 0 otherwise, and $\delta_{cd}^{ab} = \delta_{ab}\delta_{cd}$.

Stacking the above expressions and averaging them over all descriptors in an image X, we obtain the gradient of the unnormalized per-image Fisher vector $\nabla\Phi(X)$. Then the gradient of the normalized Fisher vector in Eq. (4) can be computed by the chain rule:

$$\nabla\overline{\Phi}_d = \left(\frac{\nabla\Phi_d}{2\Phi_d} - \frac{\sum_{d'} sign(\Phi_{d'})\nabla\Phi_{d'}}{2 \parallel \Phi \parallel_{L^1}}\right)\overline{\Phi}_d, \tag{16}$$

where the gradients act w.r.t. all parameters, $G = (w, \mu, \Sigma)$. Hence, the gradient of the loss term follows by applying the chain rule:

$$\frac{\partial}{\partial \boldsymbol{X}}\frac{1}{m}\sum_{i=1}^{m} v_i = \frac{1}{m}\sum_{i=1}^{m}\frac{\partial v_i}{\partial \boldsymbol{X}} = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{e}_i^T\left(\frac{\partial \boldsymbol{S}_b}{\partial \boldsymbol{X}}\cdot\frac{\partial \boldsymbol{X}}{\partial \Phi_d}\cdot\nabla\overline{\Phi}_d - \frac{\partial \boldsymbol{S}_w}{\partial \boldsymbol{X}}\cdot\frac{\partial \boldsymbol{X}}{\partial \Phi_d}\cdot\nabla\overline{\Phi}_d\right)\boldsymbol{e}_i. \tag{17}$$

The gradients computed in Eq. (17) allow us to apply backpropagation, and train the hybrid network in an end-to-end fashion. Also, a training sample X goes feed-forwadly through the network by Fisher vector encoding $\Phi_G(X)$ (parameterized by G), and supervised layers $g_\Theta(\Phi_G(X))$ (parameterized by Θ), such that the hidden representation can be outputed $\boldsymbol{X}$ accordingly. Although computing the gradients with the above expressions is computationally costly, we are lucky to have some strategies to accelerate it. For instance, as each expression contains a term $\gamma_{k'}$, it is suggested to compute the gradient terms only if this value exceeds a threshold [32], e.g., 10^{-5}. Another promising speedup can be achieved by subsampling the number of descriptors used from each image to form the gradient. In our experiments, we only use a fraction of 10% without noticeable loss, however.

References

[1] D. Gray, S. Brennan, H. Tao, Evaluating appearance models for recognition, reacquisition, and tracking, in: Proceedings of the International'l. Workshop on Perf. Eval. of Track. and Surv'l., 2007.

[2] W. Li, R. Zhao, X. Tang, X. Wang, Deepreid: Deep filter pairing neural network for person re-identification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 152–159.

[3] W. Li, R. Zhao, X. Wang, Human reidentification with transferred metric learning, in: Proceedings of the Asian conference on Computer Vision, 2012, pp. 31–44.

[4] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, Q. Tian, Scalable person re-identification: A benchmark, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1116–1124.

[5] S. Gong, M. Cristani, S. Yan, C.C. Loy, Person Re-Identification, Springer, London, 2014.

[6] A. Bedagkar-Gala, S.K. Shah, A survey of approaches and trends in person re-identification, Image Vis. Comput. 32 (4) (2014) 270–286.

[7] D.S. Cheng, M. Cristani, M. Stoppa, L. Bazzani, V. Murino, Custom pictorial structures for re-identification, in: Proceedings of the British Machine Vision Conference, 2011, pp. 68.1–68.11.

[8] M. Farenzena, L. Bazzani, A. Perina, V. Murino, M. Cristani, Person re-identification by symmetry-driven accumulation of local features, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 2360–2367.

[9] N. Gheissari, T.B. Sebastian, R. Hartley, Person reidentification using spatiotemporal appearance, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 1–8.

[10] R. Zhao, W. Ouyang, X. Wang, Learning mid-level filters for person re-identfication, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 144–151.

[11] D. Gray, H. Tao, Viewpoint invariant pedestrian recognition with an ensemble of localized features, in: Proceedings of the European Conference on Computer Vision, 2008, pp. 262–275.

[12] R. Zhao, W. Ouyang, X. Wang, Unsupervised salience learning for person re-identification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3586–3593.

[13] J.V. Davis, B. Kulis, P. Jain, S. Sra, I.S. Dhillon, Information-theoretic metric learning, in: Proceedings of the International Conference on Machine Learning, 2007, pp. 209–216.

[14] Z. Li, S. Chang, F. Liang, T.S. Huang, L. Cao, J. Smith, Learning locally-adaptive decision functions for person verification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3610–3617.

[15] L. Ma, X. Yang, D. Tao, Person re-identification over camera networks using multi-task distance metric learning, IEEE Trans. Image Process. 23 (8) (2014) 3656–3670.

[16] M. Kostinger, M. Hirzer, P. Wohlhart, P.M. Roth, H. Bischof, Large scale metric learning from equivalence constraints, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2288–2295.

[17] Y. Wu, M. Mukunoki, T. Funatomi, M. Minoh, S. Lao, Optimizing mean reciprocal rank for person re-identification, in: Proceedings of the Advanced Video and Signal-Based Surveillance, 2011, pp. 408–413.

[18] K. Weinberger, J. Blitzer, L. Saul, Distance metric learning for large margin nearest neighbor classification, in: Proceedings of the Advances in Neural Information Processing Systems, 2005, pp. 1473–1480.

[19] A. Mignon, F. Jurie, Pcca: a new approach for distance learning from sparse pairwise constraints, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2666–2672.

[20] W. Zheng, S. Gong, T. Xiang, Re-identification by relative distance comparision, IEEE Trans. Pattern Anal. Mach. Intell. 35 (3) (2013) 653–668.

[21] F. Xiong, M. Gou, O. Camps, M. Sznaier, Person re-identification using kernel-based metric learning methods, in: Proceedings of the European Conference on Computer Vision, 2014, pp. 1–16.

[22] S. Liao, Y. Hu, X. Zhu, S.Z. Li, Person re-identification by local maximal occurrence representation and metric learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2197–2206.

[23] B. Prosser, W. Zheng, S. Gong, T. Xiang, Person re-identification by support vector ranking, in: Proceedings of the British Machine Vision Conference, 2010, pp. 21.1–21.11.

[24] S. Paisitkriangkrai, C. Shen, A. van den Hengel, Learning to rank in person re-identification with metric ensembles, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1846–1855.

[25] E. Ahmed, M. Jones, T.K. Marks, An improved deep learning architecture for person re-identification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3908–3916.

[26] L. Wu, C. Shen, A. van den Hengel, Personnet. Person re-identification with deep convolutional neural networks, in: Proceedings of the CoRR abs/1601.07255, 2016.

[27] D. Yi, Z. Lei, S. Liao, S.Z. Li, Deep metric learning for person re-identification, in: Proceedings of the International Conference on Pattern Recognition, 2014, pp. 34–39.

[28] S.-Z. Chen, C.-C. Guo, J.-H. Lai, Deep ranking for re-identification via joint representation learning, IEEE Trans. Image Process. 25 (5) (2016) 2353–2367.

[29] L.-F. Chen, H.-Y.M. Liao, M.-T. Ko, J.-C. Lin, G.-J. Yu, A new lda-based face recognition system which can solve the small sample size problem, Pattern Recognit. 33 (10) (2000) 1713–1726.

[30] L. Zhang, T. Xiang, S. Gong, Learning a discriminative null space for person re-identification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[31] A. Krizhevsky, I. Sutskever, G.E. Hinton., Imagenet classification with deep convolutional neural networks, in: Proceedings of the Advances in Neural Information Processing Systems, 2012, pp. 1106–1114.

[32] V. Sydorov, M. Sakurada, C.H. Lampert, Deep fisher kernels - end to end learning of the fisher kernel gmm parameters, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1402–1409.

[33] F. Perronnin, D. Larlus, Fisher vectors meet neural networks: A hybrid classification architecture, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3743–3752.

[34] K. Simonyan, A. Vedaldi, A. Zisserman, Deep fisher networks for large-scale image classification, in: Proceedings of the Advances in Neural Information Processing Systems, 2013, pp. 1–9.

[35] G. Andrew, R. Arora, J. Bilmes, K. Livescu, Deep canonical correlation analysis, in: Proceedings of the International Conference on Machine Learning, 2013, pp. 1247–1255.

[36] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, Y. Ma, Pcanet: a simple deep learning baseline for image classification?, IEEE Trans. Image Process. 24 (2) (2015) 5017–5032.

[37] R.A. Fisher, The use of multiple measurements in taxonomic problems, Ann. Eugen. 7 (2) (1936) 179–188.

[38] R. Zhao, W. Ouyang, X. Wang, Person re-identification by salience matching, in: Proceedings of theIEEE International Conference on Computer Vision, 2013, pp. 2528–2535.

[39] S. Pedagadi, J. Orwell, S. Velastin, B. Boghossian, Local fisher discriminant analysis for pedestrian re-identification, in:Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3318–3325.

[40] D. Lowe, Object recognition from lcoal scale-invariant features, in: Proceedings of the IEEE International Conference on Computer Vision, 1999, pp. 1150–1157.

[41] G. Csurka, C.R. Dance, L. Fan, J. Willamowski, C. Bray, Visual categorization with bags of keypoints, in: Proceedings of the Workshop on statistical learning in computer vision, 2004, pp. 1–22.

[42] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1794–1801.

[43] J. Sanchez, F. Perronnin, T. Mensink, J. Verbeek, Image classification with the fisher vector: theory and practice, Int. J. Comput. Vis. 105 (3) (2013) 222–245.

[44] B. Hariharan, J. Malik, D. Ramanan, Discriminative decorrelation for clustering and classification, in: Proceedings of the European Conference on Computer Vision, 2012, pp. 459–472.

[45] J. Krapac, J. Verbeek, F. Jurie, Modeling spatial layout with fisher vectors for image categorization, in: Proceedings of the IEEE International Conference on Computer Vision, 2011, pp. 1487–1494.

[46] J. Sanchez, F. Perronnin, T. de Campos, Modeling the spatial layout of images beyond spatial pyramids, Pattern Recognit. Lett. 33 (16) (2012) 2216–2223.

[47] T.S. Jaakkola, D. Haussler, Exploiting generative models in discriminative classifiers, in: Proceedings of the Advances in Neural Information Processing Systems, 1998, pp. 487–493.

[48] K. Chatfield, V. Lempitsky, A. Vedaldi, A. Zisserman, The devil is in the details: an evaluation of recent feature encoding methods, in: Proceedings of the British Machine Vision Conference, 2011, pp. 76.1–76.12.

[49] Y. Wang, W. Zhang, L. Wu, X. Lin, M. Fang, S. Pan, Iterative views agreement: An iterative low-rank based structured optimization method to multi-view spectral clustering, in: Proceedings of the IJCAI, 2016.

[50] Y. Wang, X. Lin, L. Wu, W. Zhang, Q. Zhang, X. Huang, Robust subspace clustering for multi-view data by exploiting correlation consensus, IEEE Trans. Image Process. 24 (11) (2015) 3939–3949.

[51] F. Perronnin, J. Snchez, T. Mensink, Improving the fisher kernel for large-scale image classification, in: Proceedings of the European Conference on Computer Vision, 2010, pp. 143–156.

[52] A. Stuhlsatz, J. Lippel, T. Zielke, Feature extraction with deep neural networks by a generalized discriminant analysis, IEEE Trans. Neural Netw. Learn. Syst. 23 (4)

(2012) 596–608.
[53] D.H. Ackley, G.E. Hinton, T.J. Sejnowski, A learning algorithm for boltzmann machines, Conginitive Sci. 9 (1) (1985) 147–169.
[54] H. Osman, M.M. Fahmy, On the discriminatory power of adaptive feed-forward layered networks, IEEE Trans. Pattern Anal. Mach. Intell. 16 (8) (1994) 837–842.
[55] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: arXiv preprint 2014 arXiv:1409.1556
[56] M. Dorfer, R. Kelz, G. Widmer, Deep linear discriminant analysis, in: Proceedings of the International Conference on Learning Representations, 2016.
[57] J.H. Friedman, Regularized discriminant analysis, J. Am. Stat. Assoc. 84 (405) (1989) 165–175.
[58] J. de Leeuw, Derivatives of generalized eigen systems with applications, 2007.
[59] G.E. Hinton, S. Osindero, Y.-W. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (7) (2006) 1527–1554.
[60] P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, IEEE Trans. Pattern Anal. Mach. Intell. 32 (9) (2010) 1627–1645.
[61] B. Huang, J. Chen, Y. Wang, C. Liang, Z. Wang, K. Sun, Sparsity-based occlusion handling method for person re-identification, in: Proceedings of the Multimedia Modeling, 2015, pp. 61–73.
[62] M. Hirzer, P. Roth, H. Bischof, Person re-identification by efficient imposter-based metric learning, in: Proceedings of the IEEE Proceedings of the Ninth International Conference on Advanced Video and Signal-Based Surveillance, 2012, pp. 203–208.
[63] M. Guillaumin, J. Verbeek, C. Schmid, Is that you? Metric learning approaches for face identification, in: Proceedings of the IEEE International Conference on Computer Vision, 2009, pp. 498–505.
[64] B. Ma, Y. Su, F. Jurie, Bicov: A novel image representation for person re-identification and face verification, in: Proceedings of the British Machine Vision Conference, 2012, pp. 57.1–57.11.
[65] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: Proceedings of the CoRR, abs/1502.03167, 2015.
[66] Y. Cho, L. Saul, Kernel methods for deep learning, in: Proceedings of the Advances in Neural Information Processing Systems, 2009, pp. 342–350.
[67] Y. Zhu, S. Lucey, Convolutional sparse coding for trajectory reconstruction, IEEE Trans. Pattern Anal. Mach. Intell. 37 (3) (2015) 529–540.
[68] Y. Zhu, D. Huang, F.D.L. Torre, S. Lucey, Complex non-rigid motion 3d reconstruction by union of subspaces, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1542–1549.
[69] V.O. Andersson, R.M. Araujo, Full body person identification using the kinect sensor, in: IEEE Proceedings of the 26th International Conference on Tools with Artificial Intelligence, 2014, pp. 627–633.

Lin Wu is currently an ARC Senior Research Associate at School of Computer Science, The University of Adelaide, Australia. She obtained a Ph.D. degree from School of Computer Science and Engineering, The University of New South Wales, Sydney, Australia in 2014. Her research interest cover deep learning, computer vision and multimedia data analysis.

Chunhua Shen studied at Nanjing University, at Australian National University, and received the Ph.D. degree from the University of Adelaide. He is a professor of computer science at the University of Adelaide. His research interests are in the intersection of computer vision and statistical machine learning. In 2012, he received the Australian Research Council Future fellowship.

Anton van den Hengel received the bachelor of mathematical science degree, bachelor of laws degree, master's degree in computer science, and the Ph.D. degree in computer vision from The University of Adelaide in 1991, 1993, 1994, and 2000, respectively. He is a professor and the founding director of the Australian Centre for Visual Technologies, at the University of Adelaide, focusing on innovation in the production and analysis of visual digital media.