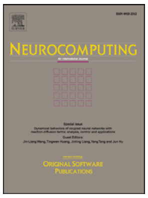
Neurocomputing 241 (2017) 191–203

Contents lists available at ScienceDirect

Neurocomputing

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



DeepDiff: Learning deep difference features on human body parts for person re-identification

Yan Huang[a,b], Hao Sheng[a,b,*], Yanwei Zheng[a], Zhang Xiong[a]

[a] *State Key Laboratory of Software Development Environment, School of Computer Science and Engineering, Beihang University, Beijing, PR China*
[b] *Shenzhen Key Laboratory of Data Vitalization, Research Institute in Shenzhen, Beihang University, Shenzhen, PR China*

ARTICLE INFO

Article history:
Received 14 June 2016
Revised 30 December 2016
Accepted 17 February 2017
Available online 24 February 2017

Communicated by M. Bianchini

Keywords:
Person re-identification
Neural network
Body parts
Difference features
Smart surveillance

ABSTRACT

Person re-identification is an important part of smart surveillance systems which match people across different scenarios. The most challenging aspect of person re-identification is to design robust features and similarity models that reduce the impact of viewpoints, lightings, background clutters, camera settings, occlusions, and pedestrian poses under different scenarios. DeepDiff is a learning method proposed in this paper which uses deep neural networks to identify the different features of various human body parts, and then evaluate the similarities between those corresponding parts. Given those two corresponding parts, we propose three subnets that show deep difference using original data, feature maps, and spatial variations. We focus on a part-based method which introduces a pyramid partition architecture through different partition granularities on human images. In order to determine whether those two parts belong to the same identity, we utilize two combinations of our three subnets. Lastly, we present a part-based integration validation method so as to achieve better performance. The DeepDiff's effectiveness is validated on three public datasets, including: CUHK03 (labeled, detected), VIPeR and CUHK01 (100 and 486 identities settings). The experimental study shows that the proposed DeepDiff model has promising potential.

1. Introduction

The demand to analyze pedestrians using intelligent surveillance systems is growing alongside the rapid development of computer vision technology. Between person detection, tracking, and re-identification, the latter underpins crucial applications such as long-term multi-camera tracking and forensic searches [1]. Person re-identification gained a lot of interest in the recent years [2–7], because it addresses regarding matching people across disjointed camera views in a multi-camera system, or identifying the same identity from different time periods within the same camera view. It can also be formulated to match persons between two sets (“probe set” and “gallery set” [8]), which contain individuals captured in different scenarios (different views or time periods); the person re-identification algorithm calculates the similarities between any two persons respectively taken from these two sets.

Different scenarios provide complex variations (e.g., viewpoints, lightings, background clutters, camera settings, occlusions and pedestrian poses (see Fig. 1)). As such, person re-identification is challenging since the biological characteristics (e.g., gait, face and physical appearance) are hardly used on low resolution images. To design person re-identification system, two fundamental modules are necessary: robust feature extraction and similarity measure. The former can reduce the impact of variations under different scenarios, which is the foundation of similarity measure. Unlike biological characteristics, the features used in person re-identification range from color histogram [9], texture character [10], body structure representation [11] or the combination of multiple features [3]. Given these handcrafted features, similarity measure algorithms are used to identify the distance between person images with the same identity (D_{same}) or different identities (D_{diff}) under certain conditions, such as $D_{same} > D_{diff}$. Although this proves effective, it depends on the quality of the selected features which requires a great deal of domain knowledge and experience [12].

The combined handcrafted features and similarity measure algorithms try to find the differences amongst different images so as to identify whether or not two images belong to the same person. However, this kind of combination is ineffective in sharing

[*] Corresponding author at: State Key Laboratory of Software Development Environment, School of Computer Science and Engineering, Beihang University, Beijing, PR China.
E-mail address: shenghao@buaa.edu.cn (H. Sheng).

http://dx.doi.org/10.1016/j.neucom.2017.02.055
0925-2312/© 2017 Elsevier B.V. All rights reserved.

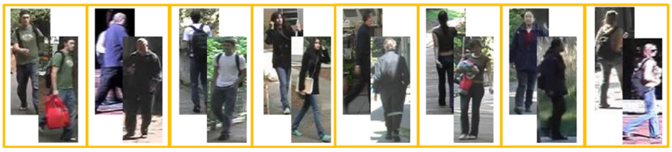


Fig. 1. Typical examples of person images captured from different camera views. Each box contains two images with the same identity. Huge variations are illustrated on each person.

these differences between two modules, because they are not integrally optimized and the information on those differences cannot be propagated to each other. To deal with this problem, the deep neural network is the most promising for the overall design because it establishes automatic interaction between the two modules [5], and has made a large contribution [5,7,13,14]. For example, the convolutional neural network (CNN), which is widely used in the re-identification problem, demonstrated its effectiveness in a variety of works including DeepID (V1, V2, V2+) [14–16] for face recognition, FPNN [5], DML [17], and IDLA [7] for person re-identification. These approaches were mainly designed for the whole image, and as such, the deep neural network's focus on body parts is originally disregarded for person re-identification. Since there is the potential for discrimination amongst different body parts, deep difference information could be exploited on local regions using partitioned parts rather than integrally trained parts. As such, this paper presents a deep difference featuring a human body parts method (DeepDiff) used for person re-identification.

In DeepDiff, a similarity value can be propagated through forward propagation once a pair of corresponding body parts is given as input. By integrating the similarity values of all parts, DeepDiff can then reach a final decision to determine whether or not the two images depict the same person. This is a three-fold process:

First, three deep neural subnets are presented to simultaneously learn deep difference features using original data, feature maps and spatial variations from DeepDiff: (1) a cross-data difference subnet is proposed to learn deep difference features from the difference of original data by using convolutional networks; (2) since variations occur under different scenarios on original data (see Fig. 1), a cross-map difference subnet is introduced to learn deep difference features using the difference between feature maps which are optimized in order to reduce the impact of these variations during the training process; (3) in most cases, spatial variations of viewpoints, occlusions and pedestrian poses occur under different scenarios, especially when person images are cropped by the pedestrian detector. Deep difference features derived from these images by comparing corresponding positions cannot effectively reduce spatial variations. As such, our DeepDiff presents a cross-space difference subnet to learn deep difference features derived from spatial variations by using a spatial traversal manner on local regions.

Second, our part-based approach needs a pyramid partition architecture that uses different image partition granularities. Generally, parts with coarse granularity partition cover more information, but when comparing the numerous details that exist between two parts, it can be a time-consuming process. In contrast, parts with fine granularity partition contain less information, but details can be exploited to extract robust difference features between local regions. Therefore, a pyramid partition architecture is presented in DeepDiff in order to balance the two partition modes, since it utilizes the different combinations of the three subnets on different partition granularities.

Third, an integration validation method is introduced in DeepDiff in order to achieve better performance. Using the part-based approach, we find that combined parts are more powerful when they are validated integrally rather than separately. In order to validate the discriminative power of different parts, our experiment presents an integration validation method based on body parts.

The rest of the paper is organized as follows: Section 2 reviews the related works on person re-identification. Section 3 introduces the three deep subnets, followed by a description of the pyramid partition architecture in Section 4. Section 5 presents the training and integration validation strategy. The experimental performance and analysis are given in Section 6. Finally, we draw conclusions and put forward future works in Section 7.

2. Related works

2.1. Overview of traditional methods

Typically, person re-identification methods include two modules: features extraction and similarity measure. In previous works, features used in person re-identification systems were mainly manually designed, and were partially invariant to variations under different scenarios. Amongst the diverse features that exist, color and texture-based features were the most popular, e.g., color histograms (RGB, HSV and LAB) [2,4,6], Gabor [18], and Schmid texture features [2,6]. Aside from that, the local features extraction included Maximally Stable Color Regions (MSCR), Recurrent High-Structured Patches (RHSP), and Spatial Covariance Regions (SCR) on salient parts by adopting perceptual principles of symmetry and asymmetry, as introduced in [3]. Besides, Ma et al. [18] proposed a covariance descriptor extracted from magnitude images, which were obtained from HSV color space images and computed with Gabor filter through different filter sizes. A Local Maximal Occurrence (LOMO) feature representation was introduced in [19], which analyzed the horizontal occurrence of local features and maximized the occurrence in order to make a stable representation against viewpoints variations. Local patch methods demonstrate its effectiveness in recent works [20–22]. Therefore, a widely used patch-based feature dColorSIFT that was robust to variations of viewpoints and illumination, was also proposed in [20].

Numerous similarity measure approaches were proposed in an effort to better understand the difference amongst features. Zheng et al. [4] utilized a metric learning method based on Mahalanobis distance. Li et al. [23] proposed a Locally Adaptive Decision Function through a locally adaptive thresholding rule. Paisitkriangkrai et al. [24] sought an optimal combination of handcrafted features which could adapt to different datasets by using two structured learning based approaches. A bi-directional weighing mechanism was used to compare the salient regions between people [20], since images of the same person were more likely to have similar salience distributions than those of different people. A mixture of linear similarity functions which was used to discover different patch matching patterns was introduced in [25]. The Regularized Pairwise Constrained Component Analysis, kernel Local Fisher Discriminant Analysis, and Marginal Fisher Analysis demonstrated the discriminative power that was used to find the difference between features in [26].

In an effort to cope with spatial variations, several works were proposed to conduct spatial alignment on people's images across different views [20,27–29]. In [20], Zhao et al. proposed an unsupervised matching approach for person re-identification; in order to prevent spatial variations in a vertical space, a relaxed spatial constraint was employed on horizontal stripes between images. To model spatial variations, Shen et al. [27] described the matching of local regions by using a "one-to-many" graph, followed by a global constraint that was integrated into the local region matching process. Huang et al. [28] compared the person re-identification problem to a tree-matching problem and used a hierarchical bipartite graph matching method so as to cope with the spatial variations between person images. In an effort to break down the spatial variations problem into sub-regions, Chen et al. [29] used a polynomial feature map to simultaneously describe different local regions with a unified learning framework. All of the aforementioned methods utilized the local regions as constraints so as to obtain accurate matches between images. However, in order to actually construct the spatial matching model, each method required in-depth domain knowledge and experiences. In our method, we coped with spatial variations using a cross-space difference subnet to model these variations within an end-to-end fashion on local body parts, which achieved good performance on several datasets.

The effectiveness was better demonstrated in the above traditional approaches, but they heavily relied on the quality of the features, primarily because the two modules of features extraction and similarity measures were handled separately or sequentially, and the difference information could not be propagated with each other. As such, the deep learning methods that established the automatic interaction between the two modules were studied.

2.2. *Deep learning for person re-identification*

Several previous works were proposed to demonstrate the effectiveness of deep learning approaches [5,7,17,30–34]. Where deep features and hash functions were simultaneously optimized in the training process, Zhang et al. [31] utilized a deep convolutional neural network in order to train models to retrieve images and re-identify persons in an end-to-end fashion. Liu et al. [30] presented a Set-Label Model which applied the Deep Belief Network and Neighborhood Component Analysis on concatenated features in order to improve the performance of person re-identification. Wu et al. [32] introduced handcrafted features as a complement to convolution neural network features in a constrained manner. In addition, Wu et al. [34] proposed a hybrid architecture which combined deep neural networks and Fisher vectors to understand person representations in a latent space where features were linearly separable. Li et al. [5] used a deep neural network to find the differences between horizontal stripes divided from feature maps of two input images. The network in [5] produced an interaction between the two input images in the shallow layer, and the difference features were generated in deeper layers through a maxout-grouping layer. Similar to [5], an improved deep learning architecture was presented in [7]; this utilized a neighborhood difference layer to compare feature maps on nearby patches, followed by a patch summary layer that summarized the difference by producing a holistic representation on each block shown on neighborhood difference maps. After that, Wu et al. [33] improved the neural network in [7] by increasing the depth of network and using smaller convolution filters with a large number of learnable parameters. Although it achieved good results on big datasets, it led to overfitting in small ones.

These approaches evaluated the similarities by utilizing the CNN and making the use of spatial correspondence relationships for difference features extraction. However, they did not fully exploit the spatial variations caused by variations of viewpoints, occlusions, pedestrian poses, and the discriminative power of different body parts. This was in part due to the fact that the methods were used on whole images, so there was little attention given to the comparison between different body parts.

In addition to our part-based person re-identification method, other deep learning methods working on body parts were also discussed in [7,17,35]. To the best of our knowledge, Yi et al. [17] was the first to handle the person re-identification problem using a part-based deep learning method; it separated each image into three overlapped parts and used a siamese convolutional network to calculate the similarity between the two parts. Following this, Ahmed et al. [7] attempted to train the neural network on five overlapped body parts which were equally divided from each person's image. Since they used a neighborhood difference in their network, their method was not competitive enough to deal with spatial variations in special datasets e.g., CUHK03 (detected) [5]. Cheng et al. [35] used multiple channels based on CNN in order to jointly train the global body and body parts feature on each input image. The combination of the improved triple loss function provided a good performance, and this was shown on different scale datasets by different network configurations (they used two kinds of networks with different depth on small and large dataset, respectively). Since it was difficult to determine the scale of some datasets, their method needed artificial experiences to configure the structure of the network among them.

In this paper, we apply a part-based DeepDiff method in order to demonstrate the effectiveness of human body parts for person re-identification. With parts partition, local spatial variations are well-handled by the proposed three deep subnets. These subnets simultaneously learn difference features from original data, feature maps and spatial variations, and are powerful enough to evaluate the similarity between the two parts. In addition, the pyramid partition architecture is proposed for our part-based neural networks through different partition granularities. Lastly, an integration validation method is introduced in DeepDiff in order to achieve better performance.

3. Deep neural subnets for difference features learning

In DeepDiff, we initially propose three neural subnets to extract deep difference features. Given the two corresponding parts (RGB images, denoted as P and $P^{'}$), the three subnets are used to learn the difference features from original data, feature maps and spatial variations. The features learned from these three subnets are used to determine whether or not the two parts depict the same person by utilizing combinations of subnets among the different partition granularities of body parts (the details are introduced in Section 4).

Fig. 2 illustrates the architecture of the three subnets, which are individually titled as a "cross-data difference subnet" (see Fig. 2(a)), a "cross-maps difference subnet" (see Fig. 2(b)), and a "cross-space difference subnet" (see Fig. 2(c)); they are respectively denoted as $\mathbb{N}_{dat}$, $\mathbb{N}_{map}$ and $\mathbb{N}_{spc}$. The Roman numerals I, II, III and IV represent the index of each layer in subnets. The kth feature map of idth layer belongs to netth subnet is formulated as $f^{net}_{id,k}$, where $net \in \{\mathbb{N}_{dat}, \mathbb{N}_{map}, \mathbb{N}_{spc}\}$ and $id \in \{I, II, III, IV\}$. In addition, x and y represent the row and column index in each feature map. The details of these three subnets are explained in the following sections.

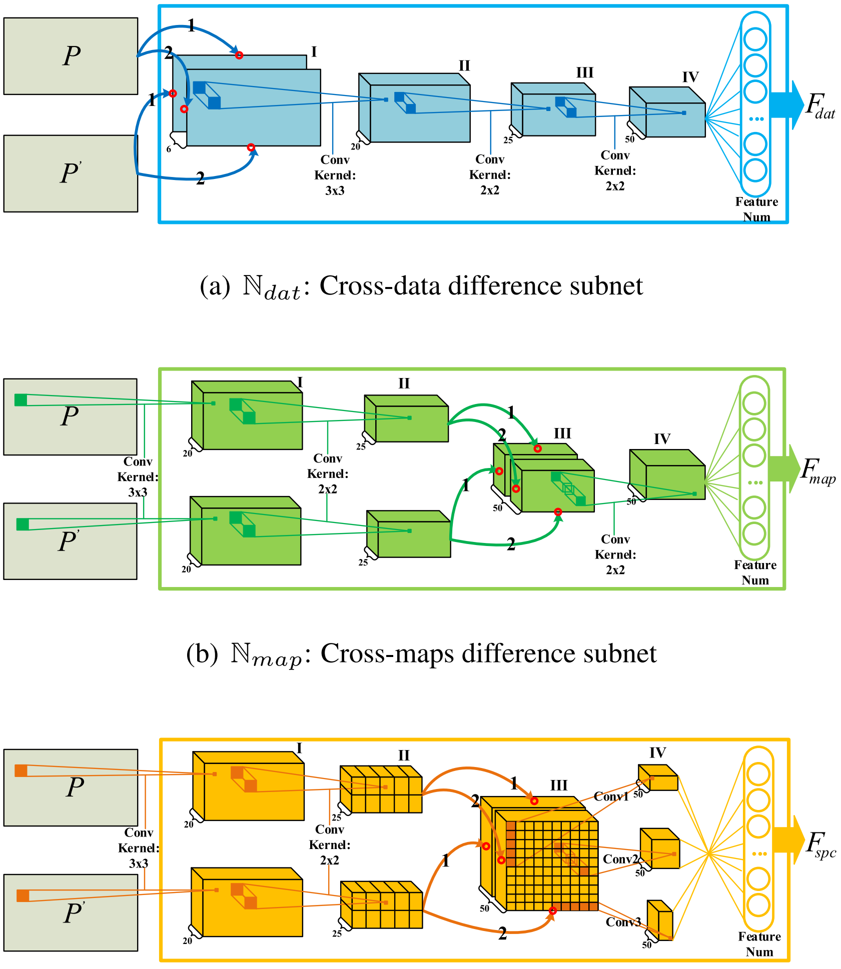
3.1. *Cross-data difference subnet*

In order to determine whether or not the two corresponding input parts belong to the same person, a cross-data difference subnet is proposed to learn the deep difference features from the original data by using convolutional networks. As shown in Fig. 2(a), given the two corresponding parts P and $P^{'}$, the feature maps in Ith layer, which represent the difference of the original data, are computed through element-wise subtraction between the two parts. More precisely:

$$\begin{aligned} f^{\mathbb{N}_{dat}}_{I,[1,3]}(x,y) &= P_{[1,3]}(x,y) - P^{'}_{[1,3]}(x,y) \\ f^{\mathbb{N}_{dat}}_{I,[4,6]}(x,y) &= P^{'}_{[1,3]}(x,y) - P_{[1,3]}(x,y) \end{aligned} \tag{1}$$

Then, $f^{\mathbb{N}_{dat}}_{I,[1,6]}$ is passed through three convolution layers with max-pooling (from layer I to layer IV):

$$f^{\mathbb{N}_{dat}}_{id,k_2} = \max_{0 \leqslant m,n \leqslant s} \left\{ \left(\sum_{k1} w^{r}_{k_1 k_2} * f^{\mathbb{N}_{dat}(r)}_{id\text{-}1,k_1} + b^{k_2(r)} \right)^{k_2}_{i \cdot s+m, j \cdot s+n} \right\} \tag{2}$$



(a) $\mathbb{N}_{dat}$: Cross-data difference subnet

(b) $\mathbb{N}_{map}$: Cross-maps difference subnet

(c) $\mathbb{N}_{spc}$: Cross-space difference subnet

Fig. 2. Three subnets $\mathbb{N}_{dat}$, $\mathbb{N}_{map}$ and $\mathbb{N}_{spc}$ are used to extract deep difference features. Paired corresponding parts are passed through the three subnets to learn deep difference features F_{dat}, F_{map} and F_{spc}, respectively. These features are used to determine whether or not the two parts belong to the same person.

Table 1
Parameters setting of $\mathbb{N}_{dat}$. Each row illustrates the parameters (kernel size and stride) of a convolution layer followed by a max-pooling layer between two adjacent layers.

$\mathbb{N}_{dat}$	Convolution		Max-pooling	
	Kernel size	Stride	Kernel size	Stride
$I \rightarrow II$	3 × 3 × 6 × 20	1	2 × 2	2
$II \rightarrow III$	2 × 2 × 20 × 25	1	2 × 2	2
$III \rightarrow IV$	2 × 2 × 25 × 50	1	2 × 2	2

where $f^{\mathbb{N}_{dat}(r)}_{id\text{-}1,k_1}$ and $f^{\mathbb{N}_{dat}}_{id,k_2}$ are the k_1th input map of layer id-1 and k_2th output map of layer id, respectively. $w_{k_1k_2}$ is the convolution kernel between k_1th input map and k_2th output map. $*$ is the convolution operation. b^{k_2} is the bias of the k_2th output map. r indicates a local region where weights are shared. The neuron in the k_2th output map pools over an $s \times s$ non-overlapping local region in the k_2th convolution map. The details of parameters setting of the three convolution layers with max-pooling are illustrated in Table 1.

Finally, a fully connected layer with ReLU nonlinearity activation function $y = max\{0, x\}$ (denoted as ReLU) is utilized after IVth layer, which is formulated as:

$$F_{dat}(j) = max\left(0, \sum_i f^{\mathbb{N}_{dat}}_{IV,k}(i) \cdot w_{i,j} + b_j\right) \tag{3}$$

where i and j denote the ith neuron in $f^{\mathbb{N}_{dat}}_{IV,k}$ and jth feature in F_{dat}, respectively. $w_{i,\,j}$ and b_j represent weights and the bias term of the fully connected layer. Moreover, the operations of convolution with max-pooling and fully connected with ReLU (Eqs. (2) and (3)) also apply to $\mathbb{N}_{map}$ and $\mathbb{N}_{spc}$.

3.2. Cross-map difference subnet

As shown in Fig. 1, variations take place under different scenarios on original data. Therefore, a cross-maps difference subnet is introduced to learn deep difference features on feature maps which are optimized to reduce the impact of these variations in training process. As shown in Fig. 2(b), given the two corresponding parts P and $P^{'}$, the two tied convolution layers with max-pooling, where weights are shared across the two parallel structures, are used to compute higher-order feature maps $f^{\mathbb{N}_{map}}_{II,[1,25]}$ and $f^{'\mathbb{N}_{map}}_{II,[1,25]}$ of each input part P and $P^{'}$ separately. Then $f^{\mathbb{N}_{map}}_{III,[1,50]}$, which represents deep difference of feature maps, is achieved through

Table 2
Parameters setting of $\mathbb{N}_{map}$. Each row illustrates the parameters (kernel size and stride) of a convolution layer followed by a max-pooling layer between two adjacent layers. Notice that the weights (kernel) of two parallel convolution layers from $P(P^{'}) \rightarrow I$ and $I \rightarrow II$ are shared.

$\mathbb{N}_{map}$	Convolution		Max-pooling	
	Kernel size	Stride	Kernel size	Stride
$P(P^{'}) \rightarrow I$	3 × 3 × 3 × 20 (shared)	1	2 × 2	2
$I \rightarrow II$	2 × 2 × 20 × 25 (shared)	1	2 × 2	2
$III \rightarrow IV$	2 × 2 × 50 × 50	1	2 × 2	2

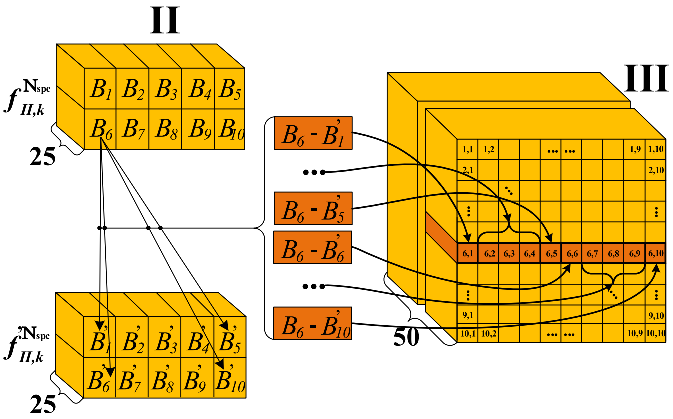

II
III
$f^{\mathbb{N}_{spc}}_{II,k}$
$f^{'\mathbb{N}_{spc}}_{II,k}$
25
25
50
B_1 B_2 B_3 B_4 B_5
B_6 B_7 B_8 B_9 B_{10}
$B^{'}_1$ $B^{'}_2$ $B^{'}_3$ $B^{'}_4$ $B^{'}_5$
$B^{'}_6$ $B^{'}_7$ $B^{'}_8$ $B^{'}_9$ $B^{'}_{10}$
$B_6 - B^{'}_1$
$B_6 - B^{'}_5$
$B_6 - B^{'}_6$
$B_6 - B^{'}_{10}$
1,1 1,2 1,9 1,10
2,1 2,10
6,1 6,2 6,3 6,4 6,5 6,6 6,7 6,8 6,9 6,10
9,1 9,10
10,1 10,2 10,9 10,10

Fig. 3. Spatial traversal manner used to calculate deep spatial variations information from *II*th to *III*th layer in $\mathbb{N}_{spc}$. Each block (B_i, $i\forall[1, 10]$) in $f^{\mathbb{N}_{spc}}_{II,[1,25]}$ is compared with every block (B_j, $j \in [1, 10]$) in $f^{'\mathbb{N}_{spc}}_{II,[1,25]}$, and vice versa.

element-wise subtraction between $f^{\mathbb{N}_{map}}_{II,[1,25]}$ and $f^{'\mathbb{N}_{map}}_{II,[1,25]}$. More precisely:

$$\begin{aligned} f^{\mathbb{N}_{map}}_{III,[1,25]}(x,y) &= f^{\mathbb{N}_{map}}_{II,[1,25]}(x,y) - f^{'\mathbb{N}_{map}}_{II,[1,25]}(x,y) \\ f^{\mathbb{N}_{map}}_{III,[26,50]}(x,y) &= f^{'\mathbb{N}_{map}}_{II,[1,25]}(x,y) - f^{\mathbb{N}_{map}}_{II,[1,25]}(x,y) \end{aligned} \tag{4}$$

where symmetry is considered in layer *III* in which we compute the difference from $f^{\mathbb{N}_{map}}_{II,[1,25]}$ to $f^{'\mathbb{N}_{map}}_{II,[1,25]}$ to get the first 25 feature maps, followed by a reverse operation to get the last 25 feature maps of $f^{\mathbb{N}_{map}}_{III,[1,50]}$. Then, $f^{\mathbb{N}_{map}}_{III,[1,50]}$ is passed through ReLU activation function, and a convolution layer with max-pooling is used on activated $f^{\mathbb{N}_{map}}_{III,[1,50]}$ in order to get the *IV*th layer of $\mathbb{N}_{map}$. Following the *IV*th layer, the deep difference feature F_{map} is achieved through a fully connected layer with ReLU. The details of the parameters setting are illustrated in Table 2.

3.3. Cross-space difference subnet

In most cases, spatial variations such as viewpoints, occlusions and pedestrian poses occur under different scenarios, especially when person images are cropped by the pedestrian detector. Deep difference features learned from these images using comparisons between corresponding positions could not effectively reduce these spatial variations. In order to cope with this problem, a cross-space difference subnet is presented to learn deep difference features from spatial variations.

Fig. 2(c) illustrates the structure of our $\mathbb{N}_{spc}$; the operations of the first two layers of $\mathbb{N}_{spc}$ are identical to $\mathbb{N}_{map}$. In order to learn difference features from spatial variations, feature maps $f^{\mathbb{N}_{spc}}_{II,[1,25]}$ and $f^{'\mathbb{N}_{spc}}_{II,[1,25]}$ are firstly divided into 10 separate blocks, which are denoted as $B_i (i \in [1, 10])$ and $B^{'}_j (j \in [1, 10])$, respectively (see Fig. 3). Amongst them, each block represents a group of spatial features with size 5 × 5 × 25.[1] Secondly, B_i and $B^{'}_j$ are compared using spatial traversal, which compared any two blocks between B_i and $B^{'}_j$ with element-wise subtraction operation (see Fig. 3). More precisely:

Table 3
Parameters setting of $\mathbb{N}_{spc}$. Each row illustrates the parameters (kernel size and stride) of a convolution layer followed by a max-pooling layer between two adjacent layers. Notice that the weights (kernel) of two parallel convolution layers from $P(P^{'}) \rightarrow I$ and $I \rightarrow II$ are shared. *w* and *h* represent the stride in dimension of width and height, respectively.

$\mathbb{N}_{spc}$	Convolution		Max-pooling	
	Kernel size	Stride	Kernel size	Stride
$P(P^{'}) \rightarrow I$	3 × 3 × 3 × 20 (*shared*)	1	2 × 2	2
$I \rightarrow II$	2 × 2 × 20 × 25 (*shared*)	1	2 × 2	2
$III \rightarrow IV$				
*Conv*1	5 × 25 × 50 × 50	$w = 5$ $h = 25$	2 × 2	2
*Conv*2	5 × 5 × 50 × 50	$w = 5$ $h = 5$	2 × 2	2
*Conv*3	25 × 5 × 50 × 50	$w = 25$ $h = 5$	2 × 2	2

$$\begin{aligned} f^{\mathbb{N}_{spc}}_{III,[1,25]}(\mathbb{R}(i,j)) &= f^{\mathbb{N}_{spc}}_{II,[1,25]}(B_i) - f^{'\mathbb{N}_{spc}}_{II,[1,25]}(B^{'}_j) \\ f^{\mathbb{N}_{spc}}_{III,[26,50]}(\mathbb{R}(j,i)) &= f^{'\mathbb{N}_{spc}}_{II,[1,25]}(B^{'}_j) - f^{\mathbb{N}_{spc}}_{II,[1,25]}(B_i) \end{aligned} \tag{5}$$

where $\mathbb{R}(i, j)$ represents a 5 × 5 × 25 block that belongs to *i*th row and *j*th column in layer *III*, symmetry is considered that we compute the difference from $f^{\mathbb{N}_{spc}}_{II,[1,25]}$ to $f^{'\mathbb{N}_{spc}}_{II,[1,25]}$ to get the first 25 feature maps; a reverse operation is made to get the last 25 feature maps of $f^{\mathbb{N}_{spc}}_{III,[1,50]}$. Then, $f^{\mathbb{N}_{spc}}_{III,[1,50]}$ is activated with ReLU, followed by three convolution layers with max-pooling (*Conv*1, *Conv*2 and *Conv*3) to extract higher-order difference information of spatial variations. Table 3 illustrates the details of parameters setting of $\mathbb{N}_{spc}$.

As shown in Fig. 2(c), the *Conv*1 and *Conv*3 are utilized to extract features of spatial variations by comparing one blocks in B_i and five blocks in $B^{'}_j$ which belong to the same row of $f^{'\mathbb{N}_{spc}}_{II,[1,25]}$. Therefore, *Conv*1 and *Conv*3 both use rectangle-shaped convolution kernels and the size of is five times that of *Conv*2, these three parallel convolution layers are passed through a fully connected layer with ReLU to extract deep difference features F_{spc}.

4. Pyramid partition architecture

The image partition granularity is crucial for our part-based person re-identification method. Parts with coarse granularity partition cover more information, but when comparing the numerous details that exist between two parts, it can be a time-consuming process. In contrast, parts with fine granularity partition contain less information, but details can be exploited to extract robust difference features between local regions. As such, a pyramid partition architecture is presented in order to balance the two partition modes, since it utilizes the different combinations of the three subnets on different partition granularities.

Our pyramid partition architecture divides parts using a Deep Decompositional Network (DDN) [13] which dissects the person's image into six semantic regions: hair, head, upper body, arms, legs and background. As shown in Fig. 4, the DDN is not the perfect

[1] The size of the input part P or $P^{'}$ is 56(*width*) × 32(*height*) (the details of the partition granularity of body parts are introduced in Section 4), after two tied convolution layers, the size of $f^{\mathbb{N}_{spc}}_{II,k}$ and $f^{'\mathbb{N}_{spc}}_{II,k}$ are 13(*width*) × 7(*height*) with our parameters setting (see Table 3). Thus, with block size 5 × 5 × 25, the stride of block segmentation is 2 for both the width and height dimension in layer *II*.

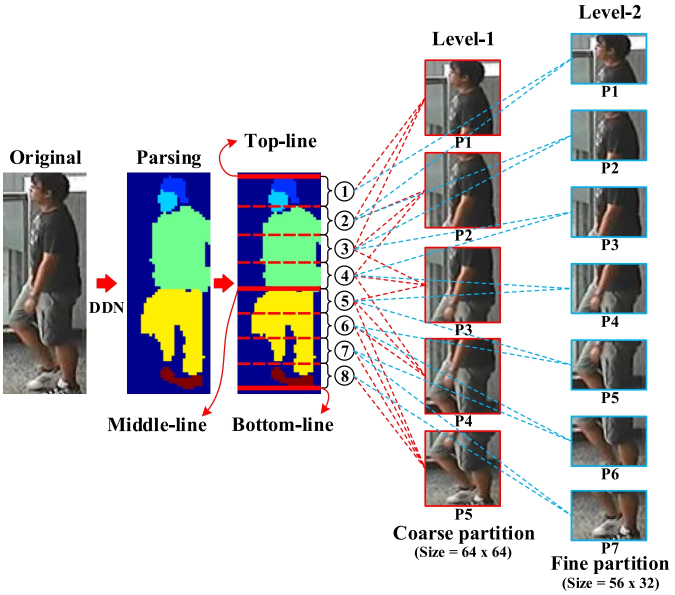


Fig. 4. Pyramid partition architecture. Each person image is divided into two levels: the Level-1 achieves through coarse granularity partition and each part with size of 64 × 64, the Level-2 achieves through fine granularity partition and each part with size of 56 × 32. The DDN algorithm is used to calculate the location of different partition lines.

method to parse different body regions, but it effectively achieves three boundaries of top, middle and bottom lines. From there six partition lines are obtained on upper body and legs using the three boundaries which equally divide the areas between the top-and-middle and the middle-and-bottom lines into four parts, respectively. We use two levels to construct our pyramid partition architecture. As Fig. 4 shows, the coarse granularity partition which contains five parts with size 64 × 64 (after resizing) is denoted as “Level-1” and the fine granularity partition which contains seven parts with size 56 × 32 (after resizing) is denoted as “Level-2”. Both are achieved through combinations of adjacent parts based on the nine partition lines (three boundaries lines + six partition lines). Finally, the two partition granularities are utilized on each image of our method. Two kinds of deep neural networks are utilized on the two levels in order to determine whether or not the two corresponding parts belong to the same person. The two deep neural networks consisted of the three subnets $\mathbb{N}_{dat}$, $\mathbb{N}_{map}$ and $\mathbb{N}_{spc}$, as described in Section 3. For Level-1, given two corresponding parts, $\mathbb{N}_{dat}$ and $\mathbb{N}_{map}$ are simultaneously optimized to evaluate whether the two parts belong to the same person. As shown in Fig. 5, $\mathbb{N}_{dat}$ and $\mathbb{N}_{map}$ are used on two input parts P and $P^{'}$ in order to learn deep difference features F_{dat} and F_{map} with size of 160, respectively. Then F_{dat} and F_{map} are passed through a fully connected layer which contains 160 hidden units to learn deeper difference features f^{LV1}. Finally a fully connected layer with 2 softmax units is used to determine if the two input parts belong to the same person, which is formulated as:

$$h_\theta(f) = \begin{bmatrix} p(y=0|f;\theta_0) \\ p(y=1|f;\theta_1) \end{bmatrix} = \frac{1}{\sum_i^{\{0,1\}} e^{\theta_i^T * f}} \begin{bmatrix} e^{\theta_0^T * f} \\ e^{\theta_1^T * f} \end{bmatrix} \tag{6}$$

where, θ represents learnable parameters of the last fully connected layer with 2 softmax units, f is deeper difference features, $p(y=1|f;\theta_1)$ and $p(y=0|f;\theta_0)$ respectively represents probability of the two inputs belonging to the same person or not. Unlike Level-1, all three subnets $\mathbb{N}_{dat}$, $\mathbb{N}_{map}$ and $\mathbb{N}_{spc}$ are utilized to evaluate whether or not the two input parts with fine granularity partition belong to the same person. As shown in Fig. 6, F_{dat}, F_{map} and F_{spc} are simultaneously learned through the three subnets. Since the input parts in Level-2 contain less information than Level-1, another 100 hidden units are used in order to learn each kind of feature in Level-2. It is worth noting that the two tied convolution layers with max-pooling of $\mathbb{N}_{map}$ and $\mathbb{N}_{spc}$ are shared with each other to extract deep features F_{map} and F_{spc}. Then, F_{map} and F_{spc} are passed through a fully connected layer with 100 hidden units to get features f_1^{LV2}, and another fully connected layer (also with 100 hidden units) is used after f_1^{LV2} and F_{dat} to get deeper difference feature f_2^{LV2}. Lastly, a fully connected layer with 2 softmax units is used to determine if the two input parts belong to the same person (see Eq. (6)).

Since the input parts of Level-1 contain more information than those of Level-2, the $\mathbb{N}_{spc}$ which is used to learn deep difference features through spatial traversal, is not suitable for use in Level-1 because it is time-consuming. In addition to that fact, the comparison between blocks with long distance in vertical direction in Level-1, is not useful to the final decision, e.g., one block on upper body in one image is unlikely to move on legs in another image. Therefore, we only use $\mathbb{N}_{spc}$ on Level-2 of our pyramid partition architecture.

5. Training and integration validation

DeepDiff for person re-identification is proposed as a binary classification problem. Using the training data, each image is first divided into 12 parts (7 + 5) based on our pyramid partition architecture. Corresponding part pairs are labeled as positive (same) and negative (different). The final decision which determines whether the two images that belong to the same person is achieved by accumulating the similarity score of those 12 parts. The optimization objective is to average the loss over all the input corresponding part pairs. Due to the fact that the datasets used for training are large, stochastic approximation of this objective is utilized with mini-batches randomly divided within the training data. The networks used in coarse and fine granularity partitions performed a forward propagation on one mini-batch to compute output and loss. Backward propagation is then used to compute gradients on a current mini-batch for updating parameters. We use stochastic gradient descent [36] to determine the parameters' update with learning rate $lr^1 = 0.01$; we gradually decrease it as the training iteration increases through an inverse policy:

$$lr^n = lr^1(1 + gamma \times n)^{power} \tag{7}$$

where $gamma = 10^{-4}$, $power = 0.75$, and n represents iterations. We use a $momentum = 0.9$ and $weight\_decay = 5 \times 10^{-4}$. As more batches pass over the networks, the model improved until it converges.

5.1. Batch normalization

It is well known fact that deep neural network is hard to optimize due to internal-covariate-shift [37]. Therefore, batch normalization [37] is performed to reduce this problem by normalizing the input distributions of every layer to the standard Gaussian distribution. A batch normalization layer is added to the output of every convolutional layer; this is critical to optimize our network with a better local optimum and a higher convergence rate.

5.2. Dropout learning

The dropout learning [38] is proposed to deal with the overfitting problem and this is applied in our training strategy. Since each part is trained in two or three parallel networks, a large quantity of parameters have to be learned in our model. As such, some datasets for person re-identification are too small for training (e.g., the VIPeR) and easily lead to over-fitting. Therefore, a 40% dropout

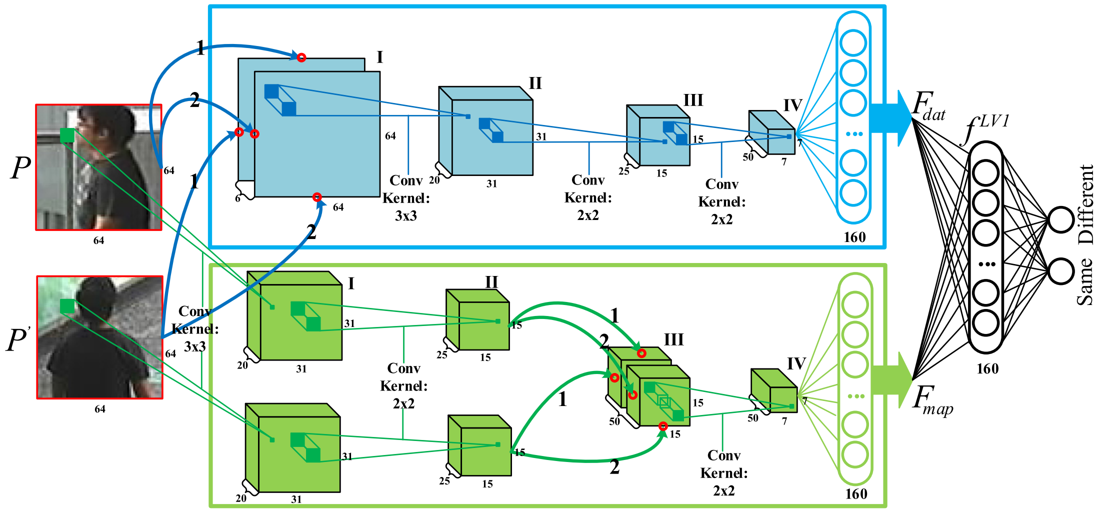


Fig. 5. Deep neural network used to evaluate whether or not the two parts with coarse granularity partition belong to the same person. The upper subnet is $\mathbb{N}_{dat}$ and the lower one is $\mathbb{N}_{map}$. Both of them are simultaneously used to learn deep feature from two input parts. With the input size (64 × 64), the size of feature maps on each layer is illustrated with corresponding parameters setting shown in Tables 1 and 2.

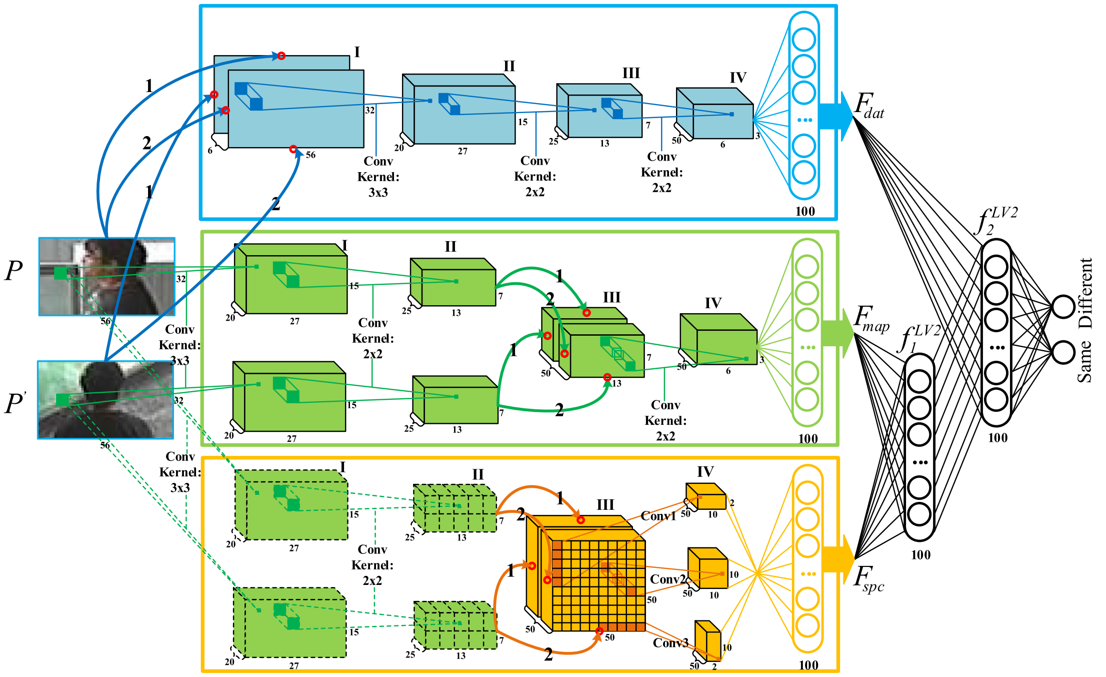


Fig. 6. Deep neural network used to evaluate whether or not the two parts with fine granularity partition belong to the same person. The upper subnet is $\mathbb{N}_{dat}$, the middle one is $\mathbb{N}_{map}$ and the lower one is $\mathbb{N}_{spc}$. All of them are simultaneously used to learn deep feature from the two input parts. With input size (56 × 32), the size of feature maps on each layer is illustrated with corresponding parameters setting which is shown in Tables 1–3. Due to the fact that the two tied convolution layers with max-pooling are shared by $\mathbb{N}_{map}$ and $\mathbb{N}_{spc}$, the *I*th and *II*th layers of $\mathbb{N}_{spc}$, which represent a copy of first two layers of $\mathbb{N}_{map}$, are illustrated with dotted line and green color.

rate is used in our training process, which means that for each training sample that is used as input at each training iteration, about 40% neurons are randomly selected from fully connected layers and set as zeros to make our model more stable.

5.3. Data augmentation

In order to reduce the over-fitting, the CUHK03 dataset [5] is proposed as a large re-identification dataset for deep learning with manually labeled and detected pedestrian images. We combine this dataset with two others, including CUHK01 and VIPeR, for pre-training the model that is used on these three dataset separately. We randomly reflect images in a horizontal direction for each iteration of the training process for the purpose of obtaining enough training samples. In addition, we use linear transformations to enlarge the size of the datasets by cropping images to augment them using a window with 90% of the image's width and height, position uniformly at random without overlapping the image's edge. However, the training set is imbalanced because data augmentation increases the number of pairs with more negative pairs rather than positive pairs. Therefore, after the data augmentation for network training, we randomly down-sample the negative pairs in order to get twice as many positive pairs.

5.4. Fine-tuning

Generally, small datasets contain few positive samples for training. As such, we pre-train our networks on the CUHK03, CUHK01 and VIPeR datasets, and the pre-trained model is then adapted to each dataset for fine-tuning. During this fine-tuning, we start a stochastic gradient descent with a learning rate $lr^1 = 0.001,$ which is 1/10th of the initial pre-training rate.

5.5. Integration validation

Our integration validation method demonstrates the power of combination by validating all the parts integrally rather than separately. During the training process, a fixed interval is used to keep intermediate models before the maximum iteration (the interval step equals to 1000 for regular training and 500 for fine-tuning). The number of trained models that belongs to one part of Level-1 or Level-2 is denoted as n_{LV1} and n_{LV2}, respectively. As such, there are combinations of $n_{LV1}^7 \times n_{LV2}^5$ which are validated by accumulating the similarity score of 12 parts. If we select the last five models for validation e.g., $n_{LV1} = n_{LV2} = 5,$ the number of combinations is $5^7 \times 5^5 \approx 10^8$ which is too large to compute. Therefore, we conduct our validation into a two-step process: validating different levels separately, and then combining them all.

In the first step, the validation is processed on Level-1 and Level-2, separately. After the network convergence, 9 and 5 latest models are respectively selected on each part of Level-1 and Level-2 before the maximum iteration. Thus, there are $9^5 = 59,049$ and $5^7 = 78,125$ combinations to be validated on Level-1 and Level-2, separately. After that, the top 200 combinations with the best performance are selected on each level. In the second step, the final decision model is achieved by 200 × 200 combinations between Level-1 and Level-2:

$$iv = \underset{(i_1,i_2)}{\operatorname{argmax}} \left(\Phi \left(\left(\left(\sum_{j=1}^{5} \mathbb{M}_{i_1 j}^{LV1} \right)_{i_1=1}^{200} + \left(\sum_{j=1}^{7} \mathbb{M}_{i_2 j}^{LV2} \right)_{i_2=1}^{200} \right) \right) \right) \tag{8}$$

where $\mathbb{M}$ is similarity matrix and each element in $\mathbb{M}$ represents the similarity score between one person's part in the probe set and another in the gallery set; $LV1$ and $LV2$ respectively represents Level-1 and Level-2 in our pyramid partition architecture; j represents the part index; i_1 and i_2 are indexes of the top 200 combinations in Level-1 and Level-2, respectively. $\Phi(*)$ represents the probability of correct matching between two sets, and iv denotes the index of the best combination in our integration validation. The total combination number of our integration validation method is $\approx 10^5$ which is effective and less time-consuming in the overall design.

6. Experiments

Experimental studies are conducted on three public datasets including CUHK03 [5], VIPeR [8] and CUHK01 [39], in order to evaluate the performance of our DeepDif method. Those results are presented using the average Cumulative Matching Characteristics (CMC) curve, with over 10 trials on each dataset by a single-shot comparison manner. In the CMC, the rank-r matching rate indicates the percentage of images in the probe set with correct matches that are found in the top r ranks against the number of person in the gallery set. Therefore, the rank-1 matching rate is the correct matching rate.

Our DeepDiff method is implemented using the deep learning framework Caffe [40], by adapting various layers from the framework and writing our own layers; all the networks are trained with six NVIDIA K20 GPUs, and the three datasets have a set $batchsize = 128$ on each GPU. After 30,000 iterations in the pre-training stage, the network is convergent on each part; in total, 12 parts take about 36–42 h. During the fine-tuning stage, 10,000 iterations are used on each part; in total, it takes about 12–16 h to convergence on each dataset. We first report the results on the largest person re-identification dataset CUHK03, followed by the results on the VIPeR dataset, and then two results on the CUHK01 with distinct setting: 100 and 486 identities in test set, respectively. Dropout learning, data argumentation and fine-tuning are used in our experiments in order to cope with over-fitting (see Sections 5.2–5.4). Finally, the integration validation method is used on those datasets (see Section 5.3) to achieve better performance.

6.1. Experimental datasets

CUHK03: the CUHK03 dataset contained 1467 identities collected from five pairs of disjointed camera views; on average, each identity has 4.8 images. This dataset provides both manually-labeled and automatically-detected person images obtained from the pedestrian detector [41]. For a fair comparison, we use the same training and testing protocols in [5], which split the CUHK03 in two sets, 1367 identities for training and 100 identities for testing. We randomly select 100 identities for validation in the training set and leave 1267 identities for training. After the data augmentation (see Section 5.3), we adapt the pre-trained model on the three datasets for each part of Level-1 and Level-2, by fine-tuning 1267 training identities.

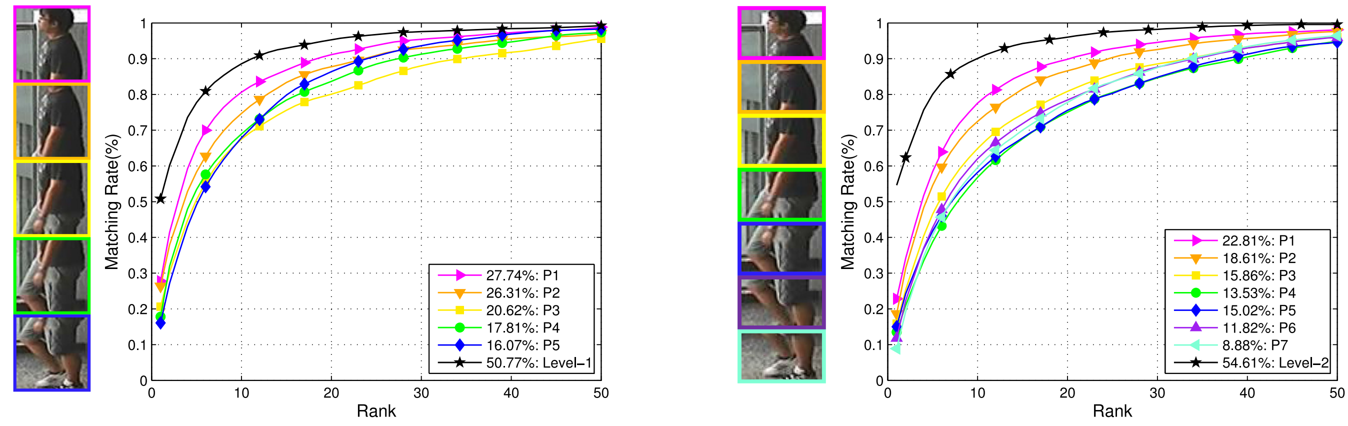
VIPeR: the VIPeR dataset contains 632 identities, each with two images captured from disjointed camera views. This dataset is one of the most challenging datasets to work with because of the various viewpoints, lightings, background clutters and pedestrian poses. In order to maintain a fair comparison, we used the same training and testing protocols in [3], which split VIPeR into two sets, with 316 identities for training and 316 identities for testing. Since the VIPeR is a small dataset, we randomly select 50 identities for validation and leave 266 identities for training. After the data augmentation, we adapt the pre-trained model to this dataset by fine-tuning the 266 training identities.

CUHK01: the CUHK01 dataset has 971 identities, each with two images captured from disjointed camera views. Two distinct settings are reported on this dataset: (1) 100 identities in the test set; and (2) 486 identities in the test set. In the 100 identities setting, 871 identities are used for training. We randomly select 100 identities from the training set for validation and leave 771 identities for training. In the 486 identities setting, 485 identities are used for training. We randomly select 50 identities from the training set for validation, leaving 435 identities for training, and then we adapt the pre-trained model onto the CUHK01 dataset on the two settings by fine-tuning the 771 and 435 training identities, respectively.

6.2. Experimental analysis

6.2.1. Analyze the discriminative power of body parts

Fig. 7 shows the performance of our networks trained on different levels and parts on the CUHK03 dataset (labeled). As we moved down the body, the performance decreases except from P4 to P5 in Level-2. The P1 performs the best on both levels; the upper body captures more discriminative power than the lower body in our experiments. Fig. 8 illustrates deep features of our part-based DeepDiff method on the CUHK03 dataset with positive and negative pairs. The L_1 displays a person image selected in the probe set and the L_2 displays two person images selected in the gallery set with the same and different identities to image in L_1. In Level-1, we divide each person's image into five parts; L_5 and L_6 show



(a) Level-1 (b) CUHK03 labeled Level-1 (c) Level-2 (d) CUHK03 labeled Level-2

Fig. 7. (a) and (c) illustrate the part partition of Level-1 and Level-2. The color of bounding box of each part in (a) and (c) is corresponding to the CMC curves in (b) and (d).

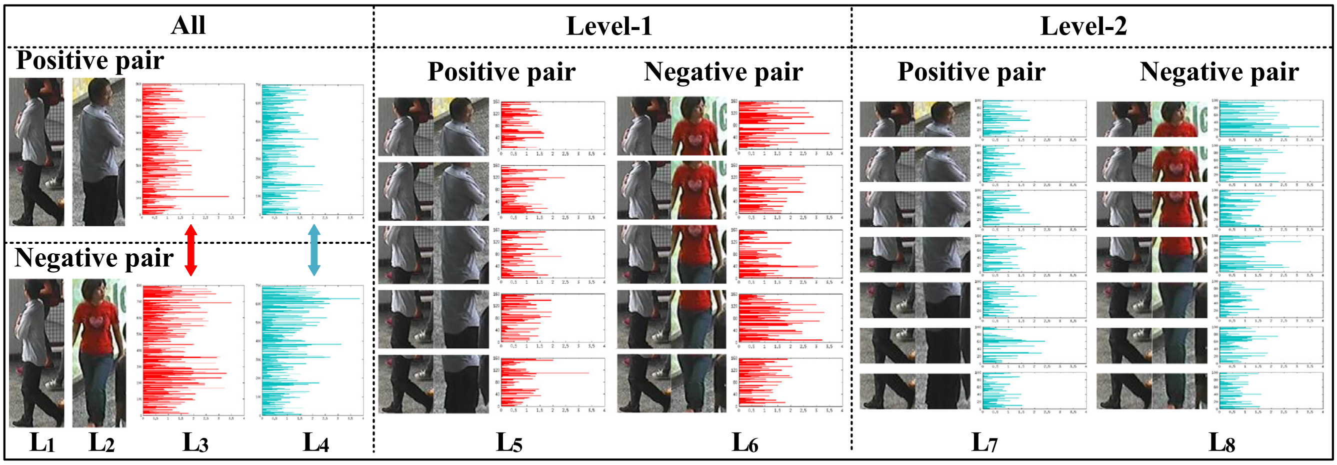


Fig. 8. Deep features of our part-based method on CUHK03 dataset with a positive and a negative pairs. The L_1 contains a person image selected in the probe set and the L_2 displays two person images selected in the gallery set with the same (the upper image in L_2) and different (the lower image in L_2) identities to image in L_1. The L_5 and L_6 show deep features of each part in the Level-1. The L_7 and L_8 show deep features of each part in Level-2. The L_3 and L_4 display features on the whole body, which contains all part-based features from Level-1 and Level-2, respectively.

deep features of f^{LV1} with 160 hidden layer units (see Fig. 5) within these five parts. In Level-2, we divide each person image into seven parts; L_7 and L_8 show deep features of f_2^{LV2} with 100 hidden layer units (see Fig. 6). The deep features of a positive pair are different than that of a negative pair, making them discriminative and capable of being used as input to an off-the-shelf classifier. The L_3 and L_4 display 800 ($5parts \times 160$) and 700 ($7parts \times 100$) features on the whole body, which contain all part-based features from Level-1 and Level-2, respectively.

6.2.2. Analyze the performance of the integration validation

Fig. 9 shows the performance of our network trained on two levels which illustrates the effectiveness of our part-based integration validation method (iv) (see Section 5.5) on the CUHK03 dataset (labeled). A 50.77% and 54.61% rank-1 matching rate can be achieved on Level-1 and Level-2, respectively; through the validation on different levels. It is due to the integration validation method that Level1-iv and Level2-iv achieve a 48.26 and 52.40% rank-1 matching rate. Although the performance is slightly less than the validation of the two levels separately, our integration validation method achieves 62.43% rank-1 matching rate, and improves 4.43% than simply accumulating the matching results of Level-1 and Level-2.

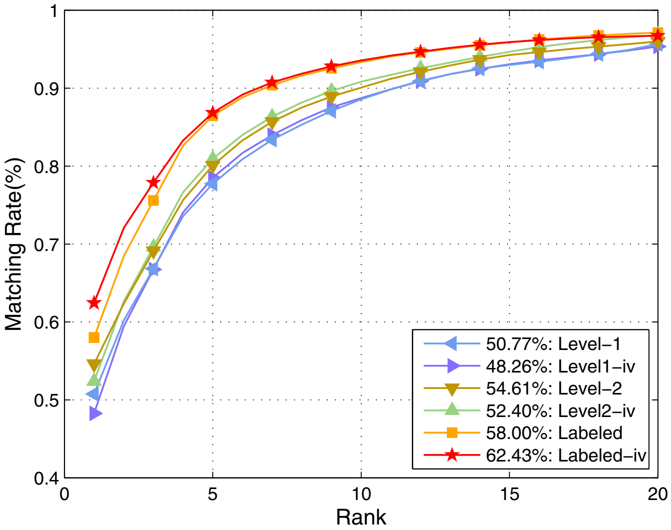


Fig. 9. CMC curves of the CUHK03 dataset (labeled). Rank-1 matching rate is marked before the name of each approach. The Level-1 and Level-2 represent the best performance after validating our method on the two levels separately. The Level1-iv, Level2-iv and Labeled-iv represent the performance by using our integration validation.

Table 4
CMC results on the CUHK03 (labeled) dataset. The performance of rank (%) 1, 5, 10 and 20 are listed.

Method	Rank=1	Rank=5	Rank=10	Rank=20
ITML [44]	5.5	18.1	28.9	44.4
LMNN [45]	7.3	19.6	30.7	47.9
LDM [46]	13.5	37.1	52.7	70.1
RANK [47]	10.4	30.2	44.4	62.1
KISSME [48]	14.1	37.46	52.2	69.4
FPNN [5]	20.7	50.9	67.01	83.0
IDLA (no hnm) [7]	50.2	84.3	92.7	97.6
IDLA [7]	54.7	86.5	93.9	**98.1**
LOMO+XQDA [19]	52.2	82.2	92.1	96.3
LOMO+MLAPG [42]	58.0	87.1	**94.7**	98.0
SS-SVM [43]	57.0	–	–	–
DeepDiff	**62.4**	**87.9**	93.6	96.7

Table 5
CMC results on the CUHK03 (detected) dataset. The performance of rank (%) 1, 5, 10 and 20 are listed.

Method	Rank=1	Rank=5	Rank=10	Rank=20
ITML [44]	5.14	17.4	27.6	43.1
LMNN [45]	6.3	17.9	28.6	45.0
LDM [46]	10.9	31.9	47.0	65.0
RANK [47]	8.5	26.2	40.0	57.9
KISSME [48]	11.7	33.5	48.1	64.9
FPNN [5]	19.9	49.4	64.8	81.1
IDLA [7]	45.0	76.0	83.5	93.2
LOMO+XQDA [19]	46.3	78.9	88.6	94.3
LOMO+MLAPG [42]	51.2	83.6	92.1	**96.9**
SS-SVM [43]	51.2	–	–	–
SI-CI [49]	52.2	**84.3**	**92.3**	–
DeepDiff	**54.8**	79.3	89.5	95.9

6.3. Compare with state-of-the-art methods

6.3.1. Experimental results on the CUHK03 dataset

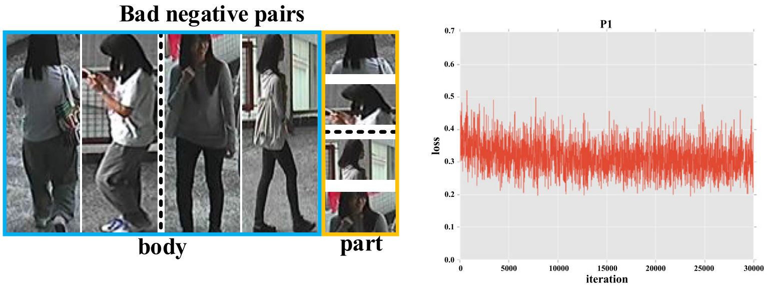
Table 4 shows the comparison of our DeepDiff with the following state-of-the-art approaches on CUHK03 dataset (labeled). Amongst them, FPNN [5] and IDLA [7] are deep learning methods which show better performance than previous state-of-the-art methods. FPNN [5] is the first using deep neural network for person re-identification; following that, IDLA [7] is proposed and its performance outperforms the state-of-the-art method on the CUHK03 dataset by more than double. Then, three traditional methods are proposed, including: LOMO+XQDA [19], LOMO+MLAPG [42] and SS-SVM [43]; among them, LOMO+MLAPG [42] and SS-SVM [43] achieve better performance than the previous deep learning method, IDLA [7]. These three traditional methods use the same feature LOMO which is proposed in [19], but with different distance metric models. As exhibited in Table 4, by comparing these state-of-the-art methods, our Deepdiff achieves the best rank-1 matching rate on the CUHK03 dataset (labeled).

In Table 4, the hard negative mining (hnm) raises the rank-1 performance of IDLA [7] from 50.19 to 54.74%. In order to perform hnm, the network introduced in IDLA [7] needs to classify all of the negative pairs; first, the negative pairs have to be collected from the networks which performs the worst (denoted as "bad negative pairs"), and then those "bad negative pairs" and the positive pairs are used to retrain the last layer of the network. We try this hnm trick on each part of DeepDiff, but the performance is unsatisfactory. Our analysis shows that "bad negative pairs" on body parts are hard to recognize across different camera views than the whole body. As shown in Fig. 10(a), it is easy to recognize the two identities from their lower bodies (since their upper bodies share the similar appearance), but it is hard to recognize them from one distinct body part. As such, networks retained on the part-based "bad negative pairs" and positive pairs are not easy to converge. We also try to randomly collect "bad negative pairs" from negative pairs on which the network classifies them as positive pairs, but the network is also not convergent (see Fig. 10(b)), and the performance is not better than our raw results.

Table 5 illustrates the comparison between state-of-the-art approaches and our DeepDiff; the latter achieved the best rank-1 matching rate on the CUHK03 dataset (detected) by 2.6%. Our DeepDiff shows its robustness of spatial misalignment that is caused by directly capturing images from the detector, thanks to the deep difference feature from our cross-space difference subnet (see Section 3.3), and the pyramid partition method that is used to align the different parts using body partition lines. Amongst the deep learning methods FPNN [5], IDLA [7] and SI-CI [49] depict in Table 5, SI-CI [49] achieves the best result compared to the traditional methods of the CUHK03 dataset (detected). Our DeepDiff outperforms SI-CI [49] at the rank-1 matching rate and beat other traditional methods.

6.3.2. Experimental results on the VIPeR dataset

In Table 6, we compare our DeepDiff method with several state-of-the-art methods on the VIPeR dataset. The VIPeR dataset is a small dataset that contains 632 persons, each with its own two images from different camera views, thus there are not enough positive samples to train our network. Over-fitting still exists even though data argumentation and dropout tricks are used on this dataset, but despite that fact, we continue to conduct experiments in order to show our method's performance. Even on the small dataset, our method outperforms other traditional approaches which are not affected by over-fitting, such as the KISSME [48], mFilter [22], LADF [23], SalMatch [21], kLFDA [26], SCNCD



(a) Whole body vs. one part

(b) Loss value of part-based hard negative mining.

Fig. 10. (a) Two identities with similar appearance in the blue boxes. The rightmost box shows four parts belonging to the four images in the left box, which are hard to recognize their identities. (b) The loss value plots with our hnm trick for 30,000 iterations, which is not convergent in our experiments.

Table 6
CMC results on the VIPeR dataset. The performance of rank (%) 1, 5, 10 and 20 are listed.

Method	Rank=1	Rank=5	Rank=10	Rank=20
KISSME [48]	27.0	55.0	70.0	83.0
LADF [23]	30.0	64.0	80.0	92.0
SalMatch [21]	30.1	52.3	65.8	–
mFilter [22]	29.1	52.5	65.9	79.9
mFilter [22] + LADF [23]	43.4	–	–	–
kLFDA [26]	32.3	65.8	79.7	90.9
SCNCD [50]	37.8	68.5	81.2	90.4
IDLA [7]	34.8	63.5	75.0	80.0
PRCSL [27]	34.8	68.7	82.3	94.9
DFRDC [52]	40.5	60.8	70.4	84.4
LOMO+XQDA [19]	40.0	68.1	80.5	91.1
LOMO+MLAPG [42]	40.7	69.9	82.3	92.4
LOMO [19]+LADF [23]	50.3	–	–	–
SR4PR [51]	41.6	71.9	86.2	95.1
SI-CI [49]	35.8	67.4	83.5	–
SS-SVM [43]	42.7	–	84.3	91.9
MCP-CNN [35]	47.8	74.7	84.8	91.1
SCSP [29]	**53.5**	**82.6**	**91.5**	**96.7**
DeepDiff	43.2	68.0	77.6	86.1

Table 7
CMC results on the CUHK01 (100) dataset (single-shot). The performance of rank (%) 1, 5, 10 and 20 are listed.

Method	Rank=1	Rank=5	Rank=10	Rank=20
ITML [44]	17.1	41.5	55.2	70.7
LMNN [45]	21.2	48.5	63.0	78.1
LDM [46]	26.5	57.7	72.8	85.5
RANK [47]	20.6	47.6	62.1	76.6
KISSME [48]	29.4	60.2	74.4	86.6
FPNN [5]	27.9	59.6	73.5	87.3
IDLA [7]	65.0	89.0	94.0	97.5
SI-CI [49]	71.80	–	–	–
DeepDiff	**72.2**	**89.0**	**95.5**	**99.3**

Table 8
CMC results on the CUHK01 (486) dataset (single-shot). The performance of rank (%) 1, 5, 10 and 20 are listed.

Method	Rank=1	Rank=5	Rank=10	Rank=20
ITML [44]	16.0	35.2	45.6	59.8
LMNN [45]	13.4	31.3	42.2	54.1
LDM [46]	9.9	22.6	30.3	41.0
RANK [47]	19.7	32.7	40.3	50.6
GenericMertic [39]	20.0	43.5	56.0	69.3
SalMatch [21]	28.5	45.9	55.7	68.0
kLFDA [26]	24.0	38.9	46.7	55.4
mFilter [22]	34.3	55.1	65.0	74.9
SR4PR [51]	31.5	52.5	65.8	77.6
IDLA [7]	47.5	71.6	80.2	87.4
MCP-CNN [35]	**53.7**	**84.3**	**91.0**	**96.3**
DeepDiff	47.9	69.6	79.2	86.9

[50], PRCSL [27], LOMO+XQDA [19], LOMO+MLAPG [42], SR4PR [51] and SS-SVM [43]. Table 6 also shows the performance of the four deep learning approaches including: IDLA [7], DFRDC [52], SI-CI [49] and MCP-CNN [35]. Compared to IDLA [7], DFRDC [52] and SI-CI [49], our DeepDiff outperforms the rank-1 matching rate on VIPeR dataset by 8.4%, 2.7% and 7.4%, respectively; the MCP-CNN [35] illustrates its rank-1 matching rate at 47.8%, which is better than ours, but it uses different types of configurations in its learning and testing strategies: small datasets such as VIPeR uses a small number of layers, whereas large datasets use more layers to construct their network. In the end, although our performance is inferior compared to MCP-CNN [35], we do not need to manually configure the structure of our network under different datasets.

Table 6 also reflects that several other traditional approaches such as SCSP [29] achieves better performance on this dataset (53.5%), as do mFilter [22]+LADF [23] (43.4%) and LOMO [19]+LADF [23] (50.3%). We believe that the reason behind this is due to the lack of positive pairs and the total number of distinct training identities compared to larger datasets. In order to improve the performance of our approaches, more datasets or data augmentation methods could be employed in the training process.

6.3.3. Experimental results on the CUHK01 dataset

Several approaches are compared with our DeepDiff method for the CUHK01 dataset (100 identifies). Table 7 shows that deep learning methods such as FPNN [5], IDLA [7] and SI-CI [49] have better performance than other traditional methods ITML [44], LMNN [45], LDM [46], RANK [47] and KISSME [48]. Since the 100 identifies setting of CUHK01 contains enough training samples which are not easily over-fitting, the two deep learning methods IDLA [7] and SI-CI [49] outperforms the other traditional methods by a large degree (e.g., SI-CI [49] (71.80%) vs. KISSME [48] (29.4%)). Nonetheless, our DeepDiff still outperforms the best rank-1 matching rate by 0.4% compared to the state-of-the-art method SI-CI [49]. Even though SI-CI [49] shows its competitive result on this dataset, it still could not match the performance of our method on small datasets (e.g., VIPeR: SI-CI [49] 35.8% vs. Ours 43.2%, shown in Table 6).

We also compare our DeepDiff with several state-of-the-art methods for the CUHK01 dataset (486 identifies). Table 8 shows that our DeepDiff achieves the second best rank-1 matching rate compared to those methods. Amongst them, IDLA [7] and MCP-CNN [35] are deep learning methods which illustrate better performance than other traditional methods. For the MCP-CNN [35] method which achieves the best rank-1 matching on this dataset, they manually configure the structure of their network under different datasets; for small datasets like VIPeR, they use less layers to cope with the over-fitting problem, whereas for large datasets, they use more layers to achieve better performance. Even though DeepDiff's performance is inferior to MCP-CNN [35], our method is more compatible than MCP-CNN [35], because we use the same network structure under various datasets.

6.3.4. Brief analyze of the experimental results

The effectiveness of our part-based deep learning method (DeepDiff) is demonstrated on three datasets, and the performance have obvious advantages compared to many state-of-the-art methods. However, the performance of our DeepDiff is inferior to other methods due to the increased number of identities to be matched (e.g., in Table 6 (316 identities to be matched); following rank-5, our method's performance become inferior to SR4PR [51], LOMO+MLAPG [42], LOMO+XQDA [19], PRCSL [27], and SCNCD [50], even though we outperforms the rank-1 matching rate). Because of our DeepDiff method accumulates the matching results of 12 parts on Level-1 and Level-2 to reach the final decision, although an integration method is proposed to enhance the final performance for our part-based approach, more effective part-based ensemble methods are expected to be exploited in our future works.

7. Conclusion and the future work

This paper presents the DeepDiff method for person re-identification. Since our DeepDiff focuses on human body parts, given the two corresponding parts, three deep subnets are created, which include learning deep features on original data, feature maps and spatial variations so as to determine whether the two parts belong to the same person. Using a part-based approach, we introduce a pyramid partition architecture with two levels partition granularities on each image. Out of the three subnets, there are two kinds of combinations optimized on different levels within that pyramid partition architecture in order to compute the similarity between the corresponding parts. Due to our part-based integration validation method, DeepDiff's performance is improved by confirming the performance of our models on different integral parts. Finally, the validity of our DeepDiff is demonstrated on three public datasets; its potential is evident from the competitive results in regards to quantitative evaluation.

Our approach is based on human body parts and the final decision is achieved by accumulating the similarity score of all parts. A part-based integration validation method is proposed to enhance the final decision, which is a preliminary attempt to assemble similar parts. For future research, we plan to exploit other part assembling methods in to enhance the performance of our DeepDiff theory.

Acknowledgments

This study is partially supported by the National Natural Science Foundation of China (No. 61472019), the National Science & Technology Pillar Program (No.2015BAF14B01). Supported by the Programme of Introducing Talents of Discipline to Universities and the Open Fund of the State Key Laboratory of Software Development Environment under grant #SKLSDE-2015KF-01. Thank you for the support from HAWKEYE Group.

References

[1] X. Wang, R. Zhao, Person re-identification:system design and evaluation overview, in: Person Re-Identification, Springer, 2014, pp. 351–370.

[2] D. Gray, H. Tao, Viewpoint invariant pedestrian recognition with an ensemble of localized features, in: Proceedings of the Tenth European Conference on Computer Vision Computer Vision - ECCV, Marseille, France, Part I, 2008, pp. 262–275.

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

[4] W. Zheng, S. Gong, T. Xiang, Person re-identification by probabilistic relative distance comparison, in: Proceedings of the Twenty-Fourth IEEE Conference on Computer Vision and Pattern Recognition,Colorado Springs, CO, USA, 20–25 June, 2011, pp. 649–656.

[5] W. Li, R. Zhao, T. Xiao, X. Wang, DeepReID: deep filter pairing neural network for person re-identification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,Columbus, OH, USA, June 23–28, 2014, pp. 152–159.

[6] C. Liu, S. Gong, C.C. Loy, On-the-fly feature importance mining for person re-identification, Pattern Recognit. 47 (4) (2014) 1602–1615.

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

[8] D. Gray, S. Brennan, H. Tao, Evaluating appearance models for recognition, reacquisition, and tracking, in: Proceedings of the IEEE International Workshop on Performance Evaluation for Tracking and Surveillance (PETS), Citeseer, 2007.

[9] C. Liu, S. Gong, C.C. Loy, X. Lin, Person re-identification: what features are important, in: Proceedings of the Computer Vision Workshops and Demonstrations, 2012, pp. 391–401.

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

[11] L. Ma, H. Liu, L. Hu, C. Wang, Q. Sun, Orientation driven bag of appearances for person re-identification, CoRR (2016). arXiv: 1605.02464

[12] L. Lin, T. Wu, J. Porway, Z. Xu, A stochastic graph grammar for compositional object representation and recognition, Pattern Recognit. 42 (7) (2009) 1297–1307.

[13] P. Luo, X. Wang, X. Tang, Pedestrian parsing via deep decompositional network, in: Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, December 1–8, 2013, pp. 2648–2655.

[14] Y. Sun, X. Wang, X. Tang, Deep learning face representation from predicting 10, 000 classes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, June 23–28, 2014, pp. 1891–1898.

[15] Y. Sun, Y. Chen, X. Wang, X. Tang, Deep learning face representation by joint identification-verification, in: Proceedings of the Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems, Montreal, Quebec, Canada, December 8–13, 2014, pp. 1988–1996.

[16] Y. Sun, X. Wang, X. Tang, Deeply learned face representations are sparse, selective, and robust, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, June 7–12, 2015, pp. 2892–2900.

[17] 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.

[18] 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, Surrey, UK, September 3–7, 2012, pp. 1–11.

[19] 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, Boston, MA, USA, June 7–12, 2015, pp. 2197–2206.

[20] 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, Portland, OR, USA, June 23–28, 2013, pp. 3586–3593.

[21] R. Zhao, W. Ouyang, X. Wang, Person re-identification by salience matching, in: Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, December 1–8, 2013, pp. 2528–2535.

[22] R. Zhao, W. Ouyang, X. Wang, Learning mid-level filters for person re-identification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, June 23–28, 2014, pp. 144–151.

[23] Z. Li, S. Chang, F. Liang, T.S. Huang, L. Cao, J.R. Smith, Learning locally-adaptive decision functions for person verification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23–28, 2013, pp. 3610–3617.

[24] S. Paisitkriangkrai, L. Wu, C. Shen, A.v. d. Hengel, Structured learning of metric ensembles with application to person re-identification, (2015). arXiv:1511.08531.

[25] D. Chen, Z. Yuan, G. Hua, N. Zheng, J. Wang, Similarity learning on an explicit polynomial kernel feature map for person re-identification, in: Proceedings of the International Conference on Computer Vision and Pattern Recognition, 2015, pp. 1565–1573.

[26] F. Xiong, M. Gou, O.I. Camps, M. Sznaier, Person re-identification using kernel-based metric learning methods, in: Proceedings of the 13th European Conference on Computer Vision - ECCV, Zurich, Switzerland, September 6–12, Part VII, 2014, pp. 1–16.

[27] Y. Shen, W. Lin, J. Yan, M. Xu, J. Wu, J. Wang, Person re-identification with correspondence structure learning, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 3200–3208.

[28] Y. Huang, H. Sheng, Z. Xiong, Person re-identification based on hierarchical bipartite graph matching, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), 2016, pp. 4255–4259.

[29] D. Chen, Z. Yuan, B. Chen, N. Zheng, Similarity learning with spatial constraints for person re-identification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1268–1277.

[30] H. Liu, B. Ma, L. Qin, J. Pang, C. Zhang, Q. Huang, Set-label modeling and deep metric learning on person re-identification, Neurocomputing 151 (2015) 1283–1292.

[31] R. Zhang, L. Lin, R. Zhang, W. Zuo, L. Zhang, Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification, IEEE Trans. Image Process. 24 (12) (2015) 4766–4779.

[32] S. Wu, Y.C. Chen, X. Li, et al., An enhanced deep feature representation for person re-identification, in: Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2016, pp. 1–8.

[33] L. Wu, C. Shen, A. van den Hengel, Personnet: Person re-identification with deep convolutional neural networks, CoRR (2016). arXiv: 1601.07255

[34] L. Wu, C. Shen, A. van den Hengel, Deep linear discriminant analysis on fisher networks: a hybrid architecture for person re-identification, Pattern Recognition 65 (2017) 238–250.

[35] D. Cheng, Y. Gong, S. Zhou, J. Wang, N. Zheng, Person re-identification by multi-channel parts-based CNN with improved triplet loss function, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1335–1344.

[36] L. Bottou, Stochastic gradient descent tricks, in: Neural Networks: Tricks of the Trade, second ed., Springer, 2012, pp. 421–436.

[37] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: Proceedings of the International Conference on Machine Learning, 2015, pp. 448–456.

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

[39] 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.

[40] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R.B. Girshick, S. Guadarrama, T. Darrell, Caffe: convolutional architecture for fast feature embedding, in: Proceedings of the International Conference on Multimedia, 2014, pp. 675–678.

[41] P.F. Felzenszwalb, R.B. Girshick, D.A. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, IEEE Trans. Pattern Anal. Mach. Intell. 32 (9) (2010) 1627–1645.

[42] S. Liao, S.Z. Li, Efficient PSD constrained asymmetric metric learning for person re-identification, in: Proceedings of the International Conference on Computer Vision, 2015, pp. 3685–3693.

[43] Y. Zhang, B. Li, H. Lu, A. Irie, X. Ruan, Sample-specific SVM learning for person re-identification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[44] 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.

[45] K.Q. Weinberger, L.K. Saul, Distance metric learning for large margin nearest neighbor classification, J. Mach. Learn. Res. 10 (2009) 207–244.

[46] M. Guillaumin, J.J. Verbeek, C. Schmid, Is that you? Metric learning approaches for face identification, in: Proceedings of the International Conference on Computer Vision, 2009, pp. 498–505.

[47] B. McFee, G.R.G. Lanckriet, Metric learning to rank, in: Proceedings of the Twenty-Seventh International Conference on Machine Learning, 2010, pp. 775–782.

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

[49] F. Wang, W. Zuo, L. Lin, D. Zhang, L. Zhang, Joint learning of single-image and cross-image representations for person re-identification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[50] Y. Yang, J. Yang, J. Yan, S. Liao, D. Yi, S.Z. Li, Salient color names for person re-identification, in: Proceedings of the European Conference on Computer Vision, 2014, pp. 536–551.

[51] Z. Shi, T.M. Hospedales, T. Xiang, Transferring a semantic representation for person re-identification and search, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 4184–4193.

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

Yan Huang received his M.S. degree from the School of Computer Science and Engineering, Beihang University, China. His research interests include computer vision, deep learning, machine learning and pattern recognition.

Hao Sheng received his B.S. and Ph.D. degrees from the School of Computer Science and Engineering of Beihang University in 2003 and 2009, respectively. Now he is an associate professor at Beihang University. He is working on computer vision and intelligent transportation systems.

Yanwei Zheng is a Ph.D. student at the School of Computer Science and Engineering of Beihang University. His research fields include computer vision and deep learning.

Zhang Xiong received his B.S. degree from Harbin Engineering University in 1982. He received his M.S. degree from Beihang University in 1985. He is a professor and Ph.D. supervisor in the School of Computer Science and Engineering, Beihang University. He is working on computer vision, information security and data vitalization.