Efficient and Deep Person Re-Identification using Multi-Level Similarity

Yiluan Guo, Ngai-Man Cheung
ST Electronics - SUTD Cyber Security Laboratory
Singapore University of Technology and Design
yiluan_guo@mymail.sutd.edu.sg, ngaiman_cheung@sutd.edu.sg

Abstract

Person Re-Identification (ReID) requires comparing two images of person captured under different conditions. Existing work based on neural networks often computes the similarity of feature maps from one single convolutional layer. In this work, we propose an efficient, end-to-end fully convolutional Siamese network that computes the similarities at multiple levels. We demonstrate that multi-level similarity can improve the accuracy considerably using low-complexity network structures in ReID problem. Specifically, first, we use several convolutional layers to extract the features of two input images. Then, we propose Convolution Similarity Network to compute the similarity score maps for the inputs. We use spatial transformer networks (STNs) to determine spatial attention. We propose to apply efficient depth-wise convolution to compute the similarity. The proposed Convolution Similarity Networks can be inserted into different convolutional layers to extract visual similarities at different levels. Furthermore, we use an improved ranking loss to further improve the performance. Our work is the first to propose to compute visual similarities at low, middle and high levels for ReID. With extensive experiments and analysis, we demonstrate that our system, compact yet effective, can achieve competitive results with much smaller model size and computational complexity.

1. Introduction

In person re-identification (ReID), given one image for a particular person captured by one camera, we need to re-identify this person from multiple images in a gallery captured by different cameras from different viewpoints. This task has attracted much attention due to its various applications in video surveillance and image retrieval. Substantial works have been proposed to accomplish this task, but improvement is still needed. The main challenge is the significant visual appearance changes caused by the illumination variation, occlusion, viewpoint change, person pose as well as background clutter. Also, the computation needs to be efficient to handle large volumes of surveillance videos.

In existing works, Person ReID is solved from two perspectives. One is to develop a powerful representation to discriminate different identities [40, 43, 20] and the other is to design an effective distance metric so that the similarity between different images can be measured [3, 4, 11].

With the great success achieved by deep convolutional nets (ConvNets) in computer vision [18, 33, 12, 9, 13], some works have been proposed to address Person ReID with deep neural networks in an end-to-end fashion. One approach is to classify the images into different identities [7, 19, 30]. During testing, each image is represented by the output of final fully connected layer before the classifier. This approach may suffer from the fact that, in ReID, there are only limited images for each identity during training. [22] formulated Person ReID as a binary classification problem with Siamese network structure. Two images are fed into the network that determines whether they are matched or not. This approach alleviated the problem of insufficient training samples and achieved state-of-the-art results at that time. The critical component of this formulation is how to measure the similarity of two input images. [22] measured the similarity by computing the product of horizontal stripes. [2] extended this idea by taking the neighborhood into consideration. The similarity is computed as the difference between one pixel on feature maps for one image and its 5×5 neighbors on feature maps for another image. [31] further enlarged the neighbor search area and computed the correlation as the similarity.

All these methods have some limitations. First, they only consider the similarity for the outputs of a certain convolutional layer. However, we argue that for ReID, it is useful to use *multi-level similarity*, i.e., similarity of the features of the bottom convolutional layers that contain low level visual information, and that of the higher layers that contain semantic information. Figure 1 contains some illustrative examples. Figure 1(a) and (b) are both non-matching pairs. For Figure 1(a), similarity based on low-level features that capture the color of the shorts can identify the difference of the two persons. For Figure 1(b), the cloth of the two
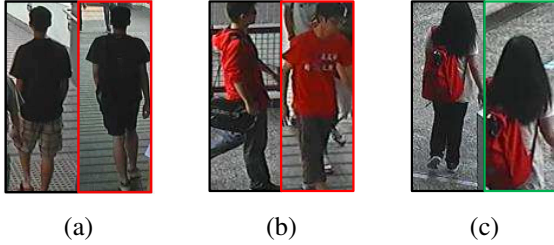
Figure 1. Some examples that are difficult for existing methods. The image in red box indicates it is unmatched and green for matched. (a) and (b) are two unmatched pairs with quite similar appearance. The images in (a) and (b) are different in low level and high level visual features respectively. The red bag in (c) translates for a large distance, which makes the similarity computation difficult.

persons is quite similar and low-level features would fail to distinguish the persons in this case. On the other hand, similarity computed from high-level features can indicate that one is carrying a bag and the other is not. Figure 1(c) is a matching pair. In this case, low and high-level features can be used simultaneously to identify that red color backpacks are present in both images, indicating that there is a high probability that the same person is captured in both images. Second, some previous works assumed that the visual features would not translate for a large distance, thus the computing of product and difference only considering the counterparts on the same or neighboring locations. Although [31] enlarged the search region for computing correlation, it still failed to incorporate the possible correlation for the entire scale, which might lead to missing information. For example, the red color backpack in the two images of Figure 1(c) translates for a substantial distance. Third, in previous works, both the product and difference are computed between the rigid parts from feature maps, which are not invariant to scale, rotation. They are inadequate to handle the case when the two matched images are captured by two cameras with similar angle but from different distances.

In this paper, we propose a fully-convolutional, Siamese network based design to address these issues. First, we compute the visual similarities at different levels of the whole network. Second, given that convolution can be viewed as the computation of "correlation" between a filter and the signal, we formulate the computation of similarity between two images as the convolution between the extracted part from one image (filter) and the other whole image (signal). In this case, the computation is not restricted to a specific search window, addressing the issue of large translation distance. Third, by leveraging Spatial Transformer Networks (STN) [15], we extract meaningful parts from the feature maps of the image. In this case, STNs introduce an attention mechanism into our system. Our contributions are:

- We propose a fully convolutional Siamese network for Person ReID. A new module named Convolution Similarity Network is proposed to improve the measurement of similarity between two input images. This new module exploits the attention mechanism and could be implemented efficiently.

- We compute visual similarities at different levels and combine them to achieve robust matching/non-matching classification.

- We conduct extensive experiments and show that our method achieves competitive results in comparison with state-of-the-arts, with a lower computational complexity and model memory.

2. Related Work

Most existing methods for Person ReID could be divided into two classes, one is traditional methods and the other is deep learning based approaches. For the traditional methods, there are usually two stages. First, handcrafted features are computed such as color histogram [40, 17], Gabor features [20] and dense SIFT [43].The handcrafted features are expected to contain as much as possible discriminative information for different persons. Following stage is similarity metric learning. Different metric learning approaches have been proposed to decide whether two images are matched or not [3, 4, 11, 23]. A suitable metric should indicate the similarity of two images based on the handcrafted features, *i.e.*, two images for the same person should have a smaller distance than those for different persons. Some work even adopted an ensemble of different metrics [26]. Since the feature extraction stage and metric learning are two independent components in traditional methods, the optimization of features and distance metric might not help each other and eventually become sub-optimal. Our proposed method is significantly different from all these traditional ones as we jointly learn the features and metric in a deep neural network.

On the other hand, deep ConvNets have achieved great success in computer vision tasks like object recognition, detection, semantic segmentation [18, 28, 25]. Recently, some published work show the promising power of deep ConvNets in person ReID. [22] proposed a Siamese network that takes a pair of images to be compared. Convolutional layers are used to extracted visual features and product is used to indicate the similarity. [2] proposed an improved architecture where neighbor difference were used to measure the similarity. [31] further extends this architecture by enlarging the neighbor search region and normalize the elements before computing product. All the above works formulate the Person ReID task as a binary classification problem. The difference between our work is that we leverage the spatial attention and integrate the computation of

similarities at different levels into the fully convolutional structure. Meanwhile, there is another line of approaches formulating this task as a ranking problem [5, 8, 35, 42]. There are two or three images as input. Contrastive loss or triplet loss are used to push the images for the same identity closer together and pull the images for different identities more far away in the embedding space. More recently, the combination of classification and ranking loss obtained promising results by taking advantage of both ranking and binary classification tasks [37, 6]. Our method also adopts multiple tasks to train the network with the difference that the ranking loss is based on the attended regions extracted by STNs instead of the descriptors of the whole images because the meaningful parts of images excluding noise and redundancy are more effective to represent the identities.

Another interesting approach treats ReID as recognition problem and classifies the images to different identities directly. [41, 30] extracted different body parts by human pose and combined local and global features for classification. [7, 27], on the other hand, considered the features at different scale. Among these works, [19] proposing to use STNs to find the meaningful local parts is similar to ours. However, our method has a different goal for the usage of STNs: we want to compute the similarity explicitly. The network structure is also distinct since we build a Siamese network and the final object is binary classification.

[32] exploited the idea of visual similarities on multi-level for face recognition, whereas our approach is proposed for the task of ReID. Moreover, we introduce attention mechanism and improve the similarity computation, which make the multi-level similarity more accurate and effective.

3. Proposed Method

3.1. Model description

The overall structure of the proposed model is shown in Figure 2. Two input images I_1 and I_2 ($60 \times 160 \times 3$ in our experiments) are processed by three successive convolution layers. Let $x_i^{(j)}$ denote the output of the j-th convolution layer for I_i. The output of the second and third convolution layers, $x_1^{(2)}, x_2^{(2)}$ as well as $x_1^{(3)}, x_2^{(3)}$ are fed into two Convolution Similarity Networks (CSNs), which have two sets of outputs: one is the similarity score maps for I_1 and I_2 while the other is feature maps for the extracted local parts of I_1 and I_2. The similarity score maps are processed by three more convolution layers. The details of convolutional layer 1-6 are listed in Table 1. Two nodes in the final layer indicate whether I_1 and I_2 are matched or not, i.e., binary classification. An additional objective function is to make the matched pair closer and unmatched pair far away in an embedding space.

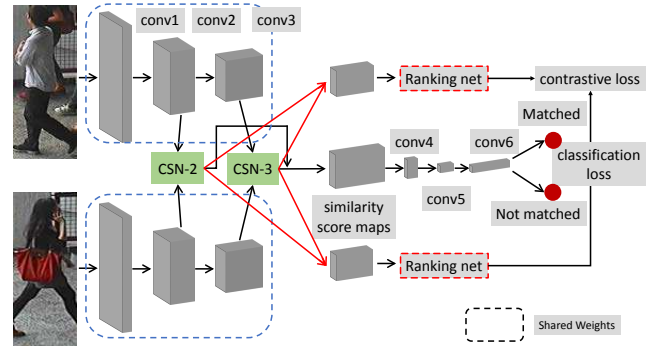**Convolution similarity network** (CSN) is proposed to

Figure 2. The structure of proposed method. The outputs of CSN-2 and CSN-3 indicated by black arrows are concatenated together for further processing. The outputs of CSN-2 and CSN-3 indicated by red arrows are fed into the ranking net. Networks in the dash line boxes with the same color means that they are sharing the same parameters. Details of the CSN can be found in Figure 3 and related context.

Table 1. Network specifications. All the convolutional layers except the conv4, conv6 and loc conv3 are followed by 2×2 max-pooling. The conv6 and loc conv3 are followed by global average pooling (GAP) layers[24].

Network	Layer	filter size	#filters
Whole structure	conv1	$5 \times 5 \times 3$	32
	conv2	$3 \times 3 \times 32$	96
	conv3	$3 \times 3 \times 96$	96
	conv4	$1 \times 1 \times 1152$	32
	conv5	$3 \times 3 \times 32$	32
	conv6	$1 \times 1 \times 32$	500
Localization Net	loc conv1	$3 \times 3 \times 96$	32
	loc conv2	$3 \times 3 \times 32$	32
	loc conv3	$1 \times 1 \times 32$	128

measure the similarity of two inputs. The framework of CSN is shown in Figure 3. Given feature maps of two images, we propose efficient comparison with CSN: first, meaningful local regions are extracted by STNs; second, the local parts are treated as filters and thus the correlation between two groups of feature maps are computed in a more efficient way with fully convolutional structure. We will describe these two stages in details.

To find the meaningful contents from a pedestrian image is of great importance and challenging due to the large view point variation and occlusion. Spatial Transform Networks (STNs) are proven to be effective for images containing one kind of objects, which suits our application well. Therefore, we decide to use STN in our network to integrate the spatial attention.

There are three components in an STN. The localization net learns the transformation parameters. We consider affine transformation which has 6 parameters. The grid generator and sampler together samples the input image and generates a new image with bilinear interpolation. In this case, the transformation is
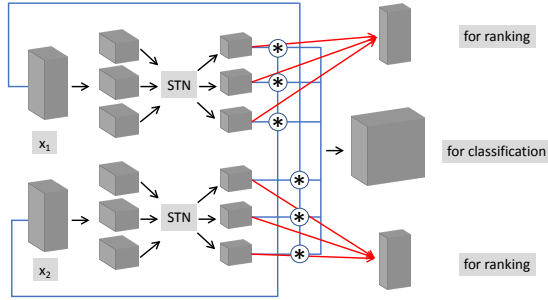
Figure 3. The framework of CSN. * denotes depth-wise convolution. All the results of the depth-wise convolution are concatenated together as the outputs. The other outputs indexed by the red arrow are the feature maps extracted by the STN, which are used as the inputs for the ranking net.

$$\begin{pmatrix} w^{in} \\ h^{in} \end{pmatrix} = \begin{pmatrix} s_w & r_w & t_w \\ r_h & s_h & t_h \end{pmatrix} \begin{pmatrix} w^{out} \\ h^{out} \\ 1 \end{pmatrix} \quad (1)$$

where w_{in} and h_{in} are the normalized pixel coordinates for input image and w_{out} and h_{out} are the normalized pixel coordinates for output image along the width and height. s, r and t are the scale, rotation, translation parameters. We suggest readers to have more details from [15].

We have two fully convolutional STNs, STN_2 and STN_3 in our model for the $x_i^{(2)}$ and $x_i^{(3)}$ respectively. The structures of their localization nets are the same, shown in Table 1, but the weight parameters are not shared. A linear embedding with dimension 6 and hyperbolic tangent activation function are followed to output the 6 transformation parameters. We found that it was difficult to find the relatively important part in a global scale from $x_i^{(j)}$ by STNs. Therefore, $x_i^{(j)}$ are divided into three parts, namely the upper, the middle and the bottom, which are overlapped each other to some extent. The overlapping between two adjacent parts makes sure that the meaningful local visual features are covered. Note that all the three parts are sharing the same localization net.

The size of the outputs of samplers is set to be $f_2 \times f_2$ for $x_i^{(2)}$ and $f_3 \times f_3$ for $x_i^{(3)}$. The value of f_2 is set to be larger than f_3 because the receptive fields for elements on $x_i^{(3)}$ are larger than those on $x_i^{(2)}$. Let $x_i^{(j,upper)}$, $x_i^{(j,middle)}$, $x_i^{(j,bottom)}$ denote the outputs of the three parts from x_i^j. Each of them is with the size $f_2 \times f_2 \times 96$ for $j = 2$ and $f_3 \times f_3 \times 96$ for $j = 3$. Now we have found the meaningful part of given feature maps. To search the corresponding similar features in another image, we treat the extracted parts as filters, slide them all over the feature maps of another image with stride 1, which is like what a convolutional layer does.

The similarity, modeled as cross-correlation between them, is described as

$$s_i^{(j,part)} = x_i^j * x_{i'}^{(j,part)} \quad (2)$$

Here $i, i' \in \{1, 2\}$ are different, indicating two input images. $part$ can be $upper, middle, bottom$. $*$ denotes depth wise convolution. This step can be implemented efficiently in existing deep learning frameworks, which gets rid of sampling the rigid parts from feature maps and comparing them with another mechanically. We choose depth wise convolution instead of traditional one due to the fact that different feature maps contain different activation patterns. Since the signal x_i^j to be convolved and the filter $x_{i'}^{(j,part)}$ have the same number of channels and x_i^j are padded with zero, $s_i^{(j,part)}$ have the same size as x_i^j. In order to be symmetric and fully exploit the similarity, depth wise convolution is performed between $x_2^{(j,part)}$ and $x_1^{(j)}$ as well as $x_1^{(j,part)}$ and $x_2^{(j)}$. Now for $j = 2, 3$, we have 6 groups of $s_i^{(j,part)}$ respectively and we concatenate all the 6 groups along the channel direction and further do 2×2 maxpooling for $j = 2$ to reduce noise and redundancy. The results are then the comprehensive similarity score maps between $x_1^{(j)}$ and $x_2^{(j)}$, denoted as sim_j.

Combination of visual similarities from different levels. It is well known that bottom convolutional layers in deep ConvNets contain low level features such as color, shape, texture, *etc.*, and higher convolutional layers learn complex and semantic information. In our case, we take the second and the third convolutional layers into account. sim_2 focuses on the low level visual similarity while sim_3 focuses on the relatively higher level visual similarity. sim_2 and sim_3 are concatenated together along the channel direction. Since there are 12 groups of depth wise convolution in total, now the size of the similarity score maps is $10 \times 4 \times 1152$, where $1152 = 96 \times 12$. These similarity score maps contain comprehensive information for the final decision.

3 convolutional layers(conv4, conv5, conv6 in Figure 2) are followed to process the similarity score maps. Specifically, 1×1 convolution is used to reduce the number of channels first. GAP is used to replace the fully connected layer to keep the fully convolutional structure.

Objective function used to train the network is the combination of classification and ranking. Softmax loss is the objective function for binary classification.

$$L_{cls} = \frac{1}{m} \sum_{i=1}^{m} [(1-y)p(y=0|\{x_1,x_2\})] + yp(y=1|\{x_1,x_2\}) \quad (3)$$

where $y = 1$ when the input images are matched and $y = 0$ otherwise. $p(y|\{x_1, x_2\})$ is the probability distribution of y given input $\{x_1, x_2\}$, computed by softmax function. m is the mini batch size.

The binary classification objective function intends to train a high accuracy model which somehow ignores the correct ranking. The combination of binary classification and contrastive loss may alleviate this issue and improve the performance substantially, which has been observed by [37, 6]. However, their computation of contrastive or triplet loss depends on the global descriptor of the whole image. We argue that the global descriptors are not ideal for ranking task since they do not highlight the more discriminative parts of the original images. This issue can be resolved by our model. For the matched image pairs, it is reasonable to believe that they have the similar extracted parts, namely, local visual features. Given $x_i^{(j,upper)}$, $x_i^{(j,middle)}$, $x_i^{(j,bottom)}$ with spatial attention, we propose a ranking net to compute the ranking loss, which only consists of 3 convolutional layers. $x_i^{(j,upper)}$, $x_i^{(j,middle)}$, $x_i^{(j,bottom)}$ are firstly go through a convolutional layer with 96 filters of size 3×3 and a max-pooling layer. Then the $x_i^{(j,upper)}$, $x_i^{(j,middle)}$, $x_i^{(j,bottom)}$ are concatenated together for each i along the vertical direction as they are extracted from different horizontal stripes. After another convolutional layer with 96 filters of size 3×3, the feature maps from the different layers, indexed by j, are concatenated again to form the descriptors for attended local parts. Then we also use GAP and linear embedding to obtain a 256 dimensional vector to represent the attended parts of one input image. The vectors for two images are L_2 normalized to make them comparable. Contrastive loss is computed for the two input images,

$$L_{ctr} = \frac{1}{2m} * \sum_{i=1}^{m} [yd^2 + (1-y)\max(0, \alpha - d)^2] \quad (4)$$

$$d = \|\mathbf{r_1} - \mathbf{r_2}\| \quad (5)$$

where $\mathbf{r_1}$ and $\mathbf{r_2}$ are the representations for two input images. d is the Euclidean distance. α is the margin set to be 1.0 in this paper. With the help of this contrastive loss, images with similar attended parts are pushed closer in the embedding space. Otherwise, they are pulled further.

The whole network is trained end-to-end with the combination of mentioned losses.

$$L_{com} = L_{cls} + L_{ctr} \quad (6)$$

During testing, one query image and one image from the gallery are fed into the network. The final similarity score is computed as

$$\text{SimiScore} = s_{softmax} + \lambda * \frac{1}{d + \epsilon} \quad (7)$$

where $s_{softmax}$ is the matched probability computed by softmax function and d is the Euclidean distance in Eq. 5. λ is set to be 0.2 empirically and ϵ is set to be a small value like 0.0001. All the images in the gallery are ranked based on their final similarity scores.

3.2. Discussion

Efficiency. We keep the implementation efficiency in mind when designing the model. In [31], the rigid local parts are sampled mechanically and compared with a restricted region of another image. However, sampling a rigid part directly from a tensor is usually avoided in existing deep learning frameworks for efficiency since it needs to index the elements from tensors. Meanwhile, in the ReID application, the local rigid parts may not cover the important visual features and are not invariant to scale, translation and rotation. In contrast, sampling local meaningful part is done by a fully convolutional STN in our network, which is more flexible and effective. The sampled parts play the role of filters in a traditional convolutional layer, which is compatible with current deep learning frameworks, thus the implementation being much easier.

Learned visual similarity from different levels. To verify that our model indeed learns the visual similarities at different levels, we conduct some visualization experiments on the similarity score maps for dataset CUHK03 detected, shown in Figure 4. The similarity score maps in column 3 and 4 are the convolution results between the feature maps for query image and the attended upper and middle parts for the test image. For the positive pair in (a), it could be inferred that the similarity score map from CSN-2 focuses on the skin texture related features so that the exposed human skin parts like face and hands in the image get higher similarity scores, which again proves that STNs successfully grasp the meaningful local features. CSN-3, on the contrary, restricts its similarity on the face part only, which proves our assumption that CSN-2 and CSN-3 are dealing with low-level and complex semantic visual similarity. In (b), the negative testing image is quite confusing for the CSN-2 since the similarity score maps in the third column almost look the same because of the red tops. However, the similarity score maps from CSN-3 are easily distinguished due to the different similarity value in the bag area. It can be inferred that the similarity score maps from CSN-2, in this case, may mislead the model. However, the significant difference of the similarity score maps from CSN-3 will make sure that the model will give the right prediction. We can conjecture that when the high level visual similarities are confusing, the low level ones will help in turn. From the visualization, we can conclude that the combination of different level similarity is necessary for final success.

Model extension. There are two aspects to explore for model extension. On the one hand, we can have more CSNs in our network structure since our proposed CSN is fully differentiable and could be inserted in the network anywhere. We consider CSN-4 in experiments, which fol-
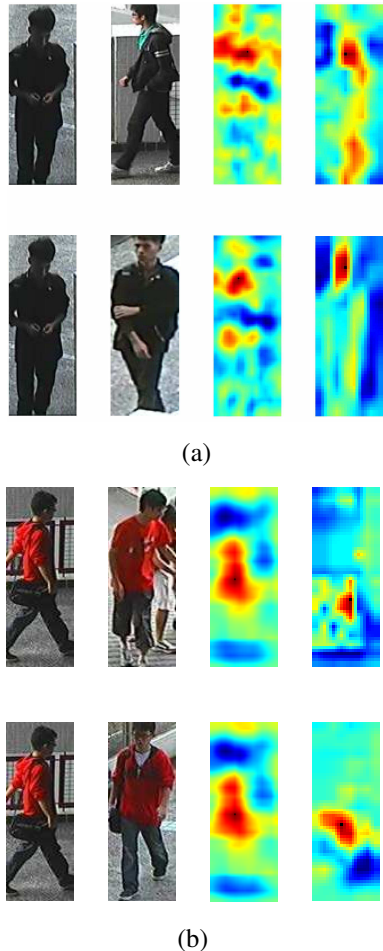
(a)

(b)

Figure 4. Visualization of the input images and similarity score maps. Query images are in the first column. Testing images in second column are unmatched and matched ones. The similarity score maps in column 3 and 4 are from CSN-2 and CSN-3 respectively.

lows another convolutional layer after conv3 in Figure.2 and has the same structure as other CSNs. The CMC results on CUHK03 dataset in Table 2 show that including more higher level visual similarity will indeed increase the performance by a large margin. On the other hand, we can achieve comparable performance with state-of-the-art methods leveraging pre-trained network, such as VGG [29], ResNet [12], *etc.*, in spite of simple three or four convolutional layers for feature extraction in our model.

4. Experiments

4.1. Datasets and evaluation metrics

We test our model on four dataset: CUHK03 detected and labeled [22], CUHK01 [21], VIPeR [10]. CUHK03 is a large dataset containing 13,164 images fro 1,360 identities captured by 6 cameras. This dataset has two kinds of

pedestrian boxes: detected by algorithms and labeled manually, both of which we will use. Following the setting as [22], we randomly choose 1160 identities for training, 100 for validation and 100 for testing. CUHK01 is a middle size dataset containing 3884 images of 971 identities. For our experiments, we follow the setting as [31] and randomly choose 871 identities for training and 100 for testing. VIPeR is a small size datasets with 632 identities, for which we randomly choose half of them for training and half for testing.

Cumulative Matching Characteristics (CMC) are reported to evaluate the performance. There are only one query image and one matched image in the gallery for each testing identity, *i.e.*, single shot setting. Rank-k accuracy stands for the accuracy that the matched image in the gallery is included in the top-k answers based on the similarity score.

4.2. Implementation details

We implement our model with TensorFlow [1]. ADAM [16] is used to optimize the network with learning rate 0.0005. We train the network for 5 epochs. Weighting decay is set to be 0.0005 to avoid over-fitting. Batch normalization [14] is used to make the training stable and fast to converge. The mini batch size is set to be 256 for CUHK03 and 128 for other two datasets. f_1 and f_2 are set to be 10 and 5 respectively. Data augmentation is also adopted for training as [2] and [31]. We randomly sample 2 images for CUHK03 and 5 for others from the original image center and also flip it horizontally. On the other hand, since the negative pairs in the training set outnumber the positive pairs significantly, the model easily falls into over-fitting and predicts all the pairs as negative. Therefore we only randomly choose two negative pairs for each positive pair. Note that we do not introduce hard negative mining, which simplifies the training process. We use one NVIDIA TitanX GPU to train the model. During inference, the model takes 1250 pairs of images as input and obtains the final score in about 1.6s.

Empirically, we found that learning the transformation parameters without constraints would cause several issues, such as negative scale parameters, falling out of the original image and so on, which are also pointed out by [19]. Therefore, some similar prior constraints are put on the 6 transformation parameters, which are used to keep the scale parameters larger than 0, which avoids the upside down case, and to keep the results staying in the original images. In addition, we simply use L_1 regularization for r_w and r_h since rotation is seldom happened in real world cases. As we discussed before, we divide the feature maps into three parts for the Localization Net. In particular, the upper part of $x_i^{(2)}$ is composed of the row 1 to 20 of $x_i^{(2)}$, middle row 10 to 30, bottom row 20 to 40. Similarly, the upper part of $x_i^{(3)}$

Table 2. The CMC results comparison between our method and other state-of-the-art methods.

Method	CUHK03 detected			CUHK03 labeled			CUHK01			VIPeR		
	top-1	top-5	top-10	top-1	top-5	top-10	top-1	top-5	top-10	top-1	top-5	top-10
FPNN	19.89	48.70	64.79	20.65	51.50	68.50	27.87	64.50	73.46	-	-	-
ImpCNN	44.96	76.50	83.47	54.74	87.80	93.88	65.00	89.00	93.12	-	-	-
Joint	52.17	85.30	91.20	-	-	-	71.80	90.00	93.50	35.76	66.70	84.50
SiameseLSTM	57.30	80.10	88.30	-	-	-	-	-	-	42.40	68.70	79.40
S-CNN	68.10	88.10	94.60	-	-	-	-	-	-	37.80	66.90	77.40
BDLatPart	67.99	91.04	95.36	74.21	94.33	97.54	-	-	-	-	-	-
ImpTriplet	-	-	-	-	-	-	-	-	-	47.80	**74.70**	84.80
X-Corr	72.04	92.10	96.00	72.43	92.50	95.51	81.23	95.00	97.39	-	-	-
Quadruplet	75.53	95.15	**99.16**	-	-	-	81.00	96.50	98.00	49.05	73.10	81.96
DGD	-	-	-	72.58	91.59	95.21	66.60	-	-	38.6	-	-
MTDNet	74.68	95.99	97.47	-	-	-	78.50	96.50	97.50	47.47	73.10	82.59
MuDeep	75.64	94.36	97.46	76.87	96.12	98.41	79.01	97.00	98.96	43.03	74.36	**85.76**
DPFL	82.00	-	-	86.70	-	-	-	-	-	-	-	-
DeepAlign	81.60	97.30	98.40	85.40	97.60	99.40	**88.50**	**98.40**	**99.60**	48.70	**74.70**	85.10
PDC	78.29	94.83	97.15	**88.70**	**98.61**	99.24	-	-	-	51.27	74.05	84.18
Spindle	-	-	-	88.50	97.80	98.60	79.90	94.40	97.10	**53.80**	74.10	83.20
JLML	80.60	96.90	98.70	83.20	98.00	99.40	-	-	-	50.20	74.20	84.30
Ours-(L2, L3)	79.45	94.70	97.90	80.30	97.10	98.35	86.55	97.70	98.70	48.03	72.90	82.15
Ours-(L2, L3, L4)	**86.45**	**97.50**	99.10	87.50	97.85	**99.45**	88.20	98.20	99.35	50.10	73.10	84.35

is composed of the row 1 to 10 of $x_i^{(3)}$, middle row 5 to 15, bottom row 10 to 20.

4.3. Comparison with state-of-the-arts

We compare our approach with several state-of-the-art methods in recent years, including FPNN [22], ImpCNN [2], SiameseLSTM [36], S-CNN [35], BDLatPart [19], MTDNet [6], X-Corr [31], Quadruplet [5], ImpTriplet [8], DGD [39], Joint [37], MuDeep [27], Spindle [41], DPFL [7], DeepAlign [42], PDC [30] and JLML [38]. In particular, we want to compare our method with MTDNet, X-Corr and BDLatPart. MTDNet combines binary classification and ranking together with the global descriptors alone for input images. X-Corr is a Siamese network computing the similarity as correlation between rigid parts of input images. BDLatPart introduces the STN into their structure to extract local meaningful parts. However, BDLatPart only considers to learn the representation for one input image, which significantly differs from our work. The results are shown in Table 2. The methods in the upper part train their models from scratch on the ReID datasets alone while the methods in the middle part either train a sub-network on another datasets for more supervision or use pre-trained networks like Inception-V3 as backbone structure. As we could guess, the middle methods usually outperforms the upper ones with the help of pre-training or more information. We implemented two models, one is the same as Figure 2 denoted as Ours-(L2, L3) since it considers visual similarities from the second and the third convolutional layers and the other including one more CSN to consider higher level visual similarity as discussed in Section 3.2, denoted as Ours-(L2, L3, L4).

Our extended model achieves the best top-1 accuracy for CUHK03 detected datasets, outperforming all the methods

by a large margin. On the other three datasets, our extended model can get performance comparable to state of the art methods with smaller model size and computation amount. Even our weaker model, ours-(L2, L3), can beat all the methods in the upper table for CUHK03 and CUHK01 datasets. As expected, X-Corr suffers from the mechanical correlation computation and restricted comparing regions while BDLatPart fails to extract discriminative enough representations for each identity even with the help of STNs, which convinces us that our model has the better way to compute similarity effectively with the usage of STNs. The performance of MTDNet is also inferior to our approach due to the lack of explicit similarity computation and spatial attention on the discriminative local parts. The results shown here demonstrate the effectiveness of our similarity computation at multiple levels.

4.4. Ablation analysis

To further understand our model, we conduct several ablation experiments for our model on CUHK03 detected dataset, which contains the images more similar to the real world application.

First, we remove the contrastive loss L_{ctr} function and train the network with the same settings carefully. The CMC results, denoted as ours-cls, are shown in Table 3. We can see that without the ranking loss, the performance degrades 3% for the rank 1 accuracy.

Then we investigate the importance of visual similarities at different levels of the proposed network. We keep only one CSN in the model and train the network under the same strategy. New models are denoted as Ours-L2, Ours-L3 and Ours-L4. The results are shown in table 3. As we discussed before, CSN-2 computes the low level visual similarity such as edge, shape, *etc.* while CSN-3 and CSN-4 focuses on the

higher level similarity containing semantical information. We can find that the high level visual similarity is more important than the lower level one. The CSN-4 alone helps the model achieves similar performance to the combination of CSN-2 and CSN-3. However, all the models with single CSN obtain inferior performances to the one utilizing combination of 3 CSNs, indicating that low level similarity provides additional information ignored by the high level one and thereby the necessity of combing low and high level visual similarities.

Table 3. The CMC results comparison between our original method and modified ones on CUHK03 detected dataset.

Method	top-1	top-5	top-10
Ours-cls	75.90	94.55	97.85
Ours-L2	74.70	93.52	96.45
Ours-L3	76.55	93.70	96.85
Ours-L4	79.15	94.45	97.80
Ours-(L2, L3)	79.45	94.70	97.90
Ours-(L2, L3, L4)	**86.45**	**97.50**	**99.10**

Last but not least, different configurations of the proposed network are studied. We examine the usefulness of dividing the images into three horizontal stripes in C1. In C2, we replace adaptive STN with fixed central cropping, i.e., we crop a center region with the same size as STN from each horizontal stripe. C3 is our proposed model with only Level 4 similarity and C4 is the original model with Multi Level similarities. Results on CUHK03 detected dataset are shown in Table 4. Comparing C1 and C4, we can observe that without dividing, it becomes difficult for STN to find the meaningful regions. In fact, the result of STN without dividing is worse than central cropping(C2). C2 achieves reasonable results when multi level similarities are used, which demonstrates the effectiveness of multi level similarities. With the combination of dividing and STN, we can compute more accurate similarity score maps from different feature levels, and this leads to the best performance as shown in C4.

Table 4. CMC results for different configurations on CUHK03 detected datasets. ML here means Multi Level(L2, L3, L4).

config.	dividing	STN	ML	top-1	top-5	top-10
C1	×	✓	✓	79.40	94.95	98.40
C2	✓	×	✓	82.00	96.40	98.55
C3	✓	✓	×	79.15	94.45	97.80
C4	✓	✓	✓	**86.45**	**97.50**	**99.10**

4.5. Complexity Analysis

We compare the proposed model with five recent proposed models in model size and computation complexity, which are measured by the number of parameters and the value of FLOPs during inference. X-Corr[31] and BDLatPart [19] are trained from scratch so we estimate the

number of parameters and FLOPs by ourselves. DPFL [7] and DeeAlign [42] use pre-trained Inception-V3 [34] and GoogLeNet [33] as their backbone structures, which are considered as main contributors for complexity. JLML [38], based on ResNet39, discloses the complexity in their paper. Table 5 shows that our original model has the smallest model size and least computation amount while outperforming X-Corr[31] and BDLatPart[19] by a large margin (8% to 10%) on CUHK03 detected dataset. The performance of our extended model, ours-(L2, L3, L4), with the second lowest complexity, can be comparable to the methods with pre-trained networks.

Table 5. Comparison of model size and complexity. #param: number of parameters. M: Million. G: Giga.

Model	#param(M)	FLOPs(G)	Depth
X-Corr	2.2	1.58	**10**
BDLatPart	1.4	1.80	25
DPFL	35	6.00	40
JLML	7.2	1.54	39
DeepAlign	6	1.57	22
Ours-(L2, L3)	**0.5**	**0.96**	**12**
Ours-(L2, L3, L4)	**0.8**	**1.31**	18

5. Conclusion

In this work, we propose a novel fully convolutional Siamese network for Person ReID. Our system extracts features from local parts of one input image and then computes the visual similarity with another input image through depth-wise convolution. By exploiting two or more CSNs at different convolutional layers, we obtain visual similarities at different levels. This approach avoids sampling the rigid parts of input images and could be implemented efficiently. We further enhance the system by considering contrastive loss based on descriptors for the extracted local parts. Extensive experiments on four Person Re-ID datasets show that our approach could achieve comparable performance with recent state-of-the-art, at a lower computational complexity and model size. Ablation and visualization experiments show that the visual similarities from different levels all contribute to the overall improvement. We also provide the comparison in model size and complexity and show that our method can achieve good performance at lower complexity.

Acknowledgment

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] E. Ahmed, M. Jones, and T. K. Marks. An improved deep learning architecture for person re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[3] D. Chen, Z. Yuan, G. Hua, N. Zheng, and J. Wang. Similarity learning on an explicit polynomial kernel feature map for person re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[4] J. Chen, Z. Zhang, and Y. Wang. Relevance metric learning for person re-identification by exploiting global similarities. In *The IEEE International Conference on Pattern Recognition (ICPR)*, August 2014.

[5] W. Chen, X. Chen, J. Zhang, and K. Huang. Beyond triplet loss: A deep quadruplet network for person re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[6] W. Chen, X. Chen, J. Zhang, and K. Huang. A multi-task deep network for person re-identification. In *AAAI Conference on Artificial Intelligence (AAAI)*, February 2017.

[7] Y. Chen, X. Zhu, and S. Gong. Person re-identification by deep learning multi-scale representations. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2017.

[8] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[9] T. Do, A. Doan, and N. Cheung. Learning to hash with binary deep neural network. In *European Conference on Computer Vision (ECCV)*, October 2016.

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

[11] M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? metric learning approaches for face identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2009.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[13] T. Hoang, T. Do, D. Tan, and N. Cheung. Selective deep convolutional features for image retrieval. In *ACM Multimedia Conference (ACM MM)*, October 2017.

[14] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, July 2015.

[15] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Annual Conference on Advances in Neural Information Processing Systems (NIPS)*, December 2015.

[16] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *The International Conference on Learning Representations (ICLR)*, May 2015.

[17] M. Koestinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof. Large scale metric learning from equivalence constraints. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Annual Conference on Advances in Neural Information Processing Systems (NIPS)*, December 2012.

[19] D. Li, X. Chen, Z. Zhang, and K. Huang. Learning deep context-aware features over body and latent parts for person re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[20] W. Li and X. Wang. Locally aligned feature transforms across views. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[21] W. Li, R. Zhao, and X. Wang. Human reidentification with transferred metric learning. In *Asian Conference on Computer Vision (ACCV)*, November 2012.

[22] W. Li, R. Zhao, T. Xiao, and X. Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[23] Z. Li, S. Chang, F. Liang, T. S. Huang, L. Cao, and J. R. Smith. Learning locally-adaptive decision functions for person verification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[24] M. Lin, Q. Chen, and S. Yan. Network in network. In *The International Conference on Learning Representations (ICLR)*, April 2014.

[25] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[26] S. Paisitkriangkrai, C. Shen, and A. van den Hengel. Learning to rank in person re-identification with metric ensembles. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[27] X. Qian, Y. Fu, Y.-G. Jiang, T. Xiang, and X. Xue. Multi-scale deep learning architectures for person re-identification. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2017.

[28] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Annual Conference on Advances in Neural Information Processing Systems (NIPS)*, December 2015.

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *The International Conference on Learning Representations (ICLR)*, May 2015.

[30] C. Su, J. Li, S. Zhang, J. Xing, W. Gao, and Q. Tian. Pose-driven deep convolutional model for person re-identification. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2017.

[31] A. Subramaniam, M. Chatterjee, and A. Mittal. Deep neural networks with inexact matching for person re-identification. In *Annual Conference on Advances in Neural Information Processing Systems (NIPS)*, December 2016.

[32] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[33] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[34] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[35] R. R. Varior, M. Haloi, and G. Wang. Gated siamese convolutional neural network architecture for human re-identification. In *European Conference on Computer Vision (ECCV)*, Octorber 2016.

[36] R. R. Varior, B. Shuai, J. Lu, D. Xu, and G. Wang. A siamese long short-term memory architecture for human re-identification. In *European Conference on Computer Vision (ECCV)*, October 2016.

[37] F. Wang, W. Zuo, L. Lin, D. Zhang, and L. Zhang. Joint learning of single-image and cross-image representations for person re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[38] S. G. Wei Li, Xiatian Zhu. Person re-identification by deep joint learning of multi-loss classification. In *International Joint Conference on Artificial Intelligence (IJCAI)*, August 2017.

[39] T. Xiao, H. Li, W. Ouyang, and X. Wang. Learning deep feature representations with domain guided dropout for person re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[40] F. Xiong, M. Gou, O. Camps, and M. Sznaier. Person re-identification using kernel-based metric learning methods. In *European Conference on Computer Vision (ECCV)*, September 2014.

[41] H. Zhao, M. Tian, S. Sun, J. Shao, J. Yan, S. Yi, X. Wang, and X. Tang. Spindle net: Person re-identification with human body region guided feature decomposition and fusion. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[42] L. Zhao, X. Li, Y. Zhuang, and J. Wang. Deeply-learned part-aligned representations for person re-identification. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2017.

[43] R. Zhao, W. Ouyang, and X. Wang. Person re-identification by salience matching. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.