End-to-End Comparative Attention Networks for Person Re-identification

Hao Liu, Jiashi Feng, Meibin Qi, Jianguo Jiang and Shuicheng Yan, Senior Member, IEEE

Abstract—Person re-identification across disjoint camera views has been widely applied in video surveillance yet it is still a challenging problem. One of the major challenges lies in the lack of spatial and temporal cues, which makes it difficult to deal with large variations of lighting conditions, viewing angles, body poses and occlusions. Recently, several deep learning based person re-identification approaches have been proposed and achieved remarkable performance. However, most of those approaches extract discriminative features from the whole frame at one glimpse without differentiating various parts of the persons to identify. It is essentially important to examine multiple highly discriminative local regions of the person images in details through multiple glimpses for dealing with the large appearance variance.

In this paper, we propose a new soft attention based model, i.e., the end-to-end Comparative Attention Network (CAN), specifically tailored for the task of person re-identification. The end-to-end CAN learns to selectively focus on parts of pairs of person images after taking a few glimpses of them and adaptively comparing their appearance. The CAN model is able to learn which parts of images are relevant for discerning persons and automatically integrates information from different parts to determine whether a pair of images belongs to the same person. In other words, our proposed CAN model simulates the human perception process to verify whether two images are from the same person. Extensive experiments on three benchmark person re-identification datasets, including CUHK01, CHUHK03 and Market-1501, clearly demonstrate that our proposed end-to-end CAN for person re-identification outperforms well established baselines significantly and offer new state-of-the-art performance.

Index Terms—Person re-identification, Comparative Attention Network, Multiple glimpses.

I. INTRODUCTION

RECENTLY, person re-identification (re-id), i.e., person or pedestrian re-identification across multiple cameras without overlapping view, has received increasing attention [1]–[17]. It aims to re-identify a person that has been captured by one camera in another camera at any new location. Person re-identification has many important applications in security systems and video surveillance of public scenarios such as stores and shopping malls. However, it is still challenging to obtain satisfactory results in terms of accuracy in real-world scenarios due to the large appearance variance across multiple cameras. For example, as shown in Fig. 1 people usually pose differently over time. Besides, other factors such as variations in color, illumination, occlusion as well as low-resolution of the captured frames also increase difficulties of the realistic person re-identification.

According to related research [1]–[17] and our daily experience, in the process of a human discerning another in a crowd, the human often abstracts the discriminative features of all the individuals and then compare the similarity and difference of them to find the specific one correctly, and this process can be repeated many times (i.e. multiple glimpses of each person). At the end of the process, the information gathered from glimpses are integrated as the comprehensive information to help the discerning. Inspired by this observation, we propose attention
based model with inherent comparative components to solve the person re-identification problems.

With the recent development of Recurrent Neural Networks (RNNs) based on Long Short-Term Memory (LSTM) [13], the attention based models have demonstrated outstanding performance on several challenging sequential data recognition and modeling tasks, including caption generation [19], machine translation [20], as well as action recognition [21]. Briefly, similar to human visual processing, attention-based algorithms tend to selectively concentrate on a part of the information, and at the same time ignore other perceived information. Such a mechanism is usually called attention and can be employed to adaptively localize discriminative parts or regions of person images. Thus it is helpful to solve the person re-identification problem, which however has been rarely considered in the literatures.

In this work, we go beyond the standard LSTM based attention models and propose an end-to-end Comparative Attention Network (CAN). The proposed end-to-end CAN framework simulates the re-identification process of human visual system by learning a comparative model from raw person images to recurrently localize some discriminative parts of person images via a set of glimpses. At each glimpse, the model generates different parts without any manual annotations. It takes both the raw person images and the locations of a previous glimpse as inputs, and produces the next glimpse local region features as the outputs. These features can be regarded as a kind of dynamical pooling feature, and we show that exploiting these features generated by our CAN model for person re-identification performs better than conventional pooling features, which is used by many existing models [13], [14], [16], [17]. In contrast, our approach is also able to achieve comparatively better performance compared to other methods, as validated by experimental results.

In summary, we make following contributions to person re-identification:

- We propose a new attention model that dynamically generates discriminative features in a recurrent way of “seeing” and “comparing” person images for automatically localizing the most discriminative parts of persons.
- We develop a comparative network that can efficiently seek discriminative parts of person image pairs by incorporating an on-line triplet selection method. Moreover, our CAN framework is able to generate attention parts directly from raw person image pairs in an end-to-end way.
- Finally, we quantitatively validate the good performance of our end-to-end CAN framework by comparing it to the state-of-the-art performance on three benchmark datasets: CUHK01 [9], CUHK03 [13] and Market-1501 [15].

The paper is organized as follows. Sec. II reviews the related work briefly. In Sec. III, the framework is described in details. Then, the experimental results on several public benchmark datasets are shown and the analyses are given in Sec. IV. Finally, a conclusion is presented in Sec. V.
In this paper, we propose an end-to-end Comparative Attention Network (CAN) based architecture that formulates the problem of person re-identification as discriminative visual attention finding and ranking optimization. Fig. 2 illustrates our network architecture (III-A). For a given triplet of person images, we first apply CNN at each one to learn and extract features. Then the features are passed to the LSTM-based comparative attention components (III-C) to obtain the discriminative attention masked features at different time steps. To combine these different time-step features and make them more discriminative, a triplet selection method (III-D) is utilized after concatenating different time-step features. Each of these components is explained in the following subsections.

### III. Model Architecture

In this paper, we propose an end-to-end Comparative Attention Network (CAN) based architecture that formulates the problem of person re-identification as discriminative visual attention finding and ranking optimization. Fig. 2 illustrates our network architecture (III-A). For a given triplet of person images, we first apply CNN at each one to learn and extract features. Then the features are passed to the LSTM-based comparative attention components (III-C) to obtain the discriminative attention masked features at different time steps. To combine these different time-step features and make them more discriminative, a triplet selection method (III-D) is utilized after concatenating different time-step features. Each of these components is explained in the following subsections.

#### A. End-to-End Comparative Attention Network Architecture

Fig. 2 illustrates the architecture of the proposed end-to-end Comparative Attention Network (CAN). The CAN network can localize and compare multiple person parts using the comparative attention mechanism. In this section, we describe how our comparative attention network works in the training phase and the test phase individually.

1) **Training Phase:** During training, the model starts from processing a triplet of images. Here, we denote the images of a triplet as \( I, I^+, I^- \), corresponding to the anchor sample, the positive sample and the negative sample respectively. \( I \) and \( I^+ \) come from the same class (positive pair), while \( I^- \) is from a different class (negative pair). The objective of CAN is to learn effective feature representation and to generate discriminative visual attention regions. Thus, in terms of the features extracted from the attention regions, the truly matched images are closer than the mismatched images by training the model on a set of triplets \( \langle I, I^+, I^- \rangle \). Fig. 2(a) shows the overall architecture used for training which consists of following two parts: feature learning and comparative attention.
comparative attention model to generate the comparative visual attention regional features, which are denoted as $H = \beta(X)$. Here, $\beta$ denotes the comparative attention generation part of our model, and $H$ correspond to local comparative attention features of persons. Note that all the person samples in a triplet share the same parameters in feature learning and comparison, as shown in Fig. 3a. Details of the comparative attention model will be given in Section III-B and III-C.

As mentioned above, our goal is to generate discriminative feature representation and visual attention regions through comparing the similarity and difference of positive and negative pairs in each triplet. Therefore, we adopt the triplet loss [25] as the final loss function. Within a triplet of $\langle H_1, H_1^+, H_1^- \rangle$, we expect features of the positive sample $H^+$ is more similar to $H_i$ than the features of the negative sample:

$$\|H_i - H_1^+\|^2_2 + \alpha < \|H_i - H_1^-\|^2_2,$$

(1)

Here $\alpha$ is a margin that is introduced to enhance the discriminative ability of learned features between positive and negative pairs. Therefore, for $N$ triplets, the loss function that CAN is going to minimize is:

$$L = \frac{1}{N} \sum_i \left[ \|H_i - H_1^+\|^2_2 - \|H_i - H_1^-\|^2_2 + \alpha \right]_+,$$

where $[\cdot]_+$ truncates the involved variable at zero. Next, we proceed to introduce how to apply the network trained for testing.

2) Test Phase: As shown in Fig. 2b, it also has two parts in the testing architecture, features extraction and comparative attention generation. After passing a set of person image pairs in the testing set into the trained CAN, the distances of them are computed. Then the ranking unit directly outputs the final ranking results. Here, we adopt average CMC (Cumulative Matching Characteristics) [26] and the accuracy at top ranks as the evaluation metrics, as in [1]–[17]. The detailed definition of CMC will be given in the Section IV-A. It is worth to mention that we also examine these items to see the performance of the whole network on the validation dataset during the training phase. This is because the training loss can only reflect the tendency of performance variance on the training set while the output evaluations on the validation set can directly indicate the true ranking performance. That is, we can train the network through directly optimizing the ranking results on the validation set.

B. Long Short-Term Memory Networks

In our CAN framework, we use a long short-term memory (LSTM) network to produce a local attention region at every time step conditioned on $D$ CNN feature maps, the previous hidden state and the previous generated attention location map. The size of the feature map is $K \times K \times D$. We use the LSTM implementation, shown in Fig. 3, introduced in [27], [19] and [21]. The formulations are shown as follows:

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t,$$

$$h_t = o_t \odot \tanh(c_t),$$

where $i_t, f_t, c_t, o_t$ and $h_t$ are the input gate, forget gate, cell state, output gate and hidden state respectively. In addition, $x_t$ represents the input feature to the LSTM at time-step $t$ and $M$ is an affine transformation containing a set of learnable weights parameters.

At each time-step $t$, the LSTM of our model predicts $l_{t+1}$, which denotes the softmax locations of size $K \times K$ and is defined as follows:

$$l_{t,i} = \frac{\exp(W_{i,h}^T h_{t-1})}{\sum_{j=1}^{K^2} \exp(W_{j,h}^T h_{t-1})}, \quad i = 1, \ldots, K^2,$$

where $W_{i,h}$ denotes the weight parameters to generate softmax locations. Then, the aggregated (via pooling) feature can be computed as follows:

$$x_t = \mathbb{E}_{p(l_t|h_{t-1})}[X_t] = \sum_{i=1}^{K^2} l_{t-1,i} X_{t,i},$$

where $X_t$ and $X_{t,i}$ correspond to the feature maps and the $i^{th}$ slice of the feature cube at time-step $t$. We adopt the similar initialization method for memory state and hidden state used
in [21]:

\[
c_0 = f_{\text{init},c} \left( \frac{1}{T} \sum_{t=1}^{T} \left( \frac{1}{K^2} \sum_{t=1}^{K^2} X_{t,i} \right) \right),
\]

\[
h_0 = f_{\text{init},h} \left( \frac{1}{T} \sum_{t=1}^{T} \left( \frac{1}{K^2} \sum_{t=1}^{K^2} X_{t,i} \right) \right),
\]

where \( f_{\text{init},c} \) and \( f_{\text{init},h} \) are two multilayer perceptrons and \( T \) is the number of time-steps in the model. These values are used to calculate the initial softmax attention location \( I_1 \) which is applied on CNN features \( X_i \) to get the initial input \( x_1 \) as shown in Fig. 3 and Fig. 4.

![Diagram of comparative attention component](image)

Fig. 4. The comparative attention component composed of several time-step LSTMs takes CNN feature maps of a same person image as its input at each time step and outputs the concatenated hidden states, which are utilized every several time-steps, as the features sent to the triplet loss layer. Here, we simply show one attention component, but there still exist another two weight-sharing attention components working simultaneously to compare positive and negative pairs in triplets and generate the comparative attention maps in training phase.

C. The Comparative Attention Model

We extract the CNN features as the inputs of our LSTM-based comparative attention components. Generally, CNN is used to obtain the global discriminative features from raw person images while the comparative attention components generate more local attention regions through comparing different person images. Intuitively, this mechanism has somewhat similarity with the function of the human visual system. In this subsection, we look inside the comparative attention components. The detailed composition shown in Fig. 3 corresponds to the attention components in the training phase and the test phase in Fig. 2. The brief working process of the comparative attention components is illustrated in Fig. 4. Note that although we simply show one attention component here, there exist three weight-sharing attention components working simultaneously in the training phase, among which two work in the test phase in practice, as shown in Fig. 2. At each time-step, the attention component takes CNN features extracted from a triplet of raw person images as its inputs. Each triplet of person images is “seen” once by our attention component in one glimpse.

Then some information is “remembered” and some “forgotten”, decided by the LSTM unit, in order to generate attention location maps and hidden states for the next time-step or “glimpse”. For the hidden states output by each time-step of comparative attention components, they contain the memory in the process of comparing person images, and are exploited to obtain the local attention maps, as introduced in III-B. Therefore, to combine all the generated attention parts information and utilize it as holistic discriminative features for comparison, a concatenation layer is applied to concatenate a few time-steps of hidden states. Our CAN framework employs the triplet loss function and the whole framework is a three-weights-sharing-branch as well as recurrent network, so it does not converge quickly. To make the network converge faster, the concatenated features are passed to the \( \ell_2 \)-normalization layer as the output of attention components:

\[
y = \frac{c}{\sqrt{\sum_{q=1}^{r} c_q^2}},
\]

where \( c = [c_1, c_2, ..., c_q] \) is the output of the concatenation layer with dimension \( r \). The \( \ell_2 \)-normalization layer ensures that the distance computed from each triplet can not exceed the margin \( \alpha \) described in Eqn. (1). It is obvious that the upper bound of the distance between two \( \ell_2 \)-normalized concatenated features of two person images is two. Therefore, more triplet constraints can take effect for the loss function, which accelerates convergence speed of the end-to-end CAN framework training.

D. Triplet Selection

It is crucial to select triplets that violate the constraint given in Eqn. (1). In particular, given \( H_i \), we want to select a positive sample \( H_i^+ \) satisfying \( \arg\max_{H_i^+} \| H_i - H_i^+ \|_2^2 \) while a negative sample satisfying \( \arg\min_{H_i^-} \| H_i - H_i^- \|_2^2 \). However, it is difficult and unrealistic to compute the argmin and argmax for the whole training set. Furthermore, our model needs to compare pair-wise images and to generate a series of attention locations for every image of each person. Therefore, it requires an efficient way to compute the argmin and argmax. There are two methods to be chosen as mentioned in [25]:

- Off-line triplets selection. The triplets are generated every few steps, and the most recent network checkpoint is employed to compute the argmin and argmax.
- On-line triplets selection. The selection can be done within a mini-batch.

Obviously, generating all possible triplets would result in overwhelming many triplets that are feasible for the constraint in Eqn. (1). But some of these triplets would not contribute to the training and slow down the convergence of model training.
Besides, they would still be passed through the network, which cause large unnecessary resource consumption. Different from off-line triplets selection method, on-line triplets selection approach selects triplets that are active and can contribute to improving the model within a mini-batch, so it is of higher efficiency and lower resource consumption. Therefore, we adopt the on-line triplets selection method in this paper. Specifically, instead of picking the hard positive, we adopt all positive pairs and randomly sample negative samples added to each mini-batch. In practice, we find that using all positive pairs makes the model more stable and converge faster than selectively using hard positive pairs in a mini-batch.

IV. EXPERIMENTS

A. Datasets and Evaluation Protocol

There exist several challenging benchmark data sets for person re-identification. In this paper, we use CUHK01 [9], CUHK03 [13] and Market-1501 [15], which are three largest benchmarks available, to conduct experiments. In experiments, for each pedestrian, the matching of his or her probe image (captured by one camera) with the gallery images (captured by another camera) is ranked. To reflect the statistics of the ranks of true matches, the Cumulative Match Characteristic (CMC) curve is adopted as the evaluation metric. Specifically, to create a CMC curve, the L2-norm distances between probe samples and those of gallery samples are computed firstly. Secondly, for each sample, a rank order of all the samples in the gallery is sorted from the sample with the smallest distance to the biggest distance. In the end, the percentage of true matches founded among the first \( m \) ranked samples is computed and denoted as \( \text{rank}(m) \). Note, all the CMC curves are computed with single-shot setting. And for the Market-1501 dataset, mean average precision (mAP) as in [15] is also employed to evaluate the performance since there are on average 14.8 cross-camera ground truth matches for each query.

1) CUHK01: The CUHK01 [9] dataset contains 971 persons captured from two camera views, and each of them has two images in each camera view. Camera A captures the individuals in frontal or back views while camera B captures them in side views. All the images are normalized to the size of \( 160 \times 60 \) pixels. We randomly divide the data set into a training set of 871 people and a test set of 100 people. This procedure is repeated 10 times and the average of CMC curves is reported.

2) CUHK03: There are 13,164 images of 1,360 identities contained in the CUHK03 dataset [13]. All pedestrians are captured by six cameras, and each person is only taken from two camera views. It consists of manually cropped person images and images automatically detected by the Deformable-Part-Model (DPM) detector [28]. This is a more realistic setting considering the existence of misalignment, occlusions, body part missing and detector errors. We evaluate the performance of CAN with the similar setting of [13]. That is, the dataset is randomly partitioned into two parts: 1,160 persons for training and 100 persons for testing. This random partition is repeated for 20 times for computing averaged performance.

3) Market-1501: Market-1501 [15] is currently the largest public available re-identification dataset, containing 32,668 detected bounding boxes of 1,501 persons, with each of them captured by six cameras at most and two cameras at least. Similar to the CUHK01 dataset, it also employs DPM detector [28]. We use the provided fixed training and test set, containing 750 and 751 identities respectively, to conduct experiments.

B. Parameters Setting

We implement our network using Caffe [29] deep learning framework. The training of the network converges in roughly 8-10 hours on NVIDIA GeForce GTX TITAN X GPU. In all of experiments, the dimensionality of the LSTM hidden state, the cell state, and the hidden layer are set to 512 for CUHK01, CUHK03 and Market-1501. All the images are resized to \( 227 \times 227 \) to train our model. As mentioned above, we first use the softmax regression network to pre-train the truncated CNN. At this stage, we perform stochastic gradient descent [30] to update the weights. We start with a base learning rate of \( \eta(0) = 0.01 \) and gradually decrease it along with the training process using an inverse policy: \( \eta(k) = \eta(0)(1 + \gamma \cdot k)^{-p} \) where \( \gamma = 10^{-4} \), \( p = 0.75 \), and \( k \) is index of the current mini-batch iteration. We use a momentum of \( \mu = 0.9 \) and weight decay \( \lambda = 5 \times 10^{-4} \). After the CNN feature learning network is pre-trained, we use the pre-trained model to initialize our end-to-end Comparative Attention Network (CAN). Here, we use the weight update parameter settings similar to those in the pre-training stage except that the initial learning rate is set to \( \eta(0) = 0.001 \). As mentioned above, we adopt the on-line triplet selection method, so the batchsize is set to 134 determined by cross-validation. We empirically set the value of the margin parameter as \( \alpha = 0.3 \). As mentioned in Sec. III-B, hidden states of LSTM at different time-steps are concatenated as the final features passed to the normalization layer. Thus, we use 8 time-steps and the extracted hidden states of the \( 2^{nd}, 4^{th} \) and \( 8^{th} \) time-step in all experiments. It is illustrated in Fig. 5(d) and is validated in Sec. IV-D.

C. Data Augmentation

In the training set, there exist much more negative pairs than positive pairs, which can lead to data imbalance and overfitting. To overcome this issue, we artificially augment the data by performing random 2D translation, similar to the processing in [13]. For an original image of size \( w \times h \), we sample ten same-sized images around the image center, with translation drawn from a uniform distribution in the range \([-0.05w, 0.05w] \times [-0.05h, 0.05h] \). For CUHK01 dataset, we also horizontally flip each image. In addition, because we use the on-line triplet selection method (see III-D), we randomly shuffle the dataset in terms of their labels. Through this shuffle strategy, more triplets can be produced in a mini-batch. Specifically, we perform this operation ten rounds for each dataset.

D. Analysis of the Proposed Model

In Sec. III, we introduce our model architecture using CNN as the features extractor. Here we compare the performance of
To further demonstrate the effectiveness of the comparative attention components in CAN, we also compare the performance of the proposed CAN with that of a similar architecture with input of each LSTM replaced by the simple average pooling or max pooling output in Fig. 4. In other words, we use the same architecture illustrated in Fig. 2 except that none of the attention mechanisms is contained in the model, and thus there are no softmax location map produced and all locations in a feature map have the same weight. Note that the model used here is also in an end-to-end form. From the results given in Table I, it is obvious that comparing positive pair and negative pair of each person triplet and staying focus on those more discriminative parts or locations can perform better than using the complete feature cube to discern different persons.

In the end, we also conduct a series of experiments to evaluate which time-steps are chosen to be concatenated can achieve the best performance for our model. We use the following three settings: i) all the time-steps; ii) last time-step; iii) 2\textsuperscript{nd}, 4\textsuperscript{th} and 8\textsuperscript{th} time-steps. The experimental results are shown in Fig. 5(a)-(d). We find that using the third setting can make our framework perform best.

### TABLE I

| Method               | Rank1  | Rank5  | Rank10 | Rank20 |
|----------------------|--------|--------|--------|--------|
| CAN using Conv5      | 32.17  | 65.13  | 84.89  | 86.06  |
| CAN using Max5       | 37.25  | 69.14  | 89.68  | 92.17  |
| Avg pooled LSTM using Max5 | 51.29  | 82.32  | 85.06  | 87.45  |
| Max pooled LSTM using Max5 | 50.51  | 81.07  | 83.17  | 86.76  |
| end-to-end CAN using Conv5 | 56.74  | 88.81  | 91.44  | 93.31  |
| end-to-end CAN using Max5 | 65.65  | 91.28  | 96.29  | 98.17  |

### E. Comparison with State-of-the-Art Methods

We compare our model with the following state-of-the-art methods: SDALF [4], LMNN [5], ITML [1], KRMCA [22], LDM [3], eSDC [10], Metric Ensembles (Ensembles) [8], KISSME [7], JointRe-id [16], FPNN [13], PersonNet [17].

1) Results on CUHK01 and CUHK03: These two datasets consist of thousands of training samples. Table 1 and Fig. 5(a) show results on CUHK01. Our method beats all compared methods at low ranks and achieves around 10% improvement in Rank1 recognition rate compared to the second best method (PersonNet [17]), which is significant. As for CUHK03, there are two settings: manually cropped person images and person images produced by DPM detector. Obviously, the performance on the latter one appears lower than that on the former, as shown in Fig. 5(b), Fig. 5(c), Table III and Table IV. However, the images produced by the detector can also reflect the algorithms in the real world. It can be seen
from Fig. [5(b) and Table [III] that, as expected, on this large
dataset, some other deep learning based methods, such as
PersonNet [17], can achieve similar performance with millions
of parameters or become much more competitive. However,
with the detector boxes, our method is less affected, especially
for the Rank1, and outperforms other approaches including
deep learning based ones by a large margin. We suppose that
the performance is not affected too much, possibly because
our model could accurately attend to different discriminative
parts of images and integrate their information which is robust
to the influence brought by the detector.

### Table II

| Method       | Rank1  | Rank5  | Rank10 | Rank20 |
|--------------|--------|--------|--------|--------|
| JointRe-id [16] | 65.00  | 88.70  | 93.12  | 97.20  |
| FPNN [13]    | 27.87  | 58.20  | 73.46  | 86.31  |
| ITML [11]    | 17.10  | 42.31  | 55.07  | 71.65  |
| LMNN [5]     | 21.17  | 49.67  | 62.47  | 78.62  |
| KRMCA [22]   | 31.22  | 57.68  | 73.55  | 86.07  |
| LDM [8]      | 26.45  | 57.69  | 72.04  | 84.69  |
| SDALF [4]    | 9.90   | 41.21  | 56.00  | 66.37  |
| eSDC [10]    | 22.84  | 43.89  | 57.67  | 69.84  |
| KISSME [7]   | 29.40  | 57.67  | 72.43  | 86.07  |
| PersonNet [17]| 71.14  | 90.07  | 95.00  | 98.06  |
| **end-to-end CAN** | **81.04** | **96.89** | **99.67** | **100.00** |

### Table III

| Method       | Rank1  | Rank5  | Rank10 | Rank20 |
|--------------|--------|--------|--------|--------|
| Ensembles [8] | 62.10  | 89.10  | 94.30  | 97.80  |
| JointRe-id [16] | 54.74  | 86.50  | 93.88  | 98.10  |
| FPNN [13]    | 20.65  | 51.50  | 66.50  | 80.00  |
| ITML [11]    | 5.53   | 18.89  | 29.96  | 44.20  |
| LMNN [5]     | 7.29   | 21.00  | 32.06  | 48.94  |
| KRMCA [22]   | 9.23   | 25.73  | 35.09  | 52.96  |
| LDM [8]      | 13.51  | 40.73  | 52.13  | 70.81  |
| SDALF [4]    | 5.60   | 23.45  | 36.09  | 51.96  |
| eSDC [10]    | 8.76   | 24.07  | 38.28  | 53.44  |
| KISSME [7]   | 14.17  | 48.69  | 52.57  | 70.53  |
| PersonNet [17]| 64.80  | 89.40  | 94.92  | 98.20  |
| **end-to-end CAN** | **65.65** | **91.28** | **96.29** | **98.17** |

### Table IV

| Method       | Rank1  | Rank5  | Rank10 | Rank20 |
|--------------|--------|--------|--------|--------|
| JointRe-id [16] | 44.96  | 76.01  | 83.47  | 93.15  |
| FPNN [13]    | 19.89  | 50.00  | 64.00  | 78.50  |
| ITML [11]    | 5.14   | 17.87  | 28.24  | 43.12  |
| LMNN [5]     | 6.25   | 18.68  | 29.07  | 45.03  |
| KRMCA [22]   | 8.14   | 20.31  | 32.96  | 49.96  |
| LDM [8]      | 10.92  | 32.25  | 48.78  | 65.63  |
| SDALF [4]    | 4.87   | 21.17  | 35.06  | 48.44  |
| eSDC [10]    | 7.68   | 21.86  | 34.96  | 50.03  |
| KISSME [7]   | 11.7   | 31.16  | 48.98  | 65.63  |
| **end-to-end CAN** | **63.05** | **82.94** | **88.17** | **93.29** |

### Table V

Comparison of our end-to-end CAN methods performance on the Market-1501 dataset with single query setting to the state-of-the-art models.

| Method       | Rank1  | mAP    |
|--------------|--------|--------|
| SDALF [4]    | 20.53  | 8.2    |
| eSDC [10]    | 33.54  | 13.54  |
| Zheng et al. [15] | 34.4  | 14.09  |
| PersonNet [17]| 37.21  | 18.57  |
| **end-to-end CAN** | **48.24** | **24.43** |

F. Visualization of Attention Maps and Discussions

In Fig. [6(a)], we visualize some comparative attention maps produced by our network for testing samples from CUHK01 dataset, which are all ranked at top 1 in re-identification results. We can see that the model is able to focus on different parts of the person images at different time-steps. Taking person 1 and person 2 in Fig. [6] as examples, the attention is focused on the lower part of the body at several first time-steps for person 1 in both camera A and camera B while on the whole body for person 2. In other words, our model is able to focus on the same parts of positive pairs, different parts of negative pairs otherwise. Note that, the model does not always attend to the foreground. We can see that some background is attended to, which means the background can also provide information to assist matching persons correctly.

In Fig. [6(b), we also visualize some failure cases of our proposed CAN model in terms of the generated attention maps based on comparison mechanism. As an example, for the person 1 in the figure, our model fails to focus on the same parts of positive person image pairs. This is partially because there are more than one persons in the image captured by camera B and this person is occluded heavily by other persons in the view of camera B. This extremely hard scenario presents challenges to our CAN model as it can not exactly
compare person image pairs and decide which local region should be selected. This phenomenon can also be observed from the generated attention maps of person 2 and person 3 in Fig. 6(b).

V. CONCLUSION

In this work, we introduced a novel visual attention model that is formulated as a triplet recurrent neural network which takes several glimpses of triplet images of persons and dynamically generates comparative attention location maps for person re-identification. We conducted extensive experiments on three public available person re-identification datasets to validate our method. Experimental results demonstrated that our model outperforms other state-of-the-art methods in most cases, and verified that our comparative attention model is beneficial for the recognition accuracy in person matching.

REFERENCES

[1] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, “Information-theoretic metric learning,” in Proceedings of the 24th international conference on Machine learning. ACM, 2007, pp. 209–216.
[2] D. Gray and H. Tao, “Viewpoint invariant pedestrian recognition with an ensemble of localized features,” in Computer Vision–ECCV 2008. Springer, 2008, pp. 262–275.
[3] M. Guillaumin, J. Verbeek, and C. Schmid, “Is that you? metric learning approaches for face identification,” in Computer Vision, 2009 IEEE 12th international conference on. IEEE, 2009, pp. 498–505.
[4] M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani, “Person re-identification by symmetry-driven accumulation of local features,” in Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, 2010, pp. 2360–2367.
[5] M. Hirzer, P. M. Roth, and H. Bischof, “Person re-identification by efficient impostor-based metric learning,” in Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on. IEEE, 2012, pp. 203–208.
[6] A. Mignon and F. Jurie, “Pcca: A new approach for distance learning from sparse pairwise constraints,” in Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012, pp. 2666–2672.
[7] M. Koestinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof, “Large scale metric learning from equivalence constraints,” in Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012, pp. 2288–2295.

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

[9] W. Li and X. Wang, “Locally aligned feature transforms across views,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3594–3601.

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

[11] ——, “Person re-identification by salience matching,” in Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 2528–2535.

[12] ——, “Learning mid-level filters for person re-identification,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 152–159.

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

[14] D. Yi, Z. Lei, S. Liao, and S. Z. Li, “Deep metric learning for person re-identification,” in 2014 22nd International Conference on Pattern Recognition (ICPR). IEEE, 2014, pp. 34–39.

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

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

[17] L. Wu, C. Shen, and A. v. d. Hengel, “Personnet: Person re-identification with deep convolutional neural networks,” arXiv preprint arXiv:1601.07255, 2016.

[18] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[19] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” in Proceedings of The 32nd International Conference on Machine Learning, 2015, pp. 2048–2057.

[20] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” arXiv preprint arXiv:1409.0473, 2014.

[21] S. Sharma, R. Kiros, and R. Salakhutdinov, “Action recognition using visual attention,” arXiv preprint arXiv:1511.04119, 2015.

[22] H. Liu, M. Qi, and J. Jiang, “Kernelized relaxed margin components analysis for person re-identification,” Signal Processing Letters, IEEE, vol. 22, no. 7, pp. 910–914, 2015.

[23] A. Graves, N. Jaitly, and A.-r. Mohamed, “Hybrid speech recognition with deep bidirectional lstm,” in Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on. IEEE, 2013, pp. 273–278.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in neural information processing systems, 2012, pp. 1097–1105.

[25] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 815–823.

[26] D. Gray, S. Brennan, and H. Tao, “Evaluating appearance models for recognition, reacquisition, and tracking,” in Proc. IEEE International Workshop on Performance Evaluation for Tracking and Surveillance (PETS), vol. 3, no. 5. Citeseer, 2007.

[27] W. Zarembo, I. Sutskever, and O. Vinyals, “Recurrent neural network regularization,” arXiv preprint arXiv:1409.2329, 2014.

[28] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 32, no. 9, pp. 1627–1645, 2010.

[29] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” arXiv preprint arXiv:1408.5093, 2014.

[30] L. Bottou, “Stochastic gradient tricks,” Neural Networks, Tricks of the Trade, Reloaded, pp. 430–445, 2012.