ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-355

Person Re-Identification via Multiple Coarse-to-Fine Deep Metrics

Mingfu Xiong[1,3] and **Jun Chen**[1,2] and **Zheng Wang**[1,3] and **Zhongyuan Wang**[2,3] and **Ruimin Hu**[1,2,3] and **Chao Liang**[1,2] and **Daming Shi**[4]

Abstract. Person re-identification, aiming to identify images of the same person from various cameras views in different places, has attracted a lot of research interests in the field of artificial intelligence and multimedia. As one of its popular research directions, the metric learning method plays an important role for seeking a proper metric space to generate accurate feature comparison. However, the existing metric learning methods mainly aim to learn an optimal distance metric function through a single metric, making them difficult to consider multiple similar relationships between the samples. To solve this problem, this paper proposes a coarse-to-fine deep metric learning method equipped with multiple different Stacked Auto-Encoder (SAE) networks and classification networks. In the perspective of the human's visual mechanism, the multiple different levels of deep neural networks simulate the information processing of the brain's visual system, which employs different patterns to recognize the character of objects. In addition, a weighted assignment mechanism is presented to handle the different measure manners for final recognition accuracy. The experimental results conducted on two public datasets, i.e., VIPeR and CUHK have shown the prospective performance of the proposed method.

1 Introduction

Person re-identification aims to judge whether two persons which come from different cameras views belong to the same person. Owing to its significance in tracking the escape route of suspects and daily life, it has been widely used in the criminal investigation and artificial intelligence [2]. Over the past decade, a large number of person re-identification methods have been proposed in the literatures [20, 1, 25, 23, 26, 19, 22, 16] and most of them have achieved satisfying performance. However, it is still a challenging problem because of various surveillance conditions, such as, view switching, lighting variations and image scaling (see Figure 1). Previous research on person re-identification can be generally classified into two categories: feature representation [25, 23, 13] and metric learning [26, 20, 9]. Since lighting and view changes can cause significant appearance variations, designing a set of discriminative and robust features is still a challenging problem [26, 21]. In order to boost the performance of person re-identification, increasing number of researches are devoted to learn a proper distance function to compare two person image features [21, 11].

Figure 1. The examples of aspects changes caused by different views, lighting conditions, scaling variations from public datasets CUHK [14], VIPeR [8], respectively. Each column shows two images of the same person from two different cameras.

Most of the existing work focus on either feature representation or metric learning step, lacking a global consideration of above two steps. It is crucial to build an automatic connection among these components in the training process for the overall system performance. More recently, deep learning, which is based on an end-to-end network, has been presented to solve the problem in a unified framework. It has attracted a lot of research interests for its superb performance in person re-identification and other visual tasks [11, 10].

Generally speaking, deep learning aims to learn features and metrics in a unified hierarchical framework directly from raw data. It has also been used in metric learning [11, 24, 1]. Unlike most previous metric learning methods which usually seek a linear distance to project samples into another linear space, the deep metric learning methods try to compute the similarities of samples via multiple layer nonlinear transformations. However, most of them just try to seek a simplex manner to measure the similarities of the persons. Such a simplicity of the metric manner may cause the problem that other similarities relationship of samples cannot be well exploited. Figure 2 is a particular example, where the persons in the two pictures are similar in shape contour and clothes, but they are not the same one in fact. Therefore, the single metric learning methods may lose helpful discriminative information for similarity comparison.

To relieve the problems with these limitations, we propose a method with multiple coarse-to-fine SAE models (SAE networks and classification networks) for deep metric learning. In our algorithm, there exist several SAEs neural networks with different hidden layers for multi-scale metric learning and the similarities of multiple

[1] State Key Laboratory of Software Engineering, Wuhan University, China. email: {xmf2013, chenj, wangzwhu, hrm, cliang}@whu.edu.cn, wzy_hope@163.com
[2] Collaborative Innovation Center of Geospatial Technology, China.
[3] National Engineering Research Center for Multimedia Software, Computer School of Wuhan University, China.
[4] School of Science and Technology, Middlesex University London, London NW4 4BT, United Kingdom. email: d.shi@mdx.ac.uk

Figure 2. Examples of dissimilar pairs. From this figure, we can see that the persons in the same column are similar in color and contour. In fact, they are not the same person. So the important information that judges whether the samples belong to the same object may be lost via a single metric manner.

levels for person image pairs are obtained via different deep neural networks in a coarse-to-fine manner. Generally speaking, we judge two persons which are the same one or not just via the physical characteristic at first glance. Then the facial features and clothes can be compared. At last, more details will be observed for final validation. This process is the information handling of our visual system. In our work, there are different deep neural networks for metric learning and it includes many neural nodes for each network which simulates the neuron of brain. There are fewer nodes in shallow neural network and vice versa. These architectures are similar to the structure of the brain in the view of bionics. In this way, we can simulate the information processing of the brain's visual system, which employs multiple different levels to recognize the character of objects. Besides, a weighted assignment mechanism is presented to handle these results which are from different SAEs networks.

The contribution of this paper can be summarized into two aspects:

Firstly, we propose a framework of multiple different SAEs networks and classification networks for metric learning to measure the similarities of the samples from coarse-to-fine.

Secondly, a weighted assignment mechanism is presented for integrating the results that come from previous different deep neural networks. The information processing mechanism of brain is simulated via this coarse-to-fine manner. Experimental results validate the effectiveness on two public person re-identification datasets.

2 Our Approach

2.1 Preliminaries

2.1.1 Person Re-identification Problem

As mentioned above, the purpose of person re-identification is to match the pedestrians observed in non-overlapping cameras via various visual methods. In other way, this problem can also be seen as a binary classification problem. For the convenience of following discussion, in our work, we consider a pair of cameras which are denoted as C_a and C_b, respectively. The persons in each camera are expressed as $\{p_a = p_a^1, p_a^2, ..., p_a^n\}$ and $\{p_b = p_b^1, p_b^2, ..., p_b^m\}$. The *n* and *m* denote the numbers of the person in each camera view. Let the label y = 1 if two pedestrian images (p_a^i, p_b^i) are matched, and y=0, otherwise. So a pair of person image is the object that we should consider in this paper. P_{ab}^I is the combination of two persons that from different cameras views, respectively.

2.1.2 The Basic Auto-Encoders

We recall the basic principles of the auto-encoder models, e.g, [3]. The classical auto-encoder tries to learn a function $h_{W,b}(x) \approx x$. In other words, the algorithm is trying to learn an approximation to the identify function, so as to the output $\widehat{x}$ that is similar to the input x. It is divided into two processes, that is "encoding" and "decoding". In the former, it is using a deterministic function of $h = f_\theta = \sigma(Wx + b)$ with parameters $\theta = \{W, b\}$. And in the process of decoding, it is used to reconstruct the input by a reverse mapping of f: $h^{'} = f_{\theta'} = \sigma(W^{'}h + b^{'})$ with $\theta^{'} = \{W^{'}, b^{'}\}$. The two parameter sets are usually constrained to be of the form $W^{'} = W^T$, using the same weights for encoding the input and the latent representation y_i. A common method to train the model is the famous Back-Propagation Algorithm [18]. The cost function is described as below. For example, we just have a set of training samples: $\{(x^{(1)}, y^{(1)}), ..., (x^{(m)}, y^{(m)})\}$. And the cost function for one training case is described as following.

$$J(W, b; x, y) = \frac{1}{2}||h_{W,b}(x) - y||^2 \quad (1)$$

In addition, cost function for the whole training set is described as formula (2).

$$J(W, b) = \frac{1}{m}\sum_{i=1}^{m} J(W, b; x, y) + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_l+1}(W_{ij}^{(l)})^2 \quad (2)$$

In order to train the model, we just need to minimize $J(W, b)$. The process of training is not belong to this range.

2.2 Multiple Coarse-to-Fine Deep Metric Learning

The architecture of the multi-scale learning method is shown in Figure 3. It includes four layers to get the coarse-to-fine deep metric learning for person re-identification. The first layer is the monitored person images that come from two different camera views. We randomly combine the two person images together to form the original input for the second layer. Then the pretreatment is executed via subtracting the mean values and normalization for each sample pair. The images are transformed into gray images and the input of the SAE networks is formed. Then a softmax classifier is followed by each of the stacked auto-encoder to get a classification result. At last, we have utilized a weighted assignment mechanism to handle the classification results obtained from the former layer. And the multiple deep metric learning framework includes two networks: the SAEs network and classification network. The detail of each layer is described as below.

There are several different SAE models for metric learning in our algorithm. For each auto-encoder network, it has three layers: the input layer, hidden layer and the output layer. In many previous work, the auto-encoder networks were used for feature representation [17]. In this work, we have used it for metric learning. In details, each of the SAE network is following by a softmax classifier (See Figure 3). Each of them is trained via the back-propagation algorithm.

From Figure 3. we can see that the input of auto-encoder networks is the person image pairs, which is reprocessed before being input into the deep neural networks. The network parameters are trained from the first hidden layer. And the output of the first hidden layer is calculated via the parameters that were trained before. The output for the next hidden layers is counted through the same way. After that, the last hidden layer is followed by a softmax classifier. The output of the last hidden layer is used to train parameters for the

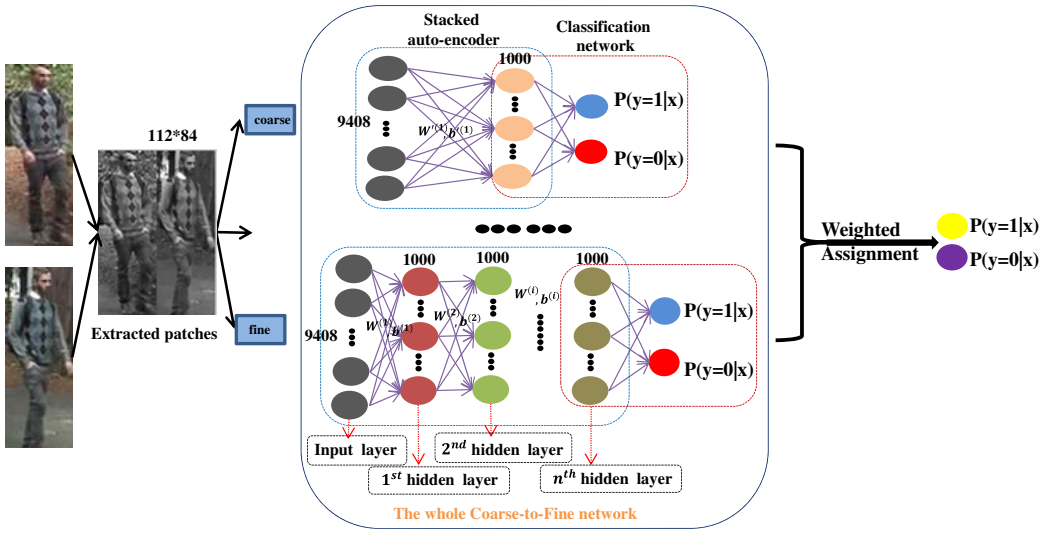

Figure 3. The Multiple Coarse-to-Fine Deep Metric Learning Framework. From left to right, there are four layers structured for last classification. For the first layer, we randomly select the two person images which come from different camera views. And the second, the combined image pairs are obtained from the previous layer that transformed into gray images. Then they are subtracted to the mean values and normalized into [0,1]. The coarse-to-fine SAEs structure is following in the third layer. This layer includes several SAEs which equipped with a softmax classifier. In other word, The whole network is composed of SAEs networks and classification networks. The parameter θ={W, b}(W is the weight, and b is bias.) is to be trained. The output of the softmax classifier is the probability that the sample pair belongs to a certain class. And the weighted assignment mechanism is used for handle the classification results for the last layer.

softmax classifier. The cost function for the softmax classifier is described as formula (3). For example, considering the training set is $\{(x^{(1)}, y^{(2)}), ..., (x^{(m)}, y^{(m)})\}$. The m denotes the numbers of samples and $x^{(i)}$ represents the feature that is the output of the last hidden layer in this work. The y^i is the classification label for each sample.

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m}\sum_{j=1}^{k} 1\{y^{(i)} = j\} \log \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}}] \quad (3)$$

In order to train the model, we just need to calculate the $J(\theta)$. The gradient decent method is used in the algorithm. And k is 2 for the person re-identification problem. The last result is classified into two classes.

In our model, there are several kinds of SAE networks and each of them has different configurations. So we can capture different levels metric results for each sample pair. A coarse metric learning is implemented via the shallow network and vice versa. So the coarse-to-fine metric manner is formed in this way. In our work, a pair of person images is generated to form the input of the SAE networks. But the final output is the probability that a person belongs to a class. Therefore, we can get multiple different results. These classified results are the sources that we can obtain the final recognition accuracy. And then handling these classified results is described as the following section.

2.3 Joint Learning for Weighted Assignment

As mentioned above, there are several kinds of SAE models for the persons binary classification. The output of each softmax classifier is a probability that the person pair belongs to a certain class. And the probability values that we can obtain are diversified. How to handle these results is remained to settle. In this work, we have utilized the weighted assignment mechanism to solve this problem. In our work, there are several SAE models for metric learning and the multiple similarities are generated via these networks. As the characteristic capability of each metric is different. So the weighted assignment mechanism is presented to get the final result. And the process of jointing is described in Figure 4.

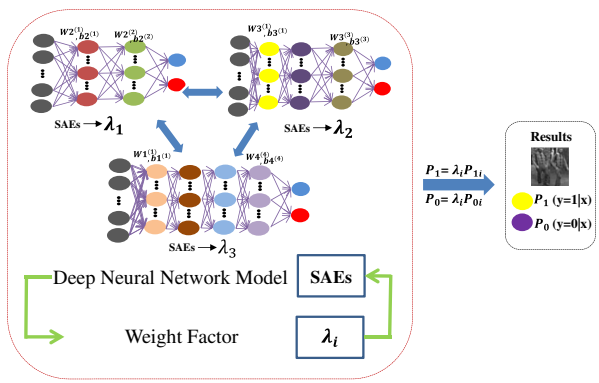

Figure 4. The process of the weights assignment mechanism. In our work, there are three kinds of SAE models for pedestrians classification. The weight factor λ_i is made via joining the deep networks. This process includes three parts: the first one is the process of these SAE models joint learning for three kinds of networks. The parameter θ={W,b} should be trained. Then the weight factor is assigned for each SAE model. At last, the final result is gotten via the operation of weight assignment.

In details, the output of the softmax classifier is two classes represented by the probabilities. From Figure 3, we assume that there are three kinds of SAE models to classify for the person pair. The notations $P(y = 1|x)$ and $P(y = 0|x)$ denote the probabilities that the sample pair x belongs to the certain class. If the person pair is matched, the label $y = 1$, and $y = 0$ otherwise. Generally speaking, the two samples are similar, the other aspects of them are similar

too. We try to consider multiple aspects of the samples for judging whether they belong to the same object. For the person images pair (p_a^i, p_b^i), the probability they matched is P_{ab}^i (P_{ab}^i is from the i-th SAE model. See Figure 3.). The weights assignment mechanism for each probability matrix distribution is represented as formula (4). After that, we reset the probabilities to get the last recognition result.

Algorithm 1 Weighted Assignment For Similar Probability

Input: N labeled a set of training samples (P_{ab}^i, y_i) where P_{ab}^i is the combination of two pedestrian images and $y_i \in \{1, -1\}$ denotes whether the two person belong to the same one.
A distribution over all the training samples: $D_1(i) = 1/N$ for i=1,...,N.

for t=1,...,K:
- Find the best localized feature λ_t for the current distribution D_t.
- Calculate the edge γ_t
$\gamma_t = \sum_{i=1}^{N} D_t(i)h(x_i)y_i$
- If $\gamma_t < 0$ break
- Set $\alpha_t = \frac{1}{2} ln \frac{1+\gamma_t}{1-\gamma_t}$
- Set $D_{t+1}(i) = \frac{1}{Z_t} D_{t+1} exp(-\alpha_t h(x_i)y_i)$, where z_t is a normalizing factor
- Add α_t, λ_t to the joint.

Output: The weight factor λ.

We assign different weights to each of the classification result to get a better representation. Generally, for the task of deep neural networks, more hidden layers lead to greater weights as well as better robust results. And the weights will be higher. In fact, these weights are learnt like the process in [6]: AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers. Our approach is quite similar in these respects, however our object is domain specific (i.e. only applicable to comparing which class the pedestrian belongs to). The proposed probability assignment is a weighted ensemble of likelihood ratio tests, made by the Algorithm 1, a brief review of which that can be found below.

In training the weights are iteratively updated. The training set is $T = \{P_{ab}^1, P_{ab}^2, ..., P_{ab}^N\}$. Initializing the weight distribution of training data is representing like formula (4). And $D_t(x)$ is the probability matrix which means the distribution for the combination of two persons. λ_t is the weight factor.

$$D_1 = (\lambda_{1,1}, .., \lambda_{1,i}, .., \lambda_{1,N}) \quad s.t. \quad \lambda_1 = \frac{1}{N}, \quad i = 1, 2, ..., N. \tag{4}$$

In algorithm 1, the $h(x_i)$ is week classifier, $x \rightarrow \{-1, +1\}$. The error of the update is γ_t and it generates in formula (5).

$$\gamma_t = P((x_i) \neq y_i) = \sum_{k}^{N} \lambda_{ti} I(h(x_i) \neq y_i) \tag{5}$$

The coefficient of $h(x_i)$ is calculated in formula (6)

$$\alpha_t = \frac{1}{2} ln \frac{1+\gamma_t}{1-\gamma_t} \tag{6}$$

The update of weight of probability distribution is representing in formula (7) (8).(t denotes the numbers of iteration.)

$$D_{t+1} = (\lambda_{t+1,1}, \lambda_{t+1,2}, ..., \lambda_{t+1,N}) \tag{7}$$

$$\lambda_{t+1,i} = \frac{\lambda_{ti}}{Z_t} exp(-\alpha_t h(x_i)y_i) \tag{8}$$

Z_t is a normalized factor and represents in formula (9). It makes the D_{t+1} become a probability distribution.

$$Z_m = \sum_{1}^{N} \lambda_{ti} exp(-\alpha_t y_i h(x_i)) \tag{9}$$

In algorithm 1, there are several iterations for the joint learning. The similarity probability is searched for the best weights w.r.t the current distribution and made the joint. And the weight for each probability matrix (i.e. the output of each SAE model) is assigned as formula (10). N denotes the numbers of the SAE model. It is 3 in our work.

$$f(x) = \sum_{t=1}^{N} \lambda_t D_t(x) \tag{10}$$

3 Experimental Results

In this section, we evaluate our multiple coarse-to-fine deep metric learning algorithm on two person re-identification benchmarks: the VIPeR and CUHK. For each dataset, we would give out the experiment result and compare with other previous methods. The detailed experimentation is described as following. We introduce the coarse-to-fine metric learning method from three aspects. i.e. the physical character, profile feature and facial feature. So there are three SAE models simulating the human's visual system. Besides, we implement our algorithm using the Andrew Ng's deep learning framework and write the code for our own architecture. It is time-consuming for roughly 6 days for the deepest network on high-performance computing platform.

3.1 The Datasets

The widely used VIPeR dataset is collected by Gray and Tao [8] and contains 1264 outdoor images obtained from two views of 632 persons. Each pair is made up of images of the same person from two different cameras, under different viewpoints, poses and light conditions, respectively. All images are normalized to 128×48 pixels. Views changes are the matched image pairs containing a viewpoint change of 90 degree.

The CUHK02 Campus dataset [14] contains 1816 persons and five pairs of camera views (P1-P5, ten camera views). They have 971, 306, 107, 193 and 239 persons respectively. Each person has two images in each camera view. This dataset is used to evaluate the performance when camera views in test are different than those in training. In our experiment, we choose view pair P1 for evaluation. And this view includes 971 subjects, 1942 images. Each subjects has two images from 2 camera views. And the instances in the two datasets could be seen as Figures.5.

3.2 Experimental Methods

Evaluation Protocol. Re-identification models are commonly evaluated by the cumulative match characteristic (CMC) curve. This measure indicates how the matching performance of the algorithm improves as the number of returned image increases. Given an algorithm and a test set of images of people with labels, each image in the test set is compared against the remaining images under the given algorithmic model and the position of the correct match is recorded.

Figure 5. Some typical samples of the two public dataset. And each column shows two images of the same person from two different cameras with significant changes on view point and illumination condition. (a) VIPeR dataset contains significant difference between different views. (b) CUHK is similar to VIPeR, but more challenge as it contains more person pairs.

The CMC curve indicates for each rank the fraction of test samples which had that rank or better. A perfect CMC curve would reach the value 1 for rank 1. Specifically, let $P = \{p_1, ..., p_{|P|}\}$ be a probe set, where $|P|$ is the size of P. And $G = \{g_1, ..., g_n\}$ a gallery set. For each probe images $p_i \in P$, all gallery images $g_i \in G$ are ranked by comparing the distance between p_i and g_i in ascending order. The image of the same person p_i in the gallery set is denoted as g_{p_i}. And the index of which in the sorted gallery is denoted as $r(g_{p_i})$. The CMC value of rank k is defined as formula (5)

$$CMC_k = \frac{\sum_{i=1}^{|P|} 1(r(g_{p_i}) \leq k)}{|P|} \tag{11}$$

where $1(\bullet)$ is the indicator function.

Data Augmentation. In the training set, the matched sample pairs (positive samples) are several orders fewer than non-matched pairs (negative samples). If they are directly used to train the deep network, the model tends to predict all the inputs as being non-matched. The easiest and most common method to solve this problem is to artificially enlarge the positive samples and randomly reduce the negative samples using label-preserving transformations [4, 5]. In our work, we exploited data augmentation by extracting random patches from the previous image pairs like [12]. For example, in the VIPeR dataset, the resolution of the combined image pairs is $128 * 96$, and we chose the size of each patch is $112 * 84$. For the CUHK dataset, the original resolution for each image is $160 * 60$. We tried to shrink the images into $128 * 48$ first. After that, the compound mode of the person images is similar with VIPeR dataset. Then, we shrank the combined image into $112 * 84$ for each negative sample pair. So the positive and negative samples pairs can be balanced via this way.

Training Platform and Training Strategy. We test the run time of the process after feedback selection. The networks are trained on the High-Performance Computing platform (HPC), which is composed of Dawning Cluster, HP Cluster, HP SMP Mainframe, GPU Cluster and Stroage System. And our algorithm is trained on the Dawning Cluster, which includes 93 computing nodes, 6 I/O nodes, 2 management nodes. For each node, there are 2 CPU with 12 cores with 2.2 GHz and the memory is 128 GB. It takes about 6 days to train the deepest network for CUHK. In addition, as the training and testing samples take up too much memory, our training algorithm adopts the mini-batch stochastic gradient descent proposed in [7]. The training data is divided into several mini-batches. And training errors are calculated upon each mini-batch in the softmax layer and get Back-Propagation to the lower layers.

3.3 Experimental Results on VIPeR Dataset

In the first experiment, we evaluated our algorithm on the VIPeR dataset. We exploited 316 person image pairs for training and 316 person image pairs for testing. Similar to [12] positive and negative sample pairs were balanced in the procedure. It means that the dimension of the feature of each image pair was 9408 (112*84). Then, the feature acted as the input of the SAE networks. Besides, each sample belongs to a certain class. As described above, if the image pair is matched. The label is $y = 1$ and $y = 0$, otherwise.

In our multiple SAE models, there were three kinds of deep networks. The hidden layers were set 2, 3 and 4 for these SAE networks, respectively. And the hidden units of each network were set 1000, correspondingly. In addition, for the single auto-encoder network, the numbers of units for each hidden layer were the same. We exploited the SAE-*K*(The *K* denotes the numbers of the hidden layers.) to represent the configuration of each SAE network. In the training phase, the input of the SAE network was 9408*100000 (we randomly selected 100000 samples which included positive and negative ones for training). There was a label for each one. At last, the output of the deep network was following by a softmax classifier. There were about 400 iterations for training the network architecture. In the testing phase, we exploited 316 samples pairs for predicting the accuracy. Three kinds of probability matrix were generated by three kinds of SAE networks. Then, the weighted assignment mechanism was used for making the final decision. After that, the probability was transformed into the recognition accuracy. The final performance would be confirmed through this way. After training an auto-encoder network, we would like to visualize the weights (filters) that learned by the algorithm and try to understand what has been learnt. Figure 6 shows some filers learned by the first hidden layer of our network. The filters of network have different texture patterns, which mean that they capture the information in a unified manner.

The experiment results and all the Cumulative Matching Characteristic (CMC) curves are shown in Figure 7. It shows not only the 3 results for 3 different networks, but also the combined results after balancing the similarities. From the figure, we conclude that the matching results of the 3 networks are different, and each of them is low. The CMC(1) for single SAE network is not very high. And the more hidden layers of the network, the higher of the recognition ac-

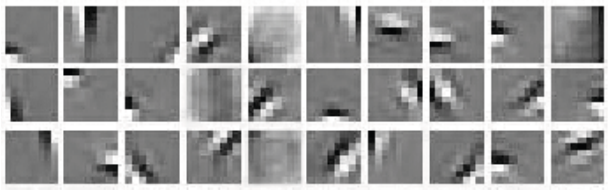

Figure 6. Visualization of Features: Visualization of some filters learned by a single auto-encoder network trained on VIPeR. The weights (called filters) of the first hidden layer for the auto-encoder network can be visualized.

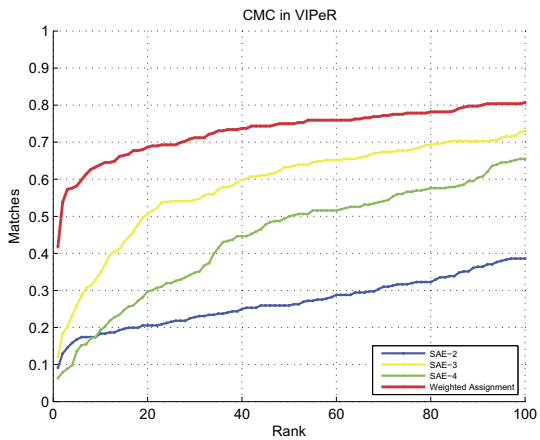

Figure 7. The recognition accuracies for different SAE models in VIPeR. The SAE-*K* denotes different hidden layers for the deep networks.

curacy for CMC(1). For the SAE-3, the accuracy is better than SAE-2. After exploiting the weighted assignment mechanism, the holistic recognition accuracy would be better than the single one and this phenomenon is consistent with our intuition.

As it is a classification problem in our work. There would be a final holistic accuracy for each SAE network and the classification network. For the test samples, each of them has a label which indicates the ground truth. It is a sign that was predicted by the softmax classifier. There are positive and negative samples in the test datasets. The holistic classification accuracy can manifest the performance of the model. The performances of the three deep models (SAE network and classification network) are shown in Figure 8. The accuracy indicates the correct classification for each test sample. If the accuracy is higher, the model is better to train. From the figure, the results are consistent with the CMC curves in Figure 7. The CMC(1) for SAE-4 is lower than SAE-2 in the figure. We think that there may exist too many connections and associated parameters between the adjacent layers in SAE-4. And they are difficult to tailor the performance.

Comparing with other metric learning methods, our algorithm has gotten the best recognition rate in CMC(1). The results can be seen as in Table 1. We compared with other six metric learning algorithms. The top two rows are the conventional metric learning methods. And the next four are the deep metric learning algorithms. We can see that our method enjoys the highest accuracy in CMC(1). When rank=10, our result is not very competitive. We guess that the results of the rank 10 may be led by the structure of SAE-4 which involves more hidden layers and associated parameters. Because the parameters are

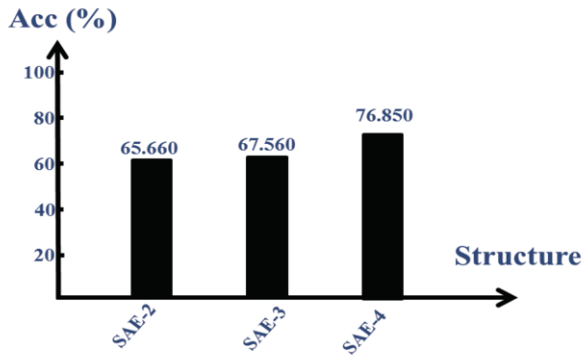

Figure 8. The overall classification performance in VIPeR for each deep model. Acc means the accuracy. From this figure, we can see that the coarse-to-fine classification mode is just like the character of the human brain which has many different levels of visual way. The SAE-*K* denotes the configurations of the SAEs.

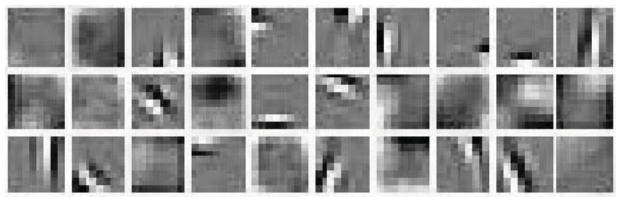

Figure 9. Some filters are learned on the CUHK.

over-fitting and the single result lower the holistic rank performance.

Table 1. Comparative results with the other metric learning algorithms on VIPeR.

Methods	Deep or Not	r=1	r=10
KISSME	Not	19.6%	62.2%
LMNN	Not	19.0%	58.1%
DML [24]	Yes	28.23%	73.45%
DDML [10]	Yes	29.56%	61.71%
DTML [11]	Yes	32.12%	65.92%
DCA [1]	Yes	34.81%	76.25%
Ours	Yes	**41.77**%	66.92%

3.4 Experimental Results on CUHK Dataset

In the second experiment, we evaluated our method on the CUHK dataset. The resolution of CUHK Campus is 60×160. Before training, we scaled them to 48×128 first. This dataset included 970 persons, which was divided into 485 for training and 485 for testing. They were also randomly selected. For each person image, we also preprocessed it like the VIPeR dataset. And the compound mode for each person image was the same as the first experiment. Some of filters learning from the first hidden layer show in Figure 9. In this experiment, there were also three kinds of deep neural networks to train. The architectures of the SAE networks were set via the same way like the previous one. In the training phase, we preprocessed the combined image pairs like the way in the VIPeR dataset. And the size for the input sample was also 9408(112*84). It was a label for each

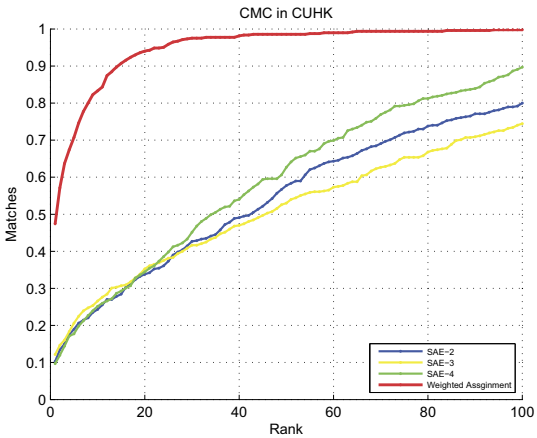

Figure 10. The recognition accuracies for different SAE networks in CUHK. The symbols denote the similar meaning like Figure 6.

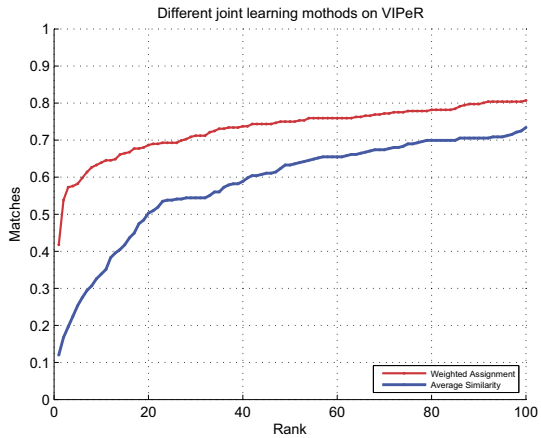

Figure 12. The different joint learning strategies on VIPeR. The weight assignment is better than average strategy.

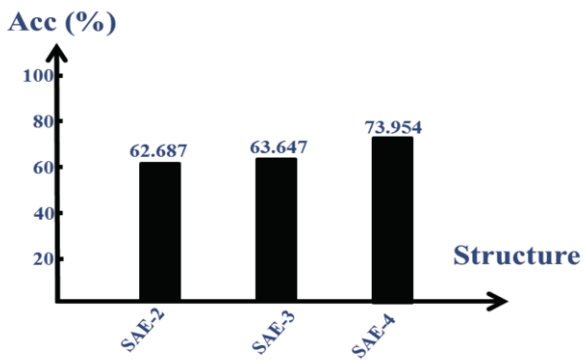

Figure 11. The overall classification performance in CUHK for each SAE network. The meaning of each notation in the figure is the same as in VIPeR.

one. The hidden layers for each SAE networks were set 2, 3 and 4, respectively. And the hidden units of each deep neural network were set 800, correspondingly. There were about 300 iterations for training the network architecture. At last, each of the auto-encoder networks was following by a softmax classifier. And the fine tuning was executed for the whole model via the back-Propagation algorithm. The output for each softmax classifier was the probability that a pair belongs to a certain class. A final accuracy was obtained via handling these probabilities. The recognition result can be seen as Figure 10.

Table 2. Comparative results with other metric learning algorithms on CUHK.

Methods	Deep or Not	r=1	r=10
KISSME	Not	29.40%	60.22%
LMNN	Not	21.17%	57.53%
FPNN [15]	Yes	27.87%	81.07%
DML [24]	Yes	16.17%	45.82%
Ours	Yes	**47.42%**	**83.29%**

From Figure 10, we can see that the single deep neural network for recognition performance is not very high. But the result is enhanced via exploiting the weighted assignment mechanism. This is in accordance with our intuition. The multi-scale metric learning is better than the single measure manner. The reason may be the same as the first experiment. And the result is not very precise. The holistic performance in this dataset is shown in Figure 11.

Comparing with other deep metric learning methods, our algorithm get the competitive result on the same dataset. The comparative results can be seen as Table 2. From it, we can see that our algorithm get the best accuracy comparing with other methods in CMC(1) and CMC(10). In addition, our model is simpler than other deep models for its architecture and training process comparing with [15, 1, 24]. Because they get involved into the structure modification, which take more time to train the deep networks. In our work, the deep models are just used to classification and the the results are joining together to get the final result. The time complexity and space complexity are simpler comparing with previous methods.

3.5 Performance Verification on Joint Learning Strategies

In this section, we would compare the joint strategies for our previous experiments. Firstly, we exploited the average similarity strategy as the joint learning on the two common datasets. Comparing with the different strategies, the experiment result can be seen as Figure as 12. From the figure, we can see that the performance of weights assignment mechanism is better than average. In CUHK dateset, the performance for compared strategy is also different. The results can be seen as Figure 13. In fact, the average similarity strategy is the special case for the weights assignment mechanism. As the architecture of the neural network is different, the degree of contribution for recognition accuracy is also different. So there are different weight factor and the joint strategy is very important for the final result.

4 Conclusions and Future Work

In this work, we have presented a method which utilizes multiple coarse-to-fine auto-encoder models to address person re-identification problem under varied environmental changes. In our algorithm, we have trained several different SAE networks, with each followed by a softmax classifier. So that the brain's visual cortex can be simulated by our established deep neural networks with different

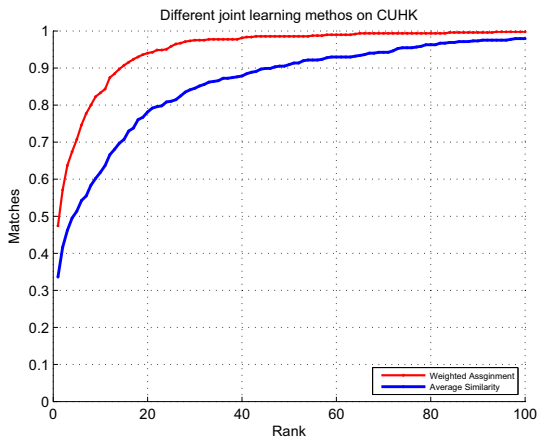

Figure 13. The different joint learning strategies on CUHK.

hidden layers. The preprocessed person image pairs via subtracting the mean value are used for network input and a couple of classification results are then produced. Finally, a weighted assignment mechanism is further used to boost recognition accuracy for the obtained classification results. Extensive experimental results on two public datasets have shown the superiority of our algorithm. Our established multiple coarse-to-fine deep metric learning approach can be extended to other visual applications, such as images classification, object detection and so on.

ACKNOWLEDGEMENTS

This research is supported by Development Program of China 863 Program (No. 2015AA016306), National Nature Science Foundation of China (No. 61231015, 61303114), The EU FP7 QUICK project under Grant Agreement No. PIRSES-GA-2013-612652*, Nature Science Foundation of Jiangsu Province (SBK2016040692). Specialized Research Fund for the Doctoral Program of Higher Education (No. 20130141120024), Nature Science Foundation of Hubei Province (2014CFB712) and National High Technology Research.

REFERENCES

[1] Ejaz Ahmed, Michael Jones, and Tim K Marks, 'An improved deep learning architecture for person re-identification', *Differences*, **5**, 25, (2015).

[2] Sola O Ajiboye, Philip Birch, Christopher Chatwin, and Rupert Young, 'Hierarchical video surveillance architecture: a chassis for video big data analytics and exploration', in *IS&T/SPIE Electronic Imaging*, pp. 94070K–94070K. International Society for Optics and Photonics, (2015).

[3] Yoshua Bengio, 'Learning deep architectures for ai', *Foundations and trends® in Machine Learning*, **2**(1), 1–127, (2009).

[4] Dan Ciresan, Ueli Meier, and Jürgen Schmidhuber, 'Multi-column deep neural networks for image classification', in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 3642–3649. IEEE, (2012).

[5] Dan C Cireşan, Ueli Meier, Jonathan Masci, Luca M Gambardella, and Jürgen Schmidhuber, 'High-performance neural networks for visual object classification', *arXiv preprint arXiv:1102.0183*, (2011).

[6] Piotr Dollár, Zhuowen Tu, Hai Tao, and Serge Belongie, 'Feature mining for image classification', in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1–8. IEEE, (2007).

[7] Ian J Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio, 'Maxout networks', *arXiv preprint arXiv:1302.4389*, (2013).

[8] Douglas Gray, Shane Brennan, and Hai Tao, 'Evaluating appearance models for recognition, reacquisition, and tracking', in *Proc. IEEE International Workshop on Performance Evaluation for Tracking and Surveillance (PETS)*, volume 3. Citeseer, (2007).

[9] Martin Hirzer, Peter M Roth, and Horst Bischof, 'Person re-identification by efficient impostor-based metric learning', in *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*, pp. 203–208. IEEE, (2012).

[10] Junlin Hu, Jiwen Lu, and Yap-Peng Tan, 'Discriminative deep metric learning for face verification in the wild', in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pp. 1875–1882. IEEE, (2014).

[11] Junlin Hu, Jiwen Lu, and Yap-Peng Tan, 'Deep transfer metric learning', in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pp. 325–333. IEEE, (2015).

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, 'Imagenet classification with deep convolutional neural networks', in *Advances in neural information processing systems*, pp. 1097–1105, (2012).

[13] Igor Kviatkovsky, Amit Adam, and Ehud Rivlin, 'Color invariants for person reidentification', *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **35**(7), 1622–1634, (2013).

[14] Wei Li and Xiaogang Wang, 'Locally aligned feature transforms across views', in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 3594–3601. IEEE, (2013).

[15] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang, 'Deepreid: Deep filter pairing neural network for person re-identification', in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pp. 152–159. IEEE, (2014).

[16] Chao Liang, Binyue Huang, Ruimin Hu, Chunjie Zhang, Xiaoyuan Jing, and Jing Xiao, 'A unsupervised person re-identification method using model based representation and ranking', in *Proceedings of the 23rd ACM international conference on Multimedia*, pp. 771–774. ACM, (2015).

[17] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio, 'Contractive auto-encoders: Explicit invariance during feature extraction', in *International Conference on Machine Learning (ICML)*, pp. 833–840, (2011).

[18] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams, 'Learning representations by back-propagating errors', *Cognitive modeling*, **5**, 3, (1988).

[19] Taiqing Wang, Shaogang Gong, Xiatian Zhu, and Shengjin Wang, 'Person re-identification by discriminative selection in video ranking', (2016).

[20] Yimin Wang, Ruimin Hu, Chao Liang, Chunjie Zhang, and Qingming Leng, 'Camera compensation using a feature projection matrix for person reidentification', *Circuits and Systems for Video Technology, IEEE Transactions on*, **24**(8), 1350–1361, (2014).

[21] Zheng Wang, Ruimin Hu, Chao Liang, Yi Yu, Junjun Jiang, Mang Ye, Jun Chen, and Qingming Leng, 'Zero-shot person re-identification via cross-view consistency', *IEEE Transactions on Multimedia*, **18**(2), 260–272, (2016).

[22] Zheng Wang, Ruimin Hu, Yi Yu, Chao Liang, and Wenxin Huang, 'Multi-level fusion for person re-identification with incomplete marks', in *Proceedings of the 23rd ACM international conference on Multimedia*, pp. 1267–1270. ACM, (2015).

[23] Mang Ye, Chao Liang, Zheng Wang, Qingming Leng, and Jun Chen, 'Ranking optimization for person re-identification via similarity and dissimilarity', in *Proceedings of the 23rd ACM international conference on Multimedia*, pp. 1239–1242. ACM, (2015).

[24] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li, 'Deep metric learning for person re-identification', in *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pp. 34–39. IEEE, (2014).

[25] Rui Zhao, Wanli Ouyang, and Xiaogang Wang, 'Unsupervised salience learning for person re-identification', in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 3586–3593. IEEE, (2013).

[26] Wei-Shi Zheng, Shaogang Gong, and Tao Xiang, 'Person re-identification by probabilistic relative distance comparison', in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 649–656. IEEE, (2011).