People Re-identification using Deep Convolutional Neural Network

Guanwen Zhang, Jien Kato, Yu Wang and Kenji Mase
Graduate School of Information Science, Nagoya University, Furo-cho, Chikusa-ku, Japan

Keywords: People Re-identification, Deep Convolutional Neural Network, Linear SVM.

Abstract: One key issue for people re-identification is to find good features or representation to bridge the gaps among different appearances of the same people, which is introduced by large variances in view point, illumination and non-rigid deformation. In this paper, we create a deep convolutional neural network (deep CNN) to solve this problem and integrate feature learning and re-identification into one framework. In order to deal with such ranking-like comparison problem, we introduce a linear support vector machine (linear SVM) to replace conventional softmax activation function. Instead of learning cross-entropy loss, we adopt a margin-based loss of pair-wise image to measure the similarity of the comparing pair. Although the proposed model is quite simple, the experimental result shows encouraging performance of our method.

1 INTRODUCTION

People re-identification refers to the problems that, recognize people when he/she leaves one camera view and enters another camera view, or recognize people when he/she reappears in the same field of view, which is crucial for inter-camera tracking and for understanding people behavior across camera network. It is a valuable task in video surveillance system and receives more and more attention with the spreading camera networks installation (Zheng et al., 2013)(Kviatkovsky et al., 2013)(Zhao et al., 2013)(Farenzena et al., 2010).

Due to low image resolution and the long distance between people and camera, biological information such as people's face or gait is general unavailable for people re-identification. In addition, because of the crossing camera issues, continuous visual tracking and intra-camera motion information of people cannot be immediately utilized. Therefore, in the current literature, studies on person re-identification mainly focus on analyzing the people appearance, with the acceptable assumption that people will not change their clothing during the observation period. The challenge in such an appearance based approach principally comes from appearance variance induced by light illumination, camera views, and non-rigid deformation of posture. This leads to the intra-camera variance being even larger than the inter-camera variance, that is, the same people could look considerably different in the videos captured by different cameras, whereas different people could look extremely similar in the videos captured by the same camera.

Figure 1: Selected images from VIPeR and CAVIAR4REID dataset. The people's appearance change with the variations in posture, illumination, and resolution. Even the same people look considerably different.

Exiting researches, which are trying to bridge the "gap" between the different appearances of the same people, can be roughly divided into two groups. The first group focuses on extracting discriminative appearance of people to form a stable feature representation. In these studies, the global and local features, such as color (Gray and Tao, 2008), shape (Wang et al., 2007), or texture (Bazzani et al., 2012), are integrated over images. Spatial information, such as pictorial structure (Cheng et al., 2011), co-occurrence representation (Wang et al., 2007), symmetry factors (Farenzena et al., 2010) and salience (Zhao et al.,

Zhang G., Kato J., Wang Y. and Mase K..
People Re-identification using Deep Convolutional Neural Network.
DOI: 10.5220/0004740302160223
In *Proceedings of the 9th International Conference on Computer Vision Theory and Applications* (VISAPP-2014), pages 216-223
ISBN: 978-989-758-009-3

2013), are also incorporated to deal with the lack of spatial information of histogram description. The second group, including a few researches, is working on measuring the similarity between representations. These methods, such as the well-known relative distance comparison (RDC) (Zheng et al., 2013), local aligned feature transform (Li and Wang, 2013), local distance comparison (Zhang et al., 2012) and large margin nearest neighbor with rejection (Dikmen et al., 2010), also achieve good performance.

All of the existing methods perform the higher and complex model by using or being based on handcrafted features. A great amount of time and manual work will be needed for different specific target re-identification task. Therefore, an end-to-end solution for re-identification task will be very valuable. As the great improvement has been achieved by deep learning in various tasks (Krizhevsky et al., 2012)(Sermanet et al., 2012), representation learning from raw input image seems to be a promising technique for people re-identification work.

In this paper, we try to utilize deep convolutional neural network (deep CNN) to solve people re-identification problem, and thus incorporate feature learning and re-identification into one framework. In practical use, people re-identification needs performing ranking-like comparison. In order to measure similarity of the comparing pair, we introduce linear support vector machine (linear SVM) to replace the traditional softmax activation function. Instead of learning cross-entropy loss for predicting class label, we measure the distance to the decision boundary that is more suitable for re-identification task. Our work follows the conventional approaches and pre-trains the layers with a unsupervised learning method in a greedy layer-wise manner. Dropout technique is also adopted in supervised learning phase to overcome the overfitting problem. Although the model we used is quite simple and results have not reached those of the state-of-the-art methods, the experiments on public datasets still show encourage performance.

There are three kinds of contribution in this paper: (1) we proposed a simple architecture of deep CNN for people re-identification problem that has not been addressed before; (2) we introduced linear SVM on the top of the network to measure the ranking comparison that is needed by people re-identification; (3) we gave a detailed discussion about the limitation of using deep learning in re-identification problem and the potential for further improvement.

The rest of the paper is organized as follows. The details of our approach are discussed in Section 2. We first explain the architecture of deep CNN (2.1). The linear SVM is then introduced in (2.2). At the end of this section, we briefly describe unsupervised learning and dropout techniques (2.3). Experimental results are discussed in Section 3. Finally, we present our conclusions and future perspectives in Section 4.

2 METHODOLOGY

People re-identification needs measuring the similarity of the comparing images. Multiple comparison results in a ranking list. Those with the highest similarity are selected as the results. The deep CNNs that have been reported so far are generally working on classification problems rather than comparison problems. In order to apply the deep CNN to re-identification problem, we design the architecture with two input branches, and introduce a linear support vector machine to measure the similarity of two input images.

In following sections, we give details about the architecture of our proposed network, and describe the techniques we used in network training.

2.1 Architecture

The architecture of our deep CNN is summarized in Fig.1. It contains 8 layers with three convolutional layers (C1, C3 and C5), three subsampling layers (S2, S4 and S6), one fully-connected layer (F7) and one weighting layer (W8). One subsampling layer follows one convolutional layer with local connection. S6 and F7 are fully-connected. F7 and W8 play the function of a linear SVM together. In the training phase, our deep CNN minimizes the squared hinge loss of the linear SVM that is equivalent to finding the max margin according to the true match (+1) and false match (-1) over training sample pair. In the testing phase, the similarity of the input image pair could be measured by the distance to the decision boundary.

The two images, with the R, G, B channel, are locally connected to the first convolutional layer (C1) through two different branches. One kernel in this convolutional layer is working on the three channels simultaneously and produces one channel. There are 32 kernels in this layer (in each branch) that produce 32 output channels totally. The neurons in the second convolutional (C3) layer are connected to both branches in the previous subsampling layer (S2), which follows the first convolutional layer (C1). The outputs of the kernels of two branches are simply added together as the kernel maps. Another convolutional and subsampling pair (C5 and S6), with more kernels, follows the second subsampling layer (S4).

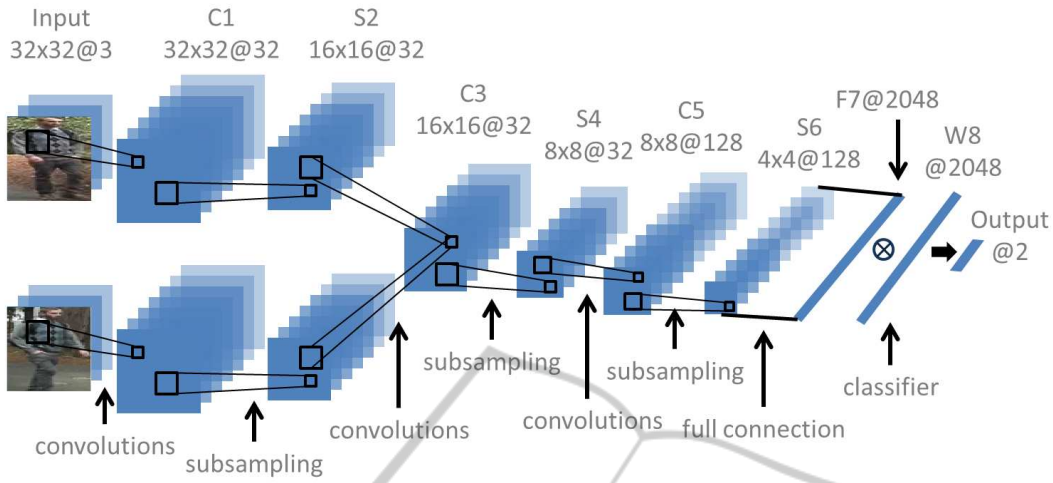

Figure 2: An illustration of the architecture of our deep CNN. There are two branches used for input of comparing pair. One subsampling layer follows one convolutional layer, and works as a pair. The last two layers are fully connected. A linear SVM classifier is added at top to replace the softmax layer for measuring similarity of input images pair.

The neurons in the fully-connected layer (F7) are connected to all neurons in the third subsampling layer (S6). The rectified-linearity is applied to the neurons of every convolutional layer. We use the max-pooling in both two branches in the first subsampling layer (S2), and use the average-pooling in the second and third subsampling layers (S4 and S6).

The first convolutional layer (C1) filters 32x32x3 input images with 32 kernels of size 5x5x3 with a stride 1 pixels (for every convolution, we pad the previous kernel map with a 2-pixel border of zeros). The input for the two branches in the first subsampling (S2) layer is 32 kernel maps with size of 32x32. Subsampling is performed for each channel and thus the number of output channels is equal to number of input. We use the filter with size of 3x3 by 2-pixel stride to reduce the input kernel map into half size. The second convolutional layer (C3) takes the both outputs of subsampling (S2) on two branches as the input, and filters it with 32 kernels of size 16x16x2 by 1 pixels stride. The third convolutional layer (C5) has 128 kernels of size 8x8x32, and the fully-connected layer (F7) has the 2048 nodes with fully connection to the 4x4x128 neurons in the third subsampling layer (S6). The processing in the second and third subsampling layers (S4 and S6) is performed as same as the first subsampling layer (S2) and is down-sampled by a factor of 2.

2.2 Softmax vs. Linear SVM

In conventional methods, it is popular to use the softmax activate function in the top layer to predict the classes. Assume there is a fully connection between softmax and penultimate layer and we have a N classes to predict, the number of the nodes in softmax layer will have the same number N as the classes. Let h_j be the activation of node j in penultimate layer, and W_{ji} be the weighting of the connection between node j in penultimate layer and node i in softmax layer. Since there is a fully connection, the input for the node i in softmax layer is then given by $a_i = \sum_j h_j W_{ji}$. The probability of class i, i.e. the output of node i, is defined as:

$$p_i = \frac{exp(a_i)}{\sum_k^N exp(a_k)}, \quad (1)$$

and $\sum_i^N p_i = 1$. In this case, the predicted class label would be $\widehat{i} = argmax(p_i)$.

In this paper, in order to measure similarity of the comparing pair, we introduce linear support vector machine to replace the softmax layer. Given the training data $\{x_n, t_n\}_{n=1}^N$, where $x_n \in R^D$ and $t_n \in \{-1, +1\}$, the linear support vector machine (linear SVM) could be formulated as the following optimization problem:

$$obj(w) = \frac{1}{2} w^T w + C \sum_{n=1}^{N} (max(1 - w^T x_n t_n, 0))^2. \quad (2)$$

This is known as L2-SVM, which is a popular optimization of SVM due to its differentiable and harder punishment of violating samples. The predicted class label could be obtain by $\widehat{t} = argmax(w^T x)t$.

In order to use the objective function as the supervised learning to train the parameters in low layers, we should back propagate the gradients of the linear SVM. The differentiate of the linear SVM, with respective to w, is given by

$$\frac{\partial obj(w)}{\partial w} = w - 2C x_n t_n (max(1 - w^T x_n t_n, 0)). \quad (3)$$

By introducing activation, $\mathbf{h} = (h_1, \ldots, h_{2048})^T$, of the neurons in penultimate layer (F7), the differentiate for each activation h_i is given by

$$\frac{\partial obj(w)}{\partial h_i} = w - 2Ch_i t_n(max(1 - w^T h_i t_n, 0)), \quad (4)$$

where t_n indicates true or false match of the input pair. By using such gradient to replace the gradient of softmax function, we could use the same back-propagation algorithm as in traditional deep learning method to train the parameters for each layer.

We have noted that the same strategy is also used in (Zhong et al., 2000)(Nagi et al., 2012)(Tang, 2013). However, rather than on the class label of the input data, we focus more on the distance of input pair to the decision boundary, where larger distance indicates the higher similarity.

2.3 Unsupervised Learning and Dropout

Generally, there are only a few training data in the re-identification task. However, our neural network architecture has thousands of parameters. Therefore, it is difficult to learn so many parameters without considerable overfitting. Below, we describe two primary ways in which we combat overfitting problem.

2.3.1 Unsupervised Learning

In conventional approaches for training a deep network, the unknown parameters are first randomly initialized, and then learned by directly searching gradient descent of supervised objective function. However this kind of methods often leads to local minimum and performance gets worse as the depth of network increasing. Hinton et al. proposed a greedy layer-wise unsupervised pre-training to deal with such problem (Hinton and Salakhutdinov, 2006). They used unsupervised learning algorithm to pre-train each layer, where out of the previous layer is used as input for training following layer. After the pre-training stage, the whole network is fine-tuned and finally realizes the optimization with a global supervised objective function. In this way, the unsupervised learning, working as initializing parameters phase, leads to much better solution in term of generalization performance.

In this paper, following the similar idea of (LeCun et al., 2010)(Sermanet et al., 2012), we use Predictive Sparse Decomposition (PSD) as unsupervised learning method to pre-train each layer of our deep CNN. PSD approximates the inputs as a sparse linear transformation on a dictionary. Similar as sparse coding, the sparse representation Z^* could be obtained by minimizing the energy function as follows:

$$E(Z,W,K) = \|X - WZ\|_2^2 + \lambda\|Z\|_1 + \|Z - C(X,K)\|_2^2 \quad (5)$$

where X is the input image, K is the filters in current layer that we want to learn and matrix W is the dictionary which is randomly initialized. The unsupervised learning is processing in two steps: in the first step, we find sparse representation Z^* and in the second step, we update the dictionary matrix W and filters K. The details can be referred to (Sermanet et al., 2012) and (LeCun et al., 2010).

2.3.2 Dropout

Dropout is an efficient technique introduced in (Hinton et al., 2012) which can reduce the generalization error of deep architecture neural network. Similar as denoising AutoEncoder, dropout randomly select a fraction of neurons in the hidden layers, and force them to be inactivated by setting the noise zeros. The selected neurons in every epoch do not contribute to the forward pass and also do not participate in back-propagation. However, unlike in denoising AutoEncoder, the dropout is performed in supervised training and could be used in all layers in a deep neural network for different purposes (Hinton et al., 2012). In this way, the neural network samples the network with different connectivity patterns, but all these architectures could share weights by the neurons that are not dropped out. By this kind of randomly sampling, the network is likely forced to learn an averaging model and finally achieve a robust model to battle against overfitting. We use dropout in the third convolutional layer (C5) and fully-connected layer (F7). We choose 0.5 as the dropout rate of parameters.

3 EXPERIMENT

3.1 Dataset Description

We evaluated our proposed deep CNN by applying it to two public datasets VIPeR of Douglas et al. (Gray et al., 2007) and CAVIAR4REID of Cheng et al. (Cheng et al., 2011). These datasets cover different genres and include different people postures, under a variety of illumination conditions, with various degrees of occlusion and camera resolution. Therefore, they are very challenging. VIPeR is originally created for viewpoint invariant pedestrian recognition. There are 632 image pairs in this dataset which are captured by two camera views in outdoor environment. Due to the arbitrary change of the view point under vary-

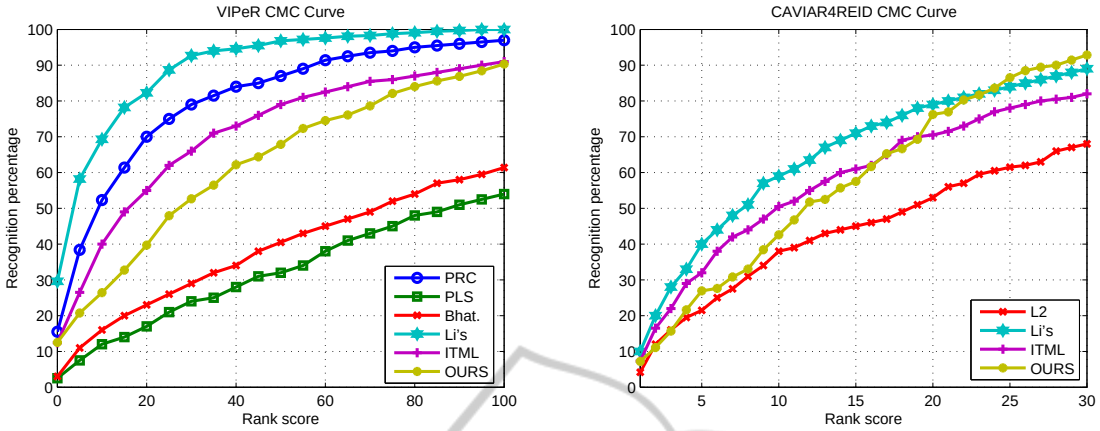

Figure 3: Evaluation of CMC curve for VIPeR and CAVIAR4REID dataset with the gallery size as 316 and 36 respectively. The state-of-the-art and base line methods are used as comparison. The corresponding results are obtained from the public papers.

Table 1: Matching rate of top ranking(%) on VIPeR dataset, gallery size is 316. The bold and red typeface are used to highlight the best results and the baseline results. Our results are shown in first the row.

Method	Top 1	Top 10	Top 20	Top 30	Top 40	Top 50	Top 60	Top 80	Top 100
OURS	12.5	26.3	39.7	52.6	62.1	67.9	74.5	85.0	90.4
RDC	15.7	53.9	70.1	79.4	83.5	87.4	90.2	92.7	96.7
PLS	2.7	10.9	17.3	24.3	28.6	32.2	38.1	48.4	53.8
Xing's	4.6	16.6	24.4	30.4	33.9	39.2	44.5	54.8	61.2
L1-norm	4.2	16.5	23.8	29.7	32.4	37.7	42.6	50.1	56.7
Li's	**29.6**	**69.3**	**82.3**	**91.7**	**94.6**	**96.8**	**97.6**	**99.1**	**100**
ITML	12.4	39.7	55.2	66.3	72.9	78.7	82.5	87.2	91.3

ing illumination between two camera views, appearance of the samples in an image pair has great difference. CAVIAR4REID is built specifically for person re-identification tasks. It consists of 72 people with $1,220$ images. 50 people are captured in two camera views, and the remaining 22 are captured in a single camera view. The images in this dataset are selected by maximizing the variance with respect to the resolution, illumination, occlusion and posture (Cheng et al., 2011).

3.2 Evaluation Method

We randomly divide each dataset into a training set and a testing set according to people's number. The training samples in each training set consist of two kinds of pairs: true pair and false pair. Given an image from one camera view, the true pair is created by selecting the image of the same people in the other view, while the false is created by randomly selecting the image of a different people from the other view. Since there are multiple images for one person in CAVIAR4REID dataset, the training pairs are created by selecting all images of each person. In this way, we have 632 and around 500 training pairs for VIPeR and CAVIAR4REID datasets respectively.

Each testing set is composed of a gallery set and a probe set. The gallery set consists of one image of each person, and the remaining images are used as the probe set. Re-identification is conducted by finding a match from the gallery for each probe. During experiments, this procedure is repeated 5 times to obtain average performance. The popular evaluation criterion, Cumulative Matching Characteristic (CMC), which represents the expectation of finding the correct match in the top n candidates, is used to measure matching rates.

Our proposed method is implemented by C++ and CUDA with library of Python. The convolution work is based on the CUDA kernels published by Alex Krizhevsky[1]. The graphic card we use in the experiments is NVIDIA GTX 660.

In the training phase, the initial parameters are set by a uniform distribution in the range $[-0.05, 0.05]$. We also initialize the basis in the convolutional layers with the constant 0.002 and set the momentum as 0.9, weight decay as 0.004 and batch size as 50. We use all training pairs of two datasets for the pre-trained unsu-

[1]http://code.google.com/p/cuda-convnet

Table 2: Matching rate of top ranking(%) on CAVIAR4REID dataset, gallery size is 36. The bold and red typeface are used to highlight the best results and the baseline results. Our results are shown in first the row.

Method	Top 1	Top 5	Top 10	Top 15	Top 20	Top 25	Top 30
OURS	7.2	26.9	42.6	57.5	76.25	**86.5**	**92.6**
L2	4.1	21.5	37.8	44.6	53.2	61.4	68.6
PS	8.5	32	48.8	59.7	66.4	79.7	86.6
SDALF	6.8	25	45	55	64.5	74	83
Li's	**10.2**	**39**	**59**	**71.4**	**79.5**	84.4	88.2
ITML	7.3	32.49	50.5	61.3	70.4	77.8	82

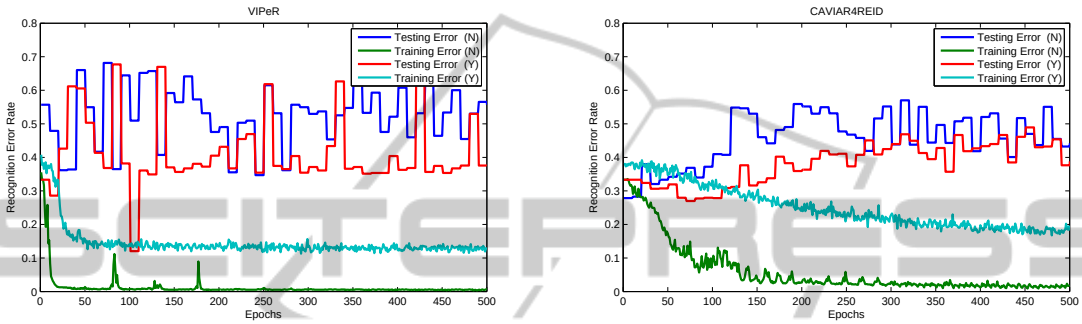

Figure 4: Training error rate and testing error rate of the identification on VIPeR and CAVIAR4REID dataset. They are evaluated by USING pre-trained unsupervised learning and dropout (Y), and NOT USING these techniques (N). The testing errors are obtained by every 10 epochs.

pervised learning. The dropout technique is only used in fine-tune stage, and the dropout rate is set as 0.5. During fine-tune stage for specific target training set, we follow the similar strategy as (Krizhevsky et al., 2012) and manually adjust learning rate throughout training.

3.3 Results

In experiments, we only choose the learning-based method that works on the two datasets. We compare our proposed method with the baseline methods, partial least squares (PLS) (Schwartz and Davis, 2009) and Bhattacharyya distance learning (Bhat.), on VIPeR, and compare with baseline method, Euclidean distance learning (L2), on CAVIAR4REID. We also compare our method with some well-known state-of-the-art methods such as relative distance comparison (RDC) (Zheng et al., 2013), information theoretical metric learning (ITML) and the method proposed by Li et al. (Li's) (Li and Wang, 2013). The results of above mentioned methods are obtained from the published papers (Zheng et al., 2013)(Li and Wang, 2013).

The comparison results are shown in Fig.3 and Table 1 and 2. It can be seen that our proposed model outperforms the baseline methods on these datasets, but it is still worse than the state-of-the-art methods. Notice that the CMC curve of our method gets competitive after rank 15 on the CAVIAR4REID. The size of training pairs of VIPeR and CAVIAR4REID are 632 and around 500 (people with 10 images may be selected as training set), respectively. However, the number of people, that will be re-identified, is 316 and 36. Therefore, the relative number of training pairs for each person in VIPeR is 2 $(= 632/316)$, much smaller than 14 $(= 500/36)$ in CAVIAR4REID. This makes the performance of CAVIAR4REID is better than that of VIPeR.

We also notice that, although we have introduced pre-trained unsupervised learning and dropout technique, overfitting is still severe. In order to show the details, we give a comparison of training and testing errors obtained by using or not using unsupervised learning and dropout technique. The testing errors are obtained from a validation set, which is created in the same way as training set, but using the images of the people in testing set. In Fig.4, it can be seen clearly that, without using unsupervised learning and dropout technique, the training errors on both datasets are close to zeros, while the testing errors are quite high. By introducing unsupervised learning and dropout technique, the divergence between training and testing error reduces on both datasets. Nevertheless, the higher and wild swing testing errors still reveal the networks suffer from overfitting.

Increasing the training data is essential way to reduce overfitting and improve performance of the net-

work. However, for re-identification task, since a positive sample (true match) consists of images from the same people, the number of the people in the dataset restricts the number of the positive samples. This makes the number of positive samples usually much smaller than that of negative samples. In the training phase, the negative samples should be limited within a certain amount to avoid overfitting.

Multiple-shot dataset of re-identification task is a good choice for solving this problem. As shown in experimental results, by creating more positive samples crossing multiple images, the performance of network on CAVIAR4REID is better than that on VIPeR. During the experiments, we observe that the performance of the network on some multiple shot datasets, such as ETHZ (Ess et al., 2007) and Person Re-ID 2011 (Hirzer et al., 2011), is not impressive. By further inspecting these datasets, we find that multiple images of the people are extracted from video sequences. The difference between images is small, and they cannot contribute to training work.

From another point of view, similar as denoising AutoEncoder, it is possible to increase positive samples by applying some transformations to original images, such as partial corruption of the input image pairs, extracting random patch of the images and so on. By such a strategy, we can not only increase positive samples to combat overfitting issue, but also can improve the robustness of the network against noise.

4 CONCLUSIONS

How to find a good feature representation to bridge the "gap" between appearances of the same people is a very challenging task. Existing methods either employ hand craft features or use machine learning method with existing features to form a specific representation. However, there are a lot of uncertainty in these methods due to human factors and specific applications. Deep learning, with ability to learn a proper feature representation from the bottom of the raw images, seems to be a promising solution for the people re-identification tasks.

In this paper, we utilize deep convolutional neural network to solve people re-identification problem. We integrate feature learning and re-identification into one framework, and accomplish learning and re-identification simultaneously. In order to deal with the ranking-like comparison problem, we introduce a linear support vector machine to replace the softmax lay for measuring the similarity of the comparing images. Since there is a large amount of parameters of the network needed to be estimated, while only a small number of training data are available, the pre-trained unsupervised learning and dropout technique are used to reduce overfitting.

Although the proposed is quite simple, we still achieve very encourage performance compared with baseline methods, which gives us great confidence. But compared with the state-of-the-art methods, our performance needs to be further improved. The careful analysis on the results shows that the serious overfitting caused by the lack of positive training samples seems to be the reason. This is our future work.

ACKNOWLEDGEMENTS

This research was supported by the National Institute of Information and Communication Technology (NICT), and by the Strategic Information and Communications R&D Promotion Programme (No. 131306004). Yu Wang is supported by Grant-in-Aid for Japan Society for the Promotion of Science and Guanwen Zhang is also supported by the Fund of the China Scholarship Council.

REFERENCES

Bazzani, L., Cristani, M., Perina, A., and Murino, V. (2012). *Multiple-shot Person Re-identification by Chromatic and Epitomic Analyses*, volume 33.

Cheng, D. S., Cristani, M., Stoppa, M., Bazzani, L., and Murino, V. (2011). *Custom Pictorial Structures for Re-identification*.

Dikmen, M., Akbas, E., Huang, T. S., and Ahuja, N. (2010). Pedestrian Recognition with a Learned Metric. *Proc. Asia Conf. Computer Vision*, pages 501–512.

Ess, A., Leibe, B., and van Gool, L. (2007). Depth and Appearance for Mobile Scene Analysis. *Proc. Int'l Conf. Computer Vision*, pages 1–8.

Farenzena, M., Bazzani, L., Perina, A., Murino, V., and Cristani, M. (2010). *Person Re-Identification by Symmetry-Driven Accumulation of Local Features*.

Gray, D., Brennan, S., and Tao, H. (2007). Evaluating appearance models for recognition, reacquisition, and tracking. In *10th IEEE Int'l Workshop on Performance Evaluation of Tracking and Surveillance*.

Gray, D. and Tao, H. (2008). Viewpoint Invariant Pedestrian Recognition with an Ensemble of Localized Features. *Proc. European Conf. Computer Vision*, pages 262–275.

Hinton, G. E. and Salakhutdinov, R. R. (2006). *Reducing the dimensionality of data with neural networks*, volume 313.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2012). *Improving neural networks by preventing co-adaptation of feature detectors*, volume abs/1207.0580.

Hirzer, M., Beleznai, C., Roth, P. M., and Bischof, H. (2011). Person re-identification by descriptive and discriminative classification. *Proc. Scandinavian Conf. on Image Analysis*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Proc. Advances in Neural Information Processing Systems 25*, pages 1106–1114.

Kviatkovsky, I., Adam, A., and Rivlin, E. (2013). *Color Invariants for Person Reidentification*, volume 35.

LeCun, Y., Kavukcuoglu, K., and Farabet, C. (2010). Convolutional networks and applications in vision. pages 253–256.

Li, W. and Wang, X. (2013). *Locally Aligned Feature Transforma accros Views*.

Nagi, J., Di Caro, G. A., Giusti, A., , Nagi, F., and Gambardella, L. (2012). Convolutional Neural Support Vector Machines: Hybrid visual pattern classifiers for multi-robot systems. pages 27–34.

Schwartz, W. R. and Davis, L. S. (2009). Learning Discriminative Appearance-Based Models Using Partial Least Squares. *Proc. 2009 XXII Brazilian Symposium on Computer Graphics and Image Processing*, pages 322–329.

Sermanet, P., Kavukcuoglu, K., Chintala, S., and LeCun, Y. (2012). *Pedestrian Detection with Unsupervised Multi-Stage Feature Learning*, volume abs/1212.0142.

Tang, Y. (2013). *Deep Learning using Support Vector Machines*, volume abs/1306.0239.

Wang, X., Doretto, G., Sebastian, T., Rittscher, J., and Tu, P. (2007). Shape and Appearance Context Modeling. *Proc. Int'l Conf. Computer Vision*, pages 1–8.

Zhang, G., Wang, Y., Kato, J., Marutani, T., and Mase, K. (2012). Local Distance Comparison for Multiple-shot People Re-identification. *Proc. Asia Conf. Computer Vision*, pages 677–690.

Zhao, R., Ouyang, W., and Wang, X. (2013). *Unsupervised Salience Learning for Person Re-identification*.

Zheng, W.-S., Gong, S., and Xiang, T. (2013). *Re-identification by Relative Distance Comparison*, volume 99.

Zhong, S., , Zhong, S., and Ghosh, J. (2000). *Decision Boundary Focused Neural Network Classifier*.