

Deep Person Re-Identification with Improved Embedding

Haibo Jin, Xiaobo Wang, Shengcai Liao*, and Stan Z. Li

Center for Biometrics and Security Research & National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences, Beijing, China

haibo.nick.jin@gmail.com, {xiaobo.wang, scliao, szli}@nlpr.ia.ac.cn

Abstract

Person re-identification task has been greatly boosted by deep convolutional neural networks (CNNs) in recent years. The core of which is to enlarge the inter-class distinction as well as reduce the intra-class variance. However, to achieve this, existing deep models prefer to adopt image pairs or triplets to form verification loss, which is inefficient and unstable since the number of training pairs or triplets grows rapidly as the number of training data grows. Moreover, their performance is limited since they ignore the fact that different dimension of embedding may play different importance. In this paper, we propose to employ identification loss with center loss to train a deep model for person re-identification. The training process is efficient since it does not require image pairs or triplets for training while the inter-class distinction and intra-class variance are well handled. To boost the performance, a new feature reweighting (FRW) layer is designed to explicitly emphasize the importance of each embedding dimension, thus leading to an improved embedding. Experiments on several benchmark datasets have shown the superiority of our method over the state-of-the-art alternatives.

1. Introduction

Person re-identification aims to re-identify a query person across multiple non-overlapping cameras. The task is challenging since pedestrian images from different camera views usually suffer from large variations in viewpoints, poses, lightings and backgrounds. Many earlier works solve the re-identification problem by dividing it into two separated parts: feature extraction [11, 12, 33, 29, 7, 9, 14, 28] and metric learning [16, 31, 32, 2, 8, 11, 12]. A large number of hand-crafted features are designed to enhance the robustness of pedestrian images to pose, viewpoint and illumination changes. After the feature is extracted, metric learning is applied to learn a metric for the features so that the

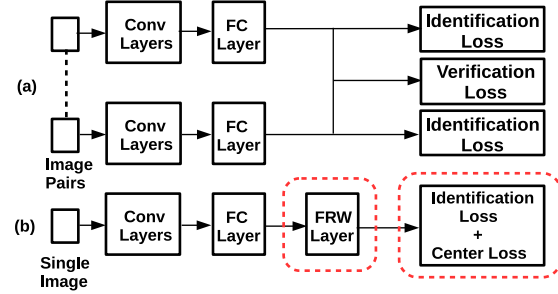


Figure 1: The difference between the state-of-the-art CNN [5, 36] and our proposed CNN for person re-identification. (a) The current best CNN has two branches, which takes a pair of images as input. (b) Our proposed CNN does not require image pairs or triplets for training since it utilizes the combination of identification loss and center loss. Moreover, a new feature reweighting (FRW) layer is designed so that the importance of each embedding dimension can be adaptively adjusted.

images of the same person are close while the ones of different pedestrians are far away from each other in the metric space.

In recent years, deep convolutional neural networks (CNNs) have achieved promising results on person re-identification [1, 3, 4, 5, 10, 13, 17, 18, 20, 21, 22, 23, 26, 30, 36] due to their advantages on feature learning. Different from previous works, CNN approaches learn features and metrics jointly from data in an end-to-end manner. Then an embedding is learned to measure the similarities between images using Euclidean distance. The loss function of a CNN plays an important role on its performance. Verification loss [1, 3, 4, 10, 17, 20, 21, 22, 23, 30] is popular among CNNs on person re-identification task benefited from its simple motivation: reducing the variance between intra-class embeddings while increasing the distinction between inter-class ones. However, verification loss takes image pairs or triplets for training, the number of which grows rapidly as the number of classes grows. When

*Corresponding author.

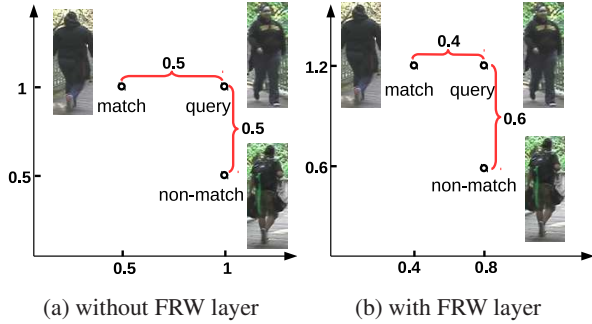


Figure 2: The effect of applying FRW layer. (a) Without FRW layer, the 2D embeddings of a matched image and a non-matched image have the same distance to the query embedding. (b) With FRW layer, the distance between the matched embedding and the query is closer than the non-matched one because the more essential dimension is enlarged while the less important one is shrunk.

there are numerous person identities, verification loss is likely to show slow convergence and unstable performance. Therefore, verification loss is not scalable to large-scale data. Identification loss is usually used for classification task, and it has also been applied to person re-identification task [26, 35, 27] due to its simplicity and discriminative ability. Though identification loss can separate inter-class embeddings efficiently, it does not explicitly reduce the intra-class variance. Consequently, the performance may be limited since the embeddings of the same person can have large distance on test data due to viewpoint, pose and background variations. To absorb the merits of the above two losses, recent works [36, 5] tend to combine them (please see Fig. 1a), and have achieved promising results. However, the inefficiency issue from verification loss still remains despite their performance improvement.

In addition to the inefficiency problem, the existing deep embedding models draw little attention to the importance of each embedding dimension. They simply accumulate the squared difference of each dimension (*i.e.* Euclidean distance) to measure the distance between embeddings [3, 4, 5, 21, 22, 26, 36]. In other words, each dimension of an embedding contributes equally to the total distance, which is a coarse-grained assumption to some extent. For instance, imagine that a matched embedding and a non-matched embedding have the same distance to the embedding of the query image (Fig. 2a), Euclidean distance method is not able to distinguish the matched one from the two. If a model learns to measure the importance of each dimension, then reweights the embeddings so that the important dimension is emphasized while the unimportant one is depressed (Fig. 2b), such problem can be alleviated. Unfortunately, few works have considered the importance of

different embedding dimensions, which limits their performance in practice.

To overcome the above two shortcomings, this paper proposes a new CNN model for person re-identification task. Specifically, we employ identification loss with center loss to train the deep model, which does not require image pairs or triplets as input. Center loss [25] aims to pull images to the corresponding class center so that the intra-class variance is reduced. It functions similarly as verification loss but the learning process is more efficient. Meanwhile, a new feature reweighting (FRW) layer to adaptively learn the importance of each dimension has been designed. The FRW layer is placed after the embedding layer, performing simple element-wise multiplication upon its input. By doing so, the model gains the freedom to explicitly adjust the scales of the learned embeddings such that some less important features could be squeezed to avoid overfitting. Fig. 1b shows the structure of our proposed CNN. For clarity, the contributions of this paper can be summarized as follows:

- We employ identification loss with center loss to train a deep CNN model without constructing image pairs or triplets as input, thus improving the training efficiency.
- We design a new FRW layer to explicitly emphasize the importance of each embedding dimension, leading to an improved embedding to boost the performance.
- Experiments on CUHK03 [10], CUHK01 [9] and VIPeR [6] have validated the superiority of our method over the state-of-the-arts.

2. Related Work

To learn effective embeddings, existing works can be classified into two categories: 1) Improving the deep CNN structure to learn discriminative embeddings; 2) Designing better loss functions for deep CNN training.

CNN structure: To improve the CNN embeddings, Li *et al.* [10] propose to jointly handle various variations with filter pairing component. Yi *et al.* [30] design a Siamese CNN to handle the divided images to finally compute a merged similarity score between images. Ahmed *et al.* [1] propose a cross-input neighborhood difference layer to capture local relationship between two images as well as a patch summary layer to summarize the features learned from the previous layers. Wang *et al.* [23] propose a joint framework of single-image representation and cross-image representation to get a merged result. Xiao *et al.* [26] propose a CNN that learns features from multiple domains. Cheng *et al.* [3] design a multi-channel parts-based CNN to learn both the global and the local features. Varior *et al.* [21] propose a gating function for CNN to emphasize fine common local patterns. Varior *et al.* [22] use Long Short-Term Memory

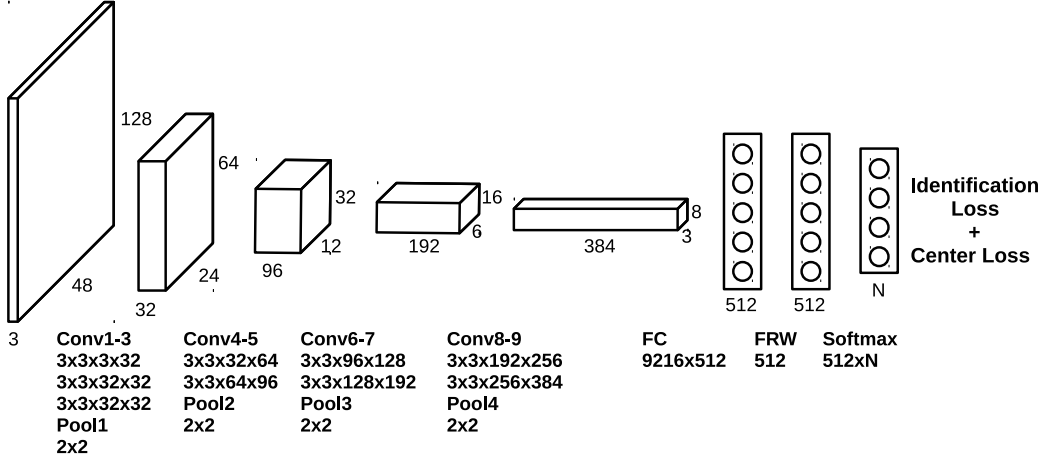


Figure 3: Our CNN architecture. A single image passes through several convolutional layers and max pooling layers, and a 512D vector is obtained by a fully connected layer. Then, FRW layer reweights the vector to get an improved embedding. Finally, the architecture is equipped with both identification loss and center loss to train the deep model.

to emphasize contextual information for learning features. Shi *et al.* [17] propose a moderate positive sample mining method to learn a variation insensitive feature. Although existing CNN structures have achieved promising results, they still suffer from learning inefficiency problem because of verification loss.

Loss function: Binary identification loss, contrastive loss and triplet loss are three main types of verification loss. CNNs with binary identification loss have been used by [10, 1]. They output a binary prediction, indicating whether the two images belong to the same identity or not. Many other deep models learn an embedding for each image, and compute the similarities between embeddings based on the Euclidean distance. The works [21, 22] use contrastive loss to train a CNN, which requires a pair of image samples for training. The methods [24, 3, 23, 4] employ triplet loss or its variations with CNN, which requires image triplets during training. For simplicity, the approaches [26, 27, 35] apply identification loss to person re-identification task since it learns discriminative features efficiently. The combination of identification loss and verification loss has been found effective on face recognition [19], and it also gives excellent performance on person re-identification [36, 5]. Recently, the work [25] propose center loss to reduce the intra-class variance on face recognition task, without constructing image pairs or triplets during training. However, for person re-identification, the mainstream loss to handle the intra-class variance is still verification loss.

3. Our CNN Model

3.1. The Overall Architecture

Our proposed CNN model is a single CNN (different from the previous Siamese CNNs) that consists of nine convolutional layers, four max pooling layers, one fully connected (FC) layer, one FRW layer, and finally a softmax classifier. Fig. 3 gives the detailed illustration of the model. All the convolutional layers use 3×3 filters, with stride 1 and zero paddings. The max pooling layers all have 2×2 filters with stride 2. Batch normalization is applied after each convolutional layer or FC layer to speed up the training. Then leaky rectified linear unit (LReLU) is used after these layers as the non-linear activation function. After the FRW layer, we get a 512D embedding equipped with identification loss and center loss.

3.2. Identification Loss and Center Loss

Identification loss aims to enlarge the inter-class distinction and is usually used for multi-class classification task. It can be formulated as follows:

$$L_I = -\frac{1}{M} \sum_{i=1}^M \log \frac{e^{\mathbf{W}_{y_i}^T \mathbf{x}_i + b_{y_i}}}{\sum_{j=1}^N e^{\mathbf{W}_j^T \mathbf{x}_i + b_j}}, \quad (1)$$

where M is the batch size, $\mathbf{x}_i \in \mathbb{R}^D$ is the i -th embedding of the batch, and y_i is the class label of the current input. \mathbf{W}_k is the k -th column of the fully connected weights $\mathbf{W} \in \mathbb{R}^{D \times N}$, b_k is the k -th item of the bias term $\mathbf{b} \in \mathbb{R}^N$, and N is the number of categories.

Center loss is proposed by Wen *et al.* [25] to reduce the intra-class variance for face recognition. It maintains a cen-

ter point for each class, and keeps pushing each image embedding to its corresponding center so that the variations between image embeddings and their centers are small. It can be formulated as follows:

$$L_C = \frac{1}{2M} \sum_{i=1}^M \|\mathbf{x}_i - \mathbf{c}_{y_i}\|_2^2, \quad (2)$$

where $\mathbf{c}_{y_i} \in \mathbb{R}^D$ is the corresponding center of the embedding \mathbf{x}_i . Specifically, unlike other CNN parameters, the updating of the class centers \mathbf{c}_{y_i} are additionally performed instead of backpropagation:

$$\begin{aligned} \Delta \mathbf{c}_j^t &= \frac{\sum_{i=1}^M \delta(y_i = j) \cdot (\mathbf{c}_j^t - \mathbf{x}_i)}{1 + \sum_{i=1}^M \delta(y_i = j)}, \\ \mathbf{c}_j^{t+1} &= \mathbf{c}_j^t - \alpha \cdot \Delta \mathbf{c}_j^t, \end{aligned} \quad (3)$$

where α is the learning rate of the centers ranging from 0 to 1, $\delta(\text{condition}) = 1$ if the *condition* is satisfied, otherwise $\delta(\text{condition}) = 0$.

During training, the two losses are optimized jointly using the formula as:

$$\begin{aligned} L &= L_I + \lambda L_C \\ &= -\frac{1}{M} \sum_{i=1}^M \log \frac{e^{W_{y_i}^T \mathbf{x}_i + b_{y_i}}}{\sum_{j=1}^N e^{W_j^T \mathbf{x}_i + b_j}} + \frac{\lambda}{2M} \sum_{i=1}^M \|\mathbf{x}_i - \mathbf{c}_{y_i}\|_2^2, \end{aligned} \quad (4)$$

where λ is a scalar to balance the two loss functions. As we can see from Eq. (4), the loss function of our model only involves batches of single image samples, which leads to the improvement of training efficiency over the existing deep person re-identification models.

3.3. FRW Layer

The importance of each embedding dimension has always been assumed to be equal in the existing works, ignoring the difference between different dimension. Here we argue that CNN should have the freedom to learn such difference. In this work, a new FRW layer is proposed to reweight the learned embedding of a CNN. More specifically, FRW layer performs an element-wise product of an embedding and the FRW weights, formulated as follows:

$$\hat{\mathbf{x}} = \mathbf{x} \odot \mathbf{w}_{frw}, \quad (5)$$

where $\mathbf{x} \in \mathbb{R}^D$ is a learned embedding, $\mathbf{w}_{frw} \in \mathbb{R}^D$ is the weights of FRW layer, and \odot denotes element-wise product. Intuitively, FRW layer enlarges certain dimensions of the embedding while reducing the other ones to strengthen the more essential features so that the similarities between embeddings can be reflected more accurately by Euclidean

distance. For example, the central area of a pedestrian image can be more important than the border areas. For stability, additional constraint on the weights of FRW layer has been developed:

$$L_F = \beta \cdot \left(\frac{1}{2} \|\mathbf{w}_{frw}\|_2^2 - C \right)^2, \quad (6)$$

where β controls the importance of the constraint, and C is a constant to constrain the norm of the weight vector.

From another perspective, FRW layer can be seen as a separated part from softmax classifier. Among deep embedding models, the weights of softmax classifier are usually discarded after training because these weights are trained specifically on training classes, which are useless for different testing classes. Nevertheless, the trained weights of softmax classifier do contain general knowledge that is irrelevant to classes. We can treat the trained softmax weights as two parts: one that contains knowledge specific to each training class and the other that learns general knowledge applicable to all classes. Accordingly, the softmax classifier and the FRW layer in our model handle the two kinds of knowledge respectively. Formally, the standard softmax weights can be decomposed as follows:

$$\begin{aligned} W_j^T \mathbf{x} + b_j &= (\hat{W}_j \odot \mathbf{w}_{frw})^T \mathbf{x} + b_j \\ &= \hat{W}_j^T (\mathbf{x} \odot \mathbf{w}_{frw}) + b_j, \end{aligned} \quad (7)$$

where W_j is the j -th column of a standard softmax classifier weight, \hat{W}_j is the j -th column of the softmax classifier weight from our model, and \mathbf{w}_{frw} is the weight of our FRW layer. From Eq. (7), we can see that the FRW layer and the softmax classifier in our model is equivalent to the standard softmax classifier. By separating a FRW layer from the standard softmax classifier, the learned general knowledge about feature importance could be merged into the embeddings, and so it is applicable in testing phase.

3.4. Training

There are two types of training paradigms in this paper: 1) For relatively large dataset (*e.g.* CUHK03 [10]), we simply train the model on its training set using stochastic gradient descent with mini-batches; 2) As for small datasets (*e.g.* CUHK01 [9], VIPeR [6]), we adopt a similar deep transfer learning method from [5]. We pre-train the model on large person re-identification datasets (CUHK03 [10]+Market1501 [34]), then fine-tune it on the corresponding training set of small data. Note that a two-stepped fine-tuning strategy from [5] is used in this paper to conduct a more effective transfer learning. After pre-training, the weights of the softmax classifier cannot be reused in the fine-tuning stage because the two datasets have different classes. Therefore, the softmax classifier weights should

be replaced by a randomly initialized one with N_s nodes, where N_s is the number of classes of the small dataset. Then, we do first-step fine-tuning by freezing the other parameters and only training the newly added weights until the classifier converges. After that, we fine tune all the parameters altogether as the second step. The reason for the two-stepped tuning is to avoid the newly added weights to backpropagate harmful gradients to the pretrained weights of the previous layers.

3.5. Testing

Testing is simple and efficient in the deep embedding scenario. We feed all the testing images to the CNN model, and we get an embedding for each of them. Then we normalize each embedding to a unit vector. Finally, we compute the Euclidean distance between all the pairs from two camera views to measure the cumulative match curve (CMC).

4. Experiments

4.1. Datasets

CUHK03: CUHK03 [10] consists of 13164 images from 1360 identities. It provides two settings, one is annotated by human (labeled) and the other one is annotated by detectors (detected). We adopt the detected setting since it is closer to practical scenarios. Following the protocol in [10], we do 20 random splits, wherein 1160 identities are for training, 100 identities are for testing. The evaluation is in single shot.

CUHK01: CUHK01 [9] contains 971 identities with two camera views, and each identity from each view has two images. Following the setting in [5], we randomly select one image for each identity in each view for both training and testing images. Then 485 identities are randomly selected for training, and the remaining 486 are for testing. The evaluation is based on 10 random splits, in single shot.

VIPeR: VIPeR [6] contains 632 identities with two camera views. Each identity from each view has one image. Half of the identities are used for training, and the other half are for testing. The evaluation is also based on 10 random splits, in single shot.

4.2. Data Preparation

To reduce overfitting, we conduct data augmentation on each dataset. Each training image is augmented by 2D random translation as in [1, 10]. We sample three images with 2D translation for each training image as well as a horizontal reflection. Each image is resized to 128×48 . The mean of the training images is subtracted from all the images for each dataset.

Method	Rank 1	Rank 5	Rank 10
XQDA [11]	46.3	78.9	88.6
MLAPG [12]	51.2	-	-
DNS [31]	54.7	84.8	94.8
LSSCDL [32]	51.2	-	-
Siamese LSTM [22]	57.3	80.1	88.3
IDLA [1]	45.0	76.0	83.5
Gated S-CNN [21]	61.8	80.9	88.3
EDM [17]	52.0	-	-
Joint Learning [23]	52.2	-	-
CAN [13]	63.1	82.9	88.2
CNN Embedding [36]	66.1	90.1	95.5
Deep Transfer [5]*	84.1	-	-
CNN-I	75.0	92.1	95.9
CNN-IV	80.2	94.9	97.3
CNN-IC	80.2	96.1	97.9
CNN-FC-IC	79.8	95.6	97.6
CNN-FRW-IC	82.1	96.2	98.2

Table 1: Accuracy on CUHK03 (detected). *Deep Transfer [5] uses ImageNet for pretraining, while CNN-IV is our implementation of [5] without ImageNet pretraining.

4.3. Models for Comparison

We compare our proposed model to a number of the existing methods, including state-of-the-art ones. In order to have a systematic comparison, we also implement several baseline models. We name the proposed model (Fig. 3) as CNN-FRW-IC. We implement a version without FRW layer to check the effectiveness of FRW layer, named CNN-IC. We also have one where the FRW layer is replaced by a fully connected layer, named CNN-FC-IC. We add the extra FC layer to check if simply increasing the depth of the network improves accuracy. There is also a version that only uses identification loss without FRW layer, named CNN-I.

To our knowledge, [5] gives the best performance among the existing deep embedding models, using identification loss and verification loss (binary identification loss) as loss function. So we implement a Siamese CNN with the two losses under our framework without FRW layer, named CNN-IV.

4.4. Training Settings

To update the parameters, we use the Adam optimizer, where the exponential decay rate for the 1st and 2nd moment estimates are 0.9 and 0.999, respectively. The number of training iterations is 25k. The initial learning rate is 0.001, decayed by 0.1 after 22k iterations. The batch size is set to 100. The weight decay is 0.001. As for the center loss, the updating rate of the centers are $\alpha = 0.5$, and its balance coefficient is $\lambda = 0.01$. The balance coefficient of

FRW layer is $\beta = 0.001$, and the constant C is set to 200.

Method	Rank 1	Rank 5	Rank 10
SCSP [2]	53.5	82.6	91.5
LSSCDL [32]	42.7	84.3	91.9
TMA [15]	43.8	-	83.9
ℓ_1 GL [8]	41.5	-	-
Siamese LSTM [22]	42.4	68.7	79.4
Metric Ensemble [16]	45.9	77.5	88.9
DNS [31]	51.2	82.1	90.5
IDLA [1]	34.8	63.6	75.6
DGD [26]	38.6	-	-
MCP-CNN [3]	47.8	74.7	84.8
Gated S-CNN [21]	37.8	66.9	77.4
EDM [17]	40.9	-	-
Joint Learning [23]	35.8	-	-
Deep Transfer [5]*	56.3	-	-
CNN-I	39.1	61.3	70.5
CNN-IV	47.2	72.6	82.3
CNN-IC	49.3	77.3	87
CNN-FC-IC	46.6	74.4	84.3
CNN-FRW-IC	50.4	77.6	85.8

Table 2: Accuracy on VIPeR. *Deep Transfer [5] uses ImageNet for pretraining, while CNN-IV is our implementation of [5] without ImageNet pretraining.

Method	Rank 1	Rank 5	Rank 10
ℓ_1 GL [8]	50.1	-	-
DNS [31]	69.1	86.9	91.8
IDLA [1]	47.5	71.6	80.3
DGD [26]	66.6	-	-
MCP-CNN [3]	53.7	84.3	91
Deep Transfer [5]*	77.0	-	-
CNN-I	63.4	84.4	90.5
CNN-IV	74.4	91.3	95.0
CNN-IC	70.1	90.5	94.8
CNN-FC-IC	66.1	88.2	93
CNN-FRW-IC	70.5	90.0	94.8

Table 3: Accuracy on CUHK01. *Deep Transfer [5] uses ImageNet for pretraining, while CNN-IV is our implementation of [5] without ImageNet pretraining.

4.5. Results on CUHK03

From Table 1, we can see that the model with only identification loss gets the worst accuracy among the baseline models. The accuracy of our implementation of identification loss and verification loss is slightly worse than the one

reported in [5] as they use extra ImageNet data for pretraining. Identification loss with center loss gets the same accuracy as identification loss with verification loss, which verifies the effectiveness of center loss. With the new designed FRW layer, the performance can be further improved. In contrast, the performance drops a little when a naive FC layer is added, which indicates that simply adding more layers does not bring any improvement. Among the models that do not use extra training data, our proposed model CNN-FRW-IC achieves the best rank 1, rank 5 and rank 10 accuracy on CUHK03 (detected).

Method	Rank 1	Rank 5	Rank 10
CNN-IV	77.0	93.1	96.6
CNN-IC	80.7	95.8	97.7

Table 4: Accuracy on CUHK03 (detected) with only 5k training iterations.

4.6. Results on VIPeR and CUHK01

The results of VIPeR and CUHK01 are shown in Table 2 and 3, respectively. Similar to CUHK03, the results of our implementation of identification and verification loss are not as good as [5] due to the lack of ImageNet pretraining. CNN-IC reaches a higher performance than CNN-IV on VIPeR but a worse performance on CUHK01. The model with FRW layer has an improved performance on both the two datasets, and it outperforms most of the existing models. Similarly, adding a FC layer reduces the accuracy.

4.7. Comparison on Convergence Speed

Intuitively, center loss is more efficient than verification loss on training since it constructs batches of single image samples as input instead of person pairs or triplets. We conduct a comparative experiment on the two types of losses to see how their convergence speed differs in practice. We reduce the training iterations to 5k, where the learning rate is decayed by 0.1 at 4k iterations. The other settings remain the same as before. From Table 4, we see that the model of center loss has a better performance than the verification model. It is worth noting that the model accuracy of center loss with 5k training iterations is slightly better than the one with 25k training iterations, indicating that it has converged with 5k iterations. On the contrary, the accuracy of verification model drops when the number of training iterations is reduced. Therefore, center loss does converge faster than verification loss. More importantly, when a larger person re-identification dataset is used, the efficiency gap between the two losses will be more significant.

5. Conclusion

In this paper, we have proposed a novel CNN architecture for person re-identification. The proposed architecture utilizes identification loss and center loss to jointly balance the intra/inter class distances. By using center loss, our model becomes more efficient compared to the one using verification loss. Our model also contains a new FRW layer that learns to reweight the learned embedding for each dimension. Thus, the network gains more freedom to distribute the importance for each dimension. Based on the experimental results on CUHK03, CUHK01 and VIPeR, our proposed CNN outperforms the state-of-the-art competitors in most cases. We also have empirically validated that CNN with identification loss and center loss gets a comparable accuracy to the one with identification loss and verification loss at a faster convergence speed.

References

- [1] E. Ahmed, M. Jones, and T. Marks. An improved deep learning architecture for person re-identification. In *CVPR*, 2015.
- [2] D. Chen, Z. Yuan, B. Chen, and N. Zheng. Similarity learning with spatial constraints for person re-identification. In *CVPR*, 2016.
- [3] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In *CVPR*, 2016.
- [4] S. Ding, L. Lin, G. Wang, and H. Chao. Deep feature learning with relative distance comparison for person re-identification. *Pattern Recognition*, 48(10):2993–3003, 2015.
- [5] M. Geng, Y. Wang, T. Xiang, and Y. Tian. Deep transfer learning for person re-identification. *arXiv preprint arxiv:1611.05244*, 2016.
- [6] D. Gray, S. Brennan, and H. Tao. Evaluating appearance models for recognition, reacquisition, and tracking. In *IEEE International Workshop on PETS*, 2007.
- [7] S. Khamis, C.-H. Kuo, V. Singh, V. Shet, and L. Davis. Joint learning for attribute-consistent person re-identification. In *ECCV*, 2014.
- [8] E. Kodirov, T. Xiang, Z. Fu, and S. Gong. Person re-identification by unsupervised ℓ_1 graph learning. In *ECCV*, 2016.
- [9] W. Li and X. Wang. Locally aligned feature transforms across views. In *CVPR*, 2013.
- [10] W. Li, R. Zhao, T. Xiao, and X. Wang. DeepReID: Deep filter pairing neural network for person re-identification. In *CVPR*, 2014.
- [11] S. Liao, Y. Hu, X. Zhu, and S. Li. Person re-identification by local maximal occurrence representation and metric learning. In *CVPR*, 2015.
- [12] S. Liao and S. Li. Efficient psd constrained asymmetric metric learning for person re-identification. In *ICCV*, 2015.
- [13] H. Liu, J. Feng, M. Qi, J. Jiang, and S. Yan. End-to-end comparative attention networks for person re-identification. In *IEEE Transactions on Image Processing*, 2016.
- [14] B. Ma, Y. Su, and F. Jurie. A novel image representation for person re-identification and face verification. In *BMVC*, 2012.
- [15] N. Martinel, A. Das, C. Micheloni, and A. Roy-Chowdhury. Temporal model adaptation for person re-identification. In *ECCV*, 2016.
- [16] S. Paisitkriangkrai, C. Shen, and A. Hengel. Learning to rank in person re-identification with metric ensembles. In *CVPR*, 2015.
- [17] H. Shi, Y. Yang, X. Zhu, S. Liao, Z. Lei, W. Zheng, and S. Li. Embedding deep metric for person re-identification: A study against large variations. In *ECCV*, 2016.
- [18] H. Shi, X. Zhu, S. Liao, Z. Lei, Y. Yang, and S. Li. Constrained deep metric learning for person re-identification. *CoRR*, 2015.
- [19] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *NIPS*, 2014.
- [20] E. Ustinova, Y. Ganin, and V. Lempitsky. Multiregion bilinear convolutional neural networks for person re-identification. *CoRR*, 2015.
- [21] R. Varior, M. Haloi, and G. Wang. Gated siamese convolutional neural network architecture for human re-identification. In *ECCV*, 2016.
- [22] R. Varior, B. Shuai, J. Lu, D. Xu, and G. Wang. A siamese long short-term memory architecture for human re-identification. In *ECCV*, 2016.
- [23] F. Wang, W. Zuo, L. Lin, D. Zhang, and L. Zhang. Joint learning of single-image and cross-image representations for person re-identification. In *CVPR*, 2016.
- [24] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. Learning fine-grained image similarity with deep ranking. In *CVPR*, 2014.
- [25] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, 2016.
- [26] T. Xiao, H. Li, W. Ouyang, and X. Wang. Learning deep feature representations with domain guided dropout for person re-identification. In *CVPR*, 2016.
- [27] T. Xiao, S. Li, B. Wang, L. Lin, and X. Wang. End-to-end deep learning for person search. *arXiv preprint arxiv:1604.01850*, 2016.
- [28] F. Xiong, M. Gou, O. Camps, and M. Sznai. Person re-identification using kernel-based metric learning methods. In *ECCV*, 2014.
- [29] Y. Yang, J. Yang, J. Yan, S. Liao, D. Yi, and S. Li. Salient color names for person re-identification. In *ECCV*, 2014.
- [30] D. Yi, Z. Lei, S. Liao, and S. Li. Deep metric learning for person re-identification. In *ICPR*, 2014.
- [31] L. Zhang, T. Xiang, and S. Gong. Learning a discriminative null space for person re-identification. In *CVPR*, 2016.
- [32] Y. Zhang, B. Li, H. Lu, A. Irie, and X. Ruan. Sample-specific svm learning for person re-identification. In *CVPR*, 2016.
- [33] R. Zhao, W. Ouyang, and X. Wang. Learning mid-level filters for person re-identification. In *CVPR*, 2014.
- [34] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. Scalable person re-identification: A benchmark. In *ICCV*, 2015.

- [35] L. Zheng, H. Zhang, S. Sun, M. Chandraker, Y. Yang, and Q. Tian. Person re-identification in the wild. *arXiv preprint arXiv:1604.02531*, 2016.
- [36] Z. Zheng, L. Zheng, and Y. Yang. A discriminatively learned cnn embedding for person re-identification. *arXiv preprint arXiv:1611.05666*, 2016.