Supervised Deep Feature Embedding With Handcrafted Feature

Shichao Kan, Yigang Cen, *Member, IEEE*, Zhihai He, *Fellow, IEEE*, Zhi Zhang, Linna Zhang, and Yanhong Wang

***Abstract*— Image representation methods based on deep convolutional neural networks (CNNs) have achieved the state-of-the-art performance in various computer vision tasks, such as image retrieval and person re-identification. We recognize that more discriminative feature embeddings can be learned with supervised deep metric learning and handcrafted features for image retrieval and similar applications. In this paper, we propose a new supervised deep feature embedding with a handcrafted feature model. To fuse handcrafted feature information into CNNs and realize feature embeddings, a general fusion unit is proposed (called *Fusion-Net*). We also define a network loss function with image label information to realize supervised deep metric learning. Our extensive experimental results on the Stanford online products' data set and the in-shop clothes retrieval data set demonstrate that our proposed methods outperform the existing state-of-the-art methods of image retrieval by a large margin. Moreover, we also explore the applications of the proposed methods in person re-identification and vehicle re-identification; the experimental results demonstrate both the effectiveness and efficiency of the proposed methods.**

***Index Terms*— Deep feature embedding, handcrafted feature, image representation, deep metric learning, image retrieval, person re-identification, vehicle re-identification.**

I. Introduction

LEARNING discriminative feature embeddings is an important task in computer vision. Image features obtained from deep convolutional neural networks (DCNNs) have achieved state-of-the-art performance in image classification [1]–[3] and image retrieval [4]–[6] tasks. Unlike the image classification task that aims to determine the classification of hyperplanes in the feature space, the image retrieval task minimizes the intra-class distance of similar images and maximizes the inter-class distance of dissimilar images. Various deep feature embedding methods based on metric learning [7] have been proposed to improve the performance of image retrieval, including: (1) Deep metric learning by constructing different metric loss functions, e.g., contrastive loss [8], triplet loss [9], lifted structured loss [10], histogram loss [11], facility location [12], global loss [13], radial basis function [14] and position-dependent deep metric [15]; (2) combining multiple loss functions, e.g., jointly optimizing contrastive loss and softmax loss [16], jointly optimizing triplet loss and softmax loss [17], [18], or combining global and triplet loss [19]; and (3) using hard negative/positive sample mining [19].

There are two major advantages in using metric loss as the network loss function. First, because the parameters of the deep feature embedding layer are optimized by the metric (Euclidean or Cosine distances) used for image retrieval, the embedded feature obtained by metric loss is more robust than that by the softmax loss for image retrieval. Second, the convergence rate with metric loss is faster than softmax loss during the network training stage. However, because the embedded features are used to compute the metric loss, it might suffer from over-fitting. Recently, to embed more information into deep features, methods combining multiple losses have been developed [16]–[18], [20]–[22]. Compared to those methods that optimize the metric loss or the softmax loss, they can boost the performance of image retrieval. However, it is difficult to determine the optimal weight for each loss function.

After studying the literature, we find that most state-of-the-art deep feature embedding models are semi-supervised learning, and they are only used for convolutional neural networks or handcrafted features, respectively. Typically, the semi-supervised deep feature embedding models only need similar and dissimilar pairs of data sets. However, more and more data are labeled with the development of supervised learning. Thus, it is best to consider the labels' information of the data in the deep feature embedding models. Moreover, in the feature fusion, some works [23]–[25] demonstrate that fusing the deep feature and handcrafted feature is an effective method for image-based applications, and these two types of features are complementary. We recognize that the handcrafted feature can boost the robustness of CNNs if its information can be merged into the network to participate in the training process.

Manuscript received August 17, 2018; revised January 14, 2019 and February 5, 2019; accepted February 19, 2019. Date of publication February 25, 2019; date of current version August 30, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61872034 and Grant 61572067, in part by the Natural Science Foundation of Guizhou Province under Grant [2019]1064, in part by the Science and Technology Program of Guangzhou under grant 201804010271, in part by the Fundamental Research Funds for the Central Universities under Grant 2018YJS046. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Xiaochun Cao. *(Corresponding author: Yigang Cen.)*

S. Kan, Y. Cen, and Y. Wang are with the School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China, and also with the Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing 100044, China (e-mail: 16112062@bjtu.edu.cn; ygcen@bjtu.edu.cn; wangyanhong@bjtu.edu.cn).

Z. He and Z. Zhang are with the Department of Electrical Engineering and Computer Science, University of Missouri, Columbia, MO 65211 USA (e-mail: hezhi@missouri.edu; zzbhf@mail.missouri.edu).

L. Zhang is with the College of Mechanical Engineering, Guizhou University, Guiyang 550025, China (e-mail: zln19770808@163.com).

Digital Object Identifier 10.1109/TIP.2019.2901407

Fig. 1. (a). The previous commonly used deep feature embedding model. (b). Our proposed supervised deep feature embedding with handcrafted feature model.

Some works have been proposed to address the problem of how to fuse multiple CNN features or handcrafted features [26]–[30]. In order to classify indoor scenes, Li *et al.* [26] proposed a method to fuse RGB and depth information based on the CNN. To detect iris presentation attack, Yadav et al. [27] proposed a method to fuse handcrafted features and VGG feature. Xiong *et al.* [28] proposed a method to fuse multiple CNN features extracted from local image patches to compose the image feature. Although the performance improved after feature fusion, there are no feature embedding in these methods. From a different point of view, Sun *et al.* [29] studied the theory of feature fusion, and feature transform was proposed in their method. However, this theory is not based on the CNN framework. Moreover, Akilan *et al.* [30] proposed a method to fuse multiple CNN features. In their method, multi-model CNN features were first extracted, and then the PCA transformation was performed on these features, respectively. Finally, these transformed features were fused and a classifier was trained. This method is started from the perspective of extracting CNN features and then fusing them. Because each part of this method is individually trained, thus it cannot benefit from end-to-end learning.

In this paper, we study how to combine image labels and handcrafted features into the deep feature embedding model based on theoretical analysis. The proposed method can effectively improve the robustness of feature embedding under the supervision of image labels and the information merger of handcrafted features. Thus, the overall image retrieval performance can be significantly improved.

We have obtained state-of-the-art results on the Stanford Online Products' data set and the In-shop Clothes Retrieval data set for general image retrieval. The performance improvement is primarily from the combination of the softmax loss and our proposed class-metric loss, as well as the embedding of merging handcrafted features. In addition, experiments are conducted on the Market-1501 [31] data set and the MARS [32] data set for person re-identification (re-ID) [33], and the VeRi-776 [34] data set for vehicle re-identification (re-ID). A variety of experiments demonstrate that our proposed methods have a wide range of applicability.

Our works have the following three major contributions.

- First, we propose a new supervised deep feature embedding with handcrafted feature model. In this model, a general fusion unit (*Fusion-Net* in Fig. 1(b)) is proposed to fuse handcrafted feature information into CNNs. For forward propagation, both CNN and handcrafted representations can be embedded directly into the final representation vector. For back propagation, the handcrafted feature information can be back propagated to CNN and participate in the parameters update of CNN.
- Second, in order to embed label information into the feature embedding, a new loss function combining the distance metric with the label information is proposed. In the proposed loss function, the sample's label information is indirectly (softmax loss) and directly (class-metric loss) embedded into the final feature embeddings at the training stage. Therefore, the ability of the final feature embedding is improved.
- Third, a variety of experiments are conducted, including the applications of image retrieval, person re-ID and vehicle re-ID. We obtain the state-of-the-art feature embedding for general image retrieval and vehicle re-ID based on the GoogLeNet [3] and 4-RootHSV [35] feature.

The rest of this paper is organized as follows. In Section II, related works about deep metric learning and multi-loss function optimization are reviewed. The idea and details of the proposed supervised deep metric learning with handcrafted feature are presented in Section III. In Section IV, algorithm implementation details, data sets, evaluation metrics, and the experimental results are presented. Section V concludes the paper.

II. Related Work

Works related to our method mainly include the following two aspects: (1) deep feature embedding with deep metric learning; (2) multi-loss function optimization.

A. Deep Feature Embedding With Deep Metric Learning

The purpose of deep metric learning is to train a matrix based on deep learning methods that can transform the input data into a low dimensional space, such that the transformed result is most suitable for the metric used for supervised learning. The original idea was proposed by Bromley *et al.* [36]. In their work, they trained a Siamese network for signature verification. Then, Chopra *et al.* [37] trained a similarity metric discriminatively for face verification. They try to minimize a discriminative loss function (contrastive loss [8], [36]) that makes the similarity metric small for faces within a same class and large for faces from different classes.

During the past few years, instead of using the contrastive loss function [8], [36], which uses the paired data $\{(\mathbf{x}_i, \mathbf{x}_j, y_{ij})\}$ ($\mathbf{x}_i \in \mathbb{R}^m$ and $\mathbf{x}_j \in \mathbb{R}^m$ represent column vectors, and $y_{ij} \in \{0, 1\}$ denotes dissimilar and similar, respectively) to train a feature embedding, the triplet loss function [9], [38], [39] is widely used to train deep feature embedding because it uses more informative triplet data $\{(\mathbf{x}_a^{(i)}, \mathbf{x}_p^{(i)}, \mathbf{x}_n^{(i)})\}$ (the dimension of these column vectors also belongs to $\mathbb{R}^m$), where $(\mathbf{x}_a^{(i)}, \mathbf{x}_p^{(i)})$ are selected from one class and $(\mathbf{x}_a^{(i)}, \mathbf{x}_n^{(i)})$ are selected from different classes. Based on these methods, Song *et al.* [10] uses all positive pairs and all negatives pairs of samples in a mini-batch and proposes a lifted structured loss function. Ustinova and Lempitsky [11] proposed a histogram loss function based on estimating two distributions of similarities for matching and non-matching sample pairs. Considering the global structure of the embedding space, Song *et al.* [12] proposed a facility location optimization method to optimize a clustering quality metric of normalized mutual information (NMI) [40]. Kumar *et al.* [13] proposed a global loss by minimizing the variance of distributions in matching and non-matching pairs. In addition, it minimizes the mean value of the distance values between matched pairs, while maximizing the mean value of the distances between non-matched pairs. Based on a radial basis function (RBF), Meyer *et al.* [14] proposed the nearest neighbor RBF solver to optimize the deep neural networks. To learn a similarity metric that adapts to a local structure, Huang *et al.* [15] proposed a position-dependent deep metric (PDDM).

All these methods are only distance-based, and there is no classification probability participated in the metric computation stage. In our work, we propose a class-metric loss by combining distances and classification probabilities of a batch samples. Similar with the previous works, in the proposed class-metric loss, the similarities of samples are also computed based on the distances of the corresponding feature embeddings. The purpose is also to minimize the intra-class distance of similar images and maximize the inter-class distance of dissimilar images.

B. Multi-Loss Function Optimization

Jointly training convolution neural networks with different loss functions is a very effective approach to improving network performance. The general formulation of the multi-loss function is usually defined as:

$$L = \alpha L_1 + \delta L_2 \tag{1}$$

where α and δ are the weights of loss L_1 and L_2, respectively. In deep feature embedding, $\alpha + \delta = 1$, and one of the losses is the softmax loss; another is a metric loss. The softmax loss contains label information, and the metric loss contains structural information. The methods in [16], [20], and [21] jointly optimized the contrastive loss and softmax loss, and the methods in [17], [18], and [22] jointly optimized the triplet loss and softmax loss. References [13] and [19] adopted a different approach that combined a global and a triplet loss to train the network.

When compute the loss value of the multi-loss function, only two or three examples are used based on the contrastive loss or the triplet loss. Thus, the examples of a batch size in the network training stage can not be fully used. In our work, we propose jointly optimizing the softmax loss and our proposed class-metric loss based on all the examples of a batch size.

III. Supervised Deep Feature Embedding With Handcrafted Feature

As shown in Fig. 1(a), the previous deep feature embedding model based on only CNNs, and a metric loss or metric combined with softmax loss (Eq.(1)) was used. We recognize that the parameter update of CNNs during back propagation and deep feature embedding can benefit from handcrafted features. Based on this idea, a supervised deep feature embedding with the handcrafted feature model is proposed in this paper. As shown in Fig. 1(b), the handcrafted feature is merged by the unit of *Fusion-Net*, and a new loss function (class-metric loss) is proposed to train CNN. Next, we will describe our model in detail from a theoretical perspective.

A. Unconstrained Metric Learning

Metric learning is a popular research area in machine learning. Given two examples $\{(\mathbf{x}_i, \mathbf{x}_j)\}$, a general Euclidean distance after transform $\phi(\cdot)$ can be defined as:

$$\mathcal{D}_\phi = ||\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)||_2. \tag{2}$$

Based on this definition, the popular unconstrained metric loss function (e.g., LMNN [38]) can be rewritten as:

$$\mathcal{L}(\phi) = \sum_{(i,j)\in\mathcal{P}} ||\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)||_2^2 + \gamma \sum_{(i,j)\in\mathcal{P},(i,k)\in\mathcal{N}} \times [1 + ||\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)||_2^2 - ||\phi(\mathbf{x}_i) - \phi(\mathbf{x}_k)||_2^2]_+ \tag{3}$$

In Eq.(3), $\mathcal{P}$ denotes a positive pair set, i.e., $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same category in a data set. $\mathcal{N}$ denotes a negative pair set, i.e., $\mathbf{x}_i$ and $\mathbf{x}_k$ belong to different categories in a data set. The symbol $[\cdot]_+$ indicates the hinge-loss $[\cdot]_+ = max(0, \cdot)$. Most metric learning tasks can be generalized to minimize Eq.(3).

We first analyze the $\phi(\mathbf{x})$ function in Eq.(2) using the Mahalanobis distance (the motivation of using this distance is analyzed in Section III-B). For a given matrix $\mathbf{G}$, the square Mahalanobis distance is defined as:

$$\mathcal{D}_\mathbf{G}^2 = (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{G} (\mathbf{x}_i - \mathbf{x}_j). \tag{4}$$

With singular value decomposition of the Mahalanobis matrix $\mathbf{G}$, we have $\mathbf{G} = \mathbf{H}\Sigma\mathbf{H}^T$ (here, $\mathbf{G}$ is a Positive Semi Definite (PSD) matrix, but some algorithms do not have to constrain the matrix $\mathbf{G}$ to be a PSD matrix, e.g., deep metric learning algorithms), where $\mathbf{H}$ is an orthogonal matrix satisfying $\mathbf{H}\mathbf{H}^T = \mathbf{I}$, Σ is a diagonal matrix containing all the eigenvalues. So, Eq.(4) can be rewritten as:

$$\begin{aligned} \mathcal{D}_\mathbf{G}^2 &= (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{H}\Sigma\mathbf{H}^T (\mathbf{x}_i - \mathbf{x}_j) \\ &= (\mathbf{H}^T\mathbf{x}_i - \mathbf{H}^T\mathbf{x}_j)^T \Sigma (\mathbf{H}^T\mathbf{x}_i - \mathbf{H}^T\mathbf{x}_j) \end{aligned} \tag{5}$$

Comparing Eq.(2) and Eq.(5), the transform function $\phi(\mathbf{x})$ in Eq.(2) can be defined as $\phi(\mathbf{x}) = \mathbf{x}^T\mathbf{M}$, and $\mathbf{M}$ can be used to approximate $\mathbf{G}$ and $\mathbf{M}\mathbf{M}^T = \mathbf{G}$. If $\mathbf{x} \in \mathbb{R}^m$, then $\mathbf{M} \in \mathbb{R}^{m\times d}$. In the past few decades, the handcrafted features of image have been commonly used as $\mathbf{x}$. However, in recent years, the deeply learned features of an image are the most commonly used features. Thus, we consider incorporating handcrafted features into the deep feature embedding model and believe that handcrafted features can enhance the discriminative power of deep feature embedding.

Motivated by this observation, in this paper, the handcrafted feature is merged into the deep feature embeddings according to the following definitions 1 and 2.

Definition 1: For representations $\boldsymbol{x}(1), \boldsymbol{x}(2), \cdots, \boldsymbol{x}(k)$ of one object, we define a transformation of these representations as:

$$F(\boldsymbol{x}(1), \boldsymbol{x}(2), \cdots, \boldsymbol{x}(k)) = [\boldsymbol{x}(1)^T\boldsymbol{W}_1, \boldsymbol{x}(2)^T\boldsymbol{W}_2, \cdots, \boldsymbol{x}(k)^T\boldsymbol{W}_k]. \quad (6)$$

where the symbol $[\boldsymbol{x}, \boldsymbol{y}]$ represents the concatenation of vector $\boldsymbol{x}$ and $\boldsymbol{y}$. Here, we introduce a transformation function $F(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{W} \in \mathbb{R}^{m_1}$ to regulate the input data, and $\boldsymbol{W} \in \mathbb{R}^{m\times m_1}$ is restricted to being a low-rank matrix. $F(\boldsymbol{x})$ is defined as the ***converter*** *of $\boldsymbol{x}$.*

Definition 2: For representations $\boldsymbol{x}(1), \boldsymbol{x}(2), \cdots, \boldsymbol{x}(k)$ of one object, we define another transformation of these representations as:

$$\phi(\boldsymbol{x}(1), \boldsymbol{x}(2), \cdots, \boldsymbol{x}(k)) = F(\boldsymbol{x}(1), \boldsymbol{x}(2), \cdots, \boldsymbol{x}(k))^T\boldsymbol{M} \quad (7)$$

where $\boldsymbol{M} \in \mathbb{R}^{m_1\times d}$ is an embedding matrix, and we also limit $\boldsymbol{M}$ to be low rank for compressed representation of high-dimensional concatenated representations. Furthermore, the rank of $\boldsymbol{M}$ must be lower than the rank of $\boldsymbol{W}$ to obtain a low-dimensional feature embedding. $\phi(\cdot)$ is defined as a ***merger*** *of multiple transformed features.*

Because we only merge one handcrafted feature into the deep feature embeddings, in this paper, we only consider $k = 2$ (i.e., deeply learned feature and handcrafted feature). Based on definitions 1 and 2, Eq.(2) can be rewritten as:

$$\mathcal{D}_{\mathbf{W}_1,\mathbf{W}_2,\mathbf{M}} = ||[\mathbf{x}_i(1)^T\mathbf{W}_1, \mathbf{x}_i(2)^T\mathbf{W}_2]^T\mathbf{M} - [\mathbf{x}_j(1)^T\mathbf{W}_1, \mathbf{x}_j(2)^T\mathbf{W}_2]^T\mathbf{M}||_2 \quad (8)$$

Based on Eq.(8), iterative methods can be used to find the values of $\mathbf{W}_1$, $\mathbf{W}_2$ and $\mathbf{M}$ (e.g., multi-layer neural networks or LogDet divergence [41]) by minimizing Eq.(3). In this paper, we consider using a multi-layer neural network to solve this problem for the following reasons: (1) the parameters of CNN, converters and the merger can be learned in an end-to-end manner through iterative methods; (2) the robustness of parameter estimation of the converters and merger can be enhanced using the mini-batch-based stochastic gradient descent (SGD) [42] method; (3) because the matrix $\mathbf{G}$ in Eq.(4) is learned by $\mathbf{M}$ and $\mathbf{W}$ as suggested in definitions 1 and 2 with SGD, we do not need to constrain the matrix $\mathbf{G}$ to being a PSD matrix; (4) the CNN can be tuned during the learning stage, and handcrafted feature information can be back propagated to CNN, which will enhance the robustness of CNN for this optimization task.

Based on the above analysis, the overall system is shown in Fig. 1(b). It consists of deeply learned representation $\mathbf{x}(1)$ and handcrafted representation $\mathbf{x}(2)$ of the input image, feature converters ($F(\mathbf{x}(1), \mathbf{x}(2))$ with unknown parameters $\mathbf{W}_1$ and $\mathbf{W}_2$), a merger network ($\phi(\mathbf{x})$ with unknown parameters $\mathbf{M}$), and multi-loss functions, which will be discussed in Section III-D.

B. The Analysis of Mahalanobis Distance-Based Metric Learning for Our Model

Considering the square of Eq.(8), we divide the matrix $\mathbf{M} \in \mathbb{R}^{m_1\times d}$ into four sub-matrices $\mathbf{M}_{11} \in \mathbb{R}^{m_1^{(1)}\times d^{(1)}}$, $\mathbf{M}_{12} \in \mathbb{R}^{m_1^{(1)}\times(d-d^{(1)})}$, $\mathbf{M}_{21} \in \mathbb{R}^{(m_1-m_1^{(1)})\times d^{(1)}}$ and $\mathbf{M}_{22} \in \mathbb{R}^{(m_1-m_1^{(1)})\times(d-d^{(1)})}$ as follows:

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix} \quad (9)$$

then we have:

$$\begin{aligned} &\mathcal{D}^2_{\mathbf{W}_1,\mathbf{W}_2,\mathbf{M}} \\ &= [(\mathbf{x}_i(1) - \mathbf{x}_j(1))^T\mathbf{W}_1, (\mathbf{x}_i(2) - \mathbf{x}_j(2))^T\mathbf{W}_2]\mathbf{M} \\ &\quad \times[(\mathbf{x}_i(1) - \mathbf{x}_j(1))^T\mathbf{W}_1, (\mathbf{x}_i(2) - \mathbf{x}_j(2))^T\mathbf{W}_2]^T \\ &= [(\mathbf{x}_i(1) - \mathbf{x}_j(1))^T\mathbf{W}_1]\mathbf{M}_{11}[(\mathbf{x}_i(1) - \mathbf{x}_j(1))^T\mathbf{W}_1]^T \\ &\quad + [(\mathbf{x}_i(2) - \mathbf{x}_j(2))^T\mathbf{W}_2]\mathbf{M}_{22}[(\mathbf{x}_i(2) - \mathbf{x}_j(2))^T\mathbf{W}_2]^T \\ &\quad + [(\mathbf{x}_i(1) - \mathbf{x}_j(1))^T\mathbf{W}_1]\mathbf{M}_{12}[(\mathbf{x}_i(2) - \mathbf{x}_j(2))^T\mathbf{W}_2]^T \\ &\quad + [(\mathbf{x}_i(2) - \mathbf{x}_j(2))^T\mathbf{W}_2]\mathbf{M}_{21}[(\mathbf{x}_i(1) - \mathbf{x}_j(1))^T\mathbf{W}_1]^T \end{aligned} \quad (10)$$

From Eq.(10), we can see that the matrix $\mathbf{M}$ is not only the metric of individual features but is also the metric of mutual features. We therefore chose the Mahalanobis distance in our model. It is different from other metric learning based on individual features.

C. Representations and Converters

In the deep feature embedding area, The GoogLeNet [3] is a commonly used CNN. To facilitate the comparison of experimental results, this paper is also based on this network. For handcrafted features, global representations such as 4-RootHSV [35] and local representations such as VLAD [43] are commonly used features. Generally speaking, global representations are more suitable for general image retrieval. However, for fine-grained image retrieval [5], local representations will be more suitable than global representations. This paper mainly focuses on the task of general image retrieval. Because color-based handcrafted features and CNN features have been experimentally proved to be heterogeneous [23], the color-based 4-RootHSV [35] is used as the handcrafted feature in our proposed model. Fig. 2 shows an example of the proposed model.

For the Fusion-Net unit in the proposed model, the **converter** is an important component. As **Definition** 1, each input representation has its own converter corresponding to the $\mathbf{W}_1$ and $\mathbf{W}_2$ in the Eq.(8). They are marked with red rectangles in Fig. 2, lying between each input representation and feature

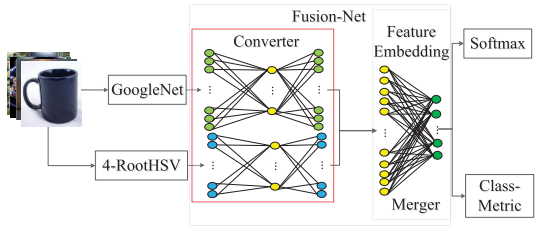


Fig. 2. An example of the proposed model: the GoogLeNet is used as the CNN and the 4-RootHSV is used as the handcrafted feature. The converter is set as an auto-encoder [44]. The merger module is a feature embedding layer. The class-metric loss is the proposed loss function.

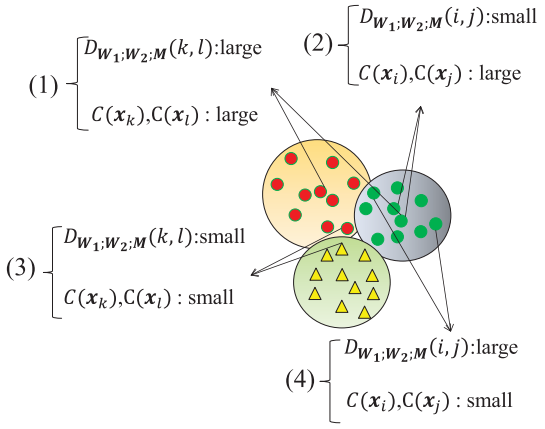


Fig. 3. Distance and classification relationship between different categories of samples (different colors in the figure indicate different categories).

embedding layer. We believe that a good converter must have the following advantages: (1) It can transform the input image representations into a relatively consistent space, which can be considered as a normalization mechanism. (2) It is able to suppress useless information in each representation for feature embedding. (3) It can extract useful information, particularly complementary information from different representations. In this work, we use the following three methods (the final performances will be verified by experiments) to obtain the parameters of the converters.

- **Extreme learning machine (ELM)** [45], [46]. It is a fully connected network. The parameters of ELM are randomly initialized and will not be updated at the stage of training or fine-tuning. It is a randomly transformed converter for an input vector. Thus, ELM can be seen as a normalization converter. Since ELM does not perform the parameter update process, it cannot learn useful information regarding the input representation for feature embedding. In addition, it cannot suppress useless information in each representation.
- **Auto-encoder** [44]. It is proposed to achieve dimensional reduction and preserve as much original information as possible. As shown in Fig. 2, the number of output nodes equal the number of input nodes in the auto-encoder. The auto-encoder can be solved by the mean square error (MSE). As a converter, the auto-encoder can be used to normalize the input representation and suppress useless information. However, because auto-encoder mainly trains according to its own information, it cannot learn complementary information with other representations.
- **Fully connected (fc) network**. By training or fine-tuning the parameters of fc, it can normalize the input representation, learn complementary information, and suppress information that may result in performance degradation after feature embedding.

Based on the solution of deep learning, the fully connected converter can be set as a non-linear function with non-linear activation in the rectified linear units (ReLUs) [47] as Eq.(11).

$$F(\mathbf{x}, \mathbf{W}, b) = max(\mathbf{0}, \mathbf{x}^T\mathbf{W} + b) \tag{11}$$

where $\mathbf{x}$ is a column vector and $\mathbf{x} \in \mathbb{R}^m$. The value of m is different for different representations. $\mathbf{W}$ is a matrix and $\mathbf{W} \in \mathbb{R}^{m \times m_1}$. b is a scalar, $\mathbf{x}^T$ denotes the transposition of $\mathbf{x}$, and $max(\mathbf{0}, \mathbf{x})$ is the activation function of rectified linear units (ReLUs) [47]. According to the two input image representations $\mathbf{x}(1)$ and $\mathbf{x}(2)$, the function of the converter can be written as $F(\mathbf{x}(j), \mathbf{W}_j, b_j)$, where $j \in \{1, 2\}$. Then, our goal is to solve the parameters of F, namely, finding $\mathbf{W}_j$ and b_j.

D. Supervised Deep Feature Embedding

According to the analysis of Section III-A, feature embedding can be realized by solving the matrix of $\mathbf{M}$ in function $\phi(\mathbf{x}) = F(\mathbf{x})^T\mathbf{M}$. For low-dimensional feature embedding, we can use the constraint that $\mathbf{M}$ is a low-rank matrix. If the output of converter $F(\mathbf{x}) \in \mathbb{R}^{m_1}$ and $\mathbf{M} \in \mathbb{R}^{m_1 \times d}$, then we let $d < m_1$. In our experiments, d is the size of feature embedding, which is set as 64, 128, 256 and 512, respectively. After the parameters of $\mathbf{M}$ are solved according to the constraint in Eq.(3), we can obtain a metric of the concatenated features (the square of Eq.(8)).

However, there are many problems with minimizing Eq.(3) directly through SGD and back propagation in deep learning. The most common issue is that the loss calculated by Eq.(3) may be very large for some mini-batches. It will cause the gradient explosion problem during the optimization process. There are several loss functions that are similar to Eq.(3) in deep feature embedding [10], [48], for example, lifted structured loss for feature embedding, which can be used as a metric loss function instead of Eq.(3).

Still, Eq.(3) or other metric loss only considers the metric information. In supervised learning tasks, the image label is a very important supervisory information. This information can be indirectly embedded into the feature embeddings if the softmax loss is used; a softmax classifier can be defined by a function $C(\mathbf{x}) = \phi(\mathbf{x})^T\mathbf{P}$ before normalization (the normalized function is $C(\mathbf{x}_i) = exp^{C(\mathbf{x}_i)} / \sum_i exp^{C(\mathbf{x}_i)}$). The unknown matrix $\mathbf{P}$ can be solved using back propagation and SGD according to the input vectors and corresponding labels in the deep learning framework. During back propagation, the information related with $\mathbf{P}$ can be back propagated to $\phi(\mathbf{x})$ and thus participates in optimizing $\phi(\mathbf{x})$ (matrix $\mathbf{M}$). Thus, the final feature embeddings also contain the label information.

Based on the above analysis, a metric loss only related to the distance between features and the softmax loss is only related to the category of features. During training, they are used to alternately update model parameters. As shown in the Fig. 3, samples at the edge of a hyperplane may have a low probability of belonging to the right category (e.g., notations (3) and (4) in the Fig. 3), where their inter-class distance may be small (e.g., notation (3) in the Fig. 3), and the intra-class distances may be large (e.g., notation (4) in the Fig. 3). In deep feature embedding, to speed up the convergence of the network at the training stage, for the cases of notations (1) and (2) in the Fig. 3, we would like to produce a small loss. For the cases of notations (3) and (4), we expect to produce a large loss. Thus, a new loss function (called class-metric loss ($L_{class-metric}$)) combined metric with class information is designed as follows:

$$\begin{aligned}\widetilde{Q}_{ij} &= \log\left\{\sum_{\substack{k\in\{i,j\},\\(k,l)\in\mathcal{N}}}[1+\frac{(p_k+p_l)}{2}]exp[e-\mathcal{D}_{\mathbf{W}_1,\mathbf{W}_2,\mathbf{M}}(k,l)]\right\}\\&\quad+[1+\frac{(p_i+p_j)}{2}]\mathcal{D}_{\mathbf{W}_1,\mathbf{W}_2,\mathbf{M}}(i,j)\\Q &= \frac{1}{2|\mathcal{P}|}\sum_{(i,j)\in\mathcal{P}} max(0,\widetilde{Q}_{ij})^2,\end{aligned}\tag{12}$$

Consistent with the notations of Eq.(3), in Eq.(12), $\mathcal{N}$ denotes the negative pair set in a mini-batch, and $\mathcal{P}$ denotes the positive pair set in a mini-batch. e is a margin parameter. $|\mathcal{P}|$ is the number of positive pairs in a mini-batch. $p_s(s\in\{i,j,k,l\})$ is defined as:

$$p_s = 1-\frac{exp^{C(\mathbf{x}_s)+\epsilon}}{\sum_r exp^{C(\mathbf{x}_r)+\epsilon}}.\tag{13}$$

In Eq.(13), $C(\mathbf{x}) = \phi(\mathbf{x})^T\mathbf{P}$, r is the image index in a mini-batch, ϵ is used to avoid over-flow. Eq.(13) is the residual of ground-truth probability with the corresponding output of softmax classifier for the image I_s. In Eq.(12), p_s can be seen as an adaptive distance weighting strategy. The back propagation gradients of Eq.(12) for the input feature embeddings can be found in Appendix.

In addition, we construct a multi-loss function by combining class-metric and softmax losses. The loss function is defined as:

$$L = \beta[\alpha L_{class-metric}+(1-\alpha)L_{softmax}]\tag{14}$$

where α is the weight of the class-metric loss, which is used to control the proportion of $L_{class-metric}$ and $L_{softmax}$ values. β is the weight of data set, which is used to scale the value of $L_{class-metric}$ and $L_{softmax}$ losses simultaneously, thus expanding or shrinking the residual of the two loss values. Compared with Eq.(1), one unique point of our multi-loss is that we consider the data set properties using the hyper-parameter β. For different types of data sets, β will affect the performance (as shown in Section IV-B). In our experiments, the notation of softmax+metric-loss is used to indicate the proposed multi-loss (Eq.(14)).

E. The Time Complexity Analysis of Class-Metric Loss

We begin to analyze the approximate time complexity of the class-metric loss start from the unit of Fusion-Net based on a mini-batch (suppose the image number of a mini-batch is n_m). First, for the converters, suppose that the inputs $\mathbf{x}(1)\in\mathbb{R}^{m^{(1)}}$ and $\mathbf{x}(2)\in\mathbb{R}^{m^{(2)}}$, the weights $\mathbf{W}_1\in\mathbb{R}^{m^{(1)}\times m_1^{(1)}}$ and $\mathbf{W}_2\in\mathbb{R}^{m^{(2)}\times m_1^{(2)}}$. We have that the approximate time complexity of converters is $T_1\approx O(n_m\times(m^{(1)}\times m_1^{(1)}+m^{(2)}\times m_1^{(2)}))\approx O(max(n_m\times m^{(1)}\times m_1^{(1)}, n_m\times m^{(2)}\times m_1^{(2)}))$. Second, for the merger, suppose that the embedding matrix $\mathbf{M}\in\mathbb{R}^{(m_1^{(1)}+m_1^{(2)})\times d}$. We have that the approximate time complexity of merger is $T_2\approx O(n_m\times(m_1^{(1)}+m_1^{(2)})\times d)$. Third, because the dimension of feature embedding is d, the approximate time complexity of Eq.(2) is $T_3\approx O(n_m\times d)$. Fourth, for the classifier, suppose the number of classes is c, for $C(\mathbf{x})=\phi(\mathbf{x})^T\mathbf{P}$, the approximate time complexity is $T_4\approx O(n_m\times d\times c)$. Five, for Eq.(12), the approximate time complexity is $T_5\approx O(|\mathcal{N}|+|\mathcal{P}|)$, and $|\mathcal{N}|+|\mathcal{P}|$ is the number of negative and positive pairs of a mini-batch. Thus, if the number of data set examples is n_d, the approximate time complexity of class-metric loss is $T\approx O(\frac{n_d}{n_m}\times max(T_1,T_2,T_3,T_4,T_5))$.

F. Model Training

The model is trained using back propagation and stochastic gradient descent (SGD) [42] with a Nesterov momentum of 0.9. We use the Caffe [49] framework for training and testing the proposed methods with and without merging handcrafted feature. The network hyper-parameter settings are chosen based on the following criteria unless otherwise specified. The maximum training iteration is set to be 20,000. The batch size is set as 128, and the initial global learning rate is set to be 0.0001. The margin parameter e in Eq.(12) is set as 1.0. The value of ϵ in Eq.(13) is set as 0.0001. Following existing methods, we also normalize the training and testing images to 256 by 256. For the converter and merger layer, we multiply the local learning rate by 10. We multiply the loss of auto-encoder by 0.01 if it is applied to the converters. For α and β in Eq.(14), we explain how to choose their values for different data sets in Section IV-B. All parameters of convolutional layers of GoogLeNet are initialized from the network pre-trained on the ImageNet ILSVRC [50] data set and fine tuned in the training stage, the parameters of converter and merger are initialized with random weights. It should be noted that before the training or fine-tuning (for GoogLeNet) step, the 4-RootHSV [35] features are computed in advance for speeding up training.

IV. Experimental Results

We conduct experiments on the CARS196 data set [51] (to verify the hyper-parameter of β), the Stanford Online Products data set [10], and the In-shop Clothes Retrieval data set [52] for image retrieval and image clustering. For the person re-ID task, experiments were conducted on the Market-1501 data set [31] and the MARS data set [32]. For the vehicle re-ID task, we use the VeRi-776 data set [34].

The **CARS196** data set has 196 classes with 16,185 images. We split the first 98 classes with 8,054 images for training and the remaining 98 classes with 8,131 images for testing.

The **Stanford Online Products** data set has 22,634 categories with 120,053 images. We split the first 11,318 classes with 59,551 images for training, and the remaining 11,316 classes with 60,502 images for testing.

The **In-shop Clothes Retrieval** data set contains 7,982 classes of clothes with 52,712 images. We use the first 3,997 classes with 25,882 images for training and the remaining 3,985 classes with 26830 images for testing. The test sets are split into the query set and gallery image set, and a successful retrieval is counted when the gallery image belongs to the same class as the query image.

The **Market-1501** data set contains 1501 persons with 32668 labeled bounding boxes, and it is currently the largest image-based person re-ID data set. Following the split method in [31], the training set has 12,936 images with 751 persons, and the testing set has 19,732 images with 750 persons. The prob set contains 3,368 hand-drawn images with 750 persons selected from the testing set. Based on the GoogLeNet, only the single-query evaluation results are reported for this data set in this paper.

The **MARS** data set is by far the largest video-based person re-ID data set. It contains 1261 persons with 1,191,003 images collected from 6 different cameras. As defined by [32], the training set has 509,914 images with 625 persons, and the testing set has 681,089 images with 636 persons. Unlike other data sets, this data set is based on video sequences. The person re-ID task is not a frame-to-frame query on this data set, but a tracklet-to-tracklet query (namely, feature embeddings are pooled across a tracklet).

The **VeRi-776** data set consists of 776 vehicles with over 50,000 images. Following the split of [34], the training set has 37,781 images with 576 vehicles, and the testing set has 11,579 images with 200 vehicles. The prob set contains 1,678 query images selected from the testing set.

A. Performance Evaluation Metrics

For image retrieval, we use the standard mean average precision (mAP) [53] and Recall@K [54] metrics to evaluate the performance of various algorithms. The Recall@K first computes K nearest neighbors of each query image from the test set. In the K nearest neighbors, if one image of the same class with the query image is obtained, the score is 1, otherwise 0. For image clustering, we use F_1 and normalized mutual information (NMI) [40] metrics. The F_1 score computes the harmonic mean of precision (P) and recall (R) ($F_1 = \frac{2PR}{P+R}$). For the NMI [40], the mutual information $I(\Omega, \Theta)$ between input clusters Ω and the ground truth classes Θ is computed. Then, we compute the average entropy of clusters $H(\Omega)$ and the entropy of ground truth clusters $H(\Theta)$. Finally, NMI is computed by Eq.(15).

$$NMI(\Omega, \Theta) = \frac{2I(\Omega, \Theta)}{H(\Omega) + H(\Theta)} \tag{15}$$

For person re-ID and vehicle re-ID, we use the standard mean average precision score (mAP) [53] and the cumulative matching curve (CMC) at rank-1 as the proposed model evaluation methods. We compute mAP and CMC scores with and without re-ranking technology by using the evaluation code provided by [55].

TABLE I

THE MAPS ON THE CARS196 (CARS) DATA SET AND THE STANFORD ONLINE PRODUCTS (SOP) DATA SET FOR DIFFERENT VALUES OF α AND β IN EQ.(14) ACCORDING TO THE MODEL OF FIG. 1(A)

	β \ α	0.1	0.3	0.5	0.7	0.9
CARS	1	**0.272**	0.260	0.242	0.241	0.230
	2	o-f	o-f	0.242	0.237	0.236
SOP	1	0.369	0.400	0.405	0.407	**0.408**
	10	**0.435**	0.405	0.375	0.358	0.328

B. Multi-Loss Function Experiments

First, we need to determine the hyper-parameters α and β in Eq.(14). We use the GoogLeNet to test our multi-loss function. The test model is shown in Fig. 1(a), and the metric loss is $L_{class-metric}$. The dimension of feature embedding is set to be 128. The fine-grained image data set CARS196 and the Stanford Online Products are used in this experiment. On the CARS196 data set, we test different values of α ranging from 0.1 to 0.9 with interval of 0.2, and the values of β are set to be 1 and 2, respectively. On the Stanford Online Products' data set, the same values of α are used, but the values of β are set to be 1 and 10, respectively. The image retrieval results are summarized in Table I.

From Table I, it can be seen that the best performance is achieved when $\alpha = 0.1$. The value of β should be set as a large value for the general image data set and a small value for the fine-grained image data set. For fine-grained image retrieval, if β is too large, the network will be over-fitting (o-f). Thus, in all subsequent experiments, we set α as 0.1, $\beta = 10$ for general image retrieval and re-ID tasks.

Moreover, we also fine-tune and test the network with only softmax loss and class-metric loss on these two data sets, respectively. The final mAPs are 0.207 and 0.221 for the CARS196 data set and 0.246 and 0.405 for the Stanford Online Products data set. From Table I, we can see that when the values of α and β are chosen appropriately, a higher mAP can be obtained with the multi-loss function in Eq.(14).

C. General Image Retrieval Results

The Stanford Online Products' data set and the In-shop Clothes Retrieval data set are used to evaluate the performances of our proposed feature embedding model for general image retrieval. For the construct of 4-RootHSV, the number of bins for H, S, V are 32, 4, 4, respectively. Thus, an HSV histogram of size 512 can be obtained for each image. Then, l_1 normalization and fourth root scaling are applied to the HSV histogram to obtain a 4-RootHSV [35] feature. For the converter, the number of nodes is 512 for both GoogLeNet and 4-RootHSV, and we design three types of converters in this paper, i.e., ELM [45], [46], auto-encoder [44] and fully connected (fc) network. For the feature embedding, we set the

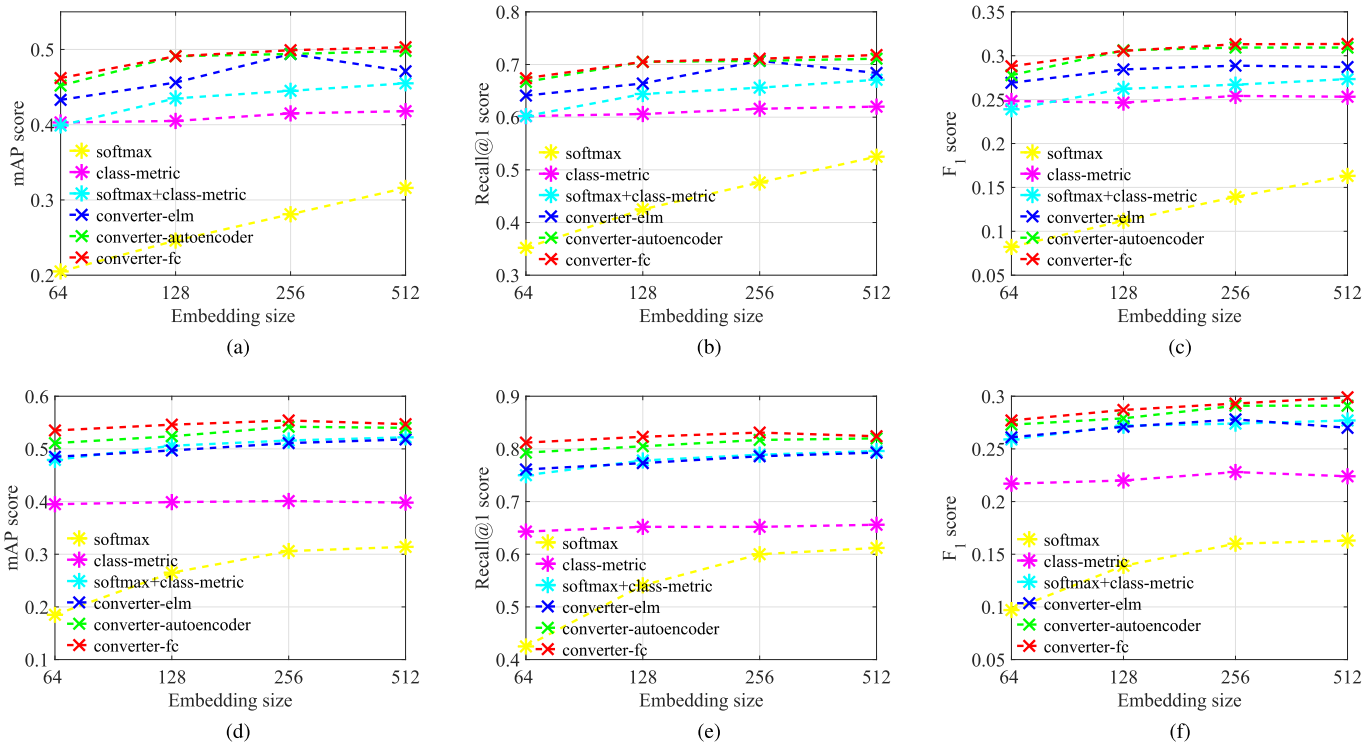


Fig. 4. Experimental results of baseline feature embedding and the proposed feature embedding with dimensions 64, 128, 256 and 512. On the Stanford Online Products' data set, (a) shows the results of mAP for image retrieval, (b) gives the average scores of Recall@1 for image retrieval, and (c) presents the scores of F_1 for image clustering. On the In-shop Clothes Retrieval data set, (d) shows the results of mAP for image retrieval, (e) gives the average scores of Recall@1 for image retrieval, and (f) presents the scores of F_1 for image clustering.

number of nodes as 64, 128, 256 and 512, respectively. The experimental results with and without 4-RootHSV are shown in Fig. 4.

Fig. 4(a) and Fig. 4(b) show the mAPs and Recall@1 on the Stanford Online Products' data set. Fig. 4(d) and Fig. 4(e) show the mAPs and Recall@1 on the In-shop Clothes Retrieval data set. In Fig. 4, the legends of softmax, class-metric and softmax+class-metric representing only the GoogLeNet without converter are used, but the legends of converter-elm, converter-autoencoder and converter-fc representing the GoogLeNet and 4-RootHSV with corresponding converters are used, and the loss is the proposed multi-loss. From these experiments, we can see that the mAP and Recall@1 results of our proposed methods are much higher than those of the other methods. Moreover, the mAPs retrieved with only 4-RootHSV are 0.385 and 0.400 for the Stanford Online Products' data set and the In-shop Clothes Retrieval data set, respectively. In addition, the Recall@1 with only 4-RootHSV is 0.606 and 0.793 for these two data sets, respectively.

For the proposed feature embedding model, as shown in Fig. 4, the best converter is fc, followed by auto-encoder, and ELM is the worst. This is consistent with the analysis in Section III-C. At the same time, the performances of converter-autoencoder and converter-fc are very close, but the converter-fc is slightly better.

Fig. 5(a) and Fig. 5(b) show some query results on the Stanford Online Products' test data set and the In-shop Clothes Retrieval test data set by using the 128 dimensional feature embeddings obtained from the proposed model (converter-fc), respectively. In Fig. 5, the first column is the input query images. Images marked with red rectangles are images that match the query images according to the ground truths.

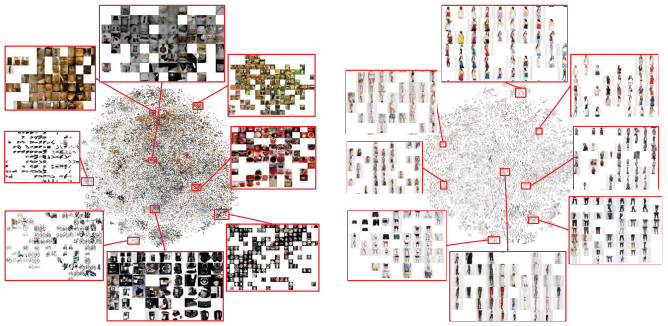
Fig. 5. (a) and (b) show some query results on the Stanford Online Products' test data set and the In-shop Clothes Retrieval test data set by using the 128 dimensional feature embeddings obtained from the proposed model, respectively.

D. Image Clustering Results

For the image clustering task, we use the K-means clustering algorithm to cluster the embedded features of the data set into 11,316 classes and 3,985 classes for the Stanford Online Products' data set and the In-shop Clothes Retrieval data set, respectively. The F_1 scores are shown in Fig. 4(c) and Fig. 4(f), and the NMI scores are listed in Table II.

TABLE II
THE NMI SCORES ON THE STANFORD ONLINE PRODUCTS' DATA SET AND THE IN-SHOP CLOTHES RETRIEVAL DATA SET

	Stanford Online Products				In-shop Clothes Retrieval			
	64	128	256	512	64	128	256	512
softmax	0.824	0.835	0.845	0.852	0.797	0.820	0.830	0.831
class-metric	0.877	0.876	0.875	0.876	0.851	0.852	0.853	0.852
softmax+class-metric	0.871	0.877	0.878	0.880	0.868	0.872	0.873	0.873
converter-elm	0.878	0.882	0.883	0.883	0.869	0.871	0.874	0.872
converter-autoencoder	0.880	0.886	0.887	0.887	0.874	0.876	0.880	**0.880**
converter-fc	**0.883**	**0.887**	**0.888**	**0.888**	**0.878**	**0.880**	**0.882**	**0.880**



(a) (b)

Fig. 6. (a) and (b) are the Barnes-Hut t-SNE visualizations of the Stanford Online Products' test data set and the In-shop Clothes Retrieval test data set by using the 128 dimensional feature embeddings obtained from the proposed model, respectively.

From Fig. 4(c), Fig. 4(f) and Table II, we can see that the performances of multi-loss-based and Fusion-Net unit-based methods are much better than single loss-based and only CNN-based methods. In addition, the best converter is fc, followed by the auto-encoder, and ELM is the worst converter for image clustering.

Fig. 6(a) and Fig. 6(b) are the Barnes-Hut t-SNE [56] visualizations of the Stanford Online Products' test data set and the In-shop Clothes Retrieval test data set by using the 128 dimensional feature embeddings obtained from the proposed model (converter-fc), respectively. More visualization results can be found on our project page or obtained by running our code.[1]

E. Person Re-Identification Results

The Market-1501 data set and the MARS data set are used to evaluate the proposed feature embedding model for the task of person re-ID. Consistent with the settings in the general image retrieval experiments (Section IV-C), the 512 dimensional 4-RootHSV [75] is used in this section, and the number of nodes is 512 for the converter of both GoogLeNet and 4-RootHSV. For the MARS data set, because it is a video-based data set, we take one frame every 16 frames from the released training set as our final training set, so only $1/16$ released training data of the MARS data set are used to train our proposed model. During the testing phase, the re-ranking [55] technique and Cross-view Quadratic Discriminant Analysis (XQDA) [57] metric are used in these two data sets. The parameters of re-ranking are set to be the same as [55]. For the MARS data set test, the average pooling is used for each tracklet for the feature embeddings. The experimental results with and without 4-RootHSV are shown in Fig. 7.

Fig. 7(a) and Fig. 7(b) show the mAPs and Recall@1 based on the Euclidean metric on the Market-1501 data set, and Fig. 7(c) shows the mAPs based on the XQDA metric on this data set. Fig. 7(d) and Fig. 7(e) show the mAPs and Recall@1 based on the Euclidean metric on the MARS data set, and Fig. 7(f) shows the mAPs based on the XQDA metric on the MARS data set. In Fig. 7, the legends of softmax, class-metric and softmax+class-metric representing only the GoogLeNet without converter, but the legends of the converter-fc representing the GoogLeNet and 4-RootHSV with corresponding converters, and the loss is also the proposed multi-loss. The legends with re-ranking representing the re-ranking [55] method are used.

For the Market-1501 data set, we can see from Fig. 7(a)-7(c) that except for the Recall@1 of converter-fc and softmax+class-metric+re-ranking in Fig. 7(b), from low to high, the best performances of person re-ID are softmax, class-metric, softmax+class-metric, converter-fc, softmax+class-metric+re-ranking and converter-fc+re-ranking. These experiments show that the feature embeddings trained by combining multiple loss functions are better than the feature embeddings trained by a single loss function. At the same time, the feature embeddings obtained by merging with 4-RootHSV are also better than that of only CNN-based feature embeddings. In addition, the re-ranking step can greatly enhance the performances of person re-ID based on our proposed model. Furthermore, the results in Fig. 7(a)-7(c) by using only 4-RootHSV are 0.032, 0.1007 and 0.0432 without re-ranking. This shows that the performance is significantly improved by the proposed Fusion-Net unit.

For the MARS data set, according to Fig. 7(d)-7(f), the performances obtained before and after merging with 4-RootHSV are comparable to softmax+class-metric and converter-fc with and without re-ranking, respectively. When the embedded dimensions are 64 and 512, the performance of converter-fc is slightly better than the performance of softmax+class-metric. However, when the embedded dimensions are 128 and 256, it is the opposite. In addition, the re-ranking step can greatly enhance the performances for this video-based data set. Unlike the previous conclusion, the loss of softmax on this data set is better than the loss of class-metric. Moreover, the results corresponding to Fig. 7(d)-7(f) using only 4-RootHSV are

[1]https://github.com/kanshichao/Supervised-Deep-Feature-Embedding

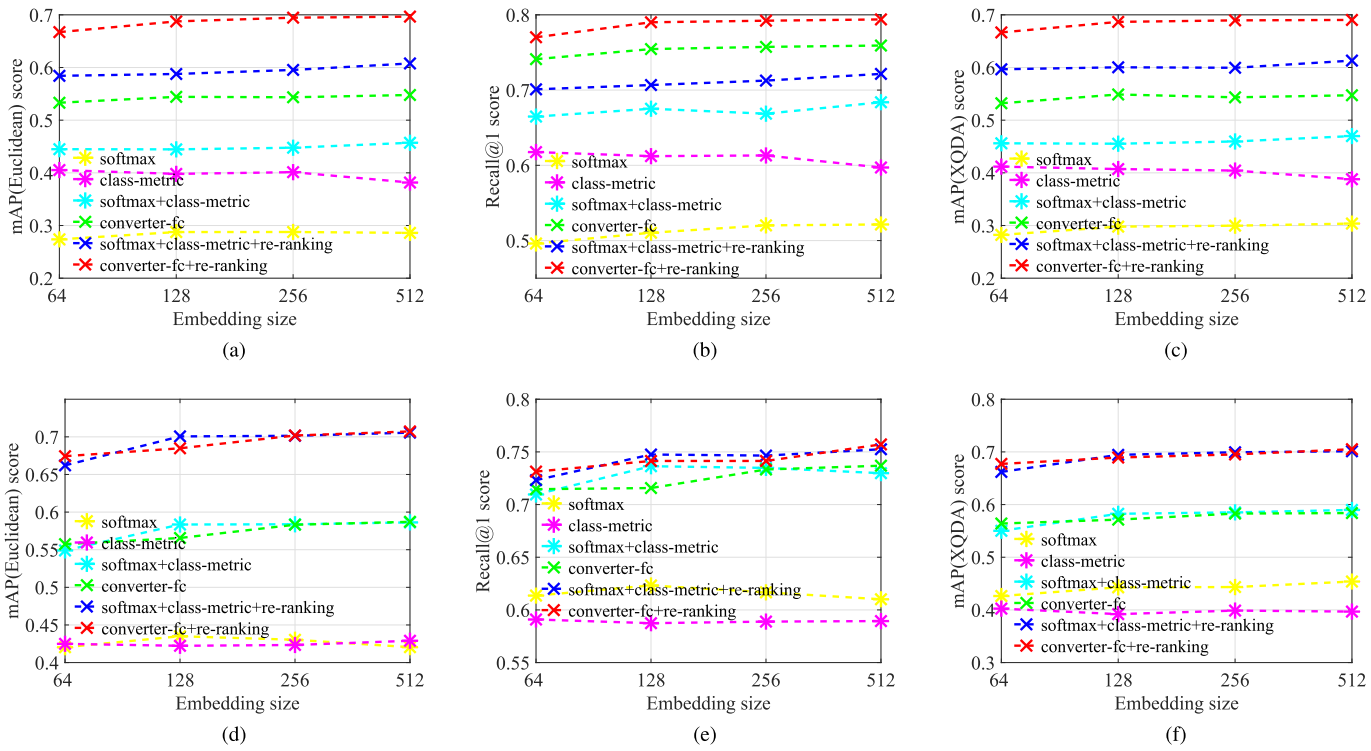


Fig. 7. Experimental results of baseline feature embedding and the proposed feature embedding with dimensions 64, 128, 256 and 512 with and without re-ranking for person re-ID. On the Market-1501 data set, (a) shows the results of mAP (based on the Euclidean metric during the testing phase), (b) gives the average scores of Recall@1 (based on the Euclidean metric during the testing phase), (c) presents the scores of mAP (based on the XQDA metric during the testing phase). On the MARS data set, (d) shows the results of mAP (based on the Euclidean metric during the testing phase), (e) gives the average scores of Recall@1 (based on the Euclidean metric during the testing phase), and (f) presents the scores of mAP (based on the XQDA metric during the testing phase).

0.0439, 0.1071 and 0.0459 without re-ranking for this data set.

From Fig. 7(a) and Fig. 7(c), Fig. 7(d) and Fig. 7(f), we can see that the mAPs of the XQDA metric and the mAPs of the Euclidean metric are consistent for our feature embeddings. Although the performances can be greatly improved by using the XQDA metric for some features, the experimental results show that the Euclidean metric is enough for the feature embeddings of our proposed supervised feature embedding model.

F. Vehicle Re-Identification Results

The VeRi-776 data set is used to evaluate the proposed feature embedding model for the task of vehicle re-ID. The network parameters are set as before (Section IV-C and IV-E). The re-ranking [55] technique is also used in this data set. The experimental results with and without merging with 4-Root-HSV are shown in Fig. 8. The meaning of legends in Fig. 8 are the same as that in Fig. 7.

From Fig. 8, we can see that the results of softmax+class-metric and converter-fc are much better than the results of the softmax and class-metric. At the same time, from Fig. 8(a) and Fig. 8(b), it can be seen that the re-ranking step can boost the mAP and Recall@1 by a large margin, but from Fig. 8(c), the conclusion is opposite for re-ranking. Similar to Fig. 7(d)-7(f), the performances of converter-fc and softmax+class-metric are comparable. This shows that the 4-RootHSV offers almost no help for this type of data set. The results of mAP, Recall@1 and Recall@5 by using only 4-RootHSV are 0.0515, 0.1532 and 0.2414 without re-ranking for this data set.

G. Comparison With the State-of-the-Art Methods

Because the results of image clustering are different with different clustering algorithms, we compare our method with the state-of-the-art methods in general image retrieval, person re-ID and vehicle re-ID.

For general image retrieval, the results of Recall@K of our methods and the state-of-the-art methods are listed in Table III for the Stanford Online Products' data set, and Table IV for the In-shop Clothes Retrieval data set. In Table III and Table IV, the superscripts of these methods denote the dimensions of the embedded features. The results of Softmax^{128}, LiftedStruct^{128} [10], LiftedStruct^{512} [10] and 4-RootHSV^{512} [35] in Table IV are the experimental results for the In-shop Clothes Retrieval data set by these methods.

As we can see from Table III, except the methods of Facility Location^{128} [12] and HDC+Contrastive^{384} [58], our methods with dimensions of 128 and 512 are better than those of the other methods. At the same time, using the converter-fc in Fusion-Net unit, our experimental results are higher than those of the state-of-the-art methods, 2.5% higher and 4.8% higher than the methods of Facility Location^{128} [12], 1% higher and 2.3% higher than the methods of HDC+Contrastive^{384} [58]

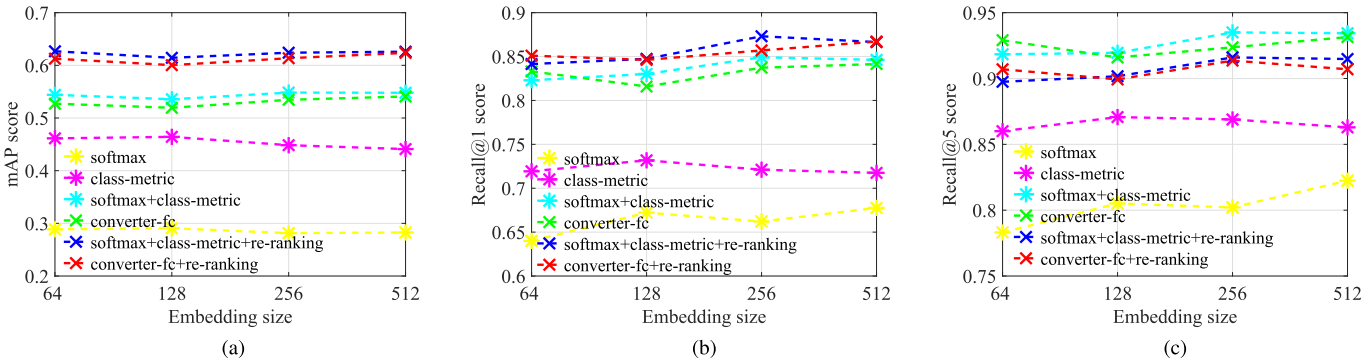


Fig. 8. Experimental results of baseline feature embedding and the proposed feature embedding with dimensions 64, 128, 256 and 512 with and without re-ranking for vehicle re-ID. Based on the Euclidean metric during the testing phase on the VeRi-776 data set, (a) shows the results of mAP, (b) gives the average scores of Recall@1, and (c) presents the scores of Recall@5.

TABLE III

SCORES OF RECALL@K(%) ON THE STANFORD ONLINE PRODUCTS' DATA SET

K	1	10	100	1000
Contrastive^{128} [16]	42.0	58.2	73.8	89.1
Triplet^{128} [9], [39]	42.1	63.5	82.5	94.8
LiftedStruct^{128} [10]	60.8	79.2	91.0	97.3
LiftedStruct^{512} [10]	62.1	79.8	91.3	97.4
Binomial Deviance^{512} [11]	65.5	82.3	92.3	97.6
Histogram Loss^{512} [11]	63.9	81.7	92.2	97.7
Facility Location^{128} [12]	67.0	83.7	92.2	$\sim$
HDC+Contrastive^{384} [58]	69.5	84.4	92.8	97.7
Softmax+class-metric^{128}	64.4	81.6	92.2	**97.7**
Converter-elm^{128}	66.4	83.0	**92.8**	**97.8**
Converter-Autoencoder^{128}	**70.5**	**85.3**	**93.6**	**98.0**
Converter-fc^{128}	**70.5**	**85.3**	**93.6**	**98.0**
Softmax+class-metric^{512}	67.1	83.5	**92.9**	**97.8**
Converter-elm^{512}	68.4	84.2	**93.2**	**98.0**
Converter-Autoencoder^{512}	**71.1**	**85.8**	**93.7**	**98.1**
Converter-fc^{512}	**71.8**	**86.3**	**94.1**	**98.2**

TABLE IV

SCORES OF RECALL@K(%) ON THE IN-SHOP CLOTHES RETRIEVAL DATA SET

K	1	10	20	30
FashionNet+Joints [52]	41.0	64.0	68.0	71.0
FashionNet+Poselets [52]	42.0	65.0	70.0	72.0
FashionNet [52]	53.0	73.0	76.0	77.0
Softmax ^{128}	61.2	84.4	88.5	90.6
HDC+Contrastive^{384} [58]	62.1	84.9	89.0	91.2
LiftedStruct^{128} [10]	65.2	88.2	91.8	93.4
LiftedStruct^{512} [10]	65.6	88.3	91.8	93.2
4-RootHSV^{512} [35]	79.3	91.9	93.6	94.5
Softmax+class-metric^{128}	77.8	**93.6**	**95.8**	**96.5**
Converter-elm^{128}	77.3	**93.0**	**95.2**	**96.2**
Converter-Autoencoder^{128}	**80.5**	**94.2**	**96.1**	**96.8**
Converter-fc^{128}	**82.3**	**95.2**	**96.7**	**97.4**
Softmax+class-metric^{512}	**79.6**	**94.1**	**96.0**	**96.8**
Converter-elm^{512}	**79.3**	**94.1**	**95.9**	**96.0**
Converter-Autoencoder^{512}	**82.0**	**95.1**	**96.6**	**97.2**
Converter-fc^{512}	**82.4**	**95.1**	**96.7**	**97.4**

with Recall@1 metric using Converter-fc^{128} and Converter-fc^{512}, respectively.

According to Table IV, except Recall@1 for Softmax+class-metric^{128} and Converter-elm^{128}, the experimental results of our methods are also better than the state-of-the-art results and 3% higher and 3.1% higher than 4-RootHSV^{512} with Recall@1 metric for Converter-fc^{128} and Converter-fc^{512}, respectively.

TABLE V

COMPARISON OF THE PROPOSED MODEL WITH THE STATE-OF-THE-ART ON THE MARKET-1501 DATA SET

	Recall@1	mAP
SCSP [59]	51.90	26.35
Gated [60]	65.88	39.55
IDE (R) + KISSME + Re-ranking [55]	77.11	63.63
Latent Parts (Fusion) [61]	80.31	57.53
IDE (R) + ML [55]	73.60	49.05
LuNet (R) [48]	81.38	60.71
LuNet (R) + Re-ranking [48]	84.59	75.62
TriNet (R) [48]	84.92	69.14
TriNet (R) + Re-ranking [48]	**86.67**	**81.07**
Softmax+class-metric^{128} (G)	67.52	44.46
Converter-fc^{128} (G)	75.45	54.43
Softmax+class-metric^{128} (G) + Re	70.67	58.77
Converter-fc^{128} (G) + Re	79.01	68.75

The results for person re-ID, Recall@1 and mAPs of our methods and the results of state-of-the-art methods are listed in Table V for the Market-1501 [31] data set and Table VI displays those for the MARS [32] data set. In Table V and Table VI, (R) represents ResNet-50 and (G) represents GoogLeNet. For our methods, the superscripts of these methods denote the dimensions of the embedded features.

From Table V and Table VI, we can see that the best result is obtained by the model trained with the ResNet-50, but our supervised feature embedding model (GoogLeNet-based) also obtain a competitive result, which exceeds all the results except [48]. However, the methods of [48] are based on the ResNet-50, with an improved loss function and hard example mining, all of these can be used in our model to further improve the performance.

Table VII lists the results of vehicle Re-ID results on the VeRi-776 data set. It can be seen that the method of Softmax+class-metric^{256} (G) obtains the best Recall@5, which is higher than the previous state-of-the-art result of 3.46%. And the method of Softmax+class-metric^{256} (G) + Re obtains the best mAP and Recall@1, which are higher than the previous state-of-the-art 4.13% and 3.82%, respectively.

TABLE VI

COMPARISON OF THE PROPOSED MODEL WITH THE STATE-OF-THE-ART ON THE MARS DATA SET

	Recall@1	mAP
LOMO + XQDA [57]	31.82	17.00
IDE (R) + KISSME + Re-ranking [55]	72.32	67.29
Latent Parts (Fusion) [61]	71.71	56.05
IDE (R) + ML [55]	70.51	55.12
LuNet (R) [48]	75.56	60.48
LuNet (R) + Re-ranking [48]	**84.59**	75.62
TriNet (R) [48]	78.48	73.68
TriNet (R) + Re-ranking [48]	81.21	**77.43**
Softmax+class-metric^{128} (G)	73.64	58.35
Converter-fc^{128} (G)	71.57	56.56
Softmax+class-metric^{128} (G) + Re	74.75	70.06
Converter-fc^{128} (G) + Re	74.14	68.48

TABLE VII

COMPARISON OF THE PROPOSED MODEL WITH THE STATE-OF-THE-ART ON THE VERI-776 DATA SET

	mAP	Recall@1	Recall@5
FACT + Plate-SNN + STR [62]	27.77	61.44	78.78
Siamese-Visual + STR [63]	40.26	54.23	74.97
Siamese-CNN [63]	54.21	79.32	88.92
Path-LSTM [63]	54.49	82.89	89.81
Siamese-CNN-VGG16 [63]	44.32	54.41	61.50
Path-LSTM-VGG16 [63]	45.56	47.79	62.63
Siamese + PathLSTM-VGG16 [63]	46.85	50.95	61.62
Siamese-CNN + Path-LSTM [63]	58.27	83.49	90.04
Softmax+class-metric^{128} (G)	53.56	83.02	91.95
Converter-fc^{128} (G)	51.94	81.59	91.60
Softmax+class-metric^{128} (G) + Re	61.43	84.74	90.17
Converter-fc^{128} (G) + Re	60.04	84.62	89.93
Softmax+class-metric^{256} (G)	54.83	84.92	**93.50**
Converter-fc^{256} (G)	53.46	83.73	92.37
Softmax+class-metric^{256} (G) + Re	**62.40**	**87.31**	91.60
Converter-fc^{256} (G) + Re	61.35	85.70	91.36

Moreover, most of our experimental results are higher than the previous state-of-the-art results.

H. Discussions

As shown in Fig. 4, in supervised deep feature embedding with a handcrafted feature model, the performance improvement is primarily from the combination of class-metric loss and softmax loss (the light blue curve in Fig. 4) and merges with the 4-RootHSV feature (the red curve in Fig. 4). In our experiments, the information of the 4-RootHSV feature is embedded into GoogLeNet. However, the features to be embedded are not limited to HSV, and the network is also not limited to GoogLeNet.

From the perspective of feature embedding, solving the unknown parameters of Eq.(8) according to minimize Eq.(3) and Eq.(12), the deep learning method has achieved good results on some data sets based on the GoogLeNet and 4-RootHSV feature. A comparison of Eq.(3) and Eq.(12) shows that they are different. Eq.(12) is suitable for solving parameters with deep learning methods. However, Eq.(3) is suitable for solving parameters with machine learning methods. It is also important to explore machine learning solutions in the future.

In addition, the proposed model can be seen as fusing deeply learned features and handcrafted features by deep feature embedding. Thus, it can be used to the feature fusion area.

V. CONCLUSIONS

In this paper, we developed a model of supervised deep feature embedding with handcrafted feature, which merges 4-RootHSV and combines deep feature embedding, deep metric learning and multi-loss function optimization into a unified framework, and achieves end-to-end learning. In the proposed model, we introduce the idea of the converter to regulate different input representations. Experimental results on the Stanford Online Products' data set and the In-shop Clothes Retrieval data set demonstrate that the proposed methods outperform existing state-of-the-art methods in terms of general image retrieval. In particular, with the converter of the fully connected network, the performances of supervised deep feature embedding with the handcrafted feature model can boost the state-of-the-art results by a large margin. Other experimental results on the Market-1501 data set, the MARS data set and the VeRi-776 data set showed the effectiveness of the proposed methods for person re-ID and vehicle re-ID tasks.

In addition, the proposed supervised deep feature embedding with the handcrafted feature model can also be used for video, text, and speech representation with other CNNs or handcrafted features.

APPENDIX
THE BACK PROPAGATION GRADIENTS OF CLASS-METRIC LOSS FOR THE INPUT FEATURE EMBEDDINGS

The class-metric loss function is defined as Eq.(12). According to the function-derived chain rules, for the positive pairs (i, j) and negative pairs (i, l) and (j, l), the corresponding derivatives are as follows (for convenience, we will use $\mathcal{D}_{i,j}$ instead of $\mathcal{D}_{\mathbf{W}_1,\mathbf{W}_2,\mathbf{M}}(i, j)$, and use $\mathcal{D}_{k,l}$ instead of $\mathcal{D}_{\mathbf{W}_1,\mathbf{W}_2,\mathbf{M}}(k, l)$ in the following derivations):

- **Positive** pairs (i, j):

$$\frac{\partial Q}{\partial f(x_i)} = \frac{\partial Q}{\partial \mathcal{D}_{i,j}}\frac{\partial \mathcal{D}_{i,j}}{\partial f(x_i)} + \frac{\partial Q}{\partial p_i}\frac{\partial p_i}{\partial f(x_i)} \tag{16}$$

$$\frac{\partial Q}{\partial f(x_j)} = \frac{\partial Q}{\partial \mathcal{D}_{i,j}}\frac{\partial \mathcal{D}_{i,j}}{\partial f(x_j)} + \frac{\partial Q}{\partial p_j}\frac{\partial p_j}{\partial f(x_j)} \tag{17}$$

$$\frac{\partial Q}{\partial \mathcal{D}_{i,j}} = \frac{(2 + p_i + p_j)}{2|\mathcal{P}|}\widetilde{Q}_{i,j}\mathbb{I}[\widetilde{Q}_{i,j} > 0] \tag{18}$$

$$\frac{\partial Q}{\partial p_i} = \frac{1}{2|\mathcal{P}|}\widetilde{Q}_{i,j}\mathbb{I}[\widetilde{Q}_{i,j} > 0]\left[\mathcal{D}_{i,j} + \delta_1\right] \tag{19}$$

$$\frac{\partial Q}{\partial p_j} = \frac{1}{2|\mathcal{P}|}\widetilde{Q}_{i,j}\mathbb{I}[\widetilde{Q}_{i,j} > 0]\left[\mathcal{D}_{i,j} + \delta_2\right] \tag{20}$$

$$\delta_1 = \frac{\sum_{(i,l)\in\mathcal{N}} exp\{e - \mathcal{D}_{i,l}\}}{exp\{\widetilde{Q}_{i,j} - \frac{(2+p_i+p_j)}{2}\mathcal{D}_{i,j}\}} \tag{21}$$

$$\delta_2 = \frac{\sum_{(j,l)\in\mathcal{N}} exp\{e - \mathcal{D}_{j,l}\}}{exp\{\widetilde{Q}_{i,j} - \frac{(2+p_i+p_j)}{2}\mathcal{D}_{i,j}\}} \tag{22}$$

- **Negative** pairs (i, l):

$$\frac{\partial Q}{\partial f(x_i)} = \frac{\partial Q}{\partial \mathcal{D}_{i,l}} \frac{\partial \mathcal{D}_{i,l}}{\partial f(x_i)} + \frac{\partial Q}{\partial p_i} \frac{\partial p_i}{\partial f(x_i)} \tag{23}$$

$$\frac{\partial Q}{\partial f(x_l)} = \frac{\partial Q}{\partial \mathcal{D}_{i,l}} \frac{\partial \mathcal{D}_{i,l}}{\partial f(x_l)} + \frac{\partial Q}{\partial p_l} \frac{\partial p_l}{\partial f(x_l)} \tag{24}$$

$$\frac{\partial Q}{\partial \mathcal{D}_{i,l}} = \frac{(2 + p_i + p_l)}{2|\mathcal{P}|} \widetilde{Q}_{i,j} \mathbb{I}[\widetilde{Q}_{i,j} > 0] \sigma_1 \tag{25}$$

$$\sigma_1 = \frac{-exp\{e - \mathcal{D}_{i,l}\}}{exp\{\widetilde{Q}_{i,j} - \frac{(2+p_i+p_j)}{2} \mathcal{D}_{i,j}\}} \tag{26}$$

$$\frac{\partial Q}{\partial p_l} = \frac{1}{2|\mathcal{P}|} \widetilde{Q}_{i,j} \mathbb{I}[\widetilde{Q}_{i,j} > 0] \left[\delta_1 + \delta_2\right] \tag{27}$$

- **Negative** pairs (j, l):

$$\frac{\partial Q}{\partial f(x_j)} = \frac{\partial Q}{\partial \mathcal{D}_{j,l}} \frac{\partial \mathcal{D}_{j,l}}{\partial f(x_j)} + \frac{\partial Q}{\partial p_j} \frac{\partial p_j}{\partial f(x_j)} \tag{28}$$

$$\frac{\partial Q}{\partial f(x_l)} = \frac{\partial Q}{\partial \mathcal{D}_{j,l}} \frac{\partial \mathcal{D}_{j,l}}{\partial f(x_l)} + \frac{\partial Q}{\partial p_l} \frac{\partial p_l}{\partial f(x_l)} \tag{29}$$

$$\frac{\partial Q}{\partial \mathcal{D}_{j,l}} = \frac{(2 + p_j + p_l)}{2|\mathcal{P}|} \widetilde{Q}_{i,j} \mathbb{I}[\widetilde{Q}_{i,j} > 0] \sigma_2 \tag{30}$$

$$\sigma_2 = \frac{-exp\{e - \mathcal{D}_{j,l}\}}{exp\{\widetilde{Q}_{i,j} - \frac{(2+p_i+p_j)}{2} \mathcal{D}_{i,j}\}} \tag{31}$$

In these functions, $\mathbb{I}[\cdot]$ is the indicator function that outputs 1 if the value of the expression is true and outputs 0 otherwise. The remaining derivatives are obvious.

REFERENCES

[1] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, Feb. 2017, San Francisco, CA, USA, 2017, pp. 4278–4284.

[2] Z. Han *et al.*, "Deep spatiality: Unsupervised learning of spatially-enhanced global and local 3D features by deep neural network with coupled softmax," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 3049–3063, Jun. 2018.

[3] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1–9.

[4] M. Paulin, J. Mairal, M. Douze, Z. Harchaoui, F. Perronnin, and C. Schmid, "Convolutional patch representations for image retrieval: An unsupervised approach," *Int. J. Comput. Vis.*, vol. 121, no. 1, pp. 149–168, 2017.

[5] X.-S. Wei, J.-H. Luo, J. Wu, and Z.-H. Zhou, "Selective convolutional descriptor aggregation for fine-grained image retrieval," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2868–2881, Jun. 2017.

[6] P. Liu, J.-M. Guo, C.-Y. Wu, and D. Cai, "Fusion of deep learning and compressed domain features for content-based image retrieval," *IEEE Trans. Image Process.*, vol. 26, no. 12, pp. 5706–5717, Dec. 2017. doi: 10.1109/TIP.2017.2736343.

[7] H. Guo, J. Wang, Y. Gao, J. Li, and H. Lu, "Multi-view 3D object retrieval with deep embedding network," *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5526–5537, Dec. 2016.

[8] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE CVPR*, New York, NY, USA, Jun. 2006, pp. 1735–1742.

[9] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 815–823.

[10] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 4004–4012.

[11] E. Ustinova and V. Lempitsky, "Learning deep embeddings with histogram loss," in *Proc. Adv. Neural Inf. Process. Syst.*, Barcelona, Spain, Dec. 2016, pp. 4170–4178.

[12] H. O. Song, S. Jegelka, V. Rathod, and, K. Murphy, "Deep metric learning via facility location," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 2206–2214.

[13] B. G. V. Kumar, G. Carneiro, and I. D. Reid, "Learning local image descriptors with deep siamese and triplet convolutional networks by minimizing global loss functions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 5385–5394.

[14] B. J. Meyer, B. Harwood, and T. Drummond, "Deep metric learning and image classification with nearest neighbour Gaussian kernels," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Athens, Greece, Oct. 2018, pp. 151–155.

[15] C. Huang, C. C. Loy, and X. Tang, "Local similarity-aware deep feature embedding," in *Proc. Adv. Neural Inf. Process. Syst.*, Barcelona, Spain, Dec. 2016, pp. 1262–1270.

[16] S. Bell and K. Bala, "Learning visual similarity for product design with convolutional neural networks," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 98:1–98:10, 2015.

[17] X. Zhang, F. Zhou, Y. Lin, and S. Zhang, "Embedding label structures for fine-grained feature representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 1114–1123.

[18] Y. Em, F. Gao, Y. Lou, S. Wang, T. Huang, and L.-Y. Duan, "Incorporating intra-class variance to fine-grained visual recognition," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Hong Kong, Jul. 2017, pp. 1452–1457.

[19] B. Harwood, B. G. V. Kumar, G. Carneiro, I. D. Reid, and T. Drummond, "Smart mining for deep metric learning," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 2840–2848.

[20] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2014, pp. 1988–1996.

[21] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," 2014, *arXiv:1411.7923*. [Online]. Available: https://arxiv.org/abs/1411.7923

[22] M. S. Sarfraz and R. Stiefelhagen, "Deep perceptual mapping for thermal to visible face recognition," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, Swansea, U.K., Sep. 2015, pp. 9.1–9.11.

[23] L. Zheng, S. Wang, L. Tian, F. He, Z. Liu, and Q. Tian, "Query-adaptive late fusion for image search and person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1741–1750.

[24] X. Zhang *et al.*, "Deep fusion of multiple semantic cues for complex event recognition," *IEEE Trans. Image Process.*, vol. 25, no. 3, pp. 1033–1046, Mar. 2016.

[25] Z. Liu, S. Wang, L. Zheng, and Q. Tian, "Robust imagegraph: Rank-level feature fusion for image search," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3128–3141, Jul. 2017.

[26] Y. Li, J. Zhang, Y. Cheng, K. Huang, and T. Tan, "DF^{2}Net: Discriminative feature learning and fusion network for RGB-D indoor scene classification," in *Proc. 32nd AAAI Conf. Artif. Intell.*, New Orleans, LA, USA, Feb. 2018, pp. 7041–7048.

[27] D. Yadav, N. Kohli, A. Agarwal, M. Vatsa, R. Singh, and A. Noore, "Fusion of handcrafted and deep learning features for large-scale multiple iris presentation attack detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPR)*, Salt Lake City, UT, USA, Jun. 2018, pp. 572–579.

[28] C. Xiong, L. Liu, X. Zhao, S. Yan, and T. K. Kim, "Convolutional fusion network for face verification in the wild," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 3, pp. 517–528, Mar. 2016.

[29] Q.-S. Sun, S.-G. Zeng, Y. Liu, P.-A. Heng, and D.-S. Xia, "A new method of feature fusion and its application in image recognition," *Pattern Recognit.*, vol. 38, no. 12, pp. 2437–2448, Dec. 2005.

[30] T. Akilan, Q. M. J. Wu, and W. Jiang, "A feature embedding strategy for high-level CNN representations from multiple convnets," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Montreal, QC, Canada, Nov. 2017, pp. 1195–1199.

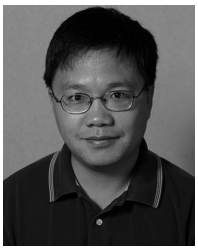
[31] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 1116–1124.

[32] L. Zheng *et al.*, "MARS: A video benchmark for large-scale person re-identification," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Amsterdam, The Netherlands, Oct. 2016, pp. 868–884.

[33] C. Sun, D. Wang, and H. Lu, "Person re-identification via distance metric learning with latent variables," *IEEE Trans. Image Process.*, vol. 26, no. 1, pp. 23–34, Jan. 2017. doi: 10.1109/TIP.2016.2619261.
[34] X. Liu, W. Liu, H. Ma, and H. Fu, "Large-scale vehicle re-identification in urban surveillance videos," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Seattle, WA, USA, Jul. 2016, pp. 1–6.
[35] S.-C. Kan *et al.*, "SURF binarization and fast codebook construction for image retrieval," *J. Vis. Commun. Image Represent.*, vol. 49, pp. 104–114, Nov. 2017.
[36] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a siamese time delay neural network," in *Proc. 7th Conf. Adv. Neural Inf. Process. Syst. (NIPS)*, Denver, CO, USA, 1993, pp. 737–744.
[37] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, San Diego, CA, USA, Jun. 2005, pp. 539–546.
[38] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, Feb. 2009.
[39] J. Wang *et al.*, "Learning fine-grained image similarity with deep ranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Columbus, OH, USA, Jun. 2014, pp. 1386–1393.
[40] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2008.
[41] B. Kulis, M. A. Sustik, and I. S. Dhillon, "Low-rank kernel learning with Bregman matrix divergences," *J. Mach. Learn. Res.*, vol. 10, pp. 341–376, Feb. 2009.
[42] Y. LeCun *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, 1989.
[43] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1704–1716, Sep. 2012.
[44] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
[45] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: A new learning scheme of feedforward neural networks," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, vol. 2. Jul. 2004, pp. 985–990.
[46] L. L. C. Kasun, H. Zhou, G.-B. Huang, and C. M. Vong, "Representational learning with extreme learning machine for big data," *IEEE Intell. Syst.*, vol. 28, no. 6, pp. 31–34, Nov. 2013.
[47] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, Haifa, Israel, Jun. 2010, pp. 807–814.
[48] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," 2017, *arXiv:1703.07737*. [Online]. Available: https://arxiv.org/abs/1703.07737
[49] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACM Int. Conf. Multimedia (MM)*, Orlando, FL, USA, Nov. 2014, pp. 675–678.
[50] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[51] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3D object representations for fine-grained categorization," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Jun. 2014, pp. 554–561.
[52] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "DeepFashion: Powering robust clothes recognition and retrieval with rich annotations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 1096–1104.
[53] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Minneapolis, MN, USA, Jun. 2007, pp. 18–23.
[54] H. Jégou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 117–128, Jan. 2011.
[55] Z. Zhong, L. Zheng, D. Cao, and S. Li, "Re-ranking person re-identification with k-reciprocal encoding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 3652–3661.
[56] L. van der Maaten, "Accelerating t-SNE using tree-based algorithms," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3221–3245, Oct. 2014. [Online]. Available: http://dl.acm.org/citation.cfm?id=2697068
[57] S. Liao, Y. Hu, X. Zhu, and S. Z. Li, "Person re-identification by local maximal occurrence representation and metric learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 2197–2206.
[58] Y. Yuan, K. Yang, and C. Zhang, "Hard-aware deeply cascaded embedding," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 814–823.
[59] D. Chen, Z. Yuan, B. Chen, and N. Zheng, "Similarity learning with spatial constraints for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 1268–1277.
[60] R. R. Varior, M. Haloi, and G. Wang, "Gated siamese convolutional neural network architecture for human re-identification," in *Proc. 14th Eur. Conf. Comput. Vis. (ECCV)*, Amsterdam, The Netherlands, Oct. 2016, pp. 791–808.
[61] D. Li, X. Chen, Z. Zhang, and K. Huang, "Learning deep context-aware features over body and latent parts for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 7398–7407.
[62] X. Liu, W. Liu, T. Mei, and H. Ma, "A deep learning-based approach to progressive vehicle re-identification for urban surveillance," in *Proc. 14th Eur. Conf. Comput. Vis. (ECCV)*, Amsterdam, The Netherlands, Oct. 2016, pp. 869–884.
[63] Y. Shen, T. Xiao, H. Li, S. Yi, and X. Wang, "Learning deep neural networks for vehicle re-ID with visual-spatio-temporal path proposals," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 1900–1909.

Shichao Kan received the B.E. and M.S. degrees from the School of Computer and Information Science, Beijing Jiaotong University, Beijing, China, in 2014 and 2016, respectively, where he is currently pursuing the Ph.D. degree. His research interests include general image retrieval, object search, object detection, large-scale image retrieval, metric learning, and deep learning.

Yigang Cen received the Ph.D. degree in control science engineering from the Huazhong University of Science Technology, Wuhan, China, in 2006. In 2006, he joined the Signal Processing Centre, School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, as a Research Fellow. From 2014 to 2015, he was a Visiting Scholar with the Department of Computer Science, University of Missouri, Columbia, MO, USA. He is currently a Professor and a Supervisor of doctoral students with the School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China. His research interests include compressed sensing, sparse representation, low-rank matrix reconstruction, and wavelet construction theory.



Zhihai He received the B.S. degree in mathematics from Beijing Normal University, Beijing, China, in 1994, the M.S. degree in mathematics from the Institute of Computational Mathematics, Chinese Academy of Sciences, Beijing, in 1997, and the Ph.D. degree in electrical engineering from the University of California at Santa Barbara, Santa Barbara, CA, USA, in 2001. In 2001, he joined Sarnoff Corporation, Princeton, NJ, USA, as a member of technical staff. In 2003, he joined the Department of Electrical and Computer Engineering, University of Missouri, Columbia MO, USA, where he is currently a Tenured Full Professor. His current research interests include image/video processing and compression, wireless sensor network, computer vision, and cyber-physical systems.

Dr. He is a member of the Visual Signal Processing and Communication Technical Committee of the IEEE Circuits and Systems Society. He serves as a technical program committee member or a session chair of a number of international conferences. He was a recipient of the 2002 IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY Best Paper Award and the SPIE VCIP Young Investigator Award in 2004. He was the Co-Chair of the 2007 International Symposium on Multimedia Over Wireless in Hawaii. He has served as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (TCSVT), the IEEE TRANSACTIONS ON MULTIMEDIA (TMM), and the *Journal of Visual Communication and Image Representation*. He was also the Guest Editor for the IEEE TCSVT Special Issue on Video Surveillance.

Zhi Zhang received the B.S. degree in electronic and information technology from Beijing Jiaotong University, Beijing, China, in 2012, and the M.S. and Ph.D. degrees in computer engineering from the University of Missouri-Columbia, Columbia, MO, USA, in 2014 and 2018, respectively. His current research interests include object detection, segmentation, and deep network acceleration.

Linna Zhang received the master's degree in mechanical engineering from Guizhou University, Guiyang, Guizhou, China, in 2010. She is currently a Lecturer with the College of Mechanical Engineering, Guizhou University. Her research interests include image processing and mechanical fault diagnosis.

Yanhong Wang received the B.E. degree from the Qilu University of Technology, Jinan, China, in 2003, and the master's degree from Shandong University in 2009. She is currently pursuing the Ph.D. degree with the Institute of Information Science, Beijing Jiaotong University. Her research interests include multimedia information retrieval, sparse representation, and computer vision.