Multi-Head Deep Metric Learning Using Global and Local Representations

Mohammad K. Ebrahimpour, Gang Qian, and Allison Beach
ObjectVideo Labs, Inc., 8281 Greensboro Dr., Tysons, VA 22102
mkebrahimpour@gmail.com, {gqian, abeach}@objectvideo.com

Abstract

Deep Metric Learning (DML) models often require strong local and global representations, however, effective integration of local and global features in DML model training is a challenge. DML models are often trained with specific loss functions, including pairwise-based and proxy-based losses. The pairwise-based loss functions leverage rich semantic relations among data points, however, they often suffer from slow convergence during DML model training. On the other hand, the proxy-based loss functions often lead to significant speedups in convergence during training, while the rich relations among data points are often not fully explored by the proxy-based losses. In this paper, we propose a novel DML approach to address these challenges. The proposed DML approach makes use of a hybrid loss by integrating the pairwise-based and the proxy-based loss functions to leverage rich data-to-data relations as well as fast convergence. Furthermore, the proposed DML approach utilizes both global and local features to obtain rich representations in DML model training. Finally, we also use the second-order attention for feature enhancement to improve accurate and efficient retrieval. In our experiments, we extensively evaluated the proposed DML approach on four public benchmarks, and the experimental results demonstrate that the proposed method achieved state-of-the-art performance on all benchmarks.

1. Introduction

Learning semantically meaningful representations has been a vital step in numerous computer vision applications such as representation learning [57, 65], content-based visual retrieval [23, 22, 33, 50], person or vehicle re-identification [66, 53, 31], and face verification [27, 15]. Deep Convolutional Neural Networks (CNNs) have proven repeatedly their effectiveness in the large spectrum of applications [55, 6, 11, 10] including Deep Metric Learning (DML). The neural networks in DML are trained to map the data to a lower-dimensional embedding space in which similar data (data in the same class) are pulled together and the dissimilar data (data in different classes) are pushed away [50, 58]. For such an embedding space, rich representations and special loss functions are inevitable.

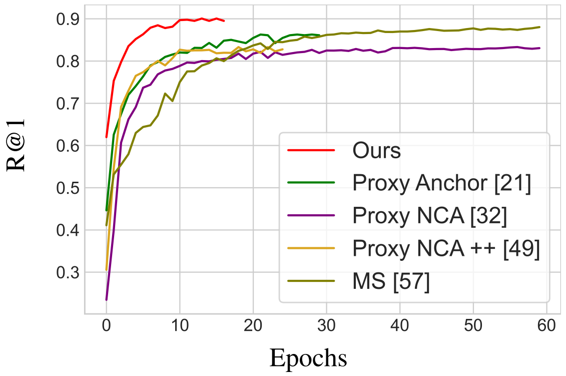

Figure 1. Accuracy in Recall@1 versus epochs on the Cars-196 [26] dataset. Note that all methods were trained on a single Quadro p5000 GPU with a batch size of 100. Our method achieves the highest accuracy while converging at the same order as the proxy-based baselines in terms of the number of epochs.

High image retrieval often requires global and local representations [7]. The global features [44, 1], or “global descriptors” compactly summarize the contents of an image. Often global descriptors are taken from the deepest layers in CNNs; therefore, they only involve the most abstract information, and the vital identifiers such as geometry and spatial information are lost. On the other hand, local features [4, 30], involve information about the geometry and spatial information of the input image. Generally speaking, global features lead to better recall, while local features are essential in better precision [7]. A typical retrieval system setup takes advantage of both global and local features in its final embeddings to obtain the best of both worlds.

Recently, self-attention or Second-Order Attention (SOA) in feature space has received a significant attraction [52, 56, 35, 51, 62]. The SOA can be considered as a spatial enhancement technique that reflects the correlation among spatial locations and enhances the highly correlated parts of the feature map. Although recent deep-learning-based global descriptors provide effective ways to aggregate features into a compact global vector, they have not

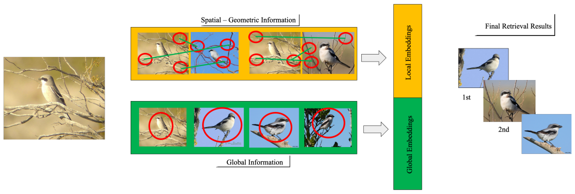

Figure 2. Our proposed deep metric learning architecture with local and global features. Our model jointly extracts deep local and global features. Both of these features will be further enhanced spatially by an SOA mechanism.

explored the correlations of low-level and high-level features within feature maps simultaneously. The other vital factor in DML is the loss function. The loss functions are essential to provide a powerful supervisory signal based on the problem objectives [22, 58]. The loss functions in the DML problems are classified into pairwise-based [58, 45, 54, 47, 38], and proxy-based [33, 22, 50] models. The pairwise-based losses are built based on comparing the pairwise distances between data in the batch. While the pairwise-based losses provide a strong supervisory signal for training the model by considering *data-to-data* relations [47, 58], they suffer from sample mining and slow convergence [22].

The proxy-based losses address the above issues by introducing a limited number of proxies [33, 22, 50]. A proxy is a representative of a subset of training data (for instance, a proxy per class) and learned along with network parameters. Since the number of proxies is substantially smaller than the data-points, the proxy-based models benefit from faster convergence rates than the pairwise-based losses. Note the proxy-based models are associated with *data-to-proxy* relations and they miss the rich supervisory information of data-to-data relations.

In this paper, we propose a multi-head network that benefits from the fast convergence of the proxy-based loss functions and rich data-to-data relation of the pairwise-based models. Although we are using a hybrid of both proxy-based and pairwise-based loss functions in our multi-head network, our approach does not introduce any hyper-parameter tuning for tuple sampling. Our framework also involves an SOA mechanism to exploit the correlation between features at different spatial locations to further enhance the deep local and global features. Also, we combine both global and local descriptors to produce the final descriptor that holds the content information as well as geometry and spatial information to efficiently select the most similar images. With the above advantages, our proposed method achieves state-of-the-art performance in terms of Recall@1 and quickly converges as exhibited in Figure 1.

The contribution of this paper unfolds as follows: (a) We propose a multi-head network that takes advantage of both pairwise-based and proxy-based methods; it leverages rich data-to-data relations and enables fast and reliable convergence. (b) We explore the SOA for further enhancement of both local and global features based on higher-order information. (c) We demonstrate the impact of using local and global descriptors, proxy-based and pairwise-based, SOA, and the embedding dimensions via a thorough ablation study on their effects. (d) An embedding neural network trained with our approach achieves state-of-the-art performance on four publicly available benchmarks for metric learning [26, 60, 39, 28].

2. Related Work

In this section, we categorize the DML approaches based on their use of descriptors and loss functions into two broad categories, then we review relevant papers in each category.

2.1. Loss Functions

Loss functions in DML can be divided into two groups, pairwise-based and proxy-based.

Pairwise-based Losses. Contrastive loss [8, 16] and Triplet loss [45, 54] are influential examples of loss functions for pairwise-based DML. Contrastive loss takes a pair of embedding vectors as input and aims to push them apart if they are of different classes or pull them together if they are of the same class. Triplet loss considers a data point as an anchor. Each anchor is associated with a positive (an embedding with an identical class label to the anchor) and a negative data point (an embedding with different class labels) and involves the distance of the anchor-positive pair to be smaller than that of the anchor-negative pair in the embedding space.

One potential issue with pairwise-based models is that a large number of tuples have a limited contribution to the learning algorithm and sometimes even diminish the quality of the learned embedding space [61]. To address this issue, most pairwise-based losses [47, 40, 59] employ hard sample mining techniques [61, 17]. However, these techniques involve tuning hyper-parameters and consequently increases the risk of over-fitting. Pairwise-based losses are rich in data-to-data relations. However, the number of tuples increases polynomially with regard to the number of training data, resulting in prohibitive complexity and significantly slow convergence [22].

Proxy-based Losses. Proxy-based metric learning endeavors to address the complexity and slow convergence issue of the pairwise-based losses. The proxy-based methods require a small set of proxies to capture the global structure of an embedding space and assign each data point to relevant proxies instead of the other data points during training. Since the number of proxies is significantly smaller than the training data, the training complexity reduces substantially. For instance, Proxy-NCA [33] loss assigns a single proxy to each class and associates data points to each proxy and

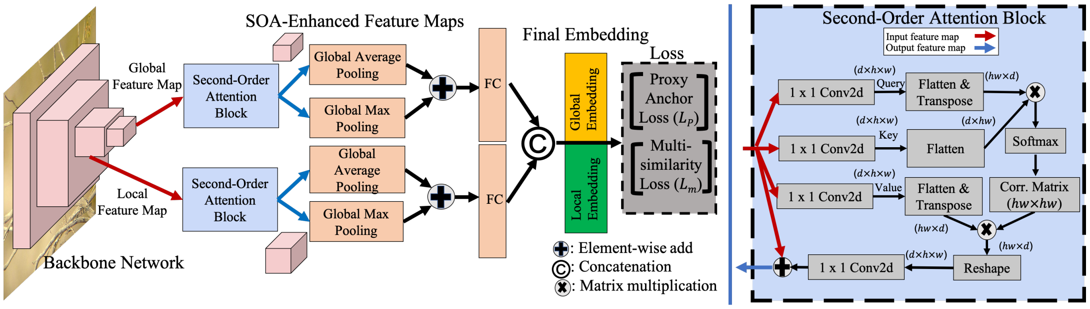

Figure 3. Our proposed multi-head metric learning architecture with joint local and global features. The local and global features are jointly extracted from the backbone and they are sent to the SOA block (the blue block) for further enhancement. Then, we apply pooling layers on top of the refined and re-weighted features. The final embedding involves the concatenation of local and global representations that are used for retrieval to efficiently select the most similar images based on both local and global identifiers simultaneously. Finally, a hybrid loss involving a proxy-based and a pairwise based is applied to the final embedding.

encourages positive pairs to be close together and negative pairs to be far apart. Proxy-NCA++ [50] is an extension of the Proxy-NCA and aims to enhance the limitations of the Proxy-NCA in terms of temperature factor and pooling layer.

SoftTriple loss [43], inspired by the Proxy-NCA yet assigns multiple proxies to each class instead of one to improve the likelihood of capturing the intra-class variance. Proxy-Anchor loss [22] assigns a proxy to each class and treats each proxy as an anchor and assigns positive and negative pairs to each anchor. Although introducing proxies in proxy-based losses significantly improves the convergence in model training, it has an inherent limitation of data-to-proxy relation instead of data-to-data relation that results in limited supervisory information. Our multi-head network overcomes this limitation by proposing a hybrid approach involves in both pairwise-based and proxy-based methods to benefit data-to-data relations as well as high convergence rates.

2.2. Descriptors

DML algorithms can be divided into three groups based on their use of descriptors: local descriptors, global descriptors, and joint local and global descriptors.

Local Descriptors. Hand-engineered features such as SIFT [30] and SURF [4] have been widely used and adopted for retrieval systems especially before the deep learning era. The key advantage of local features over global ones for image retrieval is their capacity to perform spatial matching, often by utilizing RANSAC [12]. Due to the efficiency of local features, recently, several deep learning-based local features have been proposed [63, 36, 32, 41].

Global Descriptors. Global descriptors are often involved the most abstract information about the input, leading to high-performance image retrieval. Before the deep learning era, most global descriptors were obtained using the combination of local descriptors [20, 21]. However, recently most high-performing global features are obtained based on CNNs [3, 2, 14].

Joint Local and Global Descriptors. Global descriptors are essential for high recalls yet local descriptors are necessary for better precision; therefore, researchers develop hybrid methods to take advantage of both descriptors. For instance, Taira *et al*. [49] used NetVLAD [2] to extract global features for candidate pose retrieval, followed by dense local feature matching using feature maps from the same network for indoor localization. Simeoni *et al*. 's DSM [46] detected key points in activation maps from global feature models. Activation channels are interpreted as visual words, to propose correspondences between a pair of images. Cao *et al*. [7] extracted global and local features from the same network. They utilize the global descriptors to retrieve the most similar images and then re-rank the retrieved images by local descriptors to increase the precision.

3. Proposed Algorithm

Our proposed algorithm involves two essential components: Refined deep local and global representations along with a multi-head loss function that enables the data-to-data relation as well as fast convergence. Our model design is illustrated in Figure 2.

3.1. Deep Global and Local Representations

We propose to leverage hierarchical representations from a CNN to represent different types of descriptors. While deep layers are associated with the most abstract representations and representing higher-level features, the intermediate layers are more informative in terms of local representations and lower-level features.

Given an image, from the backbone we obtain two feature maps: $f_l \in \mathbb{R}^{H_l \times W_l \times C_l}$ and $f_g \in \mathbb{R}^{H_g \times W_g \times C_g}$, representing local (l) and global (g) feature maps where H, W, C indicate the height, width, and number of channels respectively. For off-the-shelf convolutional networks, $H_g \leq H_l$,

$W_g \leq W_l$, and $C_g \geq C_l$; indicating that deeper layers have larger number of channels and spatially smaller feature maps.

3.2. Second-Order Attention (SOA)

Let (i_I, j_I) in the input image (I) correspond to location (i, j) in feature map f. To incorporate higher-order spatial information into the feature map, we adopt the second-order attention block [52, 35]. A computational flow of the SOA concept is shown in Figure 3. For each feature map, we produce two projections of feature map f named *query* q, and *key* k, each obtained through 1×1 2d-convolutions with possible reduction of number of channels (d). Then, by flattening both tensors, we obtain both q and k in $\mathbb{R}^{d\times hw}$. The second-order attention map A is computed as follows:

$$a = \text{softmax}(\zeta q^T k), \tag{1}$$

where ζ is a scaling factor and $a \in \mathbb{R}^{hw\times hw}$, indicating the correlation of each $f_{i,j}$ to the whole map f. A third projection of f and *value* v is then obtained by 1×1 2d-convolution, and after flattening, results in $\mathbb{R}^{hw\times d}$ shape. Then, the second-order attention map f^{soa} is obtained from linear combination of the first-order features f and the second-order attention map:

$$f^{soa} = f + \phi(a \times v), \tag{2}$$

where ϕ is yet another 1×1 convolution to manage the effect of the obtained attention map. Thus, a new feature $f^{soa}_{i,j}$ in the second-order map $f^{soa} \in \mathbb{R}^{h\times w\times d}$, is a function of features from all locations in f:

$$f^{soa}_{i,j} = h(a_{ij} \odot f), \tag{3}$$

where h denotes the combination of all convolutional operations within the non-local block.

3.3. Pooling

To aggregate deep activations in both global and local features, we adopt the combination of Global Max Pooling (GMP) and Global Average Pooling (GAP) as follows:

$$f = \frac{1}{W \times H} \sum_{i\in W, j\in H} f^{soa} + \max_{(i\in W, j\in H)} f^{soa} \tag{4}$$

After the aggregation, we whiten the aggregated representation for both refined local and global representations; we integrate this into our model with two separated fully-connected layers. The fully connected layer associated with enhanced local representations $F_l \in \mathbb{R}^{C_{f_l^{soa}} \times \frac{D}{2}}$, with learned bias $b_{f_l^{soa}} \in \mathbb{R}^{C_{f_l^{soa}}}$, which $C_{f_l^{soa}}$ indicates the number of channels in the f_l^{soa} and $\frac{D}{2}$ is the dimension of the local embedding space. Similarly, we have a fully connected layer associated with enhanced global representations $F_g \in \mathbb{R}^{C_{f_g^{soa}} \times \frac{D}{2}}$, with learned bias $b_{f_g^{soa}} \in \mathbb{R}^{C_{f_g^{soa}}}$, which $C_{f_g^{soa}}$ indicates the number of channels in the f_g^{soa} and $\frac{D}{2}$ is the dimension of the global embedding space.

After computing $F_l \in \mathbb{R}^{D/2}$ and $F_g \in \mathbb{R}^{D/2}$, the final embedding $F \in \mathbb{R}^D$ computes by concatenation of the F_l and F_g.

3.4. A Hybrid Loss Function

Our loss is designed to overcome the limitation of both proxy-based and pairwise-based models by introducing a hybrid loss involving the proxy-anchor [22] loss from the proxy-based category and the MS [58] loss from the pairwise-based class.

Proxy-based Loss. Proxy-anchor loss [22] assigns a proxy to each class. Proxy-anchor approach considers each proxy as an anchor and associate it with entire data in a batch to find positive and negative samples. The proxy-anchor loss defined as follows:

$$\begin{aligned}\ell_p(X) = \frac{1}{|P^+|} \sum_{p\in P^+} \log(1 + \sum_{x\in X_p^+} \exp(-\alpha(s(x,p) - \delta))) \\ + \frac{1}{|P|} \sum_{p\in P} \log(1 + \sum_{x\in X_p^-} \exp(\alpha(s(x,p) + \delta))),\end{aligned} \tag{5}$$

where $\delta > 0$ is a margin, $\alpha > 0$ is a scaling factor, P is the set of all proxies, $s(.,.)$ measures the similarity among its arguments ,and P^+ indicates the set of positive proxies of data in the batch. Also, for each proxy p, a batch of embedding vectors X is divided into the set of positive X_p^+ and negative $X_p^- = X - X_p^+$ embedding vectors.

By utilizing the proxy-anchor loss we incorporate data-to-proxy relations as well as fast convergence. For incorporating the data-to-data relation, we integrate the MS loss.

Pairwise-based Loss. We employ the MS loss [58] as a pairwise-based loss since it considers the self, negative, and positive similarities. Self-similarity ensures that the instances belonging to a positive class remains closer to the anchor than the instances associated with negative classes. The positive similarity exclusively deals with positive pairs. σ represents the similarity margin that controls the closeness of positive pairs by heavily penalizing those pairs whose cosine similarities are less or equal to σ. The negative similarity ensures that negative samples have similarity with the anchor as low as possible. The MS loss function is formulated as follows:

$$\begin{aligned}\ell_m(X) = \frac{1}{m} \sum_{i=1}^{m} (\frac{1}{\gamma} \log(1 + \sum_{k\in P_i} \exp(-\gamma(S_{i,k} - \sigma))) \\ + \frac{1}{\beta} \log(1 + \sum_{k\in N_i} \exp(\beta(S_{i,k} + \sigma))),\end{aligned} \tag{6}$$

Table 1. Recall@K (%) on the Cars-196 [26] and CUB-200-2011 [60] datasets. Superscripts indicate embedding sizes. Backbone networks of the models are denoted by abbreviations: G–GoogleNet [48], BN–Inception with batch normalization [19], R50–ResNet50 [18]. For each group of methods, the best performance is bolded and the second best is underlined.

Algorithms	BackBone	Cars-196				CUB-200-2011			
		R@1	R@2	R@4	R@8	R@1	R@2	R@4	R@8
Clustering^{64} [37]	BN	58.1	70.6	80.3	87.8	48.2	61.4	71.8	81.9
Proxy-NCA^{64} [33]	BN	73.2	82.4	86.4	87.8	49.2	61.9	67.9	72.4
Smart Mining^{64} [17]	G	64.7	76.2	84.2	90.2	49.8	62.3	74.1	83.3
MS^{64} [58]	BN	78.6	86.6	91.8	95.4	60.1	71.9	81.2	88.5
Proxy-Anchor^{64} [22]	R50	78.8	87.0	92.2	**95.5**	61.7	73.0	81.8	88.8
Ours^{64}	R50	**81.1**	**88.1**	**92.3**	95.3	**63.1**	**74.6**	**83.2**	**89.4**
Margin^{128} [61]	R50	79.6	86.5	91.9	95.1	63.6	74.4	83.1	90.0
Ours^{128}	R50	**84.9**	**90.6**	**94.0**	**96.6**	**66.6**	**77.1**	**85.2**	**91.3**
HDC^{384} [64]	G	73.7	83.2	89.5	93.8	53.6	65.7	77.0	85.6
A-BIER^{512} [42]	G	82.0	89.0	93.2	96.1	57.5	68.7	78.3	86.2
ABE^{512} [24]	G	85.2	90.5	94.0	96.1	60.6	71.5	79.8	87.4
HTL^{512}	BN	81.4	88.0	92.7	95.7	57.1	68.8	78.7	86.5
RLL-H^{512} [59]	BN	74.0	83.6	90.1	94.1	57.4	69.7	79.2	86.9
MS^{512} [58]	R50	84.1	90.4	94.0	96.5	65.7	77.0	86.3	91.2
SoftTriple^{512} [43]	BN	84.5	90.7	94.5	96.9	65.4	76.4	84.5	90.4
Proxy-Anchor^{512} [22]	BN	86.1	91.7	95.0	97.3	68.4	79.2	86.8	91.6
ProxyNCA++^{512} [50]	R50	86.5	92.5	95.7	97.7	69.0	79.8	87.3	**92.7**
Proxy-Anchor^{512} [22]	R50	87.7	92.9	95.8	97.9	69.7	80.0	87.0	92.4
Ours^{512}	R50	**90.1**	**94.2**	**96.4**	**98.1**	**70.6**	**80.9**	**88.0**	92.3

where γ, β, and σ are hyper-parameters. m is the number of samples, P_i,N_i represents positive and negative samples, and $S_{i,j}$ denotes the pairwise similarity between x_i and x_j.

Our Objective Function. Our hybrid objective function is a combination of proxy-anchor and MS losses balanced by normalization factor λ:

$$\mathcal{L} = \ell_m + \lambda \ell_p \tag{7}$$

4. Experimental Results

In this section, our method is compared with current state-of-the-art methods on four public benchmark datasets employed for deep metric learning [26, 60, 39, 28]. We also perform a thorough investigation of local and global features, the SOA, the MS loss and proxy-anchor loss, and embedding dimensionality to study their effects on the proposed method.

4.1. Dataset

We evaluated our model on the CUB-200-2011 [60], Cars-196 [26], Stanford Online Product (SOP) [39] and In-Shop Clothes Retrieval (In-Shop) [28] datasets. For CUB-200-2011, we set aside 5,864 images of its first 100 classes as a training set and 5,924 images of the other classes as a test set. For Cars-196, 8,054 images of its first 98 classes are set aside as a training set and 8,131 images of the other classes are used as a test set. For SOP, we follow the standard dataset split in [40, 22, 50] using 59,551 images of 11,318 classes as a training set and the remaining 60,502 images as a test set. Also for In-Shop dataset, we follow the setting in [22] to use 25,882 images of the first 3,997 classes as a training set and 28,760 images of the remaining

Table 2. Recall@K (%) on the SOP. Superscripts indicates embedding sizes. For each group of methods, the best performance is bolded and the second best is underlined.

Recall@K	1	10	100	1000
Clustering^{64} [37]	67.0	83.7	93.2	-
Proxy-NCA^{64} [33]	73.7	-	-	-
MS^{64}	74.1	87.8	94.7	98.2
SoftTriple^{64} [43]	76.3	89.1	95.3	-
Proxy-Anchor^{64} [22]	76.5	89.0	95.1	98.2
Ours^{64}	**77.3**	**89.5**	**95.4**	**98.3**
Margin^{128} [61]	72.7	86.2	93.8	98.0
Ours^{128}	**79.1**	**90.6**	**95.8**	**98.5**
HDC^{384} [64]	69.5	84.4	92.8	97.7
A-BIER^{512} [42]	74.2	86.9	94.0	97.8
ABE^{512} [24]	76.3	88.4	94.8	98.2
HTL^{512} [13]	74.8	88.3	94.8	98.4
RLL-H^{512} [59]	76.1	89.1	95.4	-
MS^{512} [58]	78.2	90.5	96.0	98.7
SoftTriple^{512} [43]	78.3	90.3	95.9	-
Proxy-Anchor^{512} [22]	79.1	90.8	96.2	98.7
Proxy-NCA++^{512} [50]	80.7	**92.0**	**96.7**	**98.9**
Ours^{512}	**81.7**	**92.0**	96.6	98.8

Table 3. Recall@K (%) on the In-Shop. Superscripts indicates embedding sizes. For each group of methods, the best performance is bolded and the second best is underlined.

Recall@K	1	10	20	40
HDC^{384} [64]	62.1	84.9	89.0	92.3
HTL^{128} [13]	80.9	94.3	95.8	97.4
MS^{128} [58]	88.0	97.2	98.1	98.7
Proxy-Anchor^{128} [22]	90.8	**97.9**	98.5	**99.0**
Ours^{128}	**90.9**	**97.9**	**98.6**	98.9
FashionNet^{4096} [28]	53.0	73.0	76.0	79.0
A-BIER^{512} [42]	83.1	95.1	96.9	97.8
ABE^{512} [24]	87.3	96.7	97.9	98.5
MS^{512} [58]	89.7	97.9	98.5	99.1
Proxy-Anchor^{512} [22]	91.5	98.1	98.8	99.1
ProxyNCA++^{512} [50]	90.4	98.1	98.8	**99.2**
Ours^{512}	**93.1**	**98.3**	**99.0**	**99.2**

classes for test set; the test set is further partitioned into a query and gallery sets with 14,218 images of 3,985 classes and 12,612 images of 3,985 classes, respectively. For all datasets, we set aside 20% of the training set as a validation set for hyper-parameter tuning [34].

4.2. Implementation Setup

Backbone: For a fair comparison [34] to recent works, the Resnet50 [18] pre-trained on ImageNet classification [9] is adopted as our backbone network.

Global and Local Features: For all experiments on all datasets, we obtained the global features from resnet50-conv5-x$\in \mathbb{R}^{7\times 7\times 2084}$ layer and local features are extracted from resnet50-conv4-x$\in \mathbb{R}^{14\times 14\times 1024}$.

Training: In all of the experiments, AdamW algorithm [29] has been adopted and used as our optimizer. AdamW has the same update step as Adam [25] with separate decays of weights. In all experiments, our model is trained only for 20 epochs and the initial learning rate is fixed to 10^{-4}. Note that the learning rate for proxies is set to 10^{-2} for faster convergence.

Proxy Setting: We assign a single proxy to each class as suggested in Proxy-NCA [33] and Proxy-Anchor [22]. The proxies are initialized with a normal distribution.

Input Setting: To have a fair comparison with state-of-the-art methods, the input is re-scaled to 256 × 256 and then center-cropped to 224 × 224. We used random cropping and horizontal flipping during training as the data augmentation strategy, as suggested in [22, 33]. During the test, the images are only center-cropped. The default size of cropped images is fixed to 224 × 224 [34].

Hyperparameter Setting: ζ in Eq. 1 is set to 1. α and δ in Eq. 5 is set to 32 and 10^{-1} respectively. γ, β, and σ in Eq. 6 are set to $2, 50, 1$. Finally, λ in Eq. 7 is set to 3×10^{-2}, for all experiments.

4.3. Comparison to Other Methods

We demonstrate the strength of our proposed method quantitatively by evaluating its image retrieval performance on four public benchmark datasets. For a fair comparison to the previous arts [34], the accuracy of our model is computed in three different settings: we used 64, 128, and 512 embedding dimension on Cars-196, CUB-200-2011, and SOP datasets with the default image size 224 × 224. On the In-Shop dataset the performance is only measured with embedding dimensions of 128 and 512 with the default image size 224 × 224. Results on the Cars-196 and CUB-200-2011 datasets are exhibited in Table 1. According to Table 1, our model outperforms all the previous state-of-the-art methods including the proxy-based [22, 33, 50], pairwise-based [59, 58, 43] and ensemble methods [24] in all three settings often with a large margin on top 1 recall. In particular, on the challenging Cars-196 dataset, our method improves the previous best score by a large margin, 2.3%, 5.3%, and 2.4% in Recall@1 in embedding size of 64, 128, and 512 respectively. As reported in Table 2, our model also achieves state-of-the-art performance on the SOP dataset. It outperforms previous methods in all cases except for Recall@10 and Recall@100 in 64-dimensional embedding, but even in these cases, it achieves the second best. Finally, on the In-Shop dataset, it obtains the best scores in all two settings as shown in Table 3. On the In-Shop dataset, our model outperforms the state-of-the-art by a large margin of 2.7% in Recall@1. In our experimental results, we noted that our model outperforms numerous state-of-the-art methods even with low-dimensional embedding vectors while they have higher embedding dimensions. This observation suggests that our model is capable of learning a more compact and effective embedding space. Last but not least, our hybrid method converges in the same order as the proxy-based method as results are summarized in Figure 1.

4.4. Qualitative Results

To further exhibiting the visual performance of our method, we illustrate the qualitative retrieval results of our model on four datasets in Figure 4. Note that these datasets are challenging especially due to their large intra-class variations. For instance, the CUB200-2011 has a variety of poses and background clutter, the Cars-196 has various colors and shapes, and SOP and In-Shop datasets have challenging view-points of objects that make the retrieval tasks even harder. In contrast to all of these challenges, our proposed method performs robust retrieval.

4.5. Ablation Study

Local and Global Features. To investigate the impact of the local and global features on the performance of our proposed method, we examined the Recall@1 while training

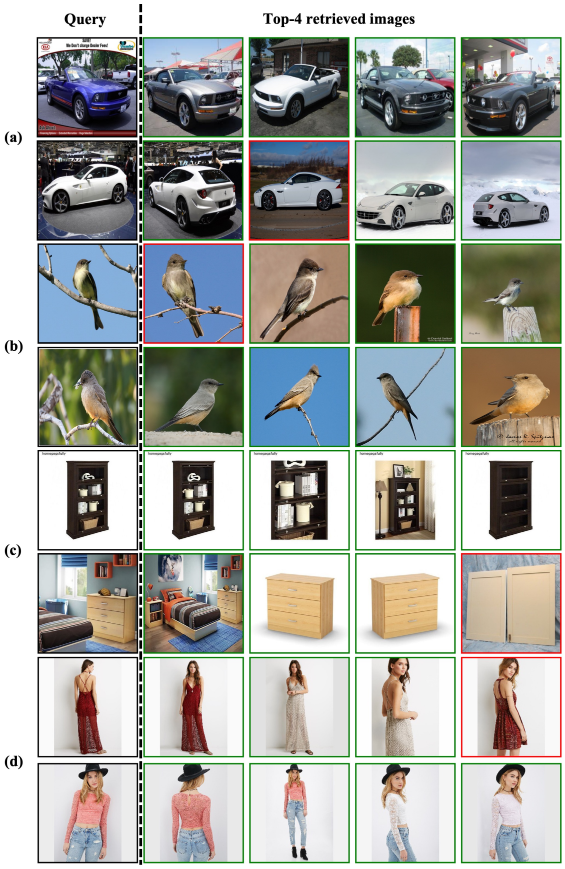

Figure 4. Qualitative results on the Cars-196 (a) CUB-200-2011 (b), SOP (c), and In-Shop (d). For each query image (leftmost), top 4 retrievals are exhibited. The results with red boundaries are false cases but they are substantially similar to the query images in terms of appearance. (rows 2,3,6, and 7).

our model with only local and global descriptors separately on the Cars-196 and CUB-200-2011 datasets. The result of the analysis is summarized in Figure 5 (a). This Figure illustrates the local descriptors on both Cars-196 and CUB-200-2011 datasets are getting slightly better performance (2%) than the global descriptors but neither of these descriptors individually achieved the performance as the combination of these descriptors obtained on both datasets based on Recall@1 performance (see Table 1).

Second-order Attention. One of the vital components of our proposed method is SOA. We employ this attention mechanism to further enhance deep features based on their correlation. To evaluate the impact of the SOA, we trained our model with and without having the SOA, and the results are summarized in Figure 5 (b). According to this figure, the performance of the model significantly improves in terms of Recall@1 when the model is trained with SOA especially on critical datasets like CUB-200-2011. This observation confirms our assumption on the essence of higher-order attention for further enhancements of the representations. We visualize the effects of second-order attention in Figure 6. This figure exhibits the attention map of samples from Cars-196 and CUB-200-2011 datasets. For each image, four parts have been selected with different stars and different colors. The attention map associated with each star has a border with identical colors. For locations in the background, interestingly, the attention from that feature is distributed within the main object in the image. It confirms the second-order attention has learned to focus on the main object in the image to accurately retrieve the most similar images to the query. On the other hand, when the star is located within the main object, the attention is on highly distinctive regions.

A Single Head Attention. Our proposed method requires two attention heads for local and global descriptors. We are interested in probing a test case with having only a single attentional head for both local and global descriptors. After extracting local descriptor $f_l \in \mathbb{R}^{14\times14\times1024}$ and global descriptor $f_g \in \mathbb{R}^{7\times7\times2048}$ from backbone, we use a focus layer [5] to reshape the local descriptor $f_l^{new} \in \mathbb{R}^{7\times7\times4096}$ to spatially match the global descriptor. Then, we concatenate the global and local descriptors resulting in a combined feature map $f \in \mathbb{R}^{\in 7\times7\times6144}$. Then, we apply the SOA for further enhancement of this feature map. We evaluated this approach on Cars-196 and CUB-200-2011 datasets. The single head attention obtained 86.6% and 66.6% in terms of Recall@1 on Cars-196 and CUB-200-2011, respectively. Note the obtained results are slightly degraded from the multi-head approach 90.1% on Cars-196 and 70.6% on CUB-200-2011 (see Table 1), that suggests the multi-head SOA is necessary.

Multisimilarity and Proxy-anchor Loss. The other crucial component of our architecture is the combination of pairwise-based and proxy-based loss functions. To study the impact of each loss function on our proposed method, we trained our model separately with MS loss and proxy anchor loss and we evaluated the performance based on Recall@1 on Cars-196 and CUB-200-2011 datasets. The results is exhibited in Figure 7. This Figure demonstrates that the combination of these two losses are crucial in our design since none of them individually achieves our performance (see Table 1).

Embedding Dimension. The dimension of embedding vectors is a vital factor that controls the trade-off between speed and accuracy in image retrieval systems. We thus investigate the effect of embedding dimensions on the retrieval accuracy in our method. We evaluated our model with embedding dimensions varying from 64 to 2048 following the experiment in [58, 22]. The result of the analysis is illustrated in Figure 8, in which the retrieval performance of our model is reported on both Cars-196 and CUB-200-2011 dataset.

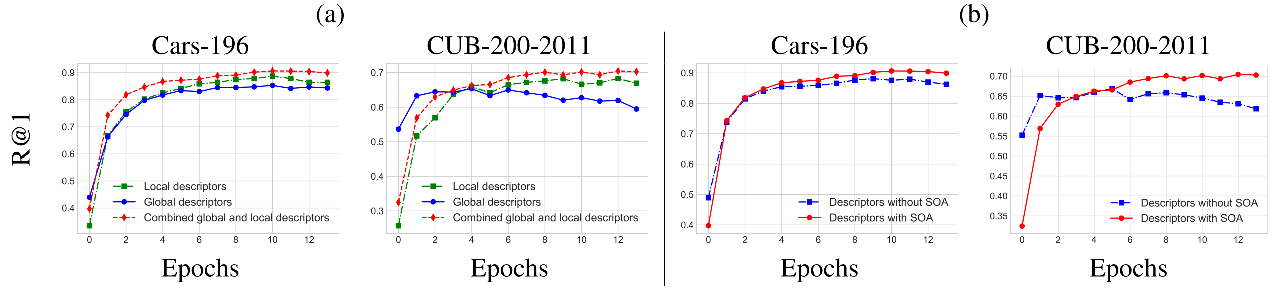

Figure 5. (a): The impact of local and global descriptors in terms of Recall@1 performance on both Cars-196 and CUB-200-2011 datasets. The blue and green colors illustrate the performance of the global and local descriptors, while the red indicates the performance of the combined global and local descriptors. The (a) part of this figure demonstrates the combination of local and global features are vital in our design. (b): The impact of the SOA component on the mentioned datasets. The blue color exhibits the model without second-order attention and red depicts the model with second-order attention.

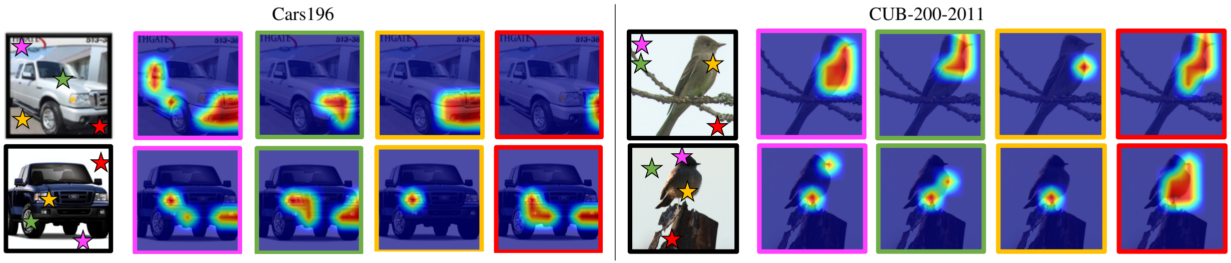

Figure 6. Qualitative examples of SOA maps on the Cars-196 and CUB-200-2011 datasets. (a) corresponds to Cars196 dataset and (b) corresponds to CUB-200-2011 dataset. Each row depicts the source image and four corresponding SOA maps obtained for specific spatial locations (marked by stars).

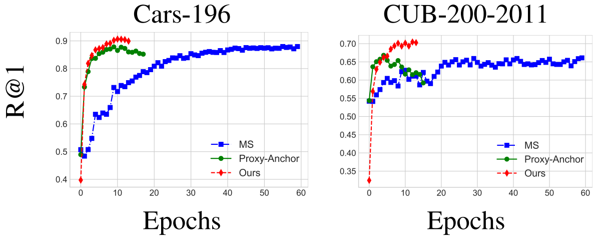

Figure 7. The impact of the MS and Proxy-Anchor loss on Cars-196 and CUB-200-2011 datasets. The red color indicates our hybrid loss while the green and blue represent the Proxy-Anchor and MS losses, respectively. The Figure demonstrates the combination of two losses is crucial in our proposed method.

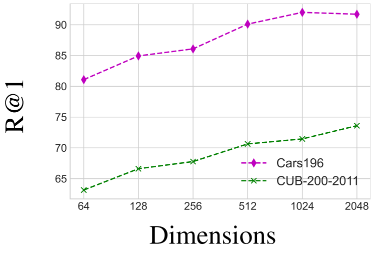

Figure 8. Accuracy in terms of Recall@1 versus embedding dimension on both Cars-196 and CUB-200-2011 datasets.

The performance of our loss is fairly stable when the dimension is equal to or larger than 128. The performance of our model on the Cars-196 dataset improves until reaching 1024 dimensional embedding and after that slightly degrades. On the other hand, the performance consistently increases with the embedding dimension on the CUB-200-2011 dataset showing that more information on that dataset helps the retrieval performance.

5. Conclusion

We have proposed a novel metric learning algorithm that takes the best of both proxy-based and pairwise-based losses. Also, it leverages enhanced local and global descriptors to improve the recall and precision simultaneously. Our method benefits from having a reach data-to-data relation as well as fast and reliable convergence. We extensively evaluated our model on 4 public benchmarks and our model has achieved state-of-the-art performance on all datasets in terms of Recall@1 accuracy. Also, our model converged quickly without any careful data sampling technique.

Acknowledgements

This research work was sponsored by the Office of Naval Research (ONR) via contract N00014-17-C-2007. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official positions, policies or endorsements, either expressed or implied, of ONR or the U.S. Government.

References

[1] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *CVPR*, 2016. 1

[2] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *CVPR*, 2016. 3

[3] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. Neural codes for image retrieval. In *ECCV*, 2014. 3

[4] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *CVIU*, 2008. 1, 3

[5] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 7

[6] Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. Sign language transformers: Joint end-to-end sign language recognition and translation. In *CVPR*, 2020. 1

[7] Bingyi Cao, Andre Araujo, and Jack Sim. Unifying deep local and global features for image search. In *ECCV*, 2020. 1, 3

[8] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, 2005. 2

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 6

[10] Mohammad K Ebrahimpour, J Ben Falandays, Samuel Spevack, Ming-Hsuan Yang, and David C Noelle. Ww-nets: Dual neural networks for object detection. In *IJCNN*, 2020. 1

[11] Mohammad K Ebrahimpour, Jiayun Li, Yen-Yun Yu, Jackson Reesee, Azadeh Moghtaderi, Ming-Hsuan Yang, and David C Noelle. Ventral-dorsal neural networks: object detection via selective attention. In *WACV*, 2019. 1

[12] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981. 3

[13] Weifeng Ge. Deep metric learning with hierarchical triplet loss. In *ECCV*, 2018. 5, 6

[14] Albert Gordo, Jon Almazan, Jerome Revaud, and Diane Larlus. End-to-end learning of deep visual representations for image retrieval. *IJCV*, 2017. 3

[15] Jianzhu Guo, Xiangyu Zhu, Chenxu Zhao, Dong Cao, Zhen Lei, and Stan Z Li. Learning meta face recognition in unseen domains. In *CVPR*, 2020. 1

[16] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006. 2

[17] Ben Harwood, Vijay Kumar BG, Gustavo Carneiro, Ian Reid, and Tom Drummond. Smart mining for deep metric learning. In *ICCV*, 2017. 2, 5

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5, 6

[19] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 5

[20] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *CVPR*, 2010. 3

[21] Herve Jegou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Perez, and Cordelia Schmid. Aggregating local image descriptors into compact codes. *PAMI*, 2011. 3

[22] Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. Proxy anchor loss for deep metric learning. In *CVPR*, 2020. 1, 2, 3, 4, 5, 6, 7

[23] Sungyeon Kim, Minkyo Seo, Ivan Laptev, Minsu Cho, and Suha Kwak. Deep metric learning beyond binary supervision. In *CVPR*, 2019. 1

[24] Wonsik Kim, Bhavya Goyal, Kunal Chawla, Jungmin Lee, and Keunjoo Kwon. Attention-based ensemble for deep metric learning. In *ECCV*, 2018. 5, 6

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[26] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *CVPRW*, 2013. 1, 2, 5

[27] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *CVPR*, 2017. 1

[28] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *CVPR*, 2016. 2, 5, 6

[29] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[30] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 1, 3

[31] Dechao Meng, Liang Li, Xuejing Liu, Yadong Li, Shijie Yang, Zheng-Jun Zha, Xingyu Gao, Shuhui Wang, and Qingming Huang. Parsing-based view-aware embedding network for vehicle re-identification. In *CVPR*, 2020. 1

[32] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *NIPS*, 2017. 3

[33] Yair Movshovitz-Attias, Alexander Toshev, Thomas K Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *ICCV*, 2017. 1, 2, 5, 6

[34] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. *arXiv preprint arXiv:2003.08505*, 2020. 6

[35] Tony Ng, Vassileios Balntas, Yurun Tian, and Krystian Mikolajczyk. Solar: Second-order loss and attention for image retrieval. *arXiv preprint arXiv:2001.08972*, 2020. 1, 4

[36] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *ICCV*, 2017. 3

[37] Hyun Oh Song, Stefanie Jegelka, Vivek Rathod, and Kevin Murphy. Deep metric learning via facility location. In *CVPR*, 2017. 5

[38] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *CVPR*, 2016. 2

[39] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *CVPR*, 2016. 2, 5

[40] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *CVPR*, 2016. 2, 5

[41] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. Lf-net: learning local features from images. In *NIPS*, 2018. 3

[42] Michael Opitz, Georg Waltner, Horst Possegger, and Horst Bischof. Deep metric learning with bier: Boosting independent embeddings robustly. *PAMI*, 2018. 5, 6

[43] Qi Qian, Lei Shang, Baigui Sun, Juhua Hu, Hao Li, and Rong Jin. Softtriple loss: Deep metric learning without triplet sampling. In *ICCV*, 2019. 3, 5, 6

[44] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Finetuning cnn image retrieval with no human annotation. *PAMI*, 2018. 1

[45] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 2

[46] Oriane Siméoni, Yannis Avrithis, and Ondrej Chum. Local features and visual words emerge in activations. In *CVPR*, 2019. 3

[47] Kihyuk Sohn. Improved deep metric learning with multiclass n-pair loss objective. In *NeurIPS*, 2016. 2

[48] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 5

[49] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. Inloc: Indoor visual localization with dense matching and view synthesis. In *CVPR*, 2018. 3

[50] Eu Wern Teh, Terrance DeVries, and Graham W Taylor. Proxynca++: Revisiting and revitalizing proxy neighborhood component analysis. In *ECCV*, 2020. 1, 2, 3, 5, 6

[51] Yurun Tian, Xin Yu, Bin Fan, Fuchao Wu, Huub Heijnen, and Vassileios Balntas. Sosnet: Second order similarity regularization for local descriptor learning. In *CVPR*, 2019. 1

[52] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 1, 4

[53] Dongkai Wang and Shiliang Zhang. Unsupervised person reidentification via multi-label classification. In *CVPR*, 2020. 1

[54] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *CVPR*, 2014. 2

[55] Li Wang, Dong Li, Yousong Zhu, Lu Tian, and Yi Shan. Dual super-resolution learning for semantic segmentation. In *CVPR*, 2020. 1

[56] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 1

[57] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *ICCV*, 2015. 1

[58] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *CVPR*, 2019. 1, 2, 4, 5, 6, 7

[59] Xinshao Wang, Yang Hua, Elyor Kodirov, Guosheng Hu, Romain Garnier, and Neil M Robertson. Ranked list loss for deep metric learning. In *CVPR*, 2019. 2, 5, 6

[60] P Welinder, S Branson, T Mita, C Wah, F Schroff, S Belongie, P Perona, and Caltech-UCSD Birds. 200. *California Institute of Technology*, 2010. 2, 5

[61] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. Sampling matters in deep embedding learning. In *ICCV*, 2017. 2, 5

[62] Bryan Ning Xia, Yuan Gong, Yizhe Zhang, and Christian Poellabauer. Second-order non-local attention networks for person re-identification. In *CVPR*, 2019. 1

[63] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *ECCV*, 2016. 3

[64] Yuhui Yuan, Kuiyuan Yang, and Chao Zhang. Hard-aware deeply cascaded embedding. In *ICCV*, 2017. 5, 6

[65] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *CVPR*, 2015. 1

[66] Zhizheng Zhang, Cuiling Lan, Wenjun Zeng, Xin Jin, and Zhibo Chen. Relation-aware global attention for person reidentification. In *CVPR*, 2020. 1