Discrimination-Aware Mechanism for Fine-grained Representation Learning

Furong Xu, Meng Wang, Wei Zhang, Yuan Cheng, and Wei Chu
Ant Financial Services Group
{booyoungxu.xfr,darren.wm,ivy.zw,chengyuan.c,weichu.cw}@antgroup.com

Abstract

Recently, with the emergence of retrieval requirements for certain individual in the same superclass, e.g., birds, persons, cars, fine-grained recognition task has attracted a significant amount of attention from academia and industry. In fine-grained recognition scenario, the inter-class differences are quite diverse and subtle, which makes it challenging to extract all the discriminative cues. Traditional training mechanism optimizes the overall discriminativeness of the whole feature. It may stop early when some feature elements has been trained to distinguish training samples well, leaving other elements insufficiently trained for a feature. This would result in a less generalizable feature extractor that only captures major discriminative cues and ignores subtle ones. Therefore, there is a need for a training mechanism that enforces the discriminativeness of all the elements in the feature to capture more the subtle visual cues. In this paper, we propose a Discrimination-Aware Mechanism (DAM) that iteratively identifies insufficiently trained elements and improves them. DAM is able to increase the number of well learned elements, which captures more visual cues by the feature extractor. In this way, a more informative representation is learned, which brings better generalization performance. We show that DAM can be easily applied to both proxy-based and pair-based loss functions, and thus can be used in most existing fine-grained recognition paradigms. Comprehensive experiments on CUB-200-2011, Cars196, Market-1501, and MSMT17 datasets demonstrate the advantages of our DAM based loss over the related state-of-the-art approaches.

1. Introduction

Fine-grained recognition aims at distinguishing each subclasses on a specific superclass dataset, such as birds, persons, cars. There are only a few effective cues that can distinguish samples of different classes due to the samples belong to a superclass. Meanwhile, images are usually captured at different times/places, resulting in various visual differences appear in the same subclass. The extremely low intra-class similarity and high inter-class similarity make fine-grained recognition very challenging. Recently, with the popularity of retrieving examples of a specific subclass from the superclass database, this topic has been attracting a significant amount of attention from academia and industry.

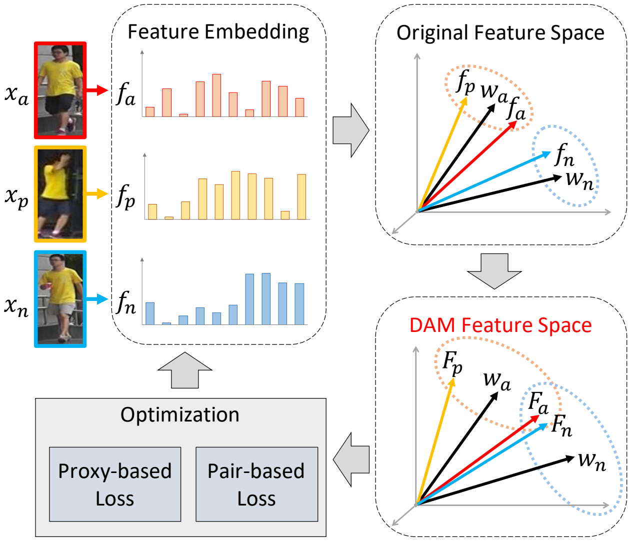

Figure 1. Illustration of DAM for representation learning. On top left, a triplet (x_a, x_p, x_n) and their features f_a, f_p, f_n are shown in different colors. On top right, black vectors indicate class centers, and the dotted circles indicate class distributions. On bottom right, f_a, f_p and f_n are mapped to a harder DAM feature space as F_a, F_p and F_n. Inter-class samples (F_a and F_n) are pushed close and intra-class samples (F_a and F_p) are pushed away, as well as the distributions of intra-class are enlarged. Then the new features by DAM are inputed to loss function. The pipeline runs iteratively and leads to a better generalization performance.

Current representation learning methods mainly focus on three directions: (1) Designing powerful loss function to extract robust feature embeddings [18, 29, 52, 53, 44], such as proxy-based loss and pair-based loss. (2) Constructing attention module to resolve local regions [15, 3, 24, 62, 66]. (3) Randomly erasing parameters or feature values during training for better generalization performance [67, 10, 43, 13, 9]. With the powerful deep networks and large-scale labeled benchmarks, these methods can obtain a relevant feature representation for the image.

Despite the significant progress in fine-grained recognition, most existing methods focus on optimizing the overall discriminativeness of the whole feature elements to distinguish it from other classes. But in fine-grained recognition scenario, the inter-class differences are quite diverse and subtle. As shown in top-right subfigure of Figure 1, in original feature space, traditional optimization mechanism may stop early when some elements has been trained to distinguish training samples well, while other elements are insufficiently trained. This would result in a less generalizable feature extractor that only captures major discriminative cues and ignores subtle ones. But these subtle cues may really discriminative for fine-grained recognition. *e.g.,* shoes and glasses of person. Therefore, there is a need for a training mechanism that enforces the discriminativeness of all the elements in the feature to capture more the subtle visual cues.

In this paper, we propose a Discrimination-Aware Mechanism (DAM) to iteratively learn elements of features more disciminative. As shown in Figure 1, we encourage the model to keep learning by mapping low-discriminative feature to a harder space, and then further optimize the new features. Specifically, by exploiting difference between various specific inter-classes, we retain elements of features with less diferences to other classes and erase rest elements for the harder features. For example, f_a, f_p and f_n are mapped to DAM feature space as F_a, F_p and F_n respectively. In DAM feature space, inter-class samples are pushed close (*e.g.,* distance between F_a and F_n) and intra-class samples are pushed away (*e.g.,* distance between F_a and F_p), as well as the distributions of intra-class are enlarged. In our DAM, whether the feature element needs to be retained depends on the difference of the feature value between different classes. For a certain element of the features, the smaller difference it is from other classes, the more it should be retained to improve its discriminativeness. By using DAM to the existing proxy-based and pair-based loss, only the selected elements are used for gradient update, and remaining elements of feature are erased during training. Finally, by iteratively mining discriminative cues, all elements of a feature are discriminative after convergence. In this way, more diverse and subtle cues are extracted, thus improving the discriminativeness of the overall feature representation.

In summary, the main contributions of our work are listed as follows:

- We propose a discrimination-aware mechanism to extract more discriminative visual cues. Compared with previous attention-based methods, our method does not need to modify network architecture. And we use differences between classes to guide feature mapping in constrast to previous erasing-based methods,
- We design two feature mapping mechanism to proxy-based loss and pair-based loss. For proxy-based loss, we selelet low-discriminativeness elements for each feature. For pair-based loss, we also select low-discriminativeness elements for paired-negative features, but high-discriminativeness elements for paired-postive features to enhance the effectiveness of triplets.
- We achievie better performance on fine-grained recognition tasks compared with the related state-of-the-art methods. i.e, CUB-200-2011, Cars196, Market-1501, and MSMT17.

2. Related Work

The core idea of fine-grained representation learning is to obtain discriminative features where semantic information between classes are fully learned. There are currently some solutions to this problem, we will introduce these methods related to our work.

Fine-grained Representation Learning. Fine-grained representation learning [15, 24, 23, 56, 49, 28, 64, 5, 11, 33, 59, 34] aims to extract discriminative features, where the intra-class distances are closer than inter-class ones. However, there are large visual differences in the same class, and much apparent similarity between classes. The existing methods to solve this computer vision task can be divided into three categories, *e.g.* loss-designed methods, attention-based methods and random erasing methods.

Loss Function Designed Methods. With the popularity of deep learning, a mass of powerful backbones [41, 32, 47, 26, 48, 46, 17, 25, 22, 7, 55, 60, 20, 61, 6, 21] emerged. By utilizing the feature extraction capabilities of the backbone, most of the previous methods focus on designing loss function to derive robust representation. Typical losses mainly include the following two categories: proxy-based loss (*e.g.,* softmax cross-entropy loss [32, 41, 17]) and pair-based loss (*e.g.,* triplet loss [18, 40, 58, 57]). The first ones use class-level labels to separate different classes by using proxies. The second ones use pair-wise labels to directly pull the same class close and push the different classes far away. The two losses are complementary, so we use softmax cross-entropy loss and triplet loss as our baseline.

Attention-based Methods. Loss designed methods mainly focus on global features, while attention-based methods expect to mining some local features. Some methods use extra information to guide attention maps. Song et al. [42] used person mask to extract person front. Han et al. [15, 24] used attribute to get attention maps. Hou et al. [19] learned spatial and channel attention maps to multiply the features. RGA [62] aims to capture the global structural information for better attention learning. Our DAM also selects some locations of feature that need to pay attention. But our work is fundamentally different from the existing attention-based methods in two folds. Firstly, **the attention manner is different.** We adopt differences between inter-classes to guide attention, but existing methods do not have direct supervision signal when there is no external label, *e.g.* mask, key points. Secondly, **attention direction is different**. We are committed to discovering locations with low-discriminativeness, but existing methods amis to find discriminative ones for further enhancement.

Random Erasing Methods. Recently, some random erasing methods achieved significant improvements in computer vision tasks. The works [67, 10] randomly erase a rectangle region of the input images during training. Dropout [43] drops the feature units randomly, which is a widely used regularization technique to prevent overfitting. DropBlock [13] randomly drops a contiguous region of the convolutional features for CNNs. In the work [9], all feature maps in the same batch are dropped in a consistent way for better metric learning. Our method also erases some feature elements for further learning the remaining elements. **But different from the existing methods by random erasing, we use differences between inter-classes to guide erasing locations for features.**

3. Proposed Approach

In this section, we firstly illustrate the details of the proposed discrimination-aware mechanism. Then, we introduce the application of DAM to proxy-based loss and pair-based loss. Thirdly, we describe the overall framework of our representation learning method. Finally, we further prove the effectiveness of our DAM from the perspective of gradient optimization.

3.1. Discrimination-Aware Mechanism

In fine-grained recognition task, there is a high degree of visual similarity between different classes. For example, the students on campus wear similar school uniforms with the same hairstyles. From a visual point of view, there are much diverse and subtle differences between the two person, *e.g,* shoes, glasses. However, for the same class, there are high visual differences, *e.g,* postures, backgrounds, illuminations and viewpoints. Therefore, it is particularly necessary to extract really effective cues to distinguish samples.

Intuitively, for a feature, when each element of the feature is discriminative, this feature can be more robust. In other words, the feature whose each element discriminative will contain as many effective cues as possible. For fine-grained recognition task, since some cues that can distinguish classes are so subtle, extracting more cues for feature representation can help improve recognition performance. To be specific, when enough cues are extracted, some cues that can really distinguish the classes will make the feature similarity between intra-class larger, and the similarity between the inter-classes smaller.

Most existing methods for representation learning optimized the overall discriminativeness of the whole feature elements. While some methods use random erasure mechanism to erase certain image/features areas to increase generalization ability. However, both of these methods have drawbacks. For the first type of methods, since deep convolutional neural networks (DCNNs) first learn some easy (visually significant) areas, then some hard (visually subtle) cues [14]. Therefore, the loss function will converge when some significant discriminative cues are learned. This will cause the learned model fits training data well but has poor generalization ability. Although the optimization target is satisfied on the training set, it is difficult to be effective for other datasets due to insufficient learning. For the second type of methods, since the erased areas are uncontrollable, the results may be trapped in local optimal. Therefore, erasing locations deserve a guided mechanism.

In this paper, we propose a discrimination-aware mechanism to extract more cues for better discriminative representation. Inspired by random erasing, we design a guided erasure mechanism to encourage the model to iteratively learn the new discriminative cues. To be specific, we obtain instructive input signals by exploiting the difference between classes. Intuitively, when two features are different in each element, that two features have a larger difference. Therefore, we can achieve high-discriminative inter-class features by constraining as many feature elements to be different as possible, thereby prompting the model to mine more plentiful cues for distinguish examples.

In order to make the feature elements between inter-classes as different as possible, we split elements of each feature into two sets every iteration during the training process. One set with low-discriminativeness elements are iteratively optimized, and the other set elements already with high-discriminativeness are erased. With the help of this erasing mechanism, we have obtained new gated features to further optimize the model parameters. In feature space perspective, DAM projects the training samples from original feature space into a more difficult (to distinguish each sample) space. The entire generation process of new features is shown in Figure 2.

To determine which elements should be erased, we utilize the differences between classes, which can be obtained by calculating the difference between the class centers. Specifically, when using softmax cross-entropy (SCE) loss to supervise the classification task, the weights w ($w \in R^{C \times D}$, where C is class number, and D is element number of a feature) of the last fully connected (FC) layer are used as proxies to pull intra-class close and push inter-classes away. Mathematically, a feature f_i is projected onto all weight vectors $[w_1, ..., w_C]$ to determine its class, where w_i is a D-dim vector as shown in Figure 2 (a). During training, SCE loss function optimizes the classification object

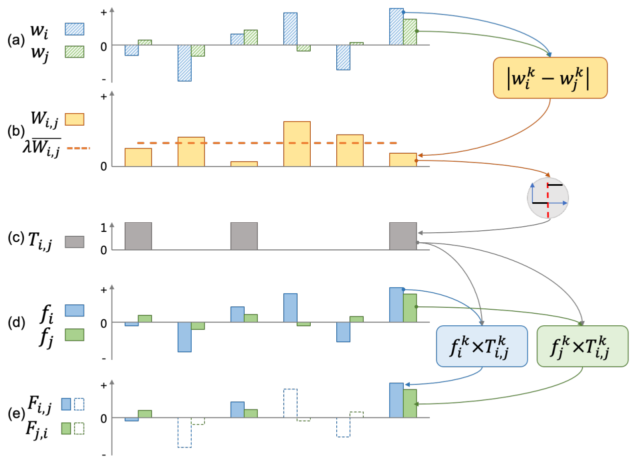

Figure 2. Illustration of our DAM to generate new gated features. w_i and w_j are D-dim weight vectors in the last FC layer.

by lengthening the projection of f_i on the true class weight w_{y_i}. In other words, the more similar the value of each dimension of the feature and class weight, the longer the projection. Therefore, class weights can represent the average feature of intra-class samples (class center), and the difference between class weights can represent the difference between inter-classes. The class differences $W_{i,j}$ between class c_i and c_j can be defined as:

$$W_{i,j} = |w_i - w_j| \tag{1}$$

where $|.|$ takes the absolute value of each vector dimension. $W_{i,j}$ is a D-dim vector, which indicates the signification of each element in features for distinguishing class c_i and c_j. The higher the value, the more discriminative of that feature element between the two classes.

With a representation of the differences between two classes, we can decide which elements of fearures for the two classes need to be further optimizing. When the value of the difference is smaller, it means that this element of features has lower-discriminativeness between the two classes. To select effective feature elements, we design a gate mechanism to cut off unnecessary parts. Specifically, we use the mean value of the difference to measure effectiveness. To increase flexibility, we set an adjustable parameter λ. As shown in Figure 2 (b-c), the new gated difference weight $T_{i,j}$ for class c_i and c_j can be defined as:

$$T_{i,j}^k = \begin{cases} 1, & W_{i,j}^k < \lambda\overline{W_{i,j}} \\ 0, & W_{i,j}^k >= \lambda\overline{W_{i,j}} \end{cases} \tag{2}$$

$$\overline{W_{i,j}} = \frac{1}{D}\sum_{l=1}^{l=D} W_{i,j}^l \tag{3}$$

where $W_{i,j}^k$ expresses the signification of the k-th element for distinguishing class c_i and c_j. After getting $T_{i,j}$, we can get a new gated feature embedding $F_{i,j}$ from the original feature f_i as shown in Figure 2 (d-e), which has low-discriminativeness to samples of class c_j. The new gated $F_{i,j}$ can be defined as:

$$F_{i,j} = f_i \times T_{i,j} \tag{4}$$

where $\times$ represents element-wise multiplication.

Besides, for feature f_i, its element set which cannot be distinguished from all other classes needs further optimization as well. To represent the discriminativeness of feature dimensions, we measure the average difference between feature f_i and all other classes. The average difference of f_i can be defined as:

$$W_{i,all} = \frac{1}{C-1}\sum_{j=1,j\neq i}^{C} W_{i,j} \tag{5}$$

By using the same gate mechanism as a specific class (Equ 2) , we can obtain the gated difference $T_{i,all}$ of the feature f_i between all other classes. Then we can get the low discriminative $F_{i,all}$ for f_i:

$$F_{i,all} = f_i \times T_{i,all} \tag{6}$$

3.2. Discrimination-Aware Mechanism Application

When our discrimination-aware mechanism generate new gated features, we can input them to proxy-loss and pair-based loss for further model optimization.

Discrimination-Aware Mechanism for Proxy-based Loss. For proxy-based loss function, multiple proxies (the weights w_i of the last FC layer) are adopted to optimize the model parameters. By iteratively optimizing new low-discriminativeness feature parts, we can improve the robustness of global features. The softmax cross-entropy based discrimination-aware mechanism (SCEDAM) can be rewrited as:

$$\mathcal{L}_{SCEDAM} = -\frac{1}{N\times C}\sum_{i=1}^{N}\sum_{k=1}^{C} y_i^l * \log\frac{e^{F_{i,k}\cdot w_k}}{\sum_{j=1}^{C} e^{F_{i,k}\cdot w_j}} \tag{7}$$

$$F_{i,k} = \begin{cases} F_{i,all}, & k = argmax(y_i) \\ F_{i,k}, & k \neq argmax(y_i) \end{cases} \tag{8}$$

where N is batch size, y_i is the label of image x_i, it is a C-dim one-hot vector. y_i^l represents the label of sample x_i is l, $y_i^l = 1$.

The difference between $\mathcal{L}_{SCEDA}$ and $\mathcal{L}_{SCE}$ is that we use new gated features for classification. In addition, when the new gated feature is projected to the weight vector of the gt class, we use features that are less discriminative than all other class. When the new gated feature is projected to the non-gt weight vector, we use the low-discriminanation

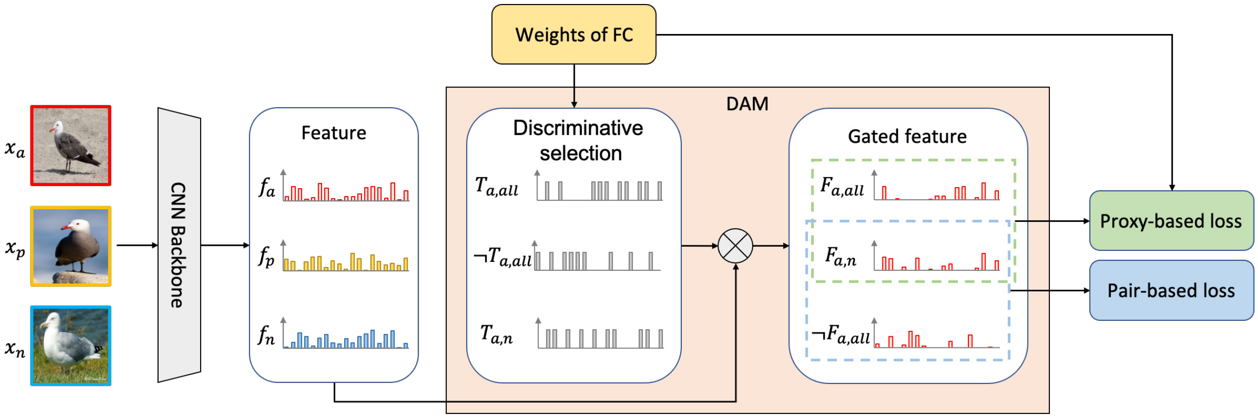

Figure 3. The framework of representation learning with the proposed discrimination-aware mechanism. We use the weights of the last FC layer to represent class center, and the difference of class center to determine discriminativeness of feature elements. Then the new gated features are inputed to proxy-based and pair-based loss for further parameters optimation.

feature elements compared with the non-gt class. This differentiated process further distinguish the learned features from all other classes. Therefore, the robustness of the overall feature representation is improved.

Discrimination-Aware Mechanism for Pair-base Loss. For pair-based loss, we choose triplet loss (*e.g.*, TriHard [18]) as our baseline. A triplet (x_a, x_p, x_n) has a shared anchor x_a, where x_a and x_p are positive pairs, while x_a and x_p are negative pairs. For triplet loss, the feature is directly used to calculate the distance, and the optimization goal is achieved by pulling the distance of the same class close and pushing the distance of the different classes far away. When use our DAM to TriHard, we can get the new gated features to continuously improve discriminativeness. In triplet loss based discrimination-aware mechanism (TriHardDAM), the triplets are harder than the previous ones for distance optimization. The TriHardDAM loss function can be rewrited as follows:

$$\mathcal{L}_{TriHardDAM} = \frac{1}{N}\sum_{a=1}^{N}[d(\neg F_{a,all}, \neg F_{p,all}) - d(F_{a,n}, F_{n,a}) + \alpha]^{+} \tag{9}$$

where $[.]^{+} = max(., 0)$, α is a pre-defined value, and $d(.)$ is the Euclidean distance calculation function, $\neg F_{a,all}, \neg F_{p,all}$ express the high-discriminativeness elements of feature f_a and f_p. To be specific, the gated mechanism sets the feature elements between intra-class with large differences to 1, and the other elements to 0.

The difference between $\mathcal{L}_{TriHard}$ and $\mathcal{L}_{TriHardDAM}$ is that we use the new gated features for distance optimization. For positive pair (f_a, f_p), we adopt features that are discriminative to all other class, which makes the distance relatively larger. For negative pair (f_a, f_n), we use the elements that has low discriminativeness between class c_a class c_n, which makes the distance relatively smaller. The new triplet has larger postive distance and smaller negitive distance, which makes learning goals more difficult. So our $\mathcal{L}_{TriHardDAM}$ can obtain better performance when the model is converged.

Discrimination-Aware Mechanism based Loss. By using the informative features to proxy-based loss and pair-based loss, we can get the overall loss by using DAM as:

$$\mathcal{L}_{DAM} = \mu L_{SCEDAM} + \nu L_{TriHardDAM} \tag{10}$$

where μ and ν are two adjustable parameters, representing the weights of L_{SCEDAM} and $L_{TriHardDAM}$ respectively.

In general, our DAM appropriately increases the learning difficulty during the training, thus improving the generalization ability of the model after convergence.

3.3. Representation Learning Framework

The architecture supervised by our DAM based loss is illustrated in Figure 3. Following the current popularity, we use CNN backbone to extract feature, and input the features to proxy-based loss *e.g.* SCE [17] and pair-based loss *e.g.* TriHard [18] to optimize model parameters. When loss convergence, we can get a feature extractor for inference. The inputs x_a, x_p, x_n from one training batch include samples from the same class and different calss, where x_a and x_p are from the same class, x_a and x_n are different class. During test phase, only feature embeddings are used for similarity comparison between samples. For simplify, we describe the training process of x_a in the schematic diagram.

For the features from DCNNs, we use weights of the last FC layer to get differences. Then features are gated by the DAM using these differences. For proxy-based loss, the elements with low-discriminativeness is retrained for further model optimization. For pair-based loss, we select high-discriminativeness elements for positive pairs while low-discriminativeness elements for negative pairs, which can increase the triplet learning difficulty during the training, thus improving the generalization ability once convergence.

3.4. Optimization for DAM based Loss

For SCEDAM loss, we can get the partial derivative of $\mathcal{L}_{SCEDAM}$ for the k-th element of weight w_i^k with respect to the following:

$$\frac{\partial \mathcal{L}_{SCEDAM}}{\partial w_i^k} = (\sum_{j=1}^{C} \frac{F_{i,j}^k w_i^k}{\sum_{l=1}^{C} F_{i,j}^k w_l^k} - 1) f_i^k \tag{11}$$

From the result of Equ 11, we can see that the elements of a feature with high-discriminativeness have smaller gradient to weight w, which shows that the optimization of SCEDAM loss slightly change the values of these elements in this iteration. On the contrary, the elements with low-discriminativeness have larger gradient. This difference makes the model parameters are further optimized to improve the element with low-discriminativeness. Therefore, our DAM can obtain more cues for better robust representation when loss convergence.

For TriHardDAM loss, when $d(\neg F_{a,all}, \neg F_{p,all}) > d(F_{a,n}, F_{n,a}) - \alpha$, we can get the partial derivative of $\mathcal{L}_{TriHardDAM}$ for f_a^k with respect to:

$$\frac{\partial \mathcal{L}_{TriHardDAM}}{\partial f_a^k} = \begin{cases} f_n^k - f_a^k, & T_{a,all}^k = 1, T_{a,n}^k = 1 \\ f_a^k - f_p^k, & T_{a,all}^k = 0, T_{a,n}^k = 0 \\ f_n^k - f_p^k, & T_{a,all}^k = 0, T_{a,n}^k = 1 \\ 0, & T_{a,all}^k = 1, T_{a,n}^k = 0 \end{cases} \tag{12}$$

According to Equ 12, the gradient of f_a^k is determined by the value of $T_{a,all}^k$ and $T_{a,n}^k$. When f_a^k has high-discriminativeness to all other classes and class c_n, it has smaller gradient ($f_a^k - f_p^k$). While f_a^k has low-discriminativeness to all other classes and class c_n, it has larger gradient ($f_n^k - f_a^k$). This difference in optimization direction can make the elements with low-discriminativeness become more discriminative.

4. Experiments

4.1. Datasets and Settings

Datasets. In this work, we use CUB-200-2011 [51], Cars196 [31], Market-1501 [65] and MSMT17 [54] to validate the effectiveness of our method. CUB-200-2011 is a dataset consists of various birds species. Following the standard splits [37], we use 5,864 images of the first 100 species for training and the rest 100 species for testing. Cars196 is a dataset contains 16,185 car images of 196 classes. We use the first 98 classes (8,054 images) for training and the rest for testing. Market-1501 and MSMT17 are person dataset, Market-1501 contains 32,668 images of 1,501 identities. It is split into 751 identities for training and 750 identities for testing. MSMT17 contains 126,441 images of 4,101 identities. Where 1,041 identities are in the training set and the rest 3,060 identities are in the testing set.

Table 1. Comparison with baselines on CUB-200-2011 and Cars196.

Method	CUB-200-2011		Cars196	
	r=1	r=2	r=1	r=2
SCE	69.8	79.4	86.9	91.8
SCEDAM	70.9	80.4	88.1	92.5
SCE+TriHard	71.2	80.7	88.5	92.8
SCE+TriHardDAM	71.9	81.0	88.6	93.1
SCEDAM+TriHard	71.8	80.8	88.4	93.0
DAM	72.3	81.2	88.9	93.4

Table 2. Comparison with baselines on Market-1501 and MSMT17.

Method	Market-1501		MSMT17	
	mAP	r=1	mAP	r=1
SCE	85.8	93.8	48.6	70.9
SCEDAM	86.7	94.8	53.2	75.6
SCE+TriHard	87.8	94.9	56.1	79.3
SCE+TriHardDAM	88.1	95.1	60.2	83.9
SCEDAM+TriHard	88.4	95.4	59.8	82.1
DAM	88.9	96.1	61.6	84.2

Implementation Details. Our method is implemented using PyTorch. For CUB-200-2011 and Cars196 dataset, we use serveral different settings: 64/512 embedding dimensions with the default image size (224×224), and the larger image size (256×256) for 512 embedding dimension also used. For Market-1501 and MSMT17 dataset, we resize images to 384 $\times$ 128, and 2,048 dimensions' features are obtained for training and testing. When our DAM used to proxy-based and pair-based loss, we set μ and ν in Equ 10 to 1 for simplification.

Evaluation Protocol. We adopt the same standard evaluation protocol as public works, *e.g.,* Recall@K and mean Average Precision (mAP), to evaluate the effect of the proposed method. The training set and testing set of all datasets do not overlap in classes. For CUB-200-2011 and Cars196, K={1, 2, 4, 8}. For Market-1501 and MSMT17, we use mAP and Rank-1 to evaluate the effect of our method.

4.2. Comparison with Baselines

Our DAM has improved feature extraction ability for proxy-based loss and pair-based loss by using DAM to obatin the new gated features. To show the effect of our

Table 3. Comparison with state-of-the-arts on CUB-200-2011 and Cars196. Superscripts denote embedding sizes and $^\sharp$ indicates models using 256 × 256 input images. Different backbones are abbreviated as: G–GoogleNet [47], BN–Inception with batch normalization [26].

Method	Backbone	CUB-200-2011				Cars196			
		r=1	r=2	r=4	r=8	r=1	r=2	r=4	r=8
Clustering^{64} [36]	BN	48.2	61.4	71.8	81.9	58.1	70.6	80.3	87.8
Proxy-NCA^{64} [35]	BN	49.2	61.9	67.9	72.4	73.2	82.4	86.4	87.8
Smart Mining^{64} [16]	G	49.8	62.3	74.1	83.3	64.7	76.2	84.2	90.2
MS^{64} [52]	BN	57.4	69.8	80.0	87.8	77.3	85.3	90.5	94.2
SoftTriple^{64}	BN	60.1	71.9	81.2	88.5	78.6	86.6	91.8	95.4
Proxy-Anchor^{64}	BN	61.7	73.0	81.8	88.8	78.8	87.0	92.2	95.5
DAM^{64}	BN	62.2	73.8	82.6	89.2	79.5	87.6	92.6	95.8
A-BIER^{512} [38]	G	57.5	68.7	78.3	86.2	82.0	89.0	93.2	96.1
ABE^{512} [30]	G	60.6	71.5	79.8	87.4	85.2	90.5	94.0	96.1
HTL^{512} [12]	BN	57.1	68.8	78.7	86.5	81.4	88.0	92.7	95.7
RLL-H^{512} [53]	BN	57.4	69.7	79.2	86.9	74.0	83.6	90.1	94.1
MS^{512} [52]	BN	65.7	77.0	86.3	91.2	84.1	90.4	94.0	96.5
SoftTriple^{512} [39]	BN	65.4	76.4	84.5	90.4	84.5	90.7	94.5	96.9
Proxy-Anchor^{512} [29]	BN	68.4	79.2	86.8	91.6	86.1	91.7	95.0	97.3
DAM^{512}	BN	69.1	79.8	87.2	91.8	86.9	92.1	95.3	97.9
$^\sharp$Contra+HORDE^{512} [27]	BN	66.3	76.7	84.7	90.6	83.9	90.3	94.1	96.3
$^\sharp$Proxy-Anchor^{512} [29]	BN	71.1	80.4	87.4	92.5	88.3	93.1	95.7	97.5
$^\sharp$DAM^{512}	BN	72.3	81.2	87.8	92.7	88.9	93.4	96.0	97.7

method, we conduct experiments on different setting.

For CUB-200-2011 and Cars196, we adopt SCE and SCE+TriHard as baselines. The experimental results are shown in Table 1. From the table, we can see that whether the DAM is adopted to proxy-based or pair-based loss, the accuracy is improved. Specially, when DAM imposed to proxy-based SCE on CUB-200-2011, SCEDAM gains +1.1% in Rank-1. For Cars196, our SCEDAM gains +1.2% in Rank-1. When our DAM imposed to pair-based TriHard loss, compared with SCE+TriHard, our SCE+TriHardDAM gains +0.7% in Rank-1 on CUB-200-2011, and +0.1% in Rank-1 on Cars196. When DAM is applied to both SCE and TriHard, we obtain a further improvement, which shows that our DAM can help learn more robust representations.

For Market-1501 and MSMT17, we also use proxy-based SCE and pair-based SCE+TriHard as baselines. The experimental results shown in Table 2 demonstrate the effect of our DAM on person re-identification (ReID) task. From the table, we can see that when our DAM used to SCE, SCEDAM gains +0.9% in mAP and +1.0% in Rank-1 on Market-1501. On MSMT17, compared with SCE, our SCEDAM gains +4.6% in mAP and +4.7% in Rank-1. When our DAM used to pair-based loss, compared with SCE+TriHard, our SCE+TriHardDAM gains +0.3% in mAP and +0.2% in Rank-1 on Market-1501, and +4.1% in mAP and +4.6% in Rank-1 on MSMT17. When DAM is applied to both SCE and TriHard, we obtain a further improvement (red color in Table 2). The results show that our DAM has better performance on both proxy-based and paired-based loss for person ReID.

4.3. Comparision with State-of-the-art Methods

In Table 3, we compare our method with state-of-the-arts on CUB-200-2011 and Cars196 dataset. For fair comparison, we use a variety of backbone networks, embedding sizes and image sizes to conduct experiments. From the table, we can see that our DAM surpasses existing methods in all indicators. Specifically, when the embedding size is 64, our DAM gains +0.5% in Rank-1 on CUB-200-2011, and gains +0.7% in Rank-1 on Cars196 than Proxy-Anchor [29]. When the embedding size is 512, we get 69.1% in Rank-1 on CUB-200-2011 and 86.9% in Rank-1 on Cars196, which are better than existing methods. For a larger image input size 256×256, we also get stronger performance and surpass the other methods. The improvements show that our method is effective to extract discriminative representation for fine-grained recognition.

For person ReID task, we compare our method to other works on Market-1501 and MSMT17, the results are shown in Table 4 and 5 respectively. From the Table 4, we can see that our method can achieve improvements than existing methods on Market-1501. Specifically, our DAM get 88.9% in mAP and 96.1% in Rank-1, which is comparable to other methods. When we compare our method with related methods on MSMT17, our method is also better than other works. Specifically, our DAM get 61.6% in mAP

Table 4. Comparison with state-of-the-arts on Market-1501.

Method	mAP	r=1
M^3+ ResNet50 [68]	82.6	95.4
RGA-SC [62]	88.4	96.1
SCSN(4 stages) [4]	88.3	92.4
ABDNet [3]	88.3	95.6
Pyramid [63]	88.2	95.7
DCDS [1]	85.8	94.8
MHN(PCB) [2]	85.0	95.1
BFE [8]	86.2	95.3
CASN(PCB) [66]	82.8	92.4
AANet [50]	83.4	93.9
IANet [19]	83.1	94.4
VPM [45]	88.8	93.0
DAM	88.9	96.1

Table 5. Comparison with state-of-the-arts on MSMT17.

Method	mAP	r=1
M^3+ ResNet50 [68]	55.0	72.8
RGA-SC [62]	57.5	80.3
SCSN(4 stages) [4]	58.5	83.8
ABDNet [3]	60.8	82.3
IANet [19]	46.8	75.5
DAM	61.6	84.2

and 84.2% in Rank-1. The improvement shows that our DAM also can extract discriminative representation for person ReID scenario.

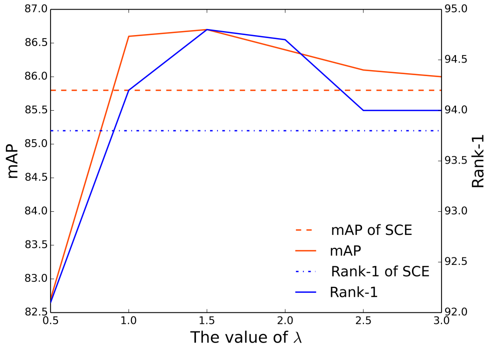

Figure 4. The influence of various λ for our DAM on Market-1501.

Table 6. Comparison with feature erasing based methods on Market-1501.

	mAP	r=1	r=5	r=10
SCE	85.8	93.8	97.9	98.8
SCE+Dropout [43]	86.3	94.4	98.1	98.8
SCE+DropBlock [13]	86.0	94.1	98.0	98.8
SCEDAM	86.7	94.8	98.4	98.9

4.4. Ablation Study

The effect of parameter λ. In this work, parameters λ (Equ 2) is used to determine which features are beneficial for further model optimization. We conduct experiments on various λ, the experimental results are shown in Figure 4. From the figure, we can see that when λ=1.5, SCEDAM has best result, *e.g.,* 86.7% in mAP and 94.8%, which achieves +0.9% in mAP and 1.0% in Rank-1 compared with SCE on Market-1501. When λ is small, *e.g,* 0.5, the effect is lower than SCE. This is because the smaller λ is, the fewer feature dimensions are selected. Feature with too few dimensions can not adequately represent the semantic information of the images, resulting disadvantaged results. On the contrary, when λ is too large, there are some redundant features with limited benefit to distinguish examples.

Comparison with erasing methods. Our DAM erase some elements of feature to enhence discriminativeness, which is similar to some feature erasing method. To show the effec of our DAM, we conduct experiments to compare with the related methods on Market-1501. The comparison results are shown in Table 6. From the table, we can see that our DAM is better than both Dropout [43] and DropBlock [13]. And the three erasing methods are higher than SCE. Specifically, compared with Dropout, our DAM gains +0.4% in mAP and +0.4% in Rank-1. Compared with DropBlock [13], our DAM gains +0.7% in mAP and 0.7% in Rank-1. The results show that erasing is an effective mechanism. The difference of our DAM to other two methods is that we erase some elements guided by the discriminativeness of feature elements, but the other two methods randomly erase some values. which shows that information to guide erasing location is beneficial.

5. Conclusion

In this paper, we proposed a discrimination-aware mechanism for proxy-based and pair-based loss, which improves the model generalization ability. To be specific, DAM can guide the model to learn more cues to enhance the discriminativeness of features. By using the proposed DAM, we erase elements of features with high-discriminativeness, and the low-discriminativeness elements are retained for further optimization. Eventually, each element of a feature becomes discriminative, resulting in a robust feature representation for fine-grained recognition. In the future work, we will move towards choosing more effective features to improve discriminativeness, and generalize it to other fields.

References

[1] Leulseged Tesfaye Alemu, Marcello Pelillo, and Mubarak Shah. Deep constrained dominant sets for person re-identification. In *ICCV*, 2019.

[2] Binghui Chen, Weihong Deng, and Jiani Hu. Mixed high-order attention network for person re-identification. In *ICCV*, 2019.

[3] Tianlong Chen, Shaojin Ding, Jingyi Xie, Ye Yuan, Wuyang Chen, Yang Yang, Zhou Ren, and Zhangyang Wang. Abd-net: Attentive but diverse person re-identification. In *ICCV*, 2019.

[4] Xuesong Chen, Canmiao Fu, Yong Zhao, Feng Zheng, Jingkuan Song, Rongrong Ji, and Yi Yang. Salience-guided cascaded suppression network for person re-identification. In *CVPR*, 2020.

[5] Yue Chen, Yalong Bai, Wei Zhang, and Tao Mei. Destruction and construction learning for fine-grained image recognition. In *CVPR*, 2019.

[6] Yunpeng Chen, Jianan Li, Huaxin Xiao, Xiaojie Jin, Shuicheng Yan, and Jiashi Feng. Dual path networks. In *NIPS*, 2017.

[7] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, 2017.

[8] Zuozhuo Dai, Mingqiang Chen, Xiaodong Gu, Siyu Zhu, and Ping Tan. Batch dropblock network for person re-identification and beyond. In *ICCV*, 2019.

[9] Zuozhuo Dai, Mingqiang Chen, Siyu Zhu, and Ping Tan. Batch feature erasing for person re-identification and beyond. *arXiv preprint arXiv:1811.07130*, 2018.

[10] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[11] Yao Ding, Yanzhao Zhou, Yi Zhu, Qixiang Ye, and Jianbin Jiao. Selective sparse sampling for fine-grained image recognition. In *ICCV*, 2019.

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

[13] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Dropblock: A regularization method for convolutional networks. In *NeurIPS*, 2018.

[14] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, 2018.

[15] Kai Han, Jianyuan Guo, Chao Zhang, and Mingjian Zhu. Attribute-aware attention model for fine-grained representation learning. In *ACM MM*, 2018.

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

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

[18] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *ICCV*, 2017.

[19] Ruibing Hou, Bingpeng Ma, Hong Chang, Xinqian Gu, Shiguang Shan, and Xilin Chen. Interaction-and-aggregation network for person re-identification. In *CVPR*, 2019.

[20] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[21] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018.

[22] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[23] Zixuan Huang and Yin Li. Interpretable and accurate fine-grained recognition via region grouping. In *CVPR*, 2020.

[24] Dat Huynh and Ehsan Elhamifar. Fine-grained generalized zero-shot learning via dense attribute-based attention. In *CVPR*, 2020.

[25] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

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

[27] Pierre Jacob, David Picard, Aymeric Histace, and Edouard Klein. Metric learning with horde: High-order regularizer for deep embeddings. In *ICCV*, 2019.

[28] Ruyi Ji, Longyin Wen, Libo Zhang, Dawei Du, Yanjun Wu, Chen Zhao, Xianglong Liu, and Feiyue Huang. Attention convolutional binary neural tree for fine-grained visual categorization. In *CVPR*, 2020.

[29] Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. Proxy anchor loss for deep metric learning. In *CVPR*, 2020.

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

[31] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCVW*, 2013.

[32] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[33] Wei Luo, Xitong Yang, Xianjie Mo, Yuheng Lu, Larry S Davis, Jun Li, Jian Yang, and Ser-Nam Lim. Cross-x learning for fine-grained visual categorization. In *ICCV*, 2019.

[34] Oisin Mac Aodha, Elijah Cole, and Pietro Perona. Presence-only geographical priors for fine-grained image classification. In *ICCV*, 2019.

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

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

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

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

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

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

[41] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[42] Chunfeng Song, Yan Huang, Wanli Ouyang, and Liang Wang. Mask-guided contrastive attention model for person re-identification. In *CVPR*, 2018.

[43] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 2014.

[44] Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. In *CVPR*, 2020.

[45] Yifan Sun, Qin Xu, Yali Li, Chi Zhang, Yikang Li, Shengjin Wang, and Jian Sun. Perceive where to focus: Learning visibility-aware part-level features for partial person re-identification. In *CVPR*, 2019.

[46] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:1602.07261*, 2016.

[47] 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.

[48] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

[49] Luming Tang, Davis Wertheimer, and Bharath Hariharan. Revisiting pose-normalization for fine-grained few-shot recognition. In *CVPR*, 2020.

[50] Chiat-Pin Tay, Sharmili Roy, and Kim-Hui Yap. Aanet: Attribute attention network for person re-identifications. In *CVPR*, 2019.

[51] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[52] 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.

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

[54] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer gan to bridge domain gap for person re-identification. In *CVPR*, 2018.

[55] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017.

[56] Wei Xiong, Yutong He, Yixuan Zhang, Wenhan Luo, Lin Ma, and Jiebo Luo. Fine-grained image-to-image transformation towards visual recognition. In *CVPR*, 2020.

[57] Furong Xu, Bingpeng Ma, Hong Chang, and Shiguang Shan. Isosceles constraints for person re-identification. *TIP*, 2020.

[58] Furong Xu, Wei Zhang, Yuan Cheng, and Wei Chu. Metric learning with equidistant and equidistributed triplet-based loss for product image search. In *WWW*, 2020.

[59] Lianbo Zhang, Shaoli Huang, Wei Liu, and Dacheng Tao. Learning a mixture of granularity-specific experts for fine-grained categorization. In *ICCV*, 2019.

[60] Xingcheng Zhang, Zhizhong Li, Chen Change Loy, and Dahua Lin. Polynet: A pursuit of structural diversity in very deep networks. In *CVPR*, 2017.

[61] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *CVPR*, 2018.

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

[63] Feng Zheng, Cheng Deng, Xing Sun, Xinyang Jiang, Xiaowei Guo, Zongqiao Yu, Feiyue Huang, and Rongrong Ji. Pyramidal person re-identification via multi-loss dynamic training. In *CVPR*, 2019.

[64] Heliang Zheng, Jianlong Fu, Zheng-Jun Zha, and Jiebo Luo. Looking for the devil in the details: Learning trilinear attention sampling network for fine-grained image recognition. In *CVPR*, 2019.

[65] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *ICCV*, 2015.

[66] Meng Zheng, Srikrishna Karanam, Ziyan Wu, and Richard J Radke. Re-identification with consistent attentive siamese networks. In *CVPR*, 2019.

[67] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *AAAI*, 2020.

[68] Jiahuan Zhou, Bing Su, and Ying Wu. Online joint multi-metric adaptation from frequent sharing-subset mining for person re-identification. In *CVPR*, 2020.