Information Sciences 643 (2023) 119120

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Multi-proxy based deep metric learning

Patrick P.K. Chan [a,*], Shute Li [b], Jingwen Deng [a], Daniel S. Yeung [c]

[a] Shien-Ming Wu School of Intelligent Engineering, South China University of Technology, Guangzhou, China
[b] School of Computer Science and Engineering, South China University of Technology, Guangzhou, China
[c] Hong Kong

ARTICLE INFO

Keywords:
Deep learning
Deep metric learning
Proxy-based method
Multi-proxy
Image retrieval

ABSTRACT

Deep metric learning (DML) achieves excellent performance in many open-set scenarios. However, the current multi-proxy methods rely on a classification framework and the performances of these methods rely on large batch size due to the representation limitation of mini-batch for neighbor distribution of proxies. This study proposes a multi-proxy method aiming to enhance the generalization ability of DML on unseen classes by enhancing the data representation ability. As a sample may not share all fine-grained semantic information of a class, our model does not require full association between samples and proxies, and is updated only according to the associated pairs. Moreover, a mini-batch may not contain sufficient information for learning. So our model also considers the proxy-proxy relation in order to provide a global view of the data structure which improves the learning on the intra-class distance. This study also investigates how the positive and negative margins, *i.e.*, the parameters of our model which control the required similarity for a class and different classes, affect our performance, and finds that less demanding on the negative margin avoids over-training and improves generalization ability to unseen classes. The outstanding performance of our model is confirmed by experimental results compared with the state-of-the-art DML methods in the benchmark image retrieval datasets.

1. Introduction

Deep metric learning (DML) [1] is an algorithm to establish a metric that aims to quantify the similarity between objects using deep neural networks. Different from the traditional numerical methods for feature selection [2,3], deep learning generates discriminative features of samples by learning the non-linear mapping with the artificial neural network. DML has demonstrated excellent performance in many open-set scenarios, for example, person re-identification [4], and few-shot classification [5].

A pair-based method [6,7] is one of the most popular DML methods. Informative sample pairs are chosen from a mini-batch in training. Samples in the same class are pulled closer, while those from different classes are pushed away. However, when the training size is large, the training complexity may increase exponentially [8]. Moreover, the training efficiency heavily depends on a sampling strategy since some sample pairs may not provide useful information.

A proxy-based method tackles these problems by providing a global view of a training set for each mini-batch [9,10]. A proxy or multiple proxies are assigned to a class to represent its data distribution and reside in the memory during the whole training. One drawback of using a single proxy [10] for a class is the limitation of the representation ability. A complicated class contains various characteristics and its distribution may not be like a hypersphere in a high-dimensional space. As a result, representing a complicated

* Corresponding author.
E-mail address: patrickchan@ieee.org (P.P.K. Chan).

https://doi.org/10.1016/j.ins.2023.119120
Received 30 December 2022; Received in revised form 10 April 2023; Accepted 4 May 2023

Available online 12 May 2023

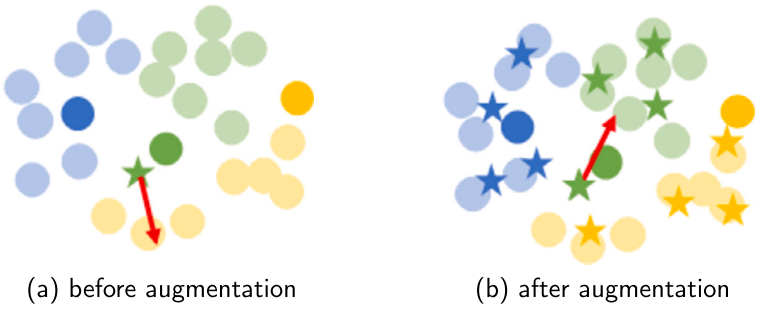

Fig. 1. Influence of data augmentation impact on the movement of proxy. The dark circles represent samples sampled in the mini-batch, and the light circles represent the distribution of other samples in the entire dataset. Pentagrams represent proxies. Different colors represent different categories. The augmentation by the global view provided by the proxies to the local view of the samples in a mini-batch causes a more desirable update.

class by only one proxy not only increases the training difficulty but also reduces generalization ability. Some multi-proxy methods, such as SoftTriple [8] and ProxyGML [11], are proposed recently. These methods consider a classification loss, which generally leads the samples to get closer to their class and further to other classes. It may cause over-training easily and does not benefit an open-set scenario in which classes in inference may be different from the ones in training.

This study proposes a multi-proxy method aiming to enhance the generalization ability of DML on unseen classes by representing the data distribution properly. A number of proxies are assigned to each class to represent its fine-grained semantics in our method. As a sample may not has all fine-grained semantics of its class, the sample should not be required to associate with all proxies of its class (called positive proxies), *e.g.*, the head and tail of a truck share few common characteristics. If a positive proxy is far away from a sample, our model assumes that they do not share the same characteristics and will not associate with them. On the other hand, a proxy with a different class to a sample (*i.e.*, a negative proxy) is associated with the sample only if they are located closely. Only the associated proxies contribute to training. Because samples are only associated with proxies partially, a proxy represents only some characteristics of a class. Therefore, the variety of inner-class structures can be represented more easily and better.

The data distribution may not be represented by a mini-batch since it only contains a few training samples, which may lead to improper updates on a proxy. It is explained in an example shown in Fig. 1a. A circle and pentagram indicate a sample and proxy, and classes are represented by different colors. The ones with dark yellow, blue, and green are selected in a mini-batch in training. As those selected samples cannot represent the whole data distribution well, the green proxy may tend to the yellow class in an update. The improper update may not only cause the slow convergence but also the local optima in learning. Our model uses the proxies which represent the structures of all classes to supplement a mini-batch. The data augmentation provided by proxies offers a global view of the data structure in addition to the local information provided by the samples in a mini-batch. Since proxies are resident during the whole training process, the local and global views on the data structure are obtained in each update in our model, which increases the efficiency of learning. In our example due to the additional information provided by the proxies, the update in Fig. 1b is more desirable.

This study also investigates how the margins affect the performance of our model. Similar to other proxy-based methods, our method also contains two margins (*i.e.*, positive and negative ones) to control the expected similarity for a class and different classes. Some studies [6,9,10] suggest that the negative margin should be larger than the positive one in order to separate the classes clearly. However, our experimental studies find that a small difference between the positive and negative margins is preferable for the generalization ability in our model. Less demanding on the negative margin avoids the over-training and encourages learning on the similar characteristics of different classes, which is beneficial to separate unseen classes. As a result, a hard sample may be retrieved more accurately. Moreover, our method is evaluated and compared with the state-of-the-art methods of pair-based, single-proxy, and other multi-proxy methods experimentally.

The generalization ability of deep metric learning is enhanced by boosting the data representation ability and determining a proper setting of margins in this study. The contributions of our study are as follows:

- A multi-proxy DML is proposed. Samples and proxies are associated partially and only the associated pairs are considered in an update. The structure of data can be presented more precisely by our model than by existing methods and our model has a better generalization ability in an open-set scenario.
- A mini-batch may only provide local information for learning due to a lack of training samples. This study devises an objective function which considers not only the sample-proxy relation but also the proxy-proxy relation providing a global view of the data structure. The distance of intra-class can be learned more efficiently.
- The influence of margins on the generalization ability of our method is explored. The performance of our method increases with the decrease in the difference between the positive and negative margins.

The rest of this paper is organized as follows. Section 2 introduces the basic concepts of DML related to this study, and our proposed model is devised and discussed in Section 3. Section 4 provides the experimental results and discussion. Finally, the conclusion and future work are given in Section 5.

2. Background

The related concepts of this study, including pair-based and proxy-based metric learning, distance-based and classification-based loss functions, and single-proxy and multiple-proxy methods, are discussed in this section.

2.1. Pair-based vs proxy-based method

A pair-based method generates sample pairs from a mini-batch. A pair containing the samples with the same label is pulled closer, while the one with different labels is separated apart. A classic representative is Triplet Loss [6,12], which refers to a triplet containing a sample as an anchor, and the other two samples chosen from the same class and different class from the anchor. The anchor gets closer to its positive sample and gets away from the negative one with a fixed margin simultaneously. Quadruplet [13], N-pair [7], and Lifted-Structure losses [14] interact a sample with more samples from the same or different classes in order to increase the stability of training. In order to solve the high training complexity, hard-sampling [15] and semi-hard sampling [6] have been proposed which only focus on the samples located in between two classes. Additional modules, *e.g.*, cross-batch memory (XBM) module [16] and Local Neighborhood Component Analysis (LNCA) module [17], are integrated into existing pair-based methods in order to extract useful information.

On the other hand, a proxy-based method [10,8,11,18,19] does not only focus on the sample-to-sample relation. Each class is represented by a proxy. The training process aims to reduce the distance between a sample and its same-class proxy, and increase the distance to the proxies from other classes. The advantages of a proxy-based method are low training complexity [10] and fast convergence [20]. Moreover, a proxy provides a global view of a class in each update and avoids the local optimum problem [9].

2.2. Distance-based vs classification-based loss function

A distance-based method [21] directly optimizes the distance between samples and samples or samples and the proxies. They are often accompanied by a margin, pulling the same labeled samples (or proxies) to make them closer than a fixed margin or pushing different labeled samples (or proxies) to make them farther than a fixed margin. On the other hand, in a classification-based method, one [22,23] or more proxies [11,24] are assigned to each class, and then SoftMax and CrossEntropy are utilized to discriminate samples in the mini-batch.

The margin-based methods can control the intra-class variety and inter-class discrimination through margins, but setting a fixed margin is not flexible due to different classes may have different intra-class varieties. To solve this problem, hierarchical metric learning [25] clusters the entire training data hierarchically, and sets different margins according to the hierarchical structure. However, this method faces the problems of complicated calculation and a lack of guarantee of the clustering effect.

Classification-based methods discriminate samples by optimizing the sample classification probability, thus avoiding the margin problem of distance-based methods. However, because the classification-based methods do not have a definite stopping boundary (margin) like the margin-based method, it will endlessly improve the discrimination ability between classes. Different classes are pushed as far as possible, which may prevent the model from learning the diverse features from the same class and the shared features between different classes, thereby impairing the generalization ability of the model.

2.3. Single proxy vs multiple proxies

A proxy provides a global view of a group of samples. Convergence and robustness of training can be improved by considering proxies in each training iteration. However, samples in the same class may be different from each other since the concepts represented by a class may be complicated. Thus, a single proxy is not enough to represent the intra-class structure well. To solve this problem, SoftTriple [8] loss assigns multiple proxies to each class as an ensemble of multiple weak classifiers to improve the performance [11] of metric learning. On the other hand, ProxyGML is also regarded as an ensemble strategy, which assigns multiple proxies for each class and reduces the number of proxies associated with each sample in order to the training complexity.

3. Proposed method

Our proposed multi-proxy DML is introduced in this section. The procedure of our model is illustrated by the flowchart shown in Fig. 2. In each update, an embedding vector of each sample in the mini-batch is extracted by a CNN model. Then the similarity between proxies and samples is calculated in the embedding space for the sample-proxy (Section 3.1) and proxy-proxy (Section 3.2) relations construction. After that, a sample's associated positive and negative proxies are identified according to the calculated similarities. Finally, the proxies and CNN model are updated in training by only considering associated positive and negative proxies' relations (Section 3.3). Since the data structure of each class is represented precisely by multiple proxies, a better performance should be achieved by our model.

3.1. Sample-proxy relation construction

Given a dataset $\mathbf{D} = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, where x_i and $y_i \in 1, 2, ..., c$ denote the ith sample features and the corresponding label, c is the number of classes, and n is the number of samples.

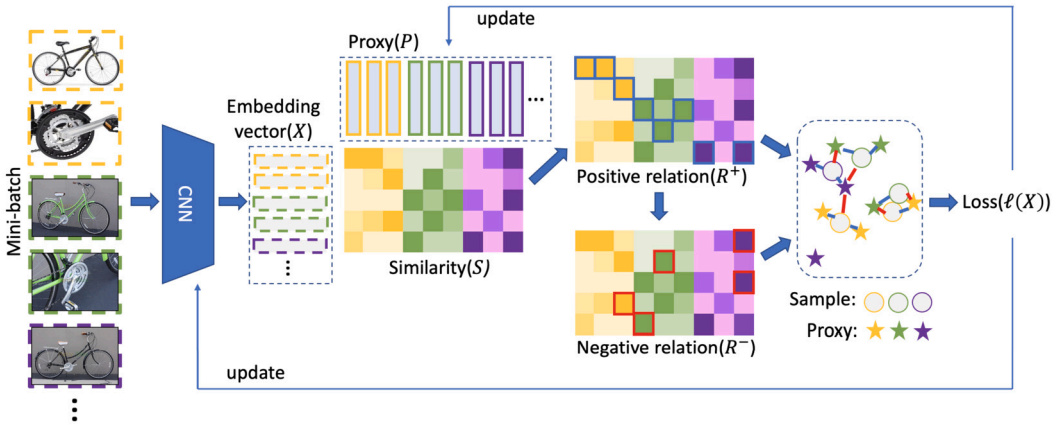

Fig. 2. The flowchart of our method. The similarity between samples in a mini-batch and all proxies is first calculated in the embedding space, and the positive and negative proxies are then identified. The proxies and CNN model are updated according to the chosen proxies only.

By assigning m proxies to each class, various semantics of a class can be represented more easily in our method. A set proxy $\mathbf{P}$ is defined as follows:

$$\mathbf{P} = \{p_j^i | i = 1, 2, ..., c, j = 1, 2, ..., m\}, \tag{1}$$

where p_j^i indicates the jth proxy of the ith class. There are $m \times c$ proxies in total. As a sample may only obtain some characteristics of a class, our method allows a sample only associates some proxies of the class. The connection between a sample and its positive proxies is identified by (2). The value υ_{x_i,p_j} represents how confident the sample x_i relates to the proxy $p_j^{y_i}$ in the high dimensional space of the DML model, where $p_j^{y_i}$ is the jth proxy of the same class as x_i.

$$\upsilon_{x_i,p_j} = \frac{e^{s(x_i,p_j^{y_i})/T}}{\sum_{k=1}^{m} e^{s(x_i,p_k^{y_i})/T}}, \tag{2}$$

where $s(a, b)$ is a cosine distance between a and b obtained from a DML model. T denotes the temperature parameter [26] indicating the confidence of a model. T indirectly determines the number of associated proxies for a sample. A lower temperature reduces the number of positive proxies to increase confidence, and vice versa.

The positive relation R^+ between a sample and proxies is defined according to υ_{x_i,p_j}. The association between a sample and a positive proxy is determined according to υ. When a sample (x_i) associates with its positive proxies (p_j), $R^+_{x_i,p_j} = 1$, otherwise, $R^+_{x_i,p_j} = 0$. R^+ is defined as (3).

$$R^+_{x_i,p_j} = \begin{cases} 1, & \text{if } \upsilon_{x_i,p_j} > \gamma \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

If the confidence value of the sample x_i and its jth positive proxy $p_j^{y_i}$ is larger than γ, *i.e.*, $\upsilon_{x_i,p_j} > \gamma$, x_i and $p_j^{y_i}$ are related. Otherwise, they are not related. γ denotes a threshold controlling the difficulty of association. When γ is large, less positive proxies are associated and the model allows larger inner-class variety. On the other hand, a class will be more compact as a sample relates to more positive proxies when γ is small.

Similarly, only negative proxies close to a sample are associated with our model. The remote proxies are not associated because further pushing negative proxies far away from a sample increases training complexity and may cause over-training. A negative proxy is associated with a sample x if its similarity is higher than the maximal similarity between x and its non-associated positive proxies. Noted that the criterion of choosing negative proxies is not fixed but is determined dynamically according to the similarity of positive proxies. It can be adaptive to different scenarios. The relation between a sample and its negative proxies is represented by the negative relation R^- defined as (4).

$$R^-_{x_i,p_j} = \begin{cases} 1, & \text{if } s(x_i, p_j) > \max\limits_{1 \le k \le m} s(x_i, p_k) \cdot (1 - R^+_{x_i,p_k}) \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

3.2. Proxy-proxy relation construction

In addition to the sample-proxy relation, our model also considers the proxy-proxy relation. Assume p_i and p_j represent the two different proxies belonging to the same class. The confidence (υ_{p_i,p_j}), positive and negative relations ($R^+_{p_i,p_j}$ and $R^-_{p_i,p_j}$) are calculated

according to (2), (3) and (4) respectively. The proxy-proxy relation is only applied to the negative relations in order to maintain the intra-class distances. During the update, two near proxies belonging to different classes are pushed away from each other. The proxy-proxy relation is defined as formula (5):

$$Q^{-}_{p_i,p_j} = \begin{cases} 1, & \text{if } s(p_i,p_j) > \max\limits_{1\le k\le n} s(x_k,p_i)\cdot(1-R^{-}_{x_k,p_i})\cdot\mathbb{I}(y_{x_k}\neq y_{p_i})\cdot\mathbb{I}(i\neq j) \\ 0, & \text{otherwise} \end{cases}, \tag{5}$$

where $s(\cdot)$ is a cosine distance between two proxies in the embedding space generated by the DML model. $\mathbb{I}(b)$ denotes an step function. When b is true, it is equal to 1; otherwise, it is 0. For each proxy, the different-class samples with the maximum similarity and not negatively associated are used as a reference for setting the threshold. Only the different-class proxies whose similarity is greater than the threshold are considered.

3.3. Loss function

The loss function used in our model consists of the components related to positive and negative relations, defined as follows:

$$\begin{aligned} \ell(X) = & \frac{1}{|P^{+}|}\sum_{p\in P^{+}}\log\left(1+\sum_{x\in X^{+}_{p}} R^{+}_{x,p}e^{-\alpha\cdot(s(x,p)-\delta^{+})}\right)+ \\ & \frac{1}{|\bar{P}^{-}|}\sum_{p\in \bar{P}^{-}}\log\left(1+\sum_{x\in X^{-}_{p}} R^{-}_{x,p}e^{\alpha\ cdot(s(x,p)-\delta^{-})}+\sum_{p'\in P^{-}_{p}} Q^{-}_{p',p}e^{\alpha\cdot(s(p',p)-\delta^{-})}\right), \end{aligned} \tag{6}$$

where P^{+} indicates the set of proxies that are positively associated with the samples in the mini-batch, and $\bar{P}^{-}$ indicates the set of proxies that are negatively associated after data augmentation. X^{+}_{p} and X^{-}_{p} indicate the same and different-class sample sets with p in the mini-batch respectively. P^{-}_{p} represents the proxy set that is different-class from the proxy p. δ^{+} and δ^{-} represent positive and negative margins aiming to control the expected similarity for a class and different classes. α is the scaling factor. The proxies (P) and DML model (s) are updated according to (6) iteratively.

A sample may not contribute to training when it is not associated with any positive and negative proxies. This kind of sample locates in a region that is not close to positive proxies and far away from negative proxies. When a sample is only associated with negative but not positive proxies, the sample is not close to its class and too close to other classes. Conversely, when a sample is close to its class but not others, it will associate only with its positive but not negative proxies. Only associated positive and negative proxies identified by R^{+} and R^{-} of a sample are considered in our loss function. This is because a sample may not share all characteristics of a class, especially the complicated ones. Moreover, only considering the negative proxies close to a sample in training avoids unnecessary updates, which may cause over-training and increase the complexity and instability.

Although the discrimination ability of a model increases when setting the positive margin larger and negative margin smaller [10,6,9], *i.e.*, $\delta^{+} = -\delta^{-}$ [10], this ability may not be generalized to novel classes in open-set scenarios. As a result, our study suggests the difference between δ^{+} and δ^{-} should be small in our model. By releasing the requirement on the negative margin, the model will not overlearn the difference between classes of the training set. It is expected that the generalization ability of our model will increase since more general characteristics between classes may be learned.

4. Experiments

This section evaluates and compares our method with the state-of-the-art (SOTA) methods on three benchmark datasets of image retrieval experimentally. Besides accuracy, this study also explores the impact of parameters and discusses the running time of our method.

4.1. Experimental settings

4.1.1. Datasets

Three benchmark datasets, including CUB-200-2011 [27], Cars196 [28], and Stanford Online Products (SOP) [14], of image retrieval is considered in our experiments. CUB-200-2011 and Cars196 are both commonly used datasets for a fine-grain classification task. CUB-200-2011 contains 11,788 images of 200 bird species collected from Flickr and filtered by Amazon Mechanical Turk, while Cars196 contains 16,185 2D images of 196 different cars manually collected for testing 3D object representation ability. SOP contains 120,053 images of the online products in eBay.com. The detailed information of each dataset is illustrated in Table 1. An open-set problem setting is used. Half classes are selected to form a training set and the rest of the classes are reserved as a test set for each dataset [10], and the classes in the training and test sets have no intersection.

4.1.2. Implementation detail

The parameters of our method are determined by maximizing the performance in preliminary experiments. 10 proxies are assigned to each class in Car196 and CUB200-2011, while 3 proxies are used in SOP. This is because the average number of samples per class is less than 5 in SOP. Moreover, in order to avoid the samples of a class being too scattered, temperature T is set as 1 in SOP, which

Table 1
Three Benchmark Datasets Used in the Experiments. Half classes are selected for training and the rest are for evaluation.

Dataset	Sample Number (Class Number)	
	Training Set	Test Set
CUB-200-2011 [27]	5,924 (100)	5,864 (100)
Cars196 [28]	8,054 (98)	8,131 (98)
SOP [14]	59,551 (11,318)	60,502 (11,316)

Table 2
Comparison Between Our proposed Methods and the SOTA Methods. The Results of the Comparison Methods Are Obtained From Their Publications. Column 'Type': 'A', 'P', and 'MP' Denotes Pair, Proxy, and Multi-Proxy Methods, While Column 'NN': 'G' and 'BN' Indicate GoogleNet [30] and Batch Normalized Inception Net [29]. 'R@k' Indicates the Success Rate of Retrieval on k Query Images. The Largest Value of Each Column Is Highlighted.

Method			Cars196			CUB200-2011			SOP		
Name	Type	NN	R@1	R@2	R@4	R@1	R@2	R@4	R@1	R@10	R@100
SemiHard [15]	A	BN	51.5	63.8	73.5	42.6	55.0	66.4	66.7	82.4	91.9
Clustering [31]	A	BN	58.1	70.6	80.3	48.2	61.4	71.8	67.0	83.7	93.2
LiftedStruct [14]	A	G	53.0	65.7	76.0	43.6	56.6	68.6	62.5	80.8	91.9
HDC [32]	A	G	73.7	83.2	89.5	53.6	65.7	77.0	69.5	84.4	92.8
HTL [25]	A	BN	81.4	88.0	92.7	57.1	68.8	78.7	57.1	68.8	78.7
DAMLRRM [33]	A	G	73.5	82.6	89.1	55.1	66.5	76.8	69.7	85.2	93.2
HDML [34]	A	G	79.1	87.1	92.1	55.1	66.5	76.8	68.7	83.2	92.4
MS [35]	A	BN	84.5	90.7	94.5	65.7	77.0	86.3	78.2	90.5	96.0
ProxyNCA [9]	P	BN	73.2	82.4	86.4	49.2	61.9	67.9	73.7	-	-
ProxyAnchor [10]	P	BN	86.1	91.7	95.0	68.4	79.2	86.8	79.1	90.8	96.2
SoftTriple [8]	MP	BN	79.1	87.1	92.1	65.4	76.4	84.5	78.0	90.3	95.9
ProxyGML [11]	MP	BN	84.1	90.4	94.0	66.6	77.6	86.4	78.0	90.6	96.2
Ours	MP	BN	**90.3**	**93.7**	**96.3**	**69.6**	**79.9**	**87.0**	**80.1**	**91.3**	**96.6**

encourages the association between samples and proxies. On the other hand, T is set as 32 in both Car196 and CUB200-2011. γ is set as $1/n$. Both δ^+ and δ^- are set as +0.15. The batch size is set as 60. The backbone of our method is BN Inception [29].

4.2. Comparison with state-of-the-art methods

Our method is compared with the SOTA methods of three kinds of DML methods: pair-based (A), single-proxy-based (P) and multi-proxy-based (MP) methods, and the results are shown in Table 2. Our method outperforms all other methods in all cases in the three datasets.

The performance of our method is significantly better than others in Car196. Our method achieves 4.2%, 2.0%, and 1.3% more than the second-best method, *i.e.*, ProxyAnchor, in R@1, R@2, and R@4. Each car in Cars196 has multiple viewpoints with special features, which may be shared by different cars with the same viewpoints. The results may indicate that our method generalizes the features shared between different classes and maintains the inner-class variety. Our method also successfully reduces the inner-class variety and learns class discrimination features in CUB200-2011 and SOP, *i.e.*, R@1 of the second-best method is improved by 1.2% and 1.0% in CUB200-2011 and SOP respectively.

4.3. Inner-class and intra-class similarity

Our method and the existing single-proxy and multi-proxy methods with the best performance, *i.e.*, ProxyAnchor and ProxyGML, are chosen for further investigation in terms of the inner-class and intra-class similarity measures. These models are trained by following the suggestions in their publications tightly.

A partition $\mathbf{D}_a$ of $\mathbf{D}$ is defined according to a class a, *i.e.*, $\mathbf{D}_a = \{(x, y)|(x, y) \in \mathbf{D}, y = a\}$. The average inter-class similarity (S_{inter}) and intra-class similarity (S_{intra}) are defined as follows:

$$S_{inter}(\mathbf{D}_a, \mathbf{D}_b) = \frac{\sum_{x_a \in \mathbf{D}_a} \sum_{x_b \in \mathbf{D}_b} \frac{x_a \cdot x_b}{\|x_a\| \|x_b\|}}{|\mathbf{D}_a||\mathbf{D}_b|} \tag{7}$$

and

$$S_{intra}(\mathbf{D}_a) = \frac{(\sum_{x_i \in \mathbf{D}_a} \sum_{x_j \in \mathbf{D}_a} \frac{x_i \cdot x_j}{\|x_i\| \|x_j\|} - |\mathbf{D}_a|)}{(|\mathbf{D}_a| - 1)(|\mathbf{D}_a| - 1)}, \tag{8}$$

Fig. 3 shows the inner-class and intra-class similarity on the training and test sets of Cars196. Both vertical and horizontal axes of each figure represent classes. Therefore, inner-class similarity values are shown in the diagonal while other regions indicate intra-class similarity values. Small and large values are shown in blue and red respectively.

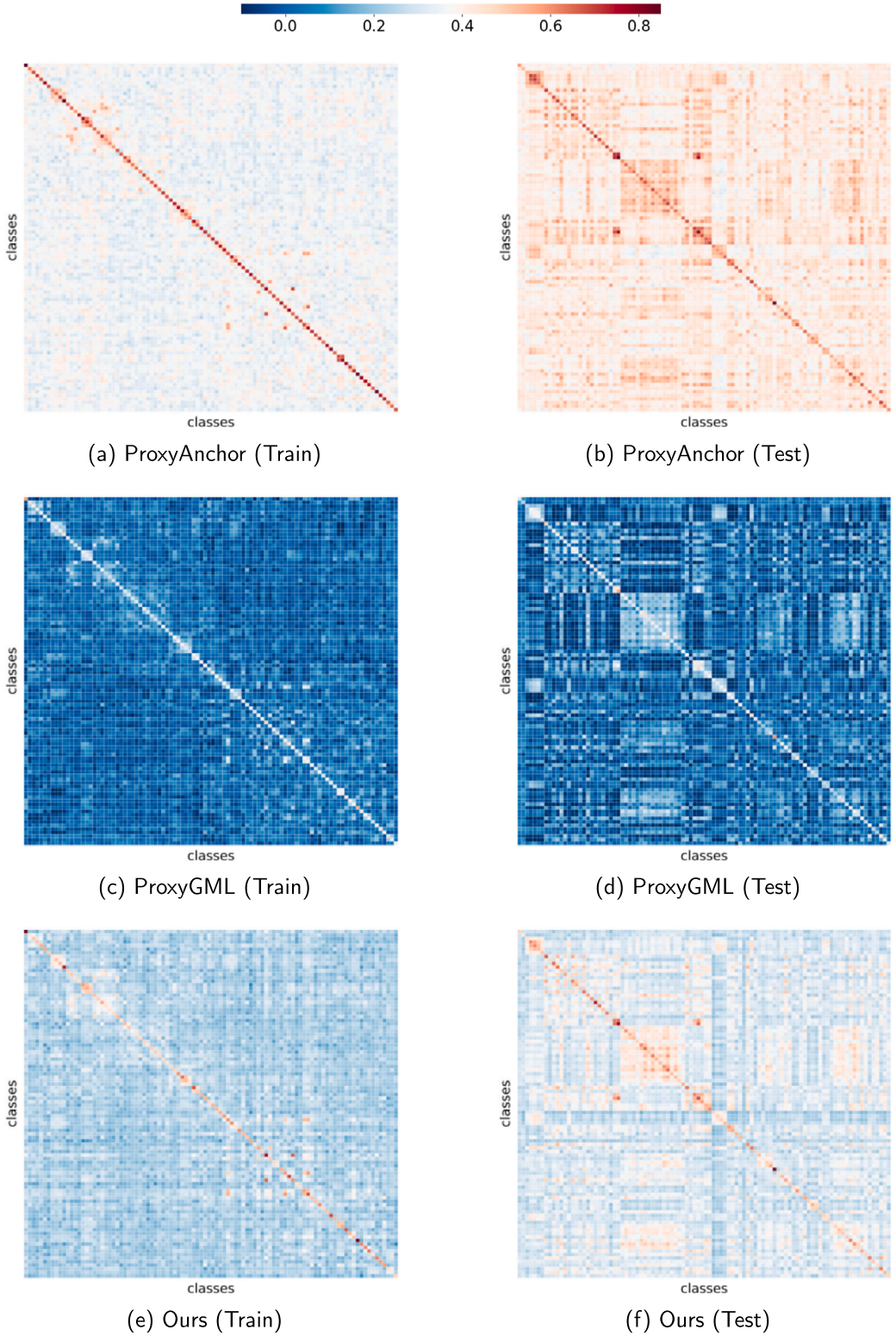

Fig. 3. Similarity value between classes of ProxyAnchor, ProxyGML and our method on the training and test sets of Cars196. The inner-class similarity is shown in the diagonal while the intra-class similarity is shown in other regions.

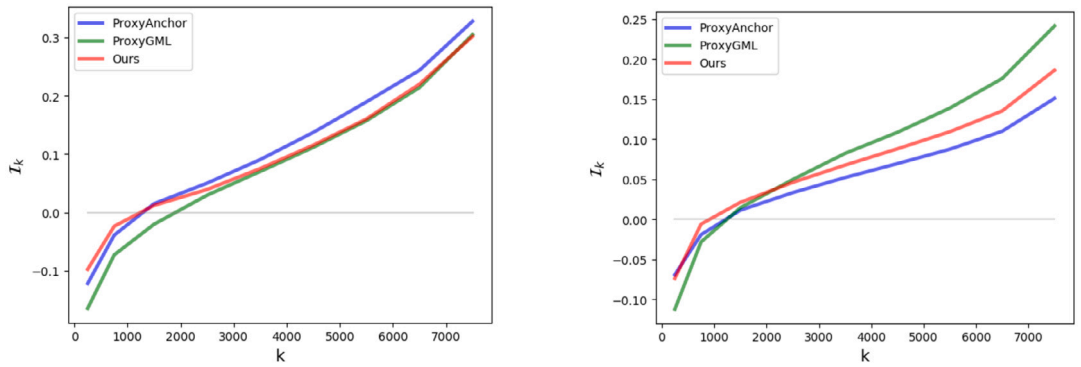

Fig. 4. Mean discrimination ability (I_k) with different k when $\upsilon = 250$ in the training and test sets of Cars196.

For ProxyAnchor, the inner-class similarity is high while the intra-class similarity is low in the training set, *i.e.*, the diagonal and other regions are in red and white respectively. However, (a) and (b) show that the generalization of ProxyAnchor from the training set to the test set is weak. The similarity values in non-diagonal values are large in the test set. On the other hand, both inner-class and intra-class similarities of ProxyGML are relatively smaller than the ones of ProxyAnchor. Similar to ProxyAnchor, the discriminative ability of ProxyGML on the test set is also weaker than that on the training set in (c) and (d). Compared with ProxyAnchor and ProxyGML, the inner-class similarity is much higher than the intra-class similarity for our method. Moreover, the discrimination ability is generalized well from the training to the test set. It explains why our method achieves better performance than others.

4.4. Sample discrimination ability

The discrimination ability of ProxyAnchor, ProxyGML, and our method is quantified. The discrimination ability $I(x)$ of a sample x is defined as the distance difference between the closest positive and negative samples of x, shown as follows:

$$I(x) = \max_{x' \in X^+} s(x, x') - \max_{x' \in X^-} s(x, x'), \tag{9}$$

where X^+ and X^- denote the sets containing samples with the same and different classes to x respectively, and s is the cosine distance. A negative $I(x)$ value means x is closer to its negative sample, *i.e.*, x is classified wrongly. I_k is defined as the mean discrimination ability on the samples ranked from $k - \upsilon$ to $k + \upsilon$ according to $I(x)$.

$$I_k = \frac{1}{2\upsilon} \sum_{k-\upsilon \le i \le k+\upsilon} I(x_{(i)}), \tag{10}$$

where $x_{(i)}$ denotes the ith sample sorted according to $I(x)$, *i.e.*, $I(x_{(1)}) \ge I(x_{(2)}) \ge I(x_{(3)}) \ge \ldots \ge I(x_{(n)})$, and υ represents the interval size. A larger I_k indicates better performance on the hard samples.

Higher discrimination ability on the hard samples is more important than on other samples since the hard samples may make mistakes easily. Fig. 4 shows I_k with different k values when υ is set as 250. When k is smaller, the samples with a shorter distance difference between its closest positive and negative samples are considered, *i.e.*, the samples are harder to be discriminated. Our method yields the largest I_k when k is less than and equal to 1,500 and 2,000 in the training and test sets respectively. It means that our method achieves stronger discriminative ability on the hard samples than other methods.

The discrimination ability of ProxyAnchor cannot be generalized from the training to test sets, *i.e.*, I_k drops dramatically on the test set. This may be because single-proxy methods cannot learn the various structures of classes, and extracted features cannot work well in a new domain. On the other hand, although ProxyGML achieves better performance on the test set than on the training set in general, it has the worst performance on the hard samples. It may be because it is a classification-based method that may not learn the similar characteristics between different classes. Thus it is easy to mistake them for the same class.

4.5. Visualization using t-distributed stochastic neighbor embedding (t-SNE) [36]

The distributions of the data extracted by ProxyAnchor, ProxyGML, and our method are visualized by t-SNE [36] in Fig. 5. All classes are isolated well by ProxyAnchor in the training set, but the separation cannot be generalized to the test set, in which some classes mix with others. The distribution of the training set is chaotic in the embedding feature space of ProxyGML. It may be because ProxyGML focuses on the local discrimination of each sample. Samples in a class may not be pulled close together due to the multiple proxies and softmax function. Classes are separately much clearer in the test set, which suggests that ProxyGML is able to learn the discriminant features successfully. Similar to ProxyAnchor, classes are clearly separated in our method but the inner-class variety is

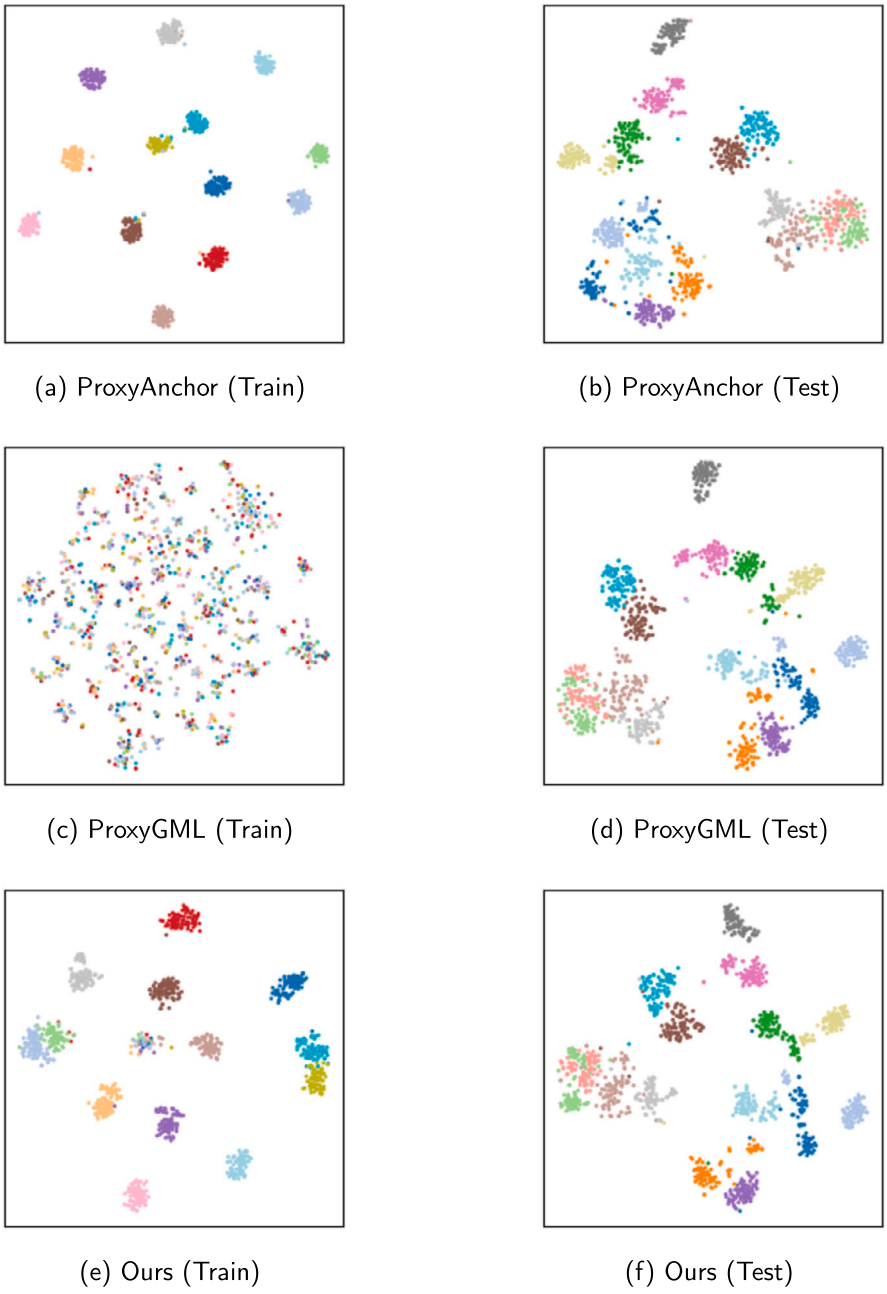

Fig. 5. The T-SNE visualizations of the feature extracted by ProxyAnchor, ProxyGML, and our method on the training and test sets of Cars196.

larger and the intra-class difference is smaller. It confirms the objective of our model and the margin setting. Our method generalizes the class discrimination ability well to the test set. Most classes form their own clusters and have a clear margin with other classes.

4.6. Intuitive comparison

Image retrieval aims to find the samples which have same label with the query image in the database by calculating the distance between them. To intuitively compare our method with the SOTA methods. This section illustrates images retrieved by ProxyAnchor, ProxyGML, and our method. Samples with $I(x)$ close to zero are randomly chosen from the test set.

Fig. 6 illustrates the query image and the top six retrieved images. An image with a red or blue outline indicates the image is wrong or correct respectively. The similarity value output by each model is also displayed on top of each retrieved image. PA and PG stand for ProxyAnchor and ProxyGML. The performance of all methods is reasonable and most retrieved images are similar to the queries. However, ProxyAnchor is confused by the angle and color of the cars and retrieves wrong images more frequently than

Fig. 6. The top six images retrieved by ProxyAnchor (PA), ProxyGML (PG), and our method for three queries as illustration. A similarity value is shown on the top of an image, and the image with a red and blue outline indicates the wrong and correct retrieval respectively. The similarity value is shown on top of an image.

ProxyGML and our method. These results confirm the limitation of using a single proxy. On the other hand, our method retrieves more relevant images than ProxyGML, which suggests that more discriminative and generalizable features are extracted by our model.

4.7. Influence of hyperparameters

4.7.1. Positive and negative margins

The influence of positive and negative margins on the performance of our method is investigated. Fig. 7 shows the results of ProxyAnchor and our method with different positive and negative margins on Cars196 in terms of R@1. The scale of our model is set as 30.

As suggested in [10], the margin between samples and negative proxies should be opposite to the margin between samples and positive proxy in ProxyAnchor. However, our experimental results shown in Fig. 7a provide different opinions. ProxyAnchor performs the best when the values of both margins are similar, *i.e.*, the cells in the diagonal are in dark red. The larger difference between the positive and negative margins may benefit the class discrimination on the training set, but the discrimination ability may not be generalizable to the test set due to over-training. The results confirm the findings in [35]. Our model has a similar performance as

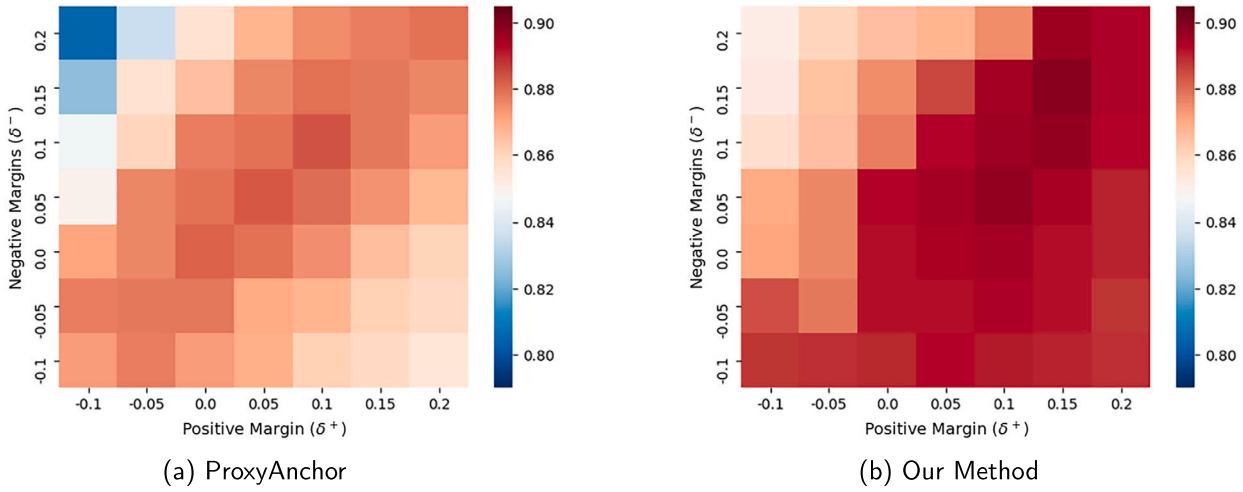

Fig. 7. R@1 of ProxyAnchor and our method with different positive and negative margins on Cars196.

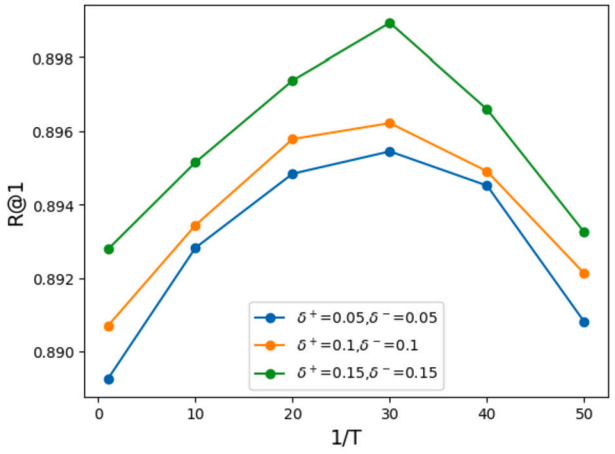

Fig. 8. R@1 of our method with δ^+ and δ^- using different temperature values on Cars196.

ProxyAnchor, and performs the best when the positive margin equals the negative margin. It explains the reason for setting $\delta^+ = \delta^-$ in the loss function.

4.7.2. Temperature of SoftMax function

The temperature (T) of the softmax function shown in (2) indirectly determines the number of associated proxies for a sample. Fig. 8 illustrates R@1 of our methods with different positive and negative margins using different T on Car196. All results basically follow the bell-shape distribution and the peak is roughly at $T = 1/30$, which suggests that too few and too many positive proxies are both not good for the performance. It is because the inner-class variety is controlled by the number of positive proxies associated with a sample. It is expected that a small temperature value is good for a larger inner-class variety, and vice versa. If the inner-class variety is large, the samples of different classes may overlap with others. However, the diversity of samples in a class may not be learned well when the inner-class variety is small. Therefore, R@1 of our model drops when T is too small and large.

4.8. Running time

In each mini-batch, the proxy-sample relations are calculated in order to indicate the association between samples and proxies. The big-O of proxy-sample relations is in $O(n \times p)$, where n and p represent the number of samples and proxies. After data augmentation, the relation between the proxies and the proxies is also processed, the time complexity is $O(p^2)$. So the comprehensive time complexity is $O(n \times p + p^2)$.

The training time on each epoch and convergence time of ProxyAnchor, ProxyGML, and our method are shown in Table 3. It is expected that our method requests a shorter running time than ProxyAnchor due to only a subset of proxies being associated with each sample. The time required by our method is longer than ProxyGML. It is because of positive and negative relation calculation.

Table 3
Training Time Per Epoch and Convergence Time of ProxyAnchor, ProxyGML, and Our Method on Car196.

Methods	ProxyAnchor	ProxyGML	Ours
Epoch time	1 m 12 s	54 s	1 m 6 s
Convergence time	53 m 16 s	48 m 31 s	50 m 14 s

5. Conclusion

Limited information provided by a mini-batch may mislead DML and downgrades the performance of its model. This study aims to provide a global view of the entire dataset by using multi-proxy in order to enhance the generalization ability of DML. Different from the existing multi-proxy methods, a positive proxy only represents a subset of samples in our model since a sample may not share all characteristics of its class. On other hand, negative proxies far away from a sample are also ignored in learning to avoid unnecessary updates and over-training. Since the flexibility of proxies increases the representation ability of our model on the data distribution, a better generalization ability is expected. A few training samples in a mini-batch may only provide local information on the data in training. Our study considers the sample-proxy and proxy-proxy relations representing local and global views of the data distribution respectively in training in order to enhance the efficiency of learning on the intra-class distance. In addition, the settings of positive and negative margins are also discussed. Our model works better when the value of positive and negative is close. The experimental results confirm the effectiveness of our model and also the margin setting. Our method outperforms all SOTA methods in all three benchmark image retrieval datasets. Although multiple proxies are used, our running time is close to other methods since only some subset of proxies are considered and the proxy-proxy relations provide useful information in learning. One possible future work is to extend the model by considering the fine-grained correlation between samples in order to enhance the local information in learning. The trade-off between local and global information should be investigated. Moreover, the computational complexity may be a concern and should be balanced with the generalization ability.

CRediT authorship contribution statement

Patrick P.K. Chan: Conceptualization, Supervision, Writing – original draft, Writing – review & editing. **Shute Li:** Conceptualization, Formal analysis, Investigation, Methodology, Software. **Jingwen Deng:** Data curation, Software, Writing – review & editing. **Daniel S. Yeung:** Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:
Patrick Chan reports was provided by South China University of Technology. Patrick Chan reports a relationship with South China University of Technology that includes: employment.

Data availability

No data was used for the research described in the article.

References

[1] R. Hadsell, S. Chopra, Y. LeCun, Dimensionality reduction by learning an invariant mapping, in: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), vol. 2, IEEE, 2006, pp. 1735–1742.
[2] Z. Abo-Hammour, O. Abu Arqub, S. Momani, N. Shawagfeh, Optimization solution of Troesch's and Bratu's problems of ordinary type using novel continuous genetic algorithm, Discrete Dyn. Nat. Soc. (2014) 2014.
[3] O.A. Arqub, Z. Abo-Hammour, Numerical solution of systems of second-order boundary value problems using continuous genetic algorithm, Inf. Sci. 279 (2014) 396–415.
[4] H.-X. Yu, A. Wu, W.-S. Zheng, Unsupervised person re-identification by deep asymmetric metric embedding, IEEE Trans. Pattern Anal. Mach. Intell. 42 (4) (2020) 956–973, https://doi.org/10.1109/TPAMI.2018.2886878.
[5] P. Li, G. Zhao, X. Xu, Coarse-to-fine few-shot classification with deep metric learning, Inf. Sci. 610 (2022) 592–604, https://doi.org/10.1016/j.ins.2022.08.048.
[6] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: a unified embedding for face recognition and clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 815–823.
[7] K. Sohn, Improved deep metric learning with multi-class n-pair loss objective, in: Advances in Neural Information Processing Systems, 2016, pp. 1857–1865.
[8] Q. Qian, L. Shang, B. Sun, J. Hu, T. Tacoma, H. Li, R. Jin, Softtriple loss: deep metric learning without triplet sampling, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 6449–6457.
[9] Y. Movshovitz-Attias, A. Toshev, T.K. Leung, S. Ioffe, S. Singh, No fuss distance metric learning using proxies, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 360–368.
[10] S. Kim, D. Kim, M. Cho, S. Kwak, Proxy anchor loss for deep metric learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 3238–3247.
[11] Y. Zhu, M. Yang, C. Deng, W. Liu, Fewer is more: a deep graph metric learning perspective using fewer proxies, arXiv preprint arXiv:2010.13636, 2020.
[12] G. Andresini, A. Appice, D. Malerba, Autoencoder-based deep metric learning for network intrusion detection, Inf. Sci. 569 (2021) 706–727, https://doi.org/10.1016/j.ins.2021.05.016.

[13] W. Chen, X. Chen, J. Zhang, K. Huang, Beyond triplet loss: a deep quadruplet network for person re-identification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 403–412.
[14] H. Oh Song, Y. Xiang, S. Jegelka, S. Savarese, Deep metric learning via lifted structured feature embedding, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4004–4012.
[15] M. Bucher, S. Herbin, F. Jurie, Hard negative mining for metric learning based zero-shot classification, in: European Conference on Computer Vision, Springer, 2016, pp. 524–531.
[16] X. Wang, H. Zhang, W. Huang, M.R. Scott, Cross-batch memory for embedding learning, in: CVPR, 2020.
[17] D. Wu, H. Wang, Z. Hu, F. Nie, Improved deep metric learning with local neighborhood component analysis, Inf. Sci. 617 (2022) 165–176, https://doi.org/10.1016/j.ins.2022.10.090.
[18] G. Gu, B. Ko, H.-G. Kim, Proxy synthesis: learning with synthetic classes for deep metric learning, Proc. AAAI Conf. Artif. Intell. 35 (2) (2021) 1460–1468, https://doi.org/10.1609/aaai.v35i2.16236.
[19] K. Roth, O. Vinyals, Z. Akata, Non-isotropy regularization for proxy-based deep metric learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 7420–7430.
[20] E.W. Teh, T. DeVries, G.W. Taylor, Proxynca++: revisiting and revitalizing proxy neighborhood component analysis, in: European Conference on Computer Vision, Springer, 2020, pp. 448–464.
[21] B. Nguyen, F.J. Ferri, C. Morell, B. De Baets, An efficient method for clustered multi-metric learning, Inf. Sci. 471 (2019) 149–163, https://doi.org/10.1016/j.ins.2018.08.055.
[22] I. Elezi, S. Vascon, A. Torcinovich, M. Pelillo, L. Leal-Taixé, The group loss for deep metric learning, in: A. Vedaldi, H. Bischof, T. Brox, J.-M. Frahm (Eds.), Computer Vision – ECCV 2020, Springer International Publishing, Cham, 2020, pp. 277–294.
[23] I. Elezi, J. Seidenschwarz, L. Wagner, S. Vascon, A. Torcinovich, M. Pelillo, L. Leal-Taixé, The group loss++: a deeper look into group loss for deep metric learning, IEEE Trans. Pattern Anal. Mach. Intell. 45 (2) (2023) 2505–2518, https://doi.org/10.1109/TPAMI.2022.3163846.
[24] Z. Yang, M. Bastan, X. Zhu, D. Gray, D. Samaras, Hierarchical proxy-based loss for deep metric learning, in: 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2022, pp. 449–458.
[25] W. Ge, Deep metric learning with hierarchical triplet loss, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 269–285.
[26] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, arXiv e-prints arXiv:1503.02531, 2015.
[27] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, P. Perona, Caltech-UCSD Birds 200, Tech. Rep. CNS-TR-2010-001, California Institute of Technology, 2010.
[28] J. Krause, M. Stark, J. Deng, L. Fei-Fei, 3d object representations for fine-grained categorization, in: 4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13), Sydney, Australia, 2013.
[29] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: International Conference on Machine Learning, PMLR, 2015, pp. 448–456.
[30] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.
[31] H. Oh Song, S. Jegelka, V. Rathod, K. Murphy, Deep metric learning via facility location, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5382–5390.
[32] Y. Yuan, K. Yang, C. Zhang, Hard-aware deeply cascaded embedding, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 814–823.
[33] X. Xu, Y. Yang, C. Deng, F. Zheng, Deep asymmetric metric learning via rich relationship mining, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4076–4085.
[34] W. Zheng, Z. Chen, J. Lu, J. Zhou, Hardness-aware deep metric learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 72–81.
[35] X. Wang, X. Han, W. Huang, D. Dong, M.R. Scott, Multi-similarity loss with general pair weighting for deep metric learning, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 5017–5025.
[36] M. LJPvd, G. Hinton, Visualizing high-dimensional data using t-sne, J. Mach. Learn. Res. 9 (2579–2605) (2008) 9.