Relative Order Analysis and Optimization for Unsupervised Deep Metric Learning

Shichao Kan[1,2], Yigang Cen[1,2,*], Yang Li[3], Vladimir Mladenovic[4] and Zhihai He[3,*]

[1]Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China

[2]Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing 100044, China

[3]Department of Electrical Engineering and Computer Science, University of Missouri, Columbia, MO 65211, USA

[4]Faculty of Technical Sciences University of Kragujevac, Cacak, Serbia

16112062@bjtu.edu.cn; **ygcen@bjtu.edu.cn**; yltb5@mail.missouri.edu

vladimir.mladenovic@ftn.kg.ac.rs; **HeZhi@missouri.edu**

Abstract

In unsupervised learning of image features without labels, especially on datasets with fine-grained object classes, it is often very difficult to tell if a given image belongs to one specific object class or another, even for human eyes. However, we can reliably tell if image C is more similar to image A than image B. In this work, we propose to explore how this relative order can be used to learn discriminative features with an unsupervised metric learning method. Instead of resorting to clustering or self-supervision to create pseudo labels for an absolute decision, which often suffers from high label error rates, we construct reliable relative orders for groups of image samples and learn a deep neural network to predict these relative orders. During training, this relative order prediction network and the feature embedding network are tightly coupled, providing mutual constraints to each other to improve overall metric learning performance in a cooperative manner. During testing, the predicted relative orders are used as constraints to optimize the generated features and refine their feature distance-based image retrieval results using a constrained optimization procedure. Our experimental results demonstrate that the proposed relative orders for unsupervised learning (ROUL) method is able to significantly improve the performance of unsupervised deep metric learning.

1. Introduction

Learning discriminative features to represent images is an important task in computer vision and machine learning. Images with the same semantic labels should have similar features being aggregated into compact clusters in the high-dimensional feature space. Meanwhile, images from different classes should be well separated from each other. Recently, methods based on deep neural networks have made remarkable progress in learning discriminative features for images [9]. In this work, we consider the unsupervised deep metric learning where the image class labels are not available. Furthermore, the test image classes are totally different from the training classes.

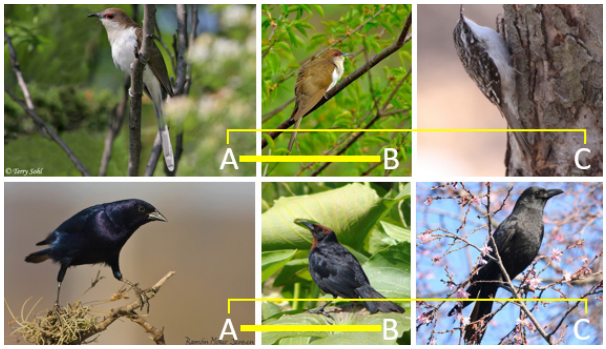

Figure 1. Illustration of the proposed idea of relative orders. If we only look at images A and B, it is hard to determine if they are from the same classes or not. But, with image C as reference, we can reliably tell that images (A, B) are definitely more similar than images (A, C).

Existing methods for unsupervised feature or representation learning attempt to construct pseudo labels using clustering methods or self-supervision labels based on self augmentation or pretext tasks [1, 4, 29]. It has been observed that the pseudo labels suffer from high error rates, especially for datasets with fine-grained object classes. It has also been recognized that features learned from self-supervision tasks cannot generalize well to other tasks or new classes [33]. The key challenge is how to effectively handle the large intra-class variation and inter-class ambiguity. Without labels, how do we tell whether two images

* corresponding authors

are from the same class or not. For example, Figure 1 shows two pairs of images (A, B) from the fine-grain CUB dataset. It is really hard to tell if images A and B are from the same class (bird species) or not, since we do not have any reference. Given a set of images without labels, when determining if two images are from the same class or not, our human visual system often performs a series of relative comparisons between images (A, B) and other images. For example, when a third image C is provided, as illustrated in Figure 1, we can definitely tell that image B is closer to A than image C to A. We denote this relative order by $A : B < C$. Here, image A is referred to as the *anchor image*. We can further extend this relative ordering to a set of images $\{F_n | 1 \leq n \leq N\}$ with anchor image A. If image F_n is visually closer or more similar to the anchor image A than F_m, we denote this as $A : F_n < F_m$.

It should be noted that this relative order analysis is different from the triplet loss or other contrastive loss developed in the metric learning literature [22, 27] since they require using the image labels or pseudo labels to construct positive and negative image pairs for contrastive analysis. However, our proposed relative order analysis does not need absolute image labels. Instead, it only needs reference-based relative comparison and ordering. In unsupervised learning, it might be very challenging and ambiguous to tell if two images are from the same classes or not. For example, many existing methods use k-mean clustering to generate pseudo labels [27], which are then used to identify positive and negative pairs for contrastive metric learning. In our experiments, we have observed that the average accuracy for positive and negative pairs during training is about 40-60%, which is quite low. However, it is often much less challenging, even with high confidence, to tell if an image is closer to one specific image than others. This motivates us to explore this relative order analysis for more effective unsupervised feature learning.

In this work, we propose to explore how these relative orders can be used to learn discriminative features with an unsupervised metric learning method. We analyze the uncertainty involved in relative order analysis and introduce a ternary relative ordering function. We construct reliable relative orders for groups of image samples and learn a deep neural network to predict these relative orders. During training, this relative order prediction network and the feature embedding network are tightly coupled, providing mutual constraints to each other to improve overall metric learning performance. During testing, the predicted relative orders are used as constraints to optimize the generated features and refine their feature distance-based image retrieval results using a constrained optimization procedure. Our experimental results demonstrate that the proposed relative orders for unsupervised learning (ROUL) method outperforms existing methods by a large margin.

2. Related Work and Major Contributions

This work is related to deep metric learning, self-supervised representation learning, and unsupervised metric learning. Most recent deep metric learning methods are focusing on design effective contrastive loss for learning the feature embedding network [23, 22]. For example, triplet loss [22] defines a positive pair and a negative pair based on the same anchor point. It encourages the embedding distance of positive pair to be smaller than the distance of negative pair by a given margin. Movshovitz-Attias *et al.* [18] proposed to optimize the triplet loss on a different space of triplets, called ProxyNCA, which consists of an anchor data point and similar and dissimilar proxy points that are learned as well. From a different point of view, Zhai *et al.* [35] proposed a NormSoftmax loss for deep metric learning. Recently, multi-similarity loss [27] considers multiple similarities and provides a more powerful approach for mining and weighting informative pairs by considering multiple similarities.

Self-supervised representation learning directly derives information from unlabeled data itself by formulating predictive tasks to learn informative feature representations. Gidaris *et al.* [7] proposed to predict the image rotation angle. Zhang *et al.* [36] proposed to predict the randomly sampled transformation from the encoded features using the Auto-encoding transformation (AET). Bachman et al. [1] proposed a method of self-supervised representation learning named augmented multiscale deep infoMax (AMDIM) based on maximizing mutual information between features extracted from multiple views of a shared context. Misra *et al.* [17] developed a pretext-invariant representation learning (PIRL) that learns invariant representations based on pretext tasks for self-supervised representation learning. Wang *et al.* [26] proposed a transformation generative adversarial networks (TrGAN) for unsupervised image synthesis and representation learning. O.Pinheiro *et al.* [21] proposed view-agnostic dense representation (VADeR) for unsupervised learning of dense representations, which learns pixel wise representations by forcing local features to remain constant over different viewing conditions.

Unsupervised metric learning is a relatively new research topic. It is a more challenging task since the training classes have no labels and it does not overlap with the testing classes. DeepCluster [2] uses k-means clustering to assign pseudo-labels to the features generated by the deep neural network and introduces a discriminative loss to train the network. [13] proposed an unsupervised method to mine hard positive and negative samples based on manifold-aware sampling. The feature embedding can be trained with standard contrastive and triplet loss. Based on deep metric learning theory, He *et al.* [8] proposed a momentum contrast (MoCo) method for visual representation learning in an unsupervised manner. Chen *et al.* [3] proposed a con-

trastive learning framework (simCLR) for unsupervised visual representation learning. The Instance method [33, 32] considered each image as an instance. It optimizes the instance feature embedding directly based on the positive augmentation invariant and negative separated properties. Nguyen *et al.* [19] proposed to use a deep clustering loss to learn centroids and generate robust pseudo-labels for better deep metric learning. Dutta *et al.* [5, 6] proposed to obtain pseudo-labels of data using a graph-based clustering approach for unsupervised deep metric learning. Ye *et al.* [31] proposed a probabilistic structural latent representation (PSLR) method, which incorporates an adaptable softmax embedding to approximate the positive concentrated and negative instance separated properties in the graph latent space.

This work is also related to the relative attribute analysis [20, 34, 16]. Relative attribute analysis was first introduced in [20], aiming to achieve zero-shot learning based on relative relationships and generate image descriptions by learning relative visual attributes. Yu *et al.* [34] proposed an active image generation approach by jointly learning attribute ranking and novel image sample generation. Min *et al.* [16] proposed a multitask deep relative attribute learning network (MTDRALN) to learn all the relative attributes simultaneously via multi-task Siamese networks. It should be noted that these methods construct the relative attributes from manual labels for supervised learning. Our work is quite unique and different in that (1) we mine high-confidence relative orders from unlabeled samples, and (2) we couple the relative order analysis with contrastive metric learning based on mutual constraints for effective unsupervised deep metric learning.

3. Method

3.1. Relative Order Analysis with High Confidence

The task of unsupervised feature or metric learning is to learn a feature embedding network from a set of unlabeled training images which can generate discriminative features to represent images from unseen classes [13, 31]. As discussed in the above section, in this paper, we hypothesize that relative orders are more reliable and efficient than pseudo labels, motivated by the fact that human eyes are much better on relative comparisons than absolution assignment of categorical labels due to the large intra-class variation and inter-class ambiguity. Given an anchor image A, for example, the query image in our image retrieval experiments, and a set of comparison images $\{F_n\}$, if image F_n is closer to A than F_m, we denote this as $A : F_n < F_m$. We recognize that for some images, it is hard to determine which one is closer to the anchor image. To address this issue, we propose to quantize the relative orders so as to decrease the amount of uncertainty being introduced into the supervised learning processing. Specifically, we introduce a high confidence relative order operation $A : F_n \prec F_m$ if image F_n is closer to image A than image F_m with high confidence. With this, we can define the following ternary relative order function:

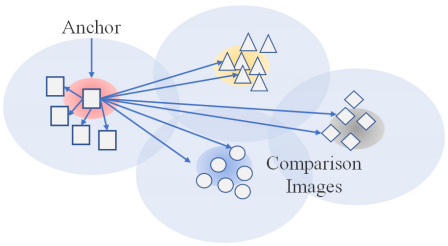

Figure 2. Construction of the relative orders between images with high confidence.

$$\mathcal{O}_A(n,m) = \begin{cases} +1, & \text{if } A : F_n \prec F_m, \\ -1, & \text{if } A : F_m \prec F_n, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

When $\mathcal{O}_A(n,m) = 0$, which indicates that the relative order analysis is uncertain, or it is not clear which image is closer to the anchor image A, we will assign a very low weight for the corresponding data during network training and network inference optimization. In the later part of this section, we will explain how to construct these high confidence relative orders. The high-confidence relative orders for N comparison images $\{F_n\}$ form an $N \times N$ matrix $[\mathcal{O}_A(n,m)]_{N\times N}$. Given an anchor image and a set of comparison images $\{F_n | 1 \le n \le N\}$, we will learn a network Φ_θ to predict their relation order matrix $[\mathcal{O}_A(n,m)]_{N\times N}$, which will be further explained in the following section.

The next question is how to select the anchor image and the set of comparison images and how to determine their relation orders with high confidence so that we can construct a training set to learn the relative order network Φ_θ. As shown in Figure 2, we start with the k-mean clustering of the training samples using the image features extracted by the feature embedding network $\boldsymbol{F}_\gamma$ that has been learned so far. Let $\{\boldsymbol{C}_m\}$ be the cluster centers and Ω_m be set of images close to the cluster center

$$\Omega_m = \{X \mid d(X, C_m) \le \Delta\}, \quad (2)$$

where Δ is a threshold. For example, we can set Δ to be 1/3 of the minimum distance between all cluster centers. To address the uncertainty issue in pseudo labels provided by the k-mean clustering, we propose to explore the following observations to construct high-confidence anchor-comparison images. (1) An image is closer to its self augmentations than those images near the centers of other clusters. (2) Images from the cluster centers are closer to each other than images from other cluster centers. Specifically, for an arbitrary image A, let A_i be its self augmentations generated

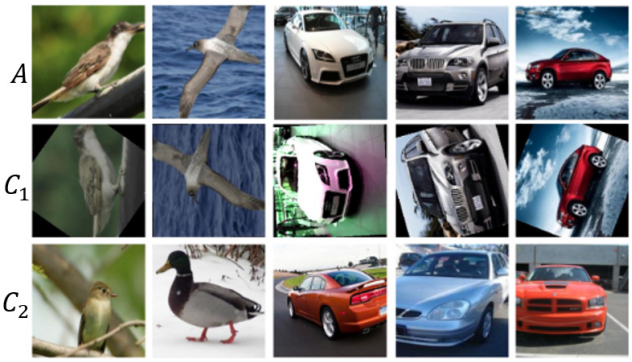

Figure 3. Constructed training examples for learning relative orders. In each column, the first image is the anchor image A, the second and third are the comparison images C_1 and C_2 satisfying $A : C_1 \prec C_2$, or C_1 is closer to A than C_2.

with small random rotations, cropping, and resizing. Then, we have

$$\mathcal{O}_A(A_i, C) = 1, \text{for } A \in \Omega_m, C \in \Omega_k, m \neq k, \quad (3)$$

with high confidence according to observation (1), and

$$\mathcal{O}_A(B, C) = 1, \text{for } A, B \in \Omega_m,\ C \in \Omega_k, m \neq k, \quad (4)$$

with high confidence according to observation (2). Certainly, we have $\mathcal{O}_A(C, A_i) = -1$ and $\mathcal{O}_A(C, B) = -1$ for the above two cases. This also implies that the relative order matrix is anti-symmetric. To construct training samples for the relative order score 0, we can explore the following observations: (1) Given two augmentations A_i and A_j of image A, it is hard to tell which one is closer to A, i.e., $\mathcal{O}_A(A_i, A_j) = 0$. (2) Given two images from the same cluster center region Ω_m, it is hard to tell which one is closer to the anchor image in the same cluster. Certainly, there are also other ways to construct training samples for the relative order analysis network. Figure 3 shows 5 examples from our training set constructed using the above procedure to train our relative order prediction network. In each column, the first one is the anchor image A. The second and third are the comparison images C_1 and C_2 satisfying $A : C_1 \prec C_2$, or C_1 is closer to A than C_2. We can see that there is a large intra-class variation between C_1 and A and also a large inter-class ambiguity between C_2 and A. These high confidence yet challenging relative order image sets provide important training sampling for our relative order prediction network to learn more discriminative features.

3.2. Relative Order and Metric Order Consistency

As illustrated in Figure 4, two networks, the feature embedding network $\boldsymbol{F}_\gamma$ which encodes the input image A into a feature vector $\boldsymbol{F}_\gamma(A)$ and the relative order analysis network which estimates the relative order matrix

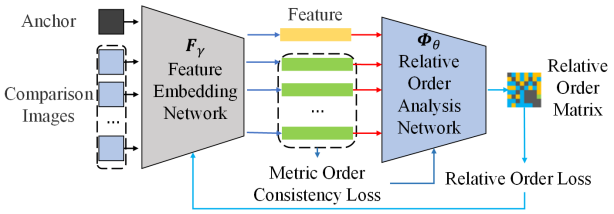

Figure 4. Illustration of the proposed method of relative order analysis for unsupervised feature learning: cooperative learning of the relative order.

$[\mathcal{O}_A(C_n, C_m)]_{N\times N}$ for an anchor image A and N comparison images C_n, $1 \leq n \leq N$. Here, our assumption is that, when these two networks are successfully trained, the generated features should satisfies the relative order constraint. On the other hand, the predicted relative order matrix should satisfy the feature distance constraint. Based on this assumption, we introduce the following two consistency loss functions: *relative order consistency* and *metric order consistency*.

(1) Relative order consistency. Here, we assume that the relative order analysis network Φ_θ is successfully trained or successfully refined. We use the relative order matrix generated by Φ_θ to guide the training or update of the feature embedding network $\boldsymbol{F}_\gamma$ with the relative order consistency loss $\mathcal{L}_{ROC}$ to be defined in the following. Specifically, in the current mini-batch during the training process, we select one image as the anchor image A and the rest as the set of comparison images. To calibrate the relative order analysis process, we also incorporate a small set of self augmentations of A into the comparison image set, denoted by $\Omega_C = \{C_1, C_2, \cdots, C_N\}$. Let $[\mathcal{O}_A(C_n, C_m)]$ be the relative order matrix generated by network Φ_θ, which will serve as a constraint for training of the feature embedding network $\boldsymbol{F}_\gamma$. The generated feature for the anchor and comparison images are $\boldsymbol{F}_\gamma(A)$ and $\{\boldsymbol{F}_\gamma(C_n)\}$. We use the L_2-norm to measure the distance $d[\cdot, \cdot]$ between two features. When constructing the relative order consistency loss, our major observation is that, if image C_n is closer to A than C_m with $\mathcal{O}_A(C_n, C_m) > 0$, then we should have

$$\Delta(n, m) = d[\boldsymbol{F}_\gamma(A), \boldsymbol{F}_\gamma(C_n)] - d[\boldsymbol{F}_\gamma(A), \boldsymbol{F}_\gamma(C_m)] < 0.$$

If $\Delta(n, m)$ becomes positive, then a larger penalty should be imposed. Note that the relative order matrix is anti-symmetric, i.e., $\mathcal{O}_A(C_n, C_m) = -\mathcal{O}_A(C_m, C_n)$. We only need to define the loss on these positive entries of the matrix. Specifically,

$$\mathcal{L}_{ROC} = \sum_A \sum_{n,m=1}^{N} \sigma[\mathcal{O}_A(C_n, C_m)] \cdot e^{\alpha \cdot \Delta(n,m)}. \quad (5)$$

Here, function $\sigma[x] = x$ for $x > 0$, and $\sigma[x] = 0$ for $x \leq 0$. $\alpha > 0$ is a model parameter with a default value of 0.1.

(2) Metric order consistency. Similarly, when the feature embedding network is successfully trained or updated, we assume that the learned image features are correct and use them to guide the training of the relative order network. Specifically, if the feature distance between image C_n and the anchor image A is smaller than the feature distance between C_m and A, or $\Delta(n,m) < 0$, then we should expect $\mathcal{O}_A(C_n, C_m) > 0$. If $\mathcal{O}_A(C_n, C_m)]$ is decreasing to 0 or even negative, then a large penalty should be imposed. Again, we only need to compute this penalty for cases with negative $\Delta(n,m)$ due to the anti-symmetric property of $\Delta(n,m)$. Based on this observation, we define the metric order consistency as

$$\mathcal{L}_{MOC} = \sum_{A} \sum_{n,m=1}^{N} [1 - \mathcal{O}_A(C_n, C_m)] \cdot \log_\alpha[1 - \Delta(n,m)] \times \sigma[-\Delta(n,m)]. \tag{6}$$

Here, $\log_\alpha[1 - \Delta(n,m)]$ is used as a weight. Note that, in this summation, the term $\sigma[-\Delta(n,m)]$ ensures that we only consider those cases with $\Delta(n,m) < 0$. Cases with large distances should have large weights in the penalty function. We choose the function $\log_\alpha$ to scale down the excessive weights by very large distance values. For these cases with negative $\Delta(n,m)$, we expect that their relative order value $\mathcal{O}_A(C_n, C_m)$ approaches 1, or $1 - \mathcal{O}_A(C_n, C_m)$ approaches 0.

3.3. Network Design and Cooperative Training

For the feature embedding network, we incorporate the marginal variance constraint [10] into the multi-similarity (MS) loss [27] to achieve effective unsupervised metric learning. The original MS method computes the similarity scores between image samples in the current mini-batch. In this work, we extend this similarity analysis to the whole training set using the approach of memory bank [8]. The similarity matrix between features of the current mini-batch and all features in the memory bank can be computed, $\mathbf{S} = \{s_{ik}\}$, $1 \le i \le m, 1 \le k \le n \times m$. Here, s_{ik} is the cosine similarity between two features. Using $\mathbf{S}$, we determine the set of positive pairs $\mathcal{P}$ and the set of hard negative pairs $\mathcal{N}$ based on their similarity scores. We define the loss for each sample I_i in the mini-batch as follows

$$\mathcal{L}^i_{FEN} = \frac{1}{\lambda_P} \log[1 + \sum_{(i,k)\in\mathcal{P}} (e^{-\lambda_P(s_{ik} - \delta)})] + \frac{1}{\lambda_N} \log[1 + \sum_{(i,k)\in\mathcal{N}} (e^{\lambda_N(s_{ik} - \delta)})], \tag{7}$$

where δ is a margin threshold and it is set as 0.5 in our experiments. According to [28], λ_P is set to 2 for positive pairs and λ_N is set to 40 for hard negative pairs. Then, combining with the relative order loss in (5), the overall loss of the current mini-batch is given by

$$\mathcal{L}_{FEN} = \mathcal{L}_{ROC} + \frac{1}{m} \sum_{i=1}^{m} \mathcal{L}^i_{FEN}. \tag{8}$$

For the relative order prediction network (ROP) , we use the feature embedding network to generate feature maps for the anchor image A and the comparison images $\{C_n\}$. These feature maps are then cascaded together and further analyzed by a network to predict the relative order matrix. The error between the predicted relative order matrix $\hat{\mathcal{O}}_A(C_n, C_m)$ and the ground truth $\mathcal{O}_A(C_n, C_m)$ is defined as the relative order matrix prediction loss

$$\mathcal{L}_{\mathcal{O}} = \frac{1}{|A|nm} \sum_{A} \sum_{n,m} [\hat{\mathcal{O}}_A(C_n, C_m) - \mathcal{O}_A(C_n, C_m)]^2. \tag{9}$$

Using the procedure described in Section 3.1, we construct the training set of samples and their corresponding relative order matrices and use them to train the relative order prediction network. Combining the metric order loss from (6) and the relative order matrix prediction loss from (9), we have the overall loss function for the relative order prediction network $\mathcal{L}_{ROP} = \mathcal{L}_{MOC} + \mathcal{L}_{\mathcal{O}}$. The feature embedding network and the relative order prediction network are then trained in an iterative and cooperative manner.

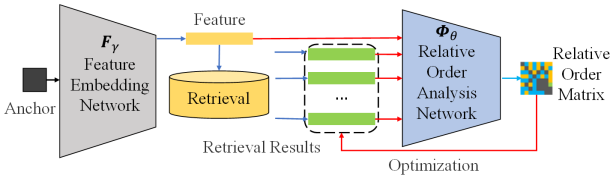

Figure 5. Optimizing the metric learning performance and image retrieval results using relative orders.

3.4. Constrained Optimization of Network Inference

As discussed in the Section 1, the task of unsupervised metric learning is to learn a network which can generate discriminative features to represent the input images. One direct and important test to evaluate the discriminative power of this feature representation is image retrieval, ranking the images based on their feature distance from the query image. The goal is to rank images from the same class as the query image on the top of the retrieval results. In this section, we will explore how the learned relative order analysis network can be used to optimize the performance of metric learning and image retrieval during the testing phase.

As illustrated in Figure 5, once the feature embedding network is learned, we use it to extract the feature of the

input query image. Based on feature distance, for example the L_2-norm, we retrieve the images from the test dataset and rank the query results based on their feature distance in an ascending order. Let $\{Q_m|1 \leq m \leq M\}$ be top M query results. The retrieval performance, measured by the Recall@K rate or the accuracy of the top K retrieval results, depends on the ordering of these query results. For example, if the image with a different label than the query image is incorrectly ranked before the another image which has the correct label, it will decrease the retrieval performance score. To address this, we propose to use the relative order analysis, which has been successfully learned in the above section, to analyze the relative ordering of these query results and optimize the image retrieval performance.

To successfully examine if the query results $\{Q_m|1 \leq m \leq M\}$ are belonging to the same class as the query image using relative order analysis, we propose to extend the query results by adding self-augmentations $\{A_i|1 \leq i \leq L\}$ of the query image A into the results, and denote this extended set by $\Omega_C = \{C_n|1 \leq n \leq N\}$, $N = M + L$. The major motivations include: (1) We can use these self-augmentations to calibrate the relative order analysis results. (2) If an image is truly from the same class as the query image, most likely, it will be close to one of its augmentations. Let $\mathcal{O}_A(C_n, C_m)$ be the relative order matrix produced by the relative order analysis network. Let $d(C_n) = d[\Phi_\theta(A), \Phi_\theta(C_n)]$ be the feature distance between image C_n and the query image. Let $r(C_n)$ be the rank of image C_n in the query results. If C_n is the self augmentation of A, its rank $r(C_n) = 0$. Initially, we set the value of $r(C_n)$ according to the rank order in the retrieval result and all query results are organized according to their feature distance to the query image A. But, this ranking could be wrong and it may not satisfy the relative order constraint. Specifically, for two images C_n and C_m with $d(C_n) < d(C_m)$, we could have $\mathcal{O}_A(C_n, C_m) < 0$. Otherwise, it leads to inconsistency between the relative order and the distance order.

Our main idea in optimizing the query results using relative orders is that, when adjusting the ranking of the queries results, the amount of changes in the distance order is small, but the amount of improvement in the relative orders is significant. In this case, we will proceed to adjust the rank of the query results. To this end, we define the follow energy function for the ranking $\{r(C_n)\}$:

$$\mathcal{E}[r(C_1), \cdots, r(C_N)] = \sum_{n,m=1}^{N} \mu[r(C_m) - r(C_n)] \times \left\{ e^{\alpha_1 \cdot [d(C_n) - d(C_m)]} + \lambda \cdot [1 - \mathcal{O}_A(C_n, C_m)] \right\}. \tag{10}$$

Here, $\mu[x] = 1$ for $x > 0$ and $\mu[x] = 0$ for $x \leq 0$. The optimized ranking of the query results $\{r^*(C_n)\}$ aims to minimize this energy function

$$\{r^*(C_n)\} = \arg\min \mathcal{E}[r(C_1), \cdots, r(C_N)]. \tag{11}$$

Figure 6 shows two retrieval examples with and without the network inference optimization based on the relative order constraint. The first column shows the two query images and the rest columns show the query results. We can see that, using the relative order constraint optimization, the number of incorrect retrieval results (highlighted with yellow) have been significantly reduced or being pushed towards lower ranks.

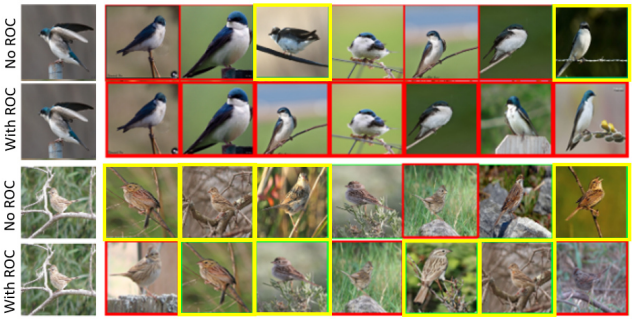

Figure 6. Inference comparison with and without relative order analysis. Images with red boxes are correct ones with the same class label as the query image. Those with yellow boxes are incorrect results from other classes.

4. Experiments

In this section, following the same procedure used by existing papers on unsupervised metric learning [32, 33], we evaluate the proposed method in image retrieval settings. As we know, image retrieval is one of the best applications to evaluate the discriminative power of learned features.

4.1. Datasets and Evaluation Protocol

We use the same benchmark datasets as in existing papers for direct performance comparison, i.e., CUB-200-2011 [25], Cars-196 [15] and the Stanford Online Product (SOP) [15] datasets. (1) The **CUB-200-2011** [25] consists of 11,788 images from 200 bird categories. We use the first 100 classes (5,864 images) for training and the remaining 100 classes (5,924 images) for testing. (2) The **Cars-196** [15] dataset contains 16,185 images of 196 cars classes. We use the first 98 classes (8,054 images) for training and the remaining 98 classes (8,131 images) for testing. (3) The **Stanford Online Product (SOP)** [23] dataset consists of 120,053 images with 22,634 classes crawled from Ebay. Following the publicly partition rule, we split the first 11,318 classes with 59,551 images for training, and the remaining 11,316 classes with 60,502 images for retrieval. In the test set, each image is also used as the query image.

For clustering, the total number of clusters is set as 100 for the CUB and Cars datasets, and 10000 for the SOP

Table 1. Comparison of retrieval performance on the CUB, Cars and SOP datasets with 128-dimensional embeddings on GoogleNet backbone network.

Methods	CUB				Cars				SOP		
	R@1	R@2	R@4	R@8	R@1	R@2	R@4	R@8	R@1	R@10	R@100
Examplar [TPAMI16] [4]	38.2	50.3	62.8	75.0	36.5	48.1	59.2	71.0	45.0	60.3	75.2
NCE [CVPR18] [30]	39.2	51.4	63.7	75.8	37.5	48.7	59.8	71.5	46.6	62.3	76.8
DeepCluster [ECCV18] [2]	42.9	54.1	65.6	76.2	32.6	43.8	57.0	69.5	34.6	52.6	66.8
MOM [CVPR18] [13]	45.3	57.8	68.6	78.4	35.5	48.2	60.6	72.4	43.3	57.2	73.2
AND [ICML19] [11]	47.3	59.4	71.0	80.0	38.4	49.6	60.2	72.9	47.4	62.6	77.1
ISIF [CVPR19] [33]	46.2	59.0	70.1	80.2	41.3	52.3	63.6	74.9	48.9	64.0	78.0
sSUML [AAAI20] [6]	43.5	56.2	68.3	79.1	42.0	54.3	66.0	77.2	47.8	63.6	78.3
aISIF [TPAMI20] [32]	47.7	59.9	71.2	81.4	41.2	52.6	63.8	75.1	49.7	65.4	79.5
CBSwR [BMVC20] [19]	47.5	59.6	70.6	80.5	42.6	54.4	65.4	76.0	-	-	-
Ortho [TAI20] [5]	47.1	59.7	72.1	82.8	45.0	56.2	66.7	76.6	45.5	61.6	77.1
PSLR [CVPR20] [31]	48.1	60.1	71.8	81.6	43.7	54.8	66.1	76.2	51.1	66.5	79.8
Ours: ROUL	**56.7**	**68.4**	**78.3**	**86.3**	**45.0**	**56.9**	**68.4**	**78.6**	**53.4**	**68.8**	**81.7**
Gain: ROUL	**+8.6**	**+8.3**	**+6.2**	**+3.5**	**+0.0**	**+0.7**	**+1.7**	**+2.0**	**+2.3**	**+2.3**	**+1.9**

dataset. We follow the standard evaluation protocol [32] and use the Recall@K [14] to evaluate the performance of our algorithm. For all datasets, our method is evaluated with the original images only, without using the object bounding box information. To compare with the state-of-the-art methods, we use GoogLeNet [24], ResNet-18 [12], and ResNet-50 [9] with an 1-layer embedding head to embed the representation to the 128-dimensional feature space.

4.2. Comparison with State-of-the-Art Methods

We compare our method with the following state-of-the-art methods recently developed in the literature: Examplar [4], noise-contrastive estimation (NCE) [30], DeepCluster [2], mining on manifolds (MOM) [13], anchor neighbourhood discovery (AND) [11], invariant and spreading instance feature (ISIF) [33], stochastic synthetic unsupervised pseudo metric learning (sSUML) [6], augmentation invariant and spreading instance feature (aISIF) [32], center-based softmax with reconstruction (CBSwR) [19], orthogonality (Ortho) [5], and probabilistic structural latent representation (PSLR) [31]. A brief review of these algorithms are provided in our Related Work. We consider two scenarios for performance comparison: (1) learning with an ImageNet pre-trained model, and (2) learning from scratch.

(1) Learning from the ImageNet pre-trained model. In this scenario, we use the network model pre-trained on the ImageNet as the initial backbone encoder and then fine-tune it on the training dataset without using the labels. The results with the GoogLeNet backbone for 128 dimensional embeddings on the CUB, Cars and SOP datasets are summarized in Table 1.

From Table 1, we can see that our ROUL method outperforms the state-of-the-art methods by large margins on the CUB data set. Specifically, our ROUL method has improved the Recall@1, Recall@2, Recall@4 and Recall@8 rates by 8.6%, 8.3%, 6.2%, 3.5%, respectively, on the CUB dataset; and 0.0%, 0.7%, 1.7%, 2.0% on the Cars dataset. On the SOP dataset, our method has improved the Recall@1, Recall@10, and Recall@ 100 rates by 2.3%, 2.3%, and 1.9%, respectively.

Table 2. Comparisons of retrieval performance on the SOP dataset with 128-dimensional embeddings on the ResNet-18 backbone network without pre-trained parameters.

Methods	R@1	R@10	R@100
Examplar [4]	31.5	46.7	64.2
NCE [30]	34.4	49.0	65.2
MOM [13]	16.3	27.6	44.5
AND [11]	36.4	52.8	67.2
ISIF [33]	39.7	54.9	71.0
aISIF [32]	40.7	55.9	72.2
PSLR [31]	42.3	57.7	72.5
Ours: ROUL	**45.4**	**60.5**	**74.8**
Gain: ROUL	**+3.1**	**+2.8**	**+2.6**

(2) Learning from scratch. Following the aISIF [32] and the PSLR [31] methods, we also test the performance using a randomly initialized ResNet-18 network without pre-training, on the large-scale SOP dataset, as shown in Table 2. Results demonstrate that the proposed method achieves much better performance than other methods, 3.1%, 2.8%, and 2.6% gain over the PSLR method for Recall@1, Recall@10, and Recall@100.

(3) Learning with different backbone networks. Following the aISIF [32] and the PSLR [31] method, we also conduct experiments with the ResNet-18 and ResNet-50 backbone encoders for our ROUL method. The embedding size is set to be 128. Results of top-1 recall rates on the CUB, Cars, and SOP datasets are shown in Table 3. Our proposed ROUL method benefits from stronger backbone encoders and outperforms the existing method. It should

Table 3. Top-1 recall rates (%) with 128-dimensional embeddings on different backbone networks.

Backbone	Methods	CUB	Cars	SOP
GoogLeNet	aISIF [32]	47.7	41.2	49.7
	PSLR [31]	48.1	43.7	51.1
	Ours: ROUL	**56.7**	**45.0**	**53.4**
ResNet-18	aISIF [32]	45.5	34.9	**54.7**
	PSLR [31]	48.9	39.2	52.2
	Ours: ROUL	**53.7**	**43.1**	52.4
ResNet-50	aISIF [32]	47.3	41.4	55.6
	PSLR [31]	49.0	42.8	**61.6**
	Ours: ROUL	**55.7**	**49.3**	58.5

Table 4. The contributions of the mutual constraints to the ROUL training with the GoogleNet backbone on the CUB dataset.

Methods	R@1	R@2	R@4	R@8
Baseline	55.4	67.0	76.7	84.9
+ Relative Order Constraint	56.1	67.7	78.1	85.9
+ Metric Order Constraint	56.7	68.4	78.3	86.3

be noted that we could only provide comparison with the aISIF [32] and the PSLR [31] papers since other papers did not report results on other backbone networks.

4.3. Ablation Studies

(1) Contributions of mutual constraints to the ROUL training. Table 4 summarizes the contributions of major components of our algorithm, namely, the relative order constraint (ROC) and the metric order constraint (MOC) used in our ROUL training based on the GoogleNet backbone. From Table 4, we can see that both constraints have significant contributions to the overall performance.

(2) Impact of the ROC loss weight. We determine the weight of the ROC loss on the CUB dataset based on GoogLeNet backbone. Results are shown in Figure 7. We can see that the best choice for the ROC loss weight is 0.1. In our experiments, we set the weight of the ROC loss as 0.1 for all the experiments.

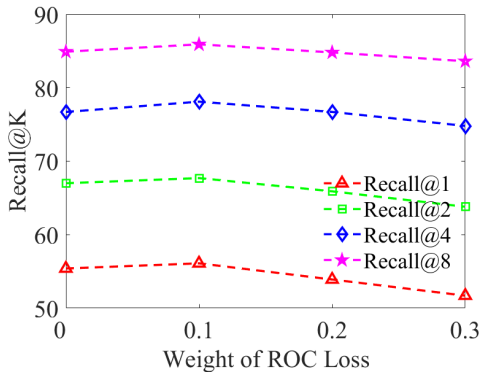

Figure 7. The impact of the weight of ROC loss with the GoogleNet backbone.

(3) Retrieval examples. Figure 8 shows some retrieval examples on the CUB dataset based on the GoogleNet backbone and trained with our ROUL method. The proposed ROUL method is able to accurately find the top matches with very few incorrect results (highlighted in green).

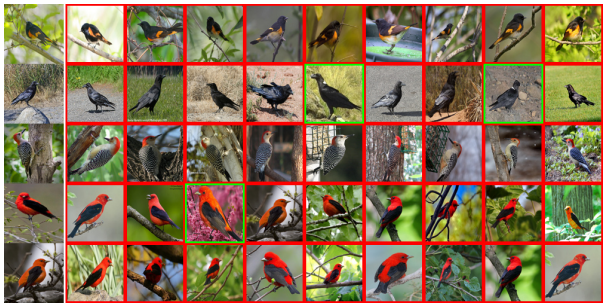

Figure 8. Retrieval examples on the CUB dataset with the proposed method. Retrieved images with red boxes are correct ones with the same class label as the query image. Those with green boxes are incorrect results from other classes.

More ablation studies and experimental results are provided in the **Supplemental Materials**.

5. Conclusion

In this work, we have successfully developed a new unsupervised deep metric learning method based on relative order analysis and optimization. Instead of resorting to clustering or self-supervision to create error-prone pseudo labels for absolute decision, we construct reliable relative orders for groups of image samples and learn a deep neural network to predict these relative orders. This relative order prediction network and the feature embedding network are tightly coupled, providing mutual constraints, namely the relative order constraint and the metric order constraint, to each other to regulate the training process and improve the learning performance in a cooperative manner. During testing, the predicted relative orders are used as constraints to optimize the generated features and refine their feature distance-based image retrieval results using a constrained optimization procedure. Our experimental results have demonstrated that the proposed relative orders for unsupervised learning method significantly improves the performance of unsupervised deep metric learning.

Acknowledgments: This work was supported in part by the National Key R&D Program of China under Grant 2019YFB2204200; in part by the National Natural Science Foundation of China under Grant 61872034, Grant 62011530042, and Grant 62062021; in part by the Beijing Municipal Natural Science Foundation under Grant 4202055; in part by the Natural Science Foundation of Guizhou Province under Grant [2019]1064.

References

[1] Philip Bachman, R. Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 15509–15519, 2019. 1, 2

[2] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIV*, volume 11218 of *Lecture Notes in Computer Science*, pages 139–156, 2018. 2, 7

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *arXiv*, abs/2002.05709, 2020. 2

[4] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin A. Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(9):1734–1747, 2016. 1, 7

[5] Ujjal Kr Dutta, Mehrtash Harandi, and Chellu Chandra Sekhar. Unsupervised deep metric learning via orthogonality based probabilistic loss. *IEEE Transactions on Artificial Intelligence*, 2020. 3, 7

[6] Ujjal Kr Dutta, Mehrtash Harandi, and C. Chandra Sekhar. Unsupervised metric learning with synthetic examples. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 3834–3841. AAAI Press, 2020. 3, 7

[7] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018. 2

[8] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv*, abs/1911.05722, 2019. 2, 5

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778, 2016. 1, 7

[10] https://github.com/kanshichao/CBML. 5

[11] J. Huang, Q. Dong, S. Gong, and X. Zhu. Unsupervised deep learning by neighborhood discovery. In *in ICML*, pages 2849–2858, 2019. 7

[12] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 448–456, 2015. 7

[13] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Mining on manifolds: Metric learning without labels. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 7642–7651, 2018. 2, 3, 7

[14] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(1):117–128, 2011. 7

[15] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *2013 IEEE International Conference on Computer Vision Workshops, ICCV Workshops 2013, Sydney, Australia, December 1-8, 2013*, pages 554–561, 2013. 6

[16] Weiqing Min, Shuhuan Mei, Linhu Liu, Yi Wang, and Shuqiang Jiang. Multi-task deep relative attribute learning for visual urban perception. *IEEE Trans. Image Process.*, 29:657–669, 2020. 3

[17] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 6706–6716. IEEE, 2020. 2

[18] Yair Movshovitz-Attias, Alexander Toshev, Thomas K. Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 360–368, 2017. 2

[19] Binh X. Nguyen, Binh D. Nguyen, Gustavo Carneiro, Erman Tjiputra, Quang D. Tran, and Thanh-Toan Do. Deep metric learning meets deep clustering: An novel unsupervised approach for feature embedding. *ArXiv*, abs/2009.04091, 2020. 3, 7

[20] D. Parikh and K. Grauman. Relative attributes. In *2011 International Conference on Computer Vision*, pages 503–510, 2011. 3

[21] Pedro O. Pinheiro, Amjad Almahairi, Ryan Y. Benmaleck, Florian Golemo, and Aaron C. Courville. Unsupervised learning of dense visual representations. 2020. 2

[22] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 815–823, 2015. 2

[23] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 4004–4012, 2016. 2, 6

[24] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 1–9, 2015. 7

[25] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. *California Institute of Technology*, 2011. 6

[26] Jiayu Wang, Wengang Zhou, Guo-Jun Qi, Zhongqian Fu, Qi Tian, and Houqiang Li. Transformation GAN for unsupervised image synthesis and representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 469–478. IEEE, 2020. 2

[27] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R. Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 5022–5030, 2019. 2, 5

[28] Xinshao Wang, Yang Hua, Elyor Kodirov, Guosheng Hu, Romain Garnier, and Neil Martin Robertson. Ranked list loss for deep metric learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 5207–5216, 2019. 5

[29] Xun Wang, Haozhi Zhang, Weilin Huang, and Matthew R. Scott. Cross-batch memory for embedding learning. 2020. 1

[30] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 3733–3742, 2018. 7

[31] Mang Ye and Jianbing Shen. Probabilistic structural latent representation for unsupervised embedding. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 5456–5465. IEEE, 2020. 3, 7, 8

[32] M. Ye, J. Shen, X. Zhang, P. Yuen, and S. Chang. Augmentation invariant and instance spreading feature for softmax embedding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 3, 6, 7, 8

[33] Mang Ye, Xu Zhang, Pong C. Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 6210–6219, 2019. 1, 3, 6, 7

[34] Aron Yu and Kristen Grauman. Thinking outside the pool: Active training image creation for relative attributes. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 708–718. Computer Vision Foundation / IEEE, 2019. 3

[35] Andrew Zhai and Hao-Yu Wu. Classification is a strong baseline for deep metric learning. In *30th British Machine Vision Conference 2019, BMVC 2019, Cardiff, UK, September 9-12, 2019*, page 91. BMVA Press, 2019. 2

[36] Liheng Zhang, Guo-Jun Qi, Liqiang Wang, and Jiebo Luo. AET vs. AED: unsupervised representation learning by auto-encoding transformations rather than data. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 2019. 2