Unsupervised Visual Representation Learning via Dual-level Progressive Similar Instance Selection

Hehe Fan, Ping Liu, Mingliang Xu and Yi Yang

Abstract—The superiority of deeply learned representations relies on large-scale labeled datasets. However, annotating data is usually expensive or even infeasible in some scenarios. To address this problem, we propose an unsupervised method to leverage instance discrimination and similarity for deep visual representation learning. The method is based on an observation that convolutional neural networks (CNNs) can learn a meaningful visual representation with instance-wise classification, in which each instance is treated as an individual class. By this instance-wise discriminative learning, instances can reasonably distribute in the representation space, which reveals their similarities. In order to further improve visual representations, we propose a dual-level progressive similar instance selection (DPSIS) method to build a bridge from instance to class by selecting similar instances (neighbors) for each instance (anchor) and treating the anchor and its neighbors as the same class. To be specific, DPSIS adaptively selects two levels of neighbors, i.e., an “absolutely similar level” and a “relatively similar level”. Instances in the absolutely similar level are used as hard labels, while instances in the relatively similar level are used as soft labels. Moreover, during training, DPSIS is able to progressively select more and more neighbors without human supervision. At the beginning of training, because CNNs are weak, most instances are distributed relatively randomly in the representation space and only a few easy-to-recognize instances are selected as neighbors. As CNN models become stronger, the semantic meaning of each instance grows clearer. Those instances originally distributed in a relatively random manner gradually move to meaningful positions. This consequently facilitates CNN training since the number of reliable samples increases. Experiments on seven benchmarks, including three small-scale and two large-scale coarse-grained image classification datasets, and two fine-grained categorization datasets, demonstrate the effectiveness of our DPSIS.

Index Terms—Unsupervised learning, deep learning, image classification, fine-grained categorization.

I. INTRODUCTION

LARGE-SCALE learning methods based on convolutional neural networks (CNNs) [1], [2] constitute the recent advances in computer vision, and play an import role for visual perception in intelligent platforms such as robots [3], [4], drones [5] and self-driving cars [6], [7]. However, most of the previous works are conducted under supervised learning and therefore require a large amount of annotated data. The expensive labeling cost limits the scalability to real applications. To alleviate this problem, in recent years, learning deep visual representations without human annotations is attracting increasing attention from the community, resulting in a number of works to study how to learn deep visual representations in an unsupervised manner.

A natural strategy is to directly borrow classification frameworks from supervised learning and employ clustering algorithms to achieve pseudo labels for supervised training [8]–[10]. For example, Fan et al. [9] proposed to transfer pre-trained deep representations to unseen domains by iteratively applying clustering and fine-tuning. However, this clustering-based strategy relies on the estimation of the number of classes in datasets. Furthermore, when the scale of datasets becomes large, clustering will become a bottleneck because it consumes much computation and costs a lot of time.

Another strategy, i.e., self-supervised learning [11]–[13], is to find or exploit the relations (or correlations) between different input signals, e.g., modifying the input and predicting what changed or ensuring that the output does not change. For example, Gidaris et al. [13] proposed to learn image representations by training CNN models to recognize a 2D rotation. However, it is not easy to decide what kind of and how many correlations are sufficient to learn representations effective for downstream tasks.

Recently, instance-wise classification [14] has shown a
promising method for unsupervised visual representation learning. This method is based on an observation that, by directly treating each instance as an individual class, CNNs can learn meaningful visual representations. The distances between two instances in the representation space can reflect their similarities. Then, \( k \)-nearest neighbor (\( k \text{NN} \)) is used to select similar instances for each instance (anchor) as neighbors, which are treated as the same class as their corresponding anchors for training [15], [16].

However, anchors usually have significantly different numbers of neighbors in practice. It is not reasonable to adopt a fixed \( k \) for all anchors to assign the same number of neighbors. Moreover, the reliability of neighbors changes during training. At the beginning, CNNs are weak and neighbors are thus not reliable enough. In this case, only a few relatively reliable neighbors should be selected for training, so as to avoid noisy neighbors. When the model becomes stronger and neighbors become more reliable, the number of neighbors should be increased, in order to further improve the model. Therefore, \( k \text{NN} \) is not robust for similar instance selection.

In this paper, we propose a Dual-level Progressive Similar Instance Selection (DPSIS) method for instance-wise classification based unsupervised visual representation learning. Specifically, after each training round, which may consist of multiple epochs, DPSIS adaptively searches different numbers of neighbors for each anchor instance, according to similarity thresholds. We design two kinds of similar level, i.e., an "absolutely similar level" and a "relatively similar level".

Instances in different levels are treated in different ways. Neighbors in the absolutely similar level are closer to the anchor, and they are more likely to share the same class label as the anchor than those in the relatively similar areas. Consequently, with the instance-level classification framework, neighbors in the absolute level are used as hard instance labels, while neighbors in the relative level are used as soft instance labels. We illustrate the motivation of our DPSIS in Fig.1.

During training, our DPSIS has the ability to automatically select more and more similar instances in a progressive way. At the early stage of training, because CNNs are weak, instances are distributed relatively randomly in the representation space. In this case, only a few instances, which are easy to be recognized, are selected as neighbors. When CNNs become stronger, the instance distribution becomes more reasonable, and the instances of the same class become closer. As a result, more instances move into similar areas, which makes class boundaries more precise. This also relatively increases the scale of the selected similar training data, and in turn facilitates CNN models. We formulate this operation as self-paced learning (SPL) [17].

To evaluate the proposed method, we conduct coarse-grained image classification on CIFAR10/100 [18], SVHN [19], ImageNet 1K [20] and Places 205 [21]. We also conduct fine-grained categorization on CUB200-2011 [22] and Stanford Dogs [23]. Experimental results demonstrate that our method can adaptively discover correct neighbors and outperform a wide of state-of-the-art methods. The contributions of this article are threefold:

- Among the early efforts, we propose a progressive similar instance selection method for instance-wise classification based unsupervised visual representation learning, which can adaptively select more and more neighbors without human supervision.
- We propose a dual-level mechanism for similar instance selection. This learning strategy facilitates DPSIS to extract accountable knowledge from instances with different reliabilities in unsupervised learning. To the best of our knowledge, the proposed DPSIS is the first attempt to integrate dual-level selection with hard and soft loss functions into SPL.
- Extensive experiments on seven datasets indicate that the proposed method noticeably improves the unsupervised visual representation learning.

The rest of the paper is organized as follows. In Section II, we discuss related works and tasks. Section III presents the proposed DPSIS method in detail. In Section IV, we conduct extensive experiments to demonstrate the efficacy of our method. Finally, Section V presents our conclusions.

II. RELATED WORK

Deep learning based methods usually require large-scale labeled datasets for training [2], [24], [25]. There are two promising solutions to address the weakness. The first one is self-supervised learning, which aims to learn without human annotations by leveraging intrinsic structures in individual instances. The second one is unsupervised learning, which directly models distributions over the entire datasets.

To learn from internal structures in individual instances, self-supervised learning methods usually aim at finding or exploiting the relations (or correlations) between different input signals, e.g., modifying the input and (1) predict what changed (rotation angle from an original image [13]) or (2) ensure that the output does not change (if transformed images belong to the same category with a seed image [12]). There are also other special self-supervised learning tasks, such as learning spatial context of patches from an image [11], learning to count objects in an image [26], predicting the missing pixels in an image [27], recovering a plausible color version of a grayscale image [28] and solving Jigsaw puzzles [29]. However, it is unclear whether one or a few self-supervision tasks are sufficient for learning effective representations and which task would be considered more. For example, to obtain better visual representations, Doersch et al. [30] even attempted to combine several self-supervised tasks. For video understanding, deep neural networks can learn temporal structures via forecasting future frames in a self-supervised learning manner [31], [32].

To directly model distributions over entire datasets, clustering-based methods [8]–[10], [33], [34] enable the conventional supervised classification frameworks to be applied into unsupervised representation learning. Specially, these methods use clustering algorithms to assign pseudo labels to each instance. Then, the unlabeled instances and their pseudo labels are used to train deep models with supervised classification frameworks. The main challenge of clustering-based methods is to precisely estimate the number of classes (or clusters) on the entire training data. Moreover, since the
complexity of clustering is exponentially proportional to both the data and cluster size, clustering algorithms become a bottleneck as the number of data instances increasing.

Methods based on generative models also aim at learning the data distributions over entire datasets in an unsupervised manner, such as Restricted Boltzmann Machines (RBMs) [35]–[37], Autoencoders [38], [39], generative adversarial networks [40], variational auto-encoder [41] and unsupervised domain adaptation [42]–[45]. The leaned latent representations by generative models can benefit downstream tasks such as object recognition.

Unsupervised domain adaptation [46]–[49] is also related to the unsupervised representation learning. The goal of unsupervised domain adaptation is to transfer knowledge learned from source domains with a large number of annotated training examples to target domains with unlabeled data only. The difference is that, unsupervised representation learning is trained from scratch with any labeled data, and thus is more challenging than unsupervised domain adaptation. Besides, few-shot learning [50]–[53] also aims to reduce the cost of annotating data for deep neural network training.

Instance-wise classification [12], [14] is another family of methods to model the distribution of the unlabeled training data. The motivation is that supervised learning of deep neural networks with classification frameworks can automatically reflect the visual similarity correlation between different classes. Therefore, the visual representations produced by instance-wise classification could also help object recognition. However, these extreme instance-wise classification methods do not explore the class decision boundaries. An improvement [15], [16], [54] for this extreme instance-wise classification is to build a bridge from instance to class by exploiting similar neighbors. However, these methods adopt a k-nearest neighbor (kNN) search, which find a fixed number of neighboring instances and ignore instance density in the representation space. This simple kNN method fails to discover the class decision boundaries and thus limits the improvement for the instance-wise classification based unsupervised visual representation learning. Different from these methods [15], [16], [54], without a predefined fixed number of neighbors, our method guarantees a fixed region scale, thus making local neighbor selection more generalizable across representation space. Moreover, we design two-level neighborhoods, and our method can adaptively discover different level neighbors.

III. Method

In this section, we present our unsupervised visual representation learning method via dual-level progressive similar instance selection (DPSIS) in detail.

A. Preliminary: Instance-wise Classification

Suppose \( \phi(\cdot; \theta) \) is a convolutional neural network (CNN) model, in which \( \theta \) is the parameter of the neural network to be learned. The goal of \( \phi(\cdot; \theta) \) is to embed the \( i \)-th image \( x_i \) to a representation or feature vector \( F_i \in \mathbb{R}^{f \times 1} \), i.e., \( F_i = \phi(x_i; \theta) \), where \( f \) is the length of the vector. For instance-wise classification, each image is simply treated as an individual class. Therefore, instance-wise classification is essentially an unsupervised method because it does not require human annotations. Usually, the serial numbers or indexes of images in a dataset are used as instance labels. Suppose there are totally \( n \) images in a dataset. The goal of the instance-wise softmax classifier is to normalize a visual representation into a probability distribution over all instances. The instance-wise softmax can be formulated as follows:

\[
P(i|F_i) = \frac{\exp(W_i^T F_i)}{\sum_{j=1}^{n} \exp(W_j^T F_j)},
\]

where \( W_j \in \mathbb{R}^{f \times 1} \) is the weight vector for the \( j \)-th label. During training, increasing this probability will encourage similar instances to move closer and dissimilar instances to move away [14].

However, when the scale of the dataset increases, it is difficult for this parametric classifier to converge, which usually leads to low classification accuracy. Non-parametric instance-wise softmax classifier [14] is then proposed to alleviate the converge problem of the parametric classifier. The key component of the non-parametric classifier is a feature memory bank \( M \in \mathbb{R}^{f \times n} \), which stores all the visual representations of instances in the dataset. The non-parametric instance-wise softmax classifier is formulated as follows:

\[
P(i|F_i) = \frac{\exp(M_i^T F_i/\tau)}{\sum_{n=1}^{n} \exp(M_j^T F_i/\tau)},
\]

where \( M_i \in \mathbb{R}^{f \times 1} \) denotes the previous visual representation of the \( i \)-th image and \( \tau \) is a temperature parameter that controls instance distribution [55]. Specifically, using a higher value for \( \tau \) will produce a softer probability distribution over instances. Note that, for this non-parametric softmax classifier, the image representation \( F_i \) is \( l_2 \)-normalized. Therefore, the inner product of two image representations is the cosine distance, which can reflect the visual similarity of the two images. The memory features in \( M \) are updated by exponential moving average during training iteration as follows,

\[
M_i = (1 - \mu)M_i + \mu F_i,
\]

where \( \mu (\geq 0) \) is a hyper-parameter (update momentum) that controls memory update rate. The representations in the memory bank are initialized as random unit vectors.

Based on the non-parametric instance-wise softmax classifier Eq. (2), the goal of instance-wise classification is to minimize the following negative log-likelihood over the training data instances,

\[
L(\theta) = - \sum_{i=1}^{n} \log P(i|F_i) = - \sum_{i=1}^{n} \log P(i|\phi(x_i; \theta)).
\]

By this instance discrimination learning, instances in the feature space can distribute in a reasonable way in which their distances reveal their visual similarities.

B. Dual-level Progressive Similar Instance Selection

The problem of the extreme instance-wise classification Eq. (4) is that it does not explicitly discover the class decision boundaries, which limits the method to learn more
effective and reasonable visual representations. To alleviate this problem, we propose DPSIS to build a bridge from instance to class. Specifically, DPSIS selects similar instances (i.e., neighbors) for each instance (i.e., anchor) according to an absolute similarity threshold $\lambda$ and a relatively similarity threshold $\gamma$, where $\lambda > \gamma$, and treats the anchor and its neighbors as the same class. Instances whose similarities are greater than $\lambda$ will be selected as absolute neighbors to the anchor instances and their labels are used as hard labels, while instances whose similarities are greater than $\gamma$ but less than $\lambda$ will be selected as relative neighbors and their labels are treated as soft labels. Suppose $A, B \in \{0, 1\}^{n \times n}$ is a matrix that indicates whether any two instances are neighbors (with the element 1) or not (with the element 0). Before training, $A$ is initialized as the identity matrix $I_n$, and $B$ is initialized as the zero matrix $0_n$.

The element $A_{i,j}$ and is $B_{i,j}$ are defined as follows,

$$A_{i,j} = 1(M_j^T F_i \geq \lambda), \quad B_{i,j} = 1(\lambda > M_j^T F_i \geq \gamma),$$

where $i, j \in \{1, \ldots, n\}$. Then, the training objective of instance-wise classification with dual-level progressive similar instance selection is to minimize the following loss,

$$L(\theta) = -\sum_{i=1}^{n} \sum_{j=1}^{n} (A_{i,j} \log P(j|\phi(x_i; \theta)) + B_{i,j} \log P(j|\phi(x_i; \theta))),$$

where $\epsilon$ is a small positive number which aims to avoid division by zero.

During training, the index of the neighbor $x_j$, i.e., $j$, is added into the labels of the corresponding anchor $x_i$ by setting $A_{i,j}$ or $B_{i,j}$ to 1. As a result, the anchor is encouraged to move close to its neighbors. Since the anchor and its neighbors potentially share the same class label, the bridge from instance to class is constructed.

The DPSIS method introduces a two-level mechanism for neighbor selection. The neighborhood is divided into two areas according to two similarity thresholds, i.e., $\lambda$ and $\gamma$. To be

**ALGORITHM 1:** Unsupervised Learning via Dual-level Progressive Similar Instance Selection

**Input:** unlabeled data $\{x_i\}_{i=1}^{n}$, absolute similarity threshold $\lambda$, relative similarity threshold $\gamma$, convolution neural network $\phi(\cdot, \theta)$, feature memory bank $M$, number of training rounds $R$, number of epochs $E$ for each training rounds.

**Initialization:** model $\phi(\cdot, \theta)$ and memory $M$.

**Initialization:** $\theta \leftarrow \theta_0$, and $M \leftarrow I_n$.

for $r \leftarrow 1$ to $R$ do

for $e \leftarrow 1$ to $E$ do

for each minibatch in dataset do

extract feature: $F_i = \phi(x_i, \theta_1)$;

instance-wise classification with Eq. (6);

update memory $M$ with Eq. (3);

end

end

select neighbors, and update $A$ and $B$;

end
specific, the $\lambda$ defines the absolutely similar area, and the $\gamma$ defines the relatively similar area. The goal of $\lambda$ is to select the real neighbors who share the same class label as the anchor with little noise. The goal of $\gamma$ is to select the potential neighbors, which allows some noise to a degree. Since neighbors in the absolutely similar area are supposed to be more accurate, their labels are treated in the same way as the anchor label, i.e., with the weight 1 on the loss. For neighbors in the relatively similar area, their losses are divided by the number of neighbors in the area, i.e., with the weight $B_{i,j}/(||B_i||_1 + \epsilon)$ on the loss. In supervised learning, we can use Bayesian approaches [56] to find $\lambda$ and $\gamma$. For unsupervised learning, those methods are not applicable because related probability estimations are not available. However, as shown in Section IV, $\lambda = 0.9$ and $\gamma = 0.6$ achieve the best accuracy on all coarse-grained datasets, while $\lambda = 0.95$ and $\gamma = 0.85$ achieve the best accuracy on all fine-grained datasets. This indicates a universality of the thresholds. We illustrate our DPSIS method in Fig. 2.

The training algorithm is listed in Alg. 1. The core procedure consists of two steps. The first step is to train the CNN model with the instance-wise classification framework, as shown in Eq. (6). Similar to conventional training for deep neural networks, this part can be performed for several epochs, so that the model learns from the label information in $A$ and $B$ well. The second step is the neighbor selection. According to the newly learned visual representations in the first step, the matrix $A$ and matrix $B$ are updated.

Given the fixed similarity thresholds, i.e., $\lambda$ and $\gamma$, the proposed DPSIS method has the ability to automatically discover more and more neighbors in a progressive manner. At the beginning of training, because the CNN model is relatively weak, instances are distributed relatively uniformly in the visual representation space and most similar instances do not lie close enough. Therefore, only a few instances are selected as neighbors for each anchor. Then, as the model becoming strong, more instances are selected as neighbors because their visual representations become more meaningful. In this more precise feature space, instances of the same class lie close to each other. This progress will in return facilitate the model training because class boundaries are better discovered, and instance labels are more precise than the beginning. From another view of point, the increment of neighbors can be seen as that the scale of the data with class labels relatively increases. Deep neural networks usually benefit from more training data. As a result, the model becomes stronger and stronger.

We illustrate the progressive learning process in Fig. 3. This progress can be seen as a variant of self-paced learning [9], [17], [57], [58], [58]--[67]. The learning process of humans and animals usually starts with easier aspects of an aimed task, and then gradually takes more difficult examples into consideration. Inspired by this process, self-paced learning considers loss as easiness or reliability measurements, and embeds easiness or reliability identification into training. Usually, with a fixed easiness or reliability threshold, self-paced learning first selects samples with small losses as the easy or reliable samples to train the model and then gradually adds hard or less reliable samples for training. This learning strategy has been demonstrated to be beneficial in avoiding bad local minima and in achieving a better generalization result. In this paper, we consider the easiness or reliability in self-paced learning as the similarity between anchors and neighbors. First, the instances with enough high similarity are discovered as neighbors to train the model. Then, more instances are automatically selected as neighbors. Note that, existing self-paced learning methods only exploit one-level easiness or reliability. However, we design a dual-level selection mechanism to facilitates CNN models to extract faithful knowledge from highly unreliable similar instances in unsupervised learning.

The learned visual representations can be used for class-wise classification. To evaluate the proposed method, we perform linear Support Vector Machine (SVM) on the CNN intermediate features, and the weighted $k$NN on the CNN output features, respectively. Linear SVM [14] is a parametric method and requires class labels to be trained. The visual representations can be extracted from different network layers. This method is realized by a fully connected layer optimized by the cross-entropy loss function. Therefore, linear SVM is, in essence, a supervised method. In contrast, the weighted $k$NN method [14] is a non-parametric method. Specifically, for a test image, we first extract its feature with the trained deep model $\phi(\cdot; \theta)$. Second, using the cosine distance metric, the similarities between the test image and all training instances are computed with the test image feature and the feature memory bank. Third, the top $k$ nearest neighbors are used to make the class prediction via weighted voting. The weighted vote of the image $x$ to be the class $c$ is calculated as follows,

$$g(\hat{x}, c) = \sum_{x \in N_k} \exp(M_x^T F_{\hat{x}}/\tau) \cdot 1(c_x = c),$$

where $N_k$ is the $k$ nearest neighbors of $x$ and $c_x$ denotes the class label of $x$. At last, the predicted label of the test image $\hat{x}$ is achieved by

$$c_{\hat{x}} = \arg\max\{g(\hat{x}, c)\}, c \in \{1, ..., m\},$$

where $F_{\hat{x}}$ is the feature vector of the test image $\hat{x}$.
where \( n \) is the number of classes. Different from linear SVM, without the requirement of an extra parametric fully connected layer, weighted \( k \)NN can directly evaluate the discriminative capability of the learned visual representations.

**Computational complexity analysis.** For a dataset which has \( n \) images, the complexity of calculating similarity matrix is \( O(n^2) \). For neighbor selection, we need to compare \( n \) times for each image. Therefore, the selection complexity is \( O(n^2) \), too. Finally, the computational complexity of our method is \( O(n^2) \). Note that, for the \( k \)NN-based AND method, there is a sort for each image. Even though we adopt Quicksort, the computational complexity is up to \( O(n^2 \log n) \). Therefore, our DPSIS is more computationally efficient than AND.

**IV. Experiments**

To demonstrate the efficacy of our method, we conduct extensive experiments on seven datasets, including three small-scale datasets, two large-scale datasets, and two fine-grained datasets.

**A. Datasets**

1) **Small-scale datasets:**
- CIFAR10/100 [18]. The dataset contains 50,000 images for training and 10,000 images for test. The images are from 10/100 object classes. Each class has 6,000/600 images with size \( 32 \times 32 \).
- SVHN [19]. The dataset is about Street View House Numbers, which consists of \( 32 \times 32 \) digit images with 10 digit classes. The training set includes 73,257 digit images and the test set includes 26,032 digit images.

2) **Large-scale datasets:**
- ImageNet 1K [20]. This is a large 1,000 classes object dataset with 1.2 million images for training and 50,000 for test.
- Places 205 [21]. A scene-centric dataset with 205 scene categories and 2.5 millions of images. The training set contains 2,448,873 images from 205 categories of Places with minimum 5,000 and maximum 15,000 images per category. The validation set contains 100 images per category and the test set contains 200 images per category (a total of 41,000 images). Following [14], we directly use the features learned on ImageNet to evaluate our method on this dataset, without any fine-tuning.

3) **Fine-grained datasets:**
- CUB200-2011 [22]. The dataset contains 200 bird species, with 5,994 for training and 5,794 for evaluation.
- Stanford Dogs [23]. The dataset contains 120 dog breeds, with 12,000 for training and 8,580 for test.

**B. Experimental Settings and Competitors**

The hyper-parameters are chosen via empirical validation. In particular, we set temperature \( \tau = 0.1 \). The update momentum \( \mu \) for the feature memory bank is set to 0.5. The dimension of visual representation is set to 128. Following [16], we perform 5 training rounds in total, and each training round contains 200 epochs. The model is trained using SGD with momentum. The batch size is set to 256 for ImageNet and 128 for other datasets. The learning rate is initialized to 0.03 and scaled down with a coefficient of 0.1 every 40 epochs after the first 120 epochs. The number of \( k \) for the weighed \( k \)NN evaluation is set to 200.

We compare our method with a range of methods, including:
- Self-supervised learning: Context [11], Color [28], Jigsaw [29], Split-Brain [68] and Count [26];
- Clustering-based method: DeepCluster [10];
- Adversarial learning: Adversarial [69];
- Instance-wise classification: Exemplar [12], Instance [14] and AND [16];
- Others: Video [70] and NAT [71].

**C. Evaluation on Small-scale Image Classification**

In this section, we conduct image classification on CIFAR10/100 [18] and SVHN [19]. By default, the absolute similarity threshold \( \lambda \) is set to 0.9, and the relative similarity threshold \( \gamma \) is set to 0.6. For the linear SVM evaluation, we use Conv5 features.

1) **Comparisons to the State-of-the-Art Methods:** Experimental results of our DPSIS and other unsupervised learning methods are listed in Tab I. With linear SVM, our method achieves 78.5% on CIFAR10, 48.8% on CIFAR100, and 94.3% on SVHN, outperforming all the state-of-the-art methods. This suggests the performance superiority of our instance discrimination and similarity learning method over alternative methods for unsupervised representation learning. By using the weighted \( k \)NN classification on the final 128 dimensional features, our method achieves 75.7% on CIFAR10, 42.3% on CIFAR100, and 91.8% on SVHN, still outperforming all methods.

The experimental results confirm the effectiveness of neighborhood discovery for instance discrimination learning. For example, when using the weighted \( k \)NN classification on SVHN, the pure instance-wise classification, *i.e.*, the Instance method, only achieves an accuracy of 79.8%. The accuracy is even lower than that of the previous DeepCluster method. The reason may be that instance similarities or relationships are especially important for these small-scale datasets. DeepCluster uses clustering to explore instance similarities while Instance does not exploit sample relationships. However, when adding neighborhood discovery, *i.e.*, AND and DPSIS, the accuracies increase to 90.9% and 91.8%, respectively.

The comparison results between DPSIS and AND demonstrates that the adaptive dual-level neighborhood discovery mechanism in our DPSIS outperforms the \( k \)NN-based and curriculum-learning-based mechanism in AND. For example, when using weighted \( k \)NN classification on CIFAR10, DPSIS achieves an accuracy of 75.7%, outperforms AND by 0.9%. This confirms that our threshold-based method, which selects different numbers of neighbors, is more reasonable than the \( k \)NN-based method.

2) **CNN Model with Different Backbones:** As the network architecture has a significant impact on the performance, we apply a few typical architectures, *i.e.*, AlexNet, ResNet-18, ResNet-50, and ResNet-101, on CIFAR10 with the weighted...
TABLE I
Top-1 Accuracy (%) on CIFAR10, CIFAR100 and SVHN with AlexNet.

<table>
<thead>
<tr>
<th>Method</th>
<th>CIFAR10</th>
<th>CIFAR100</th>
<th>SVHN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear SVM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Supervised</td>
<td>91.8</td>
<td>71.0</td>
<td>96.1</td>
</tr>
<tr>
<td>Split-Brain [68]</td>
<td>67.1</td>
<td>39.0</td>
<td>77.3</td>
</tr>
<tr>
<td>Count [26]</td>
<td>50.9</td>
<td>18.2</td>
<td>63.4</td>
</tr>
<tr>
<td>DeepCluster [10]</td>
<td>77.9</td>
<td>41.9</td>
<td>92.0</td>
</tr>
<tr>
<td>Instance [14]</td>
<td>70.1</td>
<td>39.4</td>
<td>89.3</td>
</tr>
<tr>
<td>AND [16]</td>
<td>77.6</td>
<td>47.9</td>
<td>93.7</td>
</tr>
<tr>
<td>DPSIS (ours)</td>
<td>78.5</td>
<td>48.8</td>
<td>94.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>CIFAR10</th>
<th>CIFAR100</th>
<th>SVHN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weighted kNN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Supervised</td>
<td>91.9</td>
<td>69.7</td>
<td>96.5</td>
</tr>
<tr>
<td>Split-Brain [68]</td>
<td>67.1</td>
<td>39.0</td>
<td>77.3</td>
</tr>
<tr>
<td>Count [26]</td>
<td>50.9</td>
<td>18.2</td>
<td>63.4</td>
</tr>
<tr>
<td>DeepCluster [10]</td>
<td>77.9</td>
<td>41.9</td>
<td>92.0</td>
</tr>
<tr>
<td>Instance [14]</td>
<td>70.1</td>
<td>39.4</td>
<td>89.3</td>
</tr>
<tr>
<td>AND [16]</td>
<td>77.6</td>
<td>47.9</td>
<td>93.7</td>
</tr>
<tr>
<td>DPSIS (ours)</td>
<td>78.5</td>
<td>48.8</td>
<td>94.3</td>
</tr>
</tbody>
</table>

TABLE II
Top-1 Accuracy (%) on CIFAR10 using weighted kNN with different backbones.

<table>
<thead>
<tr>
<th>Network</th>
<th>AlexNet</th>
<th>ResNet18</th>
<th>ResNet50</th>
<th>ResNet101</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance [14]</td>
<td>74.8</td>
<td>80.8</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>AND [16]</td>
<td>75.3</td>
<td>87.1</td>
<td>89.7</td>
<td>90.8</td>
</tr>
<tr>
<td>DPSIS (ours)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TABLE III
Top-1 Accuracy on CUB200 and Dogs with ResNet-18 and weighted kNN.

<table>
<thead>
<tr>
<th>Network</th>
<th>CUB200</th>
<th>Dogs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance [14]</td>
<td>11.6</td>
<td>27.0</td>
</tr>
<tr>
<td>AND [16]</td>
<td>14.4</td>
<td>32.3</td>
</tr>
<tr>
<td>DPSIS (ours)</td>
<td>16.4</td>
<td>36.1</td>
</tr>
</tbody>
</table>

$k$NN classification. Experimental results are listed in Tab II. With the CNN model becoming deeper, our DPSIS outperforms Instance and AND more. For example, when using ResNet-18, DPSIS outperforms AND by 0.8%. When using ResNet-101, DPSIS outperforms AND by 2.4%. This indicates that our method makes better use of deeper networks than other methods.

D. Evaluation on Large-scale Image Classification

In this section, we conduct image classification on the large-scale ImageNet [20] and Places [21] datasets. As these are also coarse-grained datasets, we use the same absolute similarity threshold and relative similarity threshold as the small-scale image classification’s, i.e., $\lambda = 0.9$ and $\gamma = 0.6$. For the linear SVM evaluation, we use the features from Conv1 to Conv5 and fully-connected (FC) layer. Experimental results are reported in Tab. IV. The “Random” method means that the parameters of the network are randomly initialized as a lower bound. Note that, for the results of our DPSIS method on Places, we follow the Instance method [14], in which features are trained on ImageNet without any fine-tuning on Places.

On ImageNet, with linear SVM, our method achieves an accuracy of 40.8% using Conv4 activation, an accuracy of 38.7% using Conv5 activation and an accuracy of 37.4% using FC, outperforming all the state-of-the-art methods. For example, with the Conv4 feature, DPSIS outperforms DeepCluster by 1.2%. Among the instance-wise classification methods, i.e., Instance, AND and DPSIS, our DPSIS method achieves the best accuracy on all intermediate features except for Conv1. For example, with the Conv3 feature, DPSIS outperforms AND by 0.7%. This demonstrates that our DPSIS can learn better intermediate features. With weighted $k$NN, our method achieves an accuracy of 37.4%, outperforming all other methods. Note that, compared with Instance, the $k$NN-based AND method fails to make an improvement. This is probably because that ImageNet is diverse and the number of neighbors to different images varies considerably. The $k$NN-based method cannot handle this scenario well. However, compared with Instance, our adaptive selection method still achieves an improvement of 0.6%, manifesting that DPSIS effectively exploits the potential relationship between images.

On Places, with linear SVM, our method achieves the best accuracy using Conv5 and FC, with an accuracy of 35.0% and an accuracy of 34.1%, respectively. Although directly based on features learned on ImageNet, our DPSIS outperforms other methods whose features are learned on Places, demonstrating the effectiveness of our method. With weighted $k$NN, our DPSIS outperforms Instance by 2.1%. This reflects the improvement caused by our similar instance selection mechanism.

E. Evaluation on Fine-grained Image Classification

In this section, we conduct image classification on the fine-grained CUB200 [22] and Dogs [23] datasets, which are significantly under-studied in unsupervised learning context. Compared with the coarse-grained datasets, the difference between fine-grained images is small. Therefore, we increase the similarity thresholds. Specifically, the absolute similarity threshold $\lambda$ is set to 0.95 and the relative similarity threshold $\gamma$ is set to 0.85. Experimental results with weighted $k$NN are reported in Tab. III.

Among these three instance-wise classification methods in the table, our DPSIS achieves the best accuracy, with an accuracy of 16.4% on CUB200 and an accuracy of 36.1% on Dogs. Our DPSIS outperforms AND by 2.0% on CUB200 and 3.8% on Dogs. This demonstrates the advantage of our adaptively neighbor discovery compared to the $k$NN-based AND method for instance discrimination learning on fine-grained image classification.

F. Ablation Study

1) Dual-level Selection: To study the influence of $\lambda$ and $\gamma$ on classification accuracy, we conduct the weighted $k$NN classification on CIFAR10, CIFAR100, and SVHN with different $\lambda$ and $\gamma$. We also evaluate two cases where only absolutely similar level or relatively similar level is used. Experimental results are listed in Tab V.
The combination of $\lambda = 0.9$ and $\gamma = 0.6$ achieves the highest accuracy among the datasets. For example, on CIFAR10, this combination achieves an accuracy of 89.68%. When the $\lambda$ decreases or $\gamma$ increases, the accuracy decreases. For example, when $\lambda = 0.7$ and $\gamma = 0.6$, the accuracy decreases to 86.93%. This is because a small $\lambda$ introduces too much noise in the absolutely similar area. When $\lambda = 0.9$ and $\gamma = 0.85$, the accuracy decreases to 88.69%. This is because a large $\lambda$ reduces the number of neighbors in the relatively similar area, which increases the challenge for DPSIS to discover class boundaries when given only a few neighbors. It also demonstrates that DPSIS is robust because the same $\lambda$ and $\gamma$ the highest accuracy among the datasets.

The experimental results also confirm the effectiveness of the dual-level neighborhood mechanism. The single-level neighborhood mechanism obviously decreases accuracy. For example, on the SVHN dataset, the combination of $\lambda = 0.9$ and $\gamma = 0.6$ achieves an accuracy of 92.37%. However, when using single absolutely similar instance selection, $i.e.$, $\lambda = 0.9$ and $\gamma = 0$, the accuracy decreases to 90.74%. When using single relatively similar instance selection, $i.e.$, $\lambda = 0$ and $\gamma = 0.6$, the accuracy decreases to 89.70%.

2) Progressive Selection: The proposed DPSIS method is able to automatically discover more and more neighbors. To prove this ability, we show the average number of absolute neighbors and relative neighbors per anchor instance with different similarity thresholds in Fig. 4. With training, more neighbors are discovered in the absolutely similar area. This also demonstrates that DPSIS is robust because the same $\lambda$ and $\gamma$ the highest accuracy among the datasets.
and more instances are added into the absolutely similar area and relatively similar area. As we proposed, this will benefit the CNN model training since the number of training data with reliable pseudo labels increases. Therefore, as shown in the figure, the evaluation accuracy gradually increases.

Lower similarity thresholds lead to more neighbors. For example, for the absolute similarity threshold \( \lambda \), when it equals to 0.9, only a few instances are selected as absolute neighbors. When it equals to 0.6, the number of absolute neighbors increases significantly.

The absolute neighbor selection and relative neighbor selection affect each other. In other words, they are not independent. For the case of \( \lambda = 0.8, \gamma = 0.6 \) and the case of \( \lambda = 0.9, \gamma = 0.6 \), although the relative similarity thresholds are the same, the relative neighbors with \( \lambda = 0.8 \) are more than the relative neighbors with \( \lambda = 0.9 \). This is because a low similarity threshold will change the entire visual representation distribution. It will allow too many instances to be neighbors and therefore excessively encourage instances to move close while neglect the discrimination among instances.

We also show the neighbor consistency, i.e., the proportion of neighbors sharing the same class as their anchors, in Fig. 4. When \( \lambda = 0.8 \), the consistency drops obviously. When \( \lambda = 0.9 \), the consistency drops slightly. This is the reason \( \lambda = 0.9 \) achieves a higher accuracy than \( \lambda = 0.8 \).

V. CONCLUSION

In this paper, we propose an unsupervised visual representation learning method that leverages instance discrimination and similarity. Compared with pure instance-wise discriminative learning, our method builds a bridge from instance to class by treating an instance and its neighbors as the same class. Besides, we divide the neighborhood into an absolutely similar area and a relatively similar area, in which labels of the neighbors are treated as hard labels and soft labels, respectively. Experimental results on CIFAR10/100, SVHN, ImageNet, Places, CUB200 and Dogs demonstrate the effectiveness of our method. The proposed method also has the potential application on unsupervised learning for video visual representation.

REFERENCES


