Local discriminative regions play important roles in fine-grained image analysis tasks. How to locate local discriminative regions with only category label and learn discriminative representation from these regions have been hot spots. In our work, we propose Searching Discriminative Regions (SDR) and Learning Discriminative Regions (LDR) method to search and learn local discriminative regions in images. The SDR method adopts attention mechanism to iteratively search for high-response regions in images, and uses this as a clue to locate local discriminative regions. Moreover, the LDR method is proposed to learn compact within category and sparse between categories representation from the raw image and local images. Experimental results show that our proposed approach achieves excellent performance in both fine-grained image retrieval and classification tasks, which demonstrates its effectiveness.

**key words:** fine-grained, image retrieval, image classification, attention, metric learning

1. Introduction

Different from general image analysis tasks, Fine-grained Image Analysis (FGIA) tasks aim to analyze sub-categories of images that belong to the same meta-category, such as different species of birds. In fine-grained images, different categories have the similar shape, size and even textures, and the same category could have large differences due to different poses and views. The large intra-class differences and subtle inter-class differences make FGIA tasks challenging for most mainstream deep learning networks.

To address this issue, many researches were dedicated to extracting local discriminative features to obtain better learning and inference performance. To this end, some researches [1], [2] chose to utilize auxiliary bounding boxes or part annotations, which have greatly improved the performances. However, the cost of obtaining the bounding boxes labeling of the image is expensive, and the artificial labeling of the bounding boxes does not conform to the actual application. To benefit from the performance gains brought by the bounding boxes with only category labels, more and more researches tend to use category labels to locate objects or objects’ parts under weak supervision, which has become a hot spot in fine-grained image analysis tasks. The attention mechanisms were widely adopted to locate objects in weakly supervised object location tasks. By trained with the softmax-like loss function, the attention maps could have the highest response value in the most discriminative region, which provides enlightening clues for locating discriminative regions in images. However, it may be insufficient to distinguish a category from other similar categories with only one local discriminative region. For example, as illustrated in Fig. 1, the head region is usually the most discriminative in bird classification, but there is almost no difference between the head regions of species Herring Gull and species Heermann Gull. The difference between these two species is mainly the color of beak and legs. Intuitively, locating and analyzing more local discriminative regions is beneficial to fine-grained image analysis tasks.

In our work, we propose a novel approach to search and learn multi discriminative regions in images for fine-grained image retrieval and classification tasks. To search more local discriminative regions in images, we proposed Searching Discriminative Regions (SDR) method. The SDR method adopts attention mechanism to locate the most discriminative regions in images, and by recursively dropping the previous discriminative region and adopting the attention mechanism, the method could locate multi discriminative regions for fine-grained image analysis tasks. Moreover, in Learning Discriminative Regions (LDR), we introduce a variant of arcface loss [3] named adaptive margin arcface (arc-am) loss to make the learned features more sparse between different categories and more compact within same category.

We evaluate our approach on three benchmarks fine-grained image datasets, namely CUB-200-2011 [4], FGVG-Aircraft [5] and Stanford-Cars [6]. Experimental results on fine-grained image retrieval task show that our approach achieves state-of-the-art performance with R@1 accuracy

---

**Fig. 1** Illustration of the difficulty in fine-grained image analysis. Herring Gull and Heermann Gull have similar shape and texture on head, body and other parts. Without paying attention to local discriminative regions (beak and legs regions in this figure), it is difficult to distinguish these two species.
of 79.7% and 94.6% on CUB-200-2011 and Stanford-Cars datasets, respectively. Similarly, experimental results on fine-grained image classification task show that our approach achieves strong performance with accuracy of 88.9% on CUB-200-2011 dataset and state-of-the-art performance with accuracy of 94.3% and 95.2% on FGVG-Aircraft and Stanford-Cars datasets, respectively.

2. Related Work

Fine-grained image retrieval. Given a query image, fine-grained image retrieval tasks aim to return images that are in the same category as the query image from a given fine-grained dataset. As open-set tasks, the testing categories are different from the training categories, which makes fine-grained image retrieval is more challenging than classification tasks. Wei et al. [7] adopted deep learning to solve Fine-grained Image Retrieval (FGIR) tasks, which made a great process. Zhai et al. [8] shown that classification-based metric learning approaches could achieve strong performance not only in face verification but fine-grained and general image retrieval tasks. Zheng et al. [9] proposed a centralized ranking loss to train networks and an effective weakly supervised framework to locate objects. Teh et al. [10] improved ProxyNCA [11] to get state-of-the-art performance on FGIR tasks.

Fine-grained image classification. Fine-grained image classification is a base task in fine-grained image analysis tasks. Since the differences between fine-grained categories are very subtle, the features extracted by ordinary CNN are not sufficiently discriminative for classification. To address this issue, researches on fine-grained image classification could be mainly divided into two directions. One direction is to design better CNNs to extract more discriminative features, like bilinear pooling method [12]. The other direction is to utilize object position information to provide CNNs with more refined input data, which has been the mainstream direction in fine-grained image classification. To benefit from object position information under the supervision of only category labels, many researches [13]–[15] adopted the location-classification approach. In the location process, Hu et al. [13] proposed to adopt attention mechanisms to locate objects’ parts; Yang et al. [15] proposed to locate object with Region Proposal Network (RPN) [16]. In the classification process, Zheng et al. [17] proposed to utilize knowledge distillation to force the main branch to learn the information from the part branch as an alternative feature fusion option. Yang et al. [15] concatenated original feature and partial features together to get the final classification. The location-classification methods similar to [13], [15], [17] had greatly improved the classification accuracy of fine-grained visual recognition tasks on related datasets.

3. Approach

In this section, we introduce our proposed approach in detail. Firstly, we detail SDR method which is utilized to locate local discriminative regions in Sect. 3.1. Then, we introduce LDR method in Sect. 3.2.

3.1 Searching Discriminative Regions

The convolutional neural networks maintain the spatial structure of the image during the learning process, which means that there is a position correspondence in space between the feature maps and the input image. Therefore, the most discriminative region could be located on the image according to the highest response region on the feature maps.

In our work, attentional bilinear pooling method [13], [18] is adopt to generate attention maps. It is assumed that \( F \) is the feature maps obtained from the CNN network, and \( A \) is the attention maps produced by \( 1 \times 1 \) convolutional layer from the feature maps. Attentional bilinear pooling is defined as:

\[
B = A^T F
\]  

where \( B \in \mathbb{R}^{m \times c} \) is the output of attentional bilinear pooling layer, \( A \in \mathbb{R}^{1 \times m} \), and \( F \in \mathbb{R}^{l \times s \times c} \). \( m \) and \( c \) are the number of channels, and \( s \) is the spatial size of the attention and feature maps. For each element \( B_{i,j} \) in \( B \), there are:

\[
B_{i,j} = (A^T)^t F_{j} = \sum_{k=1}^{c} A_{i,k} F_{j,k}
\]  

where \( B_{i,j} \) is the product of spatial location of the original feature in channel \( i \) of \( A \) and channel \( j \) of \( F \). \( A' \) and \( F' \) mean the feature on channel \( i \) of \( A \) and channel \( j \) on \( F \) respectively, and \( k \) means the coordinate on the space axis. With trained by back-propagation algorithm, \( A_k \) could be regarded as the richness of information on coordinate \( k \), which provide a basis for locating the most discriminative region.

It is assumed that \( I \) is the input image, the mean of the attention maps on \( I \) is determined as \( mA \) in Eq. (3).

\[
mA = Mean(Att(I)) = \frac{1}{m} \sum_{i=1}^{m} A_i
\]  

where \( Att \) is the function of attention mechanism. The foreground mask \( M^f \) for locating the most discriminative region is defined as:

\[
M^f_k = \begin{cases} 
0 & \text{if } mA_k < \delta_h \\
1 & \text{if } mA_k \geq \delta_h 
\end{cases}
\]  

The background mask \( M^b \) for marking the background region is defined as:

\[
M^b_k = \begin{cases} 
1 & \text{if } mA_k \leq \delta_l \\
0 & \text{if } mA_k > \delta_l 
\end{cases}
\]  

The most discriminative local region \( L \) is defined as the smallest rectangle region which contains all coordinates with \( M^f_k = 1 \) in the raw image \( R \). We define the location
To locate more discriminative regions in the image, we propose SDR method which is illustrated in Fig. 2. To force the attention mechanism to focus on the next discriminative region, we drop the foreground \( L \) and the background region in the input image to get the drop image \( D \).

\[
D = (1 - M^b) \ast (1 - \text{Rec}(M^f)) \ast I \\
= (1 - M^b) \ast (I \ominus L)
\]  

(7)

where \( \text{Rec}(M^f) \) means expanding the foreground region in \( M^f \) to a smallest rectangle with pixel value of 1, and \( I \ominus L \) means dropping the foreground rectangular region \( L \) from the input image \( I \), which is as illustrated in Fig. 2. Specifically, the pixel value of the \( L \) region in \( I \) is set to 0.

By iteratively dropping the previous discriminative region and locating the next discriminative region, the SDR method could propose a set of local images \( \{L_1, L_2, \ldots, L_n\} \) from high to low information. Moreover, we could locate a more complete local object image \( L_o \) by merging the proposed local regions, which is shown in Fig. 2.

To further understand our SDR algorithm, we present the algorithm in Algorithm 1.

**Algorithm 1 The SDR algorithm**

**Input**: input image \( I = R \), \( i = 1, n, \delta_h, \delta_l \)

**Output**: local images \( \{L_1, L_2, \ldots, L_n, Lo\} \)

**while** \( i \leq n \) **do**

1. Generate attention maps \( A \).
2. Generate \( mA \) by averaging the attention maps on channels as Eq. (3).
3. Binarize \( mA \) into \( M^f \) and \( M^b \) according to the threshold \( \delta_h \) and \( \delta_l \) as Eq. (4) and Eq. (5).
4. Locate the discriminative region and sample local image \( L_i \) from raw image \( R \).
5. Generate the drop image \( D \) as Eq. (7).
6. \( i \leftarrow i + 1, I \leftarrow D \).

**end while**

7. Merge the local images \( \{L_1, L_2, \ldots, L_n\} \) to \( L_o \)
3.2 Learning Discriminative Regions

Given these local images \{L_1, L_2, \ldots, L_n, L_o\}, the CNN baseline model could extract a set of local embedding features \{em_1, em_2, \ldots, em_n, em_o\}. To learn a discriminative representation from the raw features \em_i and these local features, we design Learning Discriminative Regions method.

In LDR, we introduce a variant of arcface loss [3] named adaptive margin arcface (arc-am) loss to make the learned features more sparse between different categories and more compact within the same category.

The traditional softmax loss is presented as follows:

\[
L_{\text{softmax}} = -\log \tilde{y} = -\log e^{W^T x} / \sum_{c=1}^{C} e^{W^T x},
\]

where \tilde{y} is the predicted probability of class label \(y\), \(W^T x\) means the \(j\)-th logit output from the last fully connected layer, and for simplicity, the bias \(b_j\) is determined as zero. \(x \in \mathbb{R}^d\) means the embedding feature, and \(y\) is the ground-truth class label of \(x\). \(W_j \in \mathbb{R}^d\) means the \(j\)-th column of the weight \(W \in \mathbb{R}^{d \times C}\). \(C\) means the class number.

In arcface loss, the logit is transformed as follows:

\[
W^T x = ||W_g|| \|x\| \cos \theta_g
\]

where \(\theta_g\) means the angle between the weight \(W_g\) and the feature \(x\).

By applying an additive angular margin penalty \(m\) on the angle \(\theta_g\), the arcface loss could make the angular separation between different categories, which is shown in Fig. 3. Moreover, with the normalization process on \(W\) and \(x\), the predictions only depend on the angle \(\theta_g\). With the re-scale process, the logit could be determined as \(s(\cos \theta_g)\), where \(s\) is the re-scale scalar meaning the radius of the feature hypersphere. Therefore, the arcface loss could be presented as follows:

\[
L_{\text{arc}} = -\log \frac{e^{s(\cos \theta_g + m)}}{\sum_{j=1,j \neq g}^{C} e^{s(\cos \theta_j)}}
\]

In our work, to further learn a good representation from hard examples, we introduce a method for adaptively determining the margin \(m\). The adaptive margin \(m_a\) is determined as follows:

\[
m_a = (2 - \tilde{y}) m
\]

As for hard samples, the probability \(\tilde{y}\) is small, close to 0, which could make a large angular margin penalty \(m_a\). As for easy samples, the probability \(\tilde{y}\) is large, close to 1, which could make a small \(m_a\). By applying more penalties on hard samples, the network could further narrow the distance between the hard sample features and the samples’ feature center.

In the training process, we use multiple losses to jointly guide feature extraction, including the raw loss \(L_{\text{raw}}\), the local loss \(L_{\text{local}}\), the drop loss \(L_{\text{drop}}\), and the joint loss \(L_{\text{joint}}\) shown in Eq. (12).

\[
\begin{align*}
L_{\text{raw}} & = L_{\text{arc-am}}(C_i(em_i)) \\
L_{\text{drop}} & = \sum_{i=1}^{n-1} L_{\text{arc-am}}(C_d(em_d)) \\
L_{\text{local}} & = \sum_{i=1, l_0} L_{\text{arc-am}}(C_l(em_l)) \\
L_{\text{joint}} & = L_{\text{arc-am}}(C_f(em_f))
\end{align*}
\]

where \(C_r, C_d, C_l, C_f\) are classifiers that map embedding features to category logists, and \(C_r\) and \(C_d\) share same parameters. \(em_f\) is the fused embedding features as Eq. (13).

\[
em_f = \text{Fusion}(em_1, em_2, \ldots, em_n, em_o, em_o)
\]

where \(\text{Fusion}\) means the fusion module, which is a 1D conv layer. The size of the fused embedding features could be determined by adjusting the channel size \(H\) of the 1D conv layer. Assuming that the channel size of the 1D conv layer is \(H\), the size of the fused embedding features could be determined as \((n + 2)H\).

The final loss is determined as follows:

\[
L = \frac{\alpha}{n} (L_{\text{raw}} + L_{\text{drop}}) + \frac{\beta}{n + 1} L_{\text{local}} + \gamma L_{\text{joint}}
\]

where \(\alpha, \beta, \gamma\) are the weight coefficients.

To further understand our training process, we present the training algorithm in Algorithm 2.

\textbf{Algorithm 2} The training algorithm

\begin{algorithmic}[1]
\Algrestore
\Function{\text{Train}}{\text{images } R, \text{epochs } E, \text{and } M, n, m, \delta_0, \delta_1, \alpha, \beta, \gamma}
\State \textbf{Input:} raw images } R, \text{ epochs } E, \text{ and } M; n, m, \delta_0, \delta_1, \alpha, \beta, \gamma
\State \textbf{Output:} predict the final logit
\While{the trained epochs < } E \Do
\State 1. Get the raw image } R and the category label } Y.
\State 2. Generate feature maps } F_j, \text{ attention maps } A_i, \text{ embedding features } em_i.
\State 3. Locate } n \text{ discriminative local images, and the object location image } L_o.
\State 4. Generate local embedding features, and fuse these with raw embedding features into } em_f.
\State 5. Generate the raw predict logit, the local predict logit, the drop predict logit, and the final predict logit.
\State 6. Calculate loss function } L_{\text{raw}}, L_{\text{local}}, L_{\text{drop}}, \text{ and } L_{\text{joint}}.
\State 7. Calculate gradient with back-propagation algorithm, and update the network with SGD.
\EndWhile
\EndFunction
\end{algorithmic}
4. Experiments

In this section, we describe the datasets we used, implementation details, and performance on fine-grained image retrieval and classification tasks.

4.1 Datasets

In our work, we conduct our proposed approach on three benchmark datasets, including CUB-200-2011 [4], FGVG-Aircraft [5] and Stanford-Cars [6]. In our all experiments, we train our networks in a weak supervision manner, which means that no bounding boxes or part annotations are used.

In fine-grained image retrieval tasks, CUB-200-2011 and Stanford-Cars datasets are used for evaluation. The CUB-200-2011 dataset contains 200 bird classes and 11,788 images. We follow the general method for dataset division [8], [10], [19]. The first 100 classes with 5,864 images in CUB-200-2011 dataset are used for training, and the remaining 100 classes with 5,924 images are used for testing. The Stanford-Cars dataset contains 196 car classes and 16,185 images. Similarly, the first 98 classes with 8,054 images are used for training, and the remaining 98 classes with 8,131 images are used for testing. The details of the two datasets are shown in Table 1.

In the fine-grained image classification tasks, all the three datasets are used for evaluation. We follow the standard division of training set and testing set. The details of these datasets are shown in Table 2, including the number of categories, training samples and testing samples.

4.2 Implementation Details

In the training process, the input raw images are resized to 512×512 and randomly cropped into 448×448. Besides, we also use random rotation and random horizontal flip for data augmentation. The local images proposed by SDR method are resized to 224×224. We train all models using Stochastic Gradient Descent (SGD) optimizer with momentum of 0.9, weight decay of 1e-5, and the batch size is set to 12 on one GTX 1080Ti GPU. The total epochs is set to 20 and 80 in retrieval and classification tasks, respectively. The initial learning rate is set to 0.0001 and 0.001 in retrieval and classification tasks, respectively, with exponential decay of 0.9 after every 2 epochs.

In our work, we adopt InceptionV3 as the base CNN feature extractor, which is pre-trained on ImageNet dataset [20]. We use the output of layerMixed as the feature maps. With the input image of size 448×448, the size of feature maps is 768×26×26. Without special instructions, the number of local regions \( n \) is set to 2, and the threshold \( \delta_h \) and \( \delta_l \) is are to 0.6 and 0.2, respectively. The attention \( 1 \times 1 \) convolutional layer is with channel size of \( M = 32 \). The additive angle \( m \) and the radius \( s \) of the embedding feature hypersphere are set to 0.3 and 64, respectively. The weight coefficients \( \alpha, \beta \) and \( \gamma \) are all set to \( \frac{1}{3} \).

4.3 Fine-Grained Image Retrieval

In this subsection, we show the performance of our approach in fine-grained image retrieval tasks. We use the fused embedding features processed by \( l_2 \) normalization for retrieval, and the \( l_2 \) distance is adopted to measure the similarity between features. The standard evaluation method \( R@K \) and Normalized Mutual Information (NMI) are adopted to evaluate the retrieval performance. \( R@K \) is the mean recall scores over all query images in the testing set. For each query, the top \( K \) similar images are returned by the retrieval method. If at least one of the returned images belongs to the same category as the query, the recall score will be 1, and 0 otherwise. NMI is the normalization of mutual information, which is adopted to evaluate the clustering performance of extracted embedding features in our work. When the value is 1, there is a perfect correlation between the predicted clustering result and the ground-truth clustering result, and when it is 0, there is no mutual information between the two clustering results.

4.3.1 Performance

Table 3 shows the performance comparison of our proposed approach and other state-of-the-art methods on CUB-200-2011 and Stanford-Cars testing datasets.

On CUB-200-2011 testing dataset, the baseline model achieves 73.5% and 70.5% accuracy on NMI and \( R@1 \) performances by only adopting attentional bilinear pooling method, which is a strong performance compared with other state-of-the-art methods. By searching and learning discriminative regions, under the 512-dimensional fused features, our proposed approach achieves 78.3% and 75.5% accuracy on NMI and \( R@1 \) performances, which outperforms the baseline model with 4.8% and 5.0%, respectively. Further, under 4096-dimensional fused features, our approach achieves 80.3% and 79.7% accuracy on NMI and \( R@1 \) performances. Compared to SCDA [7] which first adopted deep learning to solve FGIR tasks, our approach achieves 17.5% improvement on \( R@1 \) accuracy. Compared to ProxyNCA++ [10] which adopted proxy-based distance metric learning method in image retrieval tasks, our approach achieves 7.5% improvement on \( R@1 \) accuracy.

Similarly, on Stanford-Cars testing dataset, our ap-
To further understand our approach, we design ablation experiments to study the effects of various parts of the network on the final retrieval performance.

Table 4 shows the ablation experiments on the numbers of local discriminative regions. $n = 0$ means that the network does not search any local regions for learning the fused embedding features and only arm-am loss is adopted on the basis of the baseline. The experimental results show that if only the most discriminative local region is searched and learned, the network achieves a 2.9% improvement on $R@1$ accuracy, which shows that the local discriminative features are indeed beneficial to the performances of the network; If $n = 2$, the network improves the $R@1$ accuracy with 3.3% by learning the first two discriminative local regions and their merged region, which shows that more discriminative local regions could bring more information to the network. However, the performance of $n = 3$ shows that, under $\delta_l = 0.2$ and $\delta_h = 0.6$, two local regions are sufficient to search discriminative features in fine-grained image retrieval task.

Table 5 shows the ablation experiments on $\delta_l$ and $\delta_h$. In SDR, $\delta_l$ and $\delta_h$ are used to mark the background and foreground region, respectively. However, when $\delta_l$ is too large, it may cause SDR to erroneously drop potential discriminative regions, which would negatively affect retrieval performance. In the same way, a small $\delta_h$ makes the discriminative region located by SDR too large and introduces more background noise, while a large $\delta_h$ makes the discriminative region too small and insufficient information. Experimental results show that, compared to $\delta_l = 0$, the dropping background method is conducive to the model focusing on the discriminative regions in most cases, and under the condition of $n = 2$, $\delta_l = 0.2$ and $\delta_h = 0.6$, the proposed approach could achieve the best performances on fine-grained image retrieval task.

Table 6 shows the ablation experiments on the dropping strategy in SDR algorithm. w/o. Rec(·) means dropping the foreground region directly, rather than the smallest rectangular foreground region. Under w/o. Rec(·) strategy, Eq. (7) could be rewritten as $D = (1 - M_b) * (1 - M_f) * I$. Experimental results show that, the $R@1$ performance is reduced by 0.9%, which means that w. Rec(·) strategy could help SDR to locate more complete discriminative regions. w/o. $1 - M_b$ means not to drop the background region. Under w/o. $1 - M_b$ strategy, Eq. (7) could be rewritten as $D = (1 - rec(M_f)) * I$. Experimental results show that, the
Table 7 The retrieval performances (%) under different backbone on CUB-200-2011 testing dataset.

| Method              | NMI | R@1 | R@2 | R@4 | R@8 |
|---------------------|-----|-----|-----|-----|-----|
| Baseline-VGG16      | 69.0| 65.9| 76.1| 84.0| 89.7|
| Baseline-Resnet50   | 74.3| 71.4| 80.6| 87.5| 92.3|
| Baseline-InceptionV3| 73.8| 71.6| 80.9| 87.3| 91.9|
| Ours-VGG16          | 71.8| 69.4| 79.2| 86.5| 91.4|
| Ours-Resnet50       | 78.4| 75.0| 84.0| 90.2| 93.9|
| Ours-InceptionV3    | 77.8| 75.4| 83.6| 89.6| 93.7|

Table 8 The retrieval performances (%) under different training loss on CUB-200-2011 testing dataset.

| Network               | NMI | R@1 | R@2 | R@4 | R@8 |
|-----------------------|-----|-----|-----|-----|-----|
| Ours(softmax)         | 79.1| 76.5| 84.7| 90.6| 94.3|
| Ours(arcface)         | 80.0| 79.1| 86.3| 91.1| 94.8|
| Ours(arc-am)          | 80.3| 79.7| 86.9| 91.6| 94.7|

Fig. 4 Visualization of the fused embedding features captured by our baseline, Ours(softmax) and Ours(arc-am) models using t-SNE method.

R@1 performance is reduced by 0.2%, which means that dropping the background region strategy could help SDR locate a more accurate discriminative regions to a small extent. In addition, we also verified the impact of the drop loss \( L_{\text{drop}} \). Experimental results show that by constraining the network with \( L_{\text{drop}} \), the SDR algorithm could locate the next discriminative image more accurately.

Table 7 shows that the retrieval performances of our proposed approach with different backbones on CUB-200-2011 testing dataset. When the default batch size is 12 and the fused embedding size is 4096, our GPU’s memory is not enough to support us to replace the InceptionV3 backbone with Resnet50 directly. In order to compare different backbones fairly, we adjust the batch size and the fused embedding size to 8 and 512, respectively. Table 7 shows the ablation study on different backbone. Baseline-VGG16 means that the backbone of the baseline model is VGG16. Experimental results show that our proposed approach outperforms the corresponding baseline with at least 3.5% on R@1 performance, which proves that our SDR and LDR methods are robust on backbone models.

Table 8 shows the ablation experiments on the training loss. Compared to the traditional softmax loss, the arcface loss improves the final performance with 0.9% and 2.6% on NMI and R@1, respectively. Further, our adaptive margin arcface (arc-am) loss outperforms arcface loss with 0.3% and 0.6% on NMI and R@1, respectively.

To further understand our approach, we reduce the fused embedding features to 2D for visualization with t-SNE [23] method. As is shown in Fig. 4, compared to our baseline, the features extracted by our approach (softmax) are more compact within classes and the boundaries between classes are more obvious. Moreover, the arc-am loss further improves this situation.

4.4 Fine-Grained Image Classification

Table 9 shows the performances of our approach on fine-grained image classification tasks. The InceptionV3 with attentional bilinear pooling method is a strong baseline, which achieves 84.7%, 90.2% and 91.2% accuracy on CUB-200-2011, FGVC-Aircraft and Stanford-Cars, while our proposed approach outperforms it with a large margin of 4.2%, 4.1%, and 4.0% accuracy, respectively. Compared to MA-CNN [29] which also attempts to utilize attention mechanisms to find discriminative regions, our approach achieves 2.4%, 4.4%, and 2.4% accuracy improvement. Compared to NTS-Net [15] which proposes to utilize RPN to locate multi local images, our approach achieves 1.4%, 2.9%, and 1.3% accuracy improvement. Compared to WSDAN which proposes the baseline used and the data augmentation way to utilize local regions, our approach achieves −0.5%, 1.3%, and 0.7% accuracy improvement. It is worth noting that our proposed approach does not perform as well as WSDAN on the CUB-200-2011 dataset. According to our speculation, the reason for the poor performance of our proposed approach on CUB-200-2011 may be related to the characteristics of the dataset. As shown in Fig. 5, the background region of birds may provide information for classification. For example, images with sea in the background are generally sea birds, and those with woods are generally land birds. Therefore, attention maps may also have high-response value on the background, and our SDR and LDR methods are more susceptible to background noise in the CUB-200-2011 dataset, which leads to poor performance. However, WSDAN alleviates this problem to a certain extent through the method of using local images by data augmentation. Nevertheless, our proposed approach still achieves strong performances compared with other state-of-the-art methods on the three benchmark fine-grained datasets.

4.5 Localization

To intuitively understand our approach, we visualized the proposed local images in Fig. 5 under the condition of \( \delta_l = 0.2 \), \( \delta_h = 0.6 \) and \( n = 3 \). As shown as the visualization, SDR could effectively locate the local regions highly related to the object, thereby bringing more refined local features to the network, which is conducive to fine-grained image analysis tasks.

Table 10 show the mIoU with the ground-truth bounding boxes and classification performances under \( \delta_l = 0.2 \). mIoU is the mean of IoU, which is to evaluate the object localization performance of SDR. We determine the merged local region \( L_o \) as the detected object localization. It is shown that the larger \( \delta_h \) is, the more local regions are needed to cover the object, and the easier it is to obtain better lo-
Table 9  The classification performances comparison with state-of-the-art methods on the CUB-200-2011, FGVC-Aircraft and Stanford-Cars testing dataset. The best performance has been bolded.

| Network        | Backbone       | Auxiliary label | Bird(%) | Aircraft(%) | Car(%) |
|----------------|----------------|-----------------|---------|-------------|--------|
| BCNN [12]      | VGGNets        | ✗               | 84.1    | 84.1        | 91.3   |
| STN [24]       | InceptionV3    | ✗               | 84.1    | —           | —      |
| LRPB [25]      | VGG-16         | ✗               | 84.2    | 87.3        | 90.9   |
| FCAN [26]      | ResNet-50      | ✗               | 84.3    | —           | 91.5   |
| BoT [1]        | InceptionV3    | ✓               | 88.4    | 92.5        |        |
| RA-CNN [27]    | VGG-19         | ✗               | 85.3    | 88.4        | 92.5   |
| MAMC [28]      | ResNet-101     | ✗               | 86.5    | —           | 93.0   |
| MA-CNN [29]    | VGG-19         | ✗               | 86.5    | 89.9        | 92.8   |
| HBP [30]       | VGG-16         | ✗               | 87.1    | 90.3        | 93.7   |
| M-CNN [2]      | ResNet-50      | ✓               | 87.3    | —           | —      |
| DFL-CNN [31]   | ResNet-50      | ✗               | 87.4    | 92.0        | 93.8   |
| NTS-Net [15]   | ResNet-50      | ✗               | 87.5    | 91.4        | 93.9   |
| TASN [17]      | ResNet-50      | ✗               | 87.9    | —           | 93.8   |
| DF-GMM [32]    | ResNet-50      | ✗               | 88.8    | 93.8        | 94.8   |
| WSDAN [13]     | InceptionV3    | ✗               | 89.4    | 93.0        | 94.5   |
| Our baseline   | InceptionV3    | ✗               | 84.7    | 90.2        | 91.2   |
| Ours           | InceptionV3    | ✗               | 88.9    | 94.3        | 95.2   |

Table 10  The mIoU (%) and classification accuracy (%) performance under different δ₁ and n on CUB-200-2011 classification testing dataset.

| Method      | n = 1 | n = 2 | n = 3 |
|-------------|-------|-------|-------|
| δ₁ = 0.6   | (41.5, 85.5) | (55.3, 88.9) | (48.9, 88.3) |
| δ₁ = 0.7   | (31.8, 85.5) | (56.9, 88.9) | (53.1, 88.9) |
| δ₁ = 0.8   | (22.3, 85.4) | (51.4, 88.8) | (57.2, 88.8) |

calization performance. Under δ₁ = 0.6, the mIoU performance of n = 3 is lower than that of n = 2, which is consistent with the retrieval (in Table 4) and classification performances. However, when using a smaller δ₁, fewer n and fewer SDR iterations are required, which could greatly reduces the amount of calculation. In addition, experimental results show that the localization performance is positively correlated with the classification accuracy, but not always the better the localization performance, the higher the classification accuracy.

5. Conclusion

In this paper, we propose a novel approach for fine-grained image retrieval and classification tasks. The proposed approach could effectively locate multi discriminative local regions and learn discriminative representation from the raw features and local features. The experimental results on benchmark datasets show that, compared with other state-of-the-art methods, the proposed approach achieves excellent performance in both fine-grained image retrieval and classification tasks.

Acknowledgments

This work is supported by Shanghai Frontier Science Research Center for Gravitational Wave Detection.

References

[1] Y. Wang, J. Choi, V. Morariu, and L.S. Davis, “Mining discrim-
inative triplets of patches for fine-grained classification," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.1163–1172, 2016.

[2] X.-S. Wei, C.-W. Xie, J. Wu, and C. Shen, “Mask-cnn: Localizing parts and selecting descriptors for fine-grained bird species categorization,” Pattern Recognition, vol.76, pp.704–714, 2018.

[3] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, “Arcface: Additive angular margin loss for deep face recognition,” Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.4690–4699, 2019.

[4] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, “The caltech-ucsd birds-200-2011 dataset,” 2011.

[5] S. Maji, E. Rahru, J. Kannala, M. Blaschko, and A. Vedaldi, “Fine-grained visual classification of aircraft,” arXiv preprint arXiv:1306.5151, 2013.

[6] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, “3d object representations for fine-grained categorization,” Proc. IEEE international conference on computer vision workshops, pp.554–561, 2013.

[7] X.-S. Wei, J.-H. Luo, J. Wu, and Z.-H. Zhou, “Selective convolutional descriptor aggregation for fine-grained image retrieval,” IEEE Trans. Image Process., vol.26, no.6, pp.2868–2881, 2017.

[8] A. Zhai and H.-Y. Wu, “Classification is a strong baseline for deep metric learning,” arXiv preprint arXiv:1811.12649, 2018.

[9] X. Zheng, R. Ji, X. Sun, Y. Wu, F. Huang, and Y. Yang, “Centralized ranking loss with weakly supervised localization for fine-grained object retrieval,” Proc. 27th International Joint Conference on Artificial Intelligence, pp.1226–1233, 2018.

[10] E.W. Teh, T. DeVries, and G.W. Taylor, “Proxynca+: Revisiting and revitalizing proxy neighborhood component analysis,” Proc. European Conference on Computer Vision (ECCV), 2020.

[11] Y. Movshovitz-Attias, A. Toshev, T.K. Leung, S. Ioffe, and S. Singh, “No fuss distance metric learning using proxies.” Proc. IEEE International Conference on Computer Vision, pp.360–368, 2017.

[12] T.-Y. Lin, A. RoyChowdhury, and S. Maji, “Bilinear cnn models for fine-grained visual recognition,” Proc. IEEE international conference on computer vision, pp.1449–1457, 2015.

[13] T. Hu, H. Qi, Q. Huang, and Y. Lu, “See better before looking closer: Weakly supervised data augmentation network for fine-grained visual classification,” arXiv preprint arXiv:1901.09891, 2019.

[14] W. Ge, X. Lin, and Y. Yu, “Weakly supervised complementary parts models for fine-grained image classification from the bottom up,” Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.3034–3043, 2019.

[15] Z. Yang, T. Luo, D. Wang, Z. Hu, J. Gao, and L. Wang, “Learning to navigate for fine-grained classification,” ECCV, 2018.

[16] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” Advances in Neural Information Processing Systems, pp.91–99, 2015.

[17] H. Zheng, J. Fu, Z.-J. Zha, and J. Luo, “Looking for the devil in the details: Learning trilinear attention sampling network for fine-grained image recognition,” Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.5012–5021, 2019.

[18] K. Sun, L. Li, L. Li, N. He, and J. Zhu, “Spatial attentional bilinear 3d convolutional network for video-based autism spectrum disorder detection,” ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, pp.3387–3391, 2020.

[19] X. Zeng, Y. Zhang, X. Wang, K. Chen, D. Li, and W. Yang, “Fine-grained image retrieval via piecewise cross entropy loss,” Image and Vision Computing, vol.93, p.103820, 2020.

[20] J. Deng, W. Dong, R. Socher, L.-J. Li, and L. Fei-Fei, “ImageNet: A Large-Scale Hierarchical Image Database,” CVPR09, 2009.

[21] C.-Y. Wu, R. Manmatha, A.J. Smola, and P. Krähenbühl, “Sampling matters in deep embedding learning,” Proc. IEEE International Conference on Computer Vision, pp.2840–2848, 2017.

[22] X. Zheng, R. Ji, X. Sun, B. Zhang, Y. Wu, and F. Huang, “Towards optimal fine-grained retrieval via decorrelated centralized loss with normalize-scale layer,” Proc. AAAI Conference on Artificial Intelligence, vol.33, pp.9291–9298, 2019.

[23] L.v.d. Maaten and G. Hinton, “Visualizing data using t-sne,” Journal of Machine Learning Research, vol.9, no. Nov, pp.2579–2605, 2008.

[24] M. Jaderberg, K. Simonyan, A. Zisserman, et al., “Spatial transformer networks,” Advances in Neural Information Processing Systems, pp.2017–2025, 2015.

[25] S. Kong and C. Fowlkes, “Low-rank bilinear pooling for fine-grained classification,” Proc. IEEE conference on computer vision and pattern recognition, pp.365–374, 2017.

[26] M. Sun, Y. Yuan, F. Zhou, and E. Ding, “Multi-attention multi-class constraint for fine-grained image recognition,” Proc. European Conference on Computer Vision (ECCV), pp.805–821, 2018.

[27] H. Zheng, J. Fu, T. Mei, and J. Luo, “Learning multi-attention convolutional neural network for fine-grained image recognition,” Proc. IEEE international conference on computer vision, pp.5209–5217, 2017.

[28] Y. Wang, V.I. Morariu, and L.S. Davis, “Learning a discriminative filter bank within a cnn for fine-grained recognition.” Proc. IEEE conference on computer vision and pattern recognition, pp.4148–4157, 2018.

[29] Z. Wang, S. Wang, S. Yang, H. Li, J. Li, and Z. Li, “Weakly supervised fine-grained image classification via gaussian mixture model oriented discriminative learning,” Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020.