IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 13, 2020, accepted September 30, 2020, date of publication October 12, 2020,
date of current version October 22, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3030249

Exploiting Category Similarity-Based Distributed Labeling for Fine-Grained Visual Classification

PENGZHEN DU, ZEREN SUN, YAZHOU YAO, (Member, IEEE), AND ZHENMIN TANG
School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China

Corresponding author: Yazhou Yao (yazhou.yao@njust.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61976116, and in part by the Fundamental Research Funds for the Central Universities under Grant 30920021135.

ABSTRACT The fine-grained visual classification (FGVC) which aims to distinguish subtle differences among subcategories is an important computer vision task. However, one issue that limits model performance is the problem of diversity within subcategories. To this end, we propose a simple yet effective approach named category similarity-based distributed labeling (CSDL) to tackle this problem. Specifically, we first obtain the feature centers for various subcategories and utilize them to initialize the label distributions. Then we replace the ground-truth labels in a Deep Neural Network (DNN) with the distributed labels to calculate the loss and perform the optimization. Finally, the joint supervision of a softmax loss and a center loss is adopted to update the parameters of the DNN, the deep feature centers, and the distributed labels for learning discriminative deep features. Comprehensive experiments on three publicly available FGVC datasets demonstrate the superiority of our proposed approach.

INDEX TERMS Fine-grained classification, label distributions, category similarity, distributed labels.

I. INTRODUCTION

Distinguishing subtle differences among fine-grained categories (e.g., different kinds of birds [3], aircrafts [9], or cars [10]) is an extremely difficult computer vision task. Humans can easily distinguish a dog from a cat as the two are significantly different in appearance. However, it is challenging to identify subtle differences among fine-grained subcategories, even for an expert with specific knowledge. This is because subcategories are visually similar to each other. For example, both "Caspian Tern" and "Artic Tern" have a white head with a black cap, a white neck, and gray wings. These subcategories are thus difficult to distinguish for a non-expert because they share a similar global appearance and can only be differentiated by subtle differences in small regions. In view of this, the distinction between FGVC and traditional visual classification (e.g., ImageNet [60] categorization) lies in two aspects: (i) subcategories are visually similar and harder to distinguish, and (ii) there are fewer training samples for FGVC and therefore the training set might not be representative of the practical scenario.

The associate editor coordinating the review of this manuscript and approving it for publication was Li Zhang.

Existing FGVC works can be roughly divided into three categories [14]. The first category comprises strongly supervised methods which requires manually labeled bounding boxes or part annotations [15]–[19], [21]. However, labeling objects at the subcategory level tends to require expert knowledge, which greatly limits the feasibility of using strongly supervised FGVC algorithms for real-world applications. The second category is webly supervised methods, which trains FGVC models by directly leveraging web images [36], [37]. The advantage of these methods is that they eliminate the human cost. Nevertheless, due to the influence of errorprone automatic or non-expert annotations, web images are usually associated with noisy labels, and thus performances of webly supervised approaches are far from satisfying. The third category is weakly supervised methods, which employs only image-level labels [20], [22]–[26]. Compared to strongly supervised methods, the cost is greatly reduced. Compared to webly supervised methods, the performances of weakly supervised approaches are considerably better. In addition, these methods don't require a large number of training samples, making them the prevailing trend for the FGVC task.

The cross-entropy loss is the most commonly used loss function in visual classification, especially in FGVC. By using cross-entropy loss in a deep CNN, we can interpret the

softmax result of the last fully connected layer as a probability distribution for the input image. The typical strategy of calculating the cross-entropy loss is to use the groundtruth label to form a one-hot label distribution. By doing this, the deep CNN is forced to evolve into a model whose final probability for the input image belonging to the ground-truth category is close to 1. While this strategy works nicely for coarse-grained image classification, problems occur when it is leveraged for the FGVC task. Coarse-grained classification benefits from this strategy since the model can be pretty confident in distinguishing basic-level categories (e.g., dogs, and cats in ImageNet). However, even an expert would not be able to always identify the specific subordinate categories in the FGVC task. By using the cross-entropy loss, the deep models will more easily over-fit since the objects are very difficult to distinguish, and the models may start to focus on irrelevant features (e.g., background content).

To cope with this limitation, the maximum-entropy loss was adopted by [32] for FGVC. The motivation behind this work came from the fact that, if a certain species of bird tends to be photographed against a specific background, memorizing the background will deteriorate the generalization performance since the CNN will associate the background with the bird itself. By replacing the cross-entropy loss with maximum-entropy loss, the deep CNN model can degrade the confidence and result in better generalization abilities in the FGVC task. Nevertheless, [32] still adopted the one-hot label distribution, along with its inherent disadvantages.

In this work, we propose an approach named category similarity-based distributed labeling to learn and capitalize on discriminative feature representations in an end-to-end fashion. Our main idea is to take the visual affinity among fine-grained categories into account and learn training images less confidently. Specifically, our approach is realized by a center loss module and a distributed labeling module. The center loss module calculates a center loss [38] to jointly optimize the network with the conventional cross-entropy loss for generating category center embeddings. The distributed labeling module leverages the produced category center embeddings to formulate distributed labels. The coupling of the center loss module and distributed labeling module enables our approach to learn powerful representations in a mutually reinforcing way, leading to superior performance in finegrained visual classification. Moreover, an adaptive weighting schema is adopted to combine the distributed label and ground-truth label for guaranteeing proper ground-truth guidance while alleviating the over-fitting. Extensive experiments on CUB200-2011 [3], FGVC Aircrafts [9], and Stanford Cars [10] demonstrate the superiority of our proposed approach. The contributions of our work can be summarized as follows:

- We propose a novel mechanism that regularizes FGVC by assigning distributed labels. Meanwhile, the distributed labels are dynamically adjusted by calculating the feature embedding centers.
- We propose to jointly leverage center loss as well as cross-entropy loss to take inter-class and intra-classdiversity into consideration for learning more discriminative representations.
- Our proposed CSDL method can be easily integrated with existing fine-grained classification approaches. Our experiments show that the combination of CSDL and existing methods can achieve state-of-the-art performance.

The rest of this paper is organized as follows: the related work is described in Section I and our approach is introduced in Section II; we then report our evaluations on three publicly available fine-grained datasets in Section III and finally conclude our work in Section V.

II. RELATED WORK

A. FINE-GRAINED VISUAL CLASSIFICATION

Fine-grained visual classification essentially focuses on representing visual differences between subcategories [48], [49]. The vast majority of researchers follow either a localization-classification manner or an end-to-end encoding fashion. Based on the extent of supervision, existing works can be roughly organized into three categories.

The first one is primarily strongly supervised methods [15]–[19]. Methods of this type adopt the localization-classification pipeline and typically require manually labeled object bounding boxes or part annotations, besides image labels. Using these dense manual annotations, they manage to localize key parts by directly learning a key-parts detector [15], [17] or leveraging semantic segmentation methods [19]. After key parts are detected and localized, part features are integrated as the final visual representation for fine-grained classification. However, the practicality and scalability of these methods are limited due to the demand for time-consuming and labor-intensive manual annotation.

The second group is weakly supervised approaches. Different from strongly supervised methods, weakly supervised methods cease to use bounding boxes and part annotations. Instead, methods in this group only require image-level labels during training [20]–[32], [39], [40]. Some of them also follow a part-based pipeline but in a weakly supervised manner. For example, Zheng *et al.* [39] proposed a trilinear attention sampling network to learn fine-grained features from part proposals in an efficient teacher-student manner. Chen *et al.* [40] proposed a destruction and construction learning framework, in which input images are ''destructed'' and then ''reconstructed'' for learning more discriminative details. Others adopt an end-to-end training strategy (*e.g.*, [26], [32]). Due to the fact that methods of this group considerably reduce the cost of annotation, weakly supervised methods are becoming the prevailing trend for fine-grained tasks.

The third set is semi-supervised methods. These methods leverage web images in training fine-grained classification models. [35]–[37], [41], [42]. Although data is

augmented without adding any manual labeling overhead, these approaches still involve a certain level of human intervention. Moreover, due to the inevitable label noise issue, performances of these approaches are far from satisfying.

B. DISTRIBUTED LABELING

The idea of distributed labeling originates from the ''Pseudo Label'' that is commonly used in semi-supervised learning. Lee *et al.* proposed one-hot pseudo labeling as an efficient approach for semi-supervised deep neural network learning [43]. He trained networks in a supervised paradigm with both labeled and unlabeled data. To be specific, the model picks the class with the maximum predicted probability as the true label for unlabeled data. Zheng *et al.* further extended the one-hot pseudo-label scheme and proposed a uniformly distributed pseudo label [44]. Huang *et al.* proposed MpRL to assign pseudo labels to unlabeled data according to different contributions [45]. Ding *et al.* proposed the affinity-based pseudo-labeling method [46] to effectively leverage unlabeled data generated by Generative Adversarial Networks (GANs) [50]. However, these methods predominantly focus on semi-supervised learning (pseudo labels are only adopted on unlabeled data or synthetic data). We are inspired by this idea and propose our distributed labeling to regularize FGVC networks without any additional data.

C. LABEL SMOOTHING REGULATION

Our approach is motivated by label smoothing regularization (LSR) [51]. The commonality between LSR and our approach is that both encourage the model to be less confident. The differences between LSR and our approach lie in two aspects. Firstly, LSR assigns a small probability value to the ground-truth label and adopts a uniform distribution over all non-ground-truth categories, while our approach takes the similarities among categories into account. In other words, LSR treats all non-ground-truth categories equally, while our proposed approach considers the fact that some categories share more similarities with the ground-truth category than other categories. Secondly, to update the feature centers (used for computing similarities among categories) during training and encourage the model to learn more discriminative features, we adopt a center loss along with the conventional cross-entropy loss to jointly supervise the optimization of the neural networks.

III. THE PROPOSED APPROACH

The activation vector in deep convolutional neural networks produced by the classifier is normalized by softmax to a pseudo-probability vector:

$$p_j(x) = \frac{e^{W_j^T x + b_j}}{\sum_{k=1}^{N} e^{W_k^T x + b_k}}, \tag{1}$$

where W and b are the weight and bias parameter of the classifier, respectively. The pseudo-probability vector in conventional CNN-based image classification approaches is treated as the predicted probability distribution of the input image with respect to all categories in computing the cross-entropy loss. For simplicity, we call this pseudo-probability as the prediction probability hereafter. Specifically, the cross-entropy loss is defined as

$$L_{CE} = -\sum_i q_i(x) \log(p_i(x)). \tag{2}$$

$p_i(x)$ and $q_i(x)$ are the prediction probability and ground-truth probability, respectively. The most commonly used ground-truth label distribution is one-hot distribution. Specifically, if the input image x belongs to the k-th category, then we have

$$q_i = \begin{cases} 1, & i = k \\ 0, & i \neq k. \end{cases} \tag{3}$$

The one-hot label distribution is perfect for coarse-grained visual classification because visual differences among coarse categories are significant so that ''peaky'' prediction distributions (i.e. over-confident predictions) have little influence on the model's generalization ability [47]. However, compared to conventional coarse-grained categories, fine-grained categories are more visually similar. When using one-hot label distribution in a fine-grained visual classification task, the network predominantly focuses on increasing the confidence in prediction, while neglecting the fact that the visual alikeness among fine-grained categories makes it impossible to have nearly 100% prediction confidence. The network would start to focus on irrelevant features or sample-specific artifacts (e.g., background content) in order to achieve higher prediction confidence. Thus, using one-hot label distributions is more likely to cause overfitting in fine-grained classification tasks.

Therefore, instead of expecting an over-confident prediction (*i.e.*, a prediction probability vector with $p_k = 1$ and $p_i = 0$ $(i \neq k)$), we propose to train a less confident model to mitigate the overfitting issue caused by similarities among fine-grained subcategories. To be specific, we expect the model to produce less certain prediction probability in which $0 < p_k < 1$, while $0 \leq p_i < p_k$ $(i \neq k)$. The intuition behind this is that, when classifying a fine-grained image x, we are not entirely confident in its prediction result, as it may share too many features with other subcategory images.

To this end, we propose a simple yet effective weakly supervised approach, namely category similarity-based distributed labeling (CSDL), whose main idea is to (1) adopt the center loss to promote feature compactness and obtain class centers; (2) perform distributed labeling based on the feature similarity between class centers to mitigate over-confident predictions; (3) dynamically update the distributed labels throughout the whole training process. The framework of our proposed approach is presented in Fig. 1.

A. CENTER LOSS MODULE

To enhance CNN's feature representations and obtain as much inter-class diversity and intra-class compactness as

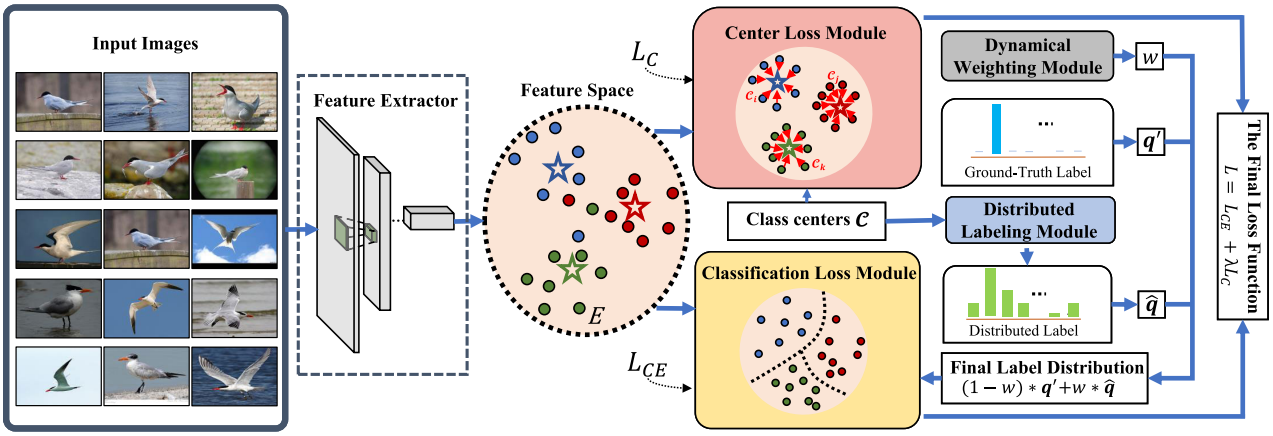

FIGURE 1. The architecture of our category similarity-based distributed labeling model. The feature vector E is extracted by a backbone network (*e.g.*, VGGNet16) and then sent into two parallel modules. The center loss module calculates the center loss between feature vector E and the class center C. The classification loss module computes the classification loss (*e.g.*, cross-entropy) using E, as well as the final label distribution. Specifically, the final label distribution is obtained by calculating the weighted sum of the ground-truth label and the distributed label (generated from C). The loss function for our proposed model is a weighted sum of the classification loss and center loss.

possible, we adopt the center loss [38] to supervise the model training in conjunction with the conventional cross-entropy loss. The center loss function is defined as:

$$L_C = \frac{1}{2}\sum_i ||x_i - c_{y_i}||_2^2, \tag{4}$$

where x_i denotes the deep feature representation of the i-th training sample that belongs to the y_i-th category. c_{y_i} represents the deep feature center of the y_i-th category. Following [38], class centers $\{c_j\}$ are dynamically updated via:

$$c_j^{T+1} = c_j^T - \alpha \cdot \Delta c_j^T, \tag{5}$$

in which α is the center updating rate used to avoid large perturbations caused by a few mislabeled samples. Different from the update equation in [38], we propose a weighted update mechanism to adjust class centers. More precisely, our update equation of Δc_j is defined as:

$$\Delta c_j = \frac{\sum_{i=1}^m \delta(y_i = j) \cdot (c_j - \beta_i \cdot x_i)}{1 + \sum_{i=1}^m \delta(y_i = j)}. \tag{6}$$

m denotes the total number of training samples and the weight parameter β_i is designed as the maximum predicted probability of sample x_i:

$$\beta_i = \max p(x_i). \tag{7}$$

The center loss module is adopted for the following three purposes: (1) enhance the intra-class representation compactness and keep features of different classes distinguishable; (2) acquire class centers for later distributed labeling; (3) update class centers based on each mini-batch instead of the entire training set, so that the model is trainable and easy to optimize in CNNs. Furthermore, compared to the original center loss [38], we encourage the center update process to favor samples with higher prediction confidence by introducing our weighting mechanism. Intuitively, samples with lower confidence are more likely to approach the classification boundary, thus their representations are more likely to consist of features that are confusing or hard to distinguish. Resorting to our weighting design, more confident samples are emphasized so that the acquired class centers are more representative and discriminative.

B. DISTRIBUTED LABELING MODULE

The motivation behind leveraging distributed labeling comes from the label smoothing regularization (LSR) technique [51]. The smoothed label distribution is formulated as:

$$q_i' = \begin{cases} 1-\epsilon, & i = k \\ \epsilon/(N-1), & i \neq k \end{cases} \tag{8}$$

where ϵ is the smoothing parameter and N is the number of categories. LSR can be deemed as a type of distributed labeling method, which reduces the prediction confidence score by assigning equal probabilities to non-target classes. Different from the LSR way of generating distributed labels, we take the category similarity in the feature space into account. Therefore, the class label of input sample x is not only decided by the ground-truth label but also contributed to by the category similarity. In our work, we adopt the following cosine similarity function as the default measurement for the category similarity:

$$S_{\text{cosine}}(c_i, c_j) = \frac{c_i \cdot c_j}{||c_i||||c_j||}. \tag{9}$$

c_i and c_j denote the deep feature center for the i-th and j-th category, respectively. In the ablation study section, we further compare the cosine similarity function with another commonly used one (*i.e.*, euclidean based similarity).

By calculating the similarity between the k-th category center c_k and all category centers, we can obtain a similarity

vector V, whose each element V_i is

$$V_i = \frac{c_i \cdot c_k}{||c_i||||c_k||} = \begin{cases} 1, & i = k \\ v, & i \neq k \end{cases} \tag{10}$$

in which $v \in [0, 1)$. Subsequently, a softmax function is applied to normalize V to the category similarity-based label distribution of the k-th category as follows

$$\hat{q}_i = \frac{e^{V_i}}{\sum_{j=1}^{N} e^{V_j}}. \tag{11}$$

To make the most of the prior knowledge of ground-truth labels and endow our model with correct guidance during the model optimization process, the final label distribution is designed as a weighted sum of the category similarity-based label distribution and the smoothed ground-truth label distribution:

$$\tilde{q}_i = (1 - w) * q_i' + w * \hat{q}_i, \tag{12}$$

where w is the hyperparameter that controls the tradeoff between the category similarity-based distributed label and the ground-truth label during the model learning procedure. It should be noted that, when the smoothing parameter ϵ is 0, the smoothed label is equivalent to the one-hot ground-truth label.

C. DYNAMICAL WEIGHTING MODULE

In Eq. (12), the weighted sum of the category similarity-based label distribution and smoothed ground-truth label distribution are leveraged to supervise the training process. The hyperparameter w is used to balance the two distributions. For obtaining further performance gains, we propose a dynamical weighting module to dynamically adjust w for better regularizing the model. To be more precise, we define the hyperparameter w as follows:

$$w(T) = \begin{cases} w_{init}, & T = 0 \\ w_{init} + \dfrac{w_{end} - w_{init}}{T_k} T, & 0 < T \leq T_k \\ w_{end}, & T > T_k \end{cases} \tag{13}$$

The "memorization" effect of deep CNNs demonstrates that deep neural networks tend to learn simple patterns in initial epochs, resulting in limited classifying capability in the early stage. With the increase of training epochs, the model will be empowered with an increasingly robust discriminating capability. Therefore, we set $w_{init} > w_{end}$ in Eq. (13), so that we emphasize the category similarity more in early epochs and then gradually let the ground-truth label dominate the model training. In the early epochs, the model has a limited classification capability. Resorting to our dynamical weighting module, a higher w is adopted to emphasize the category similarity and depreciate the ground-truth label. Accordingly, the prediction confidence score is reduced. The pace of the model fitting to the training set is subsequently slowed down and therefore the model is driven to learn more representative features. As the training proceeds, w gradually decreases so that the ground-truth label will progressively dominate the learning process and thus result in better classification performance with stronger ground-truth guidance. We investigate both fixed w and dynamic $w(T)$ in the ablation study section.

D. CATEGORY SIMILARITY-BASED DISTRIBUTED LABELING

To summarize, we focus on the fine-grained visual tasks and propose our category similarity-based distributed labeling. To be more specific, we first extract features by feeding training samples into a backbone network. Then class centers are acquired via the process of optimizing the center loss. Subsequently, distributed labels for all training samples are updated based on class centers. Afterward, we compute the classification loss by using distributed labels to calculate the cross-entropy.

$$L_{CE} = -\frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{N} \tilde{q}_j(x_i) \log p_j(x_i). \tag{14}$$

Finally, we train the model by adopting joint supervision of the center loss and classification loss.

$$\begin{aligned} L &= L_{CE} + \lambda L_C \\ &= -\frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{N} \tilde{q}_j(x_i) \log p_j(x_i) \\ &\quad + \lambda \cdot \frac{1}{2} \sum_{i} ||x_i - c_{y_i}||_2^2 \end{aligned} \tag{15}$$

where λ is a tradeoff parameter to balance the effect of the cross-entropy loss and the center loss.

E. ADVANTAGES

Our proposed approach provides three major innovations:

1) We propose to leverage the center loss and cross-entropy loss to jointly supervise the deep features learning. Compared with solely using the cross-entropy loss, our proposed approach can drive the learned deep feature representations closer and thus encourage the model to learn more representative features. Meanwhile, class centers are obtained in the process of optimizing the center loss for later distributed labeling. Moreover, we introduce a weighted update mechanism to update class centers for making them more representative by favoring high confidence samples.

2) We propose to use additional knowledge (category similarity) for generating distributed labels instead of only utilizing one-hot ground-truth labels for model optimization, resulting in a more robust model.

3) We propose a dynamical weighting module to dynamically adjust w in Eq. (12) from large values to small ones over the course of the training process. Resorting to this module, the model learns more slowly in the beginning to build a solid foundation. As the training continues, the value of w decreases, and the ground-truth label will progressively dominate the learning process and result in better classification performance with stronger guidance.

IV. EXPERIMENTS

A. DATASETS AND EVALUATION METRICS

In this section, three most commonly used fine-grained benchmark datasets (*i.e.*, CUB200-2011 [3], FGVC Aircraft [9], and Stanford Cars [10]) are employed to evaluate the effectiveness of our proposed CSDL method.

1) DATASETS

- **CUB200-2011** [3] is the most widely used fine-grained classification dataset, which is designed for classification within 200 species of birds. It contains 11788 images in total, including 5994 images in the training set and 5794 images in the test set. Each image is annotated with one image-level subcategory label, one object bounding box, 15 part locations, and 312 binary attributes. In our experiment setting, only image-level subcategory labels are leveraged in the training procedure.
- **FGVC Aircraft** [9] contains 10000 images of 100 different aircraft model variants. Conventionally, we divide this dataset as follows: 6667 images in the training set and 3333 images in the test set. The aircraft in every image is labeled with a tight bounding box and a hierarchical airplane model label. We also only adopt image-level subcategory labels in our training process.
- **Stanford Cars** [10] is another widely used fine-grained classification dataset, which consists of 16185 images of 196 classes of cars. The data is split into 8144 training images and 8041 testing images, where each class has been split roughly in a 50-50 split. Each image is labeled with an image-level subcategory label and an object bounding box. The same with the aforementioned two datasets, we only use the image-level subcategory labels in our experiments.

2) EVALUATION METRIC

We adopt the average classification accuracy (ACA) as the evaluation metric to assess the classification performance of our proposed CSDL approach. Each experiment is repeated five times and their average is reported eventually.

B. IMPLEMENTATION DETAILS

We leverage three standard deep CNN frameworks: VGG16 [57], GoogleNet [58], and ResNet50 [59]. Specifically, all of these frameworks are pre-trained on ImageNet [60] and we replace the 1000-dimensional softmax layer with a σ-dimensional one. σ is the number of categories in the fine-grained datasets (*e.g.*, 200, 100 and 196 for CUB200-2011, FGVC Aircraft and Stanford Cars, respectively). For the center loss, we follow the settings in [38] and set λ to 0.003 and α to 0.5. We initialize the feature centers $\{c_i\}$ with features extracted from the pre-trained model. During training, we first resize training samples to ensure the shortest edges are 448 while keeping the aspect ratio of the images. We then apply a random horizontal flipping to these resized training samples. Finally, we crop training samples into 448×448. We fine-tune the pre-trained models through Stochastic Gradient Descent (SGD) by setting momentum to 0.9. The learning rate, batch size, weight decay, and epochs are set to 0.01, 64, 0.0001, and 120, respectively. The learning rate is halved every 10 epochs. Our experiments are conducted on one NVIDIA V100 GPU card and the implementation is based on PyTorch.

C. BASELINES

We compare our proposed CSDL with Cross-Entropy [57]–[59] and Max-Entropy [32] by fine-tuning VGG16, GoogLeNet, and ResNet50 on the CUB200-2011, FGVC Aircraft, and Stanford Cars datasets. Performances of baseline methods are directly from Maximum-Entropy [32]. We also compared our proposed CSDL with state-of-the-art weakly supervised fine-grained methods, including BCNN [26], Kernel Pooling [30], MA-CNN [20], NTS-Net [24], DFL-CNN [25], iSQRT-COV [31], Part Model [56], TASN [39], DCL [40]. Performances of existing methods are all taken directly from their original papers.

TABLE 1. Comparison with state-of-the-art weakly supervised methods on three benchmark datasets.

Methods	CUB200	Aircraft	Stanford Cars
BCNN	84.1	84.5	91.3
Kernel Pooling	86.2	86.9	92.4
MA-CNN	86.5	89.9	92.8
NTS-Net	87.5	91.4	93.9
DFL-CNN	87.4	92.0	93.8
iSQRT-COV	88.1	90.0	92.8
Part Model	**90.4**	-	-
TASN	89.1	-	93.8
DCL	87.8	93.0	94.5
CSDL-MaxEnt	86.3	88.1	92.9
CSDL-iSQRT-COV	88.7	93.2	94.7
CSDL-DCL	88.6	**93.5**	**94.9**

D. EVALUATIONS

1) COMPARISON WITH SOTA METHODS

As stated above, our CSDL method can be integrated with existing fine-grained methods and achieve state-of-the-art performance. Table 1 shows the performance when integrating our CSDL with the existing fine-grained method iSQRT-COV [31]. As shown in Table 1, compared with state-of-the-art weakly supervised methods, our approach achieves the best performance on FGVC-Aircraft and Stanford Cars dataset. Although its performance on the CUB200-2011 dataset is not the optimal, the result is fairly comparable with other state-of-the-art methods. This is due to our leveraging of the center loss and cross-entropy loss to jointly supervise the deep feature learning. Compared with solely using cross-entropy or max-entropy loss, our approach can drive the learned deep feature representations closer and thus encourage the model to learn more representative features. Besides, we utilize additional knowledge (category similarity) to

TABLE 2. (A-C) present the performances of three fine-tuning loss functions (Cross-Entropy, Max-Entropy, and CSDL) on different datasets. Each loss function is combined with three different backbone networks (VGGNet16, GoogLeNet, and ResNet50). Improvement over the baseline model is reported as (Δ). (D) presents the average performance over three models (i.e. VGGNet16, GoogLeNet, and ResNet50) on different datasets. (E) presents the average performance over three fine-grained datasets (i.e. CUB200-2011, FGVC Aircraft, and Stanford Cars) on different modes. (A) CUB200-2011. (B) FGVC Aircraft. (C) Stanford Cars. (D) Different Datasets. (E) Different Models.

(A) CUB200-2011

Method	Top-1	Δ
Our Results		
CroEnt-VGGNet16	73.28	-
MaxEnt-VGGNet16	77.02	3.74
CSDL-VGGNet16	**82.31**	**9.03**
CroEnt-GoogLeNet	68.19	-
MaxEnt-GoogLeNet	74.37	6.18
CSDL-GoogLeNet	**81.60**	**13.41**
CroEnt-ResNet50	75.15	-
MaxEnt-ResNet50	80.37	5.22
CSDL-ResNet50	**86.11**	**10.96**

(B) FGVC Aircraft

Method	Top-1	Δ
Our Results		
CroEnt-VGGNet16	74.17	-
MaxEnt-VGGNet16	78.08	3.91
CSDL-VGGNet16	**85.75**	**11.58**
CroEnt-GoogLeNet	74.04	-
MaxEnt-GoogLeNet	79.16	5.12
CSDL-GoogLeNet	**81.61**	**7.57**
CroEnt-ResNet50	81.19	-
MaxEnt-ResNet50	83.86	2.67
CSDL-ResNet50	**85.96**	**4.77**

(C) Stanford Cars

Method	Top-1	Δ
Our Results		
CroEnt-VGGNet16	80.60	-
MaxEnt-VGGNet16	83.88	3.28
CSDL-VGGNet16	**88.11**	**7.51**
CroEnt-GoogLeNet	84.85	-
MaxEnt-GoogLeNet	87.02	2.17
CSDL-GoogLeNet	**87.99**	**3.14**
CroEnt-ResNet50	91.52	-
MaxEnt-ResNet50	**93.85**	**2.33**
CSDL-ResNet50	92.00	0.48

(D) Different Datasets

Dataset	Loss	Average	Δ
CUB200-2011	Cross-Entropy	72.21	-
	Maximum-Entropy	77.25	5.04
	CSDL	**83.30**	**11.09**
FGVC-Aircraft	Cross-Entropy	76.47	-
	Maximum-Entropy	80.37	3.90
	CSDL	**84.44**	**7.97**
Stanford Cars	Cross-Entropy	85.66	-
	Maximum-Entropy	88.25	2.59
	CSDL	**89.37**	**3.71**

(E) Different Models

Model	Loss	Average	Δ
VGGNet16	Cross-Entropy	76.02	-
	Maximum-Entropy	79.66	3.64
	CSDL	**85.39**	**9.37**
GoogLeNet	Cross-Entropy	75.69	-
	Maximum-Entropy	80.18	4.49
	CSDL	**83.73**	**8.04**
ResNet50	Cross-Entropy	82.62	-
	Maximum-Entropy	86.03	3.41
	CSDL	**88.02**	**5.4**

dynamically generate distributed labels, rather than only using one-hot ground-truth labels for model optimization.

2) COMPARISON WITH DIFFERENT LOSS FUNCTIONS

Table 2 presents the performance comparison using different loss functions. From Table 2, we find that, with the exception of the ResNet50 model on Stanford Cars, our proposed CSDL approach achieves a significantly higher performance than the other two loss functions across all three datasets, for all three backbones.

V. ABLATION STUDIES

A. EFFECTIVENESS ON REDUCING PREDICTION CONFIDENCE

Our main idea of CSDL is to introduce the category similarity to make model predictions less confident, so that the overfitting caused by overconfidence can be mitigated. The smoothness of the label distribution generated by CSDL guarantees the predicted logit vector to be smoother, benefiting generalization performance. We use the backbone network VGGNet16 on the CUB200-2011 dataset as an example. Fig. 2 presents four images from CUB200-2011 and compares the prediction confidence of Cross-Entropy (denoted as CroEnt) and CSDL-Cross-Entropy (denoted as CSDL). From Fig. 2, we can see that, by adopting our CSDL, the prediction confidence is significantly reduced and the prediction logit vector is considerably smoother.

TABLE 3. Comparison between LSR and our CSDL. Compared to LSR, our CSDL obtains a larger performance boost in fine-grained datasets (except using ResNet50 on Stanford Cars).

Backbone	Method	CUB200	Aircraft	Cars
VGGNet16	LSR	70.03	75.06	81.45
	CSDL	**82.31**	**85.75**	**88.11**
GoogLeNet	LSR	75.61	78.55	84.27
	CSDL	**81.60**	**81.61**	**87.99**
ResNet50	LSR	78.20	81.26	92.04
	CSDL	**86.11**	**85.96**	92.00

B. COMPARISON WITH LABEL SMOOTHING REGULARIZATION

Our approach is inspired by label smoothing regularization (LSR) [51]. Similarly, LSR and our method both encourage the model to be less confident to avoid peaky predictions. We compare the performance of LSR and our CSDL in Table 3 and observe that our proposed CSDL outperforms label smoothing on fine-grained tasks. It should be noted that the results of LSR in Table 3 did not involve center loss. We have also conducted further experiments to explore the case of LSR using center loss. When using VGGNet16 as the backbone, results of LSR using center loss on CUB200-2011, FGVC Aircraft, and Stanford Cars are 79.51%, 82.99%, and 85.24%, respectively. From the experimental results, we can notice that our approach works better than LSR.

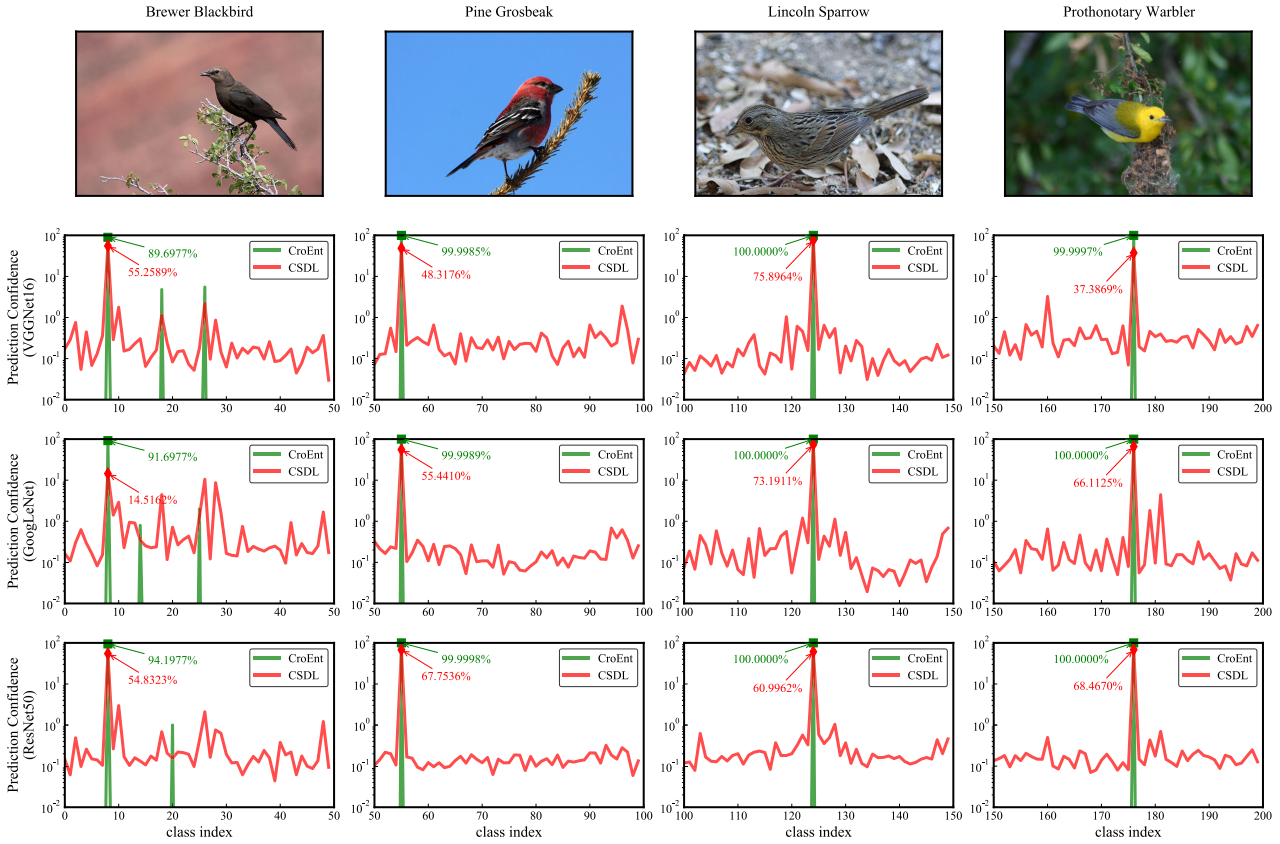

FIGURE 2. The upper four images are samples from four categories in CUB200-2011 (class index: 8, 55, 124, and 176). The bottom four graphs show comparisons of prediction confidence score between CroEnt and CSDL (for ease of presentation, we clip the x-axis to display only 50 categories).

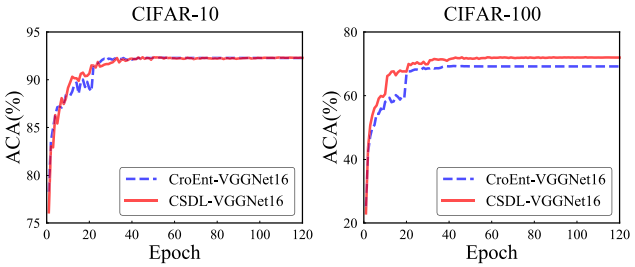

FIGURE 3. Performance comparisons between CroEnt-VGG16Net and CSDL-VGG16Net on CIFAR-10 (a) and CIFAR-100 (b). Under identical experimental settings, the performance improvement on the CIFAR-100 dataset is much higher than on CIFAR-10.

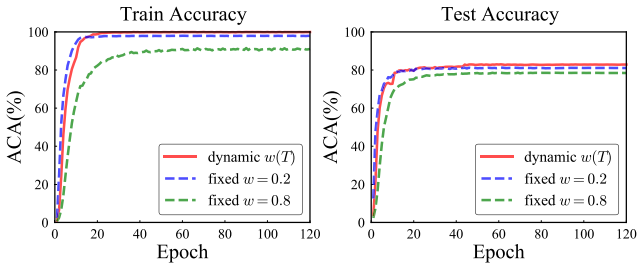

FIGURE 4. Comparison between CSDL-VGGNet16 (using fixed w) and CSDL-VGGNet16 (using dynamic $w(T)$ with $w_{init} = 0.7$, $w_{end} = 0.1$, $T_k = 10$).

C. CIFAR-10 AND CIFAR-100

In order to demonstrate that our proposed CSDL capitalizes on the nature of fine-grained datasets, we investigate the performance of our CSDL on two coarse-grained datasets (*i.e.*, CIFAR-10 and CIFAR-100 [2]). The 10 categories in CIFAR-10 dataset are completely different from each other On the contrary, categories in CIFAR-100 can be grouped into 20 super-categories and each super-category consists of five finer divisions. Given these two datasets, categorizing CIFAR-10 can be viewed as a coarser task while classifying CIFAR-100 can be deemed as a relatively finer task. Experimental results are shown in Fig. 3. As shown in Fig, 3, a larger performance improvement can be observed on CIFAR-100 compared to CIFAR-10, demonstrating that our proposed CSDL works better in the finer-grained task. While lower prediction confidence resulted from CSDL benefits the generalization ability in fine-grained tasks, coarse-grained tasks gain no advantage from CSDL because categories in coarse-grained datasets share few visual similarities between each other.

D. INFLUENCE OF DIFFERENT SIMILARITY FUNCTIONS

The cosine similarity and euclidean distance-based similarity are two commonly used similarity measurement. To investigate the influence of adopting different similarity functions,

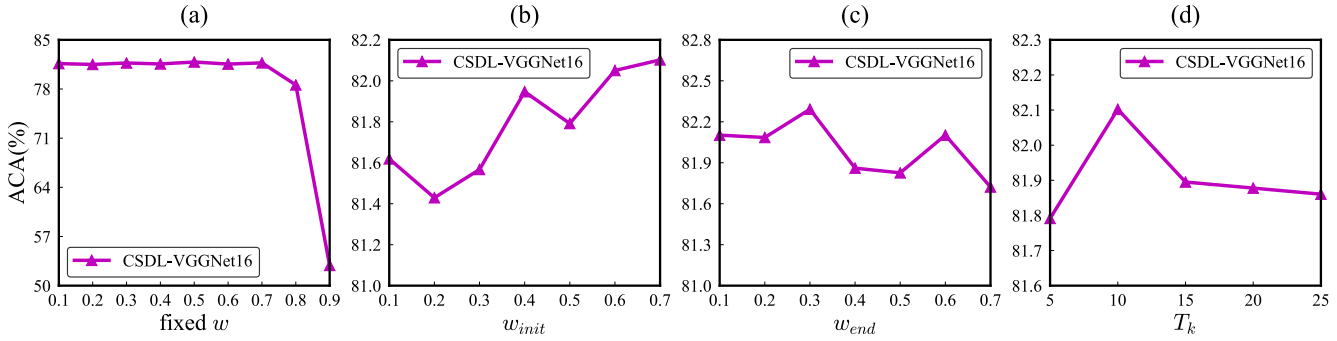

FIGURE 5. The parameter sensitivities of fixed w (a), as well as w_{init} (b), w_{end} (c), and T_k (d) for dynamic $w(T)$ in Eq. (13) w.r.t. ACA. (The default setting for dynamic $w(T)$ is $w_{init} = 0.7$, $w_{end} = 0.1$ and $T_k = 10$. When we conduct experiments to investigate one parameter, the other two are fixed.).

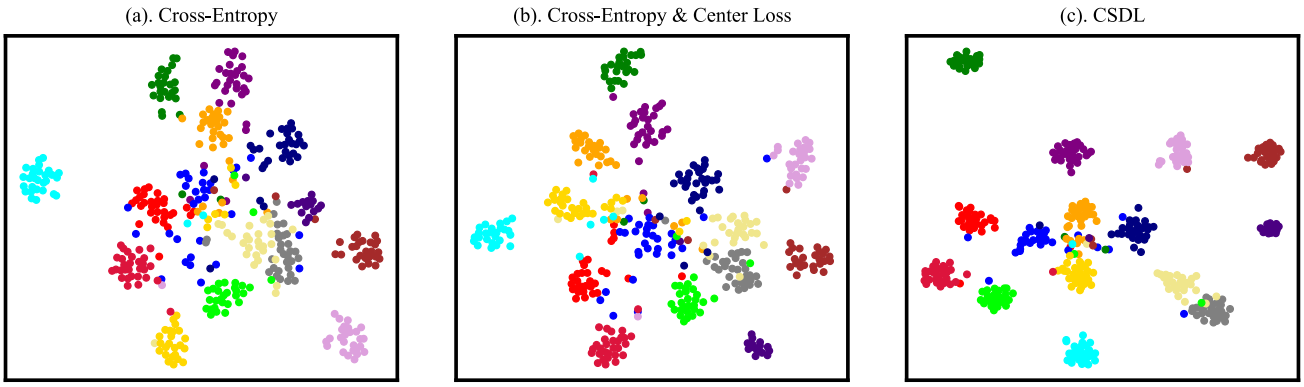

FIGURE 6. Feature compactnesses under the sole supervision of the cross-entropy loss (a), joint supervision between the cross-entropy loss and center loss (b), and our CSDL (c) are visualized using t-SNE graphs respectively.

TABLE 4. The ACA (%) of two models using different similarity functions (Cosine and Euclidean-based).

Model	Cosine	Euclidean
CSDL-VGGNet16	**82.31**	81.52
CSDL-ResNet50	**86.11**	85.97

we train CSDL-VGGNet16 and CSDL-ResNet50 using the cosine similarity function and euclidean similarity function, respectively. Experimental results are shown in Table 4. We can observe that while performances are on par with each other, the cosine similarity function yields slightly better classification results. Therefore, the cosine similarity function is selected as the default similarity measurement in our method.

E. CHOICE FOR WEIGHT OF DISTRIBUTED LABEL

In this subsection, we first analyze the performances when using different fixed weights w for the distributed labels. Then, we look at the sensitivity of each parameter in Eq. (13) when dynamically selecting the distributed label weight w. Finally, we compare performances obtained using fixed and dynamic weights.

For fixed w, we evaluate its performance by sequentially selecting w in $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. Experimental results are given in Fig. 5 (a). From this figure, we can see that our approach is robust when $w \leq 0.7$. However, as w increases beyond this, the performance decreases sharply. Towards the sensitivity of w_{int}, w_{end}, and T_k in Eq. (13), results are presented in Fig. 5 (b) - (d). To compare performances of using fixed and dynamic weights, we choose the fixed w to be 0.2 and 0.8 and train CSDL-VGGNet16 models. We also train the CSDL-VGGNet16 model with a dynamic $w(T)$.

From Fig. 4, we can observe that training with a dynamic $w(T)$ slows down the training process in the early epochs but produces a better result in the end. In contrast, training with a large fixed value for w slows down the training process too much and also prevents the model from taking full advantage of the ground-truth guidance, thus ultimately leading to a worse performance than using a dynamic $w(T)$.

F. EFFECTIVENESS OF USING DISTRIBUTED LABELING

In our proposed method, the final loss function is the weighted sum between the center loss and classification loss. To demonstrate the effectiveness of using distributed labeling as we proposed, we compare results between the case of using the ground-truth label in the final loss function and the case of using our proposed distributed labeling. We choose VGGNet16 as the backbone. The result of using

the ground-truth label is 78.51%, lower than using distributed labeling (82.31%). This verifies the effectiveness of using our proposed distributed labeling.

G. EFFECTIVENESS OF ENHANCING FEATURE COMPACTNESS

To update our distributed labels and drive the learned visual representations closer, we adopt the center loss in our CSDL. The center loss module produces loss penalties based on feature distances between images and their category centers, leading to compact feature representation in the training process. We use the backbone network VGGNet16 on the CUB200-2011 dataset as an example. Fig. 6 uses t-SNE to illustrate the compactness of learned visual representation in a 2-dimensional feature space. As presented in Fig. 6, compared with features under the sole supervision of the cross-entropy loss, the adoption of the center loss contributes to closer feature representations. Furthermore, by employing the center loss along with our distributed labeling, the feature representation compactness is even further enhanced, demonstrating that the center loss module benefits the distributed labeling module.

VI. CONCLUSION

In this work, we presented a simple yet effective approach for fine-grained visual classification. Specifically, we propose to reduce the prediction confidence by assigning distributed labels for regularizing FGVC models. Additionally, we adopted joint supervision from the center loss and cross-entropy loss to learn more discriminative deep features. Through in-depth analysis, we have quantitatively and qualitatively validated the effectiveness of our CSDL on regularizing fine-grained classification. Experiments also showed that our proposed CSDL can be integrated with existing fine-grained methods and obtain state-of-the-art performance.

REFERENCES

[1] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[2] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," School Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009, vol. 1, no. 4, p. 7.

[3] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. CNS-TR-2011-001, 2011.

[4] G. Van Horn, S. Branson, R. Farrell, S. Haber, J. Barry, P. Ipeirotis, P. Perona, and S. Belongie, "Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 595–604.

[5] A. Khosla, N. Jayadevaprakash, B. Yao, and F.-F. Li, "Novel dataset for fine-grained image categorization: Stanford dogs," in *Proc. CVPR Workshop Fine-Grained Vis. Categorization (FGVC)*, 2011, vol. 2, no. 1, pp. 1–2.

[6] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar, "Cats and dogs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3498–3505.

[7] J. Liu, A. Kanazawa, D. Jacobs, and P. Belhumeur, "Dog breed classification using part localization," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 172–185.

[8] M.-E. Nilsback and A. Zisserman, "A visual vocabulary for flower classification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2006, pp. 1447–1454.

[9] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi, "Fine-grained visual classification of aircraft," 2013, *arXiv:1306.5151*. [Online]. Available: http://arxiv.org/abs/1306.5151

[10] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3D object representations for fine-grained categorization," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Dec. 2013, pp. 554–561.

[11] Y. Kawano and K. Yanai, "Automatic expansion of a food image dataset leveraging existing categories with domain adaptation," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 3–17.

[12] S. Hou, Y. Feng, and Z. Wang, "VegFru: A domain-specific dataset for fine-grained visual categorization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 541–549.

[13] X.-S. Wei, Q. Cui, L. Yang, P. Wang, and L. Liu, "RPC: A large-scale retail product checkout dataset," 2019, *arXiv:1901.07249*. [Online]. Available: http://arxiv.org/abs/1901.07249

[14] X.-S. Wei, J. Wu, and Q. Cui, "Deep learning for fine-grained image analysis: A survey," 2019, *arXiv:1907.03069*. [Online]. Available: http://arxiv.org/abs/1907.03069

[15] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, "Part-based R-CNNs for fine-grained category detection," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 834–849.

[16] S. Branson, G. Van Horn, P. Perona, and S. Belongie, "Improved bird species recognition using pose normalized deep convolutional nets," in *Proc. Brit. Mach. Vis. Conf.*, 2014, p. 7.

[17] D. Lin, X. Shen, C. Lu, and J. Jia, "Deep LAC: Deep localization, alignment and classification for fine-grained recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1666–1674.

[18] S. Huang, Z. Xu, D. Tao, and Y. Zhang, "Part-stacked CNN for fine-grained visual categorization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1173–1182.

[19] 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 Recognit.*, vol. 76, pp. 704–714, Apr. 2018.

[20] H. Zheng, J. Fu, T. Mei, and J. Luo, "Learning multi-attention convolutional neural network for fine-grained image recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5209–5217.

[21] J. Fu, H. Zheng, and T. Mei, "Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4438–4446.

[22] C. Lei, L. Jiang, J. Ji, W. Zhong, and H. Xiong, "Weakly supervised learning of object-part attention model for fine-grained image classification," in *Proc. AAAI Conf. Artif. Intell.*, Oct. 2018, pp. 4075–4081.

[23] M. Sun, Y. Yuan, F. Zhou, and E. Ding, "Multi-attention multi-class constraint for fine-grained image recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 805–821.

[24] Z. Yang, T. Luo, D. Wang, Z. Hu, J. Gao, and L. Wang, "Learning to navigate for fine-grained classification," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 420–435.

[25] Y. Wang, V. I. Morariu, and L. S. Davis, "Learning a discriminative filter bank within a CNN for fine-grained recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4148–4157.

[26] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear convolutional neural networks for fine-grained visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1309–1322, Jun. 2018.

[27] T.-Y. Lin and S. Maji, "Improved bilinear pooling with CNNs," in *Proc. Brit. Mach. Vis. Conf.*, 2017, pp. 1–12.

[28] Y. Gao, O. Beijbom, N. Zhang, and T. Darrell, "Compact bilinear pooling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 317–326.

[29] S. Kong and C. Fowlkes, "Low-rank bilinear pooling for fine-grained classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 365–374.

[30] Y. Cui, F. Zhou, J. Wang, X. Liu, Y. Lin, and S. Belongie, "Kernel pooling for convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2921–2930.

[31] P. Li, J. Xie, Q. Wang, and Z. Gao, "Towards faster training of global covariance pooling networks by iterative matrix square root normalization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 947–955.

[32] A. Dubey, O. Gupta, R. Raskar, and N. Naik, "Maximum-entropy fine grained classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 637–647.
[33] S. Reed, Z. Akata, H. Lee, and B. Schiele, "Learning deep representations of fine-grained visual descriptions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 49–58.
[34] X. He and Y. Peng, "Fine-grained image classification via combining vision and language," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5994–6002.
[35] Y. Cui, F. Zhou, Y. Lin, and S. Belongie, "Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1153–1162.
[36] Z. Xu, S. Huang, Y. Zhang, and D. Tao, "Webly-supervised fine-grained visual categorization via deep domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1100–1113, May 2018.
[37] L. Niu, A. Veeraraghavan, and A. Sabharwal, "Webly supervised learning meets zero-shot learning: A hybrid approach for fine-grained classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7171–7180.
[38] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 499–515.
[39] 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," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5012–5021.
[40] Y. Chen, Y. Bai, W. Zhang, and T. Mei, "Destruction and construction learning for fine-grained image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5157–5166.
[41] Z. Xu, S. Huang, Y. Zhang, and D. Tao, "Augmenting strong supervision using Web data for fine-grained categorization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2524–2532.
[42] J. Krause, B. Sapp, A. Howard, H. Zhou, A. Toshev, T. Duerig, J. Philbin, and L. Fei-Fei, "The unreasonable effectiveness of noisy data for fine-grained recognition," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 301–320.
[43] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Proc. Workshop Challenges Represent. Learn. (ICML)*, vol. 3, 2013, p. 2.
[44] Z. Zheng, L. Zheng, and Y. Yang, "Unlabeled samples generated by GAN improve the person re-identification baseline *in vitro*," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3754–3762.
[45] Y. Huang, J. Xu, Q. Wu, Z. Zheng, Z. Zhang, and J. Zhang, "Multi-pseudo regularized label for generated data in person re-identification," *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1391–1403, Mar. 2019.
[46] G. Ding, S. Zhang, S. Khan, Z. Tang, J. Zhang, and F. Porikli, "Feature affinity-based pseudo labeling for semi-supervised person re-identification," *IEEE Trans. Multimedia*, vol. 21, no. 11, pp. 2891–2902, Nov. 2019.
[47] R. Müller, S. Kornblith, and G. Hinton, "When does label smoothing help?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 4694–4703.
[48] X. He, Y. Peng, and J. Zhao, "Which and how many regions to gaze: Focus discriminative regions for fine-grained visual categorization," *Int. J. Comput. Vis.*, vol. 127, no. 9, pp. 1235–1255, Sep. 2019.
[49] Y. Peng, X. He, and J. Zhao, "Object-part attention model for fine-grained image classification," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1487–1500, Mar. 2018.
[50] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
[51] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2818–2826.
[52] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8697–8710.
[53] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 4780–4789.
[54] Y. Huang, Y. Cheng, A. Bapna, O. Firat, D. Chen, M. Chen, H. Lee, J. Ngiam, Q. V. Le, and Y. Wu, "Gpipe: Efficient training of giant neural networks using pipeline parallelism," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 103–112.
[55] G. Pereyra, G. Tucker, J. Chorowski, Ł. Kaiser, and G. Hinton, "Regularizing neural networks by penalizing confident output distributions," 2017, *arXiv:1701.06548*. [Online]. Available: http://arxiv.org/abs/1701.06548
[56] W. Ge, X. Lin, and Y. Yu, "Weakly supervised complementary parts models for fine-grained image classification from the bottom up," in *Proc. CVPR*, 2019, pp. 3034–3043.
[57] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556
[58] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.
[59] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
[60] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[61] D. Wang, Z. Shen, J. Shao, W. Zhang, X. Xue, and Z. Zhang, "Multiple granularity descriptors for fine-grained categorization," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 2399–2406.
[62] J. Krause, H. Jin, J. Yang, and L. Fei-Fei, "Fine-grained recognition without part annotations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 5546–5555.
[63] Y. Wang, J. Choi, V. Morariu, and L. S. Davis, "Mining discriminative triplets of patches for fine-grained classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1163–1172.
[64] X. Liu, T. Xia, J. Wang, Y. Yang, F. Zhou, and Y. Lin, "Fully convolutional attention networks for fine-grained recognition," 2016, *arXiv:1603.06765*. [Online]. Available: http://arxiv.org/abs/1603.06765
[65] H. Yao, S. Zhang, Y. Zhang, J. Li, and Q. Tian, "Coarse-to-fine description for fine-grained visual categorization," *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4858–4872, Oct. 2016.
[66] M. Lam, B. Mahasseni, and S. Todorovic, "Fine-grained recognition as hsnet search for informative image parts," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2520–2529.
[67] C. Yu, X. Zhao, Q. Zheng, P. Zhang, and X. You, "Hierarchical bilinear pooling for fine-grained visual recognition," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 574–589.
[68] R. Ji, L. Wen, L. Zhang, D. Du, Y. Wu, C. Zhao, X. Liu, and F. Huang, "Attention convolutional binary neural tree for fine-grained visual categorization," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 10468–10477.
[69] M. Moghimi, S. J. Belongie, M. J. Saberian, J. Yang, N. Vasconcelos, and L.-J. Li, "Boosted convolutional neural networks," in *Proc. Brit. Mach. Vis. Conf.*, 2016, p. 6.

PENGZHEN DU received the Ph.D. degree from the Nanjing University of Science and Technology, in 2015. He is currently an Assistant Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology. His research interests include computer vision, deep learning, swarm intelligent, evolutionary computation, and parallel computing.

ZEREN SUN received the B.S. degree in computer science from the Nanjing University of Science and Technology, China, and the M.S. degree in robotics technology from Carnegie Mellon University. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Nanjing University of Science and Technology. His research interests include computer vision, deep learning, fine-grained classification, and learning from noise.

YAZHOU YAO (Member, IEEE) received the Ph.D. degree in computer science from the University of Technology Sydney, Australia, in 2018, with the support of the China Scholarship Council. From July 2018 to July 2019, he worked as a Research Scientist with the Inception Institute of Artificial Intelligence, Abu Dhabi, United Arab Emirates. He is currently a Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology. His research interests include multimedia processing and machine learning.

ZHENMIN TANG received the Ph.D. degree from the School of Computer Science and Engineering, Nanjing University of Science and Technology, in 2002. He is currently a Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology. His research interests include pattern recognition, intelligent systems, image processing, and object detection.