Improving Image Classification Through Joint Guided Learning

Peipei Zhao, Hang Yao, Xiangzeng Liu, Ruyi Liu, and Qiguang Miao, *Senior Member, IEEE*

***Abstract*— We propose a simple yet effective approach, called joint guided learning (JGL), based on knowledge distillation for image classification. Knowledge distillation transfers knowledge from a teacher model to a student model. The teacher model utilizes both correct and incorrect predictive labels to guide the optimization of the student model. However, the incorrect predictive labels have a negative influence on the student model. Moreover, it is not enough to rely solely on predictive labels to transfer knowledge. The student model should make full use of the features of the teacher model. To mitigate these issues, we design a JGL approach that exploits the joint guidance of features and the predictive labels. Our method involves two novel components: 1) a guided label refinery module that considers the correct predictions and ignores the incorrect predictions and 2) a channel distillation (CD) module that guides the student model to learn the attention map of each channel from feature maps in the teacher model. Experimental results show that our approach can achieve 82.18%, 66.08%, 88.36%, and 90.14% accuracy on CIFAR-100, TinyImageNet, CUB-200-2011, and Stanford Dogs, respectively. In addition, the proposed method consistently outperforms the state-of-the-art approaches on four image classification datasets. Ablation studies further show the contributions of different components in our method.**

***Index Terms*— Channel distillation (CD), guided label refinery (GLR) module, image classification, joint guided learning (JGL) approach, knowledge distillation.**

I. Introduction

Thanks to the breakthroughs in the training and design of convolutional neural networks (CNNs) [1]–[3], the performance of image classification on large-scale datasets and fine-grained datasets has undergone unprecedented improvements in the past few years. The supervised learning systems of image classification mainly consist of three parts: models, data, and labels. Models have been widely studied in recent years. The performance of image classification has shown improvements by increasing the depth of the models [1], [4], adding new loss functions and optimization strategies [5], [6], and introducing new activation and normalization layers [7], [8]. Meanwhile, various data augmentation approaches [9]–[12] have improved the performance of supervised learning systems. In addition, many studies have improved the performance from the perspective of label smoothing.

Label smoothing has been successfully utilized to improve the performance of image classification. Originally, it was proposed as a strategy [13] to improve accuracy by computing cross-entropy without “hard” targets from datasets. Since then, many image classification models [14]–[16] have added label smoothing to training procedures. Yun *et al.* [17] introduced a novel regularization approach that penalized the predictive distribution between similar samples. The method distilled the predictive distribution of deep neural networks (DNNs) between different samples of the same label during the training process. Then, knowledge distillation was explored for label smoothing for evaluating the accuracy of the teacher network and for obtaining prior knowledge of optimal logit layer geometry [18]. However, these methods typically treated models, data, and labels in isolation, neglecting their relationships between them. In this article, we propose a new joint guided learning (JGL) approach by combining models with labels.

Although knowledge distillation has been widely explored for label smoothing [13]–[18], there are some challenges for knowledge distillation to label smoothing. One is that the predictions of a teacher are not completely correct. In the training process, a student makes decisions based on the predictive results of a teacher. As shown in Fig. 1, samples are misclassified by the teacher. In addition, the incorrect predictions of the teacher are transferred to the student. These predictions have a negative impact on the student. Second is that it is not enough to rely on the predictive distribution of the teacher to transfer knowledge to the student. In addition, the student cannot accurately learn the information of feature maps in the teacher module.

To overcome the aforementioned challenges, we propose a novel approach called JGL, which integrates the guided label refinery (GLR) module and the channel distillation (CD) module for obtaining more accurate probability distribution. The GLR module not only obtains soft and informative labels but also avoids the negative influence of the incorrect predictions of the teacher on the student. Meanwhile, the importance of each channel is different, and it cannot be learned by the GLR

Manuscript received December 30, 2021; revised April 9, 2022; accepted April 20, 2022. Date of publication May 16, 2022; date of current version May 19, 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFC0807500; in part by the National Natural Science Foundations of China under Grant 61772396, Grant 61772392, and Grant 61902296; in part by the Xi’an Key Laboratory of Big Data and Intelligent Vision under Grant 201805053ZD4CG37; in part by the National Natural Science Foundation of Shaanxi Province under Grant 2020JQ-330 and Grant 2020JM-195; in part by the China Postdoctoral Science Foundation under Grant 2019M663640; in part by the Guangxi Key Laboratory of Trusted Software under Grant KX202061; in part by the Key Research and Development Projects of Qingdao Science and Technology Plan under Grant 21-1-2-18-xx; in part by the Academy of Integrated Circuit Innovation of Xidian University in ChongQing Foundation under Grant CQIRI-CXYHT-2021-06; and in part by the Fundamental Research Funds for the Central Universities under Grant 20101216855. The Associate Editor coordinating the review process was Bardia Yousefi. *(Corresponding author: Qiguang Miao.)*

The authors are with the School of Computer Science and Technology, Xidian University, Xi’an 710071, China (e-mail: qgmiao@126.com).

Digital Object Identifier 10.1109/TIM.2022.3173630

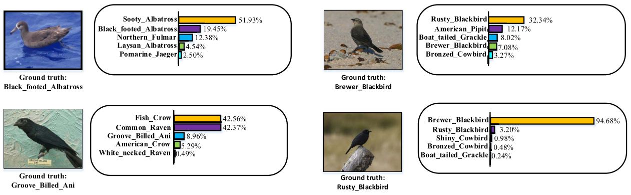


Fig. 1. Top-1 softmax scores on misclassified samples. The ground truth is the real label of an image.

module. To solve this issue, we introduce the CD module that can guide the student to learn the attention information of each channel. Extensive experiments demonstrate the effectiveness of our approach. To summarize, the contributions of this article are listed as follows.

1) We propose a GLR module to obtain more accurate probability distribution. Unlike existing approaches that transfer both correct and incorrect predictions of the teacher to the student, the GLR module only utilizes the correct predictions and ignores the incorrect predictions.
2) We propose the CD to capture more discriminative information for the student model. The CD module can force the student to learn the attention map of each channel from feature maps in the teacher.
3) We demonstrate the effectiveness of our approach using DNNs, such as deep residual network (ResNet) [1], densely connected networks (DenseNet) [2], and EfficientNet [3] trained for image classification tasks on various datasets including CIFAR-100 [19], TinyImageNet[1], Caltech-UCSD Birds-200-2011 (CUB-200-2011) [20], and Stanford Dogs [21].

The rest of this article is organized as follows. The related work is briefly reviewed in Section II. Our approach is described in detail in Section III, which includes the GLR module and the CD module. In addition, Section IV shows the experiments and analysis. Finally, we summarize this work in Section V.

II. Related Work

A. Label Smoothing and Regularization

Softening labels have been utilized successfully to improve generalization. Szegedy *et al.* [22] first proposed label smoothing as a mechanism that encouraged the inception architecture to be less confident and improved the performance of the architecture on the ImageNet dataset. Since then, label smoothing has been widely used in image classification. ScheduledDropPath [14] was introduced by linearly increasing the probability of dropping out a path over the course of training. The method was enforced with the purpose of improving generalization of models. A new regularized evolutionary algorithm, AmoebaNet-A [15], was introduced to improve the performance of standard tournament selection. In addition, LR [17] was an iterative procedure that used data and a neural network model to improve crop labels during training. Muller *et al.* [23] tried to shed light upon how label smoothing changed the representations learned by the penultimate layer of the network. Meanwhile, it demonstrated that label smoothing implicitly calibrated learned models.

B. Knowledge Distillation

Knowledge optimization is an effective method to address per-image class-level similarity. It consists of a teacher network and a student network, in which the teacher network is first trained, and then it is utilized to guide the optimization of the student network. Therefore, the per-image class-level similarity can be transmitted to the student network by the output of the teacher network. Recently, a long line of articles have paid attention to teacher–student optimization for various purposes. Initially, it was proposed to distill knowledge from a larger teacher model and compress it into a smaller model [24]–[26] or use the pre-trained weights of a shallower network [4], [27] to initialize a deeper network. Then, the teacher–student optimization was extended in many ways, including using various ways of teacher supervision [22], [28], [29], using an average of multiple consecutive student models as the teacher model to provide a better guidance [30], adding supervisory signals in intermediate neural layers [31], and allowing two networks to learn collaboratively and teach each other in the training process [13]. Recently, born-again network [32] was proposed. It was utilized for optimizing the same network in generations. Later, a snapshot distillation [33] was used to perform teacher–student optimization in one generation. Xie *et al.* [34] explored a simple and effective self-training module that the teacher network was used to generate pseudolabels and the student network was trained by labeled and pseudolabeled images. In addition, knowledge distillation was explored for label smoothing for evaluating the performance of the teacher network and for obtaining prior knowledge of optimal logit layer geometry [18]. The main difference between prior works and our method is that our approach takes into account the impact of incorrect predictions of the teacher module on the student module.

C. Attention-Guided Approaches

The attention mechanism has been widely used in many fields [35]–[38]. As we all know, it stems from the human visual system that has low resolution in the periphery and high resolution in the fovea. In addition, the human visual system focuses on part of the information while ignoring others [39]. The attention mechanism bridges this gap by making networks intentionally pay attention to useful information to improve performance of networks. To generate the global context features, squeeze-and-excitation network (SENet) [40], pointwise spatial attention network (PSANet) [41], gather–excite network (GENet) [42], and global correlation network (GCN) [43] modeled the global context by using rescaling to different channels to recalibrate dependencies among channels, or utilizing addition fusion. Jiang *et al.* [44] proposed an approach that consisted of two networks: density attention network (DANet) and attention scaling network (ASNet). DANet could generate attention masks that represented regions of

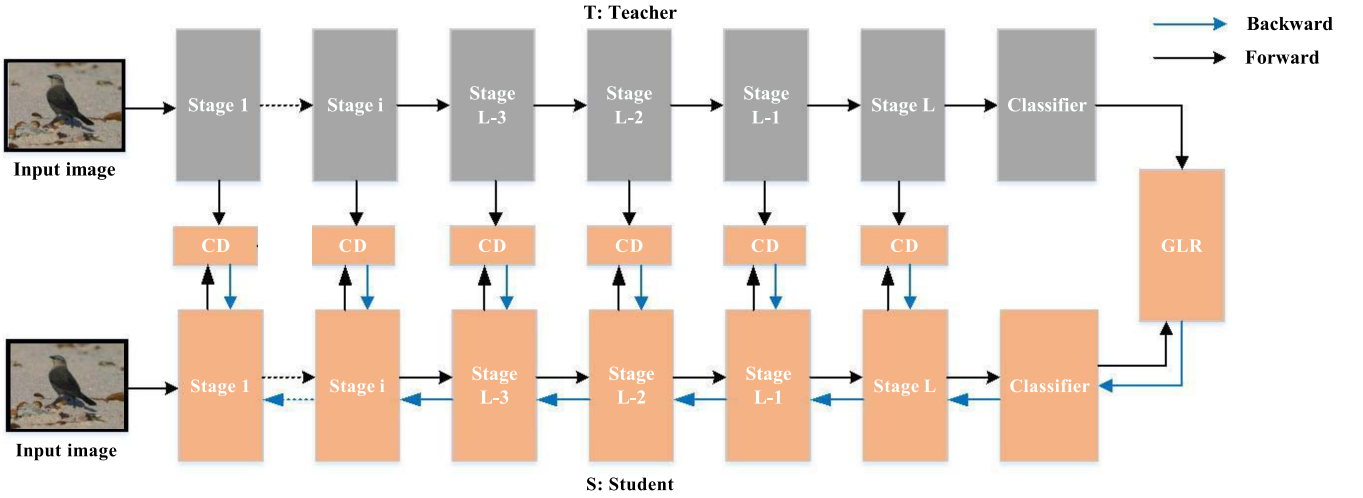


Fig. 2. Overview of our network architecture. There are two parts: the GLR model and the CD model. The GLR module can ignore the misclassified samples. In addition, the CD module can guide the student to learn the feature maps of the teacher. T and S are our teacher feature extractor and student feature extractor, which all have L stages. The Classifier represents the combination of average pooling layer and a fully connected layer with softmax layer at the end.

different density levels by performing a pixelwise density segmentation task. ASNet could output separate attention-based density maps by multiplying scaling factors and density maps by attention masks. Zhao *et al.* [45] explored pairwise self-attention which generalized the standard dot-product attention, and patchwise self-attention which was strictly more powerful than convolution. The bilinear attentional transform (BA-Transform), proposed by Chi *et al.* [46], could model a wide spectrum of global or local attentional operations. Furthermore, the grouped BA-transforms essentially utilized different attentional operations to different groups of feature channels for dealing with the discrepancy among features. The main difference between prior works and our method is that our approach utilizes the CD model to force the student to learn the attention map of each channel from feature maps in the teacher.

III. Joint Guided Learning

The framework of our proposed method is shown in Fig. 2. It consists of two parts: GLR and CD. The GLR is devised on the basis of guided knowledge distillation [39] and LR [17]. It can generate soft, collective, informative, and dynamic labels by only transferring the positive predictions to the student. The CD makes full use of the weight of each channel in the teacher module to guide the student module to learn more discriminative features.

A. Guided Label Refinery

There have been a number of label smoothing methods modifying crop labels during training. In addition, the previous works can be roughly divided into the following categories. Reed *et al.* [47] proposed a novel way that augmented incomplete and noisy labels utilizing other models to improve label consistency. To prevent overfitting, DisturbLabel [6] randomly replaced a part of labels as incorrect values in each iteration. These methods could achieve competitive recognition results on some image recognition datasets. However, these methods could not address label incompleteness. LR [17] utilized a neural network model and data to refine crop labels during training. This approach could deal with label incompleteness by updating the ground truth labels in an iterative procedure. However, this approach ignored the negative influence of incorrect predictions. In this article, we propose a novel method called GLR that adopts the idea of LR to refine crop labels and deals with label incompleteness.

1) Label Refinery [17]: Given a dataset $\boldsymbol{B} = \{(\boldsymbol{X}_i, Y_i)\}_{i=1}^{N}$, where $\boldsymbol{X}_i$ denotes the ith image, Y_i is the label of the ith image, and N is the number of samples. $Y_i \in \{1, 2, \ldots, m\}$, where m is the number of categories. We can define a new dataset $\widetilde{\boldsymbol{B}} = \{(f(\boldsymbol{X}_i), Y_i)\}_{i=1}^{N}$, where f denotes the data augmentation operation that generates random crops for the image $\boldsymbol{X}_i$. f consists of Resize, RandomCrop, and RandomHorizontalFlip functions. The Resize function resizes the image to a specific size. It can ensure that the input size of images is consistent. The RandomCrop function can randomly crop images. It can reduce data noise and increase model stability. It reduces the sensitivity of the model to missing values. The RandomHorizontalFlip function randomly flips images. It can reduce the effect of image flipping on the model.

The first LR network N_{θ_1} is trained by using the dataset $\widetilde{\boldsymbol{B}}$. The predictions of the dataset $\widetilde{\boldsymbol{B}}$ can be formulated as

$$\boldsymbol{P_i}^{(1)} = N_{\theta_1}(f(\boldsymbol{X}_i)) \tag{1}$$

where $\boldsymbol{P}_i^{(1)}$ denotes the predictive probability for the ith sample. The second LR network N_{θ_2} is trained over the same dataset, but it utilizes the predictive probability generated by N_{θ_1}. Therefore, we use the new augmented dataset $\widetilde{\boldsymbol{B}}_1 = \{(f(\boldsymbol{X_i}), \boldsymbol{P}_i^{(1)})\}_{i=1}^{N}$ instead of the dataset $\widetilde{\boldsymbol{B}}$ to train the N_{θ_2}.

2) Guided Label Refinery: There are incorrect predictions in the LR network. As shown in Fig. 1, samples are misclassified by the first network. In addition, the incorrect predictions of the first network are transferred to the second network. These predictions have a negative impact on the second network. To overcome this issue, we propose a GLR module, which only transfers the correct predictions to the second network.

The training process of the first network in the GLR is the same as that of the LR. Therefore, we obtain $\boldsymbol{P}_i{}^{(1)}$ in the first network. In addition, the final result $C_i^{(1)}$ can be expressed as

$$C_i^{(1)} = \left(\underset{\text{index}}{\operatorname{argmax}}\left(\boldsymbol{P}_i^1(\text{index})\right)\right) + 1 \tag{2}$$

where index $\in \{0, 1, 2, \ldots, m - 1\}$, m is the number of categories, and $C_i^{(1)}$ denotes the predictive label of the ith sample in the first network.

The second GLR network N_{θ_2} is trained by the same dataset $\widetilde{\boldsymbol{B}}$, but it utilizes labels generated by N_{θ_1}. There is an issue that the incorrect predictions of the first network N_{θ_1} is transferred to the second network N_{θ_2}. It will deteriorate the performance of the second network. Therefore, this article proposes the GLR module. To avoid the impact of the incorrect labels on the second network, we adopt a strategy that the network N_{θ_1} only transfers the correct predictions to the network N_{θ_2} and ignores the incorrect predictions. Therefore, the new augmented dataset can be represented as $\widetilde{\boldsymbol{B}}_1 = \{(f(\boldsymbol{X}_i), I(C_i^{(1)}, Y_i) \cdot \boldsymbol{P}_i^{(1)})\}_{i=1}^N$, where I is an indicator function and $I(C_i^{(1)}, Y_i) = 1$ when $C_i^{(1)} = Y_i$, otherwise $I(C_i^{(1)}, Y_i) = 0$.

The first GLR network N_{θ_1} is trained by the cross-entropy loss. In addition, the second GLR network N_{θ_2} is trained by minimizing the Kullback–Leibler (KL) divergence between its outputs and soft labels generated by the first GLR network N_{θ_1}. We know that our method only considers the correct predictions. Therefore, our method only back-propagates the KL-divergence loss of those samples correctly predicted by the first network and ignores those samples incorrectly predicted. Let $\boldsymbol{P}_i^{(2)} = N_{\theta_2}(f(\boldsymbol{X}_i))$ denote the probability assigned to all categories. The loss function LOSS^g for training model N_{θ_2} can be computed as

$$\text{LOSS}^g = \frac{\sum_{i=1}^N I\left(C_i^{(1)}, Y_i\right) KL\left(\boldsymbol{P}_i^{(2)}, \boldsymbol{P}_i^{(1)}\right)}{\sum_{i=1}^N I\left(C_i^{(1)}, Y_i\right)} \tag{3}$$

where I is an indicator function and $I(C_i^{(1)}, Y_i) = 1$ when $C_i^{(1)} = Y_i$, otherwise $I(C_i^{(1)}, Y_i) = 0$. Therefore, the denominator of (3) is a constant and can be ignored. The KL divergence is also called relative entropy. It can measure the difference between two probability distributions. When two probability distributions are the same, their KL divergence is zero. When the difference between two probability distributions increases, their KL divergence also increases. Therefore, we utilize KL divergence to minimize the difference between $\boldsymbol{P}_i^{(1)}$ and $\boldsymbol{P}_i^{(2)}$ in (3).

Equation (3) is simplified to

$$\begin{aligned}\text{LOSS}^g &= \sum_{i=1}^N I\left(C_i^{(1)}, Y_i\right) KL\left(\boldsymbol{P}_i^{(2)}, \boldsymbol{P}_i^{(1)}\right)\\ &= -\sum_{i=1}^N I\left(C_i^{(1)}, Y_i\right) \cdot \left(\boldsymbol{P}_i^{(1)} \log\left(\frac{\boldsymbol{P}_i^{(2)}}{\boldsymbol{P}_i^{(1)}}\right)\right)\\ &= -\sum_{i=1}^N I\left(C_i^{(1)}, Y_i\right) \cdot \left(\boldsymbol{P}_i^{(1)} \log\left(\boldsymbol{P}_i^{(2)}\right)\right)\\ &\quad + \sum_{i=1}^N I\left(C_i^{(1)}, Y_i\right) \cdot \left(\boldsymbol{P}_i^{(1)} \log\left(\boldsymbol{P}_i^{(1)}\right)\right).\end{aligned} \tag{4}$$

When training the model N_{θ_2}, the $\boldsymbol{P}_i^{(1)}$ of the model N_{θ_1} is known. Therefore, the $\boldsymbol{P}_i^{(1)} \log(\boldsymbol{P}_i^{(1)})$ can be expressed as

$$\boldsymbol{P}_i^{(1)} \log\left(\boldsymbol{P}_i^{(1)}\right) = b \tag{5}$$

where b is a constant. As $I(C_i^{(1)}, Y_i) \in \{0, 1\}$, the second term of (4) is constant and can be removed. Then, (4) can be formulated as

$$\text{LOSS}^g = -\sum_{i=1}^N I\left(C_i^{(1)}, Y_i\right) \cdot \left(\boldsymbol{P}_i^{(1)} \log\left(\boldsymbol{P}_i^{(2)}\right)\right). \tag{6}$$

By minimizing (6), the model N_{θ_2} can be trained with the network via back-propagation in an end-to-end manner. Note that the first model N_{θ_1} is trained by using cross-entropy loss with the original image-level labels. In addition, the model N_{θ_2} is trained by (6). The supervisions are applied only to the output $\boldsymbol{P_i}^{(2)}$, where $I(C_i^{(1)}, Y_i)$ is equal to 1, namely, $C_i^{(1)} = Y_i$ in the model N_{θ_1}.

B. Channel Distillation

Inspired by the recent advance in channelwise attention [39], we introduce the CD into our module for learning discriminative features and estimating more accurate probability distribution. The CD utilizes the attention information of the teacher to supervise the student to learn features. Our module design is generic and can be implemented on the top of any state-of-the-art backbone feature extractor, such as ResNet and DenseNet. $\boldsymbol{T}$ and $\boldsymbol{S}$ are our teacher feature extractor and student feature extractor, which all have L stages. The output feature map from any intermediate stages in the teacher network is represented as $\boldsymbol{T}_i^l \in \mathbb{R}^{H_l \times W_l \times C_l}$, where H_l, W_l, and C_l are the height, the width, and the number of channels of the feature map at the lth stage. Similarly, the output feature map from any intermediate stages in the student network is represented as $\boldsymbol{S}_i^l \in \mathbb{R}^{H_l \times W_l \times C_l}$. $\boldsymbol{T}_i^l$ is the feature map of the ith image in the lth layer of the teacher network. $\boldsymbol{S}_i^l$ is the feature map of the ith image in the lth layer of the student network.

Similar to SENet [40], the channelwise attention generates channelwise statistic $\boldsymbol{ZT}_i^l(c), c \in \{1, 2, 3, \ldots, C\}$ by using global average pooling (GAP). C is the number of channels. We can write the channelwise statistic as

$$\boldsymbol{ZT}_i^l(c) = \frac{1}{H \times W} \sum_{h=1}^H \sum_{w=1}^W \boldsymbol{T}_i^l(c)(h, w) \tag{7}$$

where $\boldsymbol{ZT}_i^l(c)$ is the weight of cth channel of ith sample in the lth layer of the teacher network.

From (7), we know that the importance of each channel is different. To transfer the weight of each channel from the teacher to the student, we utilize the GAP to calculate the importance of each channel in the teacher network and that of

Algorithm 1 JGL

```
Require: Given a dataset $\boldsymbol{B} = \{(\boldsymbol{X}_i, Y_i)\}_{i=1}^{N}$, we can get a new dataset $\widetilde{\boldsymbol{B}} = \{(f(\boldsymbol{X}_i), Y_i)\}_{i=1}^{N}$. The function $f$ is data augmentation operation that generates random crops for the image $\boldsymbol{X}_i$.
1. Train the teacher network $N_{\theta_1}$ with minimizing the cross-entropy loss on the image-level labels
   $\frac{1}{N}\sum_{i=1}^{N} L(Y_i, N_{\theta_1}(f(\boldsymbol{X}_i)))$
2. Utilize the trained teacher network to generate soft labels for a new dataset $\widetilde{\boldsymbol{B}}$:
   $\boldsymbol{P}_i^{(1)} = N_{\theta_1}(f(\boldsymbol{X}_i))$
3. Train an equal student model $N_{\theta_2}$ with minimizing the total loss to the student model:
   $LOSS^{gc} = (1-\lambda)\cdot LOSS^{g} + \lambda \cdot LOSS(s,t)$
```

the student network, respectively. Then, the teacher supervises the student to learn the attention information of each channel by using the CD. In a similar way, the channelwise statistics of the student network can be formulated as

$$\boldsymbol{ZS}_i^l(c) = \frac{1}{H \times W}\sum_{h=1}^{H}\sum_{w=1}^{W}\boldsymbol{S}_i^l(c)(h, w) \tag{8}$$

where $\boldsymbol{ZS}_i^l(c)$ represents the weight of cth channel of ith sample in the lth layer of the student module. To transfer the attention information of the teacher to the student, we introduce the channelwise attention loss. It is defined as

$$\text{LOSS}(s,t) = \sum_{l=1}^{L}\frac{\sum_{i=1}^{N}\sum_{c=1}^{C}\left(\boldsymbol{ZS}_i^l(c) - \boldsymbol{ZT}_i^l(c)\right)^2}{N \times C} \tag{9}$$

where $\text{LOSS}(s,t)$ denotes the channelwise attention loss error between the student and the teacher.

C. Joint Guided Learning

In general, for the training dataset $\widetilde{\boldsymbol{B}}$, the student network jointly minimizes the total loss with weight parameters λ

$$\text{LOSS}^{gc} = (1-\lambda)\cdot\text{LOSS}^{g} + \lambda\cdot\text{LOSS}(s,t). \tag{10}$$

The full training procedure with the proposed loss LOSS^{gc} is described in Algorithm 1. In addition, the structure of our network is shown in Fig. 2.

IV. Experiments

In Section IV-A, we first introduce the experimental setup including datasets, base networks, training hyperparameters, and evaluation metric. Then, ablation studies are given in Section IV-B following the quantitative analysis in Section IV-C. In addition, we give comparison with state-of-the-art methods in Section IV-D. Finally, we further give qualitative analysis and visualization results in Section IV-E.

A. Experimental Setup

Datasets: To verify the generalization of our approach, we conduct experiments on four image classification datasets, including CIFAR-100, TinyImageNet, CUB-200-2011, and Stanford Dogs. Specifically, CIFAR-100 and TinyImageNet datasets are used for conventional image classification tasks. They include multiple super-classes. CUB-200-2011 and Stanford Dogs datasets are used for fine-grained visual categorization tasks. They include a super-class with multiple sub-classes.

CIFAR-100: It contains 100 categories, each of which consists of 600 images: 500 images as training dataset and 100 images as testing dataset.

TinyImageNet: It contains 200 categories each of which consists of 500 training images, 50 validation images, and 50 testing images.

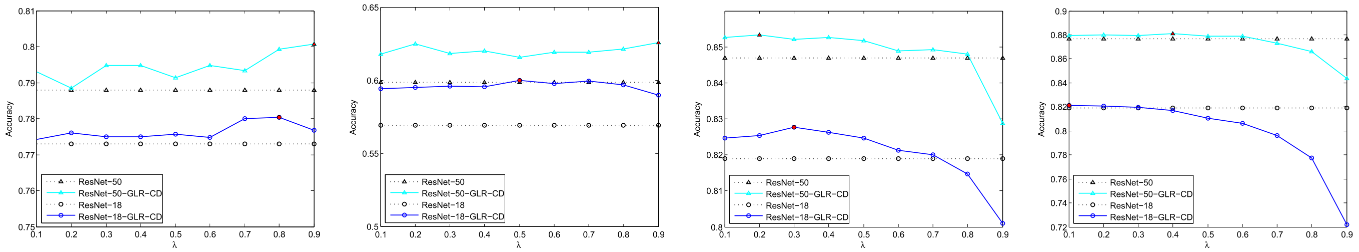
CUB-200-2011: It is North-American bird dataset. The dataset is split into two parts: 5994 images with 200 species for training and 5794 images with 200 species for testing.

Stanford Dogs: It is a dog dataset. It consists of 12 000 training images and 8580 testing images over 120 dog categories.

1) Base Networks: On CIFAR and TinyImageNet, these networks are trained from scratch. Image size and parameter setting are set with reference to ResNet [1], DenseNet [2], and EfficientNet [3]. On CUB-200-2011 and Stanford Dogs, we train these networks on the basis of the ImageNet pre-trained models which are included in Pytorch. Image size and parameter setting are set with reference to Cross-X [48].

2) Implementation Details: We utilize one state-of-the-art DNN architecture: ResNet [1]. The network is trained using stochastic gradient descent (SGD) with momentum 0.9 and weight decay 0.0005. For CIFAR-100 and TinyImageNet, the mini-batch size is 128, the input size is 32×32, and the initial learning rate is set to be 0.1. We train the networks for 200 epochs and decay the learning rate by 0.1 after epochs 100 and 150. In other words, the learning rate is set to 0.1 at $0 < \text{epoch} < 100$. Then, the learning rate is decreased to 0.01 at $100 \leq \text{epoch} < 150$. In addition, the learning rate is set to 0.001 at $100 \leq \text{epoch} \leq 150$. For fine-grained categorization tasks, we set the mini-batch size as 32, and the size of input images is 448×448. The initial learning rates on CUB-200-2011 and Stanford Dogs are set to be 0.01 and 0.001, respectively. The learning rates are divided by 10 after every 15 epochs, and total epochs are 100. Our approach is implemented with Pytorch and three Tesla P40 GPUs.

3) Choices of Hyperparameter λ: It can be seen from (10) that our formulation requires the choice of a hyperparameter λ which is important to keep the balance between the GLR and the CD. To obtain meaningful λ, we analyze it on CIFAR-100, TinyImageNet, CUB-200-2011, and Stanford Dogs datasets, respectively. As shown in Fig. 3(a), the ResNet-50, setting $\lambda = 0.9$ for CIFAR-100 dataset, obtains the best accuracy of 80.07%. The ResNet-18 with $\lambda = 0.8$ obtains the best accuracy of 78.03%. Meanwhile, as shown in Fig. 3(b), the ResNet-50 and ResNet-18, setting $\lambda = 0.9$ and $\lambda = 0.5$ for TinyImageNet dataset, obtain the best accuracy of 62.60% and 60.00%, respectively. From Fig. 3(a) and (b), our module is



(a) (b) (c) (d)

Fig. 3. Experimental results of ResNet-50 and ResNet-18 on the CIFAR-100, TinyImageNet, CUB-200-2011, and Stanford Dogs datasets. (a) CIFAR-100. (b) TinyImageNet. (c) CUB-200-2011. (d) Stanford Dogs.

TABLE I

Top-1 Error Rate (%) on Conventional Image Classification Tasks and Fine-Grained Visual Categorization Tasks. LR and GLR Denote With and Without the Incorrect Predictions. These Values Are Obtained by Averaging Three Runs. The Best Results Are Shown in Bold. "-" Denotes the Teacher Network

Model	Method	CIFAR-100	TinyImageNet	CUB	Dogs
ResNet-50	-	22.08	40.90	16.13	13.09
	LR [17]	20.97	39.00	15.10	12.23
	GLR (ours)	20.80	38.17	14.77	**12.17**
	GLR-CD (ours)	**20.53**	**37.40**	**14.67**	12.30
ResNet-18	-	24.15	42.23	19.80	19.02
	LR [17]	23.06	41.47	18.67	18.12
	GLR (ours)	22.70	40.50	18.03	18.03
	GLR-CD (ours)	**21.97**	**40.00**	**17.23**	**17.90**

always better than the baseline models on conventional image classification tasks. The baseline models are the ResNet trained using cross-entropy on these four datasets. In Fig. 3(c), setting $\lambda = 0.2$, we note that the ResNet-50 has the best accuracy of 85.33% on the CUB-200-2011. When $\lambda = 0.3$, the ResNet-18 model achieves the best accuracy on the CUB-200-2011 dataset. In the similar way, as $\lambda = 0.4$ and $\lambda = 0.1$ in Fig. 3(d), the ResNet-50 and the ResNet-18 can achieve the best accuracy on the Stanford Dogs dataset, respectively. In (10), the larger λ is, the larger the proportion of the GLR module is. As shown in Fig. 3(c) and (d), the general trend of our approach is that the accuracy decreases with the increase in λ. The reason may be that fine-grained visual categorization tasks require more accurate labels. In the following experiments, the value of λ is fixed.

4) Evaluation Metric: TP_c is the number of samples that are correctly predicted in the cth category. $c = 1, 2, \ldots, m$, where m is the number of categories. FP_c represents the number of samples that are misclassified into the cth category. FN_c means that the number of samples belonging to the cth category are incorrectly predicted as other categories.

1) *Top-1/5 Error Rate:* The top-k error rate is the percentage of test samples whose correct label is not in the top-k confidences. The Top-1 error rate can be denoted by

$$\mathrm{Top}-1\ \mathrm{error\ rate} = \frac{\sum_{c=1}^{m} \mathrm{FP}_c}{\sum_{c=1}^{m} \mathrm{TP}_c + \mathrm{FN}_c}. \tag{11}$$

2) *Recall at 1 (R@1):* Recall at 1 is the percentage of test samples that are correctly predicted to be the cth category to all samples that ground truths are cth. Then, it is defined as

$$\mathrm{R@1} = \frac{1}{m}\sum_{c=1}^{m} \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FN}_c}. \tag{12}$$

3) *Precision:* The precision is the proportion of test samples that are correctly predicted to be the cth category to all samples that are predicted to be the cth category. It can be formulated as

$$\mathrm{Precision} = \frac{1}{m}\sum_{c=1}^{m} \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FP}_c}. \tag{13}$$

4) *F_1 score:* It is the harmonic mean of the precision and the recall. The F_1 score is defined as

$$\begin{aligned}\frac{2}{F_1} &= \frac{1}{m}\sum_{c=1}^{m} \frac{2\mathrm{Precision}_c \times \mathrm{R@1}_c}{\mathrm{Precision}_c + \mathrm{R@1}_c}\\ &= \frac{1}{m}\sum_{c=1}^{m} \frac{2\mathrm{TP}_c}{2\mathrm{TP}_c + \mathrm{FP}_c + \mathrm{FN}_c}.\end{aligned} \tag{14}$$

B. Ablation Studies

To fully investigate our approach, we conduct a series of ablation experiments on different key components. All ablation studies are done on CIFAR100, TinyImageNet, CUB-200-2011, and Stanford Dogs. We evaluate the influence of the following designs: GLR and CD. The results are analyzed in detail in the following.

1) Effectiveness of Guided Label Refinery: The effectiveness of our GLR is studied in Table I. As shown in Table I, the GLR achieves a better performance than that of the LR on four datasets. The reason is that the GLR module only utilizes the correct predictions and ignores the incorrect

TABLE II

Top-1 Error Rate (%), Recall at 1 (R@1) Rate (%), Precision(%), and F_1 Score(%) on Conventional Image Classification Tasks and Fine-Grained Visual Categorization Tasks. The Best Results Are Shown in Bold. These Values Are Obtained by Averaging Three Runs. "-" Denotes the Teacher Network

Measure	Model	Method	CIFAR-100	TinyImageNet	CUB	Dogs
Top-1 error rate	ResNet-18	-	24.15	42.23	19.80	19.02
		AdaCos [49]	39.57	60.17	22.90	33.63
		Label Smoothing [23]	39.00	58.20	17.70	18.08
		Cross-entropy [50]	22.70	43.07	18.10	18.10
		LR [17]	23.06	41.47	18.67	18.12
		GLR-CD (ours)	**21.97**	**40.00**	**17.23**	**17.90**
	ResNet-50	-	22.08	40.90	16.13	13.09
		AdaCos [49]	38.30	57.13	21.43	21.60
		Label Smoothing [23]	37.19	55.83	14.93	12.80
		Cross-entropy [50]	21.20	40.07	15.30	12.30
		LR [17]	20.97	39.00	15.10	**12.23**
		GLR-CD (ours)	**20.53**	**37.40**	**14.67**	12.30
R@1	ResNet-18	-	60.02	32.12	67.03	66.00
		AdaCos [49]	60.98	33.15	69.27	67.18
		Label Smoothing [23]	61.41	34.67	72.38	72.57
		Cross-entropy [50]	69.50	39.00	73.40	74.57
		LR [17]	71.93	44.60	74.51	74.64
		GLR-CD (ours)	**72.00**	**45.07**	**75.87**	**75.30**
	ResNet-50	-	60.00	39.18	67.26	67.28
		AdaCos [49]	61.07	40.36	68.39	68.53
		Label Smoothing [23]	62.70	42.31	75.68	77.76
		Cross-entropy [50]	72.80	42.23	78.00	**83.67**
		LR [17]	71.03	47.68	78.29	82.11
		GLR-CD (ours)	**75.47**	**52.07**	**79.30**	**83.67**
Precision	ResNet-18	-	59.47	42.69	75.06	65.22
		AdaCos [49]	60.65	44.07	77.39	67.98
		Label Smoothing [23]	61.14	44.91	82.58	82.16
		Cross-entropy [50]	77.71	45.26	82.30	82.12
		LR [17]	77.50	45.32	82.75	82.56
		GLR-CD (ours)	**77.97**	**52.28**	**83.37**	**83.40**
	ResNet-50	-	59.87	39.60	77.80	77.53
		AdaCos [49]	61.41	40.53	79.18	79.95
		Label Smoothing [23]	62.66	42.26	86.00	87.83
		Cross-entropy [50]	78.20	46.03	85.35	87.23
		LR [17]	78.61	46.79	85.64	87.23
		GLR-CD (ours)	**79.76**	**53.00**	**86.26**	**88.60**
F_1 score	ResNet-18	-	59.17	42.50	75.22	65.29
		AdaCos [49]	61.56	43.40	77.81	67.94
		Label Smoothing [23]	61.11	44.29	82.18	82.19
		Cross-entropy [50]	77.50	45.77	82.60	81.11
		LR [17]	77.05	47.69	82.68	81.15
		GLR-CD (ours)	**79.05**	**48.25**	**83.02**	**82.72**
	ResNet-50	-	59.93	43.08	77.00	77.69
		AdaCos [49]	61.55	44.86	79.02	79.03
		Label Smoothing [23]	62.70	44.77	84.57	87.57
		Cross-entropy [50]	78.80	46.84	84.78	87.07
		LR [17]	78.91	48.13	84.86	87.22
		GLR-CD (ours)	**79.43**	**48.90**	**85.40**	**87.61**

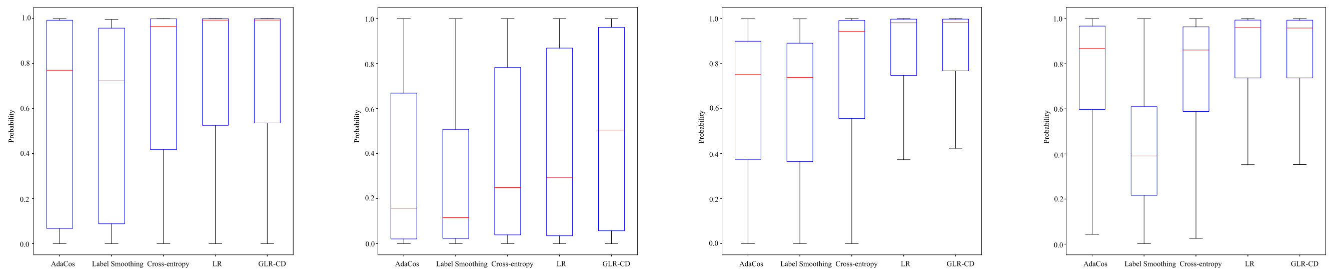
predictions. Therefore, GLR can effectively improve the performance of the ResNet-50 and the ResNet-18 on CIFAR-100, TinyImageNet, CUB-200-2011, and Stanford Dogs datasets.

2) Effectiveness of Channel Distillation: One can expect that the channel information of the teacher network can guide the student network to learn more discriminative features. To verify this, we utilize the CD to train the student module on CIFAR-100, TinyImageNet, CUB-200-2011, and Stanford Dogs. As shown in Table I, on these four datasets, our method (GLR-CD) applies the GLR module and the CD module in ResNet-50 and ResNet-18, which surpasses the GLR. The reason is that networks can learn more detailed features through the CD module. These detailed features are more beneficial for image classification. Therefore, CD can effectively improve the performance of models on the basis of the GLR.

C. Quantitative Analysis

1) Comparison With Regularization Methods: We expect that our module can capture more discriminative features and improve performance. To verify this, we measure the top-1 error rate of the GLR-CD by comparing with AdaCos [49], Label Smoothing [23], Cross-entropy [50], and LR [17] on CIFAR-100, TinyImageNet, CUB-200-2011, and Stanford Dogs.

1) *AdaCos [49]:* It is an adaptive cosine-based loss function, which is hyperparameter-free and generates



(a) (b) (c) (d)

Fig. 4. Statistical box plots of ResNet-50 on the CIFAR-100, TinyImageNet, CUB-200-2011, and Stanford Dogs datasets. The horizontal axis represents regularization methods: AdaCos, Label Smoothing, Cross-entropy, LR, and GLR-CD (our method). The vertical axis is the probability that each sample belongs to the ground truth. (a) CIFAR-100. (b) TinyImageNet. (c) CUB-200-2011. (d) Stanford Dogs.

a stronger supervision during the training process. To make the predicted class probability satisfy the semantic meaning of cosine similarities, AdaCos dynamically scales the cosine similarities between training samples and the corresponding class center vectors. In addition, AdaCos can make networks converge faster and more stable during the training process.

2) *Label Smoothing [23]:* Label Smoothing utilizes soft labels that are a weighted average of the hard labels and the uniform distribution. It significantly improves the generalization and learning speed of the network.

3) *Cross-Entropy [50]:* Cross-entropy trains networks by reducing the difference between the predicted values and the target values.

In Table II, it is observed that the GLR-CD outperforms other regularization methods on the top-1 error rate. In particular, on ResNet-50, the GLR-CD improves the top-1 error rate of the AdaCos from 38.30% to 20.53% on the CIFAR-100 dataset. Compared with AdaCos, on CIFAR-100, the GLR-CD obtains improvement of 17.60% over the ResNet-18. We also find that the top-1 error rates of the GLR-CD are often better than other methods, e.g., AdaCos, label smoothing, cross-entropy, and LR. These results indicate that our method can produce better predictive distributions than other regularization methods. Meanwhile, in Table II, we also describe quantitative results on the Recall at 1 (R@1), Precision, and F_1 score. As shown in Table II, the R@1, precision, and F_1 score of GLR-CD outperform other regularization methods on these four datasets. These further demonstrate that our method can obtain better predictive distribution than other regularization methods.

In Fig. 4, we use statistical box plots to facilitate the comparison of our approach with existing methods. We observe that the median values of GLR-CD are close to the upper edges on the CIFAR-100, CUB-200-2011, and Stanford Dogs datasets. These samples between the median value and the upper edge account for 50% of all samples. It means that the probability of these samples being predicted to the ground truth is close to the upper edge which is close to 1. In addition, on TinyImageNet, the median value of our approach is larger than that of other methods. Meanwhile, the lower quartiles of our approach are larger than those of other regularization methods on these four datasets.

TABLE III

Results of Estimating Intraclass Variance and Interclass Distance by Our Trained Cross-Entropy and GLR-CD on the CIFAR-100, TinyImageNet, CUB-200-2011, and Stanford Dogs Datasets. Var Is Variance and Dis Is Distance

	models	CIFAR-100	TinyImageNet	CUB	Dogs
Var	Cross-entropy	2.44	0.81	0.05	0.04
	GLR-CD	**1.68**	**0.69**	**0.03**	**0.03**
Dis	Cross-entropy	3.86	1.03	0.56	0.76
	GLR-CD	**5.11**	**1.75**	**1.24**	**1.76**

2) Feature Embedding Analysis: One can expect that the intraclass variations and the interclass similarities can be reduced by utilizing the GLR-CD to produce meaningful predictions. To analyze the changes of intraclass variation and interclass similarity, we obtain feature vectors of the penultimate layer of ResNet-50 and mean feature vectors of each class. The feature vectors of images are used to calculate the intraclass variance. In addition, mean feature vectors are used to calculate the interclass similarity that adopts the Euclidean distance. As can be observed in Table III, our module can obtain smaller intraclass variations and larger interclass distances. In other words, our module can increase intraclass similarities and decrease the interclass similarity. Meanwhile, in Table II, we describe quantitative results on the metric Recall at 1 (R@1) in Section IV-A. R@1 is inversely proportional to intraclass variations. In other words, the larger the value of R@1 is, the smaller the intraclass variations are. As shown in Table II, on CIFAR-100, TinyImageNet, CUB-200-2011, and Stanford Dogs datasets, R@1 values can be significantly improved when ResNet-50 and ResNet-18 are trained by our module. These further demonstrate that our module can decrease the intraclass variations.

3) The Scalability of Our Approach: In order to verify the scalability of our approach, we have evaluated our approach on these four datasets with various architectures, such as ResNet, DenseNet, and EfficientNet. As reported in Table IV, for ResNet-18, the top-1 error rates of our module are mostly smaller than those of the cross-entropy loss. As shown in

TABLE IV

TOP-1 ERROR RATE (%) ON CONVENTIONAL IMAGE CLASSIFICATION TASKS AND FINE-GRAINED VISUAL CATEGORIZATION TASKS. THE BEST RESULTS ARE SHOWN IN BOLD. THESE VALUES ARE OBTAINED BY AVERAGING THREE RUNS. THE PARAMS DENOTES THE NUMBER OF PARAMETERS

Model	Method	Top-1 error rate				Params
		CIFAR-100	TinyImageNet	CUB	Dogs	
ResNet-18	Cross-entropy	22.70	43.07	18.10	18.10	14M
	GLR-CD (ours)	**21.97**	**40.00**	**17.23**	**17.90**	20M
ResNet-50	Cross-entropy	21.20	40.07	15.30	**12.30**	22M
	GLR-CD (ours)	**20.53**	**37.40**	**14.67**	**12.30**	33M
DenseNet-161	Cross-entropy	19.70	39.50	13.57	11.87	14M
	GLR-CD (ours)	**17.82**	**37.50**	**12.58**	**11.37**	20M
EfficientNet-B0	Cross-entropy	24.38	36.90	15.72	16.56	4.8M
	GLR-CD (ours)	**23.18**	**35.68**	**14.22**	**15.87**	6.3M
EfficientNet-B1	Cross-entropy	24.59	35.93	13.86	11.59	7.5M
	GLR-CD (ours)	**22.83**	**34.02**	**12.07**	**10.26**	9.3M
EfficientNet-B2	Cross-entropy	24.24	35.78	14.00	10.45	9.3M
	GLR-CD (ours)	**23.07**	**33.92**	**11.64**	**9.86**	12.7M

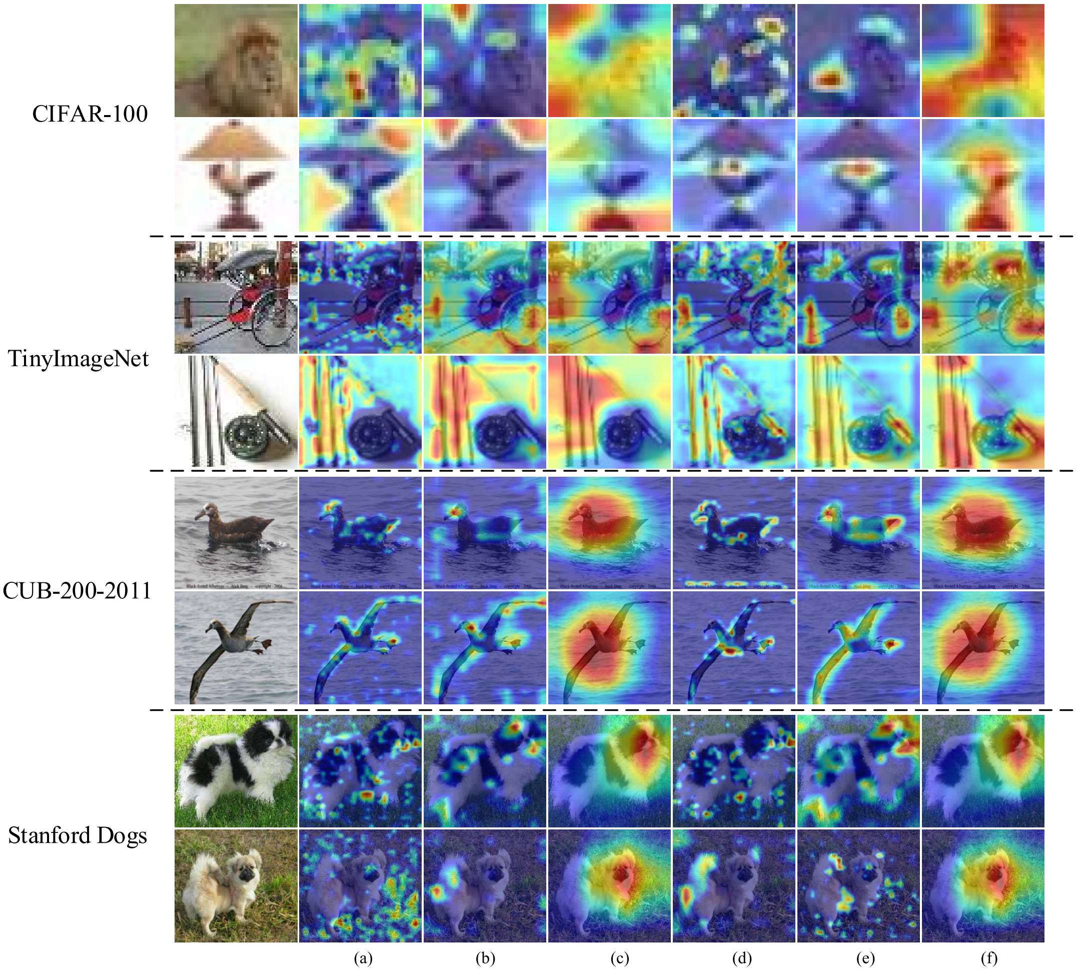


Fig. 5. Activation maps on the CIFAR-100, TinyImageNet, CUB-200-2011, and Stanford Dogs datasets with the ResNet-50 as the base model. There is visualization of the convolution layer from the second to the fourth layer of the baseline model in columns (a)–(c). There is visualization of the convolution layer from the second to the fourth layer of our model in columns (d)–(f).

Table IV, for ResNet-50, the top-1 error rates of GLR-CD are also smaller than those of the cross-entropy loss. In particular, for ResNet-50, GLR-CD improves the top-1 error rate from 40.07% to 37.40% on the TinyImageNet dataset. Meanwhile, on DenseNet-161, EfficientNet B0, EfficientNet B1, and EfficientNet B2, the top-1 error rates of GLR-CD are also smaller than those of the cross-entropy loss. Meanwhile, we introduce the number of parameters in Table IV. On different models, the GLR-CD has more parameters than the cross-entropy. However, the parameters of the GLR-CD are only increased by 20%–35% of the parameters of the cross-entropy. As mentioned earlier, our module can easily be extended to many architectures with the addition of a small number of parameters.

TABLE V
ACCURACY (%) ON CIFAR-100 AND TINYIMAGENET DATASETS

Methods	CIFAR-100	TinyImageNet
DenseNet-161 [2]	80.30	60.50
DenseNet-161-**GLR-CD**	**82.18**	62.50
EfficientNet-B0 [3]	79.20	63.10
EfficientNet-B0-**GLR-CD**	80.82	64.32
EfficientNet-B1 [3]	78.33	64.07
EfficientNet-B1-**GLR-CD**	79.37	65.98
EfficientNet-B2 [3]	78.37	64.22
EfficientNet-B2-**GLR-CD**	79.00	**66.08**

TABLE VI
ACCURACY (%) ON CUB-200-2011 DATASET

Methods	CUB
MaxEnt [56]	86.60
PC [57]	86.87
MC-Loss [58]	87.30
DFL-CNN [59]	87.40
NTS-Net [60]	87.50
Cross-X [48]	87.70
DCL [54]	87.80
DBTNet [53]	88.10
ACNet [52]	88.10
CIN [51]	88.10
EfficientNet-B2-**GLR-CD**	**88.36**

TABLE VII
ACCURACY (%) ON STANFORD DOGS DATASET

Methods	Dogs
MaxEnt [56]	83.60
PC [57]	83.75
FDL [61]	84.90
DRIFT [62]	87.30
Muti Scale [63]	87.70
SEF [64]	88.80
Cross-X [48]	88.90
FCAN [55]	88.90
EfficientNet-B2-**GLR-CD**	**90.14**

TABLE VIII
NUMBERS OF WRONG SAMPLES ARE REDUCED BY OUR APPROACH

Backbone	CIFAR-100	TinyImageNet	CUB	Dogs
ResNet-18	73	307	50	17
ResNet-50	67	267	37	2
DenseNet-161	188	200	57	43
EfficientNet-B0	120	122	87	59
EfficientNet-B1	176	191	103	86
EfficientNet-B2	117	186	137	51

D. Comparison With State-of-the-Art Methods

1) CIFAR-100 and TinyImageNet: To improve the performance of conventional image classification, many network architectures such as DenseNet-161 and EfficientNet are proposed. As shown in Table V, on DenseNet-161, our module obtains the state-of-the-art accuracy for CIFAR-100. Meanwhile, on TinyImageNet, our module achieves state-of-the-art performance with EfficientNet-B2 as the base model.

2) CUB-200-2011 and Stanford Dogs Datasets: In Table VI, it is observed that our module achieves the best performance on CUB-200-2011. Compared with the CIN [51], ACNet [52], and DBTNet [53], we obtain improvements of 0.26%. Meanwhile, as expected, our module obtains larger gains in classification accuracy on CUB-200-2011 datasets as compared with the other state-of-the-art methods, such as DCL [54] and Cross-X [48].

The results of experiments on Stanford Dogs are shown in Table VII. In addition, our module exhibits a better performance on Stanford Dogs. Our module achieves 90.14% accuracy which outperforms state-of-the-art methods (FCAN [55] and Cross-X [48]) by 1.24%.

From aforementioned results, it can be seen that our method is not only effective on conventional image classification tasks but also on fine-grained visual categorization tasks. This indicates that our method has a good generalization performance.

E. Qualitative Analysis

To show the advantages of our approach, we utilize the gradient-weighted class activation mapping (Grad-CAM) [65] to visualize the convolution layers of both the baseline model and our approach. Grad-CAM is a heatmap for input image. It is a 2-D score grid related to a specific output category. In addition, each position of the grid indicates the importance for the specific output category. For example, an image is input into the DNN model and classified as a specific category. In addition, the Grad-CAM uses a heatmap to show the degree of similarity between each position in the image and the specific category. Therefore, the heatmap can help to understand the importance of local locations in an original image for the final classification decision.

Fig. 5(a)–(c) is visualization of the convolution layers from the second to the fourth stage of the ResNet-50. Fig. 5(d)–(f) is visualization of the convolution layers from the second to the fourth stage of our model. The brighter the region, the more important it is for classification. In Fig. 5, we compare the activation map of the second stage of the ResNet-50 with that of the second stage of our model, i.e., (a) versus (d). In Fig. 4(a), the ResNet-50 focuses more on backgrounds. In Fig. 4(d), our module pays more attention to objects, such as tigers, birds, and dogs. To obtain a better performance, we should focus on objects instead of backgrounds. In the second stage, our module can focus more on objects. In addition, when it comes to the third stage, namely (b) versus (e), our model pays more attention to parts of the object, such as beaks, wings, and tail. Comparing (c) with (f), our brightest parts are closer to the center of objects. The visualization results demonstrate that our approach can concentrate on more discriminative parts. These indicate that our approach can force the model to learn more discriminative features. Therefore, our module can improve the performance of image classification.

To explore the qualitative analysis of the JGL, we visualize the top-5 softmax scores of some testing samples. In the first row of Fig. 6, the image is misclassified by the ResNet-50. However, it can be correctly classified by our module. In the second row, our module can increase the difference between the ground truth and other categories. In addition, in the third row, the image is misclassified by ResNet-50 and our module.

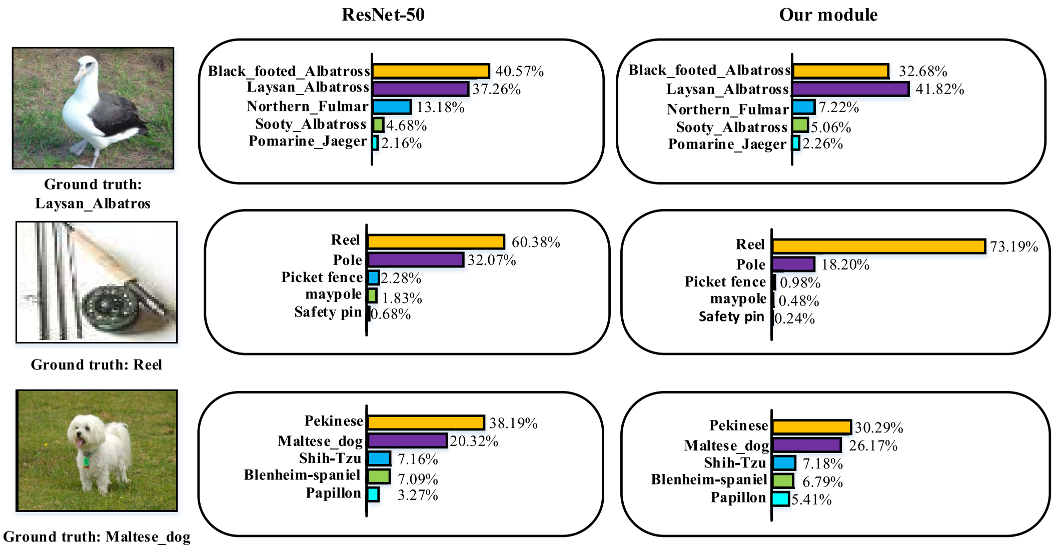


Fig. 6. Top-5 softmax scores on testing samples.

However, our module improves its probability in the ground truth.

Meanwhile, we make an experimental study to show how many wrong samples are reduced by our approach. In Table VIII, our module can reduce wrong samples on CIFAR-100, TinyImageNet, CUB-200-2011, and Stanford Dogs. These results once again prove the effectiveness of our method.

V. Conclusion

In this work, we propose a JGL approach to guide the optimization of the student model. Our approach consists of the GLR module and the CD module. First, we propose the GLR module, which only utilizes the correct predictive labels and removes the incorrect predictive labels. We remark that our idea promotes models to produce more effective predictions. Moreover, we propose the CD to make full use of the features of the teacher model to transfer the knowledge to the student model. Extensive experiments on conventional image classification tasks and fine-grained visual categorization tasks exhibit the effective performances of our approach. In addition, our approach does not require bounding box or part information. Meanwhile, it is easy to implement, and does not need to add excessive parameters during training.

Since the CD transfers both useful and useless features to the student network. In the future, we will study on attention maps of transferring knowledge from the teacher network to the student network. The attention maps only pay attention to useful features. Moreover, we will explore to integrate attention maps into the GLR in an efficient way, to further leverage features.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[2] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.

[3] M. Tan and Q. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.

[4] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, *arXiv:1409.1556*.

[5] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, *arXiv:1412.6980*.

[6] L. Xie, J. Wang, Z. Wei, M. Wang, and Q. Tian, “DisturbLabel: Regularizing CNN on the loss layer,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4753–4762.

[7] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 25, Dec. 2012, pp. 1097–1105.

[9] J. Wang and L. Perez, “The effectiveness of data augmentation in image classification using deep learning,” *Convolutional Neural Netw. Vis. Recognit*, vol. 11, pp. 1–8, Dec. 2017.

[10] S. C. Wong, A. Gatt, V. Stamatescu, and M. D. McDonnell, “Understanding data augmentation for classification: When to warp?” in *Proc. Int. Conf. Digit. Image Comput., Techn. Appl. (DICTA)*, Nov. 2016, pp. 1–6.

[11] Y. Xu *et al.*, “Improved relation classification by deep recurrent neural networks with data augmentation,” 2016, *arXiv:1601.03651*.

[12] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, “Learning from simulated and unsupervised images through adversarial training,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2107–2116.

[13] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, “Deep mutual learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4320–4328.

[14] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, “Learning transferable architectures for scalable image recognition,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8697–8710.

[15] 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.

[16] Y. Huang *et al.*, “GPipe: Efficient training of giant neural networks using pipeline parallelism,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 103–112.

[17] H. Bagherinezhad, M. Horton, M. Rastegari, and A. Farhadi, “Label refinery: Improving ImageNet classification through label progression,” 2018, *arXiv:1805.02641*.

[18] J. Tang *et al.*, “Understanding and improving knowledge distillation,” 2020, *arXiv:2002.03532*.

[19] A. Krizhevsky and G. Hinton, “Learning multiple layers of features from tiny images,” in *Handbook of Systemic Autoimmune Diseases*, vol. 1. 2009.

[20] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, “The Caltech-UCSD birds-200-2011 dataset,” Comput. Neural Syst., California Inst. Technol., Pasadena, CA, USA, Tech. Rep. CNS-TR, 2011.

[21] 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.

[22] 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. (CVPR)*, Jun. 2016, pp. 2818–2826.

[23] R. Müller, S. Kornblith, and G. E. Hinton, "When does label smoothing help?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 4694–4703.

[24] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[25] G. Urban *et al.*, "Do deep convolutional nets really need to be deep and convolutional?" 2016, *arXiv:1603.05691*.

[26] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "FitNets: Hints for thin deep nets," 2014, *arXiv:1412.6550*.

[27] T. Chen, I. Goodfellow, and J. Shlens, "Net2Net: Accelerating learning via knowledge transfer," 2015, *arXiv:1511.05641*.

[28] G. Pereyra, G. Tucker, J. Chorowski, Ł. Kaiser, and G. Hinton, "Regularizing neural networks by penalizing confident output distributions," 2017, *arXiv:1701.06548*.

[29] C. Yang, L. Xie, S. Qiao, and A. L. Yuille, "Training deep neural networks in generations: A more tolerant teacher educates better students," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 5628–5635.

[30] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1195–1204.

[31] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4133–4141.

[32] T. Furlanello, Z. C. Lipton, M. Tschannen, L. Itti, and A. Anandkumar, "Born again neural networks," 2018, *arXiv:1805.04770*.

[33] C. Yang, L. Xie, C. Su, and A. L. Yuille, "Snapshot distillation: Teacher-student optimization in one generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2859–2868.

[34] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy Student improves ImageNet classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10687–10698.

[35] Y. Cui, Y. An, W. Sun, H. Hu, and X. Song, "Lightweight attention module for deep learning on classification and segmentation of 3-D point clouds," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–12, 2020.

[36] X. Cheng and J. Yu, "RetinaNet with difference channel attention and adaptively spatial feature fusion for steel surface defect detection," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021.

[37] B. Su, H. Chen, K. Liu, and W. Liu, "RCAG-Net: Residual channelwise attention gate network for hot spot defect detection of photovoltaic farms," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–14, 2021.

[38] G. Wang, Y. Wang, and X. Sun, "Multi-instance deep learning based on attention mechanism for failure prediction of unlabeled hard disk drives," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–9, 2021.

[39] Z. Zhou, C. Zhuge, X. Guan, and W. Liu, "Channel distillation: Channel-wise attention for knowledge distillation," 2020, *arXiv:2006.01683*.

[40] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[41] H. Zhao *et al.*, "PSANet: Point-wise spatial attention network for scene parsing," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 267–283.

[42] J. Hu, L. Shen, S. Albanie, G. Sun, and A. Vedaldi, "Gather-excite: Exploiting feature context in convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 9401–9411.

[43] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "GCNet: Non-local networks meet squeeze-excitation networks and beyond," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 1–10.

[44] X. Jiang *et al.*, "Attention scaling for crowd counting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4706–4715.

[45] H. Zhao, J. Jia, and V. Koltun, "Exploring self-attention for image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10076–10085.

[46] L. Chi, Z. Yuan, Y. Mu, and C. Wang, "Non-local neural networks with grouped bilinear attentional transforms," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11804–11813.

[47] S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich, "Training deep neural networks on noisy labels with bootstrapping," 2014, *arXiv:1412.6596*.

[48] W. Luo *et al.*, "Cross-X learning for fine-grained visual categorization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8242–8251.

[49] X. Zhang, R. Zhao, Y. Qiao, X. Wang, and H. Li, "AdaCos: Adaptively scaling cosine logits for effectively learning deep face representations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10823–10832.

[50] L.-Y. Deng, "The cross-entropy method: A unified approach to combinatorial optimization, Monte-Carlo simulation, and machine learning," *Technometrics*, vol. 48, no. 1, pp. 147–148, Feb. 2006.

[51] Y. Gao, X. Han, X. Wang, W. Huang, and M. Scott, "Channel interaction networks for fine-grained image categorization," in *Proc. AAAI*, 2020, vol. 34, no. 7, pp. 10818–10825.

[52] R. Ji *et al.*, "Attention convolutional binary neural tree for fine-grained visual categorization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10468–10477.

[53] H. Zheng, J. Fu, Z.-J. Zha, and J. Luo, "Learning deep bilinear transformation for fine-grained image representation," 2019, *arXiv:1911.03621*.

[54] 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.

[55] 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*.

[56] A. Dubey, O. Gupta, R. Raskar, and N. Naik, "Maximum-entropy fine-grained classification," 2018, *arXiv:1809.05934*.

[57] A. Dubey, O. Gupta, P. Guo, R. Raskar, R. Farrell, and N. Naik, "Pairwise confusion for fine-grained visual classification," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 70–86.

[58] D. Chang *et al.*, "The devil is in the channels: Mutual-channel loss for fine-grained image classification," *IEEE Trans. Image Process.*, vol. 29, pp. 4683–4695, 2020.

[59] 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.

[60] 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. (ECCV)*, 2018, pp. 420–435.

[61] C. Liu, H. Xie, Z.-J. Zha, L. Ma, L. Yu, and Y. Zhang, "Filtration and distillation: Enhancing region attention for fine-grained visual categorization," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 11555–11562.

[62] H. Wang, V. Saligrama, S. Sclaroff, and V. Ablavsky, "Cost-aware fine-grained recognition for IoTs based on sequential fixations," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1252–1261.

[63] G. Sun, H. Cholakkal, S. Khan, F. Shahbaz Khan, and L. Shao, "Fine-grained recognition: Accounting for subtle differences between similar classes," 2019, *arXiv:1912.06842*.

[64] W. Luo, H. Zhang, J. Li, and X.-S. Wei, "Learning semantically enhanced feature for fine-grained image classification," *IEEE Signal Process. Lett.*, vol. 27, pp. 1545–1549, 2020.

[65] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

Peipei Zhao received the M.E. degree from the School of Computer Science and Technology, Xidian University, Xi'an, China, in 2016, where she is currently pursuing the Ph.D. degree.

Her research interests include computer vision, deep learning, pattern recognition, and image classification.

Hang Yao received the B.S. degree in computer science and technology from Xidian University, Xi'an, China, in 2020, where he is currently pursuing the M.S. degree with the School of Computer Science and Technology.

His research interests include computer vision, deep learning, and fine-grained classification.

Xiangzeng Liu received the M.S. and Ph.D. degrees in applied mathematics from Northwestern Polytechnical University, Xi'an, China, in 2008 and 2011, respectively.

In 2012, he joined as an Advanced Image Processing Engineer with the Xi'an Microelectronics Technology Institute, Xi'an. He is currently an Associate Professor with the School of Computer Science and Technology, Xidian University, Xi'an. His current research interests include image registration, image enhancement, object detection and recognition, computer vision, and machine learning.

Ruyi Liu received the Ph.D. degree from the School of Computer and Technology, Xidian University, Xi'an, China, in 2018.

She is currently working as a Lecturer with the School of Computer Science and Technology, Xidian University. Her current interests include image classification and segmentation, and computer vision methods with applications in remote sensing.

Qiguang Miao (Senior Member, IEEE) received the Ph.D. degree in computer application technology from Xidian University, Xi'an, China, in December 2005.

He is currently a Professor and a Ph.D. Student Supervisor with the School of Computer Science and Technology, Xidian University. In recent years, he has published over 100 articles in the significant domestic and international journals or conferences. His research interests include machine learning, intelligent image processing, and malware behavior analysis and understanding.