Exploiting Web Images for Fine-Grained Visual Recognition by Eliminating Open-Set Noise and Utilizing Hard Examples

Huafeng Liu, Chuanyi Zhang, Yazhou Yao, Xiu-Shen Wei, Fumin Shen, Zhenmin Tang, and Jian Zhang

***Abstract*—Labeling objects at a subordinate level typically requires expert knowledge, which is not always available when using random annotators. As such, learning directly from web images for fine-grained recognition has attracted broad attention. However, the presence of label noise and hard examples in web images are two obstacles for training robust fine-grained recognition models. Therefore, in this paper, we propose a novel approach for removing irrelevant samples from real-world web images during training, while employing useful hard examples to update the network. Thus, our approach can alleviate the harmful effects of irrelevant noisy web images and hard examples to achieve better performance. Extensive experiments on three commonly used fine-grained datasets demonstrate that our approach is far superior to current state-of-the-art web-supervised methods. The data and source code of this work have been made publicly available at: https://github.com/NUST-Machine-Intelligence-Laboratory/Advanced-Softly-Update-Drop.**

***Index Terms*—Noisy web images, robust learning, fine-grained recognition.**

I. Introduction

Deep neural networks (DNNs) have achieved impressive results on many computer vision tasks (*e.g.*, image retrieval [1], [2], [3], [4], image classification [5], [6], [7], [8], [9], image segmentation [10], [13], [14], [15], [16]) due to the availability of large-scale image datasets, such as ImageNet [17]. However, fine-grained visual classification (FGVC) remains challenging [18]. Dividing a category into subclasses exponentially increases the required number of labels, making it a labor-intensive and time-consuming task. Moreover, fine-grained annotation usually requires domain-specific expert knowledge, which exacerbates the labeling problem. To reduce the cost of manual labeling, several works have focused on the semi-supervised paradigm [19], [20], [21]. However, these works still inevitably involve some form of human intervention and thus remain labor-intensive.

In contrast to manually labeled image datasets, web images are a rich and free resource [11], [12], [22], [23]. For arbitrary categories, potential training data can easily be obtained from image search engines like Google or Bing. Therefore, it is intuitive to directly leverage web images to train fine-grained classification models. Unfortunately, due to the error index of image search engines, the precision of returned images is still unsatisfactory. For example, the average precision of the top 1,000 images for 18 categories from the Google Image Search Engine is merely 32% [24].

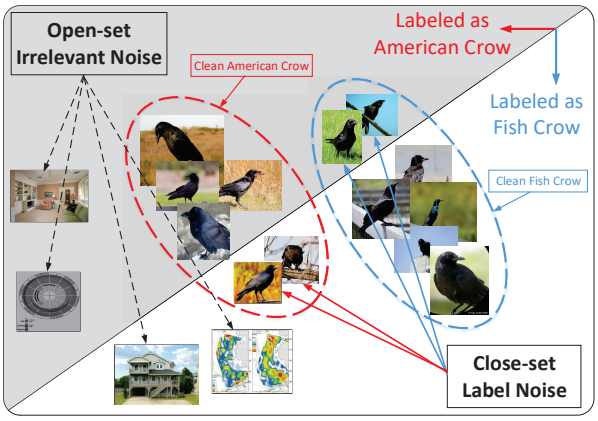


Fig. 1: An illustration of the close-set and open-set noisy web images for a bird dataset. Specifically, noisy images in the close-set have their true labels in the dataset. Open-set contains irrelevant noisy images, whose labels are outside of the dataset.

As shown in Fig. 1, noisy web images within fine-grained categories can be divided into two groups: close-set and open-set noise. Specifically, for noisy images in the close-set, their true labels can be found in the given dataset (*e.g.*, the "fish crow" images are mistakenly labeled as "American crow" in Fig. 1). The open-set, in contrast, consists of irrelevant noisy images, whose labels fall outside of the dataset (*e.g.*, the images indicated by the black dotted arrows in Fig. 1). As DNNs have a high capacity to fit noisy data [25], [26], training fine-grained recognition models directly with these noisy web images tends to result in poor performance.

To reduce the harmful influence of noise, some works have concentrated on estimating the noise transition matrix. For example, [27] leveraged a bootstrapping loss and assigned a weight to the current prediction to compensate for the erroneous guidance of noisy samples. As an extension, [29] introduced an additional softmax layer to estimate the label noise transition matrix. However, recovering the exact noise transition matrix is difficult. Alternatively, another line of works have chosen to focus on the sample section mecha-

H. Liu, C. Zhang, Y. Yao, X. Wei, and Z. Tang are with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. H. Liu and C. Zhang are the co-first authors, corresponding author: Yazhou Yao, Email: Yazhou.Yao@njust.edu.cn.

F. Shen is with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China.

J. Zhang is with the Global Big Data Technologies Center, University of Technology Sydney, Australia.

nism, which aims to separate clean samples from noise. The representative works in this area include Decoupling [30] and Co-teaching [35]. These works identify clean samples within a mini-batch and use them to update the networks. Nevertheless, they assume a close-set noisy label setting.

Such a restricted assumption contradicts the fact that open-set scenarios are more common in practice. Furthermore, ignoring the existence of irrelevant noise makes the models less robust. To address this, with the increased popularity of web-supervised learning, [36], [37] instead focused on open-set scenarios to tackle irrelevant noise. Unfortunately, neither was designed for fine-grained visual classification.

In addition to noisy images, web datasets tend to contain many hard examples. For instance, some web images are corrupted by text or watermarks, while others may contain a small object with a large background. These hard examples rarely exist in labeled training sets. Compared with easy examples, the hard examples found in web datasets carry less useful foreground information and more irrelevant background knowledge. The irrelevant information can misguide the model and consequently degrade its robustness. This can become problematic when a model is directly trained using a large number of hard examples. However, if the hard examples are correctly leveraged, the performance of the model can actually be improved, without degrading the robustness.

Therefore, we propose an approach that removes irrelevant noise from the training set, while simultaneously employing useful hard examples during training. Our work is motivated by the following observations: (1) Soft labels contain more information than one-hot labels [40], especially for fine-grained visual classification, where subcategories share obvious similarities. (2) DNNs always memorize easy instances first, and gradually adapt to hard and noisy examples [25], [26]. (3) Label smoothing prevents overfitting by encouraging models to be less confident [41].

During training, our approach aims to identify irrelevant noisy samples and dynamically drop them. In addition, it effectively utilizes hard examples to increase the robustness of the model. Unlike most existing methods, which use the loss values to find noisy samples, our approach instead leverages the cross-entropy of the softmax probability across consecutive epochs which we refer to as probability cross-entropy from here on. Our proposed approach can make effective use of the information encoded in the soft labels and is able to measure any changes in network prediction. This is because open-set noisy samples are harder to fit than clean ones, so their predictions are unstable during training, resulting in a high probability of cross-entropy. Thus, irrelevant noisy samples can be distinguished from the useful training set and dropped by calculating the probability cross-entropy. In this way, the proposed approach can alleviate the harmful effect of open-set noise and achieve better performance. To guide the model to learn more efficiently from hard examples, normalization and label smoothing are utilized to boost the model training. Extensive experiments and ablation studies demonstrate that our approach outperforms state-of-the-art methods. The main contributions of this work can be summarized as follows:

1) We propose a novel approach to remove irrelevant noisy samples and utilize hard examples from web images to train an FGVC model. Unlike existing methods, our approach can dynamically increase the drop rate, enabling it to retain more instances during early epochs and increasingly drop noisy images before they are memorized.
2) Our proposed global sampling-based approach can effectively overcome the noise rate imbalance problem common in web images. Specifically, this problem occurs when the number of noisy images fluctuates among different batches, meaning that clean samples might be dropped in some mini-batches, while noisy samples are used for training in others.
3) Extensive experiments demonstrate that our approach outperforms state-of-the-art methods. Our learning paradigm delivers a new pipeline for fine-grained visual classification, which is more practical for real-world applications.

The rest of the paper is organized as follows: Section II discusses related works. We propose our framework and associated algorithms in Section III. The experimental evaluations and discussions are presented in Section IV. Section V concludes the paper.

II. Related Work

In this section, we review related works from three aspects: fine-grained visual classification, web-supervised learning, and normalization in neural networks.

A. Fine-Grained Visual Classification

The aim of fine-grained visual classification is to distinguish objects at a subordinate level. Taking into account the similarities between subcategories, early works typically trained the network to learn discriminative features using strong annotation cues, such as bounding boxes or part annotations [44], [45], [46], [47], [48]. Although they obtain promising results, these strongly supervised methods require heavy human annotation. To overcome this drawback, recent studies have focused on weakly supervised methods, which only require image-level labels [55], [56], [57], [58], [59], [60], [61], [62]. The state-of-the-art weakly supervised methods [59], [62] have shown competitive performance with strongly supervised methods. However, the label annotation still requires expert knowledge. This limits the dataset size. In addition, to mine discriminative features on small datasets, state-of-the-art weakly supervised methods tend to involve complex operations, such as part estimation [62] or context encoding [59]. To further improve the performance, several semi-supervised methods have managed to leverage widely available web data [20], [21], [63], [64], [65], [49], [50], [51] for FGVC. For example, Niu *et al.* [20] jointly utilized web data and well-labeled auxiliary data for FGVC, while Cui *et al.* [21] leveraged an iterative approach for dataset bootstrapping and model training. Nevertheless, these semi-supervised methods still require human intervention. Different from semi-supervised methods, our approach is a pure web-supervised approach, which requires no human intervention.

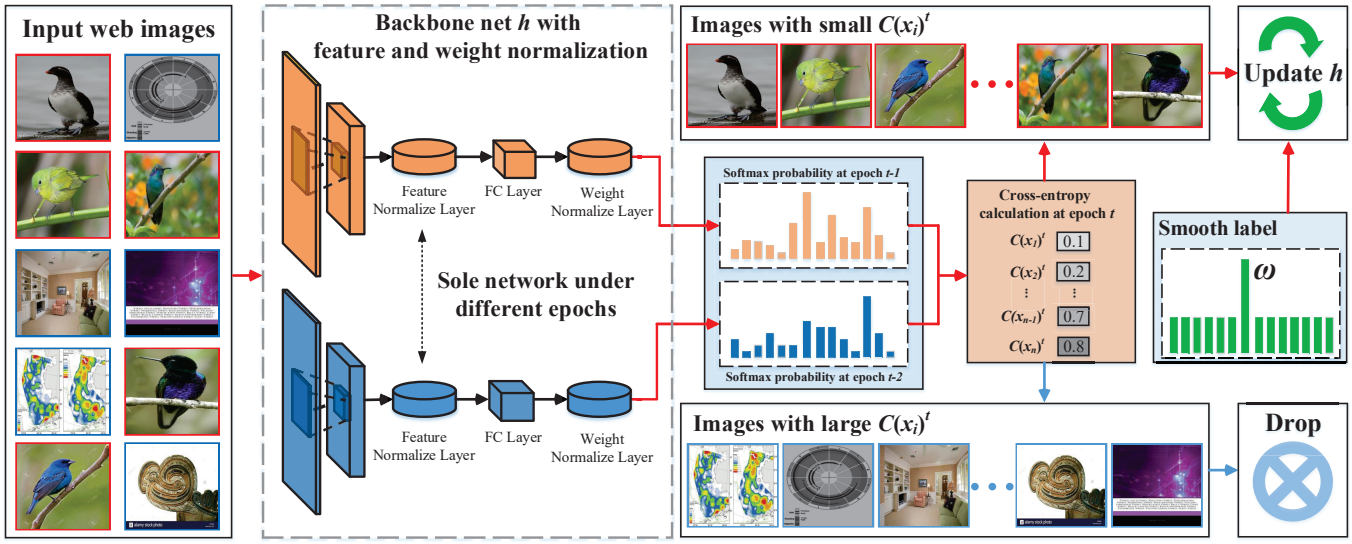


Fig. 2: The architecture of our model. For each input web image x_i, our approach first obtains the softmax probability of epoch $t-1$, $t-2$ as $\boldsymbol{p}(x_i)^{t-1}$ and $\boldsymbol{p}(x_i)^{t-2}$, respectively. Then it computes the cross-entropy $C(x_i)^t$ between $\boldsymbol{p}(x_i)^{t-2}$ and $\boldsymbol{p}(x_i)^{t-1}$ in epoch t. $C(x_i)^t$ is leveraged to supervise the separation of useful and irrelevant noisy web images. To be specific, images with large $C(x_i)^t$ are identified as irrelevant noisy images and then dropped during training. Those with small $C(x_i)^t$ are regarded as useful images and utilized to further update the network $\boldsymbol{h}$ with the smooth label, where the weight of the image label is ω.

B. Web-Supervised Learning

Training fine-grained recognition models with web images usually results in poor performance due to the presence of label noise and data bias [18]. Statistical learning has contributed significantly to solving this problem, especially in theoretical aspects [33], [34], [31], [28]. In this work, we focus on deep learning based approaches. Roughly speaking, these works can be separated into four groups. The first group involves developing novel loss functions [27], [53] to deal with label noise. The second group tries to estimate the noise transition matrix [54]. The third one applies attention mechanisms to alleviate noise and data bias [42]. The last group attempts to clean the web data as a preprocessing step [30], [35], [52]. However, none of these works are specifically designed for fine-grained visual recognition.

C. Normalization

Normalization has been widely used in person recognition [66] and face verification [38], [39]. For example, [38] utilized feature normalization to make the model pay more attention to blurry images. AM-softmax [39] further improved the effect of normalization by introducing an additive angular margin for the softmax loss. The additive angular margin makes the learned features more compact and therefore reduces the intra-class distance. Normalization significantly improves the performances of these methods in person recognition and face verification tasks. Inspired by these works, we utilize feature and weight normalization for the FGVC task.

III. The Proposed Approach

As shown in Fig. 2, our approach consists of three main steps: normalization, denoising, and hard example utilization. We provide details of each step as follows.

A. Normalization

Following [39], the feature f and weight W are normalized ($\|W_j\| = \|f_i\| = 1$) in the nonbias softmax loss:

$$\mathcal{L}_S(x_i, y_i) = -\log \frac{\mathrm{e}^{W_{y_i}^{\mathrm{T}} f_i}}{\sum_{j=1}^{M} \mathrm{e}^{W_j^{\mathrm{T}} f_i}}, \tag{1}$$

where x_i denotes the i-th sample with label y_i, f_i is the input feature of the last fully connected layer, W_j is the j -th column of the last fully connected layer, and M is the number of categories. Thus, we have

$$W_{y_i}^{\mathrm{T}} f_i = \|W_{y_i}\| \, \|f_i\| \cos\theta_{y_i} = \cos\theta_{y_i}, \tag{2}$$

where θ_{y_i} is the angle between vectors W_{y_i} and f_i. After normalization ($\|W_j\| = \|f_i\| = 1$), the network outputs the cosine distance between W_{y_i} and f_i. Then, the cosine values are scaled with a hyperparameter s and the normalized loss function becomes:

$$\mathcal{L}_N(x_i, y_i) = -\log \frac{\mathrm{e}^{s\cdot\cos\theta_{y_i}}}{\sum_{j=1}^{M} \mathrm{e}^{s\cdot\cos\theta_j}}, \tag{3}$$

where θ_j is the angle between vectors W_j and f_i. The scaling factor s is used to accelerate and stabilize the optimization. Wang *et al.* [67] pointed out that the network fails to converge without this scaling factor s. Following [39], [67], the value of s is set to 30.

Fig. 3 shows the feature distributions after normalization. As can be seen, the features are angularly distributed on a hypersphere, because their L_2-norms are scaled to 1. Accordingly, the last fully connected layer learns the center of

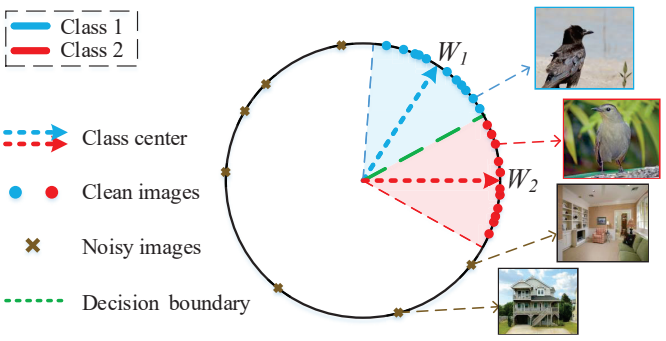


Fig. 3: After normalization, features are angularly distributed on a hypersphere, and the network learns the center of each class W_j during training.

each class W_j during training. Ranjan *et al.* [38] revealed that hard examples, especially blurry images, tend to have a lower L_2-norm. After feature normalization, features with small norms will obtain a much higher gradient compared to features that have large norms [39]. As a consequence, the model will pay more attention to these hard examples during back-propagation. In this way, feature normalization enables the model to learn more effectively from hard examples.

B. Denoising

Web-supervised learning is an open-set problem. Our approach first identifies and drops the irrelevant noisy images to purify the web training set.

1) Noise Identification: Memorization effects [25], [26] indicate that deep neural networks always fit easy examples in the initial epochs, and then gradually adapt to hard and noisy examples. Note that, irrelevant noisy samples are totally different from clean ones in the training set, with diverse patterns. As a result, the model struggles to fit to them. The prediction results of irrelevant noisy samples thus change dramatically during training, especially at the early stages. Accordingly, we can identify noisy samples by measuring the changes in prediction.

Let (x_i, y_i) be a pair comprising sample x_i and its label y_i, and $\mathcal{D} = \{(x_i, y_i)|1 \leq i \leq N\}$ be the noisy web training set. Assume that the neural network $\boldsymbol{h} = (h_1, ..., h_M)$ is trained to classify M classes. At each epoch t, we first utilize the network output logits $\boldsymbol{h}(x_i)$ to compute the softmax probability $\boldsymbol{p}(x_i)^t = (p_1(x_i)^t, ..., p_M(x_i)^t)$ for each instance x_i in the training set $\mathcal{D}$:

$$p_j(x_i)^t = \frac{\mathrm{e}^{h_j(x_i)}}{\sum_{s=1}^{M} \mathrm{e}^{h_s(x_i)}}. \tag{4}$$

Then when epoch $t > 2$, for each instance x_i, we compute the softmax probability cross-entropy $C(x_i)^t$ between $\boldsymbol{p}(x_i)^{t-2}$ and $\boldsymbol{p}(x_i)^{t-1}$ through

$$C(x_i)^t = -\sum_{j=1}^{M} p_j(x_i)^{t-1} \log p_j(x_i)^{t-2}. \tag{5}$$

The probability cross-entropy $C(x_i)^t$ reveals the changes in prediction. Since the predictions of irrelevant noisy samples change more rapidly than those of clean ones, they have higher probability cross-entropy values. For useful samples x in the selected set $\widehat{\mathcal{D}}$ and noisy samples $\widetilde{x}$ in the dropped set $(\mathcal{D}-\widehat{\mathcal{D}})$, we have

$$\frac{1}{|\mathcal{D}-\widehat{\mathcal{D}}|} \sum_{\widetilde{x}\in(\mathcal{D}-\widehat{\mathcal{D}})} C(\widetilde{x}) > \frac{1}{|\widehat{\mathcal{D}}|} \sum_{x\in\widehat{\mathcal{D}}} C(x). \tag{6}$$

Then, we can select samples that have a low probability cross-entropy $C(x)^t$ as useful images and use them to update the network at each epoch t. Different from existing methods that directly leverage cross-entropy, our method utilizes the probability, a soft label, to identify noisy samples. In this way, our approach can distinguish irrelevant noise from useful images in a web training set more efficiently than existing methods.

2) Dynamic Drop Rate and Global Sampling: Our approach selects samples from the whole training set and considers images with a low cross-entropy $C(x)^t$ as useful samples. By doing this, it can form a selected set $\widehat{\mathcal{D}}^t$. Images with a high cross-entropy $C(x)^t$ are regarded as irrelevant noisy images and are not used for training. The number of selected samples is controlled by a drop rate

$$r(t) = \tau \cdot \min\{\frac{t}{t_k}, 1\}, \tag{7}$$

which is dynamically updated during training. Parameter t_k is the initial epoch number, which controls when the drop rate reaches the maximum value τ. In the early training stage ($t \leq t_k$), $r(t)$ rises steadily until it reaches the **maximum drop rate** τ. $\widehat{\mathcal{D}}^t$ can be obtained by solving the following equation:

$$\widehat{\mathcal{D}}^t = \underset{\mathcal{D}':|\mathcal{D}'|\geq(1-r(t))|\mathcal{D}|}{\arg\min} \sum\nolimits_{x\ \in\ \mathcal{D}'} C(x)^t. \tag{8}$$

Eq. 8 indicates that our method updates $\widehat{\mathcal{D}}^t$ at each epoch t by selecting $(1 - r(t)) \times 100\%$ samples with small $C(x)^t$ from the noisy training set $\mathcal{D}$. Then, it leverages only the selected set $\widehat{\mathcal{D}}^t$ to update the parameters of network $\boldsymbol{h}$.

Similar to Co-teaching [35], our approach utilizes a linearly increased drop rate $r(t)$ in the early training epochs ($t \leq t_k$).

Algorithm 1: Softly Update-Drop Training

Input: Initialized network $\boldsymbol{h}$, training set $\mathcal{D}$, maximum drop rate τ, epoch t_k and $t_{\max}$.
for $t = 1, 2, ..., t_{\max}$ **do**
  for each instance x_i in training set $\mathcal{D}$ **do**
    if $t > 2$ **then**
      Compute $C(x_i)^t$ according to Eq. (5)
    end
    Compute $\boldsymbol{p}(x_i)^t$ according to Eq. (4)
  end
  Update $r(t)$ according to Eq. (7)
  if $t > 2$ **then**
    Obtain selected set $\widehat{\mathcal{D}}^t$ according to Eq. (8)
  else
    Set $\widehat{\mathcal{D}}^t = \mathcal{D}$
  end
  Update $\boldsymbol{h}$ according to Eq. (11)
end
Output: Updated network $\boldsymbol{h}$

TABLE I: ACA (%) performances on three fine-grained benchmark datasets. BBox/Anno (✓) indicates that human annotations are utilized during training. "Training set" shows whether the dataset is manually labeled (anno.) or collected from the web (web). iNat denotes the iNaturalist dataset.

Supervision	Method	Publication	BBox/Anno	Training Set	Datasets		
					CUB200-2011	FGVC-Aircraft	Cars-196
Strongly	Part-Stacked CNN [44]	CVPR 2016	✓	anno.	76.60	-	-
	Coarse-to-fine [45]	TIP 2016	✓	anno.	82.90	87.70	-
	HSnet [46]	CVPR 2017	✓	anno.	87.50	-	93.90
	Mask-CNN [47]	PR 2018	✓	anno.	85.70	-	-
Weakly	Bilinear CNN [56]	ICCV 2015		anno.	84.10	83.90	91.30
	RA-CNN [55]	CVPR 2017		anno.	85.30	-	92.50
	Multi-attention [57]	ICCV 2017		anno.	86.50	89.90	92.80
	Filter-bank [58]	CVPR 2018		anno.	86.70	92.00	93.80
	Parts Model [59]	CVPR 2019		anno.	90.40	-	-
	TASN [60]	CVPR 2019		anno.	89.10	-	93.80
	DCL [61]	CVPR 2019		anno.	87.80	93.00	94.50
Semi	Xu *et al.* [63]	TPAMI 2018	✓	anno.+web	84.60	-	-
	Cui *et al.* [21]	CVPR 2016	✓	anno.+web	80.70	-	-
	Niu *et al.* [20]	CVPR 2018		anno.+web	76.47	-	-
	Cui *et al.* [68]	CVPR 2018		anno.+iNat	89.29	90.70	93.50
Webly	WSDG [64]	CVPR 2015		web	70.61	-	-
	Xiao *et al.* [65]	CVPR 2015		web	70.92	-	-
	Decoupling [30]	NeurIPS 2017		web	70.56	75.97	75.00
	Co-teaching [35]	NeurIPS 2018		web	73.85	72.76	73.10
	No-correction	-		web	66.57	64.33	67.42
	Cross-entropy	-		web	77.13	70.03	78.56
	Probability Cross-entropy	-		web	77.22	72.88	78.71
	Ours	-		web	**78.17**	**77.95**	**83.50**

As indicated by memorization effects [25], [26], [35], deep neural networks have the ability to filter out noisy instances using their loss values during the early training stage, but will eventually overfit to noisy samples as the number of epochs increases. To leverage this property, our approach dynamically increases the drop rate $r(t)$ and manages to retain more instances at early epochs and increasingly drop noisy images before they are memorized.

Existing methods tend to perform sample selection using mini-batches [35]. However, the number of noisy images N_i in a mini-batch *i* follows a hypergeometric distribution. Given the noise rate $R_{\mathcal{D}}$ of dataset $\mathcal{D}$ and batch size N_b, we have

$$N_i \sim H(|\mathcal{D}|, |\mathcal{D}| \cdot R_{\mathcal{D}}, N_b). \tag{9}$$

In this distribution, the number of noisy images N_i fluctuates among different batches, resulting in a noise rate imbalance problem. Specifically, some batches may have less noisy samples, while others have more. Thus, when the drop rate $r(t)$ is fixed in each epoch, clean samples might have to be dropped in some mini-batches, while noisy samples used for training in others. Therefore, the sample selection in mini-batches is unstable and unreliable. To overcome the noise rate imbalance problem, our approach selects samples from the whole training set. By making the selection results more stable, better performance can be achieved.

C. Hard Examples Utilization

In addition to removing noisy samples, our approach is also able to properly utilize hard examples in the web training set. Since hard examples carry more irrelevant background information, they may misguide the model to learn irrelevant or invalid information. This may result in a degradation of the model's generalization ability. If, on the other hand, the hard examples are used properly, they can instead improve the robustness of the model. To address this contradiction, we propose to leverage label smoothing as it can prevent overfitting by making the model less confident about the predictions. Our approach assigns a label weight ω for image label and $\frac{1-\omega}{M-1}$ for other categories. Then the smooth loss functions are:

$$\mathcal{L}_{Smooth}(x_i, y_i) = \omega \cdot \mathcal{L}_N(x_i, y_i) + \frac{1-\omega}{M-1} \cdot \sum_{j \neq y_i} \mathcal{L}_N(x_i, j), \tag{10}$$

and

$$\mathcal{L}_{Final} = \frac{1}{|\widehat{\mathcal{D}}^t|} \sum_{x \in \widehat{\mathcal{D}}^t} \mathcal{L}_{Smooth}(x, y), \tag{11}$$

where M is the number of the categories, and j indicates each category except y_i. Parameter $\omega \in (0, 1)$ controls the confidence of the prediction. A large ω barely improves the generalization, while a small ω may cause underfitting. In our experiments, we find that a moderate ω can significantly improve the performance. The detailed steps of our proposed approach are summarized in Algorithm 1.

IV. Experiments

A. Datasets and Evaluation Metric

Unfortunately, we cannot use web images as a validation/test set, because their labels are potentially incorrect. However, we can evaluate our approach on three commonly used fine-grained benchmark datasets, CUB200-2011 [69], FGVC-aircraft [70], and Cars-196 [71].

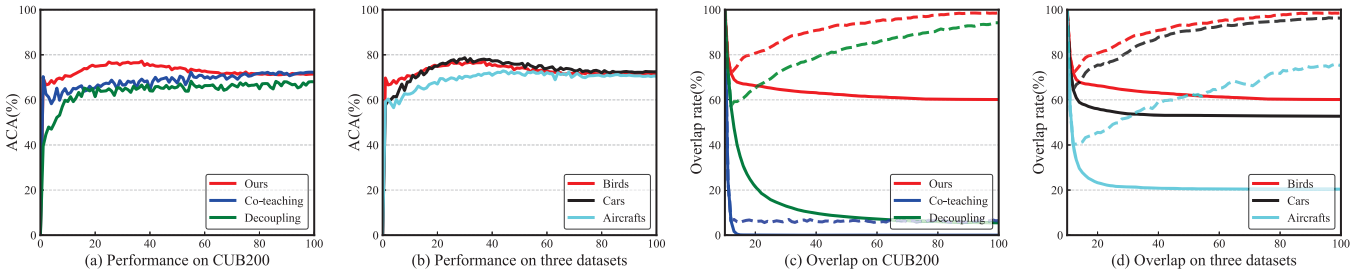


Fig. 4: Test accuracy and overlap rate vs. number of epochs. (a): Test accuracies of our approach, Co-teaching, and Decoupling on CUB200; (b): Test accuracies of our approach on three benchmark datasets; (c): Overlap rates of our approach, Co-teaching and Decoupling on CUB200; (d): Overlap rates of our approach on three benchmark datasets. The overlap rates of all previous epochs are plotted with solid lines, while the overlap rates of three contiguous epochs (*e.g.*, epoch t_{i-2}, t_{i-1} and t_i) are plotted with dotted lines in (c) and (d).

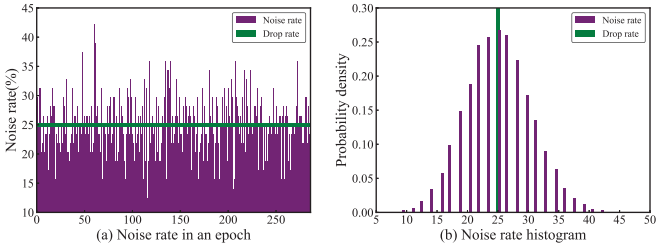


Fig. 5: Noise rate of each mini-batch in an epoch (a) and noise rate histogram of all mini-batches (b).

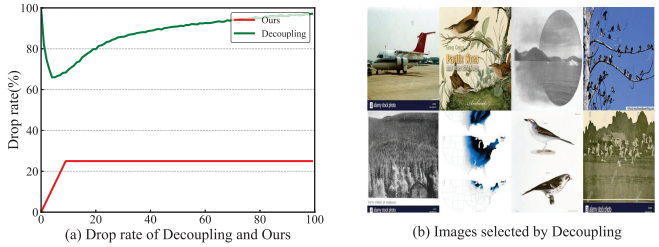


Fig. 6: Drop rate of Decoupling (a) and sample images selected for training by Decoupling (b).

CUB200-2011 covers 200 different subcategories of bird. It contains 11,788 images in total: 5,994 for training and 5794 for testing. For each subcategory, 30 images are selected for training and 11-30 for testing. Each image has detailed annotations: a subcategory label, a bounding box around the object, 15 part locations, and 312 binary attributes.

FGVC-aircraft consists of 10,000 images of 100 aircraft model variants: 6,667 images for training and 3,333 for testing. The aircraft in each image is annotated with a tight bounding box and a hierarchical airplane model label.

Cars-196: contains 196 subcategories of car, and includes 16,185 images: 8,144 for training and 8,041 for testing. For each subcategory, 24-84 images are selected for training and 24-83 for testing. Each image is annotated with a subcategory label and a bounding box of the object.

Average Classification Accuracy (**ACA**) is taken as the evaluation metric, which is widely used for evaluating the performance of fine-grained visual classification.

B. Implementation Details

We directly utilize the web images collected in [43] and set these as the training set. We adopt the testing data from CUB200-2011, FGVC-aircraft, and Cars-196 as the test set for evaluation. Note that, we use two backbone networks, VGG-16 and ResNet-18. We select the maximum drop rate τ from {0.15, 0.2, 0.25, 0.3}, epoch t_k from {5, 10, 15, 20}, and label weight ω from the range [0.1,0.9]. Based on the results of experiments, we ultimately set $\tau = 0.25$ and $t_k = 10$ as the default values for CUB200-2011 and Cars-196, and set $\tau = 0.20$ and $t_k = 10$ for the FGVC-Aircraft dataset. The label weight ω is set to 0.6 for VGG-16 and 0.5 for ResNet-18.

For the VGG-16 backbone, we follow [56] and adopt a two-step training strategy. Specifically, we first freeze the convolutional layer parameters and only optimize the last fully connected layer. Then we optimize all layers in the previously learned model. In our experiments, we use an SGD optimizer with momentum = 0.9. The learning rate and number of epochs are set to 0.01 and 80 for both steps. The batch size is set to 64 and 32 for the first and second steps, respectively. We leverage a warm-up period of five epochs and then decay the learning rate through cosine annealing. For the ResNet-18 backbone, we utilize one-step training and set the batch size to 32. The other training settings are the same for the VGG-16 backbone.

C. Baselines

To illustrate the superiority of our approach, the following state-of-the-art methods are chosen as our baselines:
1) Strongly supervised fine-grained methods: Part-Stacked CNN [44], Coarse-to-fine [45], HSnet [46], and Mask-CNN [47]; 2) Weakly supervised fine-grained methods: Bilinear CNN [56], RA-CNN [55], Filter-bank [58], Multi-attention [57], Parts Model [59], TASN [60], and DCL [61]; 3) Semi-supervised fine-grained methods: [63], [20], [21], and [68]; 4) Web-supervised methods: WSDG [64], [65], Decoupling [30], and Co-teaching [35]. For Co-teaching and Decoupling, we replace the basic network with the same VGG-16 backbone network as ours and train the fine-grained models with the

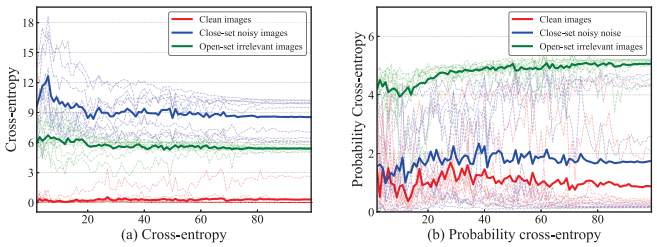


Fig. 7: Cross-entropy (a) and Probability cross-entropy (b) of clean images, close-set noisy images and open-set irrelevant images. The value of each image is plotted in dotted line and the average value is plotted in solid line.

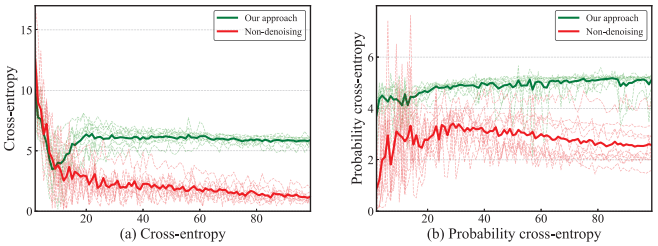


Fig. 8: Cross-entropy (a) and Probability cross-entropy (b) of irrelevant noisy images in our approach and non-denoising training. The value of each image is plotted in dotted line and the average value is plotted in the solid line.

same web datasets. To be specific, we use all the same implementation settings except for the batch sizes, which are changed to 64 and 16 in the first and second steps, respectively. For Co-teaching, we set the maximum drop rate $\tau = 0.25$ and epoch $t_k = 10$. In addition, we train the VGG-16 backbone network without any correction (No-correction) and use cross-entropy to identify noise (Cross-entropy) for comparison. Experiments are conducted on one NVIDIA V100 GPU card.

D. Performance of Denoising

We first present the denoising performance of our approach, which does not utilize the normalization and label smoothing steps.

1) Experimental Results and Analysis: Table I presents the fine-grained ACA results of various approaches on benchmark datasets. As can be seen in Table I, our proposed approach (Probability Cross-entropy) obtains significant improvements compared to other web-supervised methods on the CUB200-2011 and Cars-196 datasets. On the FGVC-Aircraft dataset, our approach achieves slightly better performance than Co-teaching.

Fig. 4 (a) presents the test accuracy vs. number of epochs for our approach, Decoupling, and Co-teaching on the CUB200 dataset. The memorization effect of networks can clearly observed in our approach, where the test accuracy quickly increases to a high level and then gradually decreases. In contrast, the test accuracies of Decoupling and Co-teaching rise slowly with obvious fluctuation, failing to reach a high level at the early stage of training. This is because our approach has a better sample selection ability, which enables it to reach a higher peak in much fewer epochs. Fig. 4 (b) shows the test accuracy vs. number of epochs on CUB200, FGVC-Aircrafts, and Cars-196. By observing Fig. 4 (b), a similar trend to that discussed above can be observed.

To further demonstrate the sample selection ability of our approach, we record the selection result of each epoch during training for epoch $t > t_k$ (we set t_k=10) and compute the *overlap rate* of selected noisy samples. We define the instance that is identified as a noisy sample in contiguous epochs (*e.g.* epoch t_{i-2}, t_{i-1} and t_i, or all epochs) as an overlapped image. Given the number of overlapped noisy images N_o, total number of training samples N, and drop rate $r(t)$, the overlap rate O is computed by $O = \frac{N_o}{N \cdot r(t)}$, which indicates the selection stability. Specifically, a higher overlap rate indicates stabler selection results. In contrast, if the selection results in different epochs are diverse, the overlap rate will be low.

Fig. 4 (c) shows the results of our approach, Co-teaching, and Decoupling on the CUB200 dataset. Fig. 4 (d) provides the results of our approach on all three datasets. Co-teaching leverages the same drop rate as ours. However, as the number of epochs increases, the overlap rate of all epochs in Co-teaching decreases to 0 rapidly, while our approach maintains a roughly stable number after a small drop (Fig. 4 (c)). Similarly, the overlap of contiguous epochs in Co-teaching remains small (around 10%), while our approach clearly contains more overlap, which rises steadily as the number of epochs increases. This means that our approach maintains a stable selection result, which becomes more stable as the training continues. This improvement can be attributed to our global selection strategy. Specifically, Co-teaching performs sample selection in a mini-batch, where it cannot tackle the noise rate imbalance problem. Thus its selection result is unstable and changes rapidly during training. This further causes the network to learn from noisy images. By overcoming this drawback, our approach has better sample selection consistency and performance. From Fig. 4 (d), we can observe that our approach maintains stable sample selection results on all three datasets, especially on CUB200 and Cars-196.

2) Validation Set and Early Stop: Given that the training set is noisy, we cannot simply separate a validation set out of the noisy web images. However, we can randomly choose 2,000 images from the CUB200 training set as a validation

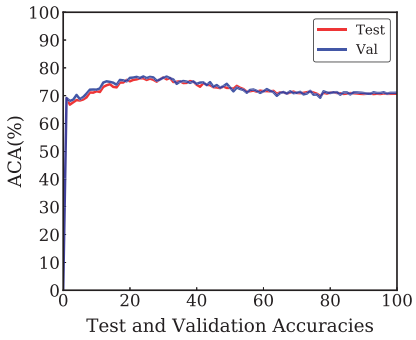


Fig. 9: Test and Validation Accuracies on CUB200.

TABLE II: ACA(%) performances of different backbones, dataset sizes, and frameworks.

Backbones	ACA (%)
VGG-16	77.22
VGG-19	75.87
ResNet-34	74.99

Dataset Sizes	ACA (%)
50	71.87
75	74.85
100	77.22

Frameworks	ACA (%)
Co-teaching	75.46
Peer networks	76.30
Single network	77.22

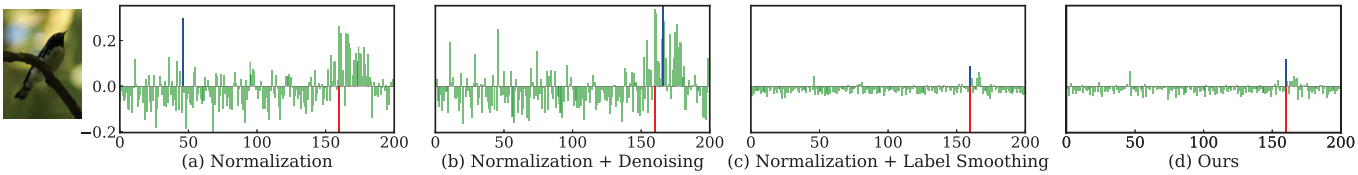


Fig. 10: The logits of four models: (a) only normalization; (b) normalization and denoising; (c) normalization and label smoothing; (d) our approach. The red and blue bars indicate the true label and prediction of the test image, respectively.

set. Specifically, we randomly choose 2,000 images from the CUB200 original training set as a validation set. The size of validation set is about one-third of the test set. We then perform the model selection based on the validation set. Finally, we use the selected model to perform the testing on the test set. We plot the accuracy vs. epoch curves by using the validation and test set, respectively. The experimental results are presented in Fig. 9. By observing Fig. 9, we can notice that the performance of our method increases firstly and then decreases. The explanation is that deep neural networks have memorization effects [26]. Deep neural networks will first learn the clean and easy patterns in the initial epochs. However, as the number of epochs increases, deep neural networks will eventually overfit on noisy labels. Therefore, a validation set for early stop and model selection is essential.

3) Noise Rate Imbalance: We investigate the noise rate imbalance problem in mini-batches with noisy bird training images. We record the number of dropped images during training for each mini-batch. Assuming that the dropped images are noisy samples, we can compute the noise rate R_i of mini-batch i. Given the number of dropped images N_i of mini-batch i and batch size N_b, we can calculate R_i by $R_i = \frac{N_i}{N_b}$. Fig. 5 (a) presents the noise rate of each mini-batch in a randomly selected epoch. It ranges from 12% to 42% with obvious fluctuation around the drop rate. To illustrate the distribution of noise rate R, we record the noise rates of all mini-batches (more than 25,000) during training and plot the histogram of R in Fig. 5 (b). By observing Fig. 5 (b), we can notice that R follows a Gaussian distribution, ranging from 6% to 48%. Fig. 5 (a) explicitly shows the noise rate imbalance in mini-batches.

Decoupling selects samples with different predictions from two peer networks to update the model. It is a global selection method and does not have the noise rate imbalance problem, so it provides a stable selection result. As shown in Fig. 4 (c), the overlap rate of three contiguous epochs in Decoupling is close to our approach. However, the drop rate of Decoupling is too high. We record the number of dropped images and compute the drop rate of Decoupling in each epoch. The results are shown in Fig. 6 (a). From Fig. 6 (a), we can observe that the drop rate of Decoupling is always larger than 60% and it climbs to nearly 100% as training continues.

The extremely high drop rate demonstrates that Decoupling unable to make full use of the clean samples. Besides, since irrelevant noise is hard to fit to, the peer networks have a high probability of producing different predictions for these samples. Then, when the samples are used for training, they misguide the networks. Fig. 6 (b) visualizes some overlapped images which are used for training in Decoupling. These images are irrelevant noisy samples, indicating that Decoupling is not capable of tackling this irrelevant noise.

Moreover, we compare the performances of global and mini-batch selection in Table III. From Table III, we can observe that mini-batch selection has a higher training accuracy but lower testing accuracy than global selection. This suggests that mini-batch selection learns more training samples, some of which are noisy. It is then misguided by the noisy samples and achieves lower performance. Performing sample selection in a mini-batch with a fixed drop rate is thus unable to tackle the noise rate imbalance problem. In contrast, our proposed approach, which leverages global sample selection, can mitigate this problem and achieve better performance.

4) Probability Cross-Entropy and Cross-Entropy: In this experiment, we compare the performance of probability cross-entropy and cross-entropy in identifying noise on noisy bird training images. We first save the models produced each epoch during training. Then we leverage them to identify clean images, close-set noisy images, and open-set irrelevant images (30 images in total, 10 of each kind). We record their cross-entropy as well as probability cross-entropy. The experimental results are shown in Fig. 7. By observing Fig. 7 (b), we can notice that the probability cross-entropy of open-set irrelevant images is much larger than that of close-set noisy images and clean data. Compared with clean images, both close-set noisy images and open-set irrelevant images have a larger loss. From Fig. 7 (a) and (b), we can conclude that selecting samples with cross-entropy cannot distinguish close-set and open-set noise. Nevertheless, leveraging our proposed probability cross-entropy to identify open-set irrelevant images is reliable. We

TABLE III: Training and testing ACA results (%) of global and mini-batch selection mechanism.

Backbone	Selection	Training (%)	Testing (%)
VGG-16	Global	76.53	**77.22**
	Mini-batch	83.21	75.66

TABLE IV: ACA (%) performances of data augmentation. Anno. and web denotes the dataset is manually labeled and collected from the web, respectively.

Models	Backbone	Training Set	Performance
Ours	ResNet-18	anno.	81.53
	ResNet-18	anno.+web	**86.19**
	VGG-16	anno.	82.79
	VGG-16	anno.+web	**86.26**
	ResNet-50	anno.	83.48
	ResNet-50	anno.+web	**87.57**

TABLE V: ACA (%) performance and improvement of VGG-16, ResNet-18 and ResNet-50. Baseline denotes the original backbone network.

Backbone	Method	Performance	Improvement
VGG-16	Baseline	66.57	11.6
	Ours	78.17	
ResNet-18	Baseline	68.59	8.58
	Ours	77.17	
ResNet-50	Baseline	73.01	7.38
	Ours	80.39	

also compare the performances when identifying noise by probability cross-entropy and cross-entropy in Table. I. As demonstrated in Table. I, probability cross-entropy shows a slightly better performance on each dataset. One possible explanation is that some hard examples are regarded as noise by cross-entropy, because they tend to have larger cross-entropy values during training.

5) Effectiveness of Denoising: To explain the effectiveness of denoising proposed in our approach, we train the network using original web images without any denoising and record the probability cross-entropy and cross-entropy of 10 irrelevant noisy images during training. We also train the network using web images selected by our proposed approach and compare the results in Fig. 8. From Fig. 8 (b), we can find that the probability cross-entropy is large and declines extremely slowly during training, meaning that using probability cross-entropy can identify irrelevant noise during training and learn robust models. From Fig. 8 (a), we also notice that the cross-entropy in the non-denoising method gradually drops during training. In contrast, the cross-entropy in our approach drops slightly at first and then climbs to a roughly constant value. The explanation is that our approach has the ability to drop irrelevant noisy images before the network fits them.

6) Influence of Different Backbones: To investigate the influence of different CNN architectures in the denoising model, we replace VGG-16 with VGG-19 and ResNet-34. As shown in Table II (left), these three backbone networks achieve similar performances on CUB200. The performances of VGG-19 and ResNet-34 are slightly worse than that of VGG-16. One possible explanation is that we use the same coefficient settings for these backbones, which is best for VGG-16.

7) Influence of Different Dataset Sizes: We investigate the impact of data scale by changing the number of web images used for each category on CUB200. Specifically, we collect 50, 75, and 100 images from the web for each category. As shown in Table II (middle), in general, the ACA performance improves steadily when using more training images. Therefore, web-supervised learning is a promising research direction as it allows large-scale datasets to easily be built.

8) Influence of Multiple Networks: We conduct two experiments to study whether using multiple networks can improve performance. In the first experiment, we combine our approach with the Co-teaching framework, letting the two networks select samples for each other. In the second experiment, we use two peer networks and leverage their outputs to compute the probability cross-entropy. Given the softmax probabilities $\boldsymbol{p}(x_i)^{t-1}$ and $\boldsymbol{q}(x_i)^{t-1}$ of two peer networks, the cross-entropy $C(x_i)^t$ is computed by $C(x_i)^t = -\sum_{j=1}^{N} p_j(x_i)^{t-1} \log q_j(x_i)^{t-1}$. The results are demonstrated in Table II (right). Both frameworks show worse performance than our proposed approach, which only utilizes a single network. Compared with methods that need two networks, our approach is lighter and more efficient.

9) Summary of Denoising: In our denoising experiments, we first illustrate that our denoising method outperforms other web-supervised methods in both performance and sample selection ability. Then we demonstrate that our global selection strategy can overcome the noise rate imbalance problem. Next, we compare our probability cross-entropy with cross-entropy, and illustrate that utilizing probability cross-entropy can identify open-set irrelevant images reliably. Finally, we investigate the influence of different backbones, dataset sizes and multi-networks.

E. Performances of Normalization and Label Smoothing

In this subsection, we demonstrate the improvements provided by the normalization and label smoothing in our proposed full framework. The experimental results are shown in Table I. By observing Table I, we can notice that the normalization and label smoothing (full) remarkably improve the ACA performance on three datasets compared with only dropping irrelevant noisy images. The improvements are 0.95%, 5.07% and 4.79% on CUB200-2011, FGVC-aircraft, and Cars-196 datasets, respectively. The explanation is that utilizing hard examples can effectively improve the robustness of the model.

1) Influence of Different Modules: Fig. 11 illustrates the impact of normalization, denoising, and label smoothing. This experiment is conducted with a ResNet-18 backbone because

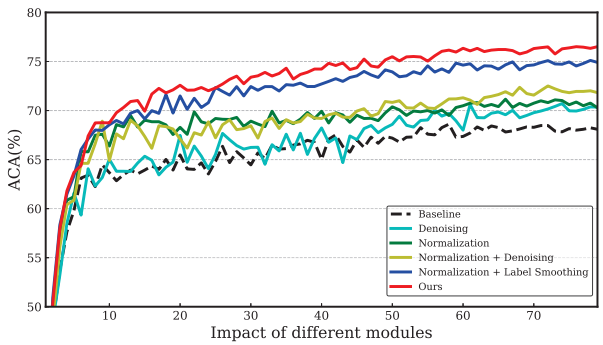


Fig. 11: Test accuracy vs. number of epochs for each module.

the impact of each module is more distinct on the ResNet-18 than VGG-16. To better demonstrate the improvements, we train a ResNet-18 model without any modules as the baseline for comparison. From Fig. 11, we can observe that the baseline is outperformed by simply utilizing the normalization. On the basis of normalization, dropping noisy images or using label smoothing further improves the performance. Specifically, label smoothing contributes more to the improvement. Compared with using each module individually, our method shows much better performance as denoising and utilizing hard web images significantly improve the performance and training efficiency. To further explore the effectiveness of denoising and label smoothing, we visualize the logits $\boldsymbol{h}(x_i)$ of four models: only normalization; normalization and denoising; normalization and label smoothing, and our full approach. The results are shown in Fig. 10.

2) Influence of Data Augmentation: To demonstrate the advantage of leveraging web images, we train our model on the CUB200 dataset with our web images as data augmentation. To be specific, we set $\omega = 0.8$ and $\omega = 0.5$ for images in labeled and web datasets, respectively. Images in the labeled dataset have less background information and deserve a higher label weight ω. In addition, we train models on the labeled dataset for comparison. The results are given in Table IV. From Table IV, we can observe that leveraging web images remarkably improves the performance across different backbone networks. The improvements on VGG-16, ResNet-18 and ResNet-50 are 3.47%, 4.66% and 4.09%, respectively. These results also indicate that leveraging web images is an effective way to improve the robustness of the model.

3) Influence of Different Backbones: To investigate the influences of different backbones in the whole framework, we conduct experiments on the CUB200 dataset with different backbone networks. The experimental results are demonstrated in Table V. By observing Table V, we can notice that our approach shows significant improvements across different backbone networks. The experimental results also indicate that our proposed approach is a robust and effective web-supervised method.

4) Visualization: Fig. 12 visualizes the sample selection results of our approach on three web datasets. From Fig. 12, we can observe that the irrelevant noisy images and useful images are clearly separated. Most irrelevant noisy images in the bird dataset have no relationship with birds. In the aircraft dataset, irrelevant noisy images are structure charts and cockpits, while in the car dataset, they tend to be logos and internal views of the car. They are related to the aircraft and car but different from images in standard datasets and harmful for training. Although the irrelevant noisy images in these datasets are totally different, our approach is still able to distinguish them from different datasets. These selection results also demonstrate that our approach is robust and can be leveraged to refurbish the web images for practical applications.

5) Summary of Normalization and Label Smoothing: In our experiments, we elaborate on improvements caused by normalization and label smoothing. We first investigate the influence of each module in the performance and output logits. We find that all modules contribute to the final performance, where label smoothing shows a more significant contribution than others. We then demonstrate the advantage of utilizing web images by conducting data augmentation. Next, we illustrate the robustness of our approach through changing backbones. Finally, we analyze the parameter sensitivity and intuitively visualize our sample selection results.

(a) Bird dataset (b) Aircraft dataset (c) Car dataset

Fig. 12: Sample selection results on three web datasets. For each dataset, the irrelevant noisy and useful images selected by our approach are shown on the left and right, respectively.

V. Conclusion

In this paper, we presented a simple yet effective training method for web-supervised fine-grained visual classification. Our key idea is to drop irrelevant noisy images and utilize hard examples to boost the model training. Specifically, we select samples that have a large probability cross-entropy as irrelevant noisy images and then drop them during training. Then, we leverage normalization and label smoothing to mine hard examples as well as prevent overfitting. Comprehensive experiments on three real-world scenario datasets demonstrate that our approach is superior to current state-of-the-art methods for web-supervised fine-grained visual classification. More importantly, our approach is a general and practical method for pure web-supervised learning, which avoids the heavy labor requirements of manual labeling. It can be applied to other tasks that lack well-labeled training data.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 61976116), Fundamental Research Funds for the Central Universities (No. 30920021135), JiangsuÂăProvinceÂăKeyÂăResearchÂăandÂăDevelopmentÂăPlanÂă(No. BE2016904), and Ark Innovation Fund (No. HT202012310145).

References

[1] H. Zhang, Y. Gu, Y. Yao, Z. Zhang, L. Liu, J. Zhang, and L. Shao, "Deep unsupervised self-evolutionary hashing for image retrieval," *IEEE Transactions on Multimedia*, 2020.

[2] B. Hu, R. Song, X. Wei, Y. Yao, X. Hua, and Y. Liu, "Pyretri: A pytorch-based library for unsupervised image retrieval by deep convolutional neural networks," *ACM International Conference on Multimedia*. 4461âĂŞ–4464, 2020.

[3] W. Wang, Y. Shen, H. Zhang, Y. Yao, and L. Liu, "Set and rebase: determining the semantic graph connectivity for unsupervised cross modal hashing," *International Joint Conference on Artificial Intelligence*, 853–859, 2020.

[4] Y. Yao, Z. Sun, F. Shen, L. Liu, L. Wang, F. Zhu, L. Ding, G. Wu, and L. Shao, "Dynamically visual disambiguation of keyword-based image search," *International Joint Conference on Artificial Intelligence*, 996–1002, 2019.

[5] Z. Li, J. Zhang, Y. Gong, Y. Yao, and Q. Wu, "Field-wise learning for multi-field categorical data," *International Conference on Neural Information Processing Systems*, 2020.

[6] G. Xie, L. Liu, X. Jin, F. Zhu, Z. Zhang, J. Qin, Y. Yao, and L. Shao, "Attentive region embedding network for zero-shot learning," *IEEE Conference on Computer Vision and Pattern Recognition*, 9384–9393, 2019.

[7] Y. Yao, J. Zhang, F. Shen, W. Yang, X. Hua, and Z. Tang, "Extracting privileged information from untagged corpora for classifier learning.," *International Joint Conference on Artificial Intelligence*, 1085–1091, 2018.

[8] G. Xie, L. Liu, F. Zhu, F. Zhao, Z. Zhang, Y. Yao, J. Qin, and L. Shao, "Region graph embedding network for zero-shot learning," *European Conference on Computer Vision*. 562–580, 2020.

[9] Z. Sun, Y. Yao, J. Xiao, L. Zhang, J. Zhang, and Z. Tang, "Exploiting textual queries for dynamically visual disambiguation," *Pattern Recognition*, 110: 107620, 2021.

[10] T. Zhou, W. Wang, Y. Yao, and J. Shen, "Target-aware adaptive tracking for unsupervised video object segmentation," The DAVIS Challenge on Video Object Segmentation on CVPR Workshop. 2020.

[11] Y. Yao, F. Shen, J. Zhang, L. Liu, Z. Tang and L. Shao, Extracting Privileged Information for Enhancing Classifier Learning, IEEE Transactions on Image Processing, 28(1): 436–450, 2019.

[12] Y. Yao, F. Shen, G. Xie, L. Liu, F. Zhu, J. Zhang, and H. Shen, "Exploiting web images for multi-output classification: From category to subcategories," *IEEE Transactions on Neural Networks and Learning Systems*, 31(7): 2348–2360, 2020.

[13] T. Zhou, S. Wang, Y. Zhou, Y. Yao, J. Li, and L. Shao, "Motion-attentive transition for zero-shot video object segmentation.," *AAAI Conference on Artificial Intelligence*, 13066–13073, 2020.

[14] T. Chen, J. Zhang, G. Xie, Y. Yao, X. Huang, and Z. Tang, "Classification constrained discriminator for domain adaptive semantic segmentation," *IEEE International Conference on Multimedia and Expo*, 1–6, 2020.

[15] J. Lu, H. Liu, Y. Yao, S. Tao, Z. Tang, and J. Lu, "Hsi road: A hyper spectral image dataset for road segmentation," *IEEE International Conference on Multimedia and Expo*, 1–6, 2020.

[16] H. Luo, G. Lin, Z. Liu, F. Liu, Z. Tang, and Y. Yao, "Segeqa: Video segmentation based visual attention for embodied question answering," *IEEE International Conference on Computer Vision*, 9667–9676, 2019.

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li, "Imagenet: A large-scale hierarchical image database," *IEEE Conference on Computer Vision and Pattern Recognition*, 248–255, 2009.

[18] C. Zhang, Y. Yao, J. Zhang, J. Chen, P. Huang, J. Zhang, and Z. Tang, "Web-supervised network for fine-grained visual classification," *IEEE International Conference on Multimedia and Expo*, 1–6, 2020.

[19] Z. Xu, S. Huang, Y. Zhang, and D. Tao, "Augmenting strong supervision using web data for fine-grained categorization," *IEEE International Conference on Computer Vision*, 2524–2532, 2015.

[20] L. Niu, A. Veeraraghavan, and A. Sabharwal, "Webly supervised learning meets zero-shot learning: A hybrid approach for fine-grained classification," *IEEE Conference on Computer Vision and Pattern Recognition*, 7171–7180, 2018.

[21] Y. Cui, F. Zhou, Y. Lin, and S. Belongie, "Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop," *IEEE Conference on Computer Vision and Pattern Recognition*, 1153–1162, 2016.

[22] Y. Yao, J. Zhang, F. Shen, L. Liu, F. Zhu, D. Zhang, and H. Shen, "Towards automatic construction of diverse, high-quality image datasets," *IEEE Transactions on Knowledge and Data Engineering*, 32(6): 1199–1211, 2020.

[23] Y. Yao, J. Zhang, F. Shen, X. Hua, J. Xu, and Z. Tang, "Automatic image dataset construction with multiple textual metadata," *IEEE International Conference on Multimedia and Expo*, 1–6, 2016.

[24] F. Schroff, A. Criminisi, and A. Zisserman, "Harvesting image databases from the web," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4): 754–766, 2011.

[25] D. Arpit, S. Jastrzębski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, and S. Lacoste-Julien, "A closer look at memorization in deep networks," *International Conference on Machine Learning*, 233–242, 2017.

[26] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," *International Conference on Learning Representations*, 1–15, 2016.

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

[28] Y. Yao, J. Zhang, F. Shen, W. Yang, P. Huang, and Z. Tang, "Discovering and distinguishing multiple visual senses for polysemous words," *AAAI Conference on Artificial Intelligence*, 523–530, 2018.

[29] J. Goldberger and E. Ben-Reuven, "Training deep neural-networks using a noise adaptation layer," *International Conference on Learning Representations*, 1–9, 2016.

[30] E. Malach and S. Shalev-Shwartz, "Decoupling "when to update" from "how to update"," *The Conference and Workshop on Neural Information Processing Systems*, 960–970, 2017.

[31] Y. Yao, F. Shen, J. Zhang, L. Liu, Z. Tang and L. Shao, "Extracting Multiple Visual Senses for Web Learning," *IEEE Transactions on Multimedia*, 21(1): 184–196, 2019.

[32] Y. Yao, J. Zhang, F. Shen, X. Hua, J. Xu, and Z. Tang, "Exploiting web images for dataset construction: A domain robust approach," *IEEE Transactions on Multimedia*, 19(8): 1771–1784, 2017.

[33] L. Ding, S. Liao, Y. Liu, L. Liu, F. Zhu, Y. Yao, L. Shao, and X. Gao, "Approximate kernel selection via matrix approximation," *IEEE Transactions on Neural Networks and Learning Systems*, 31(11): 4881–4891, 2020.

[34] Y. Yao, X. Hua, F. Shen, J. Zhang, and Z. Tang, "A domain robust approach for image dataset construction," *ACM international conference on Multimedia*, 212–216, 2016.

[35] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," *The Conference on Neural Information Processing Systems*, 8527–8537, 2018.

[36] Y. Wang, W. Liu, X. Ma, J. Bailey, H. Zha, L. Song, and S.-T. Xia, "Iterative learning with open-set noisy labels," *IEEE Conference on Computer Vision and Pattern Recognition*, 8688–8696, 2018.

[37] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," *arXiv*, 1706.02690v4, 2017.

[38] R. Ranjan, C. D. Castillo, and R. Chellappa, "L2-constrained softmax loss for discriminative face verification," *arXiv*, 1703.09507, 2017.

[39] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, 25(7): 926–930, 2018.

[40] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," *arXiv*, 1503.02531v1, 2015.

[41] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *IEEE Conference on Computer Vision and Pattern Recognition*, 2818–2826, 2016.

[42] B. Zhuang, L.-Q. Liu, Y. Li, C.-H. Shen, and I. Reid, "Attend in groups: a weakly-supervised deep learning framework for learning from web data," *IEEE Conference on Computer Vision and Pattern Recognition*, 1878–1887, 2017.

[43] C. Zhang, Y. Yao, H. Liu, G. S. Xie, X. Shu, T. Zhou, Z. Zhang, F. Shen, and Z. Tang, "Web-supervised network with softly update-drop training for fine-grained visual classification," *AAAI Conference on Artificial Intelligence*, 12781–12788, 2020.

[44] S. Huang, Z. Xu, D. Tao, and Y. Zhang, "Part-stacked cnn for fine-grained visual categorization," *IEEE Conference on Computer Vision and Pattern Recognition*, 1173–1182, 2016.

[45] H. Yao, S. Zhang, Y. Zhang, J. Li, and Q. Tian, "Coarse-to-fine description for fine-grained visual categorization," *IEEE Transactions on Image Processing*, 25(10): 4858–4872, 2016.

[46] M. Lam, B. Mahasseni, and S. Todorovic, "Fine-grained recognition as hsnet search for informative image parts," *IEEE Conference on Computer Vision and Pattern Recognition*, 2520–2529, 2017.

[47] X. S. Wei, C. W. Xie, J. Wu, and C. Shen, "Mask-CNN: Localizing parts and selecting descriptors for fine-grained bird species categorization," *Pattern Recognition*, 76: 704–714, 2018.

[48] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, "Part-based R-CNNs for fine-grained category detection," *European Conference on Computer Vision*, 834–849, 2014.

[49] Y. Yao, X. Hua, G. Gao, Z. Sun, Z. Li, and J. Zhang, "Bridging the web data and fine-grained visual recognition via alleviating label noise and domain mismatch," *ACM International Conference on Multimedia*, 1735–1744, 2020.

[50] Z. Sun, X. Hua, Y. Yao, X. Wei, G. Hu, and J. Zhang, "Crssc: salvage reusable samples from noisy data for robust learning," *ACM International Conference on Multimedia*, 92–101, 2020.

[51] C. Zhang, Y. Yao, X. Shu, Z. Li, Z. Tang, and Q. Wu, "Data-driven meta-set based fine-grained visual recognition," *ACM International Conference on Multimedia*, 2372–2381, 2020.

[52] L. Jiang, Z.-Y. Zhou, T. Leung, L.-J. Li, and F.-F. Li, "Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels," *International Conference on Machine Learning*, 1–20, 2017.

[53] K. Yi and J,-X. Wu, "Probabilistic End-to-end Noise Correction for Learning with Noisy Labels," *IEEE Conference on Computer Vision and Pattern Recognition*, 7017–7025, 2019.

[54] G. Patrini, A. Rozza, A. Krishna-Menon, R. Nock, and L.-Z. Qu, "Making deep neural networks robust to label noise: A loss correction approach," *IEEE Conference on Computer Vision and Pattern Recognition*, 1944–1952, 2017.

[55] J. Fu, H. Zheng, and T. Mei, "Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition," *IEEE Conference on Computer Vision and Pattern Recognition*, 4438–4446, 2017.

[56] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear cnn models for fine-grained visual recognition," *IEEE International Conference on Computer Vision*, 1449–1457, 2015.

[57] H. Zheng, J. Fu, T. Mei, and J. Luo, "Learning multi-attention convolutional neural network for fine-grained image recognition," *IEEE International Conference on Computer Vision*, 5209–5217, 2017.

[58] Y. Wang, V.-I. Morariu, and L.-S. Davis, "Learning a discriminative filter bank within a cnn for fine-grained recognition," *IEEE Conference on Computer Vision and Pattern Recognition*, 4148–4157, 2018.

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

[60] 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," *IEEE Conference on Computer Vision and Pattern Recognition*, 5012–5021, 2019.

[61] Y. Chen, Y. Bai, W. Zhang, and T. Mei, "Destruction and construction learning for fine-grained image recognition," *IEEE Conference on Computer Vision and Pattern Recognition*, 5157–5166, 2019.

[62] D. Korsch, P. Bodesheim, and J. Denzler, "Classification-specific parts for improving fine-grained visual categorization," *arXiv*, 1909.07075, 2019.

[63] Z. Xu, S. Huang, Y. Zhang, and D. Tao, "Webly-supervised fine-grained visual categorization via deep domain adaptation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(5): 1100–1113, 2016.

[64] L. Niu, W. Li, and D. Xu, "Visual recognition by learning from web data: A weakly supervised domain generalization approach," *IEEE Conference on Computer Vision and Pattern Recognition*, 2774–2783, 2015.

[65] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang, "Learning from massive noisy labeled data for image classification," *IEEE Conference on Computer Vision and Pattern Recognition*, 2691–2699, 2015.

[66] Y. Liu, H. Li, and X. Wang, "Learning deep features via congenerous cosine loss for person recognition," *arXiv*, 1702.06890, 2017.

[67] F. Wang, X. Xiang, J. Cheng, and A.-L. Yuille, "Normface: l_2 hypersphere embedding for face verification," *ACM International Conference on Multimedia*, 1041–1049, 2017.

[68] Y. Cui, Y. Song, C. Sun, A. Howard, and S. Belongie, "Large scale fine-grained categorization and domain-specific transfer learning," *IEEE Conference on Computer Vision and Pattern Recognition*, 4109–4118, 2018.

[69] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," *Technical Report*, CNS-TR-2011-001, California Institute of Technology, 2011.

[70] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi, "Fine-grained visual classification of aircraft," *arXiv*, 1306.5151, 2013.

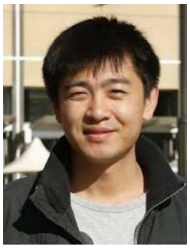
[71] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3D object representations for fine-grained categorization," *IEEE International Conference on Computer Vision*, 554–561, 2013.

Huafeng Liu is a Ph.D. student in Computer Science, School of Computer Science and Engineering, Nanjing University of Science and Technology. He received the B.Sc. Degree from Nanjing University of Science and Technology, China in 2011. His research interests include social multimedia processing, intelligent system, pattern recognition, image processing, and embedded system.

Chuanyi Zhang is a Ph.D. student in Computer Science, School of Computer Science and Engineering, Nanjing University of Science and Technology. He received the B.Sc. Degree from Nanjing University of Science and Technology, China in 2018. His research interests include social multimedia processing, intelligent system, pattern recognition, and image processing.

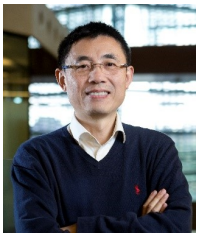
Yazhou Yao is a professor at School of Computer Science and Engineering, Nanjing University of Science and Technology. With the support of the China Scholarship Council, he received the PhD degree in Computer Science, University of Technology Sydney, Australia at 2018. From July 2018 to July 2019, he worked as a Research Scientist at Inception Institute of Artificial Intelligence, Abu Dhabi, UAE. His research interests include multimedia processing and machine learning.

Xiu-Shen Wei Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, Nanjing University of Science and Technology, Nanjing 210094, China, and also with the Jiangsu Key Laboratory of Image and Video Understanding for Social Security, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China



Fumin Shen received his Bachelor degree at 2007 and PhD degree at 2014 from Shandong University and Nanjing University of Science and Technology, China, respectively. Now he is a professor of University of Electronic Science and Technology of China. His major research interests include computer vision and machine learning, including face recognition, image analysis and hashing methods.

Zhenmin Tang received his Ph.D. degree from Nanjing University of Science and Technology, Nanjing, China. He now is a professor at Nanjing University of Science and Technology. His major research area include intelligent system, pattern recognition, and image processing, embedded system. He has published over 80 papers. He is also the leader of several key programs of National Nature Science Foundation of China.



Jian Zhang (SM' 04) received his B.S. degree in electronics from East China Normal University, China, his M.S. degree in computer science from Flinders University, Australia, and his Ph.D. degree in electrical engineering from the University of New South Wales (UNSW), Australia. From 1997 to 2003, he was a Principal Research Engineer and the Research Manager of Visual Communications Team with Motorola Australian Research Centre. From 2004 to 2011, he was a Principal Researcher and a Project Leader with Data61, Australia, and a Conjoint Associate Professor with the School of Computer Science and Engineering, UNSW. He is currently an Associate Professor in the Faculty of Engineering and IT and the Director of the Multimedia and Data Analytics Lab at the University of Technology Sydney, Australia. He has authored or co-authored over 180 paper publications, book chapters, and six issued U.S. and Chinese patents. His current interests include social multimedia signal processing, large-scale image and video content analytics, retrieval and mining, 3D-based computer vision, and intelligent video surveillance systems. Dr. Zhang was the General Co-Chair and Technical Program Co-Chair of the International Conference on Multimedia and Expo (ICME) in 2012 and 2020 respectively; and the Technical Program Co-Chair and General Co-Chair of the IEEE Conference on Visual Communications and Image Processing in 2014 and 2019 respectively. He was an Associate Editor for the IEEE Transactions on Circuits and Systems for Video Technology. Currently, he is an Associate Editor for the IEEE Transactions on Multimedia and a Member of Technical Directions Board, IEEE Signal Processing Society.