

Fine-Grained Recognition: Accounting for Subtle Differences between Similar Classes

Guolei Sun,¹ Hisham Cholakkal,² Salman Khan,² Fahad Shahbaz Khan,² Ling Shao²

¹ETH Zurich, ²Inception Institute of Artificial Intelligence

guolei.sun@vision.ee.ethz.ch, {hisham.cholakkal, salman.khan, fahad.khan, ling.shao}@inceptioniai.org

Abstract

The main requisite for fine-grained recognition task is to focus on subtle discriminative details that make the subordinate classes different from each other. We note that existing methods implicitly address this requirement and leave it to a data-driven pipeline to figure out what makes a subordinate class different from the others. This results in two major limitations: *First*, the network focuses on the most obvious distinctions between classes and overlooks more subtle inter-class variations. *Second*, the chance of misclassifying a given sample in any of the negative classes is considered equal, while in fact, confusions generally occur among only the most similar classes. Here, we propose to explicitly force the network to find the subtle differences among closely related classes. In this pursuit, we introduce two key novelties that can be easily plugged into existing end-to-end deep learning pipelines. On one hand, we introduce “diversification block” which masks the most salient features for an input to force the network to use more subtle cues for its correct classification. Concurrently, we introduce a “gradient-boosting” loss function that focuses only on the confusing classes for each sample and therefore moves swiftly along the direction on the loss surface that seeks to resolve these ambiguities. The synergy between these two blocks helps the network to learn more effective feature representations. Comprehensive experiments are performed on five challenging datasets. Our approach outperforms existing methods using similar experimental setting on all five datasets.

1 Introduction

Fine-grained recognition focuses on discriminating between children classes of a main parent category (e.g., cars (Krause et al. 2013), dogs (Khosla et al. 2011), birds (Wah et al. 2011), and aircrafts (Maji et al. 2013)). Deep CNNs have excelled immensely on traditional visual recognition tasks where categories greater differ from each other. However, fine-grained visual categorization (FGVC) poses a significant challenge mainly due to the close resemblance between subcategories e.g., different species of the same bird. The challenge is compounded by the fact that the classifier has to

Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

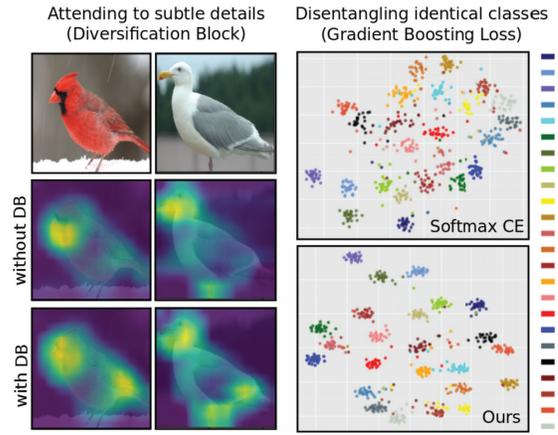


Figure 1: Illustration of two novel components of our approach. *Left*: comparison between class activation maps obtained from the model with our diversification block (DB) and the one without DB. Our DB forces the network to capture more discriminative regions. With DB (*below*), the network finds beak, tail and feet of the bird as informative regions, while without DB (*middle*), the network only focuses on beak. *Right*: visual comparison in terms of 2-d tSNE (Van Der Maaten 2014) plot for features of 24 kinds of *Walbler* (confusing and difficult classes) in CUB-200-2011 between network trained with cross entropy (CE) (*top*) and our gradient-boosting loss (*below*). By focusing on difficult classes, our gradient-boosting loss can distinguish between hard classes which are not well separated by CE.

be invariant to intra-class variations, e.g., pose, appearance and lighting changes.

Common deep learning based approaches for FGVC learn a mapping between input images and output labels. While doing so, a natural tendency during learning is to focus on only few distinguishing parts in an object to deal with confusing inter-class similarities and large intra-class variations (see Fig. 1). The analysis of attention based models provides the evidence that attention maps are often densely concentrated on a few parts, thus considering only a limited set of cues. In contrast, here we propose to spread the attention to

consolidate a diverse set of relevant cues spread across the activation map. While we diversify attention at the feature level, we do the opposite at the prediction level, i.e., focus on only the most confusing cases to achieve better discriminability. Popular loss functions such as cross entropy, consider all classes to compute the error signal for parameter update. When closely related classes are present in the data, this leads to a weak supervision signal resulting in slower convergence and low recall rates. We show that selectively attending to the hard negative classes helps in achieving much faster convergence and higher accuracy.

Our approach can also be understood as a mechanism to enhance network generalization and avoid overfitting. This consideration is of particular relevance to FGVC, since the datasets are generally smaller due to the high cost of obtaining fine-grained annotations from experts. In effort to minimize loss on training data, a high-capacity network can end up associating unrelated concepts (such as those of background) to the fine-grained object itself. By concentrating on only the closely related classes and diversifying the model’s attention, we are in fact regularizing the model to avoid overfitting the training samples. Our approach reduces classifier’s confidence on training samples and therefore makes it more generalizable. We note that regularization schemes such as label-smoothing (Szegedy et al. 2016) and maximum prediction entropy (Dubey et al. 2018b) are related to ours, but significantly different as we impose regularization on both features and output predictions.

Our main contributions are as follows:

- We introduce a *gradient-boosting* loss that seeks to resolve ambiguities among closely related classes by appropriately magnifying the gradient updates.
- Our *diversification block* masks out the salient features in order to force the network to look for subtle differences between similar-looking categories.
- The proposed method makes the convergence faster while outperforming existing methods on five datasets.

2 Related Works

Fine-grained classification has attracted much research attention in the recent years. Despite several attempts (Yang et al. 2018; Sun et al. 2018), FGVC is still an active research problem. To deal with the problem of subtle intra-class distance, many approaches focused on obtaining more relevant features (Berg and Belhumeur 2013; Lin, RoyChowdhury, and Maji 2015; Gao et al. 2016; Yang et al. 2018; Sun et al. 2018). One of the earliest but naive strategy was to exploit part annotations (Berg and Belhumeur 2013) to locate the objects so that more informative features were used. Such an approach requires more labeling effort and has therefore limited scalability. Another stream of works (Lin, RoyChowdhury, and Maji 2015; Gao et al. 2016; Li et al. 2018) developed complex pooling methods, so that complex local features can be used for classification. However, one obvious drawback of those methods is the high computation complexity. To deal with the problem of small fine-grained datasets, Cui *et al.* (Cui et al. 2018) proposed a transfer learning scheme from selected subset of the source

domain to target domain. However, it requires to re-train models on a subset of large datasets like ImageNet (Russakovsky et al. 2015) and iNaturalist (Van Horn et al. 2018).

Recent efforts (Yang et al. 2018; Sun et al. 2018; Chen et al. 2019; Zheng et al. 2019; Ge, Lin, and Yu 2019) used only class labels to automatically locate informative regions. Specifically, Yang *et al.* (Yang et al. 2018) adapted a Navigator-Teacher-Scrutinizer system under a multi-stage scheme. Sun *et al.* (Sun et al. 2018) leveraged multiple channel attentions to learn several relevant regions. Wang *et al.* (Wang, Morariu, and Davis 2018) used a bank of convolutional filters to capture discriminative regions in the feature maps. Chen (Chen et al. 2019) deconstructed and re-constructed input images to find discriminative regions and features. Zheng *et al.* (Zheng et al. 2019) proposed trilinear attention sampling network to learn features from different details. Despite the fact that the above methods perform well, they generally need to be trained in multiple stages or learn high-dimension features, resulting in increased training times. Another recent work (Ge, Lin, and Yu 2019) developed a computationally complex, three-stage pipeline for fine-grained classification. Their framework requires a weakly supervised object detector, a mask-rcnn (He et al. 2017) based instance segmentation and an LSTM for capturing the context. Moreover, the mask-rcnn needs to be pretrained on an additional dataset: MS-COCO (Lin et al. 2014). Our proposed diversification block adopts a novel way to find more relevant features by suppressing the most prominent discriminative regions in class activation maps (Zhou et al. 2016) and thus forcing the network to find other informative regions. We note that hide-and-peek (Singh and Lee 2017) is related to ours, but largely different since our module works on feature maps and selectively suppresses discriminative regions. Our module is trained end-to-end with a computational cost nearly equal to the backbone.

Lately, FGVC strategies aimed to learn optimal classifiers on top of deep features have been proposed (Dubey et al. 2018a; 2018b). Qian *et al.* (Qian et al. 2015) employed a multi-stage framework which accepted pre-computed feature maps and learned the distance metric for classification. Dubey *et al.* (Dubey et al. 2018a) adapted the idea from pairwise learning and used Siamese-like neural network. A triplet loss was used in (Wang et al. 2016) to achieve better inter-class separation. The contrastive and triplet losses, however, increase the computational cost of training. (Dubey et al. 2018b) proposed a maximum entropy loss for fine-grained classification by using the principle of maximum-entropy. All above methods do not specifically focus on differentiating confusing classes. Further, all the negative categories for a given sample are considered as equal. Our proposed gradient-boosting loss solves the problem by explicitly focusing on hard classes, incurs no additional cost and provides faster convergence rates.

3 Method

In this section, we introduce our method which can be easily plugged into any classification network. As shown in Fig. 2, to deploy our approach, we need to replace global pooling

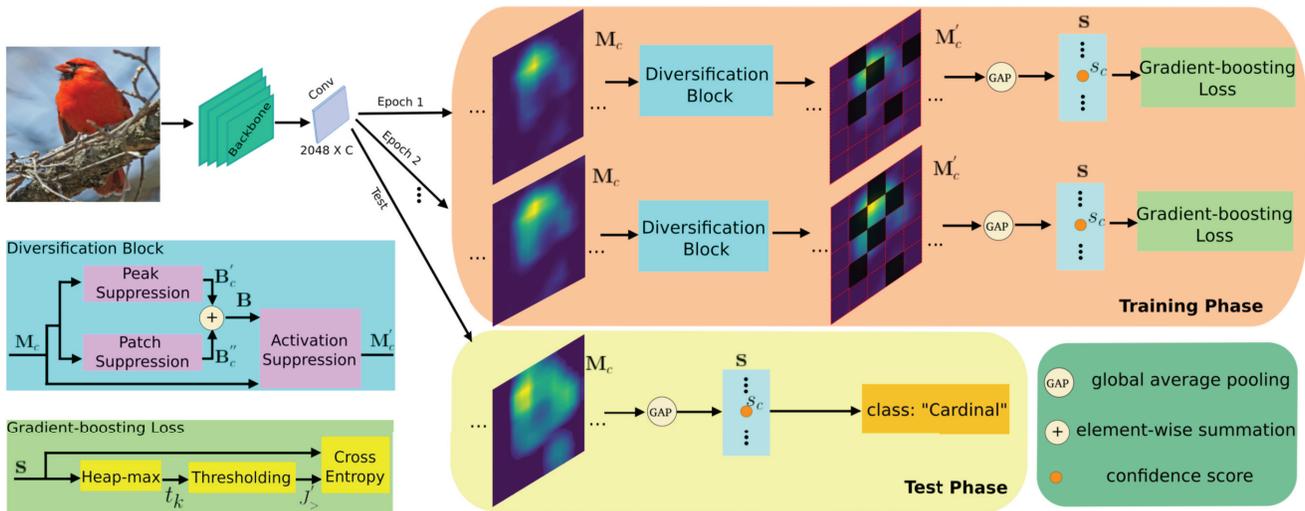


Figure 2: Overview of our overall architecture. Our method contains two novel components: diversification block and gradient-boosting loss. The diversification block suppresses the discriminative regions of the class activation maps, and hence the network is forced to find alternative informative features. The gradient-boosting loss focuses on difficult (confusing) classes for each image and boosts their gradient. As a result, the network moves swiftly (faster convergence) to discriminate the hard classes.

layer and the last fully connected layer of the backbone network with a 1×1 convolution having output channels equal to the number of classes. Our method includes two novel components: (a) A diversification module which forces the network to capture more subtle features, rather than only the most obvious ones; (b) A gradient boosting loss which trains the network to focus on highly confusing classes. These two components will be addressed in this section.

3.1 Diversification Block

Consider the multi-class image classification task with C classes as shown in Fig. 2. Let I be a training image with ground-truth label $l \in J$, where $J = \{1, 2, \dots, C\}$ is the label set containing all labels. The input to our diversification block is the category-specific activation maps $\mathbf{M} \in \mathbb{R}^{C \times H \times W}$, which is the output of the modified network. We denote $\mathbf{M} = \{\mathbf{M}_c : c \in [1, C]\}$, where $\mathbf{M}_c \in \mathbb{R}^{H \times W}$ is the individual activation map corresponding to c^{th} class. Here, H and W refer to the height and width of the output activation maps.

The basic idea of our diversification block is to suppress the discriminative regions of the activation map \mathbf{M} , so that the network is forced to look for other informative regions which is expected to enhance classification performance. In the following, we will target two relevant questions: (1) Where to suppress information? and (2) How to suppress?

Mask Generation Here, we explain the procedure to generate the mask that indicates the locations in \mathbf{M} that are suppressed. Let $\mathbf{B} = \{\mathbf{B}_c : c \in [1, C]\}$, where $\mathbf{B}_c \in \mathbb{R}^{H \times W}$ denotes the binary suppressing mask for its corresponding activation map \mathbf{M}_c . Each element in mask \mathbf{B}_c is in the domain $\{0, 1\}$, where 1 indicates the corresponding location will be suppressed while 0 means that no suppression will take place.

Peak suppression: First, we randomly suppress the peak locations of the activation maps because they are the most discriminative regions for the classifier. By suppressing the peaks, the network is forced to find alternative relevant regions in the image. Let $\mathbf{P}_c \in \mathbb{R}^{H \times W}$ be the peak map derived from c^{th} object category map (\mathbf{M}_c) such that:

$$\mathbf{P}_c(i, j) = \begin{cases} 1, & \text{if } \mathbf{M}_c(i, j) = \max(\mathbf{M}_c), \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Here, $\max(\mathbf{M}_c)$ denotes the maximum of matrix \mathbf{M}_c . We suppress the peaks of different object categories with probability p_{peak} . The masks \mathbf{B}'_c to randomly hide the peaks are generated as follows:

$$\mathbf{B}'_c = r_c * \mathbf{P}_c, \quad \text{where } r_c \sim \text{Bernoulli}(p_{peak}), \quad (2)$$

where $*$ denotes element-wise multiplication and r_c is a Bernoulli random variable that has p_{peak} probability of being 1.

Patch suppression: Peaks are the most discriminative regions, but there are other discriminative regions as well that encompass more subtle inter-class differences. In the following, we explain how to suppress locations other than peaks in the activation maps. We divide each \mathbf{M}_c into a grid of patches, where each fixed sized patch $\mathbf{M}_c^{[l, m]} \in \mathbb{R}^{G \times G}$ is indexed by row l and column m . Lets assume the set of all such patches on the grid is given by:

$$\mathbf{G}_c = \{\mathbf{M}_c^{[l, m]} : l \in [1, \frac{W}{G}], m \in [1, \frac{H}{G}]\}. \quad (3)$$

After this operation, the activation map \mathbf{M}_c will be divided into $(W \times H)/G^2$ patches. Let $\mathbf{B}''_c \in \mathbb{R}^{H \times W}$ be the mask for randomly hiding patches for c^{th} activation map \mathbf{M}_c . For each patch inside \mathbf{M}_c , we randomly hide it with probability

p_{patch} and set the elements of corresponding locations of \mathbf{B}_c'' as 1. Otherwise, the elements of \mathbf{B}_c'' are set to 0:

$$\mathbf{B}_c'' = \{\mathbf{B}_c''^{[l,m]} \in [0, 1] \sim \text{Bernoulli}(p_{patch})\}, \quad (4)$$

where, $\mathbf{0}, \mathbf{1} \in \mathbb{R}^{G \times G}$ and l, m are in the same range as Eq. 3. To consider only the non-peak locations, we then set the element of \mathbf{B}_c'' in the peak location of \mathbf{M}_c as 0,

$$\mathbf{B}_c''(i, j) = 0, \text{ if } \mathbf{M}_c(i, j) = \max(\mathbf{M}_c). \quad (5)$$

The final suppressing mask for c^{th} category is obtained as:

$$\mathbf{B}_c = \mathbf{B}_c' + \mathbf{B}_c''. \quad (6)$$

Activation Suppression Factor Setting values that replace the suppressed features is of much importance to achieve good performance. Let $\mathbf{M}' = \{\mathbf{M}'_c : c \in [1, C]\}$ represents the category activation maps obtained after our diversification module, which is generated as follows.

$$\mathbf{M}'_c(i, j) = \begin{cases} \mathbf{M}_c(i, j), & \text{if } \mathbf{B}_c(i, j) = 0, \\ \alpha * \mathbf{M}_c(i, j), & \text{if } \mathbf{B}_c(i, j) = 1, \end{cases} \quad (7)$$

where, α denotes the suppressing factor. Basically, we replace the values in the suppressing locations as α times of their initial values. In general, setting α to a low number will lead to good performance. Throughout our experiments, we set α as 0.1.

After feature masking, we perform global average pooling to get the confidence scores $\mathbf{s} \in \mathbb{R}^{1 \times C}$ as follows:

$$\mathbf{s} = \{s_c : c \in [1, C]\}, \quad s_c = \text{AvgPool}(\mathbf{M}'_c), \quad (8)$$

where, AvgPool denotes global average pooling.

3.2 Gradient-boosting Cross Entropy Loss

While diversification module aims at finding more subtle variations in the input images, our second contribution is a gradient-boosting loss function that specifically focuses on confusing classes to avoid misclassifications between them. We elaborate the proposed loss function below.

Loss Function The most widely used loss for image classification is cross entropy (CE) loss. For an image I , CE loss can be written as follows:

$$\text{CE}(\mathbf{s}, l) = -\log \frac{\exp(s_l)}{\sum_{i \in J} \exp(s_i)}, \quad (9)$$

where l is the ground-truth label for image I . Here, the loss considers all negative classes equally. However, in fine-grained classification, the ground-truth class is generally much closer to a related subset of classes than others. For example, in CUB-200-2011 (Wah et al. 2011), bird class of *Acadian Flycatcher* is more closer to categories such as *Great Crested Flycatcher*, *Least Flycatcher*, *Olive sided Flycatcher* and other kinds of *Flycatcher*, since they all belong to the same species. As a result, the network is prone to making mistakes among these similar (thus confusing) classes and predicting relatively higher confidence scores for them. Based on this observation, we argue that the loss

should focus more on the confusing classes, rather than simply considering all negative classes equally for the normalization in Eq. 9. Hence, we propose a novel and simple gradient-boosting cross entropy (GCE) loss which focuses only on k negative classes with top- k highest confidence scores among all negative classes. Here, k simply means the number of negative classes to focus on. We will show in the next section, that the proposed loss basically boosts gradients to more swiftly resolve ambiguities between closely related confusing classes.

We define J' as the set of all negative classes, where $J' = \{i : i \in [1, C] \wedge i \neq l\}$. Let $\mathbf{s}' = \{s_i, i \in J'\}$ be the set containing confidence scores of all negative classes. We get the k^{th} highest values of \mathbf{s}' by heap-max algorithm (Chhavi 2018) and denote it as t_k . Next, we split J' into $J'_>$ and $J'_<$ by thresholding \mathbf{s} using t_k , defined as follows:

$$J'_> = \{i : i \in J \wedge s_i \geq t_k\} \quad (10)$$

$$J'_< = \{i : i \in J \wedge s_i < t_k\}, \quad (11)$$

where, $J'_>$ contains the negative classes whose confidence scores are within the top- k of all negative classes, and $J'_<$ is the set of negative classes whose confidence scores rank below the top- k classification scores.

Instead of considering all negative classes in Eq. 9, our gradient-boosting cross entropy loss only focuses on confusing classes ($J'_>$). The negative classes in $J'_<$ do not contribute to the loss since the network can easily distinguish them from the ground-truth class. Our proposed loss is given by:

$$\text{GCE}(\mathbf{s}, l) = -\log \frac{\exp(s_l)}{\exp(s_l) + \sum_{i \in J'_>} \exp(s_i)}. \quad (12)$$

As shown in Eq. 10 and 12, GCE loss focuses only on $J'_>$, containing k negative classes with top- k highest confidence scores. Here, k is a hyper-parameter (we found $k = 15$ works best in our experiments). When $k = C$, GCE is equivalent to CE.

In the following analysis, we will show how our loss can boost the gradient for both the ground-truth class and confusing negative classes.

Gradient Boosting We analyze the loss from the perspective of gradient. For the original cross entropy (CE) loss, the gradient for s_c is computed as:

$$\frac{\partial \text{CE}(\mathbf{s}, l)}{\partial s_c} = \begin{cases} \frac{\exp(s_c)}{\sum_{i \in J} \exp(s_i)}, & c \neq l \\ \frac{\exp(s_c)}{\sum_{i \in J} \exp(s_i)} - 1, & c = l \end{cases} \quad (13)$$

For our gradient-boosting cross entropy loss, the gradient for s_c is computed as:

$$\frac{\partial \text{GCE}(\mathbf{s}, l)}{\partial s_c} = \begin{cases} \frac{\exp(s_c)}{\exp(s_l) + \sum_{i \in J'_>} \exp(s_i)}, & c \in J'_> \\ \frac{\exp(s_c)}{\exp(s_l) + \sum_{i \in J'_>} \exp(s_i)} - 1, & c = l \end{cases} \quad (14)$$

From our definition in Eq. 10 and Eq. 11, the following relation exists between $J'_>$ and J' ,

$$J'_> + \{l\} \subset J' + \{l\} = J. \quad (15)$$

As such, we obtain,

$$\frac{\partial \text{GCE}(s, l)}{\partial s_c} > \frac{\partial \text{CE}(s, l)}{\partial s_c}. \quad (16)$$

We can see that for both the ground-truth class and confusing negative classes, the gradient of our proposed loss is larger than the gradient of the original cross entropy loss. With our novel loss, the network can focus on differentiating difficult classes from the ground-truth class and converge faster, which is validated by our experiments.

3.3 Training and Inference

Our method is trained end-to-end in a single stage. The diversification block is only used during the training phase. As shown in Fig. 2, during the training phase, class activation maps are passed through our novel diversification block and then to the global average pooling. As a result, discriminative regions are randomly masked and the network is forced to find other relevant areas. During test phase, the whole class activation maps are passed to global average pooling directly, without being suppressed at any region so that all informative regions found during training phase contribute to the final confidence score.

4 Experiments

4.1 Datasets

We comprehensively evaluate our algorithm on CUB-200-2011 (Wah et al. 2011), Stanford Cars (Krause et al. 2013), FGVC Aircraft (Maji et al. 2013), and Stanford Dogs (Khosla et al. 2011), all of which are widely used for fine-grained recognition. Statistics of all datasets are shown in Table 1. We follow the same train/test splits as in the table. For evaluation metric, we use top-1 accuracy following (Sun et al. 2018; Dubey et al. 2018a; 2018b).

Furthermore, we also evaluate on the recent terrain dataset for terrain recognition: GTOS-mobile (Xue, Zhang, and Dana 2018) dataset and GTOS (Ground Terrain in Outdoor Scenes) (Xue et al. 2017) dataset, which have potential use for autonomous agents (automatic car). The datasets are large-scale, containing classes of outdoor ground terrain, i.e. glass, sand, soil, stone-cement, and so on. Since those terrain classes are closely related, visually similar and thus difficult to classify, we use this challenging dataset to evaluate our method. Following (Xue, Zhang, and Dana 2018), we use GTOS as training and GTOS-mobile as test.

4.2 Implementation Details

For fair comparisons with other methods (Yang et al. 2018; Wang, Morariu, and Davis 2018), we use an input image resolution of 448×448 in all experiments. We fine-tune pre-trained network (ResNet-50 (He et al. 2016)) using our proposed diversification block and gradient-boosting loss due to its popularity in existing works. Momentum SGD optimizer

Dataset	#Class	#Train	#Test
CUB-200-2011	200	5,994	5,794
Stanford Cars	196	8,144	8,041
FGVC Aircraft	100	6,667	3,333
Stanford Dogs	120	12,000	8,580
GTOS-mobile	31	31,315	6,066

Table 1: Five commonly used benchmarks.

is used with an initial learning rate of 0.001, which decays by 0.1 for every 50 epochs. We set weight decay as 10^{-4} . Our algorithm is implemented using Pytorch (Paszke et al. 2017) using two Tesla V100 GPU.

4.3 Quantitative Results

Our method does not require any part-annotation and can be trained using only class labels. Moreover, it is parameter-free and does not increase the number of parameters compared to the ResNet-50 backbone. Our results are compared with the most recent and top-performing approaches evaluated under similar experimental setting. Several approaches such as RACNN (Fu, Zheng, and Mei 2017), RAM (Li et al. 2017), and NTS-net (Yang et al. 2018) extract multiple crops at different scale from an input image. The classification score obtained from these crops are averaged to predict the final class during inference. For fair comparison, we report our ‘multi-scale’ (five crops) results, in addition to the ‘single-scale’ using one crop from an image.

The comparisons with various methods on four challenging fine-grained datasets, namely CUB-200-2011 (Wah et al. 2011), FGVC Aircraft (Maji et al. 2013), Stanford Cars (Krause et al. 2013), and Stanford Dogs (Khosla et al. 2011), are shown in Table 2. Additionally, results for GTOS-mobile (Xue et al. 2017) are shown in Table 3. Overall, our proposed method outperforms previous methods on *all* five datasets.

We observe that our method achieves the best accuracy on birds classification task (Table 2). Specifically, our method obtains an accuracy of 88.6% which outperforms TASN (87.6%) (Zheng et al. 2019). TASN (Zheng et al. 2019) performs well because it first uses a small network to find the attentive regions and then distills knowledge from various informational regions to the model. With a low parametric complexity, our method can capture more relevant regions by focusing on hard classes and diversifying informative areas in the class activation maps.

For other four datasets, our method also outperforms the compared methods. In Aircraft, we achieve 93.5% top-1 accuracy, surpassing NTS-net (Yang et al. 2018) (91.4%). In Cars, we obtain 94.9%, outperforming the best performances: 93.8% of TASN (Zheng et al. 2019). In Dogs, we obtain 87.7% top-1 accuracy compared to 87.3% obtained by RACNN approach (Fu, Zheng, and Mei 2017). Note that RACNN has much more parameters (429M) than our methods (23.9M). In GTOS-mobile, we show our result using ResNet-50 with ‘single scale’, for fair comparison with Deep-TEN (Zhang, Xue, and Dana 2017) and DEP (Xue, Zhang, and Dana 2018). We get 85.0%, which is 2.8% better than the current state-of-the-art 82.2%.

Methods	Backbone	Resolution	#Parameters	Accuracy			
				CUB-200-2011	Aircrafts	Cars	Dogs
RACNN (Fu, Zheng, and Mei 2017)	VGG-19	448	429M	85.3	88.2	92.5	87.3
RAM (Li et al. 2017)	ResNet-50	448	>23.9M	86.0	-	93.1	-
MACNN (Zheng et al. 2017)	VGG-19	448	144M	86.5	89.9	92.8	-
MAMC (Sun et al. 2018)	ResNet-50	448	434M	86.3	-	93.0	85.2
MaxEnt (Dubey et al. 2018b)	ResNet-50	-	23.9M	86.5	89.8	93.9	83.6
PC (Dubey et al. 2018a)	ResNet-50	-	23.9M	86.9	89.2	93.4	83.8
DFL-CNN (Wang, Morariu, and Davis 2018)	ResNet-50	448	26.3M	87.4	91.7	93.1	-
NTS-net (Yang et al. 2018)	ResNet-50	448	25.5M	87.5	91.4	93.9	-
TASN (Zheng et al. 2019)	ResNet-50	448	35.2M	87.6	-	93.8	-
Ours (single scale)	ResNet-50	448	23.9M	87.7	92.1	94.3	87.1
Ours (multi scale)	ResNet-50	448	23.9M	88.6	93.5	94.9	87.7

Table 2: Experimental results on four standard datasets. “-” means the information is not mentioned in the relevant paper. Our method outperforms existing approaches on four commonly used fine-grained datasets, and requires no additional parameters compared to the ResNet-50 backbone. Here, the parameters are computed on CUB-200-2011, having 200 output classes.

Methods	Accuracy
B-CNN (Lin, RoyChowdhury, and Maji 2015)	75.4
Deep-TEN (Zhang, Xue, and Dana 2017)	76.1
DEP (Xue, Zhang, and Dana 2018)	82.2
Ours (single scale)	85.0

Table 3: Experimental results on GTOS-mobile.

4.4 Ablation Study

To fully analyze our method, Table 4 provides a detailed ablation analysis on the key components of our method. It basically highlights the importance of diversification block and gradient-boosting loss. We conduct all ablation studies on CUB-200-2011 using the ResNet-50 (He et al. 2016).

Diversification block (DB). DB is important because it diversifies the informative regions by forcing the network to find relevant parts other than the most obvious ones. Integrating DB block in the ResNet-50 backbone results in a performance improvement of 0.8% (from 85.5% to 86.3%).

Methods	Accuracy
ResNet-50	85.5
ResNet-50+DB	86.3
ResNet-50+DB+Center loss (Wen et al. 2016)	86.4
ResNet-50+DB+LGM loss (Wan et al. 2018)	86.5
ResNet-50+DB+MaxEnt (Dubey et al. 2018b)	86.2
Ours (single scale)	87.7
Ours (multi scale)	88.6

Table 4: Ablation analysis on the CUB-200-2011. Our diversification block (DB) and gradient-boosting loss provide progressive improvements over the baseline.

Gradient-boosting loss. Our gradient-boosting loss is another important component that shows significant improvement. Using this loss, we improve the results from 86.3% to 87.7% (an absolute gain of 1.4%). We also compare our loss with other recent losses that aim at achieving better discriminability: Center loss (Wen et al. 2016), LGM loss (Wan et al. 2018), and max entropy loss (Dubey et al. 2018b). The re-

α	0	0.1	0.2	1.0
Accuracy	86.2	86.3	85.8	85.5

Table 5: Ablation study on suppressing factor α . Keeping suppressing factor as small leads to good performance.

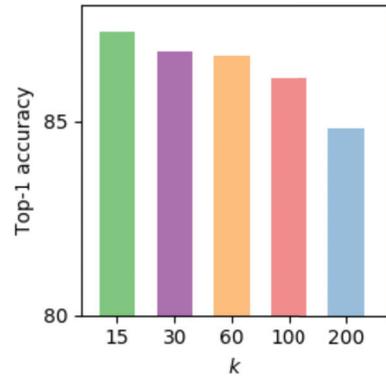


Figure 3: Ablation study on k for our loss in CUB-200-2011.

sults show that gradient-boosting loss outperforms all these loss functions. Our loss targets on difficult/confusing classes and selectively boosts the gradients for them, while other losses consider all negative classes as equal.

Suppressing Factor. Here, we show a parameter sensitivity analysis on the suppressing factor α . Top-1 accuracy with respect to different α settings is shown in Table 5. It shows that keeping α as a small value consistently leads to better performance than without using diversification block ($\alpha = 1$). Specifically, $\alpha = 0.1$ gives the best performance on CUB-200-2011 dataset.

Choices of k . Here, we show ablation study on k , the number of negative classes to focus on for gradient-boosting loss, in Fig. 3. It shows that by reducing k , our loss focuses on more confusing classes and achieves consistent improvement in top-1 accuracy.

Convergence Analysis. We compare the training curves of our methods and baseline (ResNet-50) in Fig. 5. It shows

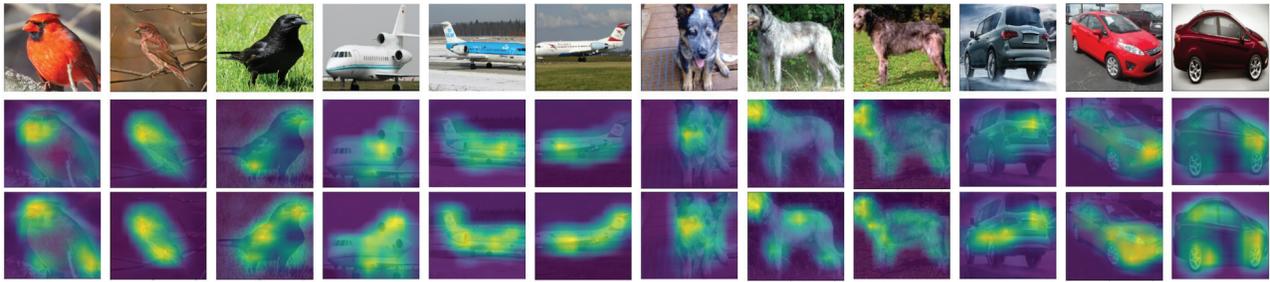


Figure 4: Class activation map (CAM) comparison between our method and baseline in different datasets. *Top to below*: original image, CAM of the ground-truth class of baseline, CAM of the ground-truth class of our method. While baseline only focuses on the most discriminative region, our method accurately diversifies attentions to other informative regions of the objects.

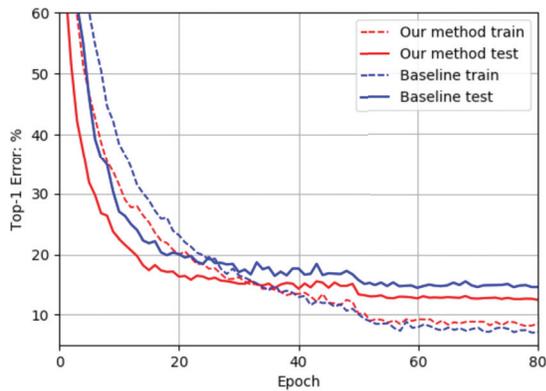


Figure 5: Training curves of our methods and the baseline (CE loss) on CUB-200-2011. Using our loss, our method converges faster and performs better than the baseline.

that our method converges much faster than the baseline, and also attains a lower error rate on test set. Remarkably, the baseline achieves a lower error rate on training set after 50 epochs, but fails to generalize well to the test set. This shows that the baseline is prone to overfitting on the train set, which our method successfully avoids. Our method uses diversification block which prevents the network to only focus on the most discriminative regions, like the beak or head of a bird. In contrast, using diversification block, the network finds various informative areas, thus reducing overfitting.

4.5 Qualitative Results

We qualitatively illustrate the comparison between our approach and the baseline in Fig. 4. We note that the diversification block indeed helps the network to find more discriminative regions in the image. In contrast, the baseline model generally focuses on the most obvious distinguishing patterns and its attention is limited to only a limited set of spatial locations. This explains why our approach generalizes better to test images, attaining a higher accuracy (Fig. 5).

4.6 ImageNet Results

To validate the generality of gradient-boosting loss in visual recognition, we apply it to a ResNet-50 backbone on Im-

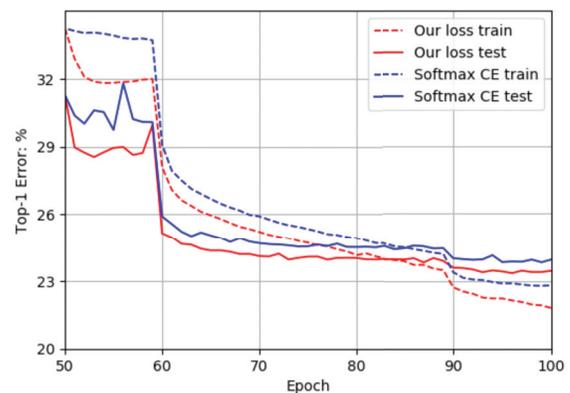


Figure 6: Training curves of using our loss and Cross Entropy (CE) on ImageNet. Our loss converges considerably faster than CE and also leads to a better performance.

geNet (Russakovsky et al. 2015). Here, we use the input size of 224×224 and follow the same training strategy as used in (He et al. 2016). Since our loss focuses on difficult classes, we apply it only half way (50 epochs) during training when easy categories are already well-classified. The comparison of training curve between using our loss and using cross entropy (CE) is shown in Fig. 6. It shows that the proposed loss converges much faster than the CE loss and achieves a lower error rate on the challenging ImageNet benchmark.

5 Conclusion

We proposed a novel approach to better discriminate closely related categories in fine-grained classification task. Our method has two novel components: (a) diversification block that forces the network to find subtle distinguishing features between each pair of classes and (b) gradient-boosting loss that specifically focuses on maximally separating the highly similar and confusing classes. Our approach not only outperforms existing methods on all studied fine-grained datasets, but also demonstrates much faster convergence rates. In comparison to previous methods, our solution is both simple and elegant, leads to higher accuracy and demonstrates better computational efficiency.

References

- Berg, T., and Belhumeur, P. 2013. Poof: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation. In *CVPR*.
- Chen, Y.; Bai, Y.; Zhang, W.; and Mei, T. 2019. Destruction and construction learning for fine-grained image recognition. In *CVPR*.
- Chhavi. 2018. k largest(or smallest) elements in an array-added min heap method.
- Cui, Y.; Song, Y.; Sun, C.; Howard, A.; and Belongie, S. 2018. Large scale fine-grained categorization and domain-specific transfer learning. In *CVPR*.
- Dubey, A.; Gupta, O.; Guo, P.; Raskar, R.; Farrell, R.; and Naik, N. 2018a. Pairwise confusion for fine-grained visual classification. In *ECCV*.
- Dubey, A.; Gupta, O.; Raskar, R.; and Naik, N. 2018b. Maximum-entropy fine grained classification. In *NIPS*.
- Fu, J.; Zheng, H.; and Mei, T. 2017. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In *CVPR*.
- Gao, Y.; Beijbom, O.; Zhang, N.; and Darrell, T. 2016. Compact bilinear pooling. In *CVPR*.
- Ge, W.; Lin, X.; and Yu, Y. 2019. Weakly supervised complementary parts models for fine-grained image classification from the bottom up. In *CVPR*.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.
- He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. B. 2017. Mask r-cnn. *ICCV*.
- Khosla, A.; Jayadevaprakash, N.; Yao, B.; and Fei-Fei, L. 2011. Novel dataset for fine-grained image categorization. In *CVPR Workshop*.
- Krause, J.; Stark, M.; Deng, J.; and Fei-Fei, L. 2013. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition*.
- Li, Z.; Yang, Y.; Liu, X.; Zhou, F.; Wen, S.; and Xu, W. 2017. Dynamic computational time for visual attention. In *ICCV*.
- Li, P.; Xie, J.; Wang, Q.; and Gao, Z. 2018. Towards faster training of global covariance pooling networks by iterative matrix square root normalization. In *CVPR*.
- Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *ECCV*.
- Lin, T.-Y.; RoyChowdhury, A.; and Maji, S. 2015. Bilinear cnn models for fine-grained visual recognition. In *ICCV*.
- Maji, S.; Kannala, J.; Rahtu, E.; Blaschko, M.; and Vedaldi, A. 2013. Fine-grained visual classification of aircraft. Technical report.
- Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in pytorch. In *NIPS Workshop*.
- Qian, Q.; Jin, R.; Zhu, S.; and Lin, Y. 2015. Fine-grained visual categorization via multi-stage metric learning. In *CVPR*.
- Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *IJCV*.
- Singh, K. K., and Lee, Y. J. 2017. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *ICCV*.
- Sun, M.; Yuan, Y.; Zhou, F.; and Ding, E. 2018. Multi-attention multi-class constraint for fine-grained image recognition. In *ECCV*.
- Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *CVPR*.
- Van Der Maaten, L. 2014. Accelerating t-sne using tree-based algorithms. *JMLR*.
- Van Horn, G.; Mac Aodha, O.; Song, Y.; Cui, Y.; Sun, C.; Shepard, A.; Adam, H.; Perona, P.; and Belongie, S. 2018. The inaturalist species classification and detection dataset. In *CVPR*.
- Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The caltech-ucsd birds-200-2011 dataset.
- Wan, W.; Zhong, Y.; Li, T.; and Chen, J. 2018. Rethinking feature distribution for loss functions in image classification. In *CVPR*.
- Wang, Y.; Choi, J.; Morariu, V.; and Davis, L. S. 2016. Mining discriminative triplets of patches for fine-grained classification. In *CVPR*.
- Wang, Y.; Morariu, V. I.; and Davis, L. S. 2018. Learning a discriminative filter bank within a cnn for fine-grained recognition. In *CVPR*.
- Wen, Y.; Zhang, K.; Li, Z.; and Qiao, Y. 2016. A discriminative feature learning approach for deep face recognition. In *ECCV*.
- Xue, J.; Zhang, H.; Dana, K.; and Nishino, K. 2017. Differential angular imaging for material recognition. In *CVPR*.
- Xue, J.; Zhang, H.; and Dana, K. 2018. Deep texture manifold for ground terrain recognition. In *CVPR*.
- Yang, Z.; Luo, T.; Wang, D.; Hu, Z.; Gao, J.; and Wang, L. 2018. Learning to navigate for fine-grained classification. In *ECCV*.
- Zhang, H.; Xue, J.; and Dana, K. 2017. Deep ten: Texture encoding network. In *CVPR*.
- Zheng, H.; Fu, J.; Mei, T.; and Luo, J. 2017. Learning multi-attention convolutional neural network for fine-grained image recognition. In *ICCV*.
- Zheng, H.; Fu, J.; Zha, Z.-J.; and Luo, J. 2019. Looking for the devil in the details: Learning trilinear attention sampling network for fine-grained image recognition. In *CVPR*.
- Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A. 2016. Learning deep features for discriminative localization. In *CVPR*.