Learning a Mixture of Granularity-Specific Experts for Fine-Grained Categorization

Lianbo Zhang[1], Shaoli Huang*[2], Wei Liu[1], and Dacheng Tao[2]

[1]Advanced Analytics Institute, School of Computer Science, FEIT, University of Technology Sydney, Chippendale, NSW, Australia
[2]UBTECH Sydney AI Centre, School of Computer Science, FEIT, University of Sydney, Darlington, NSW 2008, Australia
{lianbo.zhang@student., wei.liu@}uts.edu.au {shaoli.huang, dacheng.tao}@sydney.edu.au

Abstract

We aim to divide the problem space of fine-grained recognition into some specific regions. To achieve this, we develop a unified framework based on a mixture of experts. Due to limited data available for the fine-grained recognition problem, it is not feasible to learn diverse experts by using a data division strategy. To tackle the problem, we promote diversity among experts by combing an expert gradually-enhanced learning strategy and a Kullback-Leibler divergence based constraint. The strategy learns new experts on the dataset with the prior knowledge from former experts and adds them to the model sequentially, while the introduced constraint forces the experts to produce diverse prediction distribution. These drive the experts to learn the task from different aspects, making them specialized in different subspace problems. Experiments show that the resulting model improves the classification performance and achieves the state-of-the-art performance on several fine-grained benchmark datasets.

1. Introduction

Fine-grained visual categorization such as animal breeds recognition [10, 16, 27, 21] aims to identify under subcategories of given images. Objects in fine-grained tasks usually share small inter-class variance and large intra-class variance along with multiple object scale and complex background, leading to a more complex problem space.

In this paper, we tend to divide the fine-grained problem space into subspace problems. To this end, we develop a unified framework based on a mixture of neural network experts (ME) [9, 19, 1]. The neural network-based ME usually follows a scheme of partition and conquer, where the problem space is divided into sub-spaces. Examples like [13, 12] have been investigated on fine-grained task, but these methods focus on learning experts from a set of unique subsets, as is the case with conventional ME methods. The strategy

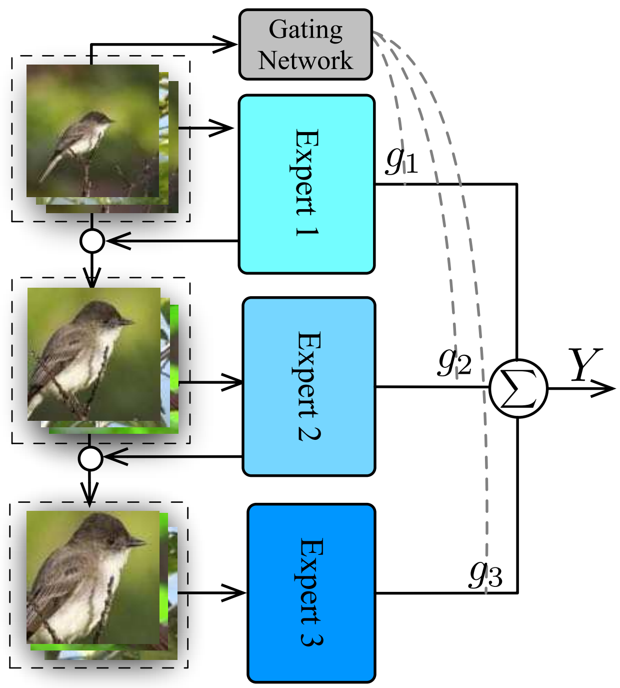

Figure 1. Overview of our framework, which consists of several experts and a gating network. Each expert learns with prior knowledge from the previous expert. The gating network determines the contribution of each expert to the final predictions.

*Corresponding author.

to learn diverse experts from a set of unique subsets is not feasible for the fine-grained task, as fine-grained training data is usually limited. If further dividing such data into subsets for training, each resulting expert model is more prone to over-fitting due to the less amount of data available.

To overcome the difficulty of learning diverse experts from limited data, we introduce a gradually enhanced strategy along with a Kullback-Leibler (KL) divergence constraint to encourage diversity among experts. The main idea of gradually enhancing is that a new expert is learned with extra informative knowledge or prior information obtained from the former expert, and therefore more specialized to the problem. Based on this, the first thing to consider is how an expert passes some task-related knowledge to the latter expert. In this work, we select attention maps from ConvNet model as a kind of carriers for such knowledge since it indicates how the neural network relates some certain regions of the image to the target task. Also, recent works [3, 29] show that attention maps reside semantic cues and can be used for visual interpretation and weakly supervised object detection.

Another explicit way to promote diversity among experts is to penalize the similarity of probability distributions. This can be simply implemented by maximizing the KL divergence between the probability distribution of experts. However, due to the limited training data for the fine-grained classification task, each expert tends to produce a vector close to one-hot. Such a result does not reflect the model's description of the inherent structure of the data. Therefore, we introduce a penalizing term that penalizes the similarity of the predicted distribution after excluding the maximum. By zeroing the maximum score and normalizing, the resulting output can better reflect the model's description of the data (such as the relationship between data and categories). Thus, maximizing the KL-divergence of two such distributions is equivalent to encouraging the two models to have different descriptions of the data. By learning with the gradually enhanced strategy and the penalizing term, our proposed methods can learn diversified experts from limited training data, which is beneficial for improving the performance in the fine-grained classification tasks.

The contributions of this paper are summarized as follows:

- we propose a gradually enhanced strategy that allows learning diversified ConvNet experts from limited training data.
- we introduce a novel constraint that is effective in promoting model diversity.
- we present a network architecture (MGE-CNN) that achieves the state-of-the-art performance on several challenging fine-grained datasets.

The rest of the paper is organised as follows. Section 2 describes the related works, and section 3 illustrates the proposed method in detail. Section 4 introduces the implementations and experimental results, followed by the conclusion in section 5.

2. Related Works

Fine-grained classification. Deep learning based methods has made significant progress in recent years [28, 35, 14, 36, 5, 4, 47, 43], especially in the field of fine-grained classification [6, 39, 37, 40, 45, 41]. One line of work [24, 11, 20] has concentrated on feature encoding. Lin *et al*. propose a bilinear pooling method [24] that computes local pairwise feature interactions from two CNN branches (shared or not shared). Despite the impressive performance, the high-dimensions of bilinear features make it challenging to optimize. Recent works improve bilinear methods using compact bilinear representation [11] with kernel method, or low-rank bilinear pooling [20] by representing the covariance features as a matrix and applying a low-rank bilinear classifier, which allows for a large reduction in computation time as well as decreasing the effective number of parameters to be learned.

Another line of work has focus on extracting discriminative part features in a weakly supervised way. To avoid using extensive annotations, Xiao *et al*. [38] apply a part-level top-down attention and combine candidates proposal attention, object-level attention to train domain-specific deep nets. Zhang *et al*. [44] propose to elaborately pick deep filters as part detectors before encoding them to final representation. Spatial transformer networks [18] perform transformation on entire feature map to allows networks to select the most relevant (attention) region. RA-CNN (Recurrent Attention CNN) [10] recursively learns discriminative region attention and region-base feature representation at multiple scale in a mutually reinforced way. MA-CNN (Multi-Attention CNN) [45] groups feature channels through clustering to generate multiple parts. Such part-based methods have become dominant in the field of fine-grained classification. Our proposed method differs from these methods in that we address the problem by leaning diversified ConvNet-based experts. More specifically, we proposed a gradually enhanced strategy and a penalizing term to promote model diversity when learning from limited data. Our experiments shows that the proposed method outperforms the state-of-the-art part-based methods.

Mixture of experts is established mainly based on divide-and-conquer principle [17, 9, 19, 1], in which the problem space is divided to be addressed by specialized experts. Recently proposed frameworks [30, 13, 12] in this field mainly consist of neural network (NN) experts and a gating network. These models focus on training each expert on a unique subset of given data. Since a deep neural

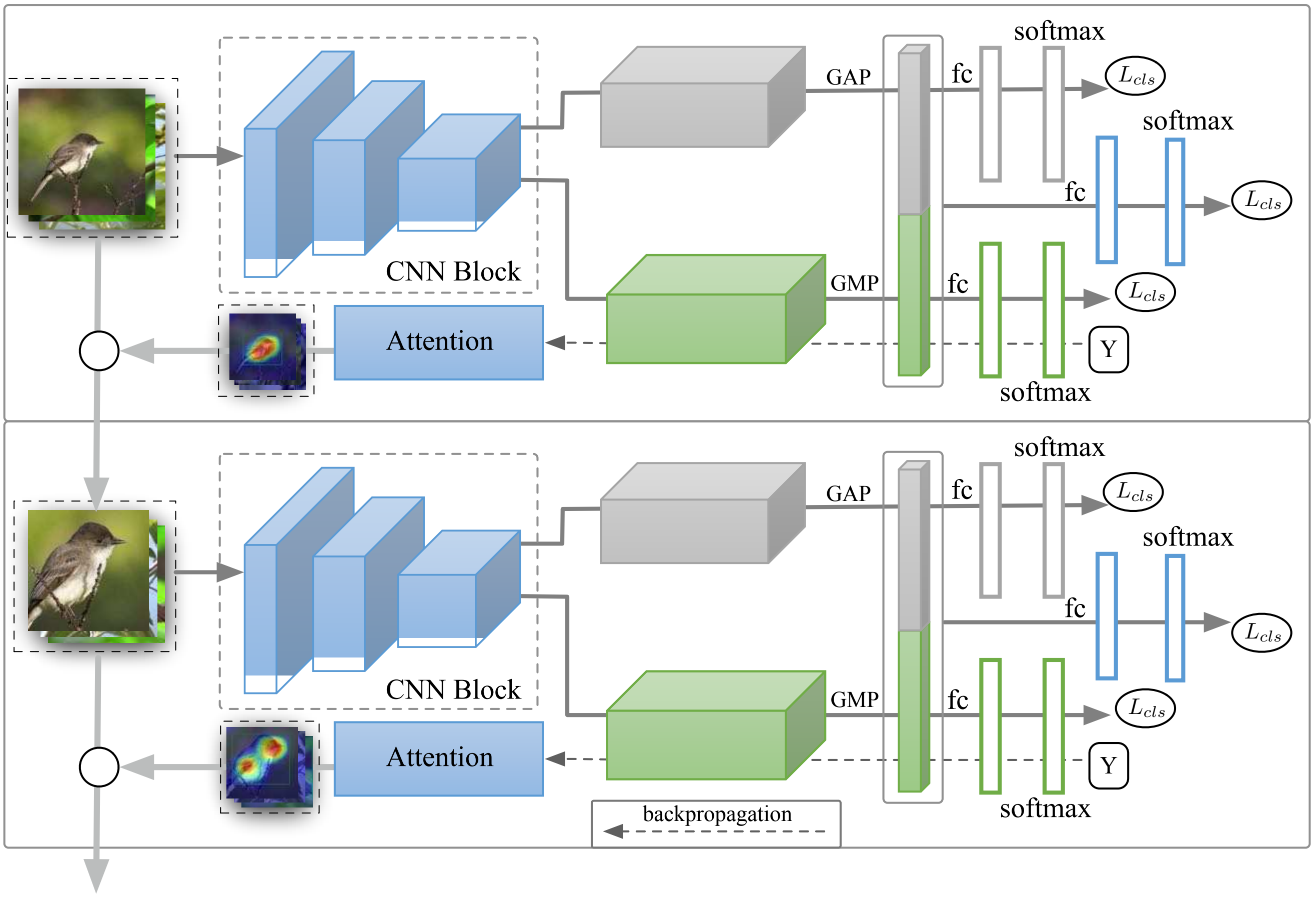

Figure 2. Network structure. The proposed MGE-CNN consists of several expert sub-networks, each of which contains a feature representation learning and attention region extraction component. The first component uses two different Conv blocks with different pooling methods on top of a shared Conv block to extract different types of feature and then concatenate them to form the overall representation. The second one is the gradient-based attention module, which is used to extract attention region and transform the training data into a new one for the following expert.

network can have millions of parameters, training a neural network requires massive amounts of data, and if we do data partitioning, it will cause serious overfitting, leading to poor performance on test data. Our method is different from these methods in two ways. First, the expert network can extract small and large part feature, which is specially designed for fine-grained classification problem. Further, we bypass the need of data division and propose a gradually enhanced strategy that allows training each expert on the full-size data yet promotes their diversity.

3. Approach

Our approach consists of several experts and a gating network. These experts are learned to be diversified by combining a gradually-enhanced learning strategy and a KL-divergence based penalizing term. The gating network is then used to combine experts for making the final decision.

We design our experts following two principles. The first one is that in order to better perform fine-grained recognition, we need to learn a good representation, and this representation needs to contain more detailed information. To achieve this, we extract both large-part features and small-part features, and each expert makes decision based on the combination of these two features. The second principle is that one expert can produce prior knowledge to build another expert. All experts can generate good but diversified predictions. To encourage diversity among experts, experts are trained in progressive enhanced way, and we feed experts with data that contains prior knowledge from the previous expert.

3.1. Experts for Fine-Grained Recognition

To meet the principles mentioned above, we need to build a strong feature extractor. For expert E_t, we use a deep Conv block with global average pooling to extract fea-

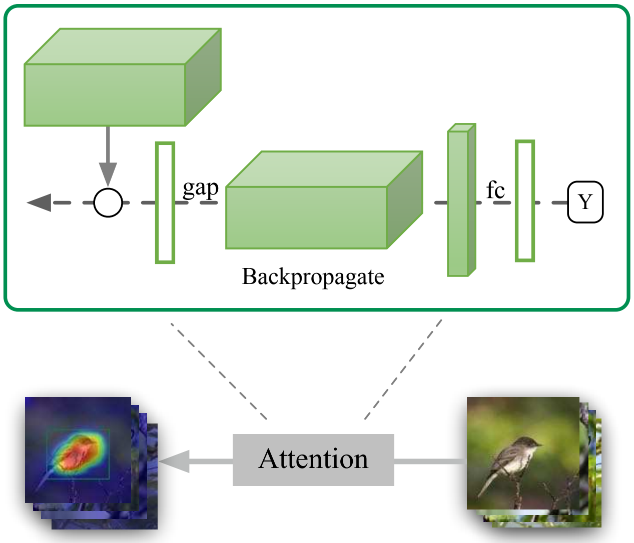

Figure 3. Attention module. We back-propagate gradients from ground-truth (predictions at test time) to obtain gradient of last convolutional layer. The gradient is then global average pooled and weighted summarized with feature maps along channel to get attention maps. The attention maps provide prior knowledge for latter expert.

tures from large-part region f_g^t, and a shallow Conv block with global max pooling[37] is used to extract features from small-part region f_l^t. By applying different global pooling methods (GAP and GMP) on two separate Conv blocks, they will learn different types of features from the same image. The unified feature f^t for the expert can be obtained by concatenating these two normalized features.

$$f^t = (\frac{f_g^t}{\| f_g^t \|_2}, \frac{f_l^t}{\| f_l^t \|_2})$$

The classification loss for an expert consists two auxiliary losses (large part and small part) and one decision making loss,

$$L_{cls}^t = -\frac{1}{N} \sum_{\theta_j \in \{\theta_g^t, \theta_l^t, \theta_c^t\}} \sum_{i=1}^{N} y_i \log(f(x_i^t, \theta_j))$$

where x_i^t is the input to expert E_t with class label y_i, and $\theta_g^t, \theta_l^t, \theta_c^t$ denote the parameters of in large region, small region, concatenate branch respectively. N is the total amount of data. All three losses are based on cross entropy.

Latter expert learns from data with prior information from the previous expert, and the prior knowledge is passed to latter experts through gradient based attention. The way we construct attention map follows Grad-CAM [29] which uses the gradient information of desired convolution layer to understand the importance of each neuron on decision of interest. To obtain the class specific attention map of width u and height v for any class c, we first compute the gradient for class c, denoted as y^c, with respected to feature map A^k of a convolution layer, *i.e.* $\frac{\partial y^c}{\partial A^k}$. These gradients that flows back are then global average-pooled to obtain the neuron importance α_k^c :

$$\alpha_k^c = \frac{1}{Z} \sum_{i=1}^{u} \sum_{j=1}^{v} \frac{\partial y^c}{\partial A_{ij}^k}$$

where the weight α_k^c denotes a partial linearization of the deep networks downstream from the activations of desired convolutional layer A, and captures the importance of feature map k for a target class c. Z is the number of neuron $(u \times v)$ in a channel and k is the channel number in layer A.

A ReLU operation is applied to the gradient before global pooling to leverage channel importance.

$$\beta_k^c = \frac{1}{Z} \sum_{i=1}^{u} \sum_{j=1}^{v} ReLU(\frac{\partial y^c}{\partial A_{ij}^k})$$

The class activation map can be constructed by performing a weighted summation of forward activation maps across channels from the desired convolutional layer. In the train phase we use ground-truth label and during test time, we use predicted class labe.

As a result, the final attention map in expert E_t can be expressed as:

$$S^c = \sum_{k=1}^{K} \beta_k^c A^k$$

After getting the attention map, we further normalize it by scaling the value between 0 and 1. Then, we can utilize a threshold to estimated bounding box for locating the significant region in the image.

$$S_{norm}^c = \frac{S^c - min(S^c)}{max(S^c) - min(S^c)}$$

By up-sampling the attention map to the size of the input image, we can identify the image regions that is most relevant to the class label.

In the training stage, we back-propagate ground-truth predictions (corresponding to category labels) to compute attention maps, while in the test phase, since we have no access to category labels, we use the predicted label.

Given attention map we construct input for next expert using technique similar to weakly supervised object localization [46, 22, 29]. One reason to do so is to include more effective regions instead of only detecting part regions. To achieve this, we fist segment the regions of which the value is above 0.2 of the max value of the attention map, which has been rescaled to between 0 and 1. Then we take the bounding box that covers the largest connected areas in segmentation map. Through this, we obtain a coarse bounding box. After that, we remap the coordinates of the bounding box to the original images, and then crop the corresponding region before zooming to original size.

3.2. KL-Divergence based Penalizing Term

To promote more diversity among experts, we introduce a KL-Divergence based constraint to penalize experts that produce the same probability distribution on the input image.

KL-Divergence is one prevailing method to measure dissimilarity among different probability distributions, and is expressed as

$$D_{KL}(P^t \parallel P^{t+1}) = \sum_{x \in X^t} P^t(x) log(\frac{P^t(x)}{P^{t+1}(x)})$$
$$= \sum_{x \in X^t} (P^t(x) log(P^t(x)) - P^t(x) log(P^{t+1}(x)))$$

where P^t is denoted as the target distribution and P^{t+1} denotes predicted distribution. We encourage latter expert to produce a probability distribution P^{t+1} different from previous one P^t.

Due to the limited training data, each expert tends to produce a very confident prediction that produces a vector close to one-hot. Such a result does not reflect the model's description of the inherent structure of the data. Therefore, we remove the maximum value and normalize it to a new distribution that better reflects the model's description of the data (such as the relationship between data and categories). Therefore, maximizing the KL-divergence of two such distributions is equivalent to encouraging the two models to have different descriptions of the data. Specifically, we change the distribution by applying a binary mask.

$$M_i^t = \begin{cases} 0, i = y^c \\ 1, otherwise \end{cases}$$

where i indicates the index of element in M, M is a mask vector, with each element corresponding to a probability in P^t for the expert E_t. It can also been treated as a gated operation to choose distribution for optimization.

Consequently, the KL-Divergence based constraints becomes

$$D_{KL}^t = \langle M, D_{KL}(P^t \parallel P^{t+1}) \rangle$$

where P^t denotes the probability distribution produced by expert E_t over all classes.

$$L_{KL}^t = \exp(-D_{KL}^t)$$

3.3. Mixture of Experts

The final optimization objective can be expressed as follows,

$$L = \sum_{t=1}^{T} L_{cls}^t + \sum_{t=2}^{T} L_{KL}^t + L_{gate}$$

The first term in this objective function indicates each expert is trained on a full-size dataset constructed by transforming the data with attention knowledge from former expert. The second term is a KL-Divergence based penalizing term that encourages experts to produce diversified probability distribution. The L_{gate} is the loss function for learning the gating network, which is expressed as:

$$L_{gate} = -\frac{1}{N} \sum_{i=1}^{N} y_i \log(\sum_{t=1}^{T} g_t * E^t(x_i)),$$

where

$$E^t(x_i) = f(x_i^t, \theta_c^t)$$

and g_t is a set of probability values predicted by the gating network. During test time, the model makes predictions by weighted prediction probability from all experts.

$$\hat{y}_i = \sum_{t=1}^{T} g_t * \hat{y}_i^t,$$

where $\hat{y}_i^t$ is the prediction made by expert E^t.

We illustrate expert design in Figure 2, and the attention module of Figure 2 is shown in Figure 3, in which white circle denotes cropping and resizing input of previous expert before generate new input for later experts. In the training process, we forward the data in the sequential way while back-propagate gradient synchronously and independently among experts. The gradient does not back-propagate from later experts to previous one.

4. Experiments

In this part we will describe the dataset used in this paper, the implementation details and experiments results. We conduct experiments on four challenging fine-grained datasets, which are Caltech-UCSD Birds (CUB-200-2011) [34], Stanford Cars [21], Flowers-102 [25] and NABirds [33] .

CUB-200-2011 dataset contains 200 birds categories with roughly 30 training images per category. The dataset also contains 5994 instances as training set and other 5794 as testing data.

Stanford Car dataset contains 196 car categories for fine-grained task. There are 8144 examples in training set, and for testing set the data number is 8041. Car images from the dataset are taken from various angles, and the categories are assigned based on production year and car model.

Flowers-102 dataset contains 1-2 flowers type with of 1,020 training, 1,020 validation and 6,149 test images.

NABirds dataset contains 23,929 training and 24,633 test images with 555 categories. There are more than 100 photographs available for each species, including separate annotations for males, females and juveniles.

4.1. Implementation Details

We first describe the basic settings of MGE-CNN. The input size to our networks is 448×448. We do not use bounding box or part annotations except for category labels. We compare our experiments results with other weakly supervised approaches (with only class labels).

In the training phase, we augment inputs by resizing images to 512×512 then randomly cropping to 448×448 with random horizontal flipping. We use ResNet-50 as our baseline and implement all of our experiments using PyTorch [26]. The output of each CNN is global average pooled from the last convolutional layer to generate a 2048-dim features vector. As for local features, we use a 1×1 filter with filter number ten times of class numbers being put into global max pooling.

After determining attention map and cropping image, we resize them to 448×448 and then fed into the ConvNets. The parameters within these ConvNets branches are not shared. For threshold related estimating bounding box, we following weakly supervised localization works and apply a scalar with value 0.2. Our model is not sensitive to the threshold, because the amplitude difference between interesting region and other areas is usually large that even changing threshold within a certain ragnge does not make too much difference. The learning rate is 0.001 for pretrained layers, and a $10\times$ multiplier is used for randomly initialized layers. The learning rate is decayed every 30 epoches with decay rate 0.1. SGD optimizer is used with momentum 0.9. We train our networks for 100 epochs with batch size 10 and measure the top-1 classification accuracy from the last epoch.

To better optimize all experts in a mutually reinforced way, we take the following training strategy.

- We initialize convolutional and fully connected layers in Figure 2 using pretrained ResNet-50 [15] weights from ImageNet [8].
- We use gradient based class activation map which computes tensor gradient as the layer weight and weighted summing across feature channels to estimate attention map. Given attentions we infer coarse bounding box and apply cropping and zooming operation to generate new input to next expert. All input to experts have same image size.
- We optimize our model in an end-to-end way. The training images are first fed to the gating network and the first expert to perform the forward propagation step, after that the first expert starts the Grad-CAM step to generate an attention map to automatically generate inputs for the next expert to perform the forward propagation, and so on, until all experts complete the forward propagation and generate predictions. Finally, all predictions are weighted by the predicted gates and fed to loss function to perform gradient backprogation and weight updating for all the networks.

4.2. Experiments Results

Since we do not use extra annotations, we compare results with methods without using human-defined bounding box/part annotations. Table 1 illustrate the results on CUB-200-2011 dataset. The baseline based on ResNet-50 is trained with simple augmentation (random flipping and random cropping) and achieves 85.4%. Our method further outperforms the baseline by 3.1%, achieving the best overall performance against other methods. Compared with DFL-CNN [37] which enhances mid-level representation learning within the CNN framework by learning a bank of convolutional filters to capture class-specific discriminative patches, we get a better result with a relative accuracy improvement of 1.1%. Our method outperform MAMC [32] which uses metrics to learn multiple attention region features by 2.0%. Although our baseline is already strong, the improvement with a large margin indicates that a better representation can still be learned even with a deeper network. A further improvement of another 1% can be seen when we use ResNet-101 as backbone (Table 4).

The classification accuracy on Stanford Cars is also presented in Table 1. We use the same baseline as CUB-200-2011. While our method is only slightly better (0.1%) than DFL-CNN(VGG-16), using same ResNet-50 as baseline, our method still achieve competitive results of 93.9% which is 0.8% better than DFL-CNN(ResNet-50).

Experiments results on Flower-102 and NABirds are shown in Table 2 separately and considerable improvement can be seen when comparing with baseline method.

Figure 4 illustrates examples from CUB-200-2011 and Stanford Cars. After training, we observe that for object with small scale, the entire object will response, which means that the first expert (first two columns) make prediction mainly based on global information. This also provides localization information, because after we estimate significant regions using technique from weakly supervised object localization, we can localization the whole object more precisely, as is shown in the third columns. The input to the second expert is cropped based on attention map from previous input before zooming into the size of first input, so the second expert learns from the object level input, and

Method	Backbone	Accracy(%)	
		CUB	Car
VGG-19	VGG-19	77.8	84.9
ResNet-50	ResNet-50	85.4	91.7
ResNet-101	ResNet-101	86.8	91.9
STN [18]	Inception	84.1	-
RA-CNN [10]	VGG-19	85.3	92.5
MA-CNN [45]	VGG-19	86.5	91.5
B-CNN [24]	VGG16	84.1	91.3
Compact B-CNN [11]	VGG-16	84.0	-
Low-rank B-CNN [20]	VGG-16	84.2	90.9
Kernel-Activation [2]	VGG-16	85.3	91.7
Kernel-Pooling [7]	VGG-16	86.2	92.4
MG-CNN [45]	VGG19	82.6	-
RAM [23]	ResNet-50	86.0	-
MAMC [32]	Resnet-101	86.5	93.0
DFL-CNN [37]	Resnet-50	87.4	93.1
DFL-CNN [37]	VGG-16	87.4	93.8
NTS-Net [42]	Resnet-50	87.5	-
MGE-CNN	Resnet-50	88.5	**93.9**
MGE-CNN	Resnet-101	**89.4**	93.6

Table 1. Comparison of different methods on CUB-200-2011 (CUB) and Stanford-Cars (Car) with out extra annotations.

Method	Backbone	Accracy(%)	
		Flower	NABirds
ResNet-50	ResNet-50	92.4	84.3
ResNet-101	ResnNet-101	92.3	85.3
NAC-CNN [31]	VGG-19	95.3	-
MGE-CNN	Resnet-50	**95.9**	88.0
MGE-CNN	Resnet-101	95.8	**88.6**

Table 2. Comparison of different methods on Flowers-102 (Flower) and NABirds without extra annotations.

the response areas in corresponding attention map (fourth column) becomes more specific. We can see from Table 4, compare to first expert using the large region in the image, the second expert achieve same performance with only cropped region, and by combing first and second expert, we get the largest increase 1.4%, which shows that these two experts have a bigger difference. While the situation is not so obvious for the third expert, the significant region in attention map still become more specific, and final results by combing all three expert (88.5%) see 0.3% increase compared to the situation with two experts. There are not so many scale changes for Stanford Cars, each expert learns good representation from given car images (last two columns from Figure 4). As a results, expert are not diverse enough to produce the stronger predictions when combining.

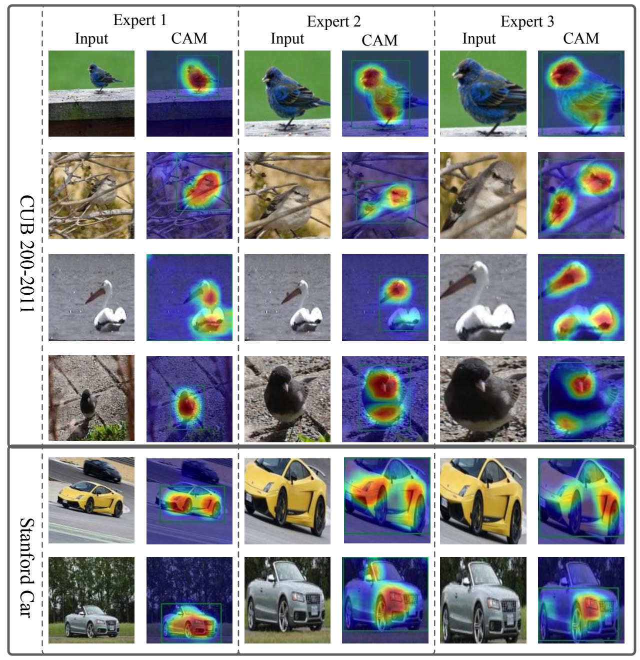

Figure 4. Visualization of the selected results from CUB-200-2011 and Stanford Cars using proposed MGE-CNN. CAM is the class specific attention map. We remap each attention map back to match origin image. For each dataset, the first, third and fifth columns shows the input images to three experts, and the second, fourth and the last columns correspond to attention maps.

Method	Accuracy (%)
Expert 1	86.8
Expert 2	87.3
Expert (1+2)	87.9
Expert (1+2)+KL	88.2

Table 3. Compared the effectiveness of KL-divergence constraints on CUB-200-2011. KL denotes expert with KL-divergence constraint.

4.3. Ablation study

To analyze the contribution of different component in the proposed framework, we conduct various experiments on CUB-200-2011 and report results.

Impact of KL-divergence constraint. We investigate the effect of the KL constraints through experiments with two experts, and one KL constraints can be applied on two distributions. The prediction distribution generated by the former expert as target distribution and the second one as predicted distribution. The performance improvement between two experts in Table 3 verifies the validity of our modified KL constraints.

Impact of different threshold. We choose 0.2 as threshold which is widely used in many methods that use attention maps for weakly supervised localization. We also con-

Expert	Method	Accracy(%)	
		ResNet-50	ResNet-100
1st	GAP	85.4	86.8
	GMP	83.8	82.3
	Concat	86.8	87.5
2nd	GAP	86.1	87.4
	GMP	84.1	84.7
	Concat	87.3	88.3
3rd	GAP	85.2	86.8
	GMP	82.2	83.9
	Concat	86.1	87.4
2 experts		88.2	89.2
3 experts		**88.5**	**89.4**

Table 4. Compared the effectiveness of large and small part information on CUB-200-2011.

Threshold	0.2	0.3	0.4	0.5
Accuracy(%)	88.19	88.44	88.32	88.14

Table 5. Experiments results using different threshold on CUB-200-2011. We only illustrate results combing two experts.

ducted experiments using different thresholds [0.2-0.5], the results in Table 5 shows minor differences.

Impact of large and small part information. As shown in Fig.5, by applying different global pooling methods (GAP, GMP) on two separate convolutional blocks, they will learn different ways of activation responses to the same image. Due to the averaging operation, a unit of the GAP output is highly depended on how many spatial locations in the feature map are activated by the corresponding filter, therefore, the GAP convolution block usually learns filters that sensitive to a large regions of the image. In contrast, the GMP convolution block only cares about if a certain spatial location is highly activated by the filter, the patterns it finds are mostly small image region. With this design, the resulting feature can encode both large and small part information. More results can be seen in Table 4. By combining large and small part together, we get stronger features. Based on these features, the accuracy increases by 1.2% from 85.4% to 86.8%. Although, margins are smaller for expert 2 and expert 3, their performances are still 0.3%, 0.9% better than only using GAP.

Impact of multiple experts. As is shown in table 4, with only one expert we achieve 86.8%. The largest performance boost can be seen when we include a second expert, the performance increase to 88.2%, which is already better than all opponents. After adding the third expert, we obtain another 0.3% growth. Note that the second expert gets better performance than other experts. One reason is that for some images, object to be recognized for the first expert is small making it hard to get more detailed information. This

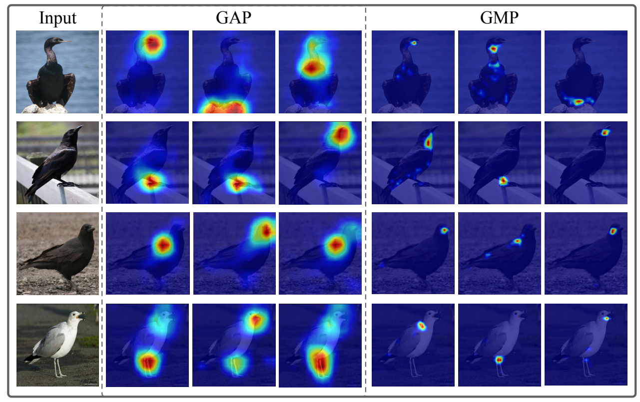

Figure 5. Visualization of the top-3 highest activation maps on selective exemplars from CUB200-2011

problem is alleviated by the second expert (Figure 5), since more details are obtained after objects are localized and enlarged. However, for the third expert, some parts of object are cut off as is shown in Figure 4, resulting in a slight drop in performance.

5. Conclusion

In this paper, we propose a unified framework for fine-grained image classification. The proposed method is based on a mixture of experts, but we divide fine-grained problem into subspaces by learning latter expert with prior information from previous expert. In this way we learn a set of gradually enhance experts on full-size data for each expert. We learn diverse experts by combining progressively enhanced strategy and KL-divergency based constraints. Finally, these experts make diverse predictions, and final predictions are made by weighted combing predictions from all experts using weights generated by a gating network. Our method can also closely integrate the large and small part features, which provides rich information when recognizing an object. The proposed method does not need bounding box or part annotations during training or test time and can be trained in end-to-end way. Experiments are conducted on several fine-grained tasks (CUB-200-2011, Stanford Cars, Flowers-102, NABirds) and achieve better performance than baseline methods.

6. Ackonwledge

This work was supported by the Australian Research Council Project FL-170100117 and DP-180103424.

References

[1] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic

neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

[2] Sijia Cai, Wangmeng Zuo, and Lei Zhang. Higher-order integration of hierarchical convolutional activations for fine-grained visual categorization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 511–520, 2017.

[3] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *IEEE Winter Conference on Applications of Computer Vision*, pages 839–847, 2018.

[4] Xinyuan Chen, Chang Xu, Xiaokang Yang, Li Song, and Dacheng Tao. Gated-gan: Adversarial gated networks for multi-collection style transfer. *IEEE Transactions on Image Processing*, 28(2):546–560, 2018.

[5] Xinyuan Chen, Chang Xu, Xiaokang Yang, and Dacheng Tao. Attention-gan for object transfiguration in wild images. In *European Conference on Computer Vision (ECCV)*, pages 164–180, 2018.

[6] Yue Chen, Yalong Bai, Wei Zhang, and Tao Mei. Destruction and construction learning for fine-grained image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5157–5166, 2019.

[7] Yin Cui, Feng Zhou, Jiang Wang, Xiao Liu, Yuanqing Lin, and Serge Belongie. Kernel pooling for convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[9] David Eigen, Marc'Aurelio Ranzato, and Ilya Sutskever. Learning factored representations in a deep mixture of experts. *arXiv preprint arXiv:1312.4314*, 2013.

[10] Jianlong Fu, Heliang Zheng, and Tao Mei. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In *IEEE conference on computer vision and pattern recognition*, pages 4438–4446, 2017.

[11] Yang Gao, Oscar Beijbom, Ning Zhang, and Trevor Darrell. Compact bilinear pooling. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 317–326, 2016.

[12] ZongYuan Ge, Alex Bewley, Christopher McCool, Peter Corke, Ben Upcroft, and Conrad Sanderson. Fine-grained classification via mixture of deep convolutional neural networks. In *IEEE Winter Conference on Applications of Computer Vision*, pages 1–6, 2016.

[13] ZongYuan Ge, Christopher McCool, Conrad Sanderson, and Peter Corke. Subset feature learning for fine-grained category classification. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 46–52, 2015.

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *EEE international conference on computer vision*, pages 2961–2969, 2017.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[16] Shaoli Huang, Zhe Xu, Dacheng Tao, and Ya Zhang. Part-stacked cnn for fine-grained visual categorization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1173–1182, 2016.

[17] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, Geoffrey E Hinton, et al. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

[18] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.

[19] Michael I Jordan and Robert A Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 6(2):181–214, 1994.

[20] Shu Kong and Charless Fowlkes. Low-rank bilinear pooling for fine-grained classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7025–7034, 2017.

[21] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *IEEE International Conference on Computer Vision Workshops*, pages 554–561, 2013.

[22] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *IEEE International Conference on Computer Vision*, pages 3524–3533, 2017.

[23] Zhichao Li, Yi Yang, Xiao Liu, Feng Zhou, Shilei Wen, and Wei Xu. Dynamic computational time for visual attention. *arXiv preprint arXiv:1703.10332*, 2017.

[24] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnn models for fine-grained visual recognition. In *IEEE International Conference on Computer Vision*, pages 1449–1457, 2015.

[25] M-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, 2008.

[26] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[27] Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. Learning deep representations of fine-grained visual descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 49–58, 2016.

[28] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[29] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *IEEE International Conference on Computer Vision*, pages 618–626, 2017.

[30] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[31] Marcel Simon and Erik Rodner. Neural activation constellations: Unsupervised part model discovery with convolutional networks. In *IEEE International Conference on Computer Vision*, pages 1143–1151, 2015.

[32] Ming Sun, Yuchen Yuan, Feng Zhou, and Errui Ding. Multi-attention multi-class constraint for fine-grained image recognition. *European Conference on Computer Vision*, 2018.

[33] Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 595–604, 2015.

[34] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[35] Xinchao Wang, Engin Türetken, François Fleuret, and Pascal Fua. Tracking interacting objects optimally using integer programming. In *European Conference on Computer Vision*, pages 17–32, 2014.

[36] Xinchao Wang, Engin Türetken, Francois Fleuret, and Pascal Fua. Tracking interacting objects using intertwined flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11):2312–2326, 2015.

[37] Yaming Wang, Vlad I Morariu, and Larry S Davis. Learning a discriminative filter bank within a cnn for fine-grained recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4148–4157, 2018.

[38] Tianjun Xiao, Yichong Xu, Kuiyuan Yang, Jiaxing Zhang, Yuxin Peng, and Zheng Zhang. The application of two-level attention models in deep convolutional neural network for fine-grained image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 842–850, 2015.

[39] Zhe Xu, Shaoli Huang, Ya Zhang, and Dacheng Tao. Augmenting strong supervision using web data for fine-grained categorization. In *IEEE International Conference on Computer Vision*, pages 2524–2532, 2015.

[40] Zhe Xu, Shaoli Huang, Ya Zhang, and Dacheng Tao. Webly-supervised fine-grained visual categorization via deep domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(5):1100–1113, 2016.

[41] Zhe Xu, Dacheng Tao, Shaoli Huang, and Ya Zhang. Friend or foe: Fine-grained categorization with weak supervision. *IEEE Transactions on Image Processing*, 26(1):135–146, 2016.

[42] Ze Yang, Tiange Luo, Dong Wang, Zhiqiang Hu, Jun Gao, and Liwei Wang. Learning to navigate for fine-grained classification. In *European Conference on Computer Vision*, pages 420–435, 2018.

[43] Xiaoqing Yin, Xinchao Wang, Jun Yu, Maojun Zhang, Pascal Fua, and Dacheng Tao. Fisheyerecnet: A multi-context collaborative deep network for fisheye image rectification. In *European Conference on Computer Vision*, pages 469–484, 2018.

[44] Xiaopeng Zhang, Hongkai Xiong, Wengang Zhou, Weiyao Lin, and Qi Tian. Picking deep filter responses for fine-grained image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1134–1142, 2016.

[45] Heliang Zheng, Jianlong Fu, Tao Mei, and Jiebo Luo. Learning multi-attention convolutional neural network for fine-grained image recognition. In *IEEE international conference on computer vision*, pages 5209–5217, 2017.

[46] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2921–2929, 2016.

[47] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision*, pages 2223–2232, 2017.