IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 12, 2019, accepted July 30, 2019, date of publication August 5, 2019, date of current version September 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2933169

Probability Fusion Decision Framework of Multiple Deep Neural Networks for Fine-Grained Visual Classification

YANG-YANG ZHENG[1], JIAN-LEI KONG[1,2], XUE-BO JIN[1,2], XIAO-YI WANG[1,2], TING-LI SU[1], AND JIAN-LI WANG[1]

[1]School of Computer and Information Engineering, Beijing Technology and Business University, Beijing 100048, China
[2]Beijing Key Laboratory of Big Data Technology for Food Safety, Beijing Technology and Business University, Beijing 100048, China

Corresponding authors: Jian-Lei Kong (kongjianlei@btbu.edu.cn) and Xue-Bo Jin (jinxuebo@th.btbu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFC1600605, in part by the Beijing Municipal Education Commission under Grant KM201910011010, and in part by the National Natural Science Foundation of China under Grant 61673002.

ABSTRACT Fine-grained visual classification tasks often suffer from that the subordinate categories within a basic-level category have low inter-class discrepancy and high intra-class variances, which is still challenging research for traditional deep neural networks (DNNs). However, different models extract local parts' features in isolation and neglect the inherent correlations and distribution in high-dimensional space, which limit the single model to achieve better accuracy. In this paper, we propose a novel probability fusion decision framework (named as PFDM-Net) for fine-grained visual classification. More specifically, it first employs data-augmented tricks to enlarge the dataset and pretrain the basic VGG19 and ResNet networks on high-quality images datasets to learn common and domain knowledge simultaneously while fine-tuning with professional skill. Next, refined multiple DNNs with transfer learning are applied to design a multi-stream feature extractor, which utilizes the mixture-granularity information to exploit high-dimensionality features for distinguishing interclass discrepancy and tolerating intra-class variances. Finally, a probability fusion module equipped with gating network and probability fusion layer is developed to fuse different components model with Gaussian distribution as a unified probability representation for the ultimate fine-grained recognition. The input of this module is the various features of multi-models and the output is the fused classification probability. The end-to-end implementation of our framework contain an inner loop about the EM algorithm within an outer loop with the gradient back-propagation optimization of the whole network. Experimental results demonstrate the outperforming performance of PFDM-Net with higher classification accuracy on different fine-grained datasets compared with the state-of-the-arts methods. More discussions are provided to indicate the potential applications in combination with other work.

INDEX TERMS Deep neural network, weakly-supervised learning, multi-steam feature extractor, probability fusion module.

I. INTRODUCTION

Deep neural networks (DNNs) are the most important research branches in machine learning. Thanks to the breakthroughs in the design and training of DNNs with complex structures consisting of multiple processing layers or non-linear transformations, unprecedented improvements have penetrated into many aspects of artificial intelligence, especially the performances of visual classification on large-scale datasets (such as ImageNet [1], MSCOCO [2]). Such fast-pacing progresses in research have also drawn attention of the related research corporations and industries like Google, Facebook to build software and hardware to recognize everything snapshotted by the consumers in real life [3].

Although great successes have been achieved for basic-level classification in the last few years, fine-grained visual classification (FGVC) aiming to identify the subordinate-level categories under a basic-level category of daily objects such as animal species or aircraft types is still a quite challenging task for existing DNNs-based methods [4]–[9],

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

such as AlexNet, VGGNet, DenseNet. For example, the subordinate-level recognition between a ''Yellow breasted chat'' and a ''Yellow headed Blackbird'' with similar appearance is significantly more difficult than the basic-level recognition divided as a coarse basic-level category ''bird''. It is hard to obtain accurate classification results only by the state-of-the-art coarse-grained DNNs because of three main reasons: (1) Minor interclass variances. The global geometry and appearances of samples belong to different categories may be very similar apart from some subtle differences of several key parts, which is a vital identification puzzle for DNNs; (2) Major intra-class variances. Samples that belong to the same category usually present significantly different poses, viewpoints and locations in different images; (3) Limited magnitudes of training data. The sample number of each fine-gained category cannot meet the basic requirements of DNNs training for deep features representation and excellent recognition.

To distinguish fine-grained categories with very similar outline, it requires specialized knowledge focusing on feature representation of discriminative object parts to expand the application of existing DNNs on FGVC. According to whether the method requires additional part location annotation, current stage-of-the-arts can be divided into two groups: strongly-supervised learning (SSL) [10] and weakly-supervised learning (WSL) [11]. In the training process, the former requires additional location information apart from image-level category labels, such as bounding-boxes or key-points of discriminative parts. Location annotation heavily rely on more expensive manual labeling and time-cost, which makes it hard to be prevalently applied in practice. Consequently, researchers pay more attention on WSL frameworks, which only employ image-level annotation to achieve FGVC tasks. For instance, the attention mechanism can be implemented to capture local feature in a translationally invariant manner, which is particularly suitable for classifying fine-gained categories without manual location annotations. Despite the great progresses, these WSL methods universally suffer lower performance than the best SSL models, especially when small objects appear in clutter background. More importantly, most WSL works tend to extract local part features in isolation, while neglect their inherent correlations and high-dimensional distribution. Given the learned location features of objects' parts, single WSL model is likely to focus on the constant architecture of parts distribution and lack the capability to distinguish interclass variances between similar fine-grained classes. Moreover, diverse WSL models are interested in multiple object's parts with different preferences, which aggravate intra-class variances of the same class. As a consequence, it is very likely to bring about the wrong category when these parts are occluded due to pose or viewpoint variances, which lead to the diverse recognition performance of different WSL models.

From extensive experimental studies, it is observed that WSL models with different location features have potential information complementary characteristics for each other. Therefore, the information fusion mechanism is introduced to integrate multiple WSL model advantages for discovering discriminative parts in fine-grained visual classification. To verify the application value, an effective probability fusion framework in decision-level viewpoint named as PFDM-Net was designed, which aims at utilizing the mixture-granularity information of multiple DNNs by only using image-level labels. In detail, the PFDM-Net first generates input images with various data augmented preprocessing and batch initialized tricks. Next, the framework is trained by the multi-stream DNNs architecture to exploit the high-dimensional feature maps representing discriminative and non-discriminative parts as well for interclass variances. As for each stream, some advanced WSL models were adopted separately achieving excellent performance on public FGVC datasets. Then, a novel probability fusion module is developed to fuse different features as a unified probability representation for the ultimate fine-grained recognition. The end-to-end implementation of fusion module has an inner loop about the expectation maximization algorithm (EM algorithm) used in fusion layer and an outer loop about backward gradient propagation of the whole network in the training process. This optimization design offers a high capacity of learning complementary yet correlated information for intra-class variances among multi-grained feature maps of different models, which make the proposed PFDM-Net more suitable for identifying slight discrepancy when distinguishing the fine-grained categories in FGVC tasks. The experimental results on CUB-200-2011 [12], Stanford Cars [13] and Stanford Dogs [14] show the robustness and superiority of PFDM-Net, which achieves the top performance outperforming state-of-the-art methods.

The rest of the paper is organized as the follows: Section II outlines an overview of the related works. The proposed framework is explained in Section III. Experimental results and evaluation performance of our technique compared with other state-of-the-art methods are discussed in Section IV. Finally, Section V make some further discussion and Section VI concludes the whole work with future research prospects.

II. RELATED WORK

At present, there are four main types of fine-grained image classification methods: strongly-supervised learning methods, weakly-supervised learning methods, attention mechanism methods, higher order representation learning strategies and methods. We review recent works in the literature in this section.

A. STRONGLY-SUPERVISED LEARNING METHODS

To focus on the local features, many strongly-supervised learning methods rely on the manual annotations of parts location or attribute. Reference [15] finds that the migration of Deep Convolutional Activation Feature (DeCAF) to fine-grained classifications yields better classification performance. Reference [16] is another early work based

on the point of view of deep-learning migration, which transfers the object detection model(R-CNN) as novel Part R-CNN [17] to extracting important local information for fine-grained classification. The Part R-CNN follows the routine of generating semantic part prediction (head, body, etc.) of the object (bird) with geometric constraints and classifying images by use of pose-normalized representation. It shows that better localization of parts does lead to further improvement of classification, yet greatly affect the arithmetic speed. Reference [18] proposes a feedback-control framework Deep-LAC with a valve linkage function connecting the localization and classification modules to back-propagate alignment and classification errors to localization. Reference [19] introduces the collaborative segmentation into fine-grained image classification, and propose a novel local region detection algorithm. The algorithm named SPDA-CNN only relies on the label box to segment object parts and combines them with alignment operations for fine-grained classification. Nevertheless, it is also dependent on extra location annotation with expensive manual labeling, which makes it hard to be prevalently applied in practice.

B. WEAKLY-SUPERVISED LEARNING METHODS

In order to reduce the human labeling cost, many weakly-supervised learning methods that only require image-level annotation has gradually been proposed in recent years. Reference [20] proposes the Two-Level Attention models (TLAN) to extract object-level and part-level features in bottom-to-up way at the same time. Reference [21] designs a novel bilinear network model (Bilinear CNN) is to combine two stream features at each location using the outer product, which considers their pairwise interactions in the end-to-end training process. Similarly, reference [22] proposes the object-part attention model (OPAM) for weakly supervised FGVC without neither object or part annotations, which avoids the heavy labor consumption of labeling This model integrates two level attentions: object-level attention localizes objects of images and part-level attention selects discriminative parts. Both are jointly employed to learn multi-view and multi-scale features to enhance their mutual promotion. Spatial Transformer Network (ST-CNN) [23] also chooses a weakly-supervised way. The model can also locate several object parts simultaneously to achieve more accurate classification performance by first learning a proper geometric transformation and align the image before classifying.

C. ATTENTION MECHANISM METHODS

There are also many other models based on attention mechanism in fine-grained classification. Reference [24] proposes Recurrent Attention CNN (RA-CNN) and progressively learns coarse to fine region attention areas by training an attention sub-network. To generate multiple attention locations in a mutually reinforced way, reference [25] proposes Multi-Attention CNN (MA-CNN) to locate multiple part attentions from feature channels at same time. However, the number of object's parts is still limited, which might constrain further accuracy improvement. Reference [26] proposes Weakly Supervised Bilinear Attention Network (WS-BAN) to solve the above issue. It jointly generates a set of attention maps to indicate the locations of object's parts and extracts sequential part features by bilinear attention pooling. The generating process of attention maps is flexibly adjusted according to the parts quantity. However, training by softmax with cross entropy loss usually leads the model to pay attention to the most discriminative location, whose output neglect that inter-layer part feature interaction and fine-grained feature learning are mutually correlated. To reinforce locations of the whole object and parts, reference [27] propose a hierarchical bilinear pooling framework (HBP) to integrate multiple cross-layer bilinear features to enhance their representation capability, which results in superior performance compared with other bilinear pooling-based approaches.

D. HIGHER ORDER REPRESENTATION LEARNING STRATEGIES AND METHODS

Besides attention models, higher order representation learning strategies and methods are also explored. On basis of object-level and part-level features, reference [28] proposes a discriminative filter bank of convolutional filters (DFL) to capture mid-level class-specific patches. Reference [29] develops the deep metric learning to construct a novel multi-attention multi-class constraint neural network (MAMC), which regulates multiple object parts and extracts high level abstractions from different input images-pairs. Reference [30] applies clustering mechanism to utilize discriminative parts from feature maps of CNN's mid-layers, and a Gaussian mixture layer (GMNet) is proposed to model distribution of part features, which are treated as data points to generate output features based on combination of cluster center. Hence, reference [31] have shown increasing interests in generalizing average pooling and bilinear pooling to covariance pooling layers, which is identified as Matrix Power Normalized Covariance (MPN-COV) method. Reference [32] further expand second-order pooling to higher-order global covariance pooling, and proposes an iterative matrix square root normalization method designed with loop-embedded directed graph structure for fast end-to-end training of global covariance pooling networks (iSQRT-COV).

Despite the great progresses, these WSL methods universally suffer lower performance. However, diverse WSL models are interested in multiple object parts with different preferences, which could make up for each other to predict better results. Therefore, several recent works have developed a great progress in investigation of information fusion strategies with different models. Reference [33] and [34] attempt to fuse different models as a two-branch HybridNet architecture. Reference [35] proposes a novel self-supervision model termed Navigator-Teacher-Scrutinizer Network (NTSN) to effectively localize informative regions. However, multi-scale fine-grained image

classification remains a challenging task because multiple WLS models cannot be guided with an effective fusion strategy, which will reduce the overall classification accuracy especially when low-quality image distortion occurs. Inspired by the mixture-of-experts layer [36], an effective probability fusion decision framework named PFDM-Net was designed to integrate multiple WSL models for discovering discriminative parts in fine-grained visual classification. Different from the previous studies, this work presents a probability fusion module by combing different models for each example, where model capacity is critical for absorbing the vast quantities of knowledge available in the training corpora. The experimental results show that the proposed PFDM-Net performs better in terms of classification accuracy. This work can be also combined with other newly updated models for yielding better results in the future.

III. METHODOLOGY

In this section, we first introduce the motivation of the whole framework. Then, we focus on the proposed model named PFDM-Net, and the details of model training are chaired in the end.

A. MOTIVATION

It is well known that more abstract features are performed to describe the structure, shape and sketch of objects in a more complete manner as the depth of the CNN layer increases. Therefore, DNNs with flexible CNN layers and tricks become the foundational instruments for feature extraction of WLS models. However, images of fine-grained visual task usually contain more complex texture, rich color and complicated spatial relationship than ones of basic-level classification in which one or more objects are dominated. At the same time, detailed information becomes gradually lost in deeper CNN layers, and consequently most regions in a scene image possess no longer discriminative property.

To further illustrate the above analysis, we take three models including ResNet101, WS-BAN and iSQRT-COV currently performing well on fine-grained image classification as an example. As aforementioned, the ResNet model is employed to mitigate network training with deeper structure owing to its well-designed architecture, impressive performance and low model complexity. This model explicitly reformulates the CNN layers as residual functions with reference to the basic rule that partial data of the input goes directly to the output without passing through the neural network, which is proven to preserve some original information and prevent effectively the gradient dispersion problem in back propagation. In addition, there are off-the-shelf pre-trained ResNet models including various auxiliary branches, which are convenient for implementation of coarse-grained classification. Hence, the ResNet-101 model was taken, which the major novel component of ResNet, as the traditional DNNs without special design focusing on subtle differences of local information compared with other WSL models. Moreover, WS-BAN jointly learns the location of a large num ber of discriminative object's parts and their feature representation to improve the accuracy of FGVC. This architecture generates feature maps and attention maps by using the lightweight VGG16 as the basic feature extractor, which has a good effect on small target features with adequate semantic information. Reversely, iSQRT-COV modifies the first-order convolutional pooling with ResNet architectures as the second-order matrix power normalized covariance pooling. This has shown that, given an order through compact explicit low-dimensional features, matrix power is consistent with a robust feature extractor characterizing higher order feature interactions, which achieves impressive improvements over traditional WSL models.

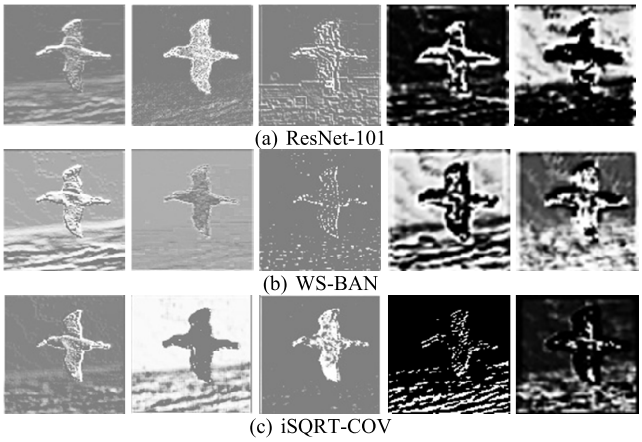

FIGURE 1. Feature maps from 1st, 3rd, 5th, 7th and 9th (from left to right) convolutional layers of ResNet-101(a), WS-BAN(b) and iSQRT-COV(c) models are markedly different, which illustrates that various models have different learning preferences to the same object.

In this work, we use these above models trained on the fine-grained CUB-200-2011 image dataset to demonstrate that the feature maps obtained from coarse-grained extractor, first-order fine-grained extractor and second-order fine-grained extractor are distinctly different. The visualized feature results of the same image extracted from different layers of various models are shown in Figure 1.

Although feature maps from different models exhibiting multi-view visual characteristics, all above models obtain inherent representation abilities to achieve preeminent performance on the fine-grained recognition task. Experiments are presented on fine-grained datasets of birds. Various feature extraction architectures are initialized by using models trained on the ImageNet. Without additional local information and tricks, the fine-tune ResNet101 does remarkably well and features prior to classifier achieve 86.57% accuracy on the CUB-200-2011 dataset. Fine-tuning bilinear attentions improve the performance of WS-BAN model in certain degree, outperforming a number of existing methods that additionally rely on object or part annotations. In comparison a bilinear model consisting of first-order convolutional pooling, the second-order matrix covariance pooling applied in the training process makes iSQRT-COV model achieve the better outperforming accuracy.

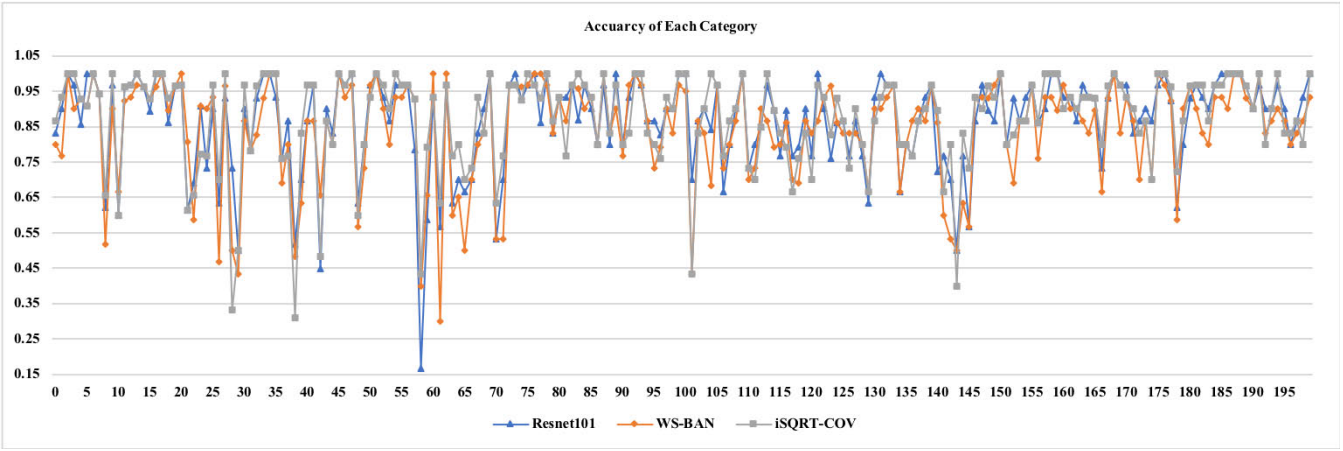

FIGURE 2. Accuracy results of each category offered by ResNet101, WS-BAN and iSQRT-COV models on the CUB-200-2011 dataset.

However, the WSL models do not achieve the highest accuracy at present, which extract location features of objects' parts in isolation, neglecting their inherent correlations and high-dimensional distribution. Therefore, single model is likely to focus on the constant architecture of parts distribution and lack the capability to distinguish interclass variances between similar fine-grained classes. Moreover, various models with discrepant distribution architecture are interested in different categories with inconsistent preference. The accuracy differences of above WSL models for each category are illustrated as following Figure 2. It is shown that different models have different recognition capabilities for each category, and the single model has discrepant accuracy performance for different categories. For example, WS-BAN only obtains 16.7% accuracy for the 59th category, then get the accuracy up to 66.7% for the 67th class and 100% for the 122th class. As a result, the relatively large accuracy difference limits the overall accuracy of the single model to further enhance. Correspondingly, ResNet101 achieves the accuracy to 40% for the 59th category and iSQRT-COV obtains the accuracy up to 45.4%. The accuracy difference of the same category among three models is relatively large due to layer depth of feature extractor, object discriminative attributes and detailed information loss. Since features from different models exhibit different visual characteristics and discernible recognition accuracies, we consider to combine the inherent representation capabilities of several WSL methods to construct a novel fused model for further improving fine-grained image recognition performance. The idea of the fused method is similar to the idea of multi-view learning, which takes advantage of features from different views, since multi-view features can complement each other and acquire better understanding of intra-class differences.

On the other hand, each model is interested in multiple parts of objects with different preferences, which bring about intra-class variances among quantity-changed images of the same category. Moreover, complicated background with discriminative properties including color, illumination,

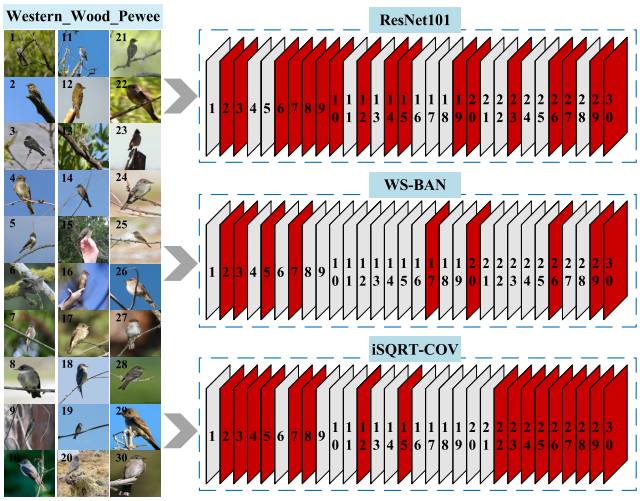

FIGURE 3. Testing classification results of different WLS models on the 101st class.

poses or viewpoint further aggravate the diverse recognition performance of different WSL models for each fine-grained class. An example of the intra-class recognition achieved from different models is experimented to further demonstrate the above discussion. The experimental analysis is depending on the 101st class (biological category is Western Wood Pewee) with a total of 30 test images in the CUB-200-2011 dataset. Then, the variant results predicted by ResNet101, WS-BAN and iSQRT-COV are illustrated in following Figure 3. Pewee) with a total of 30 test images in the CUB-200-2011.

As Figure 3 shown, the red pictures belonging to the 101st class represent the wrong recognition result of corresponding model, which mistakenly divide these images into other bird categories. Yet, the grey images are predicted as the correct category with the guide of image-level annotations. It is obviously known than different models have different recognition capabilities for the same category and focus on the dominant parts of the same species object with

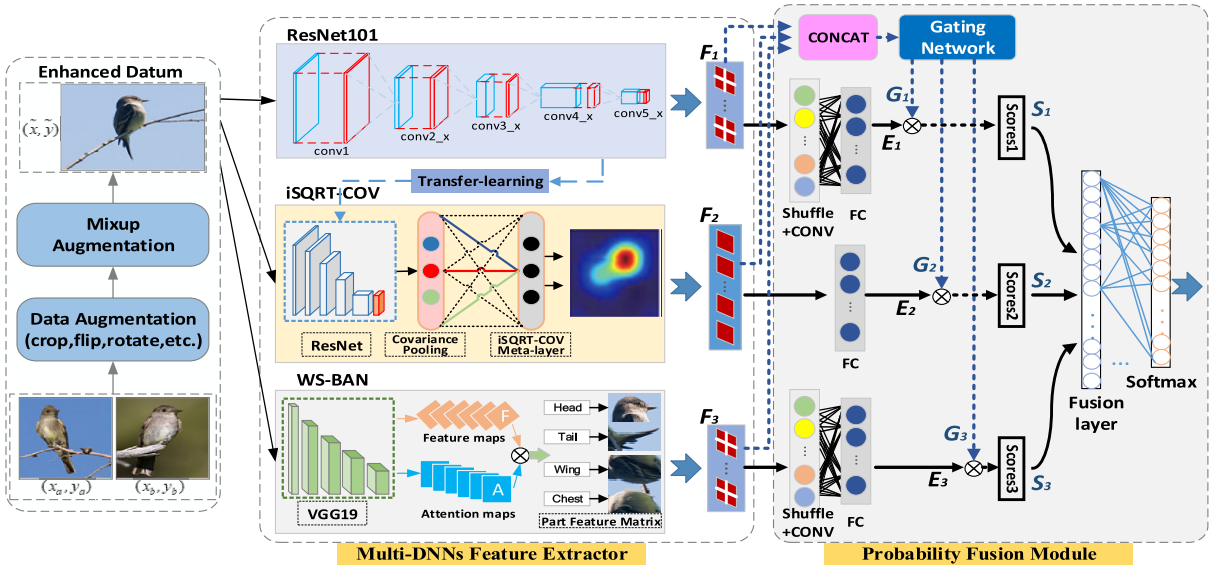

FIGURE 4. Schematic diagram of the fusion model structure.

significant differences. Another interesting analysis is that ResNet101 and iSQRT-COV don't perform as well as the WS-BAN model with a shallower VGG structure of feature extractor for this 101st class, even though both of them have the deeper feature extractor with better representation ability.

However, excepting for a few same cases where everyone makes mistakes, various models make most mistakes at different points, which contain the potential rule that information complementation of multiple models can reduce the predicted error of test image samples. Thus, the addition rule should be assigned to fusing prediction probability scores of multiple models' classifiers, which are relatively independent from each other, to compute the global weighted score for better intra-class representation of various images and samples in the same category. Therefore, the probability fusion strategy applicable to amplifying interclass discrepancies and suppressing intra-class differences is adopted to design the proposed PFDM-Net model on basis of ResNet101, WS-BAN and iSQRT-COV models. The fusion framework takes parallel advantages of multi-models in multi-dimensional feature extraction and multi-view decision complementation, which is experimentally proven to improve the overall accuracy of FGVC task. The following presents how to incorporate the probability fusion rule into our multi-models framework.

B. PFDM-Net ARCHITECTURE

By analyzing the differences in the recognition ability of individual models for different categories and the recognition of multiple models in the same category, we find that there are different feature extraction capabilities of various models for pictures. Therefore, a fusion strategy was designed to fuse the feature results of the multi-stream models through the fully connected layer output, so that the models complement each other and improve the whole accuracy. The fusion model diagram is shown in Figure 4.

1) DATA AUGMENTATION

To avoid over-fitting of the network, some data augmentations are applied to enhance a larger number of dataset's images with high quality before training. Generally, various adjusted forms and distortion are needed to resize the input images before sending them to the network. Hence, the more complicated the tasks are, the more images the DNN models need to nonlinearly estimate massive parameters adopted by most classification works, especially when using images with low resolution. To address this problem, we formulate a series of augmented methods to increase the general training datum, which consists of following six steps one-by-one:

1) Randomly crop a rectangular region whose aspect ratio is randomly sampled in [3/4, 4/3] and area randomly sampled in [8%, 92%], then resize the cropped region into a 448-by-448 square image.
2) Randomly flip each image 180 degrees horizontally and vertically with 0.5 probability in order to increase the diversity of the image.
3) Randomly rotate each image in 90, 180, and 270 clockwise in order to improve distortion adaptability.
4) In the HSV color space of the image, exponentially changing the saturation S and brightness V components of each pixel, keeping the hue H constant, to increasing the illumination variation. The S and V channels are respectively scaled with coefficients uniformly drawn from [0.25, 4].
5) Randomly sample an image and decode it into 32-bit floating point raw pixel values in [0, 255]. And randomly add PCA, salt-pepper and Gaussian noises with a coefficient [37] sampled from a normal distribution N(0, 0.1) to increase the perturbation resistance of each pixel.
6) Finally, the mixup augmentation method proposed in [38] is select to regularize the network models to

favor simple linear behavior in-between training examples for alleviating undesirable behaviors. In the mixup step, each time two examples (x_a, y_a) and (x_b, y_b) are randomly sampled from training data to form a new virtual training example by a weighted linear interpolation:

$$\widetilde{x} = \lambda x_a + (1-\lambda) x_b$$
$$\widetilde{y} = \lambda y_a + (1-\lambda) y_b \quad (1)$$

where x_a and x_b are raw input image vectors, y_a and y_b are one-hot encoding labels, λ is a random number drawn from the Beta(α, α) distribution with the value range [0,1]. And the mixup hyper-parameter α controls the strength of interpolation between vector-label pairs, of which value is recommended as tending to 0. In this paper, we set $\alpha = 0.18$ in the Beta distribution and increase the epoch number asking for a longer training progress to converge better performance. Thus, we achieve additional high-quality examples $(\widetilde{x}, \widetilde{y})$ through the enhanced data augmentation for subsequent model training.

Those above steps can obtain improved generalization and robustness abilities of the network architecture by the augmented datum.

2) MULTI-STEAM FEATURE EXTRACTOR

The feature extractor of the proposed PFDM-Net consists of multiple classification deep neural networks which are trained concurrently on the augmented datum from the first stage. In this section, ResNet101, WS-BAN and iSQRT-COV models will be used to obtain multi-stream feature maps. For each componential model, some minor adjustments were made to the network architecture, such as changing the stride or kernel size of a particular convolution layer. Such a tweak often barely changes the computational complexity but might have a non-negligible effect on the model accuracy.

Firstly, we briefly present the ResNet architecture as a coarse-fined feature extractor to investigate the effects of model tweaks. A basic ResNet network consists of an input stem, four subsequent stages and a final output layer, which is detailed illustrated in [5]. The input stem has a 7×7 convolution with an output of 64 channels and stride 2, followed by a 3×3 max pooling layer also with a stride of 2. Starting from conv2_x stage, the input stem begins with a downsampling block, which is then followed by several residual blocks. There two path candidates in the downsampling block, where the path-A respectively consist of 1×1, 3×3 and 1×1 kernel and the path-B uses a 1×1 convolution to transform the input shape to be the output shape of path-A. A residual block is similar to a downsampling block except for only using convolutions with a stride of 1. In this paper, the highly rated ResNet-101 was selected as the first pipeline of multiple architecture with 3, 4, 23, 3 residual blocks respectively in conv2_x, conv3_x, conv4_x, conv5_x stages. Especially, we revisit some popular ResNet model tweaks. One basic tweak is replacing the 7×7 convolution in the input stem with five conservative 3×3 convolutions, which lower the computational cost and permit the input of augmented datum with larger 448×448 size. Then, we introduce the Inception-v4 module with residual connections module as the similar implementations of [39] and adopt batch normalization module right after each convolution and before activation to improve the single-frame recognition performance. Finally, the eventual average pooling layer was abandoned, the 1000-d full convolution layer and the softmax layer to extract the feature map vector F_1 as shown in Equation (2).

$$F_1 = H_{resnet}(\widetilde{x}, \widetilde{y}, \{W_1, b_1, \delta\}) \quad (2)$$

where the function H_{resnet} can represent multiple convolutional layers of the ResNet architecture with the inputs $(\widetilde{x}, \widetilde{y})$ denoted to the first of these layers. W_1 denotes a square weight matrix asymptotically approximating complicated combination of multiple layers. b_1 can perform the biases of linear projection by the shortcut connections to match the dimensions, channel by channel. And δ denotes the nonlinear activation functions, which was selected as ReLU. We initialize the weights and train all plain/residual nets from scratch with a weight decay of 0.0001 and a momentum of 0.9. The learning rate starts from 0.1 and is divided by 10 when the error plateaus. Then, the feature map vector F_1 was obtained performed on $N \times c_1 \times w_1 \times h_1$ dimensionality, where N represents the mini-batch size, c_1 represents the channel number of feature map, w_1 and h_1 denote the width and height of each map. Thus, the SGD with a mini-batch size of 512 was used, and feature map is converted to the size as $w_1 \times h_1 = 7 \times 7$ and the channel number as $c_1 = 2048$.

Then, the iSQRT-COV is employed to extract fine-grained feature maps of small-scale object's parts. This model is an iterative matrix square root normalization method for fast end-to-end training of global covariance pooling networks, which consists of a basic classification network (AlexNet or ResNet), some covariance pooling layers and an iSQRT-COV Meta-layer. To further improve the performance and efficiency of our proposed architecture, we adopt the transfer-learning strategy to learn professional representation capability of object's parts on the basis of the coarse-grained domain knowledge from abovementioned ResNet model. Therefore, the trained ResNet101 network was transferred as the classification network of iSQRT-COV, which avoids the repeated parameter calculation in much less epochs, further accelerating network training. After the last convolutional layer of ResNet101, some 11 convolution with $c_2 = 256$ channels were added to down sample the outputted feature tensor, which outputs a $14 \times 14 \times 256$ tensor. Then a second-order pooling is performed to estimate the covariance matrix. Subsequently, the model designs a meta-layer with loop-embedded directed graph structure for computing approximate square root of covariance matrix. The meta-layer consists of three nonlinear structured layers, performing pre-normalization, coupled Newton-Schulz iteration and

post-compensation, respectively. The first pre-normalization layer is to guarantee the following iteration convergence, achieved by dividing the covariance matrix by its trace. The second layer is of loop structure, repeating the coupled matrix equations involved in Newton-Schulz iteration a fixed number of times, for computing approximate matrix square root. Then the third post-compensation layer is set to counteract the adverse effect by multiplying trace of the square root of the covariance matrix. In this work, we erases the subsequent ConvNet layers of iSQRT-COV and directly take the outputting symmetric matrix of the meta-layer as an $c_2(c_2+1)/2$ dimensional vector F_2 as shown in Equation(3).

$$F_2 = H_{iSORT}(\widetilde{x}, \widetilde{y}, \{H_{resnet}, \sum\nolimits_{cova}, Y_{ns}, Z_{ns}\}) \tag{3}$$

where the function H_{iSORT} can represent multiple iterative layers of the iSQRT-COV with the inputs $(\widetilde{x}, \widetilde{y})$ and the trainsfering papermeters H_{resnet} of ResNet101. $\sum_{cova}$ denotes the covariance matrix of the output of the last convolution layer. Based on the pre-normalization of $\sum_{cova}$ by trace norm, Y_{ns} and Z_{ns} are intermediate variables of Newton-Schulz iteration, which are suitable for parallel implementation on GPU, deriving the corresponding gradients of back propagation. Hence, with both architectures, the covariance matrix $\sum_{cova}$ is of size 256×256 and F_2 outputs an 32,896-dimensional vector as the image representation.

Subsequently, the WS-BAN model was described, which consists of bilinear attention pooling, weakly supervised attention learning and post-processing, to complete the proposed overall network structure for the fine-grained classification and object localization. The WS-BAN applies the VGG19 neural network backbone to generate feature maps and attention maps in two-steam parallel structure size by one or several convolutional operations from input image batches. Attention maps a_k are then split into M maps, reflecting the region of kth object's part. After that, feature maps F_V are element-wise multiplied by each attention map a_k with the same size in order to generate M part feature maps, which then are injected into additional local feature extraction function $g()$ to extract discriminative local feature representation. The final part feature matrix F_3 is concatenated by concatenating these local features with the bilinear attention pooling Γ,which can be represented by Equation(4)

$$F_3 = \Gamma(\Pi_{k=1}^{M} g(a_k \odot F_V\{\widetilde{x}, \widetilde{y}, H_{vgg}\})) \tag{4}$$

where H_{vgg} presents the hyper-parameters of VGG19 network. $\odot$ indicates element-wise multiplication for two feature tensors. In the following experiments, $g()$ is set as the global average pooling operation. During training, the initial learning rate is set to 0.001, with exponential decay of 0.85 after every 5 epochs. The weight of attention regularization is set to 1.0 and attention dropout factor is set to 80%. Then, the local feature map vector F_3 was obtained performed on $c_3 \times w_3 \times h_3$ dimensionality, where the size of feature map is $w_3 \times h_3 = 14 \times 14$ and the channel number is $c_3 = N^M$ with $N = 512$ and $M = 7$.

In addition to the proposed multi-stream structure, we also propose pre-training strategy to learn professional domain knowledge from the large-scale dataset. We initially pre-train our ResNet101 and VGG19 models on the ImageNet 2012 classification dataset [1] that consists of 1000 classes with the 1.28 million training images and the 50k validation images. Then fine-tune those models on a small-scale fine-grained dataset (*i.e.* CUB-200-201)). In this way, the network learns the common classification information and acquires domain knowledge during the pre-training process, and masters the fine-grained discriminative information during the fine-tuning process. This strategy enables the network to learn the features of the target dataset accurately and comprehensively, which can effectively improve the representation performance of neural networks on fine-grained small-scale datasets.

3) PROBABILITY FUSION MODULE

Given aforementioned feature extractor offering multi-dimensional object and part feature maps, we further propose a probability fusion module for decision fusion of various models. Our work aim to construct a general neural network component allowing for different gating decisions at each parts, and demonstrate its use as a practical way to massively increase model capacity. In this fusion module, the input is the various features of multi-models and the output is the fused classification probability, while the whole structure is divided as three main steps:

The first kernel step is to train a gating network which selects a sparse combination of three models to process each input and the inherent relations. The motivation behind this step is that we wish to recalibrate the strengths of different input representations of three feature extractors through a self-gating mechanism. Thus, a gating network was proposed inspired by the MOE introduced recently in reference [39], which considers complex non-linear interactions among activations of the input representation. To combine the input feature maps of the various size offered by different models, a concatenation layer is added in front of the gating network, which concatenates all features maps as an integral feature vector $f = concat(F_1, F_2, F_3)$. Then, the feature vector was passed to the gating network to capture the prior structure of the output label. For the output j-th samples of the i-th feature extractor, we denote the output of the gating network by G_{ij} on basis of the non-sparse softmax as Equation (5):

$$G_{ij} = Softmax(TopK(f \cdot W_g, k)) \tag{5}$$

where W_g present a trainable weight, matrix multiplied by the input vector for parameterizing the dependencies among each feature activations. Before taking the softmax function, a sparsity operation $TopK_i$ was added to keep only the top k values. Otherwise, this above form of sparsity set the rest to $-\infty$, which causes the corresponding gate values to equal 0. If we choose k $>$ 1, the gate values for the top k model have nonzero derivatives with respect to the weights of the gating network, which can be trained by simple

back-propagation, along with the rest of the model. With creating some theoretically scary discontinuities, the gating network chooses a sparse weighted combination of each extractor with computation save, whose output is a sparse $N \times 3n$ dimensions vector.

The second important step is to obtain the prediction scores of these three models, which respectively indicates the probability that the input belongs to the corresponding category in the i-th extractor. Based on the sparsity weighted vector, the gating network was used to transform the feature maps of each model before passing them to the fusion layers. We compute the output prediction probabilities of each model independently by written as

$$S_i = G_{ij} \odot E_{ij}(F_i) \tag{6}$$

where i is the number of component model with $i = \{1, 2, 3\}$ and j is the number of samples in the mini-batch with the upper limit value 512. G_{ij} is the sparsity component weight for the i-th model. There is a constraint that $\sum_{i=1}^{3} G_{ij} = 1$, so that the total probability distribution is normalized. S_i indicates the probability score of the i-th component model, and E_{ij} represents the information distribution representation of the corresponding i-th model extracted from input feature vector F_i, which should be defined to the same size matching with the G_{ij} dimension. Each of the three models will produce a feature vector with $N \times n$ dimensions variable uniformly, which is set as $n = 2048$.

However, as above Equation(2)-(4), F_1 and F_3 have more input channels comparing to F_2, which will increase time consumption if be calculated completely by convolution layers. Specifically, we employ the channel shuffle units [40] incorporate with convolution layers to encode the coarse-grained object features from ResNet101 and the fine-grained part features from WS-BAN synchronously. The channel shuffle operation first reshapes output channel dimension of F_1 and F_3 into several groups, transposing and flattening them back as the input of next convolution layer, which still takes effect even if the two convolutions have different numbers of groups. Moreover, channel shuffle is also differentiable, which means it can be embedded into network structures for end-to-end training. Taking advantage of this operation, the proposed probability fusion module can encode more information to compute the corresponding distributions E_{ij} efficiently with lightweight computational budget and low theoretical complexity. After statistical analysis, we choose the Gaussian distribution functions to fit multimodal distributions E_{ij} of each models for extracting key information from input data. The i-th component model's Gaussian distribution is defined as follows equation:

$$E_{ij}(\tilde{y}|\theta_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp(-\frac{(\tilde{y} - \mu_i)^2}{2\sigma_i^2}) \tag{7}$$

where $\theta_i = (\mu_i, \sigma_i^2)$ is estimated parameters consisting of the mean vector μ_i and covariance matrix σ_i^2 respectively. $\tilde{y} = \{\tilde{y}_1, \tilde{y}_2, \cdots, \tilde{y}_N\}$ is the output label vector corresponding to input feature maps, which are supposed to reflect substantive characteristics of original images. Though different features coming from different local parts and samples from three models exhibit tend to be more similar for the same category with semantic relevance, which helps to distinguish the small inter-class differences and combine the diversiform intra-class variations.

Therefore, **the third step** of the proposed module is applying Gaussian mixture method to construct the probabilistic fusion-based layer, which aims at adaptively clustering similar features at different models cluster centers and further combining three models probability scores to generate the final classification estimation. With the gated importance allocation of each models' Gaussian distribution in the whole architecture, the final perdition probability is defined as the linear combination of each model center. Let S denote the final output score, and it is calculated as follows:

$$S(\tilde{y}|\theta) = \sum_{i=1}^{3} S_i = \prod_{j=1}^{N} \prod_{i=1}^{3} G_{ij} E_{ij}(\tilde{y}|\theta_i) \tag{8}$$

where $\theta = (\theta_1, \theta_2, \theta_3)$ represents the Gaussian mixture parameters of the fusion layer. Thought the observation label vector $\tilde{y}$ is known, the detail translation from the i-th model's features to $\tilde{y}_j$ is still uncleaned. Thus, we introduce implicit variables γ_{ji} to represent this relation. By integrating the joint probabilities between $\tilde{y}_j$ and γ_{ij}, the fusion score S in Equation (8) is redefined to solve the likelihood function as follows:

$$\begin{aligned} S(\tilde{y}, \gamma|\theta) &= \prod_{i=1}^{3} S\left(\tilde{y}, \gamma_{j1}, \gamma_{j2}, \gamma_{j2}|\theta\right) \\ &= \prod_{i=1}^{3} \prod_{j=1}^{N} \left[G_{ij} E_{ij}\left(y_j|\theta_i\right)\right]^{\gamma_{ji}} \end{aligned} \tag{9}$$

Perform a log operation on the above expression, and take account of expression (7) to get the log likelihood expression:

$$\begin{aligned} \log S(\tilde{y}, \gamma|\theta) = \sum_{i=1}^{3} \Bigg\{ \sum_{j=1}^{N} \gamma_{ji} \log G_{ij} + \sum_{j=1}^{N} \gamma_{ji} \Bigg[\log\left(\frac{1}{\sqrt{2\pi}}\right) \\ - \log \sigma_i - \frac{1}{2\sigma_i^2}\left(y_j - \mu_i\right)^2 \Bigg] \Bigg\} \end{aligned} \tag{10}$$

Then, we choose the EM algorithm to estimate the hyper-dimensional parameters of the fusion layer. It can approach a local minimum of the maximum likelihood through differentiating the log likelihood strictly increases during iterations. The EM algorithm consists of two repeatedly iterative steps: E-step and M-step. Given the gating weight G_{ij}, the E-step calculates the expectation of implicit variables γ_{ji} defined as $\hat{\gamma}_{jk} = E\left(\gamma_{ji}|\tilde{y}, \theta\right)$, which denotes the responsivity of the i-th component model to the label y_j. Subsequently, The M-step updates the parameters by maximizing the expectations of the given log likelihood function in Equation (10).

We run EM algorithm for several iterations until parameters reaching the convergence value in fusion layer. Therefore, we obtain the detailed responsivity in E-step shown as following:

$$\begin{aligned}\hat{\gamma}_{ji} &= E\left(\gamma_{ji}|\tilde{y},\theta\right) \\ &= \frac{S\left(\gamma_{ji}=1,\tilde{y}_j|\theta\right)}{\sum_{i=1}^{3} S\left(\gamma_{ji}=1,\tilde{y}_j|\theta\right)} = \frac{G_{ij}E_{ij}\left(y_j|\theta_i\right)}{\sum_{i=1}^{3} G_{ij}E_{ij}\left(y_j|\theta_i\right)}\end{aligned} \tag{11}$$

Then in the M-step, we define the expectations of $\log S(y,\gamma|\theta)$ as M function as following:

$$\begin{aligned}M\left(\theta,\theta^{(t)}\right) &= E\left[\log S(\tilde{y},\gamma|\theta)|\tilde{y},\theta^{(t)}\right] \\ &= \sum_{i=1}^{3}\left\{\sum_{j=1}^{N}\hat{\gamma}_{ji}\log G_{ij} + \sum_{j=1}^{N}\hat{\gamma}_{ji}\left[\log\left(\frac{1}{\sqrt{2\pi}}\right)\right.\right. \\ &\quad \left.\left. -\log\sigma_i - \frac{1}{2\sigma_i^2}\left(y_j-\mu_i\right)^2\right]\right\}\end{aligned} \tag{12}$$

After t-step iteration, we hope that the new parameter estimation $\theta^{(t)}$ could make the $\log S(y,\gamma|\theta)$ function keep increasing until to the maximum. Otherwise, the parameters need to be continuously updated during each iteration, and the new iteration parameters are calculated in Equation (13):

$$\theta^{(t+1)} = \arg\max_{\theta} M\left(\theta,\theta^{(t)}\right) \tag{13}$$

Since θ_i is consisting of the mean vector and covariance matrix, we acquire the corresponding parameters $\hat{\mu}_i$ and $\hat{\sigma}_i^2$ by finding the partial derivative of variable μ_i and σ_i^2 with respect to Equation(12) set to 0. Similarly, we also update the gating parameter $\hat{G}_i$ under the constraint $\sum_{i=1}^{3} G_{ij} = 1$. The overall process of two-step loops from Equation (11) to (14) is terminated if any parameters move will no longer lead to further minimization of the M function, leading to the final estimation of labeling result. Therefore, the next iteration parameters are rewritten as Equation (14).

Notice that the inner EM-iteration loop is inside the fusion layer for parameter estimation, which is conducted in every external forward loop of the whole network's training. Therefore, the fusion layer with Gaussian mixture optimization is particularly independent, but it still can back propagate gradient through neighboring layers of the whole probability fusion module in a jointly trained way. With aforementioned operations, our proposed PFDM-Net gain an overall representation of prediction score in decision-level perspective, which actually is the joint posterior probabilities by integrating several prior probabilities from each component model of Multi-DNNs feature extractor. Finally, a min-max normalization layer and a softmax layer after the fusion layer were added to output the normalized classification result.

$$\begin{aligned}\frac{\partial M\left(\theta,\theta^{(t)}\right)}{\partial\mu_i} \rightarrow \hat{\mu}_i &= \frac{\sum_{j=1}^{N}\hat{\gamma}_{ji}\tilde{y}_j}{\sum_{j=1}^{N}\hat{\gamma}_{ji}} \\ \frac{\partial M\left(\theta,\theta^{(t)}\right)}{\partial\sigma_i^2} \rightarrow \hat{\sigma}_i^2 &= \frac{\sum_{j=1}^{N}\hat{\gamma}_{ji}\left(\tilde{y}_j-\mu_i\right)^2}{\sum_{j=1}^{N}\hat{\gamma}_{ji}} \\ \frac{\partial M\left(\theta,\theta^{(t)}\right)}{\partial G_i} \rightarrow \hat{G}_i &= \frac{\sum_{j=1}^{N}\hat{\gamma}_{ji}}{N}\end{aligned} \tag{14}$$

As the softmax operator obtaining predicted probabilities, the cross entropy loss is used to estimate the inconsistency degree between the predicted score and the true label. The optimal solution of CE loss is to minimize the error gap to small enough value with some regularization constraints including L1 or L2 terms. However, CE updates model parameters to make these two probability distributions similar to each other. It encourages the output scores dramatically distinctive, which potentially leads to overfitting for intra-class description. This easily leads to the low inter-class recognition in dealing with other categories. Thus, during training we optimize the following multi-part loss function:

$$\begin{aligned}Loss &= L_{fuse} + \lambda_1 L_{gate} + \lambda_2 L_{feature} \\ &= -\sum_{c=1}^{W}\tilde{y}_c\log(S_c) - \lambda_1\sum_{i=1}^{3}\sum_{j=1}^{N}G_{ij}TopK_i \\ &\quad -\lambda_2\sum_{i=1}^{3}\sum_{c=1}^{W}\tilde{y}_c\ softmax(F_i)\end{aligned} \tag{15}$$

where $L_{feature}$ indicates the loss of multi-DNNs feature extractor, L_{gate} indicates the information loss representation for the gating network, and L_{fuse} denotes the partial loss of probability fusion module. Moreover, we introduce two weighting factors $\lambda_1 \in [0, 0.5]$ and $\lambda_2 \in [0, 0.5]$ to balances the importance of each loss. Our proposed loss is a simple extension to CE that we consider as an experimental baseline to differentiate inter-class discrepancy among fine-grained categories. Specifically, W indicates the number of categories. $\tilde{y}_c$ indicates the indicator variable (0 or 1) if the category and sample have the same category, otherwise 0. S_c denotes the predicted probability that the observed sample belongs to category c. Similarly, we obtain the whole loss of three models extracting various feature maps F_i with the softmax function. Then we construct the gating loss based on the Equation (5) by comparing G_{ij} and $TopK_i$. In subsequent experimental results, the above loss form was used to optimize the entire model structure, which is demonstrated to be effective to improve the performance for FGVC tasks.

TABLE 1. The statistics of three fine-grained datasets.

Datasets	Category	Training	Testing
CUB-200-2011[9]	200	5994	5794
Stanford Cars[10]	196	8144	8041
Stanford Dogs[11]	120	12000	8580

IV. EXPERIMENTS

A. EXPERIMENTAL SETTINGS

We conduct experiments on three challenging fine-grained image recognition datasets, including CUB-200-2011 (Caltech-UCSD Birds), Stanford Cars and Stanford Dogs. The detailed statistics with category numbers and data splits are summarized in Table 1.

1) CUB-200-2011

It is one of the most popular datasets for fine-grained image classification. It has 11,788 images from 200 bird subordinates. 5,994 images are selected for training, while the rest 5,794 images for testing. Approximately, 30 images are used in training for each subordinate, and 11 to 30 images for testing. In addition, it provides the most detailed annotations among datasets, including a subordinate label, a bounding box, 15-part locations and 312 binary attributes for each image.

2) STANFORD CARS

It is a collection of car models, which contains 16,185 images of 196 subordinates. 8,144 and 8,041 images are selected as training set and testing set respectively. Each subordinate has 24-68 training images and 24-68 testing images variably. Labels at the level of Make, Model and Year, along with a bounding box, are provided.

3) STANFORD DOGS

It has 20,580 images of 120 dog breeds. It is divided as follows: 12,000 images for training and 8,580 images for testing. Each breed has 100 training images and 48-to-152 testing images. Class labels and bounding boxes are annotated.

To avoid over-fitting of the network, aforementioned augmentations consist of geometrical transformations (resizing, random crop, rotation and horizontal flipping, aspect ratio, encoding, mixup) and intensity transformations (contrast and brightness enhancement, color, noise) are applied to enhance a larger number of images in the dataset before training and testing on three datasets. With the pre-training of ResNet101 and VGG19 on ImageNet, the WSL models consisting of ResNet101, WS-BAN and iSQRT-COV are training performed on three Fine-grained image classification datasets. All models are trained and tested on an Intel Core i7 3.6 GHz processor with four NVIDIA Tesla p40 GPUs and 512G RAM. The training is proceeded on the training set, after that the evaluation is performed on the validation set for minimizing overfitting. When the training process and parameter selection are achieved, the final evaluation is done on the unknown testing set for evaluating the performance. Training batches of size 512 are created by uniformly sampling from all available training images as opposed to sampling uniformly from the classes. We fine-tuned all networks from pretrained weights with a learning rate of 0.0001, decayed exponentially by 0.94 every four epochs, and RMSProp optimization with momentum and decay both set to 0.9. Images are performed within 448 pixels in size, with a single centered crop at test time.

TABLE 2. Comparison of results on CUB-200-2011.

Method	Train Anno.	Accuracy
Part-RCNN(AlexNet) [18]	√	76.4
Deep-LAC [19]	√	80.3
SPDA-CNN [20]	√	85.2
ResNet101 [5]		86.6
TLAN(AlexNet) [21]		77.9
B-CNN [22]		84.1
OPAM [23]		85.8
ST-CNN [24]		84.1
RA-CNN [25]		85.3
WS-BAN [27]		88.8
HBP [28]		87.1
DFL [29]		87.4
MAMC [30]		86.5
GMNet [31]		86.3
iSQRT-COV [33]		88.7
NTS-Net [36]		87.5
PFDM-Net (WS-BAN + iSQRT-COV)		**90.3**
PFDM-Net (WS-BAN + ResNet101)		**89.7**
PFDM-Net (WS-BAN+ResNet101+ iSQRT-COV)		**91.2**

B. EXPERIMENTS ON CUB-200-2011

Our proposed PFDM-Net model is performed to achieve better or comparable results for fine-grained image classification on the CUB-200-2011 dataset. The feature maps of the three models ResNet101, WS-BAN, iSQRT-COV is fused to obtain the final classification results. The accuracy (the ratio between the number of correctly classified images and the number of testing images) is used to evaluate the performance. The whole accuracy of the proposed model comparing with some previous methods are shown in following Table 2.

As shown in Table 3, the results obtained by using the probability fusion decision model are much higher than the current excellent models. The accuracy result obtained by the PFDM-Net model after gating weighted fusing multi-stream component models is 91.2 %. In contrast, the best strong supervised method (SPDA-CNN) using the training label box to classify the fine-grained data only obtain 85.2%

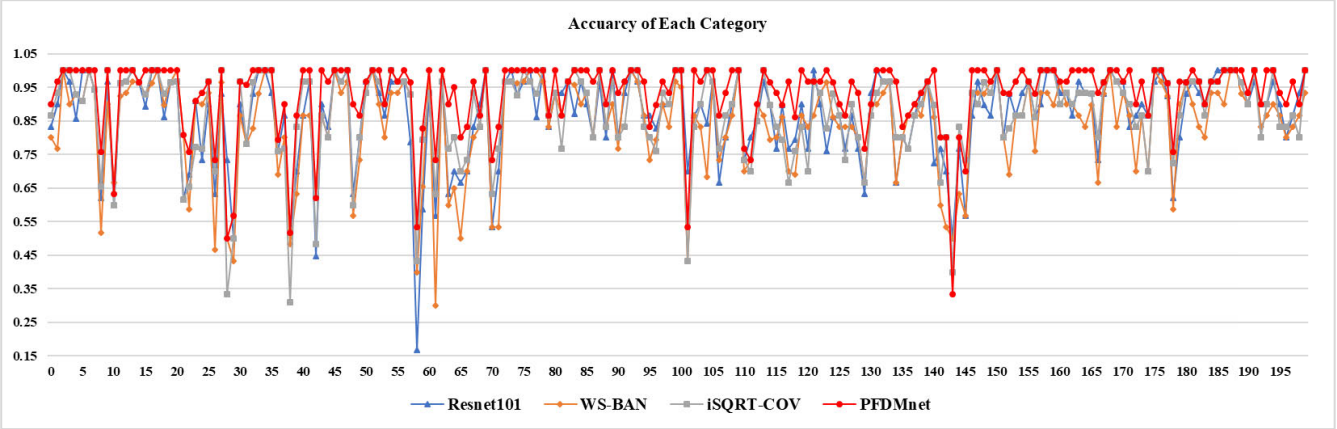

FIGURE 5. Accuracy of Each type achieved by ResNet101, WS-BAN, iSQRT-COV and PFDM-Net models on CUB-200-2011 dataset.

TABLE 3. Comparison of results on stanford cars.

Method	Train Anno.	Accuracy
SPDA-CNN [20]	√	93.1
ResNet101 [5]		92.9
B-CNN [22]		91.3
OPAM [23]		92.2
RA-CNN [25]		92.5
WS-BAN [27]		93.6
HBP [28]		93.7
DFL [29]		93.8
MAMC [30]		93
GMNet [31]		93.5
iSQRT-COV [33]		93.3
NTS-Net [36]		93.9
PFDM-Net (WS-BAN + iSQRT-COV)		**94.1**
PFDM-Net (WS-BAN + ResNet101)		**93.9**
PFDM-Net (WS-BAN+ResNet101+ iSQRT-COV)		**95.2**

accuracy with 6% lower than the accuracy of our fusion model, which only need subordinate labels and performs even better than those requiring extra information. For weak supervision training without training box, two baselines NTS-Net and GMNet achieve the accuracies with 87.5% and 86.3% respectively, which also prove the higher effectiveness of our framework.

As for each component model of multi-stream feature extractor, the accuracy of the ResNet101 model is 86.6% only rely on the coarse image-level labels. It is proven that the deeper structure of ResNet101 could extract impressive and effective feature maps, which is suitable for focusing on further local information. Similarly, the accuracy of WS-BAN is 88.8% and the accuracy of iSQRT-COV is 87.2%, which illustrates the actual rule that more tricks and operations acquiring abundant local feature representation of discriminative object's parts do improve the higher accuracy for solve FGVC tasks. After our fusing processing, the accuracy relying on the proposed probability fusion module increase much higher in the range of 2.4% to 4.6% than the previous single model. Even with two models merged, our fusion module can still play the important role in excavating complementary characteristic of different models to get higher precision performance. Among the results, the combining form of WS-BAN and iSQRT-COV acquires the 90.3% accuracy, and the other accuracy by combining WS-BAN and ResNet101 is up to 89.7%, both of which perform better than single component model or other WLS methods. It is certified that the designed probability fusion module in decision-level viewpoint do utilize the mixture-granularity information of multiple DNNs by only using image-level labels. And the end-to-end implementation of fusion module with an inner-to-outer loop effectively improve the overall accuracy of the FGVC problem well.

Based on the obtained results, we farther analyze the detail accuracy of each subclass offered by our PFDM-Net and three baseline models before fusion on the CUB-200-2011 dataset. Quantitative analysis of each model are as shown in Figure 5. Although the individual models have different recognition capabilities for different categories, it can be seen that the trend of accuracy curve owing to the PFDM-Net (red line) is relatively flat and stable. It means that the fusion strategy can mine some small inter-class differences which is not available for a single model to improve the recognition rate of different sub-classes, further balance uncertainty among different models. With combining complemental feature maps, our model performs the high capabilities of distinguishing inter-class discrepancy for each image of different fine-grained type. For example, ResNet101 has an accuracy of 16.7% in the 58^{th} category and 66.7% in the 66^{th} category. After fusion module, the accuracy of the 58^{th} category is improved up to 53.3% and the corresponding accuracy of the 66^{th} category is up to 83.3%. The probability fusion module sincerely reduces the recognition difference of an individual models covering different categories, thereby improving the overall accuracy.

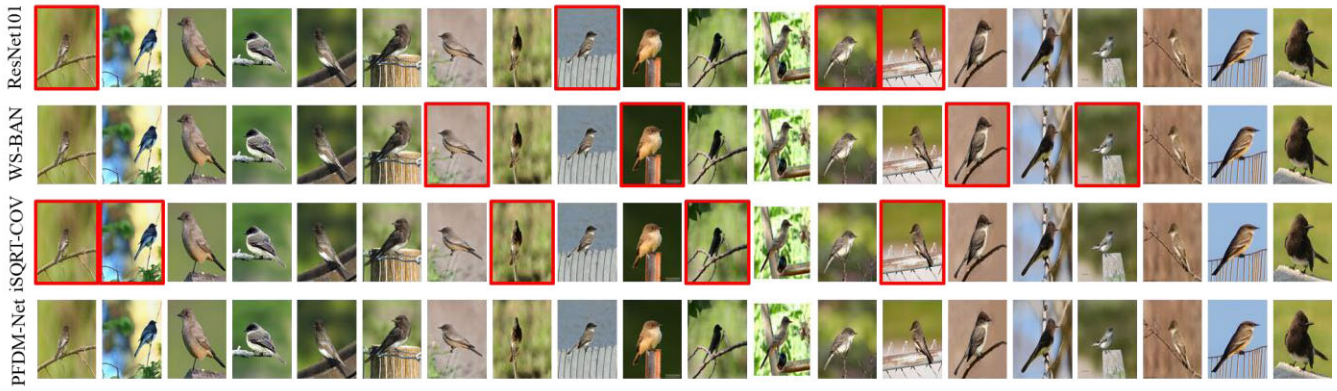

FIGURE 6. Model classification results on the 102nd class of CUB-200-2011.

Relatively, the accuracy of different models in the same category is obvious large. For example, in the 102nd category with the species name as Sayornis, the accuracy of ResNet101 is 80.3%, while the accuracy of WS-BAN and iSQRT-COV is 86.7% and 90.2% respectively. It can be found that the intra-class variables among different images and samples of the same sub-class are effectively suppressed by the fusion technology. The interference factors such as position changes or lighting varieties preventing the accuracy improvement are limited in a certain degree for all categories in our fusion method, which leads to the surprising accuracy up to 100% in the 102nd category. We detailed analyze the complementation process of intra-class error in the 102nd class as shown in Figure 6. For the 20 images of Sayornis class, the red boxes represent that the model predicts the inside picture as other wrong categories, and the rest represent the right predicted images. The results show that three component models all make some mistakes, and only our PFDM-Net model is completely correct. It is demonstrated that the fusion layer weighted by the gating network could reasonably select domain information from various feature maps extracted from each individual model, and enhance the controllability of interclass and intra-class variables to improve the holistic performance of FGVC task.

C. EXPERIMENTS ON STANFORD CARS

Similarly, we used PFDM-Net model to perform FGVC experiments in the Stanford Cars dataset. The final accuracy of the PFDM-Net model and some state-of-the-arts competitors as listed in Table 3.

As above shown. each accuracy of different models are relatively higher than their results on CUB-200-2011. The main reason is that there are more trainable and testable images and fewer categories, which meets the needs of deep learning training to maximize their performance. Moreover, the differences between vehicles are obvious compared to birds in complex environmental background, which make it easy to achieve the better accuracy results. Even so, our proposed PFDM-Net model still obtains the best performance on the accuracy indicator with 95.2%, which is 1.3% higher than the current NTS-Net with 93.9% accuracy. This illustrates that our model is adapted to different datasets and tasks with good generalization performance, which also provides an effective way to solve other practical FGVC applications with more complex backgrounds.

In contrast, each component model in the multi-stream architecture also acquire decent result severally. The accuracy of the ResNet101 model is 92.9%, which is close to the 93.1% accuracy of SPDA-CNN with affluent part annotations. while, the accuracy of WS-BAN is 93.6% and the accuracy of iSQRT-COV is 93.3%. After comparison, it can be found that the accuracy of the two model merged is still much higher than that of the previous single model, and can be increased by up to 94.1% (combining WS-BAN and iSQRT-COV) and 93.9% (combining WS-BAN and ResNet101). Both of two results are better or as good as other single WLS models.

As shown in Figure 7, it shows that each single model has different preference in recognizing different analogical categories. However, the fusion model can balance the advantages and disadvantages of each component model to achieve better local performance in the same category. For example, in the 70th category with the basic-level label ''Chevrolet Express Van 2007'', the accuracy of iSQRT-COV is 58%, and the accuracy of WS-BAN is 51.4%. while ResNet101 only achieve the 40.0% accuracy, which is relatedly low probability even less than random guessing 50%. Thought the fusion operation with iteration and gradient back-propagation optimization, the accuracy of the 70th category is improved up to 69.6%. Similar result is illustrated in Figure 8 by analyzing the 24th class car named as ''Audi TT RS Coupe 2012''. As shown 20 pictures of this category, ResNet101 incorrectly identifies 9 samples as other categories, such as Tesla and BMW types. WS-BAN and iSQRT-COV also have similar poor performance on Audi prediction with 5 red wrong images and 4 representatives severally. It shows that our PFDM-Net model can fuse the feature extraction capabilities for better fine-grained car type classification.

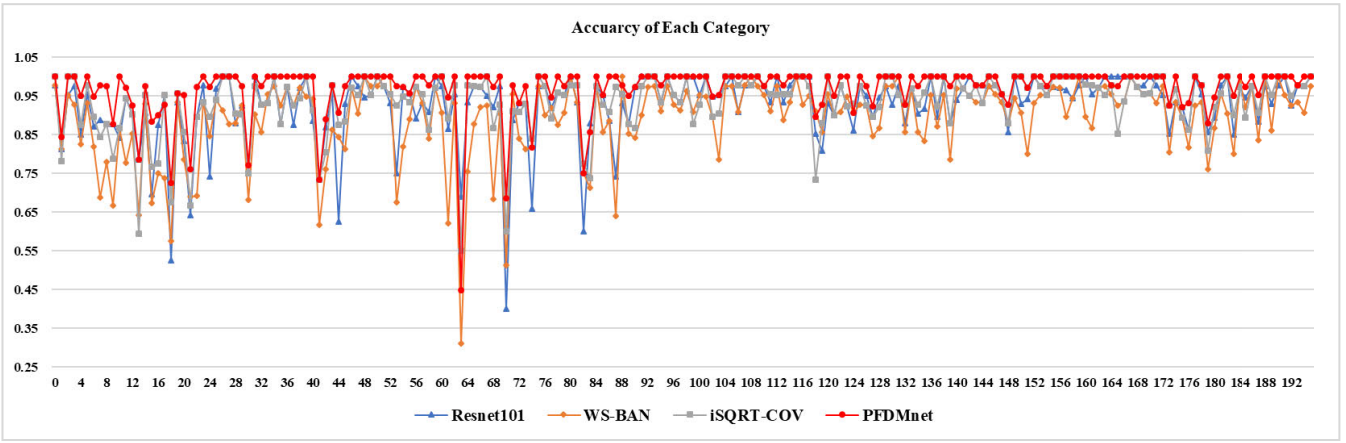

FIGURE 7. Accuracy of each type achieved by the ResNet101, WS-BAN, iSQRT-COV and Fusion models on Stanford Cars dataset.

TABLE 4. Comparison of results on stanford dogs.

Method	Train Anno.	Accuracy
Part R-CNN[18]	√	78.9
ResNet-50 [5]		81.1
ResNet-101 [5]		84.9
RA-CNN [25]		87.3
WS-BAN [27]		87.5
MAMC(ResNet-50) [30]		84.8
MAMC(ResNet-101) [30]		85.2
GMNet [31]		88.1
iSQRT-COV [33]		88.1
PFDM-Net (WS-BAN + iSQRT-COV)		**88.8**
PFDM-Net (WS-BAN + ResNet101)		**88.7**
PFDM-Net (WS-BAN+ResNet101+ iSQRT-COV)		**89.6**

D. EXPERIMENTS ON STANFORD DOGS

Further, we take the comparable experiments to demonstrate the classification results on Stanford Dogs in Table 4. Part R-CNN is one of the early works in this field. It extracts traditional features for different parts of image and introduces both some supervised alignments to the object. It only gets a 79.8% accuracy on this dataset, which demonstrates that the feature extraction process is crucial for fine-grained classification. RA-CNN designs an unsupervised part model discovery method by selecting prominent parts, it obtains 87.3% accuracy. Currently, the best result is achieved by the GMNet with 88.1% accuracy. While the three component models consisting of ResNet101 WS-BAN and iSQRT-COV achieve the approximate results with 84.9%, 87.5% and 88.1%. On this dataset, PFDM-Net also outperforms all state-of-the-arts SSL and WSL methods, reaching 89.6% accuracy increased by at least 1.5%.

The identification curves of PFDM-Net and its component models for each type are illustrated in Figure 9. The red curve denoting our models is relatively stable without much fluctuation, which indicates the better ability for distinguishing small interclass differences. Thought the fusion model can perform well overall on the Stanford Dogs with more training data, but the result does not match the better performance comparing with the abovementioned databases in each category. This is because the image number of each type is very unbalanced, and the biological morphology of pet dogs varies greatly and changes with the growth cycle. What's worse, the background of this dataset is more complicated filled with vehicles, humans, and daily necessities, which extend the intra-class variation and make it more difficult for fine-grained recognition.

Figure 10 illustrates the 1st class in the Stanford Dogs dataset, the category name is Chihuahua. For the 20 pictures selected randomly, our PFDM-Net model outperforms other methods and only make two mistakes represented by red boxes. One wrong image of them has a dog with black and white pattern, which is significantly different from the other samples. And in another wrong image, the dog is blocked by its master and could not be seen, which made all networks invalid. These special cases frequently occurrence in this Stanford Dogs dataset, which limited the farther performance improvement in the local and global accuracy of our PFDM-Net model.

V. DISCUSSION

A. LOSS FUNCTION

The loss function is used to estimate the degree of inconsistency between the predicted value f(x) of your model and the true value Y. It is a non-negative real-valued function, usually expressed by L (Y, f(x)). The smaller the loss function, the better the robustness of the model. The loss function is the core part of the empirical risk function. Figure 11 shows the loss function diagram for the four models in the CUB-200-2011 dataset.

Figure 11 shows the loss function diagram for the four models. A comparison of the loss functions shows that the

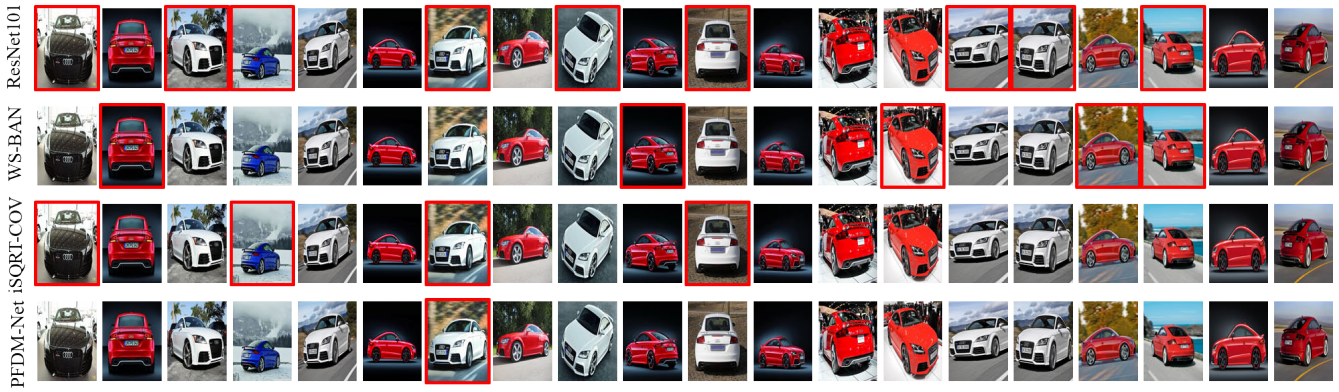

FIGURE 8. Model classification results on the 24th class of Stanford Cars.

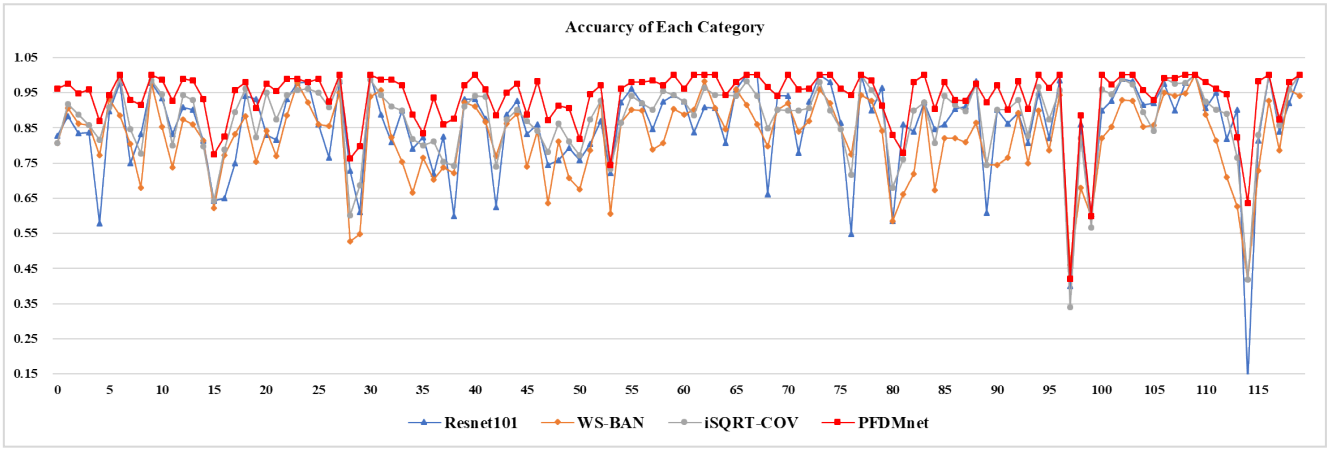

FIGURE 9. Accuracy of each type achieved by the ResNet101, WS-BAN, iSQRT-COV and Fusion models on the Stanford Cars dataset.

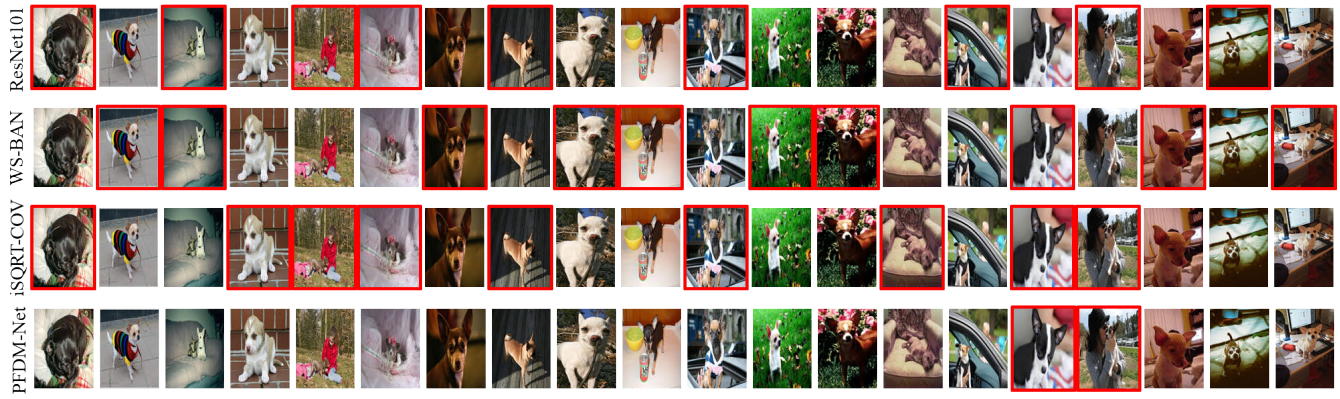

FIGURE 10. Model classification results on the 1st class of Stanford Dogs.

trend of the loss function values of the four classification networks is decreasing in the CUB-200-2011 dataset. It stabilizes for about 90 generations and the predicted value is closer to the true value. Between them, the PFDM-Net model has the fastest loss function, and the final loss function drops to about 0.0084. The loss function of ResNet101, WS-BAN and iSQRT-COV decreases slowly, and the loss function drops to around 0.0186. The PFDM-Net model fuses the confidence of the three baseline models, combining the ability of the three baseline models to extract features. The ability to extract features from the fusion model is enhanced, and the model has better generalization. The fusion model is more capable

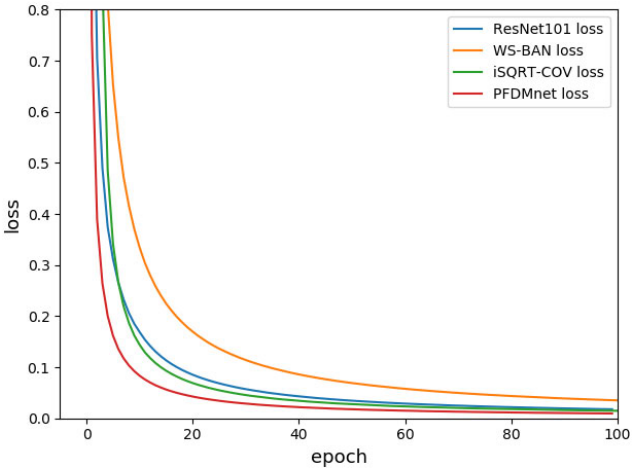

FIGURE 11. Loss function graphs of each classification network.

of extracting features of fine-grained images, and can better classify fine-grained images, thereby improving the accuracy of model classification.

B. COMPLEXITY ANALYSIS

The PFDM-Net model developed a probabilistic fusion module with a gating network and a probabilistic fusion layer to fuse different component models with a Gaussian distribution. This process produces a relatively large number of parameters, which increases the time for picture testing. Table 5 shows the parameter quantities and classification times for each model. Experiments show that the classification time of the PFDM-Net model is 25MS per picture, compared with ResNet101, WS-BAN and iSQRT-COV, the classification time is about 14ms slower. But this speed can still achieve real-time classification.

The complexity of ResNet101, WS-BAN, iSQRT-COV and PFDMnet models was analyzed, while we also used other weakly supervised models and four models for multi-stream feature extraction. It is found that the model fusion of ResNet101, WS-BAN and iSQRT-COV has the highest classification accuracy compared with other models. At the same time, the classification accuracy of the four models used for multi-stream feature extraction is not significantly

TABLE 5. Model complexity and time analysis.

Model	parameter	Time
Resnet101	343M	0.13ms
WS-BAN	278M	0.09ms
iSQRT-COV	534M	0.18ms
PFDMnet	686M	0.25ms

improved, but the model complexity becomes very large. Therefore, we decided to use three models as multi-stream feature extraction to find a balance between accuracy and model complexity.

C. CONFUSION MATRIX

Because the patterns displayed in each category have different complexity, especially in terms of categories and backgrounds, the system does not accurately distinguish between several categories, resulting in reduced accuracy. In Figure 12, we present a confusion matrix for the final CUB-200-2011 test results. Four confusion matrixes, which compared the true category (Ordinate) against the predicted category (Abscissa), were calculated to describe the individual classification rate of each model. By analyzing the confusion matrix, we can directly evaluate the performance of the network. In addition, the confusion matrix helps us further analyze the program to avoid re-confusion between these different classes. It can be seen from the Figure 13 that some of the Resnet101, WS-BAN and iSQRT-COV models have lower predictions and true correlations. This indicates that the network has some confusion for the similarity category, resulting in low detection accuracy. For our PFDM-Net model, the PFDM-Net network can be well classified for the similarity categories, so that the prediction and the true correlation are relatively high, thus improving the detection accuracy.

The results of the confusion matrix for the test set of the Stanford Cars dataset and the Stanford Dogs dataset are shown in Figure 13 and Figure 14. By analyzing the following two figures, the confusion matrix helps us further analyze the program and avoid re-ambiguity between these

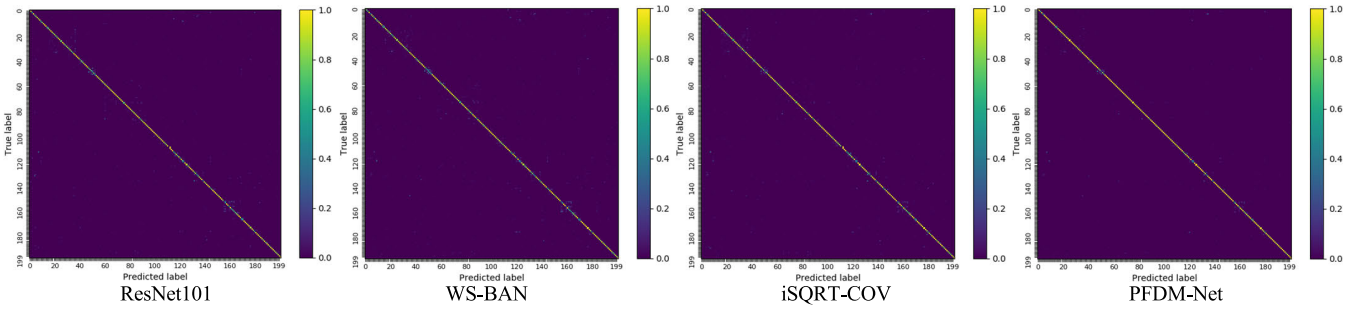

FIGURE 12. Confusion matrix of detection results on the CUB-200-2011.

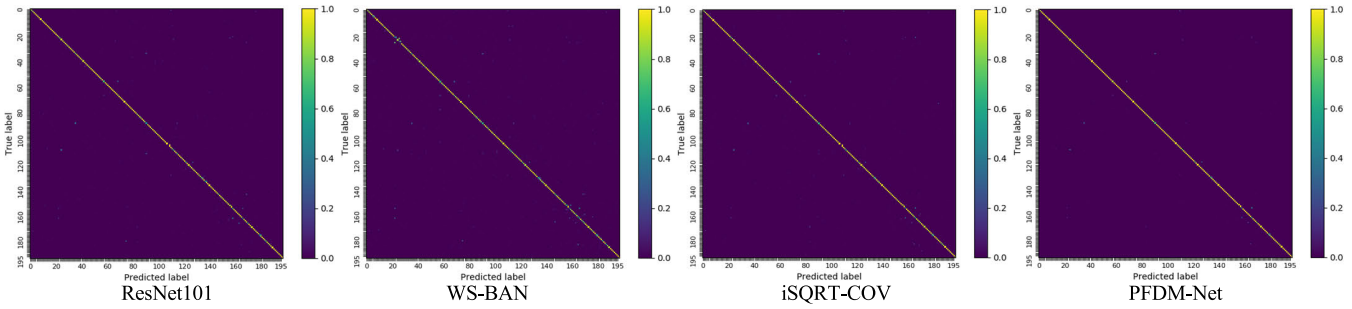

FIGURE 13. Confusion matrix of detection results on the Stanford Cars.

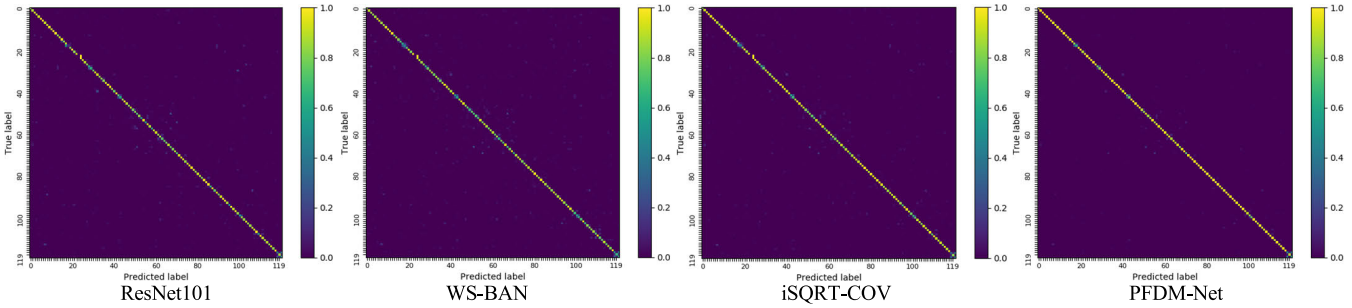

FIGURE 14. Confusion matrix of detection results on the Stanford Dogs.

different classes. As seen from those figures, the Resnet101, WS-BAN and iSQRT-COV models have lower predictions and true correlations in the Stanford Cars dataset and the Stanford Dogs dataset, resulting in lower classification accuracy. The PFDM-Net model fuses the characteristics of its baseline model, so it has high prediction and true correlation in the Stanford Cars dataset and Stanford Dogs dataset, thus improving classification accuracy.

VI. CONCLUSION

In this paper, we design a novel probability fusion decision framework named as PFDM-Net for fine-grained visual classification tasks. It contains three elements, Data Augmentation, Multi-steam Feature Extractor and Probability fusion module. The image can be accurately classified through these three parts. Where Data Augmentation employs data augmentation tricks including mixup to enlarge the dataset. In addition, refined multiple DNNs consisting of ResNet101, WS-BAN and iSQRT-COV are applied to design a multi-stream feature extractor, which utilizes the mixture-granularity information to exploit features distinguishing interclass and intra-class variances. Finally, a probability fusion module equipped with gating network and probability fusion layer is developed to fuse different components model with Gaussian distribution. It constructs a general neural network component allowing for different gating decisions at each part, and demonstrate its use as a practical way to massively increase model capacity. The whole framework contains an inner loop about the EM algorithm and an outer loop with the gradient back-propagation optimization. Experiments demonstrate the effectiveness of the PFDM-Net with higher accuracy up to 91.2% on CUB-200-2011, 95.2% on Stanford Cars, and 89.6% on Stanford Dogs, which outperform state-of-the-arts FGVC methods.

The network generates large number of parameters, which increases the time for model classification. Therefore, the model can only be trained on the GPU and the image cannot be classified online on a convenient mobile device. Our future works are to try lightweight network to compress the model parameters and speed. And we will attempt to combine our method with the newly updated works for other practical FGVC applications, such as agricultural Internet of Things, food supply chain security.

REFERENCES

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

[2] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, vol. 8693. Cham, Switzerland: Springer, 2014, pp. 740–755.

[3] L. Han, C. Yu, K. Xiao, and X. Zhao, "A new method of mixed gas identification based on a convolutional neural network for time series classification," *Sensors*, vol. 19, no. 9, pp. 1960–1982, 2019.

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556

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

[6] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4700–4708.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012, pp. 1097–1105.

[8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.

[9] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: https://arxiv.org/abs/1502.03167

[10] H. Azizpour and I. Laptev, "Object detection using strongly-supervised deformable part models," in *Proc. Eur. Conf. Comput. Vis.*, vol. 7572. Berlin, Germany: Springer, 2012, pp. 836–849.

[11] B. Yu, J. Pan, D. Gray, J. Hu, C. Choudhary, A. C. A. Nascimento, and M. De Cock, "Weakly supervised deep learning for the detection of domain generation algorithms," *IEEE Access*, vol. 7, pp. 51542–51556, 2019.

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

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

[14] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei, "Novel dataset for fine-grained image categorization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. 1st Workshop Fine-Grained Vis. Categorization*, Colorado Springs, CO, USA, Jun. 2011, pp. 1–2.

[15] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "DeCAF: A deep convolutional activation feature for generic visual recognition," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 647–655.

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[17] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, "Part-based R-CNNs for fine-grained category detection," in *Proc. Eur. Conf. Comput. Vis.*, vol. 8689. Cham, Switzerland: Springer, 2014, pp. 834–849.

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

[19] H. Zhang, T. Xu, M. Elhoseiny, X. Huang, S. Zhang, A. Elgammal, and D. Metaxas, "SPDA-CNN: Unifying semantic part detection and abstraction for fine-grained recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1143–1152.

[20] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, and Z. Zhang, "The application of two-level attention models in deep convolutional neural network for fine-grained image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 842–850.

[21] T. Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear CNN models for fine-grained visual recognition," in *Proc. IEEE Conf. Comput. Vis.*, Dec. 2015, pp. 1449–1457.

[22] Y. Peng, X. He, and J. Zhao, "Object-part attention model for fine-grained image classification," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1487–1500, Mar. 2018.

[23] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2017–2025.

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

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

[26] T. Hu, J. Xu, C. Huang, H. Qi, Q. Huang, and Y. Lu, "Weakly supervised bilinear attention network for fine-grained visual classification," 2018, *arXiv:1808.02152*. [Online]. Available: https://arxiv.org/abs/1808.02152

[27] C. Yu, X. Zhao, Q. Zheng, P. Zhang, and X. You, "Hierarchical bilinear pooling for fine-grained visual recognition," in *Proc. ECCV Conf. Comput. Vis.*, 2018, pp. 595–610.

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

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

[30] J. Liang, J. Guo, X. Liu, and S. Lao, "Fine-grained image classification with Gaussian mixture layer," *IEEE Access*, vol. 6, pp. 53356–53367, 2018.

[31] P. Li, J. Xie, Q. Wang, and W. Zuo, "Is second-order information helpful for large-scale visual recognition?" in *Proc. IEEE Conf. Comput. Vis.*, Oct. 2017, pp. 2070–2078.

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

[33] T. Robert, N. Thome, and M. Cord, "HybridNet: Classification and reconstruction cooperation for semi-supervised learning," in *Proc. ECCV Conf. Comput. Vis.*, Sep. 2018, pp. 153–169.

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

[35] Z. Yang, T. Luo, D. Wang, Z. Hu, J. Gao, and L. Wang, "Learning to Navigate for Fine-grained Classification," in *Proc. ECCV Conf. Comput. Vis.*, Sep. 2018, pp. 420–435.

[36] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," 2017, *arXiv:1701.06538*. [Online]. Available: https://arxiv.org/abs/1701.06538

[37] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of tricks for image classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 558–567.

[38] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," 2017, *arXiv:1710.09412*. [Online]. Available: https://arxiv.org/abs/1710.09412

[39] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, May 2017, pp. 4278–4284.

[40] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6848–6856.