UTS UNIVERSITY OF TECHNOLOGY SYDNEY

Deep Learning Based Fine-Grained Species Identification

by Qiyu Liao

Thesis submitted in fulfilment of the requirements for the degree of

C02029 Doctor of Philosophy

under the supervision of Min Xu, Dadong Wang

University of Technology Sydney
Faculty of Engineering and Information Technology

March, 2021

Certificate of Original Authorship Template

Graduate research students are required to make a declaration of original authorship when they submit the thesis for examination and in the final bound copies. Please note, the Research Training Program (RTP) statement is for all students. The Certificate of Original Authorship must be placed within the thesis, immediately after the thesis title page.

Required wording for the certificate of original authorship

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, Qiyu Liao declare that this thesis, is submitted in fulfilment of the requirements for the award C02029 Doctor of Philosophy, in the Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.
**If applicable, the above statement must be replaced with the collaborative doctoral degree statement (see below).*

**If applicable, the Indigenous Cultural and Intellectual Property (ICIP) statement must be added (see below).*

This research is supported by the Australian Government Research Training Program.

Signature: Production Note:
Signature removed prior to publication.

Date: 18/11/2021

Collaborative doctoral research degree statement

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of the requirements for a degree at any other academic institution except as fully acknowledged within the text. This thesis is the result of a Collaborative Doctoral Research Degree program with Commonwealth Scientific and Industrial Research Organisation (CSIRO).

Indigenous Cultural and Intellectual Property (ICIP) statement

This thesis includes Indigenous Cultural and Intellectual Property (ICIP) belonging to UTS and CSIRO, custodians or traditional owners. Where I have used ICIP, I have followed the relevant protocols and consulted with appropriate Indigenous people/communities about its inclusion in my thesis. ICIP rights are Indigenous heritage and will always remain with these groups. To use, adapt or reference the ICIP contained in this work, you will need to consult with the relevant Indigenous groups and follow cultural protocols.

Abstract

Fine-Grained Visual Categorization (FGVC) is a challenging research topic in computer vision. It deals with the classification of visual data at a subordinate level. This thesis investigates four categories of FGVC methods based on deep learning, including general convolutional neural networks, object part localization methods, approaches using CNN ensemble or higher-order feature encoding, and methods utilizing recurrent visual attention. Overall performance comparison has been conducted to analyse their advantages and disadvantages.

We proposed a new regression-based part detection structure and a novel part-based model, which increased the classification accuracy of PS-CNN from 76.4% to 82.4% on the CUB-200-2011 benchmark dataset.

Inspired by the second-order pooling, we proposed a highly interpretable method with a compressed structure to significantly reduce the computation complexity while improving the fine-grained categorization accuracy. The proposed model provides a supervised selection of the most discriminative second-order channels. With the proposed method, the computation and the feature dimension are linearly reduced to 4% of the original bilinear pooling. By applying matrix normalization and a Fisher-Recurrent-Attention structure, we achieved the best result among the VGG-16 based FGVC models.

Following the conception of attention crop and attention drop in the Fisher-Recurrent-Attention model, we proposed a forcing module to constrain the network to extract more diverse features for FGVC. The forcing module focuses more on confusion regions which are essential for the fine-grained classification. Experimental results show that the proposed forcing module can improve the attention and prediction of the network when an input image is panned or zoomed, and the double prediction performs better than the single prediction.

The existing FGVC methods often come with enormous amounts of computation and require large memory space. This makes these models inadequate for mobile applications. We proposed a Category Attention Transferring Convolutional Neural Network (CAT-CNN) to transfer the attention knowledge from a large-scale FGVC network to a small but efficient network to improve its presentation capability. Using the proposed model, we improved the classification accuracy of the efficient networks by up to 5.7% on the CUB-2011-200 dataset without increasing computation time or memory cost, which makes FGVC feasible on mobile devices.

We also conducted abundant studies to investigate the relationship between attention and classification accuracy of our proposed deep learning models, visualized and analysed the attentional activations of these models. We hope that our findings may inspire further research efforts to advance the FGVC for a wide range of real-world applications.

keywords: Image classification, Fine-grained classification, Deep learning, Species identification.

Contents

List of Figures

Chapter 1

Introduction

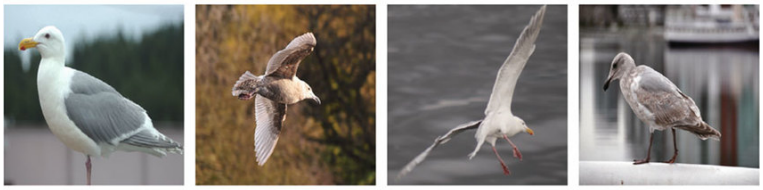
In the past years, deep learning has immensely excelled on many challenging visual tasks, such as image classification, speech recognition, and natural language processing (NLP). Different from traditional image processing systems in which the features are artificially designed, the deep features are extracted from plentiful annotated data with a general training regulation. Numerous variants of the deep learning structures have been proposed in recent years, most of which are deuterogenic from several well-known parent structures.

The convolutional layer is first proposed in [24]. The layer consists of learnable filter kernels with parameters. The image's receptive field is shrunk through the convolutional layers, but the image features are extended to the full depth of the output volume. [53] first proposed the convolutional layers' backpropagation approach and introduced the first Convolutional Neural Network (CNN) structure. The CNN structure was defined as an artificial neural network that contains one or several convolutional layers. A standard MultiLayer Perceptron (MLP) was constructed following CNN using one or more fully connected layers. With the backpropagation method, we can train a CNN structure end-to-end using a set of pre-annotated image data. It's prevalent to apply an optimization solver (usually gradient descent) to calculate the gradient of a loss function and update all the network weights to minimize the loss between the label and the CNN network's output.

The CNN model is an efficient way to extract semantic image features for that it learns deep information from the real image data and obtains an approximate optimal solution. Deep CNN structures have many variants, and they are widely used in real applications like object detection, image classification, semantic segmentation, and image retrieval systems.

Different from the traditional image classification task where categories display a huge difference in morphology, fine-grained visual categorization (FGVC) mainly

(a) Calilfornia gull



(b) Glaucous gull

Figure 1.1: Two subclasses of gulls from the general FGVC dataset, CUB-2011-200, illustrate the major challenge in FGVC. (a) and (b) shows California gulls and Glaucous gulls, respectively. The inner-class variances, e.g., pose, scale, etc., are more visually obvious than the inter-class variances.

focus on distinguishing between subordinate-level categories within the same basic-level category, e.g., different kinds of birds [90], dogs [49], cars [51], and aircrafts [66]. FGVC is more challenging than the traditional image classification tasks because of high intra-class variances and low inter-class variances between the subordinate-level categories. There are minute differences between different categories. It is difficult for a model to correctly distinguish between remarkably similar-looking categories until it extracts the subtle and discriminative difference. Figure 1.1 presents two different gulls with a small variance between species but an enormous variance within the same class. The pose of the objects, viewpoints, and background of the same species' images differ very vastly, while different subclasses are very similar to each other. Moreover, we could only discriminate these species by their very tiny components, e.g., the legs or tails.

Compared with the well-studied traditional image visual classification that distinguishes an object's main class, FGVC is closer to real-world applications. There are plenty of demands from different fields of agriculture, animal husbandry, health protection, environmental protection, taxonomy, and commodity management. Some scenarios (e.g., biological control of agricultural pests) require high identification accuracy, while other specific applications have some other requirements. For example, a fine-grained image retrieval system prefers low feature dimension to decrease storage cost; online image identification needs faster processing to boost server concurrency capacity; industrial inspection drone needs an energy-efficient and portable classifica-

tion model. These demands provided new directions for FGVC research in the past few years.

1.1 Background, Motivation and Challenges

There are four main branches of existing fine-grained image classification approaches based on deep learning. According to the role and utilisation of CNN networks and additional information such as part annotation or extra datasets, the approaches can be classified into the following categories.

- The first is to use generic deep neural networks such as CNN to classify the fine-grained images. Researchers are motivated by the impressive performance of CNN [15] to transfer the CNNs pre-trained on the large-scale ImageNet dataset to the datasets from other domains, such as the fine-grained datasets. However, fine-grained image classification needs to obtain more discriminative image representation than the general image classification tasks like the ImageNet. Efforts have been made [6][12] to extend the fine-grained dataset via image augmentation, alignment, or building large transferring datasets. It is the most intuitive way to improve the performance of generic models on fine-grained datasets. Usually, the net structure of such methods is more straightforward and more generalizable compared to parallel or recurrent models. However, the annotation of fine-grained datasets involves a large amount of labor, which is expensive and time-consuming. On the other hand, complex augmentation of image data could slow down the training of a classification network and makes a limited improvement on the classification accuracy.

- The second is part detection and alignment-based approaches. They use deep CNN to localize an object's discriminative parts and conduct classification over the manually defined object parts. The semantic object part localization promotes the fine-grained visual categorization by specifically isolating and zooming in to clarify the subtle differences on specific object parts. Therefore, object parts localization is an essential step for constructing the correspondence between object instances and disregarding the variations caused by the changes of object posing, position, or camera view. Besides accuracy, model interpretability is another highly valued factor in some cases, such as taxonomy. The localization-classification scheme conforms to traditional taxonomic conventions and can be applied to other fields as well, such as image caption,

literal search engine, and discrimination interpretation. The drawback of the part-based method is that they often require "strong" labels of the accurate locations of discriminative semantic object parts, which adds extra workload and makes it only applicable to very small or "one-shot" datasets.

- The third direction of research involves multiple deep neural networks or multiple orders of CNN features to discriminate highly visually-similar fine-grained images. Some previous works applied multiple neural networks to vote on the classification results to improve classification accuracy. Others automatically split the entire dataset into multiple visually similar subsets to improve each subset's professionalism and enhance the prediction. In contrast, others encode the CNN features to higher orders to discover fine-grained discrimination. These duplicated methods show significant enhancement in fine-grained tasks. However, they often suffer from high feature dimension and high computation complexity and are difficult to extend to deep network structures such as ResNet and DenseNet.

- The last branch finds the fine-grained images' discriminative regions using visual attention produced by the deep CNN concerning the most activated regions. They classify by the image's discriminative region and discard the inessential background information. The attention models achieve the state-of-the-art in widely used fine-grained datasets. The drawback of the attention-based models is that they ordinarily use recurrent and multiple CNN structures. Some of them even apply long short-term memory with deep CNN features, making the training and testing computation extremely slow and computation consuming.

These four research directions of fine-grained classification may cross each other to obtain better performance than working solo. The details of the previous works on these four categories will be discussed in Chapter 2.

Our research is motivated by a CSIRO project, "Deep Learning-Based Visual Identification of Large Scale Insect Species". It aims at developing a large-scale digital image library of insects in Australia and discover unknown species. So far about 85,269,426 insect occurrence records and 10,839 datasets have been collected. The project is particularly important to understand the impact of the environment for the protection of natural species. However, the image library contains millions of insect species, with subtle inter-class variances. Another challenge is that the images are shot from both laboratory and natural environment, this could result in apparent intra-class variances. This thesis focuses on the investigation and development

of deep CNN-based models for fine-grained species identification with high classification accuracy. For a largescale online image retrieval system, a low dimension and highly representative feature set is required; for few-shot or one-shot identification cases, part-based fine-grained classification methods with prior knowledge such as part annotation methods have shown great potential in FGVC. For the outdoor service environment, the efficiency of a machine learning model is very important. This thesis proposes several novel approaches to addressing the challenges in FGVC, and presents extensive experimental results to show the potential of the proposed models. For comparison of the performance of the proposed and state-of-the-art methods, the standard FGVC datasets are used in this thesis.

1.2 Main Contributions

A thorough literature review was conducted to gain an understanding of the existing research in FGVC, especially state-of-the-art methods, evaluate different methods, and identify gaps where future research is needed. Methods presented in the literature were compared, evaluated, and summarised into four categories or four research directions.

The main contributions of this thesis include:

- **Proposed a strongly supervised part-based classification method, the Part-Collaboration CNN (PC-CNN).** Similar to previous part-based approaches, the proposed method detects the accurate locations of semantic object parts and extracts multi-view visual features to train the PC-CNN classifier. We designed a detection-localization regression strategy to enhance the robustness of localization against object occlusion and part deformation. To avoid dimension boosting caused by additional part streams, we applied the ResNet structure and fully-convolutional-network to reduce the multi-view feature dimension and implemented an independent part feature extractor without increasing the computation complexity and memory requirement.

- **Investigated the well-performed bilinear CNN model and proposed a novel and compact model, named Squeezed Bilinear Pooling (SBP).** The model can reduce both the feature dimension and computation significantly. With the same dimension of feature sets, the proposed method outperforms the other compressed models (e.g., Compact Bilinear Pooling [25], and Low-Rank Bilinear Pooling [50]) and is two orders of magnitude faster. The integration of

the matrix square root layer is also investigated to enhance the categorization performance. Based on selected discriminative feature channels, we proposed a Fisher-Recurrent-Attention structure to achieve state-of-the-art classification accuracies on the three general FGVC datasets. Based on the selective Squeezed Bilinear Pooling, we designed a novel compressed higher-order feature encoding method, which is more adaptive to general structures such as MobileNet [40], and ShuffleNet [65].

- **Proposed a novel framework named "Forcing Network", referred to F-Net.** There are two modules in F-Net, the feature extracting module and the Forcing module. The forcing module consists of an original branch and a forcing branch. The original branch focuses on discriminative parts and generates class activation maps to locate the most discriminative parts. In the forcing branch, based on the class activation maps, we generate suppressive masks and utilize the suppressive features to classify an abject using the hybrid outputs of both branches. Different from the original framework, F-Net can learn confused parts besides the most discriminative part. After gradient descent and backpropagation, more diverse and enhanced features can be acquired for the fine-grained classification. The Forcing Network can also accommodate the situation for objects whose principal part is occluded. Because of the weak translation invariance of CNN, we crop and rescale objects according to class activation maps and infer again. This inference mechanism can reduce the loss caused by a desperate gamble. In the training phase, we drop the most discriminative regions on the cropped and rescaled images to force the network to pay attention to more regions.

- **Proposed a novel network structure for the fine-grained visual classification on mobile platforms, named Category Attention Transferring CNN (CAT-CNN).** Based on the conception of fine-grained classification with efficient net structures, we introduced the "Category Attention Teacher" with the assumption that a model with better activation to the discriminative regions has better classification ability. Instead of only using the one-hot class label in the training phase, we combined the label loss and the category attention loss so that the student net in the CAT-CNN was capable of learning "what it is" and "where to pay attention" at the same time.

1.3 Research Objectives

The main research objectives of this study are:

- Developing new part-based FGVC methods aiming for high classification accuracy and low computation and memory costs by improving part localization performance, avoiding overfitting from the less discriminative object parts.
- Innovating an efficient bilinear-based FGVC model for mobile platforms by (1) developing a squeezed bilinear pooling module, (2) combining the matrix normalization with compact bilinear methods, and (3) solving the super high-dimension problem of applying the higher-order feature to deep backbones to reduce the computation cost and memory consumption.
- Inventing new attention-based methods to improve the FGVC classification accuracy and reduce the computation cost and memory consumption for the strongly supervised part-based fine-grained object classification.
- Developing the Category Attention Teaching method to transfer knowledge from large-scale deep networks to light-weight models for efficient and accurate FGVC.

1.4 Thesis Outline

This thesis is structured as follows. Chapter 2 reviews available literature works related to fine-grained visual classification and evaluates the performance of different FGVC approaches. Chapter 3 to Chapter 6 present our proposed methods listed in Section 1.2, together with experimental results. In Chapter 7, we conclude the thesis by summarizing the main findings of this research and suggestions and recommendations for future research as the extension of this study. Chapter 8 lists the publications produced during the Ph.D. study.

Chapter 2

Literature Review

This chapter presents the background information and evaluates existing research, especially state-of-the-art methods for fine-grained visual classification (FGVC). The literatures reviewed in this study can be summarised in four main categories:

- **Part-based methods** use semantic object parts to discover more delicate but smaller object features, which often needs extra part annotations to train a part localizer before classifying the object.
- **The traditional methods** utilize deep net structures or apply complex data augmentation in their training phase, which is also performed in general image classification missions.
- **The duplicated model methods** use more than one CNN model in training and prediction or employ higher-order feature encoding to enhance the feature presentation.
- **The attention-based methods** learn the most discriminative regions in the image and focus on them while ignoring background noise. They often need multiple inferences in both training and testing phases.

The literature review revealed that the four categories' roadmaps often cross over each other to achieve better performance. In this chapter, we review these methods in detail. Section 2.1 introduces several traditional methods; the part-based methods are reviewed in Section 2.2; methods using multiple CNN models or higher-order image features are introduced in Section 2.3. Section 2.4 describes the recurrent attention-based approaches. Some illustrations and tables in this chapter are duplicated from the original papers to make the description more transparent. The citations are annotated in the captions of each duplicated illustration or table.

2.1 Traditional Methods

CNN has become the most popular computer vision model because of its superiority in object detection and image classification. Since LeCun et al. [53] first introduced the convolutional layer's backpropagation method and proposed the convolutional net structure, the CNN models have been surpassing other computer vision learning-based approaches, e.g., Support Vector Machine (SVM) [83] and Random Decision Forest (RDF) [39]. Large-scale image datasets with category-level annotation, e.g., the ImageNet [15], has accelerated the development of CNN-based computer vision approaches. In recent years, a tremendous number of researches have verified the superior performance and effectiveness of CNN structures in large-scale visual categorization. On the other hand, motivated by the impressive discrimination performance of CNN [15]] on the large-scale image datasets, researchers transferred the visual knowledge from the large image datasets (usually ImageNet [15]) to other more specific but comparatively much smaller domains, e.g., the fine-grained or medical image datasets. Usually, a model was pretrained using the large-scale dataset, then adopted to the smaller dataset and fine-tuned to obtain the final model. The ability of CNN models to produce the image's discriminative features is also crucial for fine-grained visual categorization. By replacing the last several fully connected layer, we can apply most of the current state-of-the-art CNN structures to FGVC datasets.

This section presents fine-grained visual classification using traditional CNN-based methods, including transfer learning and advanced image data augmentation. Some basic and commonly used CNN structures are also introduced in this section.

2.1.1 General CNN Backbones

AlexNet [52] is the first winner of the ILSVRC2012 competition based on a deep convolutional neural network. It achieved a superior top-5 error rate of 15.3% in the testing phase, which is 10.9% higher than the second-best entry (26.2%). As shown in Figure 2.1, AlexNet architecture consists of five convolutional layers and three fully connected layers, and all of the eight layers are end-to-end learnable.

The GoogLeNet [84] is the new state-of-the-art for detection and classification in the ImageNet Large-Sale Visual Recognition Challenge 2014 (ILSVRC14). The major hallmark of the GoogLeNet architecture is the improved utilization efficiency of the computation resources of the network. This is obtained by the carefully crafted architecture, named "inception module", enhancing the network in depth and width

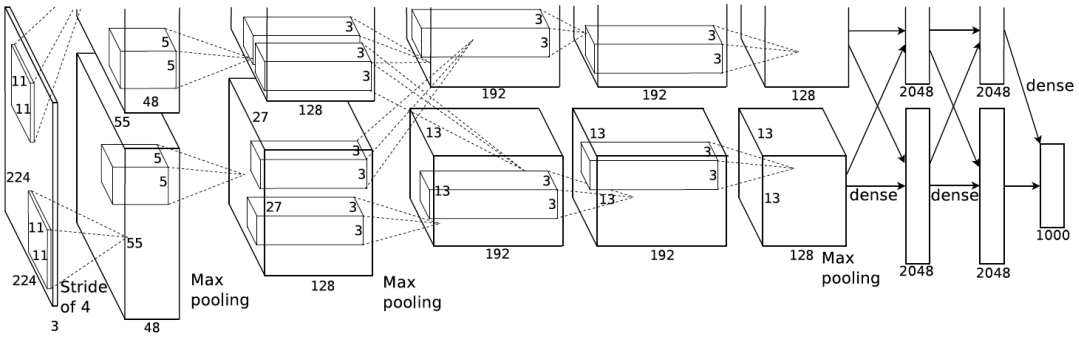


Figure 2.1: Framework of AlexNet [52].

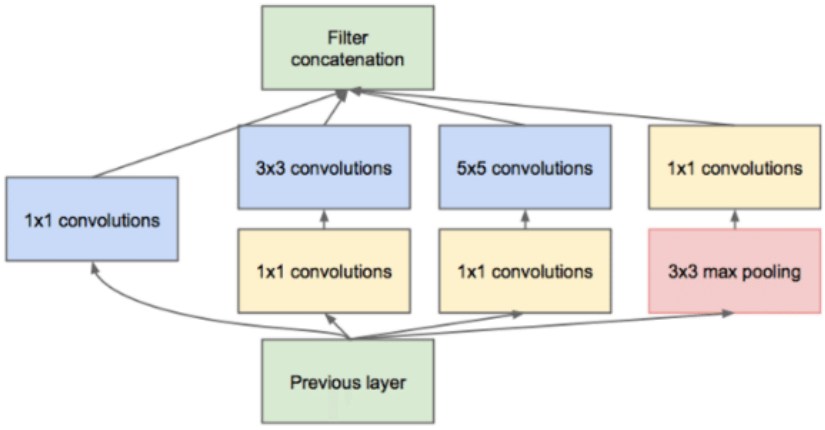


Figure 2.2: Inception module of GoogLeNet [84].

while keeping the computational consumption acceptable. GoogLeNet consists of 27 layers (including five pooling layers). It stacks the inception modules to reduce the number of parameters to 1/12 of that in the structure of AlexNet [52]. Figure 2.2 explains the workflow of an inception module. The 1×1 convolutional layers are applied to perform dimension reductions before the more computationally expensive 3×3 and 5×5 convolutional layers. Moreover, The inception module also consists of the rectified linear activation function.

The VGG net [80], as its name claims, increases the convolutional neural networks' depth by using filters with the smallest kernel size (3×3) to reduce the receptive field. The architecture of the VGG-16 is displayed in Figure 2.3. It contains 13 convolutional layers, followed by the classifier with three fully connected layers. VGG-19 consists of 16 convolutional layers and has a better representation ability than the shallower VGG-16. Non-linear rectification was applied on all hidden layers.

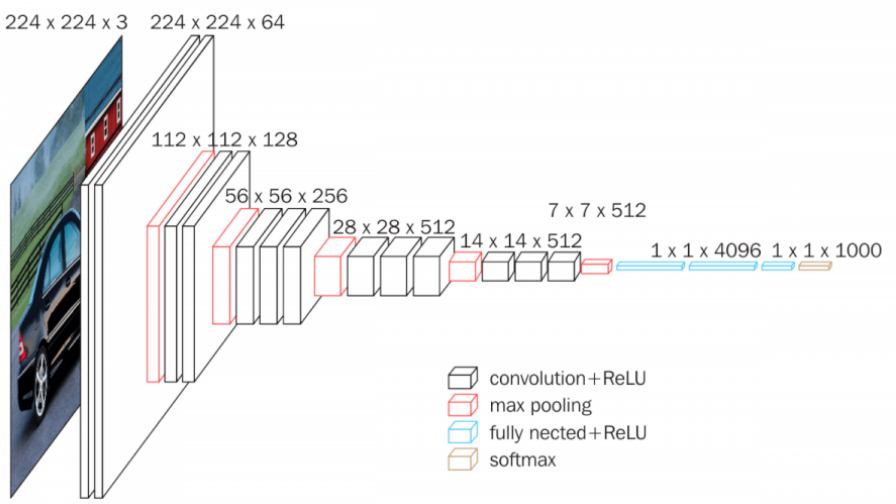


Figure 2.3: The architecture of VGG-16 [80].

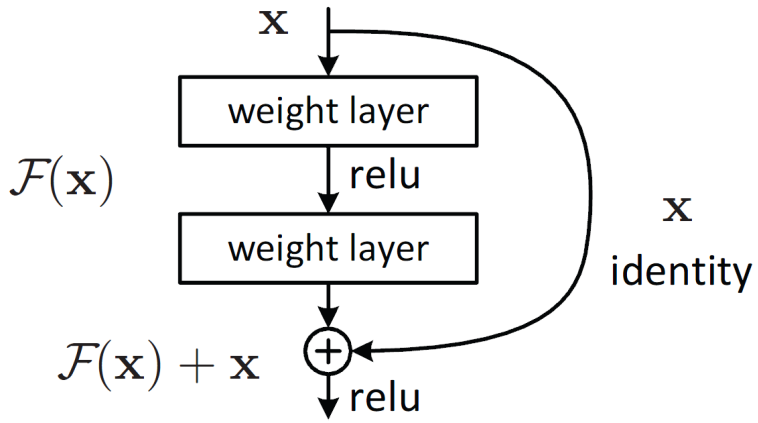


Figure 2.4: A residual block.

The VGG architecture is the winner of the ILSVRC localization and classification competition in 2014 and can be transferred to other tasks easily.

Following the idea to deal with the vanishing gradient issue in VGG [80], the ResNet introduces a so-called "identity shortcut connection" that skips one or more layers, as demonstrated in Figure 2.4. It argues that stacking layers shouldn't degrade the network performance because we could stack identity mappings (layers that do nothing) upon the current network. The resulting architecture would perform the same. This indicates that the deeper model should not produce a training error higher than its shallower counterpart. They hypothesize that it is easier for the stacked layers to fit a residual mapping than to fit the underlying mapping.

EfficientNet [85] is a convolutional neural network architecture and a scaling method that uniformly scales all dimensions of depth/width/resolution using a com-

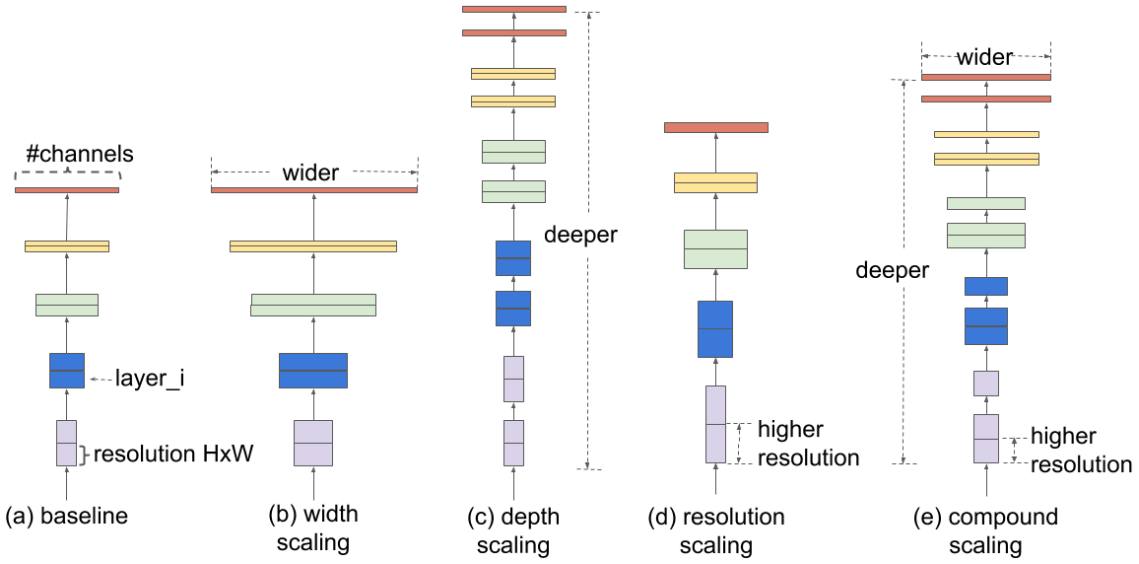
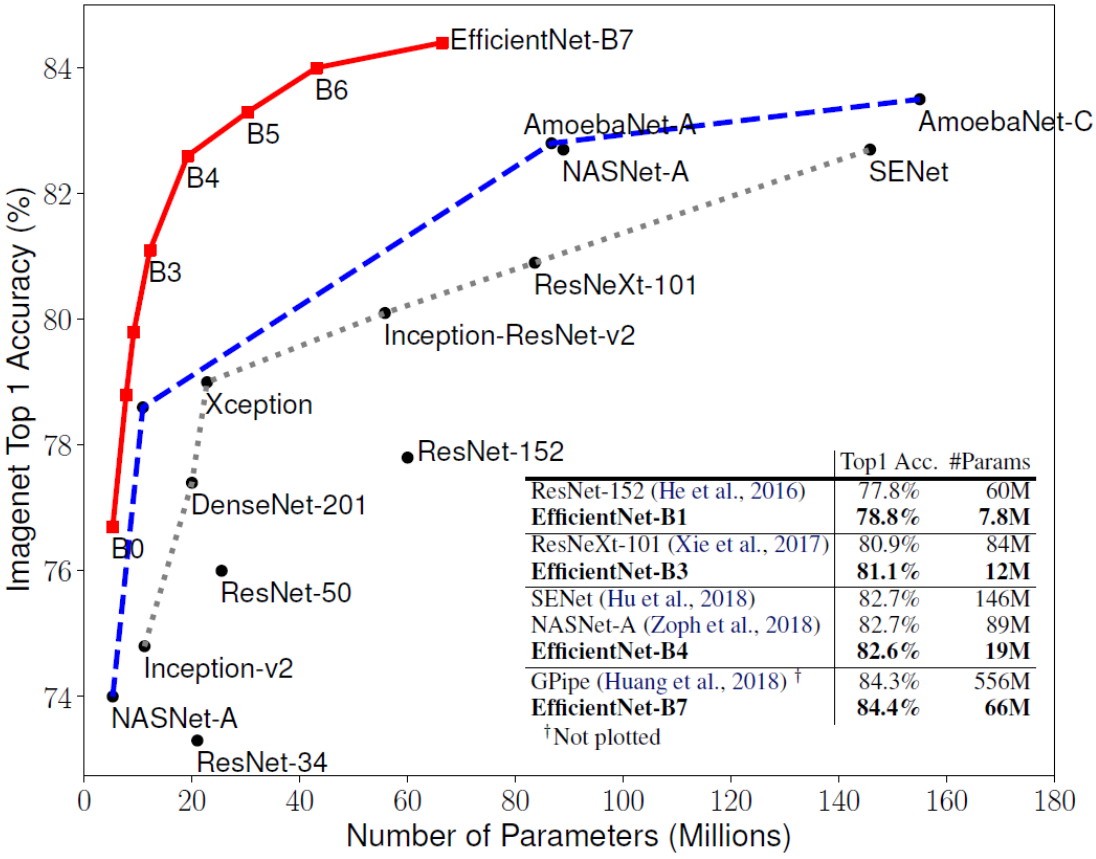


Figure 2.5: Model scaling in [85]. The proposed componding scaling (e) combines the three scaling (b,c,d) into a single strategy.

pound coefficient. Unlike conventional practice that arbitrarily scales these factors, the EfficientNet scaling method uniformly scales a network in width, depth, and resolution with a set of fixed scaling coefficients. For example, suppose we want to use 2^N times more computational resources. In that case, we can increase the network depth by α^N, width by β^N, and image size by $gamma^N$, where α, β, γ are constant coefficients determined by a small grid search on the original small model. EfficientNet uses a compound coefficient to uniformly scale network width, depth, and resolution in a principled way. As presented in Figure 2.5, the compound scaling method is justified by the intuition that if the input image is of more resolution, then the network needs more layers to increase the receptive field and more channels to capture more fine-grained patterns on an image with higher resolution.

Figure 2.6 presents the comparison among the EfficientNets, and other benchmarks with the dataset of ImageNet [15]. To the best of our knowledge, the EfficientNets outperform all the other generic image classification structures in both accuracy and efficiency.

There are some other generic deep convolutional feature extractors for image classification [78][97][99]. The CNN structures can be easily transferred from large-scale image datasets to fine-grained image datasets. The last fully connected layer is randomly reset. The output number of the layer is set to the same as the class number of the objective image dataset (for instance, 200 in the CUB-Bird-2011 dataset). The results of transferring directly from the ImageNet dataset to CUB-200-2011 using the



	Top1 Acc.	#Params
ResNet-152 (He et al., 2016)	77.8%	60M
EfficientNet-B1	**78.8%**	**7.8M**
ResNeXt-101 (Xie et al., 2017)	80.9%	84M
EfficientNet-B3	**81.1%**	**12M**
SENet (Hu et al., 2018)	82.7%	146M
NASNet-A (Zoph et al., 2018)	82.7%	89M
EfficientNet-B4	**82.6%**	**19M**
GPipe (Huang et al., 2018) †	84.3%	556M
EfficientNet-B7	**84.4%**	**66M**

†Not plotted

Figure 2.6: The prediction accuracy and model size comparison with multiple models on the ImageNet dataset [85].

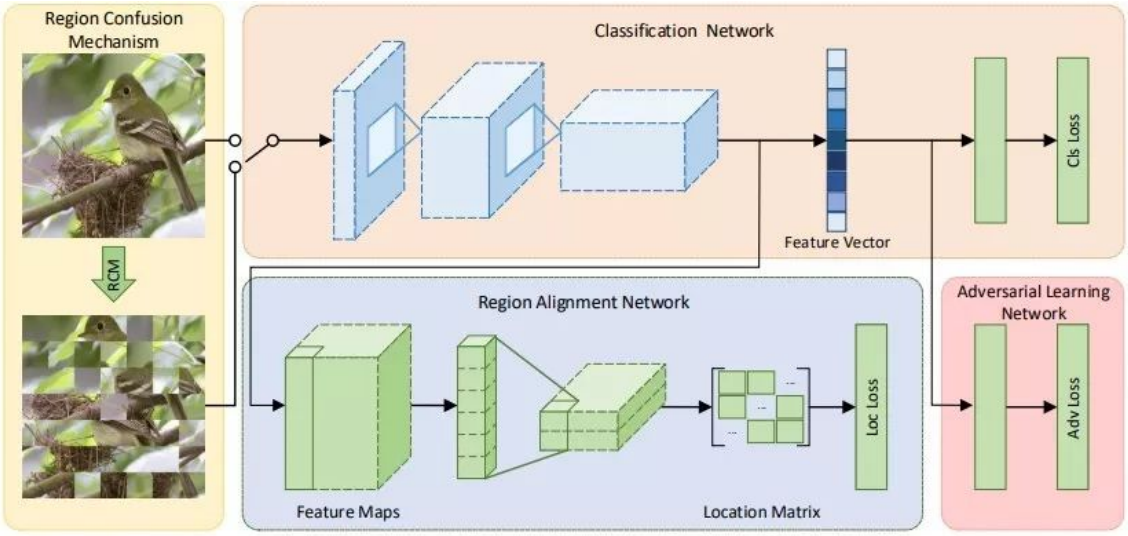


Figure 2.7: The general framework of the Destruction and Construction Learning [10].

previous CNN structures are shown in Section 2.6. The classification results have proved the effectiveness of the CNNs pre-trained from the generic image dataset.

2.1.2 Destruction and Construction Learning (DCL)

The DCL[10] proposed a novel image classification structure named Destruction and Construction Learning to enhance the "expert learning" of the CNN networks. It introduced two modules except for the typical image classification network. Firstly, the Destruction Module uses the Region Confusion Mechanism (RCM) to break the image construction and shuffle the subtle regions randomly. Thus, the DCL can force the classification network to learn the local features without object structures. Secondly, to avoid the negative effect of presentation noise from the RCM, Adversarial Learning Network (ALN) was used to distinguish whether an image is an original image or a destructed image. Region Alignment Network used the destructed image features to reconstruct the original image distribution. The framework of DCL is shown in Figure 2.7

By collecting the most discriminate object regions and learn from them, DCL outperforms other methods on a wide range of FGVC datasets. It is worth noting that the DCL can also significantly improve the network performance on general image classification datasets.

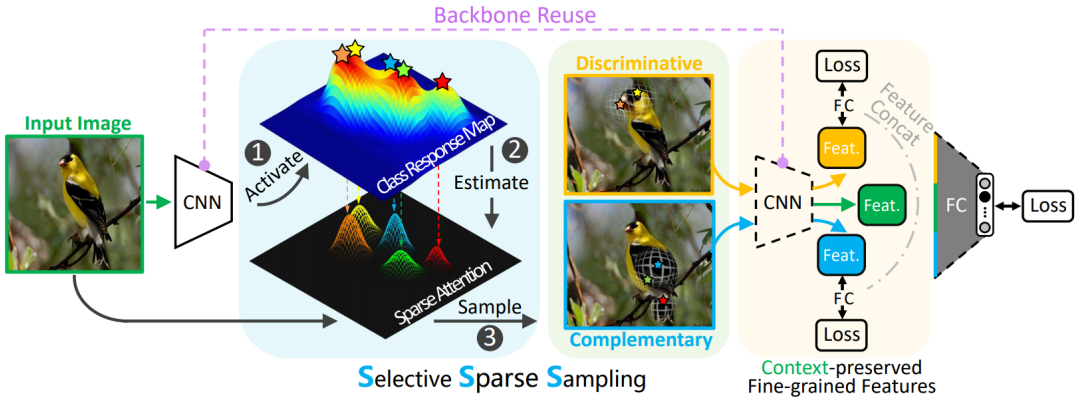


Figure 2.8: The general framework of Selective Sparse Sampling Net [16].

2.1.3 Selective Sparse Sampling Net (S3N)

A Selective Sparse Sampling framework was proposed [16] to solve the problem of keeping the image construction while highlighting the discriminative regions, as illustrated in Figure 2.8. The system mimics the human visual system to predict dynamic sparse attention regions in the image content. S3N is trained under image-level supervision, firstly collecting the peak responses to each class. Then, each class's size of the peak responses is estimated to generate a set of sparse attention maps. The image is inhomogeneously transformed to highlight the corresponding regions and fed again to the network to learn discriminant features and complementary features.

2.1.4 Anti-Perturbation Inference Net (API-Net)

Aiming to solve the generative classification problem, an Anti-Perturbation Inference (API) approach was proposed [18]. The API searches for anti-perturbations to maximize the lower bound of the joint log-likelihood of inputs and classes. By leveraging the lower bound to approximate Bayes' rule, it constructed a generative classifier called Anti-Perturbation Inference Net (API-Net) upon a single discriminator, as demonstrated in Figure 2.9. It benefits from the generative properties to tackle the off-manifold examples while maintaining a concise structure for effective optimization. This introduces a similar idea from another hot topic, the Generative Adversarial Network (GAN), and its performance is competitive to approaches in other traditional routes.

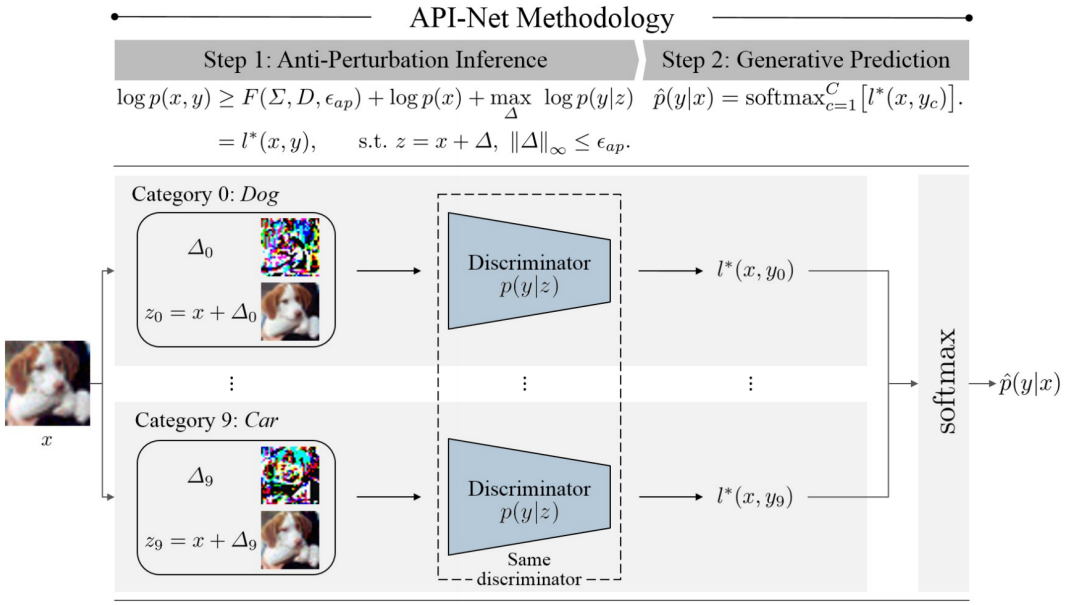


Figure 2.9: Overview of the inference procedure of Anti-Perturbation Inference Net [18].

2.1.5 Mutual-Channel Loss (MC-Loss)

Unlike most FGVC approaches, a novel loss function was proposed [9] aiming at delving into individual feature channels and ensuring channel presentation diversity. As Figure 2.10 shows, the net structure needs no more than an extra component named MC-Loss. The proposed Mutual-Channel Loss consists of four parts: (a) Softmax for classification; (b) Cross-Channel Max Pooling which concentrates the peak activation of every attention channel onto a single activation map; (c) Summarisation of the pixels in each channel into a single value; and (d) Calculation of the average throughout all the channels. The workflow and outputs of MCL are illustrated in Figure 2.11.

The Mutual-Channel Loss uses a novel loss function as extra supervision to force the model to learn diverse discriminative regions on an object and significantly improve the performance of some basic models on FGVC datasets.

2.1.6 Attribute Mix+

It is believed that the significant challenge of FGVC is the limitation of the dataset scale. The FGVC data are very difficult to obtain and expensive to annotate. Addressing the problem of overfitting caused by a small-scale dataset, Attribute Mix+

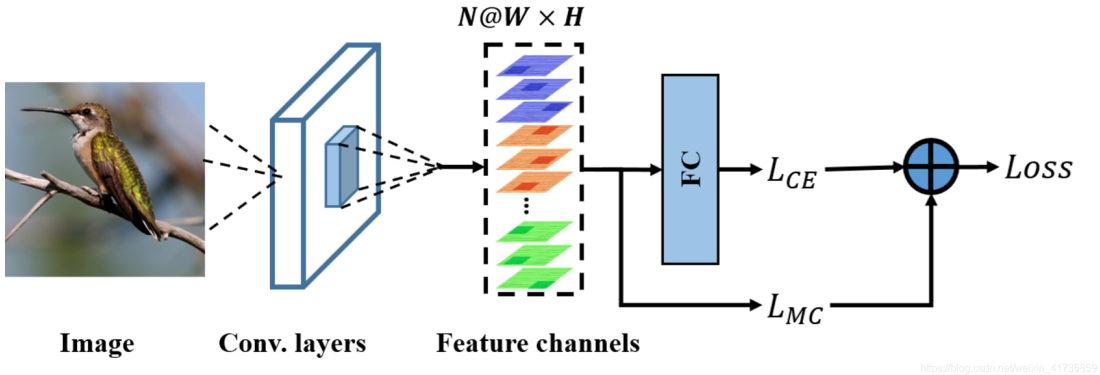


Figure 2.10: MC-Loss layer in a CNN structure [9].

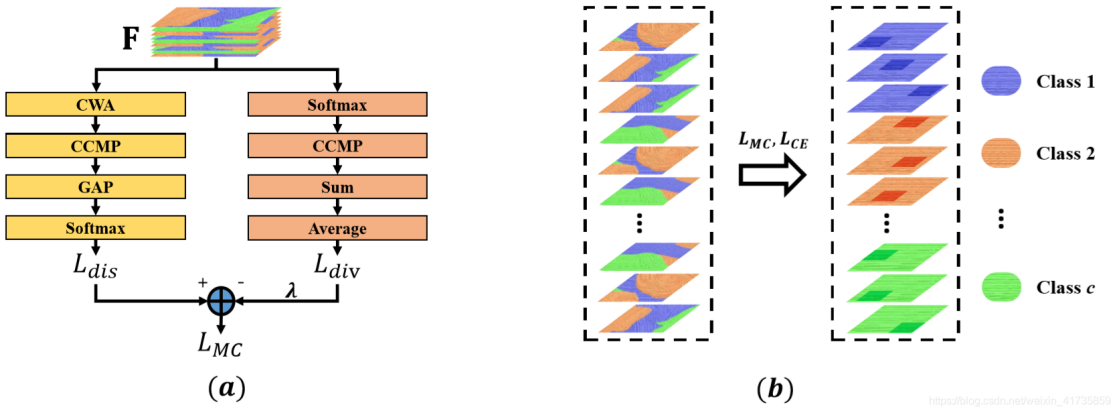


Figure 2.11: The inference structure of the MCL [9]. (a) is the MCL components and (b) is the output feature from CNN with/without MCL.

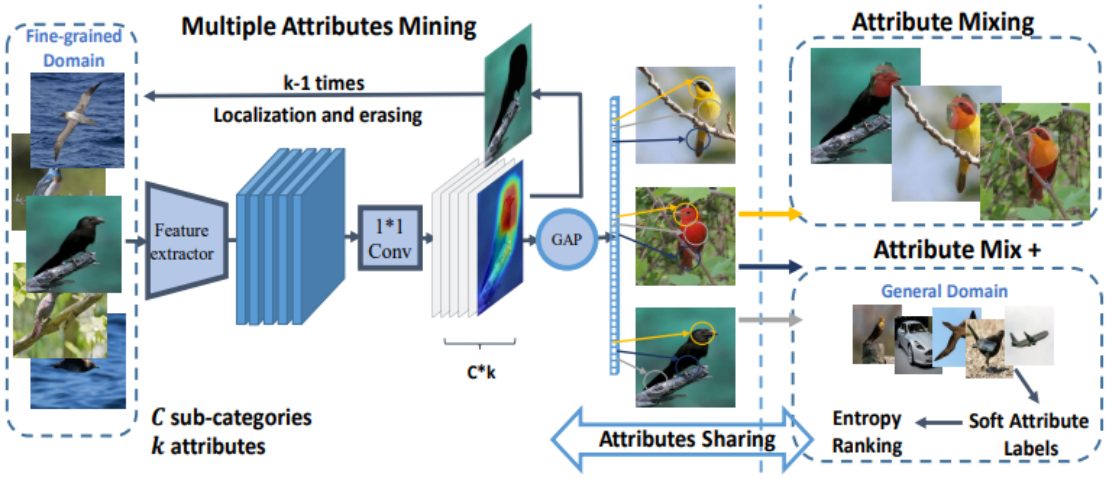


Figure 2.12: The workflow of Mix+ model [54].

[54] was proposed as a method for data auto augmentation/generation for FGVC applications.

The first conception of this paper [54] is the Attribute Mix, a data augmentation strategy. As illustrated in Figure 2.12, after the category attention maps learned each class's discriminative attributes, the Attribute Mix module mixed the attributes to generate new image samples. The Attribute Mix+ module mines the attributes that belong to the same category and inserts the eligible image samples into the dataset to enhance the training.

2.2 Part Based Methods

The fine-grained visual categorization can be enhanced by semantic part localization, for that it can help isolate subtle appearance differences correlated with particular object parts explicitly. In humans' procedure to recognize a fine-grained object class, object parts localization is necessary to degrade the impact of object pose variations and camera view position variations and establish the correspondence between object instances in different image samples. Figure 2.13 presents the workflow of many traditional part-based approaches. The object parts on fine-grained datasets (e.g., beak and head in CUB-200-2011) are localized in each image in the first step. Then the part alignment is conducted to normalize the part distribution. The last step is to extract the aligned parts' feature to do classification.

Following this roadmap, Thomas et al. [4] use data mining to extract intermediate image features. Each of the learned intermediate features is specialized to discrim-

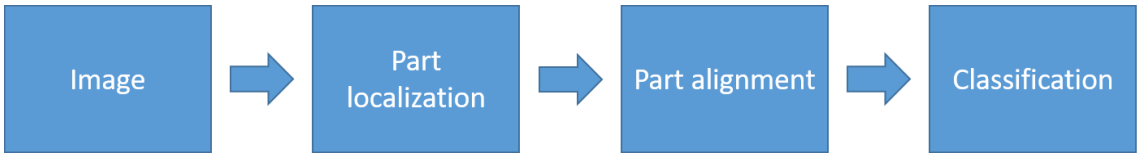


Figure 2.13: Rough framework of traditional part detection-based methods.

inate two particular categories on a particular object part. To find the accurate location of object parts like the ears and eyes of a dog, Liu et al. [63] built a geometric model of dog breeds and their face components. To discover the object parts' common geometric patterns, Yang et al. [101] learned the co-occurrence statistics of the geometric patterns of the fine-grained objects using a template model. For fine-grained visual classification, Yang et al. [101] extracted the object features in the images aligned with the cooccurred patterns. Similarly, Chai et al. [7], and Gavves et al. [26] segmented the images and conducted unsupervised alignment on the image fragments. Then they use the alignments to transfer the part patterns from the training set to the testing set. In the final step, the part features are extracted from the testing images for classification.

Besides the works, the following subsections introduce several more recent part-based works using deep-learning technology.

2.2.1 Part-Based R-CNN (PBR-CNN)

Previous research has shown that deep convolutional neural networks can significantly enhance part localization performance. The Part-based R-CNN [108] proposed a structure to conduct object part localization using deep convolutional features from the bottom-up semantic region proposals. The framework of the PBR-CNN is presented in Figure 2.14. It developed the object detection structure, RCNN [31] to conduct object detection and part localize at the same time using a geometric prior. It began with finding several proposal regions of interest with the selective search technology [86]. Then the object detector and the part detector are trained respectively using the deep convolutional neural network. The detectors estimated all proposals' confidence in the testing phase and applied geometric constraints to fine-tune the detection proposals' confidence score and confirm the optimal proposals for the detections of the object and its parts. The next stage is to use the extracted features of the localized object and its semantic parts to train a deep CNN model to classify the image in a pose-normalized representation.

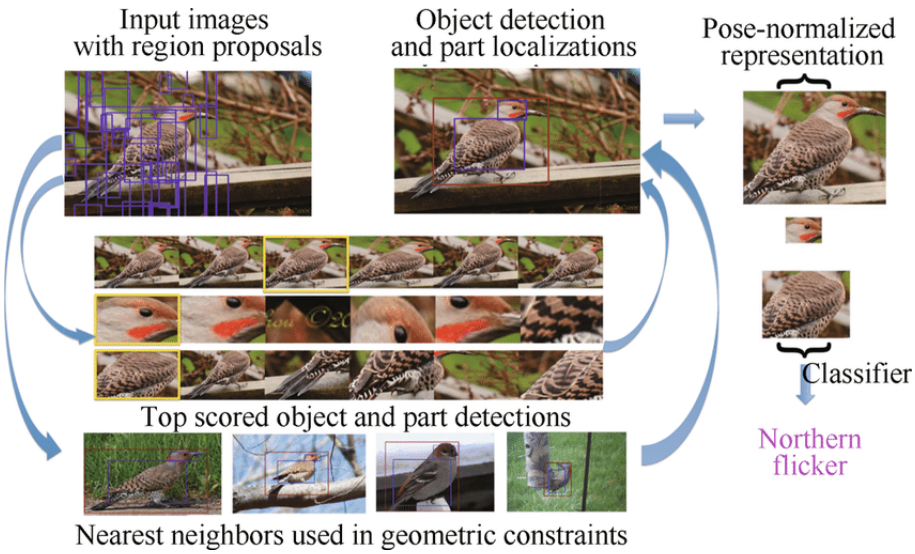


Figure 2.14: Workflow of the Part-based R-CNN [108].

The PBR-CNN model used the ground truth bounding boxes from the original CUB-200-2011 dataset to fine-tune the CNN pretrained on the ImageNet dataset to the 200-way bird classification task. They claimed that the transfer fine-tuning enhances the deep CNN features for the fine-grained bird classification. In practice, they replaced the original 1000-way fully connected layer in AlexNet with a randomly initiated 200-way fully connected layer as the classifier. On the other hand, they use the bounding box annotations of the object and its semantic parts that indicate the bounding boxes of the head and the body to train the detectors. The training is based on a one-vs.-all linear SVM, which takes the region proposals' deep convolutional feature as input and the label of whether the region belongs to an entire object or a particular object part as output. A single part detector is not robust enough, so the window with the best confidence from a single part detector is not mean to be perfectly accurate. The PBR-CNN utilized a geometric constraint to filter out the incorrect detection by considering the relative location of the detected object and its semantic parts.

In the testing phase, the detectors score all the region proposals produced by the selective search. The geometric non-parametric constraints are employed to the detected windows and their scores to find the optimal object and part detections. Finally, an AlexNet pretrained on ImageNet was applied to fine-tune for the FGVC dataset. Object feature and part features are extracted using the deep network and concatenated to form the final feature representation, which is used to train a one-vs.-all linear SVM to produce the category's final prediction.

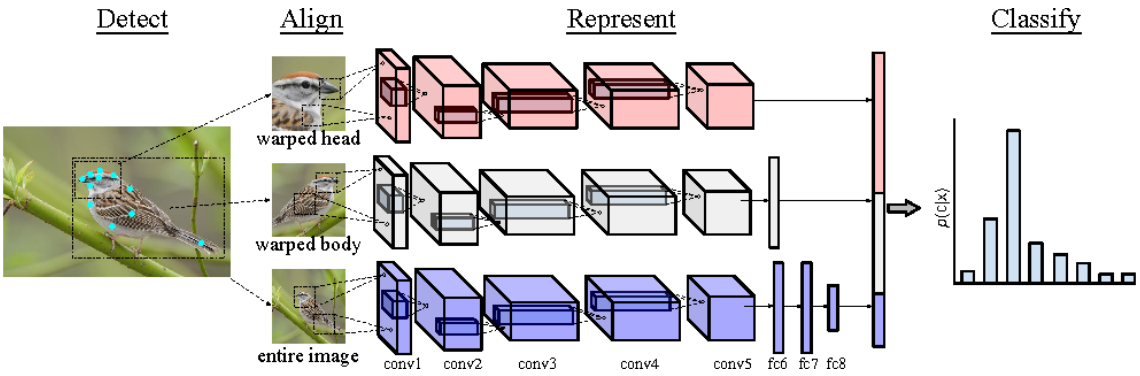


Figure 2.15: Pose normalized nets [6].

2.2.2 Pose Normalized Nets

The core idea of the pose normalized net [6] is to estimate the object pose and use it to extract the object parts' local features. Deep convolutional neural networks are applied to the image patches localized and normalized by the pose estimation to extract the local features and conduct the final classification. Figure 2.15 demonstrates the workflow of the pose normalized net. The lower level of the CNN features (conv5 and fc6) are used to pose normalized zooming-in feature extraction, while the higher level of the CNN features (fc8) are used to extract the entire images' feature. The features are concatenated to produce the final image feature for classification.

Pose normalized net performs the Deformable Part Model (DPM) [5] to localize the key points and predict their visibilities in the training phase. The entire object and its parts are cropped and fed to independent deep CNNs to extract the image features. Then the concatenated image features are used to train the classifier (one fully connected layer). The paper also implemented a prototype using the bounding boxes and part annotations provided by the dataset directly.

In the testing phase, the keypoints detected by the DPM are used to crop the regions of the test images that are aligned with the learned object models. The image region patches are used to extract local features using the deep CNNs [52]. The experiments show that it performs better when extracting the pose normalized features using lower-level representation and unaligned image features using lower-level representation.

2.2.3 Multi-Proposal Consensus (PL-MPC)

The Multi-Proposal Consensus [79] uses a single convolutional neural network, the AlexNet [52] to localize the keypoint and region of the object parts (eyes, tail, wings).

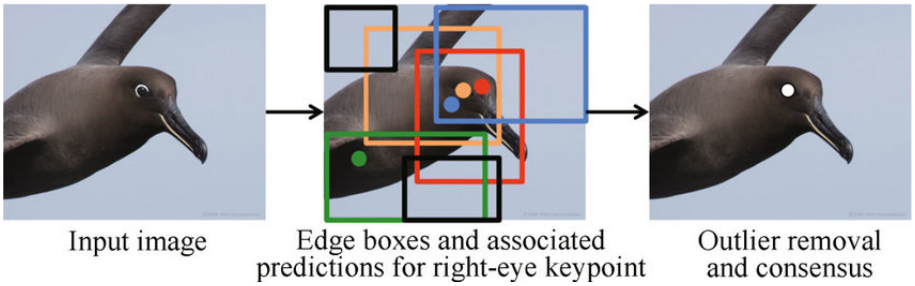


Figure 2.16: The workflow of the multi-proposal consensus [79].

It is more advanced than the PBR-CNN, which performed the manually designed geometric constraint to predict the object parts' locations.

The Multi-Proposal Consensus applied the regression-based modification on the last fully connected layer of the AlexNet to localize all the keypoints and predict their visibilities simultaneously. Two separate output fully connected layers are adopted to replace the original fc8 layer of the AlexNet. The two layers are for keypoint localization and their visibility, respectively. The ImageNet [15] dataset is used to pre-train the AlexNet model. The model then learns on edge box crops [113] that can be directly obtained from the CUB-200-2011 dataset. After the keypoint location and their visibility scores are obtained, the detections with low visibility confidences are ignored. The rest of the detections display a normal distribution around the ground truth of keypoint locations. The medoid is considered a robust estimation of the detection.

Figure 2.16 illustrates a typical procedure to find the key point of the right eye. Confidence thresholding is performed to determine the optimal localization of the right eye. The predictions with a confidence lower than the pre-set threshold are shown in boxes with black edges without associated dots. The green box stands for an outlier with high confidence.

Three partial regions of each bird are localized using the keypoints predicted from the network: the whole body, head, and torso. The tightest bounding box that contains the forehead, nape, eyes, beak, crown, and throat is defined as the head region. Similarly, the box containing the tail, breast, throat, legs, back, wings, and belly is defined as the torso. The whole body region is provided in the annotation of the CUB-200-2011 dataset. The whole body, head, and torso regions are fed into CNN to produce each part's representation features. Then the features are concatenated to form the final feature for the 200-way linear one-vs-all Support Vector Machine to perform classification.

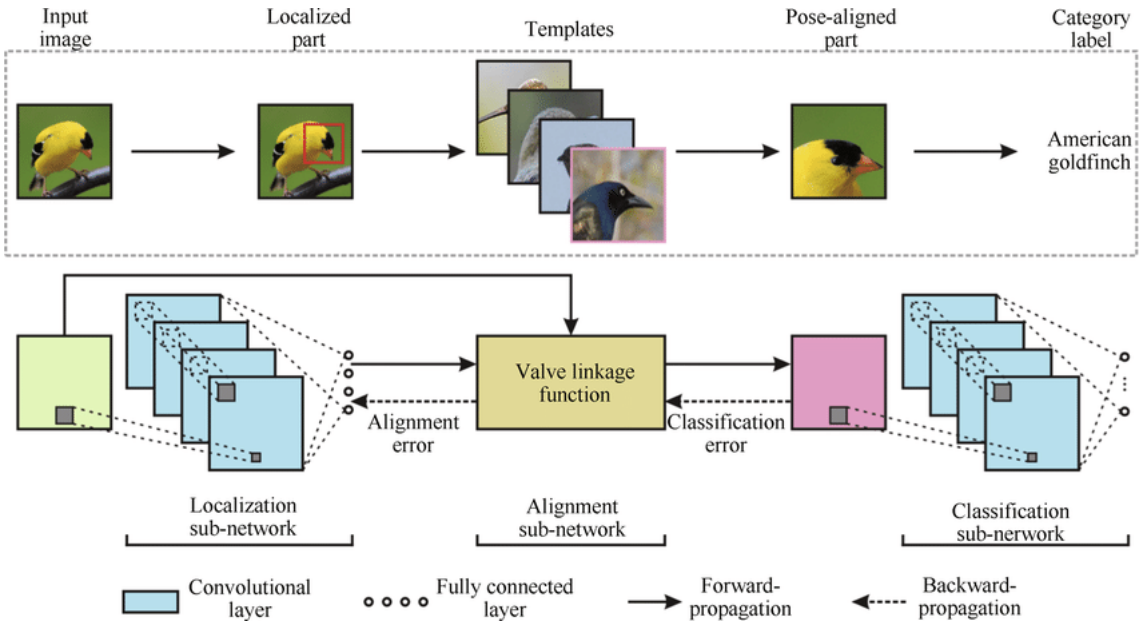


Figure 2.17: Framework of deep Localisation, Alignment and Classification [59].

2.2.4 Deep Localisation, Alignment, and Classification (Deep LAC)

Deep Localisation, Alignment, and Classification (Deep LAC) [59] uses a single deep CNN to perform part localization, alignment, and classification simultaneously. The Valve Linkage Function (VLF) is designed to backpropagate the localization, alignment, and classification losses in a single system. When the LAC model is trained, the classification error and the alignment error are adaptively compromised to form the final error and help localization. Figure 2.17 demonstrates the framework of the Deep LAC.

Following the AlexNet, the part localization stream contains five convolutional feature extraction layers and three fully connected layers for classification and alignment. The part stream's input and output are the images for fine-grained classification. The coordinates normalized to range from 0 to 1 using the relative position to the top-left corner of the view, respectively. It regresses the part bounding boxes with the ground truth provided by the dataset to learn the object parts' distribution in the training phase. In the testing phase, LAC takes the images as input and outputs the corresponding object parts' normalized relative coordinates.

The part locations detected by the localization stream are fed into the alignment stream, which conducts align the input image with a template following the approach in [74], and output the aligned part image segments to the classifier. For higher

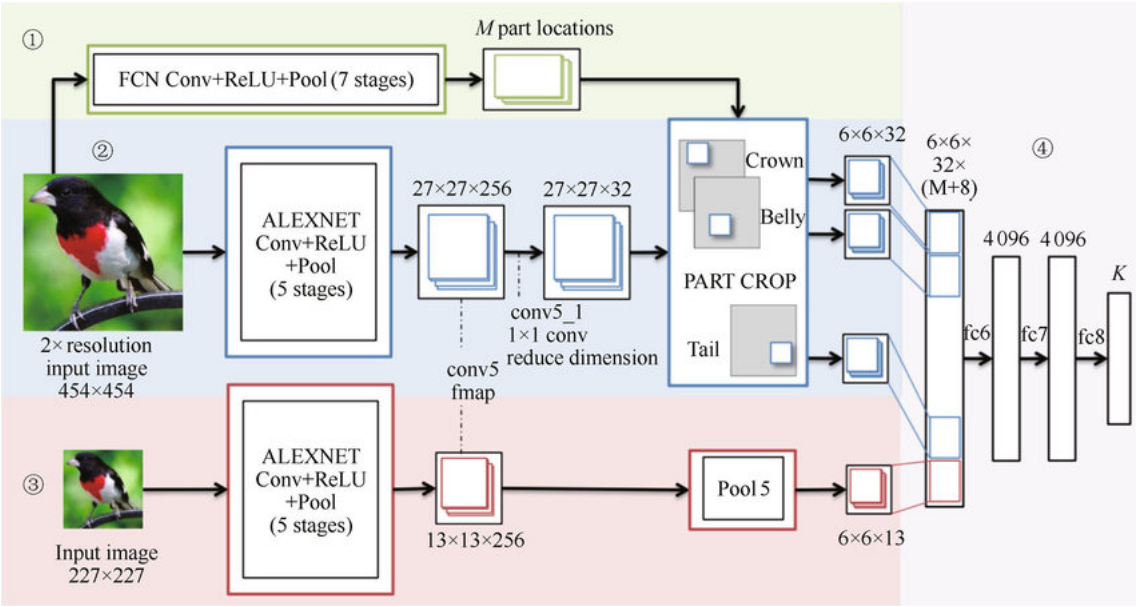


Figure 2.18: Part-stack CNN [45].

prediction accuracy in the classification phase, the alignment stream performs offsets translation, scaling, and rotation on the input part images. Besides the function of pose alignment, this stream also bridges the entire LAC's backpropagation by using the alignment and classification result to fine-tune the localization stream.

The VLF plays a critical role in the deep LAC architecture because it provides the optimal connections between the localization and classification streams. All three subnetworks are connected in a compromising way: the LAC model can take the classification accuracy and alignment errors into a combined consideration to confirm the training data's effectiveness. The classification can have better confidence if the alignment is sound enough. Otherwise, if the alignment is not correct, the classifier's loss can have less impact on the training of the entire network.

2.2.5 Part-Stack CNN (PS-CNN)

The workflow of the Part-stacked CNN (PS-CNN) [45] is explained in Figure 2.18. PS-CNN introduces a part detection model that contains a fully convolutional network to localize the major object and its multiple semantic parts. Base on the part detection, PS-CNN encodes the object-level and part-level features together with a two-stream feature extraction network. Following the previous research works, it also used strong supervision from the manually-labeled part annotation of the fine-grained datasets like CUB-200-2011.

The fully connected layers in generic CNN structures are replaced with an 1×1 convolutional layer to construct a Fully Convolutional Network (FCN). The FCN takes an RGB image as input and outputs a set of activation maps with lower resolution than the input image. The activation of each pixel in the output maps is only related to the visual representation of its receptive field, a region with a corresponding location, and a fixed size in the input image. The localization net chose FCN mainly for three reasons: (1). Compared with fully connected layers generating the coordinates of the detected keypoints, FCN produces activation maps indicating not only the part location but also the size of a particular object part, which can be used in the classification networks directly. (2). FCN can learn to localize the objects' different parts in a united framework, considering the relative position relation between the semantic parts in an object. (3).FCN is more efficient to fully connected layers in both the training and the inference phase.

The localization net based on a fully convolutional network [68], takes the object images as input and the dense feature maps of the correspondent object parts as output. The annotation of the K key points at the center of each semantic object part is used to generate the CNN's activation to the image. A Gaussian filter is applied to the generated feature maps to remove the noise. Finally, the output of the localization net is K conv5 feature maps indicating K 2D locations of the object parts in the image. The coordinates of the locations are extracted using the indexes of the maximum responses in each feature map. The part locations are used to crop the image regions covering the entire object and the discriminative object parts. Each of the image segments is fed into the classification net.

The PS-CNN classification net is two-fold: the object-level classification using the bounding boxes of the objects and the part-level classification using the part landmarks. The part-level classification is implemented with a shared feature extraction structure, which contains the first five convolutional layers of the AlexNet. A bottle-neck convolutional layer has a lower output channel number to reduce the memory cost and computation complexity. The object-level classifier is a CNN net structure that takes the bounding box level image as input and the conv5 feature maps as output. The final classification is conducted using the grouped features concatenated from the object feature and the part features. Following the architecture of the AlexNet, the classification net contains three fully connected layers. Experiments have been conducted on both original part annotation and the part locations predicted by the localization network.

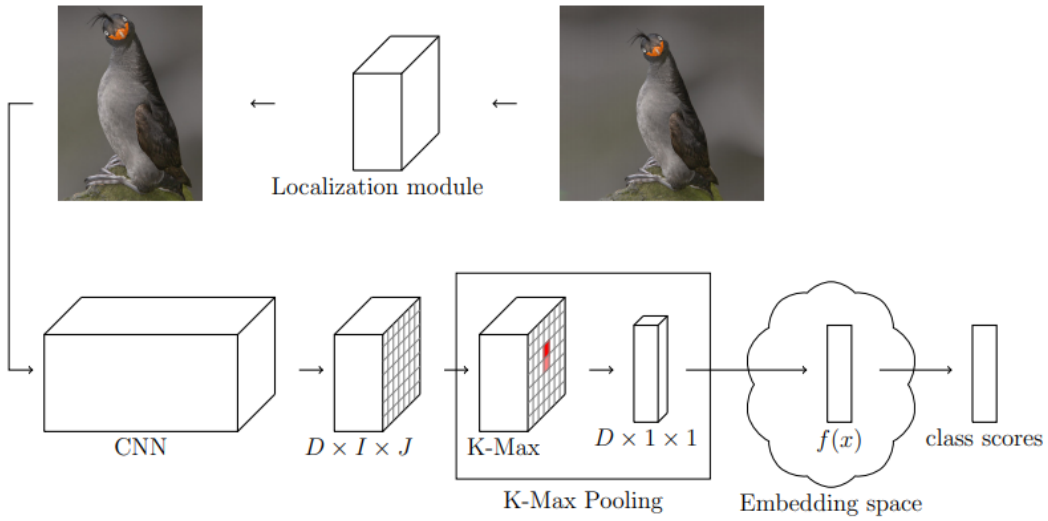


Figure 2.19: The workflow of AttNet&AffNet [35].

2.2.6 AttNet&AffNet

A specific backbone CNN structure can be enhanced with a limited increase in memory and computational complexity [35]. It only needs a single inference during training and testing. As manifested in Figure 2.19, for most of the CNN structure, the last convolutional layer has a large resolution (e.g., ResNet-50 with 448×448 input has 14×14 output resolution). Instead of using global average pooling, global k-max pooling is used [35] to approximate the part-based recognition. Moreover, an efficient bounding box detector is trained that can be applied before the image is processed by the backbone CNN. The localization module is lightweight and trained using the class labels only.

A hybrid model was proposed [35] to integrate the auto-part-localization and the attention-based approaches. We classify this work as a part-based method because its innovation is mainly on the localization module.

2.3 Model Duplication and Feature Encoding Methods

Another popularly applied approach in many CNN-based fine-grained visual categorization researches is to split the whole FGVC dataset into several subsets, and each of the subsets has a similar appearance representation. In this way, the profession-

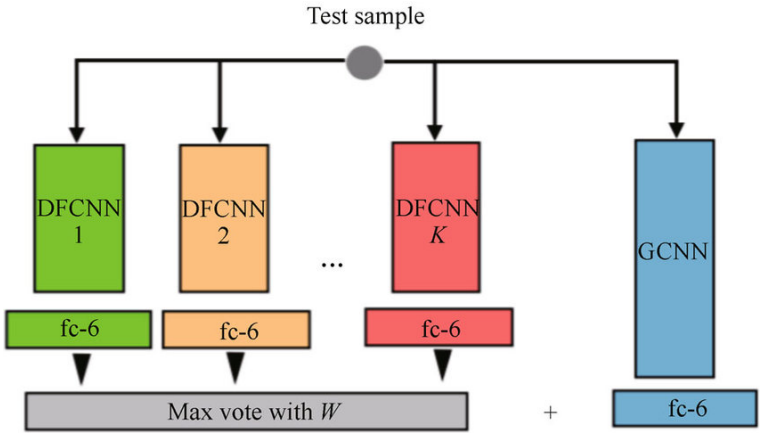


Figure 2.20: Framework of subset feature learning networks [30].

alism of the classifiers can be improved since they only focus on smaller but more delicate classification problems. On the other aspect, "more hands make light work". Using multiple convolutional neural networks was also proven to improve fine-grained visual classification performance. Inspired by the duplicated model approaches, some other researchers used the higher-order image feature encoding to extend the features learned by CNNs to a higher dimension and improved their representation abilities. The three routes can be concluded with the model duplication and feature encoding methods. The following subsections describe the methods using these ideas.

2.3.1 Subset Feature Learning Networks

The subset feature learning networks [30] contains two major streams. The first is the Generic Convolution Neural Network (GCNN). The second stream consists of several Domain Feature Convolutional Neural Networks (DFCNNs). The AlexNet, adopted as the backbone of the GCNN, is firstly pre-trained on a large-scale image dataset that contains a similar domain as the target fine-grained image dataset. Then the target dataset is used to fine-tune the GCNN model. The output of the first fully connected layer, fc6, of the GCNN model is used as the representation feature. Linear Discriminant Analysis (LDA) is applied to the CNN feature to reduce the dimension. The target dataset classes are clustered into several subsets by their visual similarities to train the subset classifiers in the second stage. Figure 2.20 presents the framework of the subset feature learning networks.

In the second stage, correspondingly, K separate domain feature CNNs are learned from the K subsets pre-clustered. The DFCNNs are supposed to learn subsets' features and can discriminate the images in the same subset easier. Following the GCNN,

the fc6 is also extracted as the subset DFCNN features. Therefore, each DFCNN is a specialist in classifying the images in a single subset. The model's core problem is the method to select an optimal DFCNN for a particular input image.

Addressing the problem of choosing the optimal DFCNN for a given image sample, the subset selector CNN (SCNN) is introduced. SCNN follows the generic CNN-based classification model but uses the rough class clustering output as the class label and replaces the last fully connected layer, fc8, with a new fully-connected layer with K outputs. Therefore the prediction of the SCNN is a particular subset. Max voting is performed to the prediction for the final determination of the subset that the input image belongs to. Following the previous training procedure, SCNN is trained via backpropagation using the optimizer of stochastic gradient descent (SGD). It uses the AlexNet[52] pretrained on ImageNet as the initial state.

2.3.2 Mixture of Deep CNN (MD-CNN)

Following the subset specialist's idea in subset feature learning networks, MixDCNN [29] contains several CNN-based experts. Unlike previous works, the MixDCNN does not need to split the dataset into several subsets with a similar visual representation. Instead, the K CNNs perform classifications independently on the entire fine-grained visual classification dataset, and their suggestions are combined to generate the final decision. Different from the subset-based learning architectures that train the specialist CNNs separately, the occupation probability equation is applied in MixDCNN structure to end-to-end train the K CNN specialists at the same time. Figure 2.21 demonstrates the workflow of the MixDCNN.

The occupation probability function is designed as:

$$\alpha_k = \frac{e^{C_k}}{\sum_{c=1}^{K} e^{C_k}} \tag{2.1}$$

where C_k stands for the k-th CNN's classification result. The CNNs with higher confidence of ttheir classification result are granted higher weight from the occupation probability function.

Based on each CNN expert's classification confidence, the occupation probability function allows the MixDCNN structure to train the CNNs jointly with a single united label vector. Gating network to determine which CNN is trustworthy in the classification is not required. Instead, the MixDCNN mixed the outputs of all components in a more confederate way. By multiplying the occupation probability of the

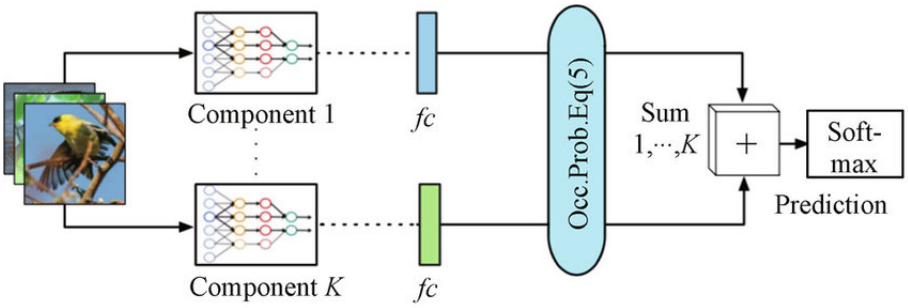


Figure 2.21: Framework of MixDCNN [29].

CNNs' outputs and summing up the features, the classification is simply performed by applying the softmax function. The experimental results show that the united framework outperforms the previous separated methods.

2.3.3 CNN Tree

It can be observed from many fine-grained visual classification datasets that a particular class with low classification accuracy only confuses with a few other categories. The subset of such classes is defined as the confusion set. Another observation is that an expert CNN can learn more discriminative features from a smaller but more elaborate subset. Based on that, Wang et al. [94] proposed the CNN tree to learn the fine-grained visual features from the confusion sets.

As illustrated in Figure 2.22, a CNN is first trained on entire dataset classes. Then the trained CNN is used to estimate the confusion sets of the dataset. The estimated confusion sets are packed to be fed to the next stage's training. From the root node CNN that classifies the entire dataset to the leaf nodes that only classify two confused classes, the procedure repeats until the tree ends up to the two-class classification leaf or reaches its maximum depth. Different from the previous methods, the CNN tree learns fine-grained features by progressively learning the discriminative features from smaller but more confused subsets. The features are more robust than those learned from the entire datasets directly. In the testing phase, an image that the top CNN misclassifies can be correctly classified by the CNN tree.

2.3.4 Multiple Granularity CNN

It can be observed that the subordinate-level annotations also carry the labels' hierarchy in the fine-grained datasets. For example, in the CUB-200-2011 dataset,

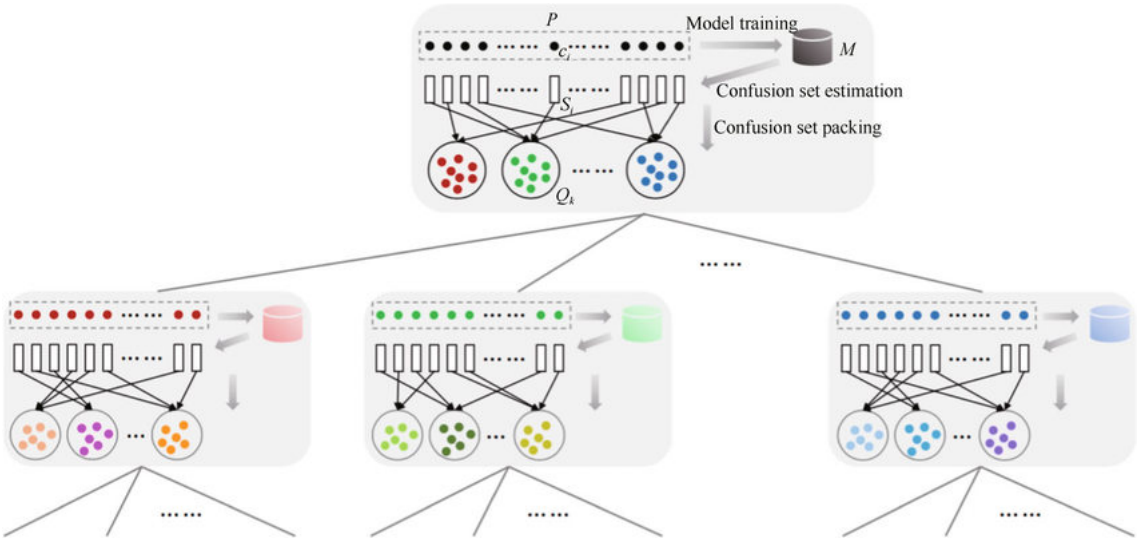


Figure 2.22: Framework of CNN tree [94].

Melanerpes formicivorus belongs to Melanerpes at the genus level and belongs to Picidae at the family level. Following the ideas to split the FGVC dataset into subsets with a hierarchy structure in the CNN tree, these genus level and family level annotations can also be utilized to train a set of CNN classifiers and form the tree structure. Each CNN only concentrates on a one-grain level classification.

As shown in Figure 2.23, the multiple granularity CNN [91] contains a series of deep convolutional neural networks in parallel structure. Each of the CNNs is designed to classify for a particular granularity, which means that they are classifiers responsible for a set of single-grained classifications. The selection of regions of interest (ROI) is guided by the saliency in the CNNs' feature maps, and the ROI is then selected from a general pool of image patches that has the higher activation from the CNNs. Finally, the ROIs are fed into the next stage classifiers for feature extraction. The features of different granularity are then combined to generate the classification results.

2.3.5 Bilinear CNN Models

Bilinear CNN (B-CNN) [62] proposed a two-stream CNN structure for fine-grained visual classification. The network structure is presented in Figure 2.24. After inference through the Convolutional layers, the two streams' output feature maps are multiplied with the inner product function, which means that the feature maps are compressed into a compact representation. This function plays a similar role to the Global Average Pooling (GAP) [60]. Still, the output feature dimension is the square

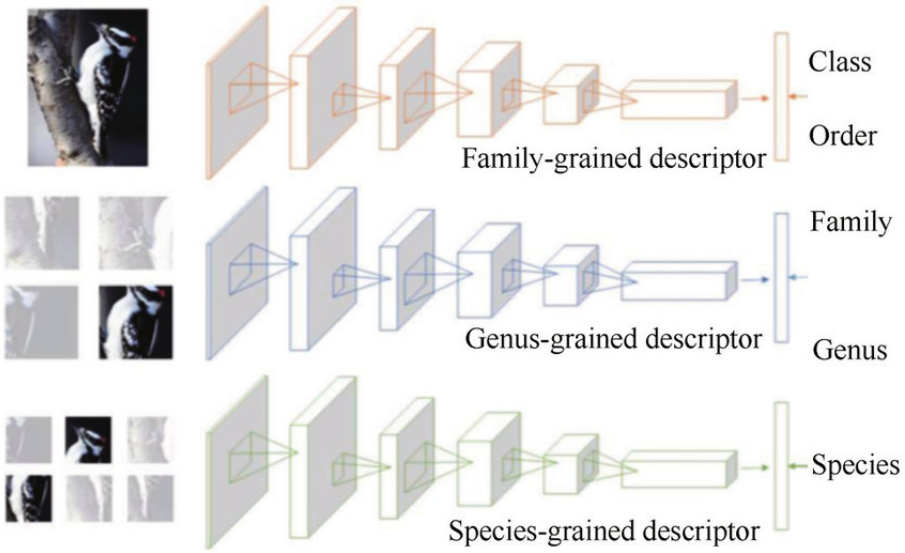


Figure 2.23: Framework of Multiple Granularity NN [91].

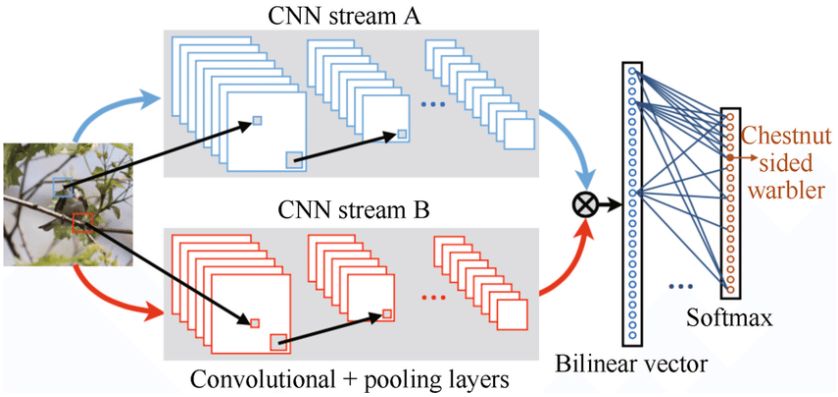


Figure 2.24: Framework of bilinear CNN model [62].

magnitude of the GAP. The bilinear function in CNN structures is also named "Bilinear Pooling" (BP). It provides very high dimension representative image features, which is especially beneficial to the fine-grained visual classification.

The bilinear CNN structure B contains four main components: $B = (f_A, f_B, P, C)$. Here f_A and f_B are two parallel feature-extraction functions, P is a pooling function and C is a classification function. A feature function is a mapping $f : L \times I \rightarrow R_c \times D$ that takes as input an image I and a location L and outputs a feature with size $c \times D$. The locations include information on position and scale. The feature outputs are multiplied at each location with the matrix outer product, i.e., the bilinear feature combination of f_A and f_B at a location l is given by bilinear function $(l, I, f_A, f_B) = f_A(l, I)^T f_B(l, I)$. Both f_A and f_B are designed to have the same feature dimension

c to be compatible. Image bilinear features are obtained by aggregating the pooling function P across all image pixels.

The original paper used the output feature maps of the last convolutional layers of two independent CNNs pre-trained with the ImageNet [15] as f_A and f_B. The bilinear CNN benefits from pre-training, for that it provided prior knowledge and made up the scarcity of the single domain dataset. Only applying the convolutional layers of the pre-trained CNNs provides another advantage that arbitrary resolution of input images can be adopted. It means that we can use high-resolution image data to discover more discriminative features for fine-grained visual classification.

For the effectiveness of bilinear pooling shown in fine-grained classification, lots of follow-up studies have been conducted to improve the performance. An end-to-end trainable matrix power normalization method was employed to improve the representation of bilinear features [61]. In a parallel research track, low dimension compact bilinear pooling has been explored. PCA low-rank approximation of CNN features before bilinear pooling was investigated [62]. The bilinear pooling and the linear classifier with a second-order polynomial kernel were bridged [25] by adopting the off-the-shelf kernel approximation methods, Random Maclaurin [47], and Tensor Sketch [73]. Tensor Sketch was iteratively generated [13] to achieve a higher-order feature polynomial. Inspired by the bilinear SVM, an imposed low-rank constraint was proposed to reduce the feature dimension [50]. After all, none of the above compact methods considered the implementation of matrix-power normalization, which requires the computation of singular value decomposition (SVD) of the output of tensor product and the same to the backpropagation. Sub-matrix square root was proposed [34] to normalize the CNN feature before tensor production to approximate the matrix-square root operation on bilinear features. Although the performance of this method is better than that obtained by applying element-wise normalization only, the classification accuracy is sacrificed by 1% comparing with the full MoNet baseline.

2.4 Attention-Based Methods

Attention is the universal phenomenon of the biological visual system. Instead of encoding the entire image into a compressed feature representation, animals (including humans) usually focus on the most discriminative regions and ignore the rest. Similarly, in computer vision, salient features can be prior extracted by a visual attention system and come to the forefront dynamically as required by the fine-grained classification applications. Especially for the image datasets with a lot of clutters. In

this subsection, fine-grained visual classification methods using attention structures are described in detail.

2.4.1 FCN Attention

FCN attention [64] is a fully convolutional attention localization network that is based on reinforcement learning. The purpose of the architecture is to select several task-driven visual attention regions from the input image adaptively. Different from the previous FGVC models based on reinforcement learning [2][70][77], the proposed approach is significantly more efficient on computational complexity in both the training and the testing phase. The reason can be ascribed to its fully-convolutional architecture, and it is capable of simultaneously focusing its glimpse on multiple visual attention regions.

The fully convolutional attention localization network's architecture is illustrated in Figure 2.25. It localizes multiple discriminative object parts with the attention-based method. Different parts cover different pre-defined sizes on an object. The network consists of two components: a part localization and a classification. A fully convolutional neural network is applied in the part localization component. For a particular input image, the presentation feature maps are extracted using the convolutional layers of the VGG-16 model [80]. The model is pre-trained on the large-scale ImageNet dataset [15] and fine-tuned on the target dataset. By generating a confidence map for each part using the feature maps produced by the localization CNN, multiple object parts are located with the attention localization network. Two stacked convolutional layers and one spatial softmax layer are used in the localization CNN to generate each part's confidence map. Sixty-four 3×3 kernels and one 3×3 kernel are used in the first and the second convolutional layer, respectively, to produce a part localization confidence map. Then the confidence map is fed into the spatial softmax layer and transferred to localization probability. The activated region with the highest localization probability is determined as the corresponding part region. This procedure is adopted to a pre-set fixed number of time steps for the parts localization. A single particular part is generated with each time step.

The classification stream consists of an independent deep CNN classifier for the whole image and every object part. Different parts of the same object might cover different visual sizes on the object. A local part image segment is cropped around the parts' locations concerning its pre-set size. The whole image and the local image segments of different parts are used to train an image classifier. All the output results from the part classifiers are averaged to produce the final classification result. Each

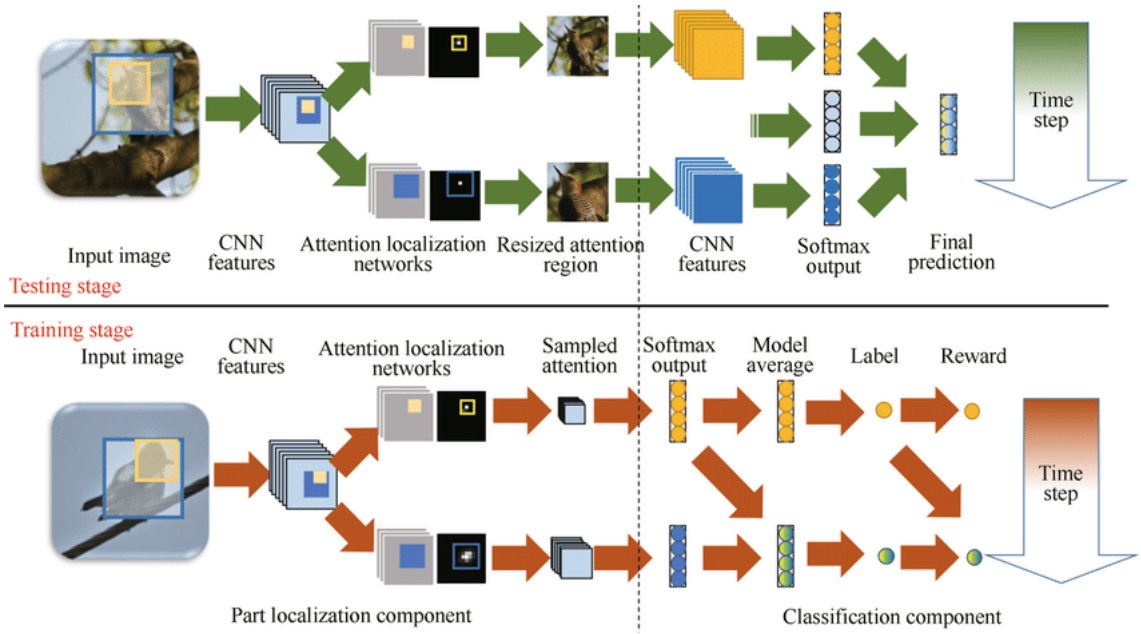


Figure 2.25: Framework of FCN attention [64].

local part region is resized to a higher resolution to discriminate the subtle visual differences better.

2.4.2 Diversified Visual Attention Net (DVAN)

To persevere the diversity of multiple attention and extract the most discriminative information from the image, the Diversified Visual Attention Network (DVAN) [111] is proposed. Figure 2.26 presents the architecture of the DVAN model, which indicates that the model includes four main components: (1) attention canvas generation function; (2) CNN feature learning network; (3) diversified visual attention extraction; and (4) the final classification. Firstly, DVAN performs localization of several regions with different sizes and crops those image regions to generate the "canvas" for the next stage's visual attention extraction. The representation features in each attention canvas are learned with a pretrained CNN (here, VGG-16). A multiple visual attention CNN is adopted to generate the activation maps to localize each canvas's important visual regions. In this way, the critical part locations in the canvas are extruded, and discriminative information obtained from the attention canvases is maximized. Different from the traditional attention-based structures paying attention only to a single discriminative region, DVAN performs classification using the multiple interest regions in an image, which benefits from a customized diversity-enhancing loss function.

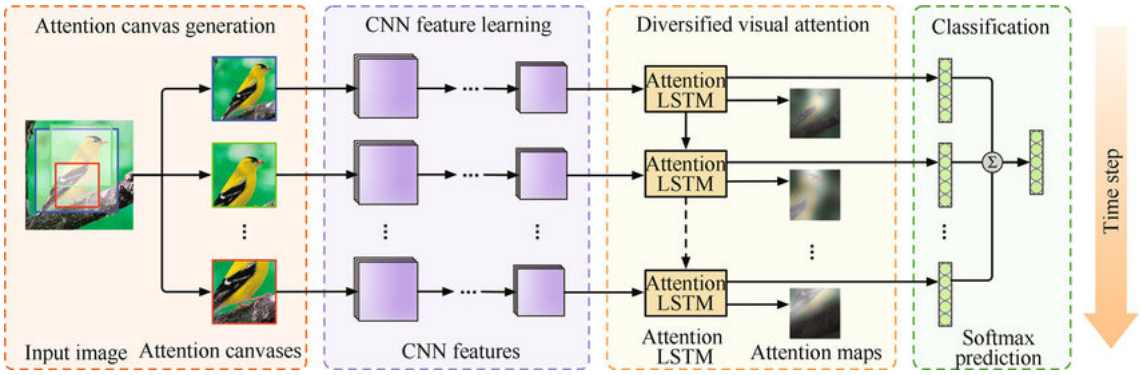


Figure 2.26: The framework of diversified visual attention networks [111].

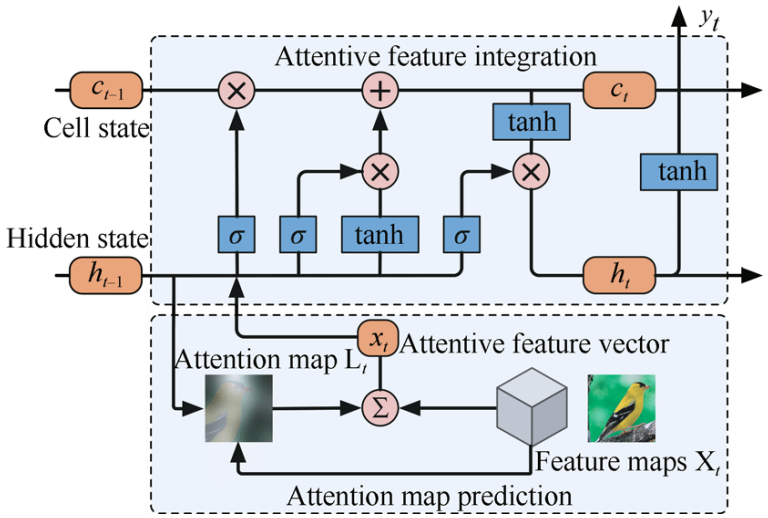


Figure 2.27: The DVAN attention component [111].

As illustrated in the top and bottom panels in Figure 2.27, the visual attention adopted in DVAN contains two major components: the attentive feature integration module and the attention map prediction module. DVAN also applied novel diversity-enhancing loss function and attention canvas construction techniques to diversify the attention regions and improve the prediction accuracy.

2.4.3 Recurrent Attention CNN (RA-CNN)

Unlike previous attention-based models, the Recurrent Attention CNN (RA-CNN) [22] recursively learns discriminative region attention and region-based feature representation in a mutually reinforced manner. The proposed RA-CNN is a stacked network that takes the input from full images to fine-grained local regions at multiple scales. First, the same network architecture is shared by the multi-scale networks. The network architecture uses different parameters at each scale to fit the inputs with

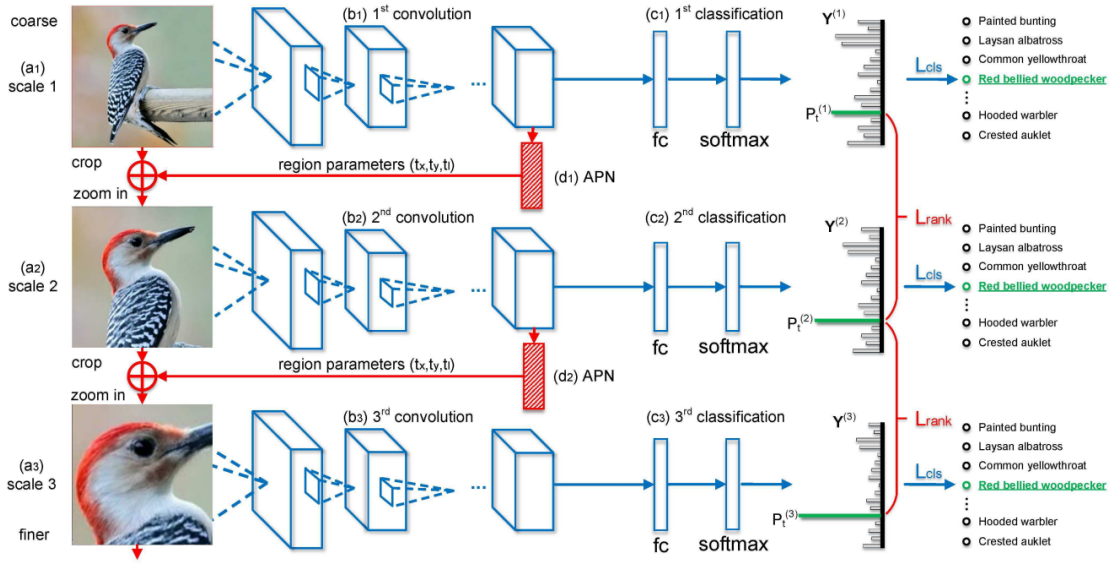


Figure 2.28: The workflow of recurrent attention CNN [22].

different resolutions. The learning procedure at each scale contains a classification sub-network and an Attention Proposal sub-Network (APN), ensuring adequate discrimination ability at each scale and generating an accurate attended region for the next finer scale. Second, a finer-scale network dedicated to high-resolution regions takes as input an amplified attended region for extracting more fine-grained features. Third, the recurrent network is alternatively optimized by an intra-scale softmax loss for classification and an inter-scale pairwise ranking loss for the attention proposal network. The network is optimized by the ranking loss to generate higher confidence scores on correct categories than the previous prediction.

Since finer-scale networks can be stacked recurrently, RA-CNN can gradually attend to the most discriminative regions from coarse to fine (e.g., from the body to head, then to beak for birds). Note that accurate region localization can help discriminative region-based feature learning and vice versa. Thus the proposed RA-CNN can benefit from the mutual reinforcement between the object part localization and the feature learning. To further leverage the benefits from the ensemble learning, features from multiple scales are genuinely fused to classify an image by learning a fully-connected fusion layer. The workflow of the RA-CNN is demonstrated in Figure 2.28.

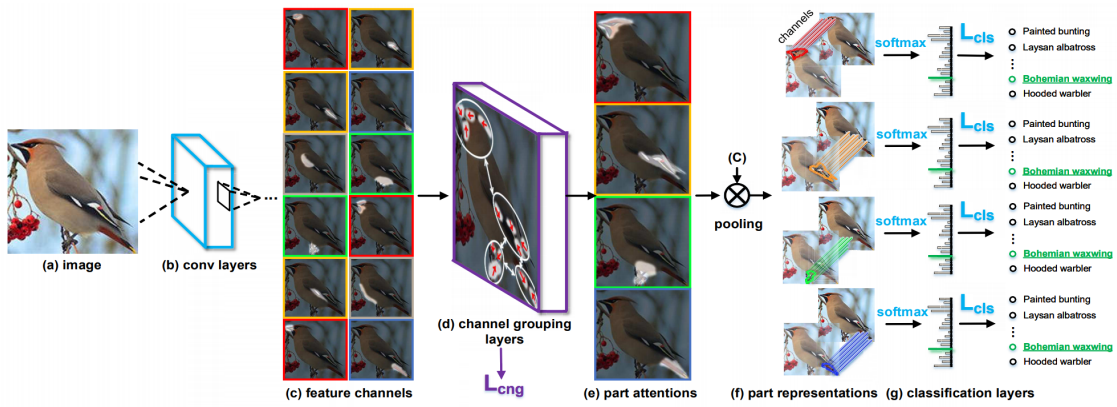


Figure 2.29: The workflow of Multi-attention CNN [112].

2.4.4 Multi-Attention CNN (MA-CNN)

Following the approach in [22], the Multi-attention CNN (MA-CNN) model was proposed [112]. It is a novel framework with significant improvement in two aspects. Firstly, it proposed the one-squeeze multi-excitation module (OSME) to localize different parts inspired by the ImageNet winner in 2018, Squeeze-and-Excitation Networks (SENet) [41]. It is fully differentiable and can directly extract part features with a budgeted computational cost. Secondly, inspired by metric learning loss, the multi-attention multiclass constraint (MAMC) was proposed to enforce the correlations among different parts in training coherently. The workflow of MA-CNN is shown in Figure 2.29.

The paper made the following contributions to solve the problem of the attention-based fine-grained visual classification: 1) The detected parts should be well spread over the object body to extract noncorrelated features; 2) Each part feature alone should be discriminative for separating objects of different classes; 3) The part extractors should be lightweight in order to be scaled up for practical applications. The experimental results show that MA-CNN achieved substantial improvements on four benchmark fine-grained classification datasets.

2.4.5 Navigator-Teacher-Scrutinizer Network (NTS-Net)

Following the typical structure of attention-localization and classification workflow, the Navigator-Teacher-Scrutinizer Network (NTS-Net) [103] was proposed. It is equipped with a navigator agent, a part teacher agent, and a scrutinizer agent as three stages of a classification strategy, as illustrated in Figure 2.30. Different from the previous methods, the NTS-Net took into consideration the intrinsic consistency

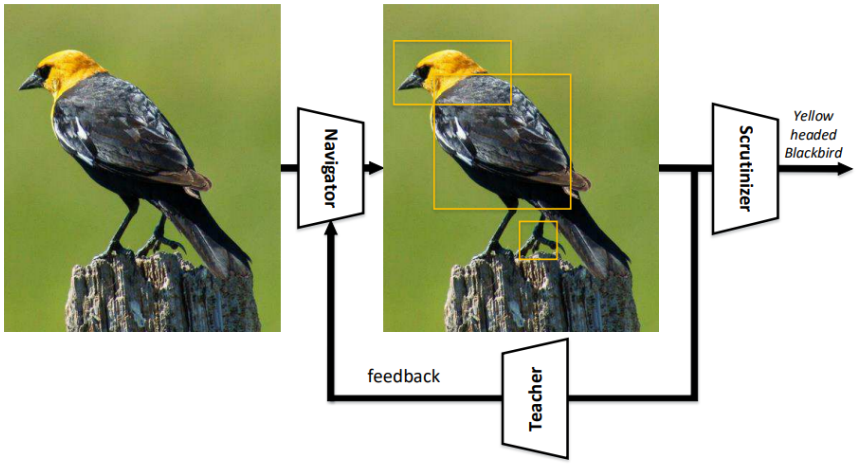


Figure 2.30: The workflow of the NTS-Net. The feature extractor extracts the deep feature map from each input image, and feeds the feature map into to Navigator network to generate the activations. Here we select the top-3 activations and crop the correspondent regions as the second layer classification navigator [103].

between informativeness of the regions and their probability being ground-truth class was taken into consideration. Hence a training paradigm was designed in the NTS-Net to detect the most discriminative regions.

The NTS model can also be viewed as a part-based or multiple-model method. Since the most significant improvement is the Navigator, which uses the attention map to discover the most activated regions and train the part teacher agent, we classify it as an attention-based method. The model can be trained end-to-end and has no requirement for extra annotations except class labels. This makes the model different from the typical part-based or duplicated model-based methods.

2.4.6 Mixture of Granularity-Specific Experts CNN (MGE-CNN)

Different classes of objects in fine-grained classification tasks typically have small inter-class variances and large intra-class variances. A simple solution for this problem is to divide an FGVC dataset into some class subsets and train several "experts". However, the generated expert model is prone to overfitting because small datasets are used in training.

The Mixture of Granularity-Specific Experts CNN (MGE-CNN) [107] was proposed to address the overfitting problem. It is a novel structure consisting of several classification experts and a Gating Network (presented in Figure 2.31). For each input

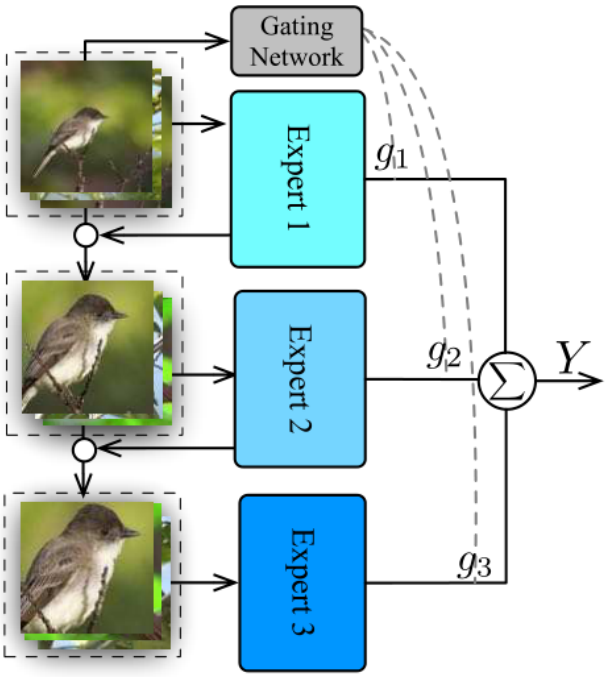


Figure 2.31: The workflow of the MGE-CNN [107].

image, a classification expert generates a Category Attention Map (CAM) and crops attentional regions to feed into the next expert. In MGE-CNN, the KL divergence was introduced into the loss function to force different experts to focus on different attention regions:

$$L = \sum_{t=1}^{T} L_{cls}^{t} + \sum_{t=2}^{T} L_{KL}^{t} + L_{gate}. \tag{2.2}$$

The Gating Network in MGECNN combines the class scores from class experts and outputs the final classification result:

$$\hat{y}_i = \sum_{t=1}^{T} g_t * \hat{y}_i^t \tag{2.3}$$

2.4.7 Weakly Supervised Data Augmentation Network (WS-DAN)

The Weakly Supervised Attention Learning was proposed in [42]. It can generate attention maps to represent the spatial distribution of discriminative object parts and

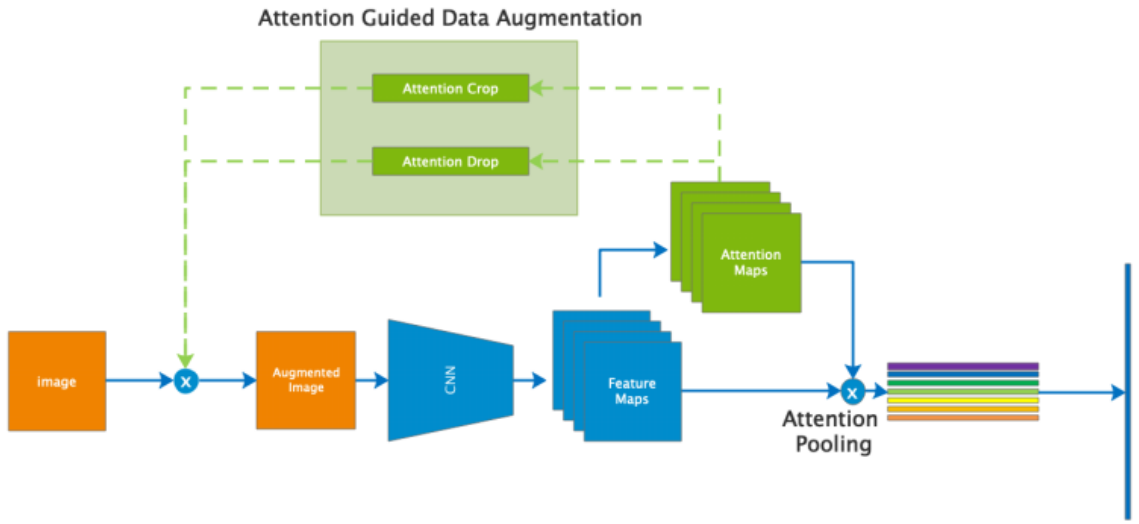


Figure 2.32: The workflow of Weakly Supervised Data Augmentation Network [42].

extract sequential local features to solve the fine-grained visual classification problem. The center loss, which is widely used in face landmark detection, is introduced into the fine-grained visual classification to obtain the feature maps of discriminative object parts. Based on the attention maps extracted by weakly supervised learning, attention-guided data augmentation is conducted to improve data augmentation efficiency, including attention cropping and attention dropping. The attention cropping crops randomly and resizes one of the attention parts to enhance the local feature representation. The attention-dropping function randomly erases one of the attention regions out of the image to encourage the model to extract features from multiple discriminative parts.

In WSDAN, Inception-V3 was used as the backbone, and the bilinear inner product was employed to extract discriminative local visual features. The training and testing flows are illustrated in Figure 2.32.

2.4.8 Weakly Supervised Complementary Parts Models

Weakly supervised complementary parts models [28] were designed in weakly supervised convolutional neural networks to retrieve discriminative features in dominant object regions. The model extracts rough object instances using image-level annotations only. Weakly-supervised object detection and instance segmentation are performed using Mask R-CNN and CRF-based approaches. In the next step, under the principle of preserving as much diversity as possible, the model evaluates to seek the optimal parts model for each object instance in the input images. The corresponding

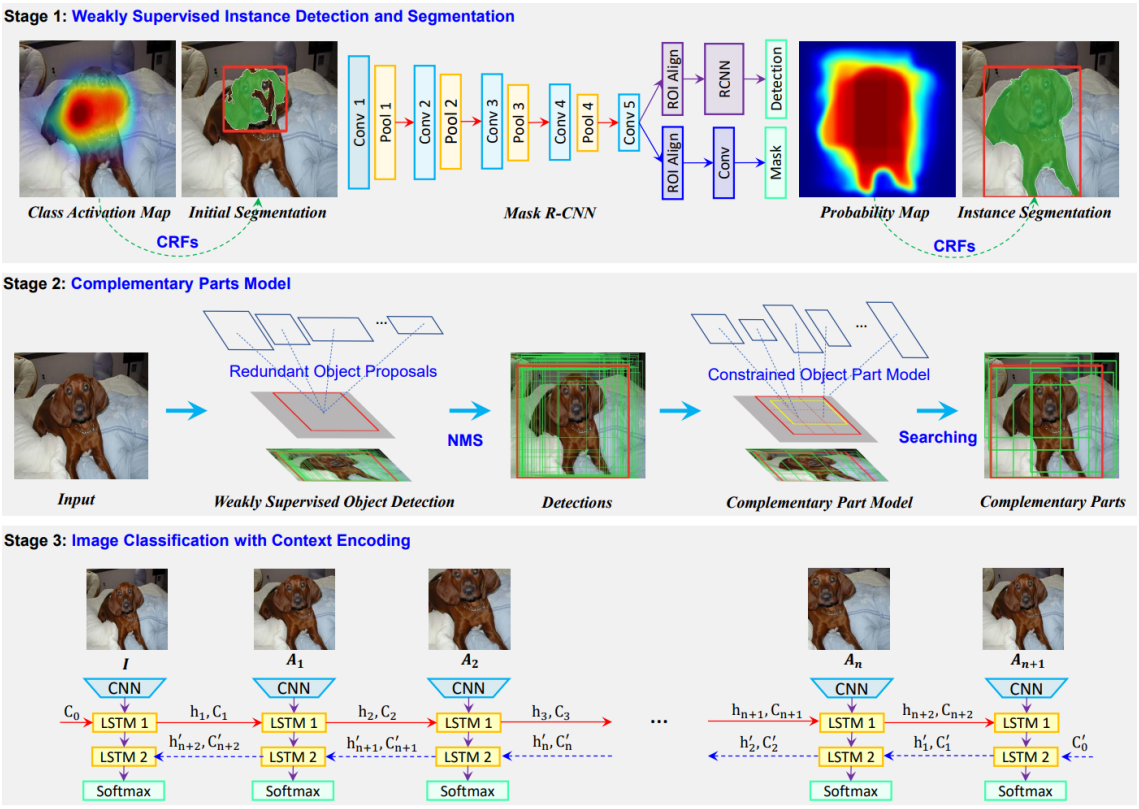


Figure 2.33: The workflow of Weakly Supervised Complementary Parts Models [28].

object parts' representation features are combined and encoded by a bi-directional long short-term memory (LSTM) network to generate a comprehensive feature set for image classification. The workflow is demonstrated in Figure 2.33.

2.4.9 Diversification Block (DB)

Diversification Block (DB) [81] was proposed for the fine-grained classification tasks to discriminate category subsets with a close relationship better. As shown in Figure 2.34, the model consists of two modules: (1) the diversification module forcing the network to discover subtle but discriminative features between each pair of categories; (2) a gradient-boosting loss function focusing explicitly on distinguishing class pairs with high visual similarity. The diversification block addresses where and how to suppress the attention activation by using a peak and a patch suppression module. The Gradient-boosting Loss introduces the Heap-max and Thresholding module before applying the typical cross-entropy Loss function to avoid the extra noise produced by the diversification block. The DB solution has a simple and elegant structure, demonstrates a superior classification performance, and performs lower computational

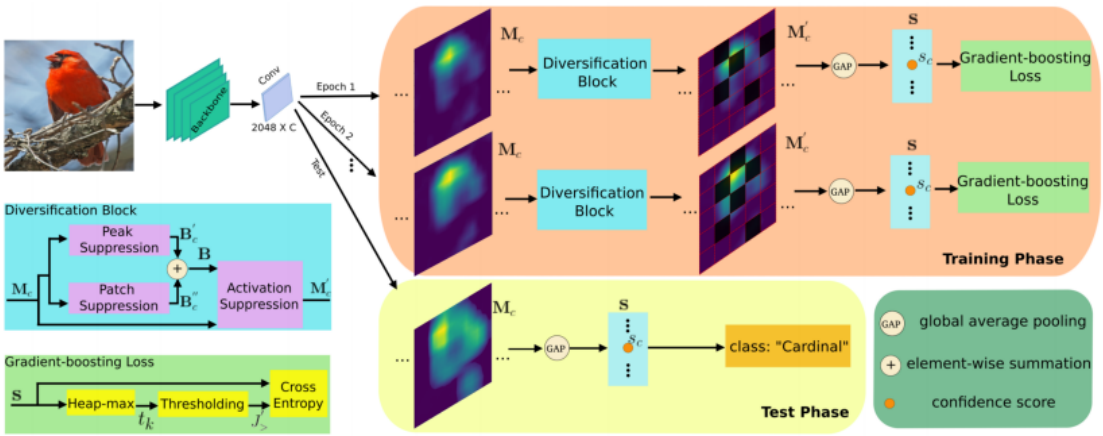


Figure 2.34: The workflow model with Diversification Blocks [81].

	train	test	class
CUB-200-2011([90])	5994	5794	200
FGVC-Aircrafts([66])	6667	3333	100
Stanford Cars([51])	8144	8041	196

Table 2.1: Summary of the benchmark datasets used in the experiments.

complexity.

2.5 Fine-Grained Visual Classification Datasets

This section introduces the benchmark fine-grained visual classification datasets popularly used by FGVC researchers. Our experiments were also conducted on these datasets, including CUB-200-2011 Bird dataset [90], FGVC-Aircrafts [66] and the Stanford Cars [51]. All these datasets provide a fixed train and test split. The statistics of these datasets are summarised in Table 2.1.

The CUB-200-2011 Bird dataset [90] contains 11,788 images of 68 primary and 200 subclasses of birds in natural environments. Differences among subclasses could be very tiny, making it very challenging for vision systems to discriminate automatically and accurately. In the dataset, each image contains only one bird, and its bounding box and 15 key-point locations (beak, back, breast, belly, forehead, crown, left eye, left leg, left-wing, right eye, right leg, right-wing, tail, nape, and throat) are annotated. It should be noted that due to occlusion or limitation of camera views, some part locations are missing in the annotation of some image samples. A dataset splitter

is provided to produce a training set with 5,994 images and a testing set with 5,974 images.

The Stanford Cars [51] dataset contains 16,185 images of 196 car models. The dataset is pre-split into 8,144 images in the training set and 8,041 images in the testing set. Each class has been split roughly in 50%-to-50%. For the finest-grained level of annotation, the dataset is categorized by make, model, and year (e.g., BMW M3 Coupe 2012 or Tesla Model S 2012). Besides, it also has annotations of color (e.g., yellow vs. red) and body type (e.g., sedan vs. SUV).

The FGVC-Aircrafts [66] dataset contains 10,200 images of 100 different aircraft models, with 102 images for each model. The annotation of the dataset includes a bounding box and a model label for the primary airplane in each image. The label annotation includes information about the model, family, and manufacturer. The dataset is split into training and testing sets by 66.66%-to-33.33%.

Figure 2.35 displays sample images from each of the datasets.

2.6 Performance Comparison and Analysis

Table 2.2 compares the classification accuracy of each of the deep learning models in the four categories on the CUB200-2011 dataset [90]. The classification accuracies mentioned in the rest of this thesis are defined as the average classification accuracy of all classes in the FGVC dataset. The dataset contains 11,778 images from 200 bird species. The dataset also provides rich annotations, including image-level class labels, bounding boxes of the major object, attribute annotations, and fifteen components' landmarks. Note that the results from some models are not presented in Table 2.2 because these models were not evaluated with this dataset.

Table 2.2 groups the experimental results of each method into four categories. The first category of models includes baselines whose backbone structures are widely used in traditional object classification. Some data augmentation and transfer learning-based FGVC models are also listed in the category. The second category shows the performances of the part-based models. The methods shown in the third category are based on ensembled multiple neural networks or encoded image features to improve the performance of fine-grained classification. The attention-based models are evaluated in the fourth category. They usually do not require the bounding box or part landmark annotations and mimics human beings' recognition procedure. Depending on the time they were proposed, these approaches might use different CNN structures such asAlexNet [52], VGGNet [80], GoogLeNet [84], or DenseNet [44]. Moreover, some

Figure 2.35: Samples from the three benchmark FGVC datasets. For each dataset, three images from each of randomly selected two classes are presented to show intuitively the FGVC datasets used in our experiments.

of these approaches may utilize the annotation of object bounding boxes or part landmarks in the training or testing phase, while other models only require the category labels for the training. The backbone CNN model and the annotation they are using are noted in the table.

For traditional approaches, the net structures before 2017, such as AlexNet [52], GoogLeNet [84], VGG-16 [80] and ResNet50 [36], could only obtain an accuracy lower than 79%. However, DenseNet161 [44] which was proposed in 2017 achieved an accuracy of 84.2% with a single model. With the latest auto-searching based EfficientNet-b7 [85], an accuracy of 88.5% was obtained with only a single model and a single training phase. Meanwhile, some other methods using auto data augmentation or customized loss function also achieved impressive performance. For example, both API [18] and Mix+ [54] produced an accuracy of over 90% on the CUB-200-2011 dataset.

Generally, part-based fine-grained classification methods can localize critical regions using a series of manually pre-defined semantic parts. However, It is both difficult and expensive to obtain the part landmarks for so many images. More recent fine-grained visual classification techniques can automatically learn one or more discriminative region detectors using category-level annotations with hierarchical reinforcement learning. The disadvantage of the attention-based models is that It is usually difficult for them to localize the semantic object accurately without utilizing the explicit part annotation. The discovered attention parts lack semantical interpretability. The attribute annotation is more manageable considering the great labor and time consumption of part annotation for fine-grained visual classification. Therefore, we can utilize the attribute label information as weak supervision to improve the classification accuracy further. The best traditional part-based method we have investigated is the Deep LAC [59], which only achieved 80.3% accuracy on CUB-200-2011 dataset. However, with the attention cropping module and deeper backbone structure, the AttNet [35] achieved an accuracy of 88.9%, which was also in the first echelon of the top-performing FGVC approaches.

Duplicated models or higher-order feature encoding methods achieved better accuracy and required more affordable human labor than part-based methods. Bilinear models, particularly, obtain an impressive accuracy of 88.7% with the backbone of ResNet-101. Meanwhile, the recurrent visual attention models can effectively localize the discriminative regions and learn their representations with an end-to-end procedure. In recent years, a lot of recurrent visual attention-based models have been

proposed. The roadmaps of the attention-based approaches fall into two main categories: soft attention-based or hard attention-based. Models based on soft attention [22][112] extract the attention regions deterministically. Consequently, it is differentiable and can be trained end-to-end using backpropagation. Hard attention models [111][64][42][28] use the stochastic attention points of the images. Reinforcement learning is usually applied to the training of the hard attention models. Generally, the soft attention-based approaches are more efficient than the hard attention-based approach, for that it usually requires duplicated training procedure in hard attention models. In contrast, soft attention-based models can be trained in a single end-to-end procedure.

In practice, there are still several drawbacks to visual attention-based models. Firstly, performance improvements are not satisfying using soft attention-based models. A more robust visual attention model is supposed to improve FGVC accuracy more significantly. Secondly, reinforcement learning is used in hard attention-based methods, making it difficult to implement and usually not as efficient as soft attention-based methods. Approaches to improve the hard attention-based methods' efficiency and assemble the different attention extraction stages should be explored further.

In conclusion, attention-based methods provide the best performance and have immense research prospects.

Method	Architecture	Input	Train	Test	Accuracy(%)
AlexNet[52]	AlexNet	224	—	—	69.1
GoogLeNet[84]	GoogLeNet	224	—	—	68.2
VGG-16[80]	VGG-16	224	—	—	73.3
ResNet50[36]	ResNet50	224	—	—	78.2
ResNet50[36]	ResNet50	448	—	—	85.4
DenseNet161[44]	DenseNet161	448	—	—	84.2
EfficientNet-b0[85]	EfficientNet-b0	448	—	—	84.7
EfficientNet-b7[85]	EfficientNet-b7	448	—	—	88.5
DCL[10]	ResNet-50	448	—	—	87.8
S3N[16]	ResNet-50	448	—	—	88.5
API[18]	DenseNet161	448	—	—	90.0
MC-Loss[9]	Bilinear CNN	448	—	—	86.4
Mix+[54]	ResNet-50	448	—	—	90.2
PB-R-CNN[108]	AlexNet	224	BBox+Parts	BBox	76.4
PB-R-CNN[108]	AlexNet	224	BBox+Parts	—	73.9
MPC[79]	AlexNet	224	BBox	BBox	80.3
PoseNorm[6]	AlexNet	224	BBox+Parts	—	75.7
PS-CNN[45]	AlexNet	224	BBox+Parts	BBox	76.2
Deep LAC[59]	AlexNet	224	BBox	BBox	80.3
AttNet[35]	ResNet-101	448	—	—	88.9
Subset FL[30]	AlexNet	224	—	—	77.5
MixDCNN[29]	AlexNet	224	BBox	BBox	74.1
MG-CNN[91]	VGG-16	224	BBox	—	83.0
MG-CNN[91]	VGG-16	224	—	—	81.7
Bilinear CNN[62]	VGG-16	448	BBox	BBox	85.1
Bilinear CNN[62]	VGG-16	448	—	—	84.1
BoostCNN[71]	VGG-16	448	—	—	86.2
PC[19]	DenseNet-161	448	—	—	86.8
iSQRT-COV[55]	ResNet-101	448	—	—	88.7
DVAN[111]	VGG-16	224	—	—	79.0
FCN attention[64]	GoogLeNet	224	BBox	—	84.3
FCN attention[64]	GoogLeNet	224	—	—	82.0
RA-CNN[22]	VGG-16	448	—	—	85.3
MA-CNN[112]	VGG-16	448	—	—	86.5
NTS-Net[103]	ResNet-50	448	—	—	87.5
MGE-CNN[107]	ResNet-50	448	—	—	88.5
DB[81]	ResNet-50	448	—	—	88.6
WS-DAN[42]	Inception-V3	448	—	—	89.3
WS-CPM[28]	GoogleNet	448	—	—	90.3

Table 2.2: Comparison of classification performance of different FGVC models on CUB-200-2011 dataset. The baselines of AlexNet, GoogLeNet, VGG-16, ResNet50, DenseNet161, EfficientNet-b0, and EfficientNet-b7 are conducted by this thesis, while the rest are reported in the original papers.

Chapter 3

Part Collaboration Convolutional Neural Network

Part based approaches have been proven effective in previous works [59][108][6][79][45]. By localizing parts in an object and construct their feature correspondences among different species, these methods can amplify the subtle distinguishing visual characteristics. Moreover, the localization-classification scheme conforms to traditional taxonomic conventions and can be applied to other fields like image caption, literal search engine, and discrimination interpretation. Plagued by increasing model complication from multi-view end-to-end parallel deep network, however, previous works benefited far not enough from part level detail information.

This chapter presents an end-to-end part-based Part Collaboration CNN (PC-CNN) architecture for fine-grained categorization. Similar to previous part-based approaches, the proposed method extracts multi-view visual features to train the FGVC classifier. To avoid dimension boosting caused by additional part streams, ResNet structure and a fully convolutional network were applied to reduce the multi-view feature dimension, and independent part feature extractors were implemented. The major contributions of this work include:

- A novel multi-view categorization architecture was proposed based on deep CNN, which can take into consideration an unlimited number of object parts and the pose information.
- A training strategy was proposed, which was capable of avoiding under fitting caused by the unbalance among the discrimination abilities of different parts while keeping the globe optimization achievable.

- The proposed method achieved an accuracy of 84.1% with the ground truth part annotation for the CUB-200-2011 dataset, superior to the accuracy of 82.02% reported in the state-of-the-art works [108][45].
- The parameter number of the proposed model is 93.3 million for 15 parts, which is 28.7% less than that in [45]. The processing speed for the proposed 7-parts model is 73 frames per second on a TITAN X (Pascal) GPU, which is marginally slower than the basic ResNet-50 model [36] (84.5 frames/sec).

The work of this chapter has been published in the proceedings of 2018 Digital Image Computing: Techniques and Applications (DICTA) [56].

The chapter is organized as follows. Detailed introduction of PCCNN nets and their training method is described in Section 3.1. Experimental results and analysis on the performance of the proposed model and its interpretability are detailed in Section 3.2, followed by conclusions and discussions on the structure of the PC-CNN and future work in Section 3.3.

3.1 PC-CNN Structure

While Zhang et al. [108] and Huang et al. [45] proved the effectiveness of side net and increased the accuracy of the primary AlexNet [52] by 6%, with only two rough parts (head and body) utilized, their methods were limited by the R-CNN-based localization framework and 2-step classification network structure. To address these problems, an FCN part-localization architecture was designed [45] that was capable of precisely detecting coordinated points of quantity object parts. Based on that, a 2-stream end-to-end network architecture was proposed, containing an object stream for bounding-box level feature extraction, a shared part stream for the feature extraction of all candidate parts, and a three fully connected-layer classifier. However, its FCN based localization network is not accurate enough due to the limitation of feature map resolution, and the FCN often overlooks the potential relationship among the locations of different parts. Moreover, sub-classifiers should be customized based on the visual appearance of each part. However, this was not taken into consideration in [45]. Their model also ignored the pose information that can be potentially useful in the classification to improve the generalization capacity of the model.

A novel structure was proposed to address these problems by adopting ResNet-50 [36] as the object level feature extractor, regression-based localization network, and AlexNet 5-layer convolutional network as the part feature extractor. A joint layer

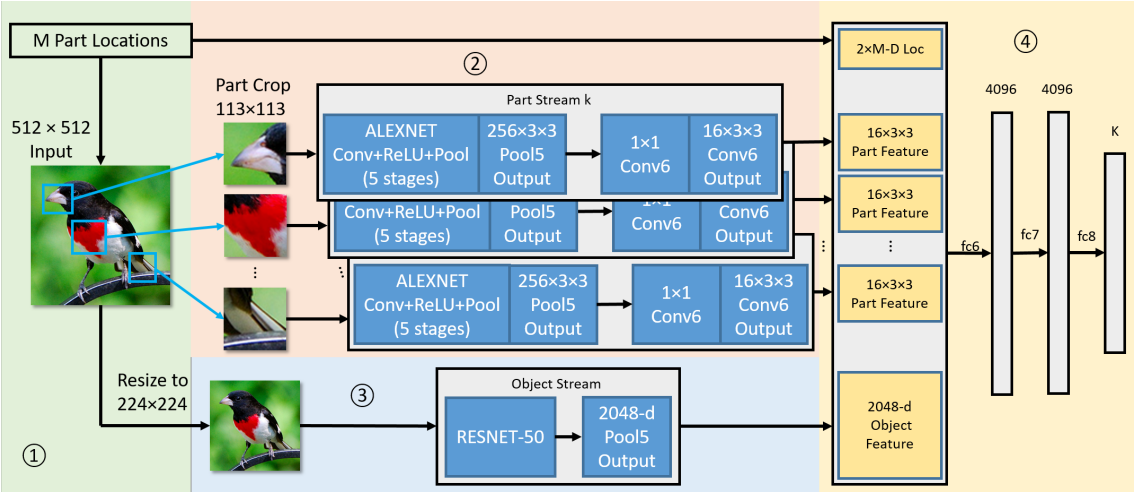


Figure 3.1: The network architecture of the proposed PC-CNN categorization model. With the PC-CNN, 1) the input image is resized into 224×224 for object stream, and cropped into M 113×113 parts for part streams; 2) M part streams take cropped part images as input, and extract the most representative $M \times 16 \times 3 \times 3$ features independently; 3) the object stream takes the object image as input and extract a global 2048-dimensional feature set; 4) unify object and part features and utilize pose information to achieve the final multi-view feature for the classifier with 3 fully connected layers.

is designed to fuse global, part, and pose features. The uniqueness of the proposed architecture is its customized part classifier and feature fusion strategy. The experimental results show that the proposed architecture works well, and its classification accuracy has been significantly improved with minimal increase in computation cost. Figure 3.1 illustrates the proposed framework.

3.1.1 Part Localization Network

The first stage of the proposed method is part localization that aims to detect the 2-D locations of each object part defined in CUB-200-2011 [90]. Different from [45], which applies a fully convolutional network, a novel net structure is proposed to reduce location sampling errors from convolutional feature maps substantially. Moreover, compared with the fully convolutional network, the fully connected layers can learn more in-depth into the potential correspondence among each object's parts, leading to high localization accuracy.

Unlike most of the regression-based keypoint localization applications such as human pose estimation [102] or face keypoint detection [82], object part localization for fine-grained classification suffers a lot from occlusion and deformation. Figure 3.2

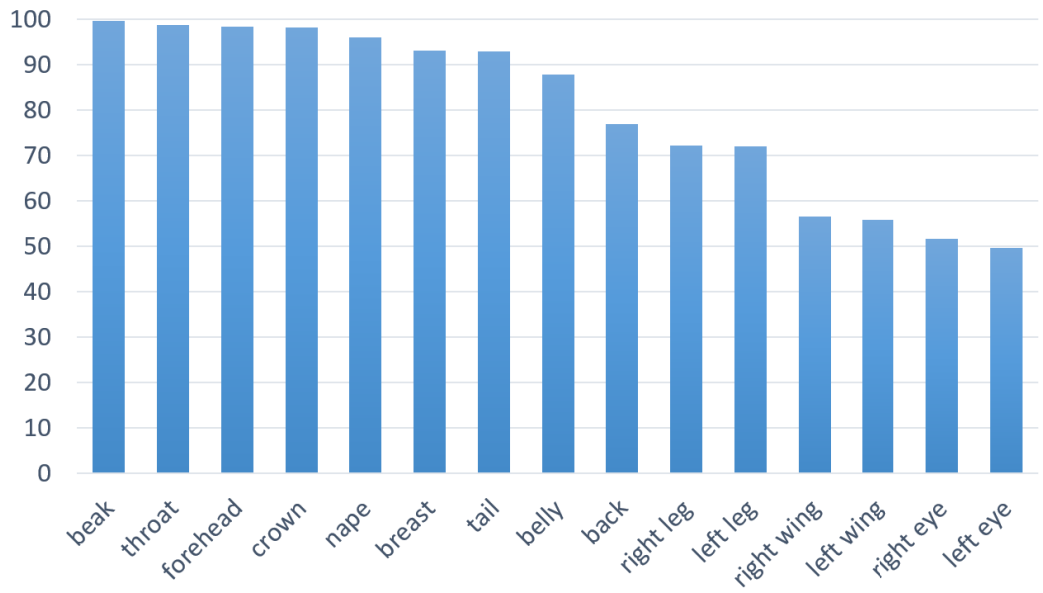


Figure 3.2: The percentage of the images in CUB-2011-200 dataset that contains each part of a bird. 99.64% of the images in the dataset contain beak, while the back only appears in 76.89% of the images. Due to occlusion, right wing, left wing, right eye, and left eye only show in around half of the images.

shows the unbalanced percentage of the images in CUB-2011-200 dataset that contains each part of a bird. As a consequence, lots of parts are annotated but are hard to detect in CUB-200-2011. To address this issue, a fully convolutional network was introduced [45] to transform localization into a pixel classification problem. However, the localization accuracy is limited by the resolution of feature maps produced by CNN. The output of CNN has a resolution of 27×27. Therefore, some artificial postprocessing like smoothing and maximum detection must be done to determine the final location.

To address this problem, a regression-based keypoint localization method is proposed. By separating part detection and localization networks and designing detection and localization loss functions, the proposed method can accurately learn part locations while detecting missing parts caused by occlusion. As illustrated in Figure 3.3, the proposed network consists of three parts: feature extraction network, part detection network, and part localization network. The feature extraction network applies the convolutional layers of ResNet-50 [36] and provides 2048-dimension output. Supposing N is the number of parts utilized in the proposed model, part detection and part localization networks take the feature vector as input, generate N dimension detection $(d_1, d_2, ..., d_N)$ and $2 \times N$ rough localization coordinates $(x_1, x_2, ..., x_N, y_1, y_2, ..., y_N)$

as output.

Part detection network consists of two fully connected layers, the first layer produces 512 dimension feature vector and the second one output N dimension detection. If the $i-th$ part appears in the image, set annotation $\hat{d}_i = 1$, else set $\hat{d}_i = -1$. Then model output d_i can determine the possibilities of each parts appearing in the image. Set loss function for the $i-th$ part is :

$$loss_{detect}(i) = \begin{cases} 0 & d_i \times \hat{d}_i = 1 \\ (d_i - \hat{d}_i)^2 & otherwise. \end{cases} \tag{3.1}$$

Then the overall detection loss is the summation of the loss of each part:

$$loss_{detect} = \sum_{i=1}^{N} loss_{detect}(i) \tag{3.2}$$

In the backward processing of the detection network, it takes $loss_{detect}(i)$ as the diff vector. In the testing stage, a threshold $\delta \subset (-1, 1)$ is set to determine whether a part appears in an image: if the output $d_k > \delta$, the model labels the $k-th$ part as being detected.

Part localization network structure is similar to the detection net, but its output number is 2×N. The annotated localization coordination $(\hat{x}_i, \hat{y}_i)$ is normalized to $(0, 1)$ in preprocessing. The loss function of the i-th part is defined as follows:

$$loss_{loc}(i) = \begin{cases} 0 & \hat{d}_i = -1 \\ (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 & otherwise. \end{cases} \tag{3.3}$$

In the back propagation, the proposed model only calculates the detected parts, the diff of the coordinates of undetected parts is set to 0. Similarly, the overall localization loss is defined as follows:

$$loss_{loc} = \sum_{i=1}^{N} loss_{loc}(i) \tag{3.4}$$

3.1.2 Two Stream Feature Extraction Structure

As described in Figure 3.1 (3), the object stream is constructed on the general architecture of ResNet-50[36], which takes a 224 × 224 RGB image as input and the 2048-dimensional feature set from the pool5 layer as output. To capture detailed semantics for finer-grained classification, the previous work [108] trained a set of independent part-based feature extraction side nets and cascaded these part features

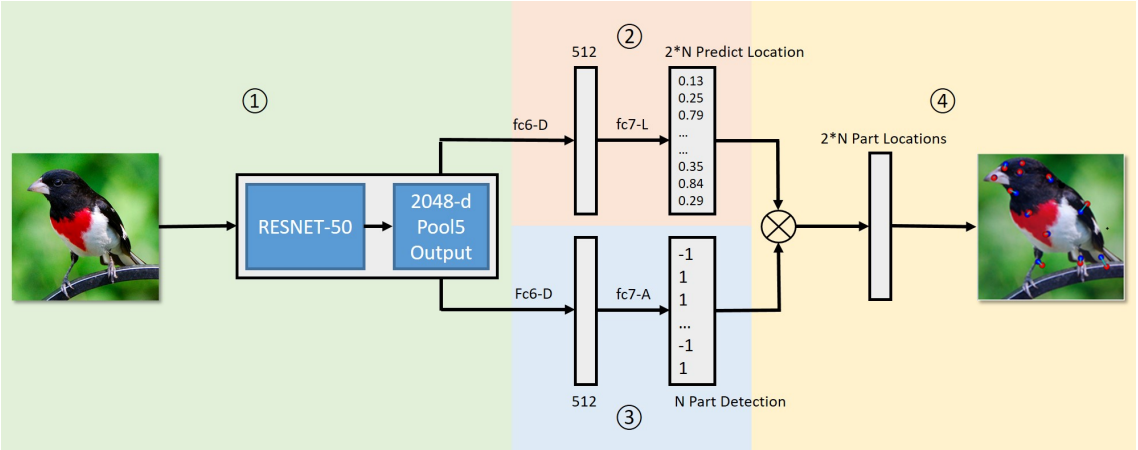


Figure 3.3: The localization network architecture. The model consists of: 1) image feature extraction network; 2) 2 fully connected layer part location prediction network; 3) 2 fully connected layer object part detection network; 4) unify part detection and predicted location output to generate the final localization.

with the global features to classify an object. In [45], the part level side net was combined with the object-level structure. However, only a unified part feature extractor for all different object parts was used, which means that the part classifiers are not concentrated enough comparing with the customized part streams in [108]. With the proposed method, customized side nets and a unified structure are introduced as the core of the proposed strategy. These side nets utilize finetuned part-level supervision to capture detailed semantics. As criticized in [45], independent feature extraction CNNs for multiple parts will lead to inapplicable high computation and memory cost. In the proposed framework, efforts are made to keep recognition performance while reducing the computation and memory cost of the customized part classifiers to a maximum extent.

The first step of the proposed method is to downsample the input resolution of part streams to 113×113 to reduce the calculation cost. In the meantime, the pool5 layer in each part stream was discarded to save $M \times 256 \times 3 \times 3$ feature map output. This design avoids fixed receptive field in [45] while providing more flexible part-cropping options. For example, we designed larger receptive field cropping for wings was designed with 25% of the bounding-box size and smaller ones for beaks (15% of the bounding-box size). This makes the part feature extractor more adaptable to scale changes.

AlexNet part stream K outputs $3 \times 3 \times 256$ features, and M part streams provide $M \times 3 \times 3 \times 256$ dimension part features for the classification net. For our general classification net structure, the major overhead is the massive parameters in the first

fully connected layer. For PC-CNN that involves more than ten parts, the first fully connected layer would contain over one million parameters, making both computation time and memory cost unacceptable. To address this problem, similar to [45], FCN is utilized to reduce the dimension of conv5 outputs of part streams. Therefore, the final output dimension of each part stream is only $3 \times 3 \times 16$, and the overall dimension of part steams is $M \times 3 \times 3 \times 16$.

3.1.3 Feature Unifying Layer and Classifier

After extracting feature maps at both the object and part levels, a unification layer is defined to cascade feature maps from different extractors. To handle this, output feature maps of different streams were put together outside the network [108]. The separation of the feature extraction and classification in the framework makes it harder to achieve global optimization. To solve the problem, a parameter sharing structure was proposed in [45] to extract detailed part feature maps with a shared side net and flatten feature maps as the input of 3 fully connected classification networks. In some instances, this solved the global optimization problem, but sharing the same set of parameters and computation by all parts raised the issue of under-customization as CNN finetuned at specific dataset outperforms general one.

We proposed three strategies on the unification layer to improve the performance of both computational complexity and the final classification accuracy. The first involves the pose information in the classification by simply taking a $2 \times M$ dimension (M point X-Y coordinates) pose vector as part of the input. The pose information is an instructive extra feature for fine-grained classification. The previous studies we have reviewed only utilized these coordinates to crop images/features from objects. The proposed structure takes the pose coordinates into the forward and backward propagation of the three fully connected layers.

The second is discarding blocking parts in both the forward and backward propagation. In the forward propagation, the unifying layer reads location vectors and checks whether a part is blocked or not. If it is blocked, then the data vector is set to 0. Similarly, if an object part is blocked, the difference vector is set to 0 for the backward propagation. This strategy effectively reduces redundant computation while minimizing the negative impact of the blocked parts.

Unlike the single-view classification in which over-fitting results are only from limited image samples, over-fitting in the multi-view classification also comes from limited part correlations in samples. Following the idea of dropping layer to avoid part over-fitting, we randomly discard some less discriminative parts in each sample.

Supposing the discriminative capability of each part can be represented by the top-1 accuracy of classification by the part alone, to measure the discriminative capability of each part, we use the CaffeNet model [52] as the feature extractor and train the last three fully connected layers for each part. As presented in Table 3.4, some parts such as the eye, crown, and beak produce higher prediction accuracies than others. The proposed framework randomly discards some of the existing part features by setting both the data and difference of the correspondent position to 0. The possibility $p_{discard}$ of a part being discarded depends on its top-1 prediction accuracy Acc_{part}:

$$p_{discard}(part) = \frac{\rho}{1 + e^{\sigma \times (Acc_{part} - \epsilon)}} \tag{3.5}$$

In practice we set ρ=0.8, σ=10 and ϵ=0.1. The unified feature vector is input into the classic three fully connected classification layers after sorting out in the unifying layer. In particular, we replaced the last original 1000-way fc8 layer with a customized 200-way fully connected layer to generalize the final classification output.

3.1.4 Training Methodology

Considering that the streams in the proposed PC-CNN are unbalanced in both representation capability and sample number as some parts are missing in some samples due to occlusion or camera views, training the entire PC-CNN from the beginning will cause under-fitting in some low representative part streams. Feature extraction nets should be trained separately and finetuned together. We propose a three-step training process for our PC-CNN structure.

Given data symbols D_G, D_L and D_i respectively stand for object-level image data, part localization vector data, and the i-th part image data. Parameters P_{net}, P_{netCNN}, and P_{netOrg} respectively stand for the parameters for network structure, convolutional part of the structure, and original model parameters pre-trained from the ILSVRC ImageNet dataset. ResNet, AlexNet, and PC-CNN$_n$ respectively stand for 50-layer ResNet[36], classic 8-layers AlexNet[52] and the proposed PC-CNN that consists of n part streams. Operator({D}, {P}) $\underrightarrow{\text{NET}}$ P_{save} stands for utilizing data/dataset {D} and pre-trained model P_{Org} to fine-tune network NET, producing and saving network model parameters P_{save}. The training process is demonstrated in Table 3.1.

A significant advantage of the proposed three-step training is that it can avoid the potential under-fitting of some weak representative part streams because the multiple-step training can avoid the conflict between different steams. In the meanwhile, it can keep the distinctive capabilities of each referring object part.

Step 1: fine-tuning object and part streams separately;

(D_G, P_{ResOrg}) $\xrightarrow{\text{ResNet}}$ P_{ResCNN}
for (i = 1; i < M; i++){
(D_i, $P_{AlexOrg}$) $\xrightarrow{\text{AlexNet}}$ $P_{AlexCNN}$
}

Step 2: fix P_{ResCNN} and $P_{AlexCNN}$, training conv6 of each part streams;

(D_i, $P_{AlexCNN_i}$) $\xrightarrow{\text{Sort by independent accurracy}}$ {D'$_i$, P'$_{AlexCNN_i}$}
for (i = 1; i < M; i++){
({D_G,D'$_1$,D'$_2$,...D'$_i$}, {P_{PC-CNN_i},$P_{AlexCNN_i}$}) $\xrightarrow{\text{PC-CNN}_i}$ P_{PC-CNN_i}
}

Step 3: fine-tuning entire network.
(D_G,D'$_1$,D'$_2$,...D'$_M$, P_{PC-CNN_M}) $\xrightarrow{\text{PC-CNN}}$ P_{PC-CNN_M}
Output: P_{PC-CNN}

Table 3.1: **The 3-steps training process for PC-CNN**

3.2 Experiment Result of PC-CNN

This section presents the experimental results of PC-CNN. Section 3.2.2 illustrates the localization accuracy of PC-CNN and presents the comparison of the PC-CNN with the previous PS-CNN. Section 3.2.3 compares the classification accuracy of the proposed PC-CNN and other state-of-the-art part-based methods. In Section 3.2.4, activated feature maps are visualized to validate the effectiveness of PC-CNN in extracting discriminative features.

3.2.1 Implementation Details

We carried out the experiments on the widely-cited fine-grained benchmark Caltech-UCSD birds (CUB-200-2011 [90]) dataset, which has been described in Section 2.5. The open-source package Caffe [46] was used to extract deep features and finetune our CNNs. For the localization network and the object stream of the classification network, we employed the ResNet-50 [36] model, which provided the least output feature-map dimension ($2048 \times 1 \times 1$), and was one of the best-performing structures overall. Meanwhile, to produce ancillary detail features for fine-grained classification, we took the Caffe reference AlexNet model in classification side nets in our proposed

part streams. The AlexNet model is almost identical to the model used by Krizhevsky et al. in [52].

3.2.2 Part Localization Result

According to [102][82], the localization accuracy of the proposed model is quantitatively assessed with three metrics: MRK (Mean Precision of Keypoints over images), MPK (Mean Recall of Keypoints over images), and APK (Average Precision of Keypoints). Following [102], an output location is considered correctly predicted if the prediction is within an Euclidean distance of 0.1 times the maximum width of the bounding box. Suppose that an image I in image set $\mathcal{I}$ contains n_{gt} ground-truth parts, and the proposed model predicts n_{pd} parts in which n_{tp} parts are correctly located, MPK and MRK are:

$$MPK = \frac{1}{N}\sum_{I\in\mathcal{I}}\frac{n_{tp}}{n_{pd}}, MRK = \frac{1}{N}\sum_{I\in\mathcal{I}}\frac{n_{tp}}{n_{gt}}, \tag{3.6}$$

By adjusting threshold $\delta \in (-1, 1)$ and observing the model performance, we can get a MPK-MRK function, $MPK = f_\delta(MRK)$, then APK is defined as follows:

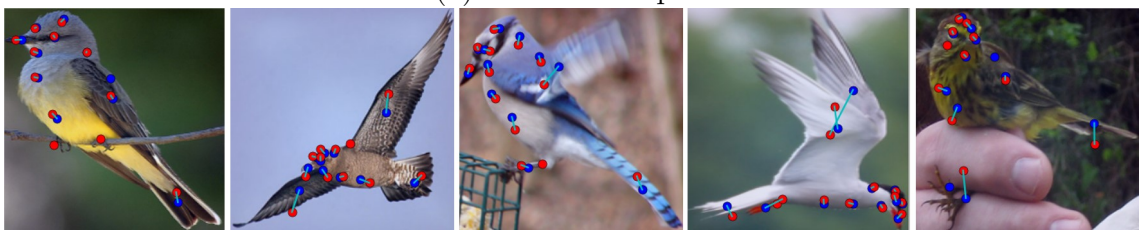
$$APK = \int_0^1 f_\delta(MRK)dMRK \tag{3.7}$$

The overall comparison of APK between the proposed PC-CNN and model presented in [45] is presented in Table 3.2. PC-CNN achieved impressive improvement over the recent state-of-the-art model by 5.8%. In our case, the δ is set to 0.12, and the MPK/MRK is 0.900/0.906, which is much higher than that in [45] (0.800/0.838). The localization result is presented in Figure 3.4. As Figure 3.4.(a) illustrates, the proposed model can effectively handle pose transmission, object part deformation, and slight occlusion. Moreover, in Figure 3.4.(b) we show several typical misprediction samples caused by mis-annotation, extreme deformation, or occlusion, through which we can see that the majority of essential parts are correctly located and the performance of the proposed model remains robust even in these extreme situations.

3.2.3 Categorization Results

To generate the categorization results, we firstly examined the extra discrimination of part streams. In step two, described in Table 3.1, we incrementally added the most discriminative object parts to the entire framework and iteratively trained the proposed model. The classification accuracy was recorded for each increment to observe

(a) Positive samples



(b) Negative samples

Figure 3.4: Localization samples: predicted part locations and ground-truth part locations are shown in red and blue dots, respectively, the green lines show their correspondences. (a) Correct localizations on different species and environments, and (b) Some representative samples mis-predicted by the proposed model.

Part	throat	beak	crown	forehead	r-eye	nape	l-eye	back
PS-CNN[45]	0.908	0.894	0.894	0.885	0.861	0.857	0.850	0.807
Ours	0.982	0.982	0.977	0.980	0.967	0.955	0.956	0.929
Part	breast	belly	r-leg	tail	l-leg	r-wing	l-wing	overall
PS-CNN[45]	0.799	0.794	0.775	0.760	0.750	0.678	0.670	0.866
Ours	0.942	0.956	0.815	0.836	0.783	0.859	0.820	0.924

Table 3.2: Comparison of APKs obtained from state-of-art methods on the CUB-200-2011.

FCN	BB	+1	+2	+3	+4	+5	+6	+7	+8	+9	+10
8	76.0	77.2	79	78.5	79.3	79.6	80.2	80.4	80.6	80.3	80.5
16	76.0	80.7	81.8	82.7	82.9	83.2	83.7	84.1	84.1	83.5	83.6
32	76.0	81.0	81.4	81.8	82.2	83.0	83.6	83.3	83.3	83.4	83.2

Table 3.3: The classification accuracy with different part number and FCN output dimension (tested with ground-truth part annotation).

Part	BB	Eyes	Beak	Crown	Forehead	Throat	Wing
Acc	63.50	56.20	36.40	35.20	34.30	31.20	26.00
Part	Breast	Nape	Belly	Back	Tail	Legs	
Acc	20.90	17.00	16.10	16.00	12.30	10.30	

Table 3.4: Part accuracy.

the contribution of the part streams. As shown in Table 3.3, the most discriminative parts (eyes, crown, beak, wings) can significantly improve the classification accuracy. While further increment of less discriminative parts also contributed to the gradual improvement of the classification accuracy, it soon became saturated. In the end, it was observed that the environmental noise in barely discriminative parts had a negative impact on the classification accuracy of the proposed PC-CNN model.

The impact of the conv6 output number was also evaluated. We respectively applied 8, 16, and 32 full convolution layer output feature maps on the proposed structure to determine the best practice for the low-level part feature extraction. Table 3.3 reveals the impact of part streams. BB stands for using bounding box only, +k stands for using the most discriminative parts besides BB, each row stands for using the outputs of 8, 16, 32 FCN. As shown in Table 3.3, further increasing the FCN output number to 32 did not improve the accuracy. However, it vastly increased the input dimension of the first fully connected layer, which was useful for explicit model interpretation.

In the two steps of the localization-classification scheme, we concentrated on the classification structure by utilizing the ground truth localization in our testing process. Table 3.4 compares the classification accuracies among the bounding-box level object and different parts alone trained and tested with the AlexNet model. Table 3.5 demonstrates the comparison of the proposed and other state-of-the-art models on the same dataset. Our proposed PC-CNN achieved a classification accuracy of 82.4%, which is inspiringly higher than state-of-the-art part-based methods, including the part-based RCNN [108] and the part-stacked CNN [45] by over 6%. Compared with the former state-of-the-art, Bilinear CNN (BCNN) [62], which used 448×448 input

Method	Training anno	Testing anno	Speed/fps	Para/million	Acc/%
Alignment[27]	BBox	BBox	—	—	67.0
No parts[67]	BBox	BBox	<1	135	74.9
ResNet[36]	BBox	BBox	130	25.5	76.2
Bilinear[62]	BBox	BBox	8	70	85.1
PR-CNN[108]	BBox+Part	n/a	—	—	73.9
PN-CNN[6]	BBox+Part	n/a	—	—	75.7
POOF[4]	BBox+Part	BBox+Part	—	—	73.3
PC-CNN	BBox+Part	BBox+Part	73	92	**84.1**
POOF[4]	BBox+Part	BBox	—	—	56.8
DeCAF[17]	BBox+Part	BBox	—	—	65.0
PR-CNN[108]	BBox+Part	BBox	<1	60	76.4
PS-CNN[45]	BBox+Part	BBox	20	130.5	76.2
PC-CNN	BBox+Part	BBox	47	120	**82.4**

Table 3.5: Comparison of classification accuracies among the proposed PC-CNN and other state-of-the-arts. BBox and Part stand for using ground-truth bounding box and part location annotation, respectively.

image resolution, the PC-CNN only use 1/4 of the input resolution (224×224) and hence is nearly six times faster (8 fps vs. 47 fps), to achieve a comparable classification accuracy.

3.2.4 Discussion on the Proposed PC-CNN

For CNN-FCN classification schemes, the FCN structure largely determines the memory and calculation cost of the entire network model. A comparison was conducted to evaluate the parameter number and testing speed of some well-known structures and the proposed method, as shown in Table 3.5. By applying the ResNet structure to reduce the input parameter number of the fully connected layers, an M-part PC-CNN updates only $49.5 + 2.92 \times M$ millions of parameters, which is much less than the AlexNet based PS-CNN structure ($60.02 + 4.72 \times M$ millions). At the localization stage, our proposed 15 part PC-CNN achieved 47 fps on TITAN X (Pascal) GPU, which is marginally slower than the basic ResNet-50 model; however, the classification performance has been significantly improved by about 6.2%.

Instead of sharing a feature extractor for all parts as presented in [45], our proposed model uses separated part streams. We particularly compared the activation of the conv5 output of an image sample from the CUB-200-2011 dataset. We also applied several independent AlexNet feature extractors in the proposed framework to produce the activation heatmap for each part. On the contrary, a shared feature ex-

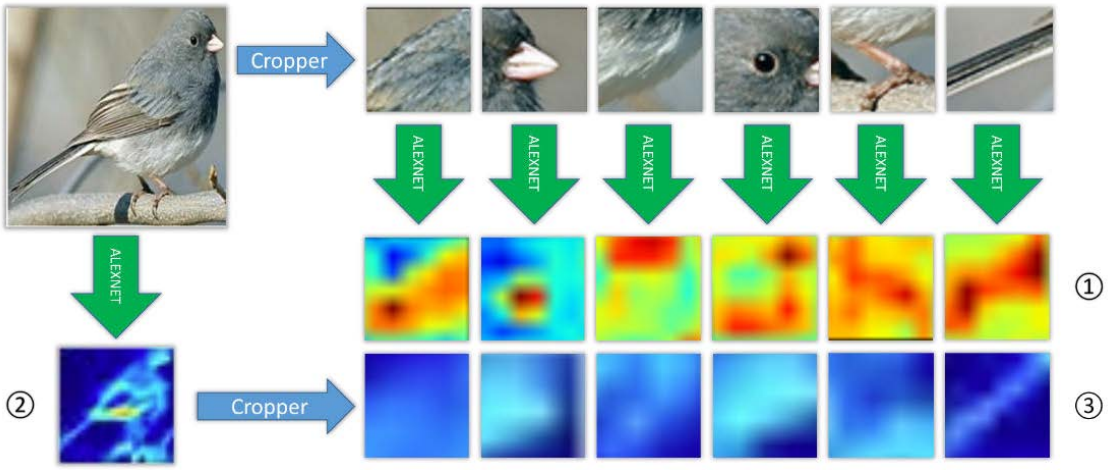


Figure 3.5: Comparison among the output heatmaps of independent and shared parameter CNNs. (1) the maximal activation maps from the outputs of part streams; (2) the maximal activation map from the outputs of object stream; (3) the heatmaps of different parts cropped from (2).

tractor was employed in [45], and part features were cropped from output heatmaps. Figure 3.5 explains that shared parameters lead to mediocre discrimination of part features. In contrast, customized feature exactors for each stream are more sensitive to discriminative parts like feather textures, beak outlines, or tail shapes in part images.

3.3 Conclusion

In this chapter, a novel PC-CNN model for fine-grained classification was proposed and presented. By defining separated part side nets, the proposed model introduces detailed part-level representations into the classification. This has led to significant improvement of the fine-grained classification on the benchmark CUB-200-2011 dataset. Examining the output feature maps of each part stream reveals that independently trained part feature exactors can produce more meaningful activation and discriminate part differences among different species with subtle appearance variances. By applying ResNet and 1×1 convolution layer to reduce the input dimension of the FCN classifier, we reduced the model size to a level similar to or smaller than most of the commonly used networks.

The future work of the PC-CNN will focus on an end-to-end localization-classification strategy. Based on the predefined skeleton of an object, a deep network should have the capability to learn the tridimensional structure of the object and find discrimi-

native parts automatically. In this way, we may even catch some hidden biological characteristics still undiscovered by human biologists.

Chapter 4

Squeezed Bilinear Pooling

Bilinear pooling was first proposed to address the challenge of Fine-Grained Visual Classification (FGVC) by Lin et al. [62]. Based on the bilinear pooling, Lin et al. [61] investigated matrix square-root normalization to significantly improve the representation of the bilinear feature. However, a neglected problem of the above feature encoding method is its extremely high output feature dimension. The tensor product makes c CNN output channels to c^2 dimension of pooled features. A relatively low $c = 512$ VGG-16[80] structure produces a $512 \times 512 \approx 262k$ dimension bilinear features. To deal with this problem, Tensor Sketching was investigated in [25] and similar accuracy was reported with 8K compact features. However, the linear combination significantly increases the computational complexity of bilinear features. To solve the computation dilemma, a low-rank approximation-based method was proposed in [50] and obtained a similar performance of the original full bilinear pooling. Given these methods reduced the dimension and computational complexity by two orders of magnitude, one vital problem is that the matrix power function cannot propagate through the compact layer. Sub-normalization was employed in [34] to solve the problem. However, the performance is not as good as expected since the categorization accuracy drops around 1% compared with that obtained with the baseline structure. It remains a problem to combine a compressed bilinear structure with matrix power normalization.

This chapter proposes a novel efficient module for the fine-grained visual classification, named Iterative Fisher Bilinear Pooling (IFBP). The proposed module is a pooling layer inspired by the Bilinear CNN (BCNN) [62], and its variants CBP [25] and LRBP [50]. Unlike the previous works, the proposed IFBP can directly replace the global average pooling in a range of deep CNN structures to improve transfer learning capability while keeping the computation cost and memory consumption

	Fast Computation	Matrix Normalization	Attention Interpretable
CBP	×	×	×
LRBP	√	×	×
MoNet	×	√	×
SBP	√	√	√

Table 4.1: Comparison on the proposed SBP and other compressed bilinear pooling based methods.

low. The advantages of the proposed model over the other compact bilinear models are illustrated in Table 4.2.

The highlights of this chapter include:

- With the proposed IFBP, we investigated the distribution of the bilinear features. We compressed the bilinear features with Fisher Selector (FS), which outputs a second-order feature with a much lower dimension, and at the same time, linearly reduces the computational complexity.
- With the improved version of Squeezed Bilinear Pooling (SBP), Matrix Normalization (SBP-MN), we designed a backpropagation procedure for the matrix normalization of the selected features.
- We proposed an SBP-based recurrent attention structure, called Fisher Recurrent Attention Squeezed Bilinear Pooling (FRA-SBP), which significantly improved the categorization accuracy on the common FGVC datasets.

The work of this chapter has been published in the proceedings of the 2019 IEEE International Conference on Computer Vision Workshops [57].

4.1 Squeezed Bilinear CNN

In this section, we describe a novel bilinear-based supervised compressed method. The overview of the proposed SBP architecture is shown in Figure 4.1. For each input image I, the convolutional neural network outputs a feature matrix $M = \{m_1, m_2 \ldots, m_c\}$, where m_i is the expansion tensor of the i-th channel of CNN features. Following [62], the co-inner production is conducted on M to produce $c \times c$ second order map $\hat{M} = \{\hat{m}_1, \hat{m}_2, \ldots, \hat{m}_{c^2}\}$. After that, the novel Fisher Selection Layer (FSL) and Global Average Pooling (GAP) are applied to generalize the d-dimension

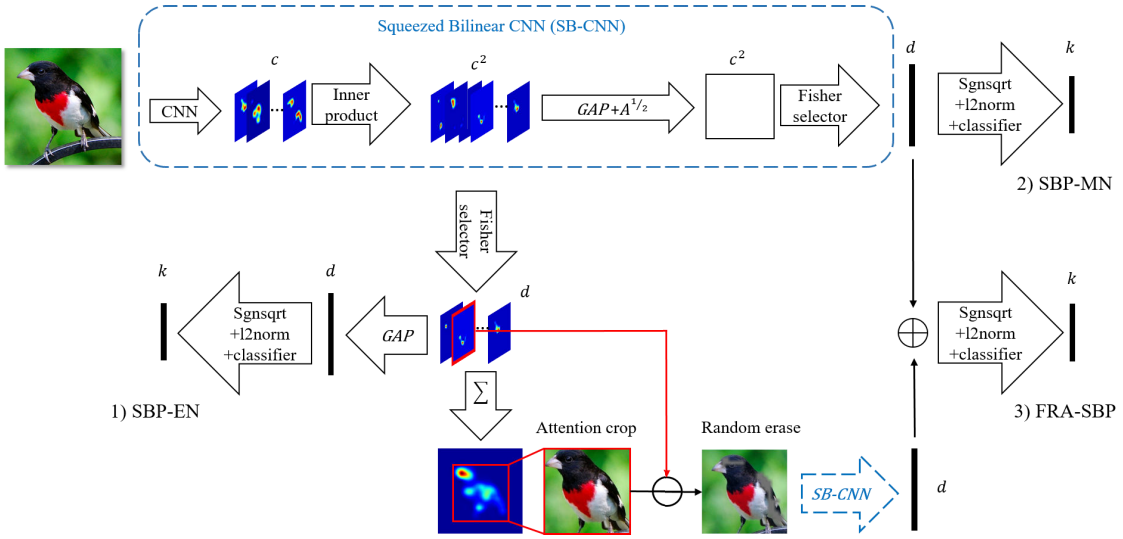


Figure 4.1: The proposed network architecture with three SBP based models: 1) the Squeezed Bilinear Pooling with Element-wise Normalization (SBP-EN) for fast computation, 2) the Squeezed Bilinear Pooling with Matrix Normalization (SBP-MN) by inserting the matrix square root function before the squeezing layer, and 3) the Fisher Recurrent Attention Squeezed Bilinear Pooling (FRA-SBP).

	Full Bilinear	CBP/MoNet_TS	LRBP	SBP-EN
Feature Dim.	c^2 [262K]	d [10K]	m^2[10K]	d [10K]
Feature Comp.	$O(hwc^2)$	$O(hw(c+dlogd))$	$O(hwm(c+m))$	$O(hwd)$
Classify Comp.	$O(Kc^2)$	$O(Kd)$	$O(Khwrm)$	$O(Kd)$
Feature Para.	N/A	$2c$ [4KB]	cm [200KB]	d [20KB]
Classifiy Para.	Kc^2 [200MB]	Kd [6.4MB]	krm [600KB]	Kd [6.4MB]

Table 4.2: Comparison on feature dimensions, computations, and the parameter numbers for different pooling methods and classification layers.

squeezed second-order feature vector, followed by the off-the-shelf element-wise square root regularisation, l_2-normalization layers, and a full connected classification layer.

In the parallel workflow, a matrix square root layer is applied as a bridge between the full bilinear layer and the FSL to improve the proposed SBP further. The two flows are named as Squeezed Bilinear Pooling with element-wise (SBP-EN, as illustrated in Figure 4.1.(1)), and matrix normalization (SBP-MN, as shown in Figure 4.1.(2)), respectively. Based on the single model approaches, we designed the Fisher Recurrent Attention SBP (FRA-SBP, shown in Figure 4.1.(3)). We will discuss these structures in detail in the following sections.

4.1.1 Fisher Feature Selector

Fisher Discriminant Analysis (FDA) has been widely investigated [3][69][23]. It discriminates patterns using the low-dimensional projection of high-dimensional features with linear transformations. Its main idea is to maximize the inter-class variations and minimize intra-class variations. Although Fisher's discriminant analysis does not perform feature selection projection from high dimensional to low dimensional space via linear transformation, it maximizes the class separation measurement via the feature selection.

Denote the intra, inter-class, and total scatter matrix by S_w, S_b, and S_t:

$$\begin{aligned} S_w &= \frac{1}{n}\sum_{j=1}^{g}\sum_{i=1}^{n_j}(x_i^{(j)}-\mu_j)(x_i^{(j)}-\mu_j)^T, \\ S_b &= \frac{1}{n}\sum_{j=1}^{g} n_j(\mu_j-\mu)(\mu_j-\mu)^T, \\ S_t &= \frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)(x_i-\mu)^T = S_w + S_b, \end{aligned} \tag{4.1}$$

where $x_i^{(j)}$ represents the i-th feature observation in Class C_j, μ_j and μ are sample means for Class C_j and the hole dataset, respectively, i.e. $u_j = \frac{1}{n_j}\sum_{i=1}^{n_j} x_i^{(j)}$, and $\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$.

Fukunaga [23] proved that the traces of these scatter matrices could be used to measure the class separation of the features. The scatter measurement using means and variances is based on an implicit assumption that the features in each class follow a normal distribution. For large data machine learning cases, it usually works well [89]. To address the linearly non-separable problem, non-linear kernels are incorporated into the above scatter matrices. Note that $\phi(\cdot)$ is the kernel that maps the original feature space R^p into the objective kernel space R^d:

$$R^p \xrightarrow{\phi} R^d, \phi(x) = z, \tag{4.2}$$

the inner product in the kernel space becomes $\langle\phi(x_1, x_2)\rangle = k(x_1, x_2)$, where $k(\cdot,\cdot)$ is the kernel function [89]. After mapping into and calculating the scatter matrices in the kernel space, the trace of the intra, inter-class, and total scatter matrices in

kernel space, i.e., $\tilde{S}_w$, $\tilde{S}_b$, and $\tilde{S}_t$ can be obtained as:

$$\begin{aligned} Tr(\tilde{S}_w) &= \frac{1}{n}Tr(K) - \frac{1}{n}\sum_{i=1}^{g}\frac{1}{n_i}Sum(K^{(i)}), \\ Tr(\tilde{S}_b) &= \frac{1}{n}\sum_{i=1}^{g}\frac{1}{n_i}Sum(K^{(i)}) - \frac{1}{n^2}Sum(K), \\ Tr(\tilde{S}_t) &= \frac{1}{n}Tr(K) - \frac{1}{n^2}Sum(K), \end{aligned} \tag{4.3}$$

where the operators $Sum(\cdot)$ and $Tr(\cdot)$ calculate, respectively, the summation of all elements and the trace of a matrix, and K and $K^{(i)}$ are the $n \times n$ and $n_i \times n_i$ sized matrices defined by:

$$\{K\}_{kl} = k(x_k, x_l), \{K^{(i)}\}_{uv} = k(x_u^{(i)}, x_v^{(i)}). \tag{4.4}$$

The feature selector is denoted by $\alpha = [\alpha_1, ..., \alpha_p]^T \in \{0,1\}^p$ with $\alpha_k = 1$ indicating that the k-th feature is selected or 0 not-selected, $k = 1, \ldots, p$. Then the selected feature set from the original feature vector x is given by:

$$x(\alpha) = x \odot \alpha, \tag{4.5}$$

where the operator $\odot$ performs the Hadamard product, with the feature selector, K and $K^{(i)}$ become the function of α:

$$\begin{aligned} \{K(\alpha)\}_{kl} &= k(x_k \odot \alpha, x_l \odot \alpha), \\ \{K^{(i)}(\alpha)\}_{uv} &= k(x_u^{(i)} \odot \alpha, x_v^{(i)} \odot \alpha), \end{aligned} \tag{4.6}$$

for $k, l \in \{1, ..., n\}$, $u, v \in \{1, ..., n_i\}$, and $i = 1, ..., g$. Similarly, the trace of the scatter matrices are noted by $Tr(\tilde{S}_w)(\alpha)$, $Tr(\tilde{S}_b)(\alpha)$, $Tr(\tilde{S}_t)(\alpha)$. To maximize the class separation, the objective function of optimization can be formulated as:

$$\arg\max_{\alpha \in \{0,1\}^p} \{Tr(\tilde{S}_b)(\alpha) - \lambda Tr(\tilde{S}_t)(\alpha)\}. \tag{4.7}$$

To handle the robustness issues with the linearly non-separable and highly noisy datasets, especially for extremely high dimensional covariance bilinear features in our case, the l_0-norm constraint is utilized in the feature selector to regularize the discrimination measure [21][96]. Hence, the objective function is modified to:

$$\arg\max_{\alpha \in \{0,1\}^p} \{Tr(\tilde{S}_b)(\alpha) - \lambda Tr(\tilde{S}_t)(\alpha) - \beta\|\alpha\|_0\}. \tag{4.8}$$

The parameters λ and β are constraint factors, and their ranges and the optimal values were discussed and experimented in detail in [11]. In the Fisher selection layer

in our squeezed bilinear model, we directly use the value provided in [11] with the best performance on other machine learning-based classification tasks.

Substitute the scatter matrix expression (4.3) into the object function (4.8), we can get the final Fisher feature selector objective function:

$$\begin{aligned}\arg\max_{\alpha\in\{0,1\}^p}\{\frac{1}{n}\sum_{i=1}^{g}\frac{1}{n_i}Sum(K^{(i)}(\alpha)) - \frac{\lambda}{n}Tr(K(\alpha)) \\ +\frac{(\lambda-1)}{n^2}Sum(K) - \beta\|\alpha\|_0\}.\end{aligned} \tag{4.9}$$

The general Fisher selector formulation is a combinatorial optimization problem for many kernels, it is far from feasible in our bilinear features whose p is up to over 262K. Surprisingly, the polynomial kernel provides efficient and global optimization for (FS) for large p [89][76][88]:

$$k(x_1, x_2) = (1 + \langle x_1, x_2\rangle)^D \tag{4.10}$$

By incorporating the feature selector α with the degree parameter $D = 1$, the kernel becomes:

$$\begin{aligned}k_1(x_1, x_2)(\alpha) &= 1 + \langle x_1 \odot \alpha, x_2 \odot \alpha\rangle \\ &= 1 + \sum_{i=1}^{p} x_{1i}x_{2i}\alpha_i \\ or \quad k_1'(x_1, x_2)(\alpha) &= \sum_{i=1}^{p} x_{1i}x_{2i}\alpha_i\end{aligned} \tag{4.11}$$

Substitute the $k_1(\cdot,\cdot)$ or $k_1'(\cdot,\cdot)$ into the objective function (4.9), we can get the Fisher discriminative score:

$$\begin{aligned}\theta_j =&\frac{1}{n}\sum_{i=1}^{g}\frac{1}{n_i}\sum_{u,v=1}^{n_i} x_{uj}^{(i)}x_{vj}^{(i)} \\ &-\frac{\lambda}{n}\sum_{i=1}^{n}x_{ij}^2 + \frac{(\lambda-1)}{n^2}\sum_{u,v=1}^{n}x_{uj}x_{vj}\end{aligned} \tag{4.12}$$

and considering of the preset d object dimension of the squeezed bilinear model, the Fisher optimization objective function can be depicted as follows:

$$\begin{aligned}&\arg\max_{\alpha\in\{0,1\}^p}\sum_{j=1}^{p}(\theta_j - \beta)\alpha_j, \\ &s.t. \|\alpha\|_0 = d.\end{aligned} \tag{4.13}$$

With the first-order polynomial kernel, the globe Fisher score θ_j only varies for the observation of the j-th feature dimension. For a given λ, the optimization process

can be transformed into calculating the θ_j for each bilinear feature dimension, and select d largest scores θ_J:

$$j \in J \longleftrightarrow \alpha_j = 1 \tag{4.14}$$

This is the globally optimal solution, and the computational complexity for the calculation of α with n training samples and p feature dimensions is $O(n^2p)$.

4.1.2 Squeezed Bilinear CNN

With the full bilinear feature $M = X^T X \in R^{c^2}$, the Fisher selector α, and the objective projection dimension d, the projection function for the Fisher selection layer can be represented as:

$$R^{c^2} \xrightarrow{\psi} R^d, \psi(M) = M \circ \alpha, \tag{4.15}$$

where the operator $(a \circ b)$ requires the same size of the tensors a and b, aiming at extracting the a values of position that is not "0" in b, to form a new d dimensional feature tensor ($\|a\|_0 = d$). Note that the calculation of each element in the bilinear features is independent. Discarding the less discriminative features can lead to faster computation. By combining the bilinear pooling layer and the Fisher selection, named as squeezed bilinear pooling, we can directly obtain $x_i \cdot x_j$where $\alpha_(\tau(i,j)) \neq 0$ ($\tau(i,j)$ is the corresponding transformed position of the inner product of the i-th and j-th CNN channels in the bilinear feature tensor. The computational complexity for a d dimensional squeezed bilinear feature with CNN feature maps of size $h \times w \times c$ is $O(hwd)$. It is linear only with the compact feature size d, which means that our proposed squeezed bilinear pooling can be efficiently implemented into larger-scale CNN structures such as ResNet and DenseNet with 2048 and 1024 CNN output channels respectively.

The backpropagation is the converse process of the forwarding (4.15). The propagation function can be written as:

$$\frac{\partial L}{\partial \psi(M)} = \frac{\partial L}{\partial M_i \circ \alpha_i}. \tag{4.16}$$

The operator $\circ$ is not differentiable but is an element-wise first-order linear combination, hence it can be solved by the combination of element-wise derivatives. For each $\alpha_i \neq 0, \rho(i)$ is the projected index of i by the selection function ψ (if projected), the backpropagation function can thus be depicted as below:

$$\frac{\partial L}{\partial M_i \alpha_i} = \frac{\partial L}{\partial \psi(M)_{\rho(i)}}. \tag{4.17}$$

For selected elements, $\alpha_i = 1$, we can directly pass the back gradient to the corresponding channels of CNN feature maps.

The implementation of the matrix power normalization $M' = M^{1/2}$ requires a positive definite forward input feature matrix, and a symmetric backward gradient matrix [61], which makes it not practical to be embedded into the linear combination based compact bilinear structures [61][50][34]. As shown in Figure 4.1, the SBP-MN structure satisfies the first requirement by inserting the matrix normalization layer as a bridge between the full bilinear pooling layer and the Fisher selection layer. The second requirement can be met by a matrix diagonalization. Suppose $\delta(i)$ is the diagonal position of index i in the bilinear feature matrix M, we can describe the improved backpropagation function for matrix power normalization as follows:

$$\frac{\partial L}{\partial M'_i \alpha_i} = \frac{\partial L}{\partial M'_{\rho(i)} \alpha_i} = \frac{1}{2}\left(\frac{\partial L}{\partial \psi(M')_{\rho(i)}} + \frac{\partial L}{\partial \psi(M')_{\rho(\delta(i))}}\right) \tag{4.18}$$

This can promise the convergence of Newton's iteration in [61]. But due to the necessity of a positive definite full bilinear feature matrix for SVD in the matrix power function, the computational complexity is increased back to $O(hw(c^2))$. The memory and computation cost of SBP with/without the matrix normalization are compared with other compressed structures in Table 4.2. With the same encoded feature dimension, the proposed SBP-EN can reduce the computational complexity by magnitude. Despite that the SBP-MN requires a similar computation as the Full Bilinear Pooling, it uses matrix normalization to enhance the classification accuracy, which does not apply to CBP and LRBP. SBP-MN and MoNet_TS require, on the same order of magnitude, as much of computation and memory resources, their classification performance will be discussed in Section 4.2.3.

4.1.3 Fisher Recurrent Attention Structure

Recently, lots of research on fine-grained classification focus on the recurrent attention structures [22][112][43]. Noting that the Squeezed Bilinear Pooling selects the most discriminative second order features, it is also an ideal method to localize the object region and suppress the irrelevant background activation. As illustrated in Figure 4.1.(3), the d selected second order feature maps $\{m_1, m_2, ..., m_d\} \in \psi(M)$ are element-to-element summed and produce the average activated map m_a. Then the m_a is linearly resized to input image size, and its values are normalized into the range of $(0, 1)$, called the normalized attention map m_n. A threshold ε is applied to m_n to segment the attention activation map m_s:

$$m_s = m_n > \varepsilon. \tag{4.19}$$

Following the method of [95], we randomly select an second order map $m_k \in \psi(M)$, and erase the activated region of m_k from m_s to obtain the attention erase map m_e:

$$m_e = (m_k == 0) \cdot m_s. \tag{4.20}$$

The attention erosion is conducive to learn from global discriminative object parts [95] and avoid overfitting. After the random erosion, we crop the TRUE region of m_e from the input image I to create the attention image I_a. I_a is recurrently inputted into the SB-CNN and outputs the d dimensional attention feature f_a. The features of the two stages, f and f_a are cascaded, and after sgnsqrt and l_2 normalization layer, are classified with a fully connected layer. The feature dimension of FRA-SBP is 20,000. The computational complexity of the FRA-SBP model is the summation of two SBP-MN models and the computation of the cropping processing. Compared with the convolutional computation, the cropping is negligible, for which we can consider the FRA-SBP has two times of computational complexity of SPB-MN.

4.2 Experimental Results of SBP

In this section, we detail our experiments on the proposed SBP-CNN in three aspects. (1) In Section 4.2.2, we investigate the impact of the most determinative parameter, the selected dimension number, on the proposed squeezed bilinear method and compare it with the commonly used compact bilinear pooling under different dimensions. (2) In Section 4.2.3, we conduct an overall comparison of our proposed squeezed models against other methods on a variety of fine-grained datasets. (3) In Section 4.2.4, we look into the activation of different channels and their covariances to verify the effectiveness of the Fisher score and the squeezed function for bilinear features. To begin with, we provide experiment details in Section 4.2.1.

4.2.1 Implementation Details

In the experiments, We only used the category label without any part or bounding box annotation provided by the datasets when training all models. When evaluating our model on the VGG-16[80] structure (D-net in [62][25][61]), we used the first 30 layers of VGG-16 as the local feature extractor and retained the output of the Conv5_3+ReLU layer for the second-order encoding, as conducted in [62]. We then

compared the proposed squeezed bilinear pooling with full bilinear and other compact feature encoding methods. All experiments are conducted using Caffe [46] platform and implemented in Python. We resized the input images to the dimension of 512×512 and randomly cropped them to the size of 448×448 for training in all the second-order models. The training batch size was set to 8, and the weight of decay is 5×10^{-6}. We conducted the training in two steps: the first step was to fine-tune the last fully connected classification layer, and the second step is to fine-tune the whole network with CNN to obtain the final model. For FSL, λ was assigned a value of -0.5, and for attention cropping in FRA-SBP, the value of ε was assigned to 0.005.

4.2.2 Configuration and Comparison with Compact Bilinear

The selected dimension d of the squeezed bilinear pooling can be adjusted manually. We conducted experiments with the selected dimension range of 100 to 15,000 on the CUB-200-2011 dataset to investigate the potential impact of the selected dimension number on the squeezed bilinear pooling. To choose the proper dimension needed for SBP, we used a convolutional net of VGG16 as the backbone of the proposed approach. We also compared the performance of SBP with CBP. For fairness, we did not utilize the matrix normalization in this section to compare the representation capabilities of different compression functions. Except for the pooling method and the projected dimension mentioned in the experimental results, the other components of the network structure and training configurations are the same for all models.

As summarised in Figure 4.2, with the increasing of projected dimension, the top-1 errors of both the CBP [25], and the SBP come down to a similar level of the full bilinear. The difference after fine-tuning the whole network is less than 0.2% with over 10K projected dimension. When only the last classification layer is fine-tuned, the squeezed bilinear can achieve a slightly lower error rate than the full bilinear and compact bilinear. We credit to the discarding of the redundant non-semantical high dimensional covariances.

With a lower dimension, the performance of SBP is more promising than the Tensor Sketch. Without fine-tuning, SBP outperforms CBP by around 1.5% when the dimension ranges from 1K to 5K, and the gap widens as the dimension decreases. With 500 projected dimensions, a significant disparity can be observed between the two methods, 4.6% without and 2.6% with fine-tuning. In deficient dimension cases, e.g., 100, SBP produces acceptable accuracy loss (22.7%) comparing with CBP (42.8%). This makes sense for a vast area of applications. For example, large-scale fine-grained

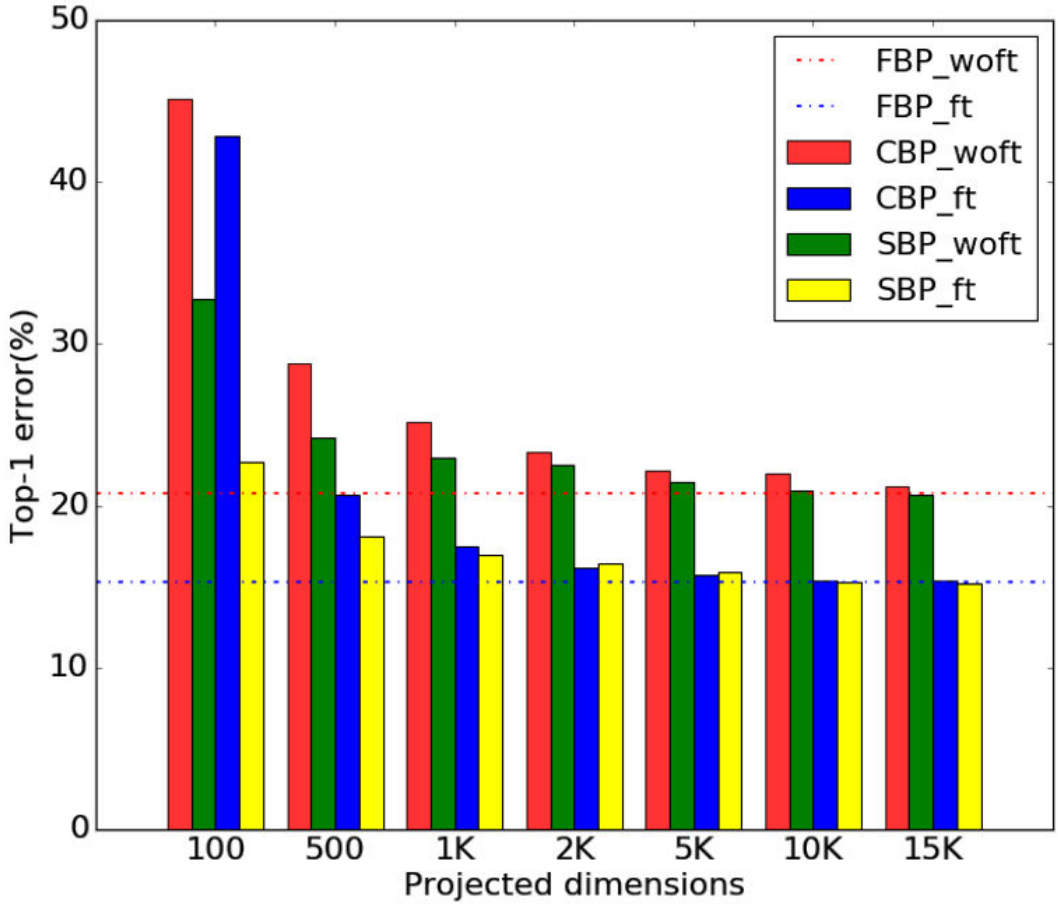


Figure 4.2: Classification error rate on the Cub dataset. Comparisons are made on the proposed SBP without matrix normalization and Compact Bilinear Polling (CBP) with Tensor Sketch. Horizontal lines are the baseline performances of Fully Bilinear Pooling (FBP). ft and woft stands for with and without global fine-tuning of CNN, respectively.

	Method	Dim.	Mul.	Cub	Airplane	Car
Bilinear & Improved Versions	BCNN[27]	262K	205M	84.0	86.9	90.6
	iBCNN[61]	262K	205M	85.8	88.5	92.0
	G2DeNet[93]	263K	206M	87.1	89.0	92.5
	MoNet[34]	263K	206M	86.4	89.3	91.8
Compressed	CBP_TS[25]	8.2K	105M	84.0	87.2	90.2
	LRBP[50]	10K	48M	84.2	87.3	90.9
	SMSO[104]	2K	—	**85.0**	—	—
	MoNet_2U_TS[34]	10K	105M	**85.0**	86.1	89.5
	SBP-EN	10K	**7.8M**	84.6	**87.8**	**90.9**
Compressed +Matrix Normalization	MoNet_2_TS[34]	10K	105M	85.7	86.7	90.3
	MoNet_TS[34]	10K	105M	85.7	88.1	90.8
	SBP-MN	10K	205M	**86.1**	**89.2**	**91.6**
	FRA-SBP	20K	—	**86.8**	**90.4**	**93.2**
Other SotAs	KP[13]	14.3K	420M	86.2	86.9	92.4
	BoostCNN[71]	—	—	86.2	88.5	92.1
	PC[19]	—	—	85.6	85.8	92.5
	iSQRT-COV[55]	—	—	**87.2**	90.0	92.5
	MA-CNN[112]	—	—	86.5	89.9	92.8

Table 4.3: Comparison of the classification performance among the proposed SBP based models and other methods on various FGVC datasets. From top to bottom, the four blocks respectively list fully bilinear based methods, compressed bilinear methods, compressed structures with matrix normalization, and other state-of-the-art methods obtained on multi-datasets. Dim. and Mul. stand for feature dimension and multiplies required for pooling, respectively. The results of the baseline methods are duplicated from the original papers.

image retrieval [92] often needs quick, high representative but low dimensional image features.

The experiments suggest that while the proposed squeezed bilinear pooling linearly reduces the feature dimension and computational complexity, it can provide a reliable approximation of the full bilinear pooling with 262K dimension of output features. For FGVC, 10K selected features performed well. The experimental results also show that SBP significantly outperforms CBP, especially with low dimension constraints.

4.2.3 Experiments with Different Datasets

We compared the proposed squeezed bilinear pooling against other approaches for the categorization of the following commonly used FGVC datasets: CUB-200-2011 [90], FGVC-Aircrafts [66], Stanford Cars [51]. The experimental results are summarised

in Table 4.3, including four parts. The first part adduces results from the original [62] and the improved versions of bilinear, including iBCNN [61] and the first-order embedded G2DeNet [93] and MoNet [34]. The second part lists the results using the compressed bilinear features, including CBP [25], LRBP [50], SMSO [104], Monet with Tensor Sketch but without the first-order embedding (MoNet_2U_TS) [34], and the proposed SBP with element-wise normalization (SBP-EN). Both utilizing compressed structure and matrix normalization, the proposed SBP with matrix normalization (SBP-MN), the derived Recurrent Attention SBP (RA-SBP), and competitive Monets are shown in part three. The other state-of-the-art methods, including the boostCNN [71], KP [13], Pairwise Confusion(PC) [19], iterative matrix square root normalization of covariance pooling (i-SQRT-COV) [55] and attention-based multi-model MACNN [112], despite not exactly in the same research route, are shown in the part 4. Note that for fairness, we only compare the results reported on the backbone of VGG-16.

From Table 4.3, we can see that the fast version of the proposed SBP, SBP-EN obtained the best accuracy with the planes and cars dataset and achieved an accuracy of 84.6% on CUB-200-2011, only 0.4% lower than the recent proposed MoNet [34]. However, our SBP requires only 24% of the computation required for the Tensor Sketching in Monet. Comparing with CBP [25] and LRBP [50] with the same dimension, the accuracy of the proposed SBP-EN is around 0.5% higher on average on the three datasets.

In the other aspect, the matrix square root is not directly applicable to CBP [25] and LRBP [50]. Hence, comparing the compressed structures with matrix normalization was conducted between the SBP-MN and Tensor Sketching Monet with and without the first-order information. SBP-MN without the first-order information outperforms MoNet_TS [34] by 0.4% to 1.0% on the three fine-grained datasets. When comparing with MoNet_2_TS [34], the classification accuracy of the proposed SBP-MN is 0.4% to 2.4% higher. To the best of our knowledge, the performance of our SBP-MN model is state-of-the-art among all compressed bilinear models on these datasets.

The accuracy of the proposed FRA-SBP is 0.4% lower than the state-of-the-art fine-grained model, iSQRT-COV [55] on CUB-200-2011 dataset, but around 0.5% higher on the other two datasets. Note that iSQRT-COV needs to pre-train on ImageNet [15], while the FRA-SBP model achieved the overall better accuracy with a transferred model. Comparing with other duplicate or recurrent models, e.g. BoostCNN [71], KP [13] and MA-CNN [112], the accuracy of the FRA-SBP is 0.2% to 3.5% higher on the three datasets.

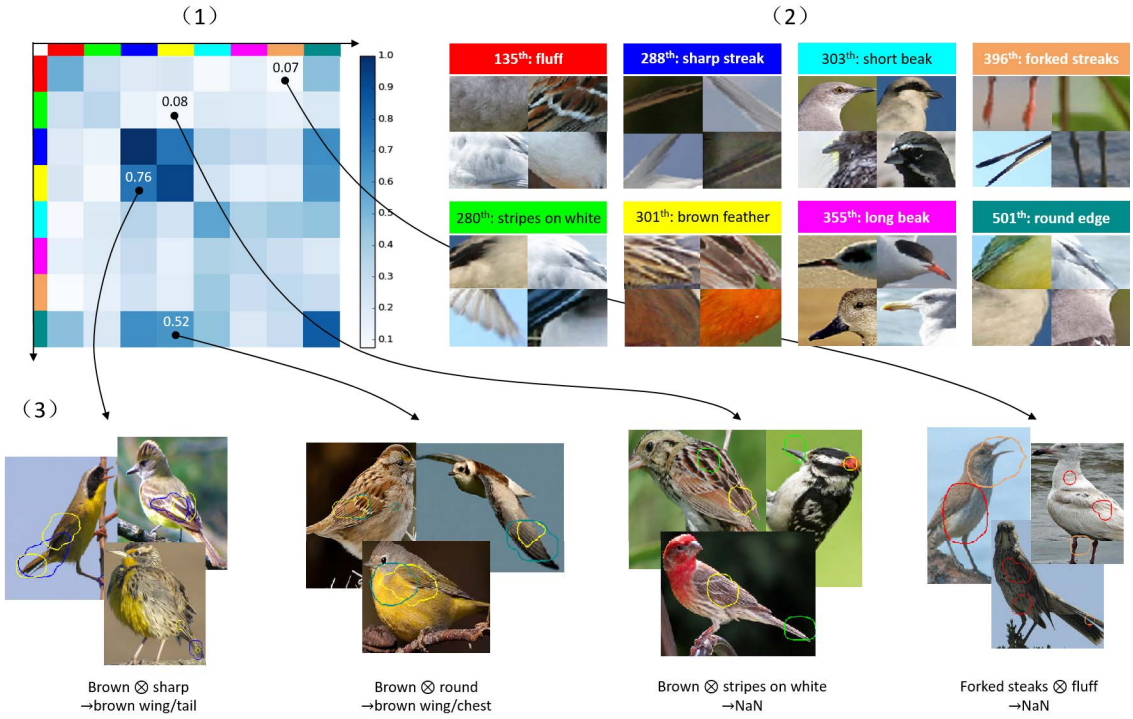


Figure 4.3: Visualization of Squeezed Bilinear Pooling and its activation across CNN channels. 1) shows the normalized Fisher scores for the inner products of the eight most activated channels. 2) crops the activated regions of the channels and abstracts the semantemes. 3) marks overlapped regions of activations across channels to verify the effectiveness of the Fisher score measurement.

4.2.4 Validation via Visualization

We visualized the output of CNN channels, and the inner products cross them to see what happens to features with different ranges of Fisher scores. As shown in Figure 4.3, we chose the eight most covariant channels with the largest mean Fisher score from all other channels, and they are labeled with the colors of red, green, blue, yellow, cyan, magenta, brown, and dark cyan. The activations of these channels for different classes of images were observed to abstract the semantic representation. For example, the $288th$ channel, marked as blue, was mostly activated to sharp streak on wings or tails; and the $301st$ channel, marked as yellow, activated to brown feather regions, which are highly probably wings or bellies.

We calculated the Fisher score of the inner product of the eight channels to obtain an 8×8 Fisher score matrix (illustrated in Figure 4.3.(1)). We then looked into the activation of channel pairs with different ranges of Fisher scores. As shown in Figure 4.3.(3), the $288th \bigotimes 301st$ (Fisher score: 0.76) are sensitive to the combination of the semantics of two single channels and activate in the areas of brown tails or wings,

which most probably represent sparrows. Similarly, the $301st \bigotimes 501st$ (Fisher score: 0.52) activates strongly in the areas of brown breasts and wings.

We also analyzed the features with low Fisher scores to discover the fundamental cause of sparsity in the bilinear features. For example, the $280th \bigotimes 301st$ (Fisher score: 0.08) provides rare activation throughout the whole dataset, for that the $280th$ channel activates to the black and white regions, and the $301st$, to the brown regions. The other example is the $135th$ channel, which is activated to the fluff regions, and the $396th$ channel for streaks (Fisher score: 0.07). The experiments suggest that the semantic conflicts of different channels suppress the discrimination of most of the co-inner channels and lead to the sparsity of bilinear features. The Fisher score can be used to measure the representation capability effectively and select the most discriminative features.

4.3 Conclusion

We presented a novel Squeezed Bilinear Pooling (SBP) network to solve the fatal problem of bilinear pooling, the extremely high feature dimension, and obtained state-of-the-art results using VGG as the backbone. Using the Fisher feature selector, we obtained the global optimized selection of bilinear features with $O(n^2p)$ computational complexity, making it practical to deal with the high dimensional bilinear features. Our model outperforms other compressed bilinear models and low-rank approximation in terms of classification accuracy and computation performance, especially with low dimensional features. The proposed models are capable of matrix normalization and provide the best performance over other compact models to the best of our knowledge. The computational complexity of the proposed SBP increases linearly with the output feature dimension. This is a promising step for the second-order pooling towards replacing global average pooling in other deep structures, e.g., ResNet, Inception, and DenseNet.

Chapter 5

Learning Enhanced Features and Inferring Twice

The recent works [81][16][103] have shown that paying attention to multiple discriminative parts plays a vital role in FGVC. In the early work [98][8][4], extra manual bounding-box/part annotations are employed to extracting discriminative features in multiple object parts. Recent efforts [103][112][106] utilize only class labels to automatically localize the object's parts. [81][16] show that without external interference, CNNs [80][36][84] usually excel at extracting the most discriminative feature but ignores the complementary information that is crucial as well. Recently, the study of translation invariance [48][109][1] in CNN indicates that small translation or rescaling on the input image can drastically change the prediction of a deep network. Since the phenomenon occurs in CNN, we can reasonably speculate that the FGVC based on CNN will also follow the rule.

Motivated by the recent study, a novel framework named "forcing network" is proposed, referred to as F-Net, to address the challenges of FGVC. The diverse and enhanced features are obtained in F-Net by the forcing module, which is composed of the original branch and the forcing branch. The original branch generates the class activation maps (CAM) to localize the most discriminative parts. In the forcing branch, the suppressive mask is generated to suppress the primary discriminative parts and force the network to pay attention to the secondary discriminative regions that are usually overlooked. After the back gradient propagation, enhanced features are extracted for classifiers. To reduce the prediction error, the subtle regions are magnified. According to the CAM, the object is cropped and zoomed as the second input to predict the class of the object again. The first and second prediction probabilities are fused as the final result.

In the training phase, the most discriminative region of the cropped image is dropped to force the network to pay attention to more regions. The main contributions of this work can be summarised as follows:

- A novel "forcing network" structure was proposed. A forcing branch is introduced as an auxiliary branch to force the network to focus on multiple regions and extract diverse features that contain the primary discriminative features and confusion features for fine-grained visual categorization.
- Based on class activation maps, an object to be classified is cropped to the center of its image, and the subtle regions are magnified for the second prediction. The sum of the two predicted confidence scores serves as the final prediction.
- Extensive experiments were conducted on the widely-used fine-grained benchmark datasets, including CUB-200-2011, FGVC-aircraft, and Stanford-cars. The experimental results demonstrated that our method outperforms the majority of existing methods and achieved state-of-the-art performance on FGVC-Aircraft.

The methodology and experiment result are shown in Section 5.1 and 5.2, respectively.

The work in this chapter has been submitted to the Multimedia Tools and Applications [100].

5.1 Method

In this section, the F-Net and the CAM-based cropping module are described in detail. The overview architectures of the two modules are illustrated in Figure 5.1 and Figure 5.2, respectively. F-Net consists of two components, the feature extracting module and the forcing module. The feature extracting module is the convolutional backbone of ResNet-50 [36]. The forcing module and CAM-based cropping module are described in this Section 5.1.1 and 5.1.2, respectively. To acquire class activation maps, the fully connected layer for classification is replaced with a 1×1 convolutional layer. The number of the output channels of the 1×1 convolutional layer is the same as the number of classes. Given an input image, the feature maps for classification are produced by the feature extraction module. We denote the extracted feature maps as $F \in R^{NWH}$, with height H, width W, and the number of channels N.

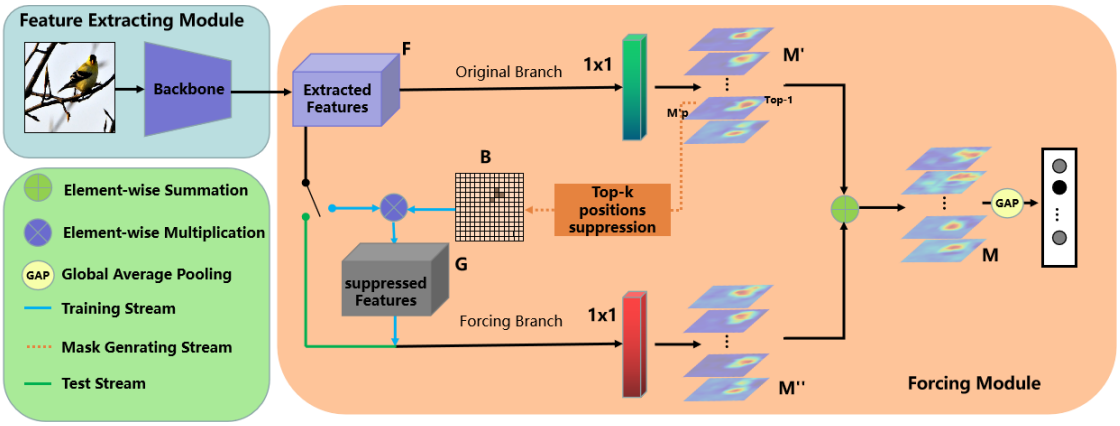


Figure 5.1: Overview of the proposed F-Net which consists of the feature extracting module and the forcing module. The feature extracting module is convolutional layers that extract features. The forcing module contains the original branch and the forcing branch.

5.1.1 Forcing Net

The proposed forcing module is inspired by DB [81]. The module aims to force the network to extract more diverse features for the classifier. It consists of an original branch and a forcing branch. The original branch and the forcing branch share the same feature extraction model, but the inputs of the two branches are different. The destination of the original branch is to generate the class activation maps and localize the primary discriminative regions. After the feature maps F are convoluted by an 1×1 convolutional layer, the class activation maps $M' \in R^{CWH}$ are obtained, where W, H, and C represent the width and height of the feature, and the number of classes, respectively. Then a global average pooling was conducted to obtain the predicted class activation maps $M'_p \in R^{WH}$, where p is the index of maximum in the predicted vector $V \in R^C$. Here, C refers to the number of classes. We have:

$$V = g(M'), \tag{5.1}$$

where $g()$ represents the Global Average pooling. For the forcing branch, M'_p is utilized to generate a mask to suppress the top-k discriminative positions of F. Since the top-k positions are suppressed, the forcing branch is forced to pay attention to other confused positions. Here, we describe the procedure to generate the input for the forcing branch in detail. Firstly, M'_p is reshaped to a vector of size $W \times H$, i.e., WH and is sorted in descending order, then, the k-th values T is obtained as the threshold value:

$$T = Sort(M'_p)[k], \tag{5.2}$$

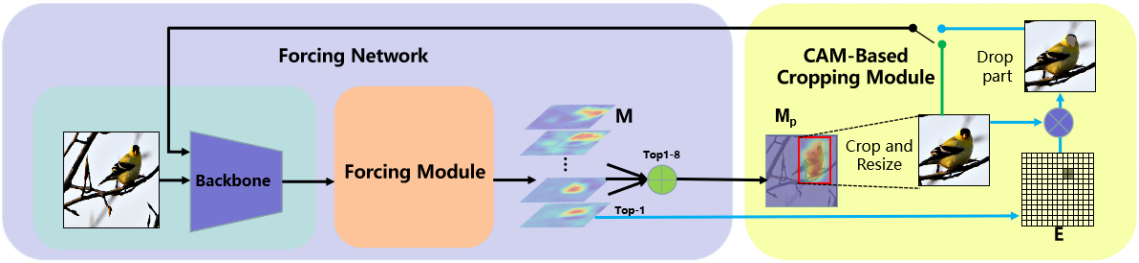


Figure 5.2: Overview of CAM-based cropping module.

where $Sort()^{\cdot}$ denotes the sorting in descending order, $[]^{\cdot}$ represents getting value from the vector, and k is a hyperparameter that denotes the number of suppressive positions. Let B be the suppressive mask derived from M'_p such that:

$$B(i,j) = \begin{cases} \alpha & M'_p(i,j) \geq T \\ 1 & M'_p(i,j) < T \end{cases}, \tag{5.3}$$

where i and j represent row and column of the position of the feature, respectively, and α is a hyperparameter that denotes suppressing factor. Finally, the input of the forcing branch $G \in R^{CWH}$ is obtained, which is generated as follows:

$$G = B \bigodot F, \tag{5.4}$$

where $\bigodot$ denotes the element-wise multiplication of the two tensors. After the classification convolution is performed, the output of the forcing branch $M'' \in R^{CWN}$ is obtained. Let M be the output of the forcing module, M is obtained as:

$$M = M' + M'', \tag{5.5}$$

The confidence scores are obtained after M is fed to global average pooling.

5.1.2 CAM-Based Cropping

The CAM-based cropping module is proposed to crop the object region as the center of the input image for the second prediction. The first prediction always focuses on the obvious regions, while the second prediction may pay attention to some subtle regions which are amplified after the CAM-based cropping module. The summation of raw prediction and the second prediction is deemed as the final prediction. Here, we explain the procedure of cropping the object. In the forcing module, the generation of class activation maps $M \in R^{CWN}$ was described. Since the top-1 map usually highly responds to the object region and the high response regions from other maps

are other parts of the object. Instead of using the top-1 map to localize the discriminative region, top-8 maps are used to crop the whole object. Denote $M_p \in R^{WN}$ as the element-wise summation of top-8 maps. M_p consists of the object and background. The threshold value t is set to distinguish the object and background, and t is generated as follows:

$$t = max(M_p) \times d, \tag{5.6}$$

where d is the random coefficient for the diversity of the samples. d follows the uniform distribution between 0.4 and 0.6 in the training phase and is set to the minimum value of 0.4 in the test phase to ensure the whole major object is included. Then, the crop mask B_2 is obtained as follows:

$$B_2(i,j) = \begin{cases} 1 & M_p(i,j) \geq t \\ 0 & M_p(i,j) < t \end{cases}. \tag{5.7}$$

The response values greater than or equal to t belong to the object. Otherwise, they belong to the backgrounds. We generate a bounding box covering all positions of 1 in B_2 and crop the object from the raw image as the second input. In the training phase, the discriminative parts are dropped in the second input. It should be noted that discriminative parts do not drop in the test phase. The drop mask is obtained as follows:

$$E(i,j) = \begin{cases} 0 & M_1(i,j) \geq m_1 \times 0.75 \\ 1 & M_1(i,j) < m_1 \times 0.75 \end{cases}, m_1 = max(M_1) \tag{5.8}$$

where M_1 is the top-1 map of M and m_1 is the maximum of M_1. As the training progresses, the size of the high response area changes all the time, so the threshold value is set to a fixed value of 0.75 instead of random values. In the second inference, the positions with the value of E=0 will be discarded due to low discrimination.

5.2 Experimental Results of F-Net

This section presents the experiments conducted to verify the effectiveness of F-Net. Firstly, the implementation details are described in Section 5.2.1. Then the proposed model is compared with other methods on the three benchmark fine-grained visual classification datasets in Section 5.2.2, followed by analysis on the contribution of each component in the proposed framework in Section 5.2.3.

5.2.1 Implementation Details

In the following experiments, following previous attention based FGVC works [9][35][16], ResNet-50 [36] implemented in Pytorch [72] is adopted as the backbone, and the fully connected layer is replaced with a 1×1 convolutional layer, which has the same number of output channels as the number of classes. The feature extraction convolutional layers is initialized with pre-trained ResNet-50 weights on ImageNet [15], and the classification layer is initialized using Xavier initialization [32].

In the training phase, the input images are resized to 515×512 and then randomly cropped to 448×448 with random horizontal flipping. The threshold for the cropping, d, is randomly selected from 0.4 to 0.6 for every sample, and the threshold for the dropping is set to 0.75, as described in Section 5.1.2. We train our network using Stochastic Gradient Descent (SGD) with the momentum of 0.9, epoch number of 100, weight decay of 0.0001, and a mini-batch of 6 on GTX-1080(8G) GPU. The initial learning rate is set to 0.001 and decayed on the 30th epoch with a decay rate of 0.1. Source code is released at https://github.com/boxyao/Forcing-Network.

5.2.2 Quantitative Results

No manual annotations except for the class labels are used in the experiments. For a fair comparison, our method is compared with the methods without using human-defined bounding boxes or part annotations. The comparison was conducted between various recent and top-performing methods on three challenging datasets, including CUB200-2011, FGVC aircraft, and Stanford-cars. Table 5.1 illustrates the comparison of the performance of different FGVC methods on the three datasets.

On the CUB-200-2011 dataset, the baseline based on ResNet-50 achieved an accuracy of 85.4%. Our method outperforms the baseline by 3.0%. A further improvement of 0.7% can be observed when we use DenseNet-161 [44] as the backbone. Comparing with MGE-CNN [107] based on ResNet-50, which used multi-experts, we obtained almost the same accuracy by adding an auxiliary classifier. Both our approach and the DB [81] extracted diverse features by feature suppression. Although the DB method outperforms our method by 0.2%, our forcing module outperforms DB without Gradient-boosting loss by 1%.

On the FGVC-aircraft dataset, the proposed F-Net on ResNet-50 and DenseNet-161 [44] achieved an accuracy of 93.3% and 94.4%, respectively. Comparing with the methods based on ResNet-50, our methods outperform most of the state-of-the-

Method	Backbone	Resolution	Parameters	Cub	Airplane	Car
ResNet-50[36]	ResNet-50	448	23.9M	85.4	88.5	91.7
NTS-Net[103]	ResNet-50	448	25.5M	87.5	91.4	93.9
DCL[10]	ResNet-50	448	24.7M	87.8	93.0	94.5
S3N[16]	ResNet-50	448	>101.5M	88.5	92.8	94.7
MGE-CNN[107]	ResNet-50	448	>25.1M	88.5	-	93.9
DB[81]	ResNet-50	448	23.9M	88.6	93.5	94.9
Stacked-LSTM[37]	ResNet-50	800	-	90.4	-	-
API-Net[18]	DenseNet-161	448	30.3M	90.0	93.9	95.3
AttNet&AffNet[35]	ResNet-50	448	23.8M	88.9	94.1	95.6
MC-Loss[9]	ResNet-50	448	67.1M	86.4	92.9	94.4
Ours	ResNet-50	448	24.3M	88.4	93.3	94.5
Ours	DenseNet-161	448	27.3M	89.1	**94.4**	94.8

Table 5.1: Comparison with the state-of-the-art on the CUB-200-2011, Stanford Cars, and FGVC Aircraft benchmarks.

Method	Accuracy
ResNet-50	85.6%
ResNet-50+Forcing Module	87.3%
ResNet-50+CAM-based Copping Module	88.1%
ResNet-50+Forcing Module+CAM-based Copping Module	88.4%

Table 5.2: Ablation analysis on the CUB-200-2011.

art methods except DB. Our method based on DenseNet-161 shows state-of-the-art performance, outperforms DenseNet-161 based APINet [18] by 0.5%.

On the Stanford-cars dataset, our ResNet-50 based method obtained an accuracy of 94.5%, which is 2.8% higher than the accuracy of the baseline, 91.7%.

Figure 5.3 shows some examples of experimental results. The results show that the proposed structure is activated to different parts of a raw input image and its cropped image. The examples with the wrong prediction from the original image and correct prediction from the cropped image indicate that the two-step strategy can reduce prediction error.

5.2.3 Ablation Study

To further analyze the contribution of different components in our method, extensive experiments were conducted on CUB-200-201 using ResNet-50. Table 5.2 illustrates the detailed contribution of each key component. It shows both the forcing branch

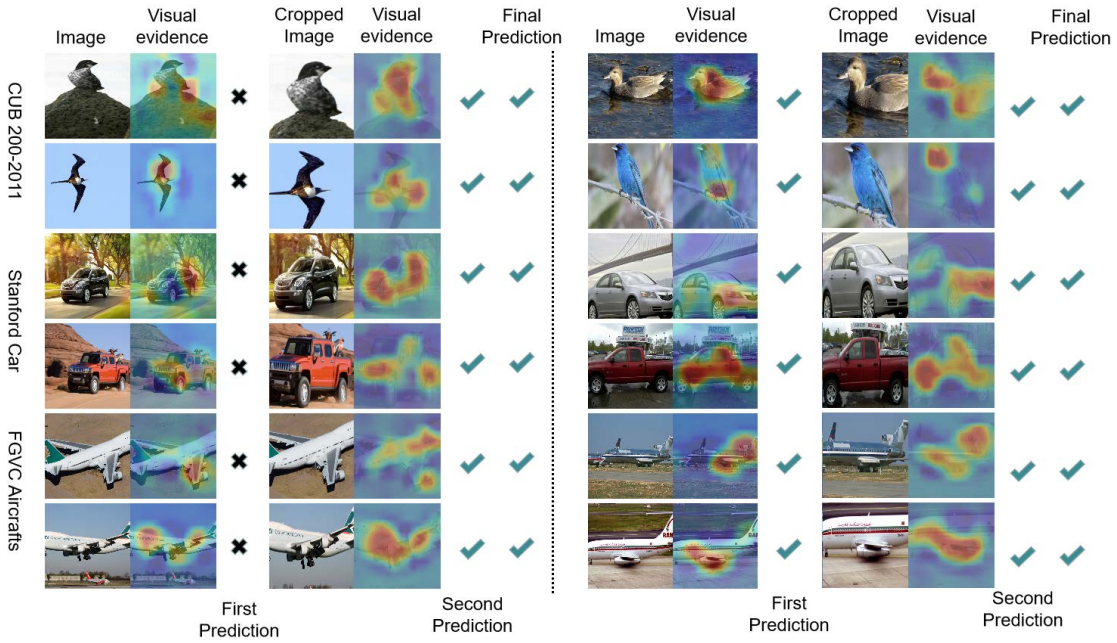


Figure 5.3: Visualisation of our method. The one to the left of the dotted line is where the first prediction was wrong and the second prediction was correct, and the final prediction was correct. The examples shown to the right of the dotted line are those with the first, the second, and final predictions are all correct. Each of these examples from left to right includes the original image, top-1 class activation map of the original image, prediction of the original image, cropped image, top-1 class activation map of the cropped image, prediction of the cropped image, the summation of the prediction from the original image, and the prediction from the cropped image.

k	0	1	2	3	4	5	6
Accuracy	86.9	87.0	87.1	86.8	**87.3**	87.2	87.0

Table 5.3: Ablation study on the number of suppressing positions k.

and the crop inference are effective in improving the performance of FGVC.

Impact of forcing branch - Basic ResNet-50 with the forcing branch achieved a top-1 accuracy of 87.3%. Since the primary discriminative part for the forcing branch classifier is suppressed, the network is forced to focus on other equally important parts rather than the primary discriminative part. It also means we enhanced the weight of the secondary discriminative regions in the extracted features. In the inference phase, the diverse features are acquired by CNN, and the classifier of each branch pays attention to different parts. Our experimental results show that the forcing branch improved the accuracy of the backbone network by 1.7%.

CAM-based cropping module - Because we have conducted panning and rescaling of the input images, the prediction from the cropped image is different from that of the raw image. The results in Figure 5.3 show that the network always pays attention to different parts when the object is panned or zoomed. Double prediction improves the result from 85.6% to 88.1%. The 2.5% improvement shows that the second prediction can reduce the classification error, and the combination of the forcing model and the double prediction leads to an improvement of the classification accuracy by 0.3%. Using the cropped object to the center of the image, the network pays attention to more object parts as the object is more evident than the first to the network. The experiments show that the forcing module can force the network to focus on other confusing parts as well. This can further improve the classification accuracy from 88.1% to 88.4%.

Two hyperparameters: suppressing factor α and the number of suppressing positions k - The classification accuracy is impacted by both the number of suppression positions, k, and the suppressing factor, α, as shown in Table 5.3 and Table 5.4. When the top-k positions are suppressed based on the class activation maps, the classification accuracy changed as well. It is observed that suppressing too many positions or setting an over small α will result in lower accuracy. We first fix α to 0.5 and compare the performance of different k. Specifically, when $k = 4$, the best performance was achieved. Then we fix k to 4 and compare the performance of different α. The experiments indicate that when $\alpha = 0.5$, the best classification performance was obtained on CUB-200-2011.

α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Accuracy	87.0	87.1	86.7	86.9	**87.3**	86.9	86.9	87.0	86.9	86.9

Table 5.4: Ablation study on suppressing factor α.

5.3 Conclusion

In this chapter, we proposed a forcing network to focus on multiple regions as well as extract diverse features for fine-grained visual categorization. The first prediction using a raw input image is combined with the second prediction using a copped image of the raw input image to predict the final classification result. The cropped image was obtained from the raw image based on class activation maps from the first prediction. The forcing network does not require bounding boxes or part annotations and can be trained end-to-end. Our method outperforms most methods of FGVC on the benchmark datasets, including CUB-200-2011, FGVCAircraft, Stanford-Cars, and achieved state-of-the-art performance on FGVC-Aircraft. We have experimented with our model using ResNet and DenseNet backbones, which proved that our model could be embedded to multiple structures and improve the classification performance. Although our method has improved the accuracy greatly, the suppressed region is highly dependent on hyperparameters. We will reduce the dependence on hyperparameters while maintaining higher accuracy in future research.

Chapter 6

Category Attention Teaching CNN

Previous FGVC works have shown that the attention of multiple discriminative parts plays a vital role in FGVC. Early works ([8]) used manual bounding-boxes or part annotations to extract discriminative local features in multiple object parts. Significant efforts were made ([112]; [103]) to utilize only class labels to find and localize the most discriminative object parts automatically. Some research focuses on attention crop and attention drop to extract the secondary but discriminative features to better describe the presentation ([43]). These efforts are impressive as they have greatly improved the classification accuracy in FGVC datasets, and experimental results show that the activation of the networks is semantically more representative.

One common defect that disadvantages the previous FGVC structures to handle the mobile scenarios is that they often complicate the backbone CNN models. For example, BCNN ([62]) extended the number of the feature channels from 512 to 512×512; Stacked LSTM ([37]) applied multi-inference with LSTM to identify the most discriminative regions; and WS-DAN ([43]) introduced attention cropping and attention pooling in the inference phase, which doubled the inference time and increased the feature dimension as well. The other problem is that most of the current solutions are customized for a specific network structure. A large number of hyper-parameters like channel number, attention cropping size, dropping rate, cropping threshold, etc., make the structures extremely difficult to be transformed to other backbones.

With the rapid development of mobile devices such as mobile phones, auto robotics, drones, and auto driving devices, a great demand has been raised to use deep-learning models with lower power consumption and smaller model size. Addressing the problem, previous works [85][40][65] have successfully embedded deep-learning models to mobile platforms in detection, general classification, segmentation, etc. However, they

usually degrade the presentation performance while increasing the efficiency, which makes them insufficient to the mission of fine-grained visual classification.

Motivated by the recent works, this chapter presents a novel network structure for the fine-grained visual classification on mobile platforms, named Category Attention Transferring CNN (CAT-CNN). Based on the assumption that a model with better activation to the discriminative regions has better classification capability, we introduced the "Category Attention Teacher". Instead of only using the one-hot class label in the training phase, we combine the label loss and the category attention loss to learn "what it is" and "where to pay attention" simultaneously.

The main contributions of this study include:

- We proposed the CAT-CNN to transfer the attention knowledge of a large-scale FGVC network to multiple efficient models.
- We investigated the relationship between the category transferring rate and the classification performance over three common efficient CNN models and confirmed the best transferring rate overall models.
- Our structure achieved a classification accuracy of 84.1% with the ground truth part annotations on the CUB-200-2011 dataset, superior to the accuracy of 82.02% reported in the state-of-the-art works [108][45].
- We conducted extensive experiments to compare the CAT-CNN with other efficient structures and state-of-the-art in FGVC. The experimental results show that when using the large-scale network, the performance of the proposed CAT-CNN is similar to the state-of-the-arts and outperforms the basic efficient networks by up to 6.7%.
- The proposed CAT-CNN uses the original network structure and single inference in the testing phase. This is unique and makes the proposed CAT-CNN more efficient and suitable for mobile platforms.

The work of this chapter has been published in Pattern Recognition Letters 2021 [58].

6.1 Category Attention Teaching CNN

This section describes a novel structure called Category Attention Teaching CNN (CAT-CNN) in detail. As shown in Figure 6.1, the CAT-CNN contains two streams. The attention teacher stream has a deeper and more complex CNN structure like Resnet-152 [36], and EfficientNet-b7 [85], which has a more robust learning capability of fine-grained features. The attention student stream is an efficient structure like ShuffleNet and MobileNet. The representation capabilities of these structures are comparably weaker than the teacher nets, but the computation and memory consumption are orders lower than the teacher CNNs.

To investigate the activations of the CNN to specific categories, we replaced the last fully connected layer with an 1×1 convolutional layer. The number of the input and output channel is the same as the output channel number of the last convolutional layer and the number of classes in the dataset, noted as c. We denote the output feature map of the corresponding class as the "category activation" in the rest of this chapter. The category activation Att of the input image I from the specific network Net is defined as $Att = Net(I)$, where $Att \in \mathbb{R}^{c \times h \times w}$, $I \in \mathbb{R}^{3 \times size_{input} \times size_{input}}$, $size_{input}$ is the input image size, $h = w$ is the size of output attention map. Class score $S_{cls} = GAP(Att)$ is the global average pooling result of the category activation Att, and we have $S_{cls} \in \mathbb{R}^c$.

In Figure 6.1, we showed the comparison of class activation of pretrained EfficientNet-b7 and MobileNet-V3 on an image from CUB-200-2011 in the first and third lines, The class activations of EfficientNet-b7 are more obvious than the MobileNet-V3, and cover most discriminative regions of the birds in the image. Hence, the prediction accuracy of EfficientNet-b7 on the CUB-200-2011 dataset is 4.6% higher than that of the MobileNet-V3 (88.7% vs. 84.1%).

Inspired by the Knowledge Distilling (KD) [38][33], we proposed the class attention supervised CNN for FGVC on mobile platforms. The CAT-CNN uses the class activation of the deeper teacher net to guide the student net's training. Compared with using the one-hot class labels as the only supervision, the class-map can provide much more valid information of where the student CNN should pay attention to and enhance the representation capability of the student net. To better tackle the gap between the more complexed teacher net and simpler student net, we trained a Category Attention Teaching Assistant (CATA) using class labels and the category attention output from a pretrained Category Attention Teacher. The structure of CATA is the same as the teacher net, which means the same learning ability of the

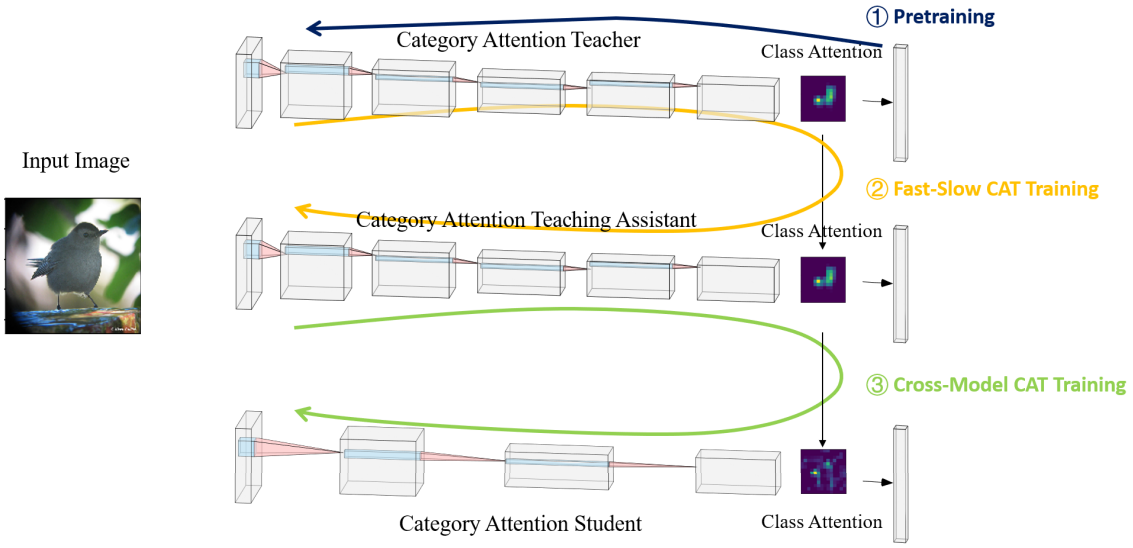


Figure 6.1: The proposed network architecture with three training phases: 1) pre-training the category attention teacher using the class labels, 2) training the category attention teaching assistant with the supervision from the class attention of the phase one and the class labels, and 3) cross-model category attention teaching for efficient networks using the supervision from the class attention of the phase two and the class labels.

CAT and CATA. The way in which CAT learns and helps the training of CATA is defined as "Fast-Slow CAT Training". The loss function can be described as follows:

$$L_{cat} = (1 - r_t)L_{cls} + r_t L_{ap}, \tag{6.1}$$

where L_{cls} is the typical cross-entropy classification loss between the output class scores and the label. L_{ap} is the mean square loss between two same-sized attention maps (Att1 and Att2), which is defined as:

$$L_{ap}(Att1, Att2) = \sum_{i,j,k}^{c,h,w} (Att1_{i,j,k} - Att2_{i,j,k})^2 / c/h/w. \tag{6.2}$$

The factor r_t is a parameter that indicates the participation of category attention teaching. If $r_t = 0$, the training of CAT-CNN degrades into regular classification training using a single model. The result is the same as that when we use the student net only.

Using the categorization attention supervision, the training includes three steps: (1) We use the class labels only to fine-tune a teacher network. (2) Using the class activation maps from the teacher net and the class labels, we train a teaching assistant

network using the same backbone as the teacher net. (3) We use the class attention maps and class labels of the input images to supervise the training of the student net.

6.2 Experimental Result of CAT-CNN

In this section, we detail our experiments in three folds. In Section 6.2.2, we investigate the impact of the most crucial parameter, the mixing rate of teacher's activation, on the proposed category attention teaching method and compare with the commonly used efficient structures without teaching. In Section 6.2.3, we conduct an overall comparison of our proposed CAT-EfficientNet-b7 model against other fine-grained methods on various datasets. Section 6.2.4 compares the proposed CAT-CNN's accuracy and efficiency with other efficient structures. In Section 6.2.5, we investigate the activation of different structures and their covariance to verify the effectiveness of the category attention teaching. We start with providing experiment details in Section 6.2.1.

6.2.1 Dataset and Implementation Details

We conducted our experiments on three widely cited FGVC datasets: CUB-200-2011 Bird dataset ([90]), FGVC-Aircrafts ([66]) and the Stanford Cars ([51]). All these datasets provide a fixed train and test split. The statistics of these datasets are summarized in Table 2.5. We only use the category label without any part or bounding box annotation provided in the datasets when training all models. To compare the performance of our proposed model with other state-of-the-art methods with extended training datasets, we also applied the NABirds ([87]) dataset in Section 6.2.3.

We evaluated our model based on four different efficient CNN backbones: the EfficientNet-b0 and EfficientNet-b7 proposed by [85], the MobileNet-V3-Large-1.0 by [40] and ShuffleNet-V2-Large-1.0 by [65]. We used the CNN layers of the backbones as the local feature extractor and replace the last fully connected layer with an 1×1 convolutional layer with the same input and out channel numbers as that of the fully connected layer to produce c output feature maps (c is the number of classes of the dataset used in each experiment, for CUB-200-2011, $n = 200$). We then compared the proposed CAT-CNN with the original structure and other FGVC models. All experiments are conducted using the Pytorch platform and implemented in Python. Following practice in previous single inference work ([16]), we use the "RandomResizedCrop" function of Pytorch to augment the input image to the resolution of

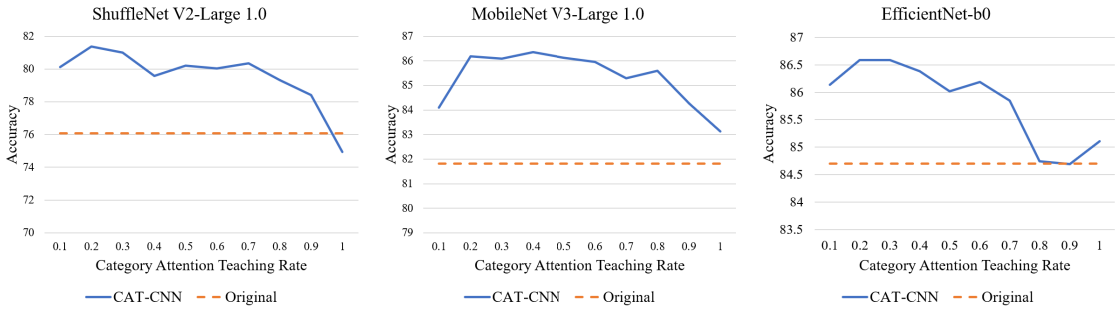


Figure 6.2: Visualization of the accuracy of Category Attention Teaching model with its teaching rate r_t. From left to right, the three graphs stand for ShuffleNet-V2-Large-1.0, MobileNet-V3-Large-1.0 and EfficientNet-b0 taught by a pretrained EfficientNet-b7 model. The blue lines represent the prediction accuracies of CAT-CNNs with different teaching rates. The orange lines demonstrate the baselines when we only use the corresponding student model without CAT structure.

448×448 for training. For testing, we resize the input image to 512×512 and center crop the image to 448×448. The training batch size is 16, and the weight of decay is 1×10^{-5}. For all layers, the initial learning rate is 0.01. We conducted testing at the end of each training epoch. If the testing accuracy does not increase for seven epochs, we reduce the learning rate by multiplying 0.1. If the learning rate is lower than 0.00001, we terminate the training process. The training method is consolidated for all structures.

6.2.2 Configuration and comparison with the baseline

The category attention teaching rate r_t, which indicates the degree of participation of the teacher in the training phase, can be adjusted manually. To investigate the potential impact of teaching rate on the category attention teaching, we conducted experiments with the teaching rate in the range of 0.1 to 1.0 with the step of 0.1 on the CUB-200-2011 dataset. We used the EfficientNet-b7 model to train a category attention teacher on the dataset. To choose a proper teaching rate for CAT-CNN, we employed three net structures, EfficientNet-b0, MobileNet-V3-Large-1.0, and ShuffleNet-V2-Large 1.0, as the backbones of the student streams in our proposed CAT-CNN structure. We also compared the performance of CAT-CNN with the original structure. For a fair comparison, the training configurations are the same for all models.

As demonstrated in Figure 6.2, the top-1 errors of both the EfficientNet-b0, MobileNet-V3-Large-1.0, and ShuffleNet-V2-Large-1.0 approached a maximum value

Method	Cub	Airplane	Car
NTS-Net ([103])	87.5	91.4	93.9
DCL ([10])	87.8	93.0	94.5
S3N ([16])	88.5	92.8	94.7
WS-DAN ([43])	89.4	93.0	94.5
MGE-CNN ([10])	88.5	-	93.9
DB ([81])	88.6	93.5	94.9
stacked-LSTM ([37])	**90.4**	-	-
API-Net ([18])	90.0	93.9	95.3
AttNet ([35])	88.9	94.1	**95.6**
MC-Loss ([9])	86.4	92.9	94.4
Mix+ ([54])	90.2	93.1	94.9
TBMSL-Net ([106])	89.6	**94.5**	94.7
EfficientNet-b7 ([85])	88.5	91.7	93.1
CAT-EfficientNet-b7	**88.8**	**92.0**	**93.9**
CAT-EfficientNet-b7+NABirds	**90.2**	-	-

Table 6.1: Comparison of the classification performance among the proposed CAT-Efficient-b7 and baselines on various FGVC datasets. The results of the previous works are duplicated from corresponding publications.

when the teaching rate was around 0.2 to 0.4. The ShuffleNet-V2-Large-1.0 reached its optimal solution when the teaching rate was 0.2, and the classification accuracy was 81.67%, which was 5.6% higher than the original ShuffleNet-V2-Large-1.0. For the MobileNet-V3-Large-1.0, the accuracy of the original structure was 81.82%, while the CAT-CNN achieved an accuracy of 86.37% with the teaching rate of 0.4, which indicated a 4.55% improvement on the accuracy. As the latest compact structure, EfficientNet-b0 achieved 86.66% and 84.70% on the CUB-200-2011 dataset with and without the CAT-CNN structure. The most significant improvement achieved was 1.96% when the teaching rate r_t was set to 0.3.

The experiments suggested that while the proposed CAT-CNN kept the compact structure and low computational consumption, it could significantly enhance the performance of general CNN structures for FGVC. Considering the individual deviation between different net structures, we used the median optimal teaching rate $r_t = 0.25$ in the rest of experiments in this chapter.

6.2.3 Fast-Slow CAT-CNN for General FGVC

We did not use any manual annotations except for class labels. Using the same backbone as the original EfficientNet-b7 but with the Fast-Slow CAT training strategy, we

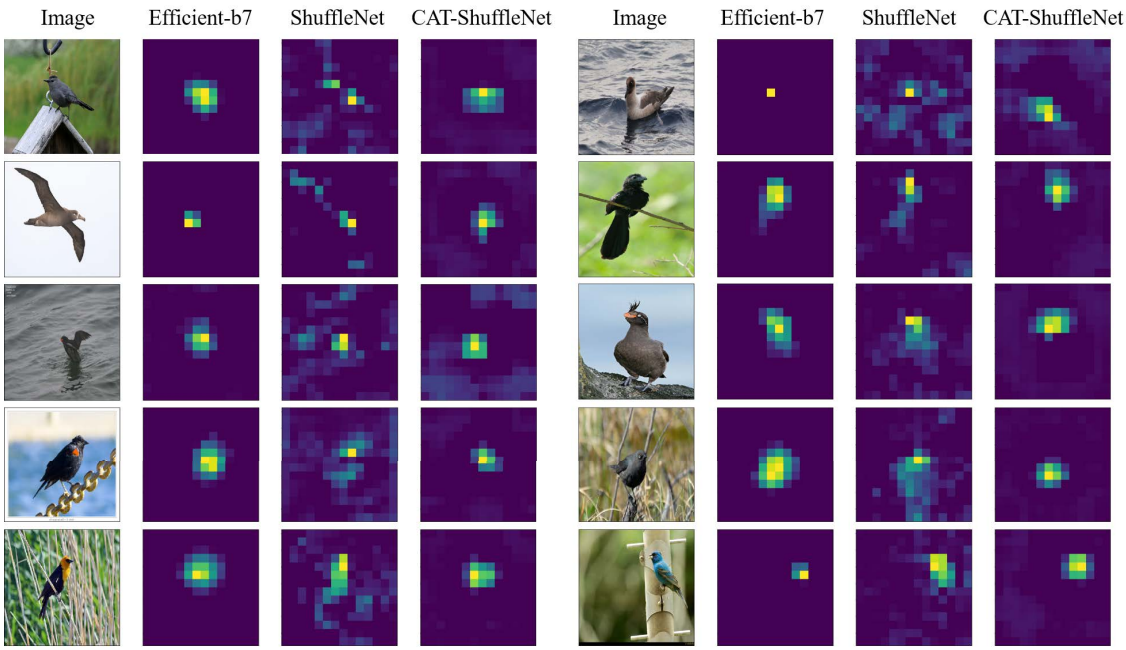


Figure 6.3: Visualization of the category activation of the CAT-ShuffleNet and comparison with the teacher steam (EfficientNet-b7) and the original ShuffleNetV2-Large-1.0. The column labeled with "Image" is the original input image. The "Efficient-b7" column shows the output of the maximum activated category of the category attention maps. The "ShuffleNet" shows the maximum category activations of the ShuffleNet-V2-Large-1.0, which is pretrained with the one-hot label only. The column labeled with "CAT-ShuffleNet" shows the maximum category activations of the proposed CAT-ShuffleNet model, which is trained with Category Attention Teaching from a pretrained EfficientNet-b7 model.

proposed the CAT-EfficientNet-b7 network. The network was compared with those without using human-defined bounding boxes or part annotations for fair comparison. The comparison was conducted with a recent top-performing method on the following datasets, CUB-200-2011, FGVC aircrafts, and Stanford-cars. Table 6.1 illustrates the results on the three datasets.

On the CUB-200-2011, the baseline based on EfficientNet-b7 achieved an accuracy of 88.5%. Our approach outperformed the baseline by 0.3% to 88.8%. A further improvement of another 1.4% was observed when we used NABirds and pretrained the CAT stream. Compared with the DB, which extracted diverse features by feature suppression and used attention cropping and double inferences, our proposed CAT-CNN outperformed the DB by 1.0%. Comparing with stackedLSTM and Mix+, which used multi-experts, we obtained almost the same accuracy by single inference and two training steps. With an extensive dataset of NABirds, we achieved the state-of-the-art accuracy of 90.2% among all single inference approaches. On other datasets like FGVC-aircraft and Stanford cars, our method achieved similar accuracies to the latest state-of-the-art FGVC methods, which was higher than the original EfficientNet-b7.

6.2.4 Cross Model CAT-CNN for Efficient FGVC

The CAT-CNN can transform latent knowledge of the teacher model to the weaker student model. Thus it can improve the student net's performance, especially when the student net has a strong limitation of memory and computation, which leads to a large gap of presentation capability between the student and the teacher streams. This section investigates the impact of the CAT structure with the EfficientNet-b7 as the teacher model and three efficient structures, the EfficientNet-b0, MobileNet-V3-Large-1.0, and ShuffleNet-V2-Large-1.0 as the student models. Experiments were conducted on three common FGVC datasets described in Section 6.2.1. To compare horizontally with other models that are widely used in image classification, we also listed the experimental results of ResNet-50. Memory consumption and computation efficiency were also compared.

The performance comparison of different models is shown in Table 6.2. For the most compact base model of ShuffleNet-V2-Large-1.0, we observed improvements of 5.7%, 4.9%, and 2.3% on the three datasets. The accuracies of CAT on MobileNet-V3-Large-1.0 are improved by 6.8%, 4.9%, and 2.7%, respectively, on the three datasets listed in Table 2.5, and the average performance is much better than the common deep-CNN structure, ResNet50, which consumes five times of memory and costs 16 times of computation. The classification performance of CAT-EfficientNet-b0 is only

Method	GFLOPs	Parameter	Cub	Airplane	Car
ResNet-50 ([36])	16.48	25.56M	85.4	89.9	91.7
EfficientNet-b7 ([85])	0.32	66.35M	88.5	91.7	93.1
ShuffleNet-V2 ([65])	0.60	2.28M	76.0	79.5	87.0
CAT-ShuffleNet-V2	0.60	2.28M	**81.7**	**84.4**	**89.3**
MobileNet-V3 ([40])	1.03	5.15M	81.8	83.8	89.5
CAT-MobileNet-V3	1.03	5.15M	**86.5**	**90.5**	**92.2**
EfficientNet-b0 ([85])	0.06	5.29M	84.7	90.0	91.9
CAT-EfficientNet-b0	0.06	5.29M	**86.7**	89.7	**92.7**

Table 6.2: Comparison of the classification performance among the proposed CAT based models and baselines on various FGVC datasets. Parameter and GFLOPs stand for the number of model parameters and giga FLOPS required for the inference of each model, respectively.

improved by 2.0%, -0.3%, and 0.8%, respectively, on the three datasets, compared with the original EfficientNet-b0 because its accuracy is close to the teacher's.

6.2.5 Validation via Visualization

We extracted the category activation maps and visualized the channel with the maximum average value from three different models: EfficientNer-b7, ShuffleNet-V2-Large-1.0, and the proposed CAT-ShuffleNet-V2-Large-1.0. As shown in Figure 6.3, the brighter region indicates stronger activation of the convolutional network. The "Efficient-b7" column shows great affinity and accuracy of the category activation from the Efficient-b7 net structure. The most activated regions are the heads and bodies of the birds. The activation on the background is close to zero in most cases, which means that the shape and texture of the non-discriminative background have less influence on the classification result. In the "ShuffleNet" column, we can see that ShuffleNet-V2-Large-1.0 provides much less accurate and shape-related activations with the correct category. This is caused by the background noise, like the grasses, bushes, or reflections on the water, and leads to a 12.5% decrease of accuracy on the CUB-200-2011 from EfficientNer-b7 to ShuffleNetV2-Large-1.0 (88.5% vs 76.0%).

In the column labeled with "CAT-ShuffleNet", we show the category activation of the proposed CAT-ShuffleNet-V2Large-1.0. Comparing with the original ShuffleNet, the CATShuffleNet is much more concentrated. Most of the activations are concentrated on the discriminative regions such as heads and bodies. In the other aspect, CAT-ShuffleNet extracts more discriminative features than the original ShuffleNet, for that the activation on the background is suppressed to a low level. Compared with

the Efficient-b7, the activated region and intensity are similar to that of the original ShuffleNet. This indicates that our proposed CAT-ShuffleNet has learned the attention patterns from the Category Attention Teacher. The classification accuracy is 81.7%, which is lower than the attention teacher but is much higher than that of the original structure (76.0%).

6.3 Conclusion

We presented a novel Category Attention Teaching CNN (CAT-CNN) framework to satisfy the demand for Efficient Fine-Grained Visual Classification. Using the cross model CAT-CNN, we can transfer the latent discriminative knowledge of a larger structure to a more efficient one to significantly enhance the compact CNNs in FGVC applications. Our model outperforms the commonly used single inference classification model, ResNet-50, with about 1/5 of the memory utilization and about 1/16 of the computation cost. Furthermore, we use the Fast-Slow CAT-CNN framework to enhance the EfficientNet-b7 and achieve the same level of accuracy as that of the state-of-the-art approaches without cropping or second inference. Deploying such an application on mobile devices requires small, simple but accurate models which are less resource-intensive. So our proposed model is promising to be deployed as a mobile FGVC app to low-power platforms like smartphones, drones, and automatic robots.

Chapter 7

Conclusions and Future Work

This chapter concludes the research work, the thesis's achievement, and the potential future work that needs to be done. Section 7.1 illustrates the conclusions of the thesis. Section 7.2 lists two possible future working roots that may be novel and applicable in the field of fine-grained visual classification.

7.1 Conclusions

In this thesis, we presented our achievements on the fine-grained visual classification in four stages. First, we proposed a regression-based deep localisation structure and a novel PC-CNN model for the fine-grained classification. We handled occlusion and deformation in images by designing a detection-localisation mechanism to increase the localisation accuracy. By defining separated part side nets, the proposed model introduced detail part-level representations into the classification. This has led to significant improvement of the fine-grained classification on the benchmark CUB-200-2011 dataset. Examining the output feature maps of each part stream reveals that independently trained part feature exactors can produce more meaningful activation and discriminate part differences among different species with subtle appearance variances. By applying ResNet and a 1×1 convolution layer to reduce the input dimension of the FCN classifier, we reduced the model size to a level similar to or smaller than most of the commonly used structures.

In the meantime, based on the second-order feature encoding methods, we presented a novel Squeezed Bilinear Pooling (SBP) network to solve the fatal problem of bilinear pooling, the exceptionally high feature dimension. Using Fisher feature selector, we obtained the global optimised selection with $O(n^2p)$ computation complexity, which made it practical to deal with the high dimension bilinear features.

Our model has the best overall performance compering with other compressed bilinear models and low-rank approximation in terms of classification accuracy and computation efficiency, especially with low dimension features. To the best of our knowledge, the proposed models are capable of matrix normalisation and provide the best performance over other compact models. The computation complexity of the proposed SBP increases linearly with the output feature dimension. This is a promising step for bilinear-based pooling towards replacing global average pooling in other deep structures, e.g., ResNet, Inception, and DenseNet.

Following the Recurrent Attention models, we proposed a forcing network to force the network to focus on multiple parts and extract diverse features for fine-grained visual categorisation and infer the object twice to reduce misclassification rates. Our method outperforms most methods of FGVC on CUB-200-2011, FGVC-aircraft, and Stanford-cars. We took a further step by transferring the attention knowledge from a large-scale deep network to a smaller but more efficient network to significantly improve the performance of FGVC on mobile platforms. Experiments have shown that the proposed CAT-CNN structure can enable the compact networks to activate similarly as a large FGVC network and achieve a high classification accuracy.

7.2 Future Work

The proposed models have achieved promising results on both part-based, recurrent attention-based, and efficient fine-grained image visual classification, and have achieved our pre-set research objectives described in Section 1.3. However, there is still some buffer for future improvements.

In this section, we present some research ideas for further improvement of the fine-grained visual classification.

7.2.1 Extend the Squeezed Bilinear Pooling to General Structure

The bilinear pooling obtains the covariance of very high dimension CNN features. The representative of covariance features drops as the number of CNN output feature maps increase. As shown in Table 7.1, the bilinear pooling has great potential for further performance improvement with the backbone of VGG-16, which outputs 512 feature maps from the Relu5_3 layer. With ResNet-50 as the backbone and 2,048 output feature maps, the bilinear pooling has achieved an accuracy of 81.6%, which is only 3.4% higher than that of the baseline. For DenseNet with 1,024 output feature

maps, the classification accuracy of the bilinear pooling is dropped by 2.1% (82.1% vs. 84.2%) comparing with that of the baseline. Surprisingly, both ResNet and DenseNet have achieved lower classification accuracies than VGG-16, though their structures are much deeper than VGG's.

Backbone	Pooling	Output maps	Error rate
VGG-16	GAP	512	73.3%
VGG-16	FBP	512	84.1%
DenseNet-161	GAP	1024	84.2%
DenseNet-161	FBP	1024	82.1%
ResNet-50	GAP	2048	78.2%
ResNet-50	FBP	2048	81.6%

Table 7.1: Classification accuracy of Bilinear Pooling with different backbones on CUB-200-2011

To address this problem, some research such as iSQRT-COV [55] and WS-DAN [42] investigated PCA trainable functions to reduce the output feature dimension and achieved promising but not satisfactory enough results. This is because such dimension compression leads to considerable accuracy loss [62]. The proposed SBP can be a potential solution for the compression of the extremely high dimension ResNet or DenseNet features.

We have conducted some initial experiments by applying the proposed SBP to the following efficient backbones: the MobileNet V3 large 1.0, the ShuffleNet-V2-large-1.0, and the EfficientNet-b0 and EfficientNet-b7. The experimental results are shown in Figure 7.1.

We can see from the figure that with a similar feature dimension (around 1,000), the SBP led to the improvement of classification accuracies by 1.8%, 3.1%, and 0.7%, respectively, over the original global average pooling on MobileNet-V3, ShuffleNet-V2, and EfficientNet-b0 backbones. The improvements proved that the proposed SBP could be easily generalised to other networks to improve their performance. If we increase the feature dimension of the SBP, the classification accuracies of MobileNet-V3 and ShuffleNet-V2 can be improved by 2.3% and 4.3%, respectively, which indicates that a higher feature dimension of the SBP can improve the representation of the squeezed bilinear feature. However, for the very-large-scale net structure, EfficientNet-b7, the SBP failed to improve the performance. This is because the performance of EfficientNet-b7 is already close to the state-of-the-art on the CUB-200-2011 dataset.

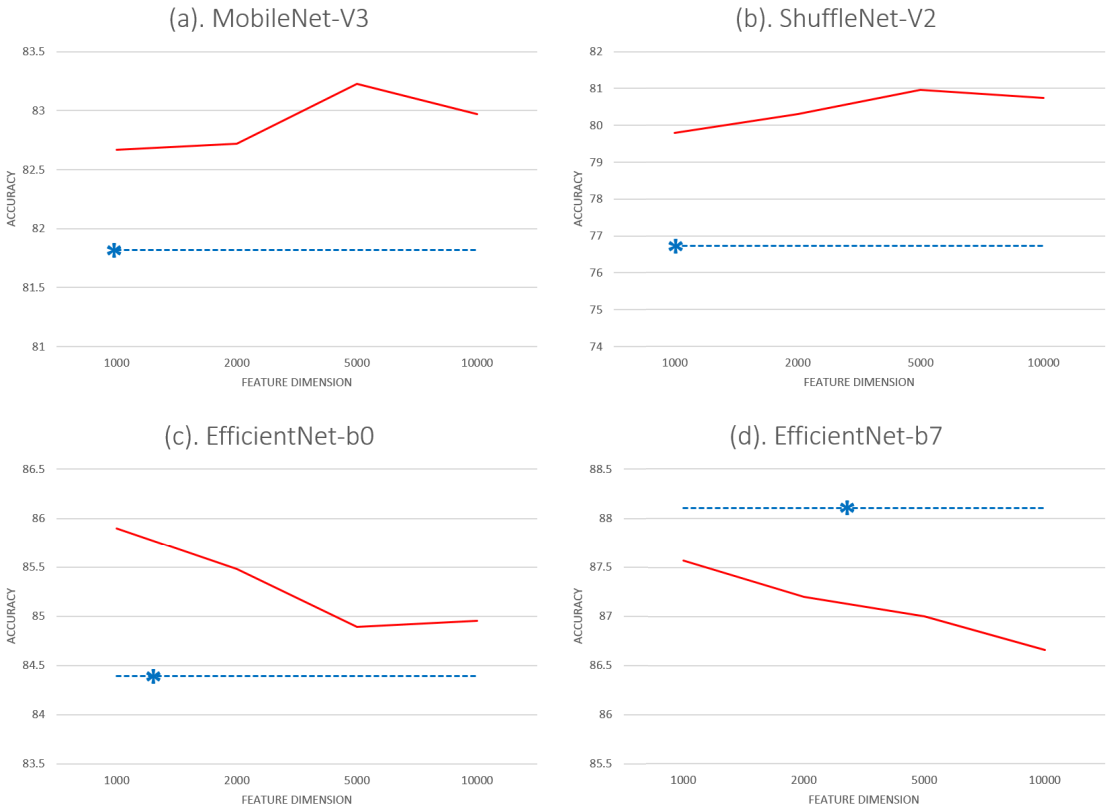


Figure 7.1: The CUB-200-2011 prediction accuracy of SBP on four different backbones: (a). MobileNet-V3; (b). ShuffleNet-V3; (c). EfficientNet-b0; (d). EfficientNet-b7. The red lines indecate the accuracies using different featrue dimension. The blue lines of dashes are the baseline of the original structures without SBP, and the blue stars is the feature dimension of the orginal structures.

Therefore, future research can focus on how to generalise the squeezed bilinear pooling and embed it into popular deep learning networks as a plugin component.

7.2.2 Weakly Supervised Part Discovery

Most of the attention-based methods in the fine-grained classification rely on local activation grouping to localise the discriminative semantic object parts. For example, a channel grouping loss is employed [112] to roughly localise four parts (head, wings, feet, and tail) for classification in detail. There is a need to develop a more accurate and robust part segmentation method without semantic part annotations. The method can be used for Weakly Supervised Part Discovery (WSPD).

Based on the semi-supervised semantic segmentation, an autoencoding formulation was proposed [110] to discover landmarks as explicit structural representations. Combined with the SBP fine-grained features, a novel framework for weakly supervised part discovery and segmentation can be developed to further improve the performance of the attention-based fine-grained classification models. Therefore, we suggest this as another potential area for future work.

Chapter 8

Publications

1. Qiyu Liao, Hamish Holewa, Min Xu, and Dadong Wang. Fine-grained categorization by deep part-collaboration convolution net. In *2018 Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–8. IEEE, 2018

2. Qiyu Liao, Dadong Wang, Hamish Holewa, and Min Xu. Squeezed bilinear pooling for fine-grained visual categorization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019

3. Liao Qiyu, Aristizabal David, Ahmedt, Tychsen-Smith Lachlan, Wang Dadong, Petersson Lars, and Salvado Olivier. Abattoir track and trace solution. *CSIRO Report No. EP198430*, 12 Dec.2019

4. Wang Dadong, Arzhaeva Yulia, Devnath Liton, Qiao Maoying, Amirgholipour Saeed, Liao Qiyu, McBean Rhiannon, Hillhouse James, Luo Suhuai, Meredith David, Newbigin Katrina, and Deborahz Yates. Automated pneumoconiosis detection on chest x-rays using cascaded learning with real and synthetic radiographs. In *Proceedings of the 2020 Digital Image Computing: Techniques and Applications (DICTA)*, pages pp. 1–6, DOI: 10.1109/DICTA51227.2020.9363416, 29 Nov. – 2 Dec. 2020

5. Edwards Everard, J, Thomas Mark, Gensemer Stephen, Lagerstrom Ryan, Khokher Rizwan, Liao Qiyu, Sun Changming, Wang Dadong, Hargrave Chad, and Ralston Jonathon. New non-destructive technologies for simultaneous yield, crop condition and quality estimation. *the final project report submitted to Wine Australia, Wine Australia Project No. CSA 1602*, July 2020

6. Arzhaeva Yulia, Wang Dadong, Alam Md, Shariful, Momeni Saba, Liao Qiyu, Salvado Olivier, Sowmya Arcot, Park Eun-Kee, and Yates Deborah. Optimization and pilot development of pneumoconiosis detection software. *The first progress report for Coal Services Health and Safety Trust, Coal Services Health and Safety Trust Project No. 20656, CSIRO Report No. EP21935, 25 pages*, 22 Feb. 2021

7. Nie Xuan, Wang Luyao, Liao Qiyu, and Xu Min. Learning enhanced features and inferring twice for fine-grained image recognition. *submitted to Multimedia Tools and Applications*, 2021

8. Qiyu Liao, Dadong Wang, and Min Xu. Category attention transfer for efficient fine-grained visual categorization. *Pattern Recognition Letters*, 2021

Bibliography

[1] Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *arXiv preprint arXiv:1805.12177*, 2018.

[2] Jimmy Ba, Volodymyr Mnih, and Koray Kavukcuoglu. Multiple object recognition with visual attention. *arXiv preprint arXiv:1412.7755*, 2014.

[3] Peter N. Belhumeur, Joao P Hespanha, and David J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):711–720, 1997.

[4] Thomas Berg and Peter Belhumeur. Poof: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 955–962, 2013.

[5] Steve Branson, Oscar Beijbom, and Serge Belongie. Efficient large-scale structured learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1806–1813, 2013.

[6] Steve Branson, Grant Van Horn, Serge Belongie, and Pietro Perona. Bird species categorization using pose normalized deep convolutional nets. *arXiv preprint arXiv:1406.2952*, 2014.

[7] Yuning Chai, Victor Lempitsky, and Andrew Zisserman. Bicos: A bi-level co-segmentation method for image classification. In *2011 International Conference on Computer Vision*, pages 2579–2586. IEEE, 2011.

[8] Yuning Chai, Victor Lempitsky, and Andrew Zisserman. Symbiotic segmentation and part localization for fine-grained categorization. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 321–328. IEEE, 2013.

[9] Dongliang Chang, Yifeng Ding, Jiyang Xie, Ayan Kumar Bhunia, Xiaoxu Li, Zhanyu Ma, Ming Wu, Jun Guo, and Yi-Zhe Song. The devil is in the channels: Mutual-channel loss for fine-grained image classification. *IEEE Transactions on Image Processing*, 29:4683–4695, 2020.

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

[11] Qiang Cheng, Hongbo Zhou, and Jie Cheng. The fisher-markov selector: fast selecting maximally separable feature subset for multiclass classification with applications to high-dimensional data. *IEEE transactions on pattern analysis and machine intelligence*, 33(6):1217–1233, 2011.

[12] Yin Cui, Yang Song, Chen Sun, Andrew Howard, and Serge Belongie. Large scale fine-grained categorization and domain-specific transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4109–4118, 2018.

[13] Yin Cui, Feng Zhou, Jiang Wang, Xiao Liu, Yuanqing Lin, and Serge Belongie. Kernel pooling for convolutional neural networks. In *CVPR*, volume 1, page 7, 2017.

[14] Wang Dadong, Arzhaeva Yulia, Devnath Liton, Qiao Maoying, Amirgholipour Saeed, Liao Qiyu, McBean Rhiannon, Hillhouse James, Luo Suhuai, Meredith David, Newbigin Katrina, and Deborahz Yates. Automated pneumoconiosis detection on chest x-rays using cascaded learning with real and synthetic radiographs. In *Proceedings of the 2020 Digital Image Computing: Techniques and Applications (DICTA)*, pages pp. 1–6, DOI: 10.1109/DICTA51227.2020.9363416, 29 Nov. – 2 Dec. 2020.

[15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[16] Yao Ding, Yanzhao Zhou, Yi Zhu, Qixiang Ye, and Jianbin Jiao. Selective sparse sampling for fine-grained image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6599–6608, 2019.

[17] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655, 2014.

[18] Xinshuai Dong, Hong Liu, Rongrong Ji, Liujuan Cao, Qixiang Ye, Jianzhuang Liu, and Qi Tian. Api-net: Robust generative classifier via a single discriminator. In *European Conference on Computer Vision*, pages 379–394. Springer, 2020.

[19] Abhimanyu Dubey, Otkrist Gupta, Pei Guo, Ramesh Raskar, Ryan Farrell, and Nikhil Naik. Pairwise confusion for fine-grained visual classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 70–86, 2018.

[20] Edwards Everard, J, Thomas Mark, Gensemer Stephen, Lagerstrom Ryan, Khokher Rizwan, Liao Qiyu, Sun Changming, Wang Dadong, Hargrave Chad, and Ralston Jonathon. New non-destructive technologies for simultaneous yield, crop condition and quality estimation. *the final project report submitted to Wine Australia, Wine Australia Project No. CSA 1602*, July 2020.

[21] Dean P Foster and Edward I George. The risk inflation criterion for multiple regression. *The Annals of Statistics*, pages 1947–1975, 1994.

[22] Jianlong Fu, Heliang Zheng, and Tao Mei. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In *CVPR*, volume 2, page 3, 2017.

[23] Keinosuke Fukunaga. *Introduction to statistical pattern recognition*. Elsevier, 2013.

[24] Kunihiko Fukushima and Sei Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pages 267–285. Springer, 1982.

[25] Yang Gao, Oscar Beijbom, Ning Zhang, and Trevor Darrell. Compact bilinear pooling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 317–326, 2016.

[26] Efstratios Gavves, Basura Fernando, Cees GM Snoek, Arnold WM Smeulders, and Tinne Tuytelaars. Fine-grained categorization by alignments. In *Proceedings of the IEEE international conference on computer vision*, pages 1713–1720, 2013.

[27] Efstratios Gavves, Basura Fernando, Cees GM Snoek, Arnold WM Smeulders, and Tinne Tuytelaars. Local alignments for fine-grained categorization. *International Journal of Computer Vision*, 111(2):191–212, 2015.

[28] Weifeng Ge, Xiangru Lin, and Yizhou Yu. Weakly supervised complementary parts models for fine-grained image classification from the bottom up. *arXiv preprint arXiv:1903.02827*, 2019.

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

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

[31] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[32] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[33] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021.

[34] Mengran Gou, Fei Xiong, Octavia Camps, and Mario Sznaier. Monet: Moments embedding network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3175–3183, 2018.

[35] Harald Hanselmann and Hermann Ney. Elope: Fine-grained visual classification with efficient localization, pooling and embedding. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1247–1256, 2020.

[36] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[37] Yaya Heryadi and Harco Leslie Hendric Spits Warnars. Learning temporal representation of transaction amount for fraudulent transaction recognition using cnn, stacked lstm, and cnn-lstm. In *2017 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom)*, pages 84–89. IEEE, 2017.

[38] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[39] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

[40] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1314–1324, 2019.

[41] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[42] Tao Hu and Honggang Qi. See better before looking closer: Weakly supervised data augmentation network for fine-grained visual classification. *CoRR*, abs/1901.09891, 2019.

[43] Tao Hu and Honggang Qi. See better before looking closer: Weakly supervised data augmentation network for fine-grained visual classification, 2019.

[44] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

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

[46] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[47] Purushottam Kar and Harish Karnick. Random feature maps for dot product kernels. In *Artificial Intelligence and Statistics*, pages 583–591, 2012.

[48] Osman Semih Kayhan and Jan C van Gemert. On translation invariance in cnns: Convolutional layers can exploit absolute spatial location. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14274–14285, 2020.

[49] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, June 2011.

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

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

[52] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[53] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[54] Hao Li, Xiaopeng Zhang, Qi Tian, and Hongkai Xiong. Attribute mix: semantic data augmentation for fine grained recognition. In *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, pages 243–246. IEEE, 2020.

[55] Peihua Li, Jiangtao Xie, Qilong Wang, and Zilin Gao. Towards faster training of global covariance pooling networks by iterative matrix square root normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 947–955, 2018.

[56] Qiyu Liao, Hamish Holewa, Min Xu, and Dadong Wang. Fine-grained categorization by deep part-collaboration convolution net. In *2018 Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–8. IEEE, 2018.

[57] Qiyu Liao, Dadong Wang, Hamish Holewa, and Min Xu. Squeezed bilinear pooling for fine-grained visual categorization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019.

[58] Qiyu Liao, Dadong Wang, and Min Xu. Category attention transfer for efficient fine-grained visual categorization. *Pattern Recognition Letters*, 2021.

[59] Di Lin, Xiaoyong Shen, Cewu Lu, and Jiaya Jia. Deep lac: Deep localization, alignment and classification for fine-grained recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1666–1674, 2015.

[60] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[61] Tsung-Yu Lin and Subhransu Maji. Improved bilinear pooling with cnns. 2017.

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

[63] Jiongxin Liu, Angjoo Kanazawa, David Jacobs, and Peter Belhumeur. Dog breed classification using part localization. In *European conference on computer vision*, pages 172–185. Springer, 2012.

[64] Xiao Liu, Tian Xia, Jiang Wang, and Yuanqing Lin. Fully convolutional attention localization networks: Efficient attention localization for fine-grained recognition. 2016.

[65] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018.

[66] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[67] Subhransu Maji and Gregory Shakhnarovich. Part and attribute discovery from relative annotations. *International journal of computer vision*, 108(1-2):82–96, 2014.

[68] Ofer Matan, Christopher JC Burges, Yann LeCun, and John S Denker. Multi-digit recognition using a space displacement neural network. In *Advances in neural information processing systems*, pages 488–495, 1992.

[69] Geoffrey McLachlan. *Discriminant analysis and statistical pattern recognition*, volume 544. John Wiley & Sons, 2004.

[70] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *Advances in neural information processing systems*, pages 2204–2212, 2014.

[71] Mohammad Moghimi, Serge J Belongie, Mohammad J Saberian, Jian Yang, Nuno Vasconcelos, and Li-Jia Li. Boosted convolutional neural networks. In *BMVC*, 2016.

[72] 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. 2017.

[73] Ninh Pham and Rasmus Pagh. Fast and scalable polynomial kernels via explicit feature maps. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 239–247. ACM, 2013.

[74] Josien PW Pluim, JB Antoine Maintz, and Max A Viergever. Mutual-information-based registration of medical images: a survey. *IEEE transactions on medical imaging*, 22(8):986–1004, 2003.

[75] Liao Qiyu, Aristizabal David, Ahmedt, Tychsen-Smith Lachlan, Wang Dadong, Petersson Lars, and Salvado Olivier. Abattoir track and trace solution. *CSIRO Report No. EP198430*, 12 Dec.2019.

[76] Bernhard Scholkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

[77] Pierre Sermanet, Andrea Frome, and Esteban Real. Attention for fine-grained categorization. *arXiv preprint arXiv:1412.7054*, 2014.

[78] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813, 2014.

[79] Kevin J Shih, Arun Mallya, Saurabh Singh, and Derek Hoiem. Part localization using multi-proposal consensus for fine-grained categorization. *arXiv preprint arXiv:1507.06332*, 2015.

[80] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[81] Guolei Sun, Hisham Cholakkal, Salman Khan, Fahad Khan, and Ling Shao. Fine-grained recognition: Accounting for subtle differences between similar classes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12047–12054, 2020.

[82] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep convolutional network cascade for facial point detection. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3476–3483. IEEE, 2013.

[83] Johan AK Suykens and Joos Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.

[84] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[85] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019.

[86] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[87] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 595–604, 2015.

[88] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[89] Vladimir Naumovich Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

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

[91] Dequan Wang, Zhiqiang Shen, Jie Shao, Wei Zhang, Xiangyang Xue, and Zheng Zhang. Multiple granularity descriptors for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision*, pages 2399–2406, 2015.

[92] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1386–1393, 2014.

[93] Qilong Wang, Peihua Li, and Lei Zhang. G2denet: Global gaussian distribution embedding network and its application to visual recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 3, 2017.

[94] Zhenhua Wang, Xingxing Wang, and Gang Wang. Learning fine-grained features via a cnn tree for large-scale classification. *Neurocomputing*, 275:1231–1240, 2018.

[95] Yunchao Wei, Jiashi Feng, Xiaodan Liang, Ming-Ming Cheng, Yao Zhao, and Shuicheng Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1568–1576, 2017.

[96] Jason Weston, André Elisseeff, Bernhard Schölkopf, and Mike Tipping. Use of the zero-norm with linear models and kernel methods. *Journal of machine learning research*, 3(Mar):1439–1461, 2003.

[97] Lingxi Xie, Richang Hong, Bo Zhang, and Qi Tian. Image classification and retrieval are one. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 3–10. Acm, 2015.

[98] Lingxi Xie, Qi Tian, Richang Hong, Shuicheng Yan, and Bo Zhang. Hierarchical part matching for fine-grained visual categorization. In *Proceedings of the IEEE international conference on computer vision*, pages 1641–1648, 2013.

[99] Lingxi Xie, Liang Zheng, Jingdong Wang, Alan L Yuille, and Qi Tian. Interactive: Inter-layer activeness propagation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 270–279, 2016.

[100] Nie Xuan, Wang Luyao, Liao Qiyu, and Xu Min. Learning enhanced features and inferring twice for fine-grained image recognition. *submitted to Multimedia Tools and Applications*, 2021.

[101] Shulin Yang, Liefeng Bo, Jue Wang, and Linda G Shapiro. Unsupervised template learning for fine-grained object recognition. In *Advances in neural information processing systems*, pages 3122–3130, 2012.

[102] Yi Yang and Deva Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2878–2890, 2013.

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

[104] Kaicheng Yu and Mathieu Salzmann. Statistically-motivated second-order pooling. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[105] Arzhaeva Yulia, Wang Dadong, Alam Md, Shariful, Momeni Saba, Liao Qiyu, Salvado Olivier, Sowmya Arcot, Park Eun-Kee, and Yates Deborah. Optimization and pilot development of pneumoconiosis detection software. *The first progress report for Coal Services Health and Safety Trust, Coal Services Health and Safety Trust Project No. 20656, CSIRO Report No. EP21935, 25 pages*, 22 Feb. 2021.

[106] Fan Zhang, Guisheng Zhai, Meng Li, and Yizhao Liu. Three-branch and mutil-scale learning for fine-grained image recognition (tbmsl-net). *arXiv preprint arXiv:2003.09150*, 2020.

[107] Lianbo Zhang, Shaoli Huang, Wei Liu, and Dacheng Tao. Learning a mixture of granularity-specific experts for fine-grained categorization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8331–8340, 2019.

[108] Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell. Part-based r-cnns for fine-grained category detection. In *European conference on computer vision*, pages 834–849. Springer, 2014.

[109] Richard Zhang. Making convolutional networks shift-invariant again. In *International conference on machine learning*, pages 7324–7334. PMLR, 2019.

[110] Yuting Zhang, Yijie Guo, Yixin Jin, Yijun Luo, Zhiyuan He, and Honglak Lee. Unsupervised discovery of object landmarks as structural representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2694–2703, 2018.

[111] Bo Zhao, Xiao Wu, Jiashi Feng, Qiang Peng, and Shuicheng Yan. Diversified visual attention networks for fine-grained object classification. *IEEE Transactions on Multimedia*, 19(6):1245–1256, 2017.

[112] Heliang Zheng, Jianlong Fu, Tao Mei, and Jiebo Luo. Learning multi-attention convolutional neural network for fine-grained image recognition. In *Int. Conf. on Computer Vision*, volume 6, 2017.

[113] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *European conference on computer vision*, pages 391–405. Springer, 2014.