Faculty of Engineering and Information Technology
University of Technology Sydney

Rail Infrastructure Defect Detection Through Video Analytics

A thesis submitted in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

by

Huaxi Huang

February 2022

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, Huaxi Huang declare that this thesis, is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Electrical and Data Engineering, 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.

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

Production Note:
Signature: Signature removed prior to publication.
Date: 07/02/2022

Acknowledgments

First of all, I would like to express my most profound appreciation to my principal supervisor, Prof. Jian Zhang, for his professional guidance, reliable help, and perpetual support during my pursuit of the Ph.D. degree and three and a half years of research.

I would also like to express my sincere appreciation to my co-supervisor Prof. Qiang Wu and the collaborators: Dr. Junjie Zhang, Dr. Chang Xu, and Dr. Jingsong Xu, for not only their comments on revising my manuscript but also for the insightful guidance, which have incented me to improve my research ability and broaden my research horizon.

I am grateful to my colleagues and friends at the UTS Multimedia and Data Analytics Lab: Yazhou Yao, Xiaoshui Huang, Zhibin Li, Yongshun Gong, Lu Zhang, Anan Du, Lingxiang Yao, Guofeng Mei, Wenbo Xu, Litao Yu and all other labmates. I enjoyed the time we spent together.

Finally and most essentially, I would like to thank my parents. Their selfless support and continuing encouragement helped me defeat the obstacles I encountered during my Ph.D. study.

Huaxi Huang
February 2022 @ UTS

Contents

List of Figures

List of Tables

List of Publications

Papers Published

- **Huaxi Huang**, Junjie Zhang, Jian Zhang, Qiang Wu, Chang Xu, *PTN: A Poisson Transfer Network for Semi-supervised Few-shot Learning*, in Proceeding of the 35th AAAI Conference on Artificial Intelligence (AAAI-21), pp: 1602-1609, 2021.
- **Huaxi Huang**, Junjie Zhang, Jian Zhang, Qiang Wu, Chang Xu, *TOAN: Target-Oriented Alignment Network for Fine-Grained Image Categorization with Few Labeled Samples*, IEEE Transactions on Circuits and System for Video Technology (TCSVT), vol: 32, pp: 853-866, 2022.
- **Huaxi Huang**, Junjie Zhang, Jian Zhang, Jingsong Xu, Qiang Wu. *Low-Rank Pairwise Alignment Bilinear Network For Few-Shot Fine-Grained Image Classification.* IEEE Transactions on Multimedia (TMM), vol: 23, pp: 1666-1680, 2021.
- **Huaxi Huang**, Junjie Zhang, Jian Zhang, Qiang Wu and Jingsong Xu, *Compare More Nuanced: Pairwise Alignment Bilinear Network for Few-Shot Fine-Grained Learning,* in IEEE International Conference on Multimedia and Expo (ICME-19), pp. 91-96, 2019.
- **Huaxi Huang**, Jingsong Xu, Jian Zhang, Qiang Wu, et al. *Railway Infrastructure Defects Recognition using Fine-grained Deep Convolutional Neural Network.* in IEEE International Conference on Digital Image Computing: Techniques and Application, pp: 1-8, 2018.

Abstract

Compared with the traditional railway infrastructure maintenance process, which relies on manual inspection by professional maintenance engineers, inspection through automatic video analytics will significantly improve the working efficiency and eliminate the potential safety concern by reducing physical contact between maintenance engineers and infrastructure facilities. However, the defect does not always have a stable appearance and involves many uncertainties exposed in the clutter environments. On the other hand, various brands of the same devices are used widely on the railway, which shows diverse physical models. Therefore, it creates many challenges to the existing computer vision algorithms for defect detection. In this thesis, two key challenges are abstracted with regard to video/image analytics using computer vision techniques for railway infrastructure defect detection, resulting from the fine-grained defect recognition and the limited labeled learning (few-shot learning). This thesis summarizes the works that have been conducted on utilizing different methods to solve the two challenges.

The first challenge is fine-grained defect recognition. For railway infrastructure defect inspection, damaged or worn equipment defects are usually found in some small parts compared to the whole object. That is, the differences between the defective ones and standard ones are fine-grained. How to find these subtle defects is a fine-grained recognition problem. This thesis proposes a bilinear CNNs model to tackle the defect detection problem, which effectively captures the invariant representation of the dataset and learns high-order discriminative features for fine-grained defect recognition.

Another challenge is the limited labeled data (few-shot learning). In many scenarios, obtaining abundant labeled samples is laborious. For example, in industrial defect detection, most defects exist only in a few common categories, while most other categories only contain a small portion of defects. Moreover, annotating a large-scale dataset of railway infrastructure defects is labor-intensive, which requires high expertise in railway maintenance. Thus, how to obtain an effective model with sparse labeled samples remains an open problem. To address this issue, this thesis proposes a framework to simultaneously reduce the intra-class variance and enlarge the inter-class discrimination for both fine-grained defect recognition and general fine-grained recognition under the few-shot setting. Three models are designed according to this framework, and comprehensive experimental analyses are provided to validate the effectiveness of the models. This thesis further studies the few-shot learning problem by mining the unlabeled information to boost the few-shot learner for defect/general object recognition and proposes a Poisson Transfer Model to maximize the value of the extra unlabeled data through robust classifier construction and self-supervised representation learning.

Chapter 1

Introduction

1.1 Background

Railway power supply infrastructure is one of the essential components in a rail transportation system. Therefore, Railway-Power-Supply-Infrastructure (RPSI) defects detection plays a vital role in railway maintenance and railway safety. Figure 1.1 illustrates an example of the RPSI, which is captured by an infrared camera equipped on a maintenance vehicle in Sydney Trains. There are multiple types of equipment in the image with different sizes and complex backgrounds. Traditional RPSI defects detection task is performed by railway maintenance engineers or related experts manually. Reviewers will check every frame of the infrastructure surveillance videos to find possible flaws in different pieces of power supply equipment. Firstly, they need to locate the specific objects and then focus on these objects to determine whether they are defective. This detection process is low-efficiency, time-consuming, and high-labor costing. More importantly, strong domain knowledge of railway infrastructure is needed while assessing the defects of a device. It is also difficult to obtain a large-scale well-labeled RPSI defect dataset through this labeling process.

The advances in digital video technology, the increasing availability of computing resources and high-resolution cameras, and the growing need for

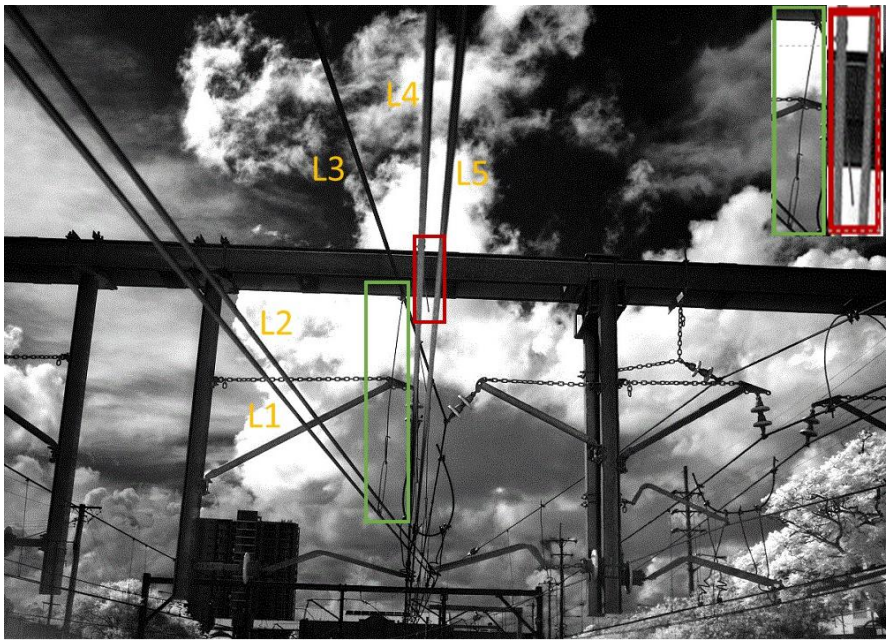


Figure 1.1: An example of Railway Power Supply Infrastructure. We labeled two pieces of Dropper equipment with red and green boxes. The green one presents a standard Dropper equipment where the red one contains a defect Dropper equipment that is disconnected at the end of line 5 (L5). L1 (L2/3/4/5) indicates the power supply cable.

automatic video analytics have sparked great interest in visual recognition and object detection algorithms in computer vision and multimedia. However, developing automated video/image analytics methods in many real-world scenarios is still challenging due to the noise in videos, cluttered background, complex target movement, incomplete or complete occlusions, illumination variances, real-time computing requirements, *etc.* For example, in railway infrastructure defect detection, the changing weather, the train's motion, and the aging of the equipment generate problems when developing robustness and high-performance video analytics technology to reinforce artificial intelligence based decision support for railway infrastructure maintenance.

In automatic industrial defect inspection, there mainly exist three types

of methods: naive-vision-based detection (Otsu 1979, Ng 2006, Chan & Pang 2000, Lowe, Alleyne & Cawley 1998, Kumar & Pang 2002), hand-craft features with machine-learning-based inspection frameworks (Marani, Palumbo, Galietti, Stella & D'Orazio 2016, Jia, Murphey, Shi & Chang 2004, Huang, Hu, Wang, Zhang, Li & Guo 2017) and deep-learning-based detect methods (Jia, Lei, Lin, Zhou & Lu 2016, Lu, Wang, Qin & Ma 2017, Faghih-Roohi, Hajizadeh, Núñez, Babuska & De Schutter 2016). Naive-vision-based industrial inspection methods use traditional image features such as color, gray value , and so on to conduct defects detection. This type of methods (Otsu 1979, Ng 2006, Chan & Pang 2000, Lowe et al. 1998, Kumar & Pang 2002) are fast and efficient in some common industrial processing tasks. But it works lousily when the images are cluttered as well as with complex backgrounds, as Figure 1.1 indicates. Some researchers have attempted to solve these problems from the viewpoint of machine learning with hand-craft features (Marani et al. 2016, Jia et al. 2004, Huang et al. 2017). Nevertheless, with the advent of the Big Data era, massive amounts of data need to be processed in a more precise and robust way which is difficult for traditional machine learning methods.

Nowadays, deep neural networks are widely applied to computer vision and other areas due to their powerful data processing capacity and excellent performance in different applications like AlexNet (Krizhevsky 2014), VGGNet (Simonyan & Zisserman 2014), and ResNet (He, Zhang, Ren & Sun 2016) for image classification, Faster-RCNN (Ren, He, Girshick & Sun 2015) and YOLO (Redmon, Divvala, Girshick & Farhadi 2016) for object detection. There are some works that build deep networks to deal with industrial defects inspection such as (Jia et al. 2016, Lu et al. 2017, Faghih-Roohi et al. 2016, Jia et al. 2016, Lu et al. 2017). However, the adopted datasets are relatively simple, and the deep architectures are comparatively shallow.

Motivated by the above observations, this thesis chooses RPSI for our case studies and explores the feasibility of detecting their defects with different deep neural network models.

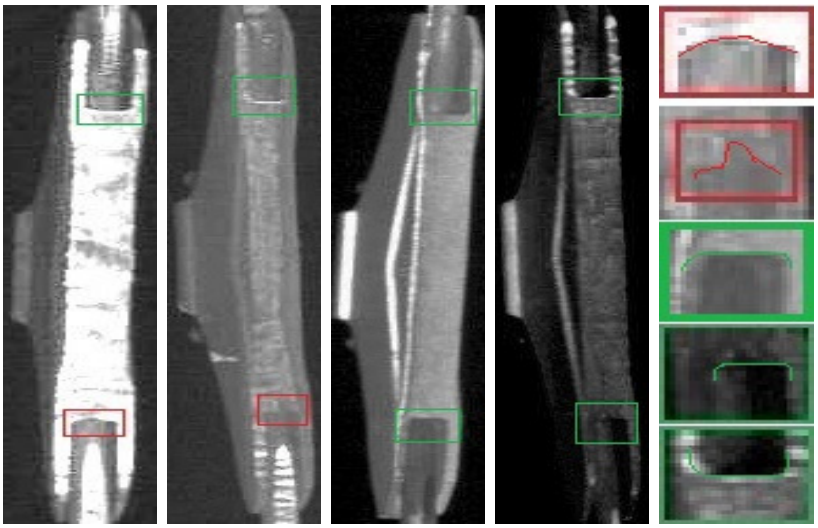
1.2 Research Challenges

Instead of directly performing RPSI defect detection in the full resolution images (video frames), we break the defect detection into two steps: general object (no matter it is defective or not) detection and defective object classification on given cropped objects. The general object detection is well-studied in current deep learning object detection approaches, such as Mask R-CNN (He, Gkioxari, Dollár & Girshick 2017), Faster R-CNN (Ren et al. 2015), and YOLO (Redmon et al. 2016). Therefore, this thesis focuses on the second part. We aim to solve two research challenges regarding defect recognition for RPSI, resulting from the fine-grained defect recognition and the limited labeled learning (few-shot learning).

1.2.1 Fine-grained Defect Recognition

For RPSI, the appearance of component defects and symptomatic conditions will be different, although they have the same functions. Moreover, flaws of damaged or worn equipment are usually found in some small parts. That is, the differences between the defective equipment and the standard one are subtle, while the variations in each type of defects are large. For instance, a typical RPSI equipment named Splice is used to connect two different electric wires that can extend the whole power line, and the most common defect of this equipment is worn at the joints, as seen in Figure 1.2. The green and the red bounding boxes illustrate the standard joints and worn joints, respectively. Distinguishing the subtle differences between standard Splice and defective Splice is imperative while challenging for naive-vision-based detectors and traditional machine-learning-based detectors.

On the other side, even though deep models have made significant progress in defects detection and recognition, considering that RPSI defects recognition is a fine-grained problem, it should be better to use a specific fine-grained model to tackle it. For example, in animal species classification, vehicle type discrimination, and food recognition, where classes in these datasets



(a) Defect (b) Defect (c) Standard (d) Standard (e) Detail

Figure 1.2: An example of Splice defect. (a) and (b) show V-wear defects in connection of wire and splice labeled by red boxes, (c) and (d) show standard Splices. (e) is the magnification of the first four images details. The first row of (e) shows the defect part of (a) which is a V-wear, the second row shows the defect part of (b) and the remaining rows are samples of standard parts of Splice (c)' bottom part and (d) (both top and bottom parts) which we draw with green lines.

have small inter-class variance yet large intra-class gaps. It is reported that fine-grained methods achieve better performance than classical deep neural networks in dealing with these types of tasks (Zhang, Donahue, Girshick & Darrell 2014, Lin, Shen, Lu & Jia 2015*a*, Peng, He & Zhao 2018*a*, Lin, Roy-Chowdhury & Maji 2015*b*, Jaderberg, Simonyan, Zisserman et al. 2015, Cui, Zhou, Wang, Liu, Lin & Belongie 2017*a*, Chen, Wang, Qi, Li & Sun 2017).

Based on the above analysis, in this thesis, we claim that the RPSI defect identification is a typical fine-grained recognition task and try to solve it using fine-grained deep neural networks.

1.2.2 Few-shot Learning for Defect Recognition

Deep neural network models have advanced abilities on various computer vision tasks (Krizhevsky 2014, Simonyan & Zisserman 2014, He et al. 2016, Ren et al. 2015, Redmon et al. 2016), which depends on large-scale training data with well-annotated ground truth. However, it is unrealistic always to obtain such annotation in real-world applications, such as industrial defect detection (Lu et al. 2017) and rare species identification (Wah, Branson, Welinder, Perona & Belongie 2011). Recently, Few-shot learning (FS), as an attempt to address the shortage of training samples, has made significant progress in generic classification tasks (Vinyals, Blundell, Lillicrap, Kavukcuoglu & Wierstra 2016, Snell, Swersky & Zemel 2017, Sung, Yang, Zhang, Xiang, Torr & Hospedales 2018*b*, Liu, Lee, Park, Kim, Yang, Hwang & Yang 2019*a*, Li, Xu, Huo, Wang, Gao & Luo 2019, Li, Wang, Xu, Huo, Gao & Luo 2019). Nonetheless, it is still challenging for current FS models to distinguish the subtle differences for industrial defect categories given limited training data.

In RPSI defect recognition, the labeled image/video data is limited for a specified device due to the high labor-costing and strong domain knowledge requirements in the manual labeling process. For example, according to the Sydney Trains Maintenance Centre, there mainly exist 39 defect types in RPSI maintenance. However, the data distribution of defects is highly imbalanced. The most labeled defect type contains 138 images, while around 20 types of defects only has 1 to 5 images per type. It is not easy to devise or train a decent individual recognition model for each defect with such sparsely labeled data. How to design a model that can effectively handle limited labeled data and identify the unlabeled incoming defect equipment is challenging but necessary for realistic RPSI maintenance. To such a degree, this thesis investigates the feasibility of solving the FS problem for RPSI defect recognition. More specifically, we study the FS RPSI defect recognition from two aspects, fine-grained few-shot learning, and semi-supervised few-shot learning.

As discussed in Section 1.2.1, RPSI defect recognition is a typical fine-

grained classification problem. Therefore, the first research issue of FS RPSI defect recognition is the Fine-Grained Few-Shot image recognition (FGFS). In the general FGFS task, Wei *et al.* (Wei, Wang, Liu, Shen & Wu 2019*a*) proposed the first FGFS model by employing two sub-networks to tackle the problem jointly. The first component is a self-bilinear encoder, which adopts the matrix outer product operation on convolved features to capture subtle image features, while the second one is a mapping network that learns the decision boundaries of the input data. Li *et al.* (Li, Xu, Huo, Wang, Gao & Luo 2019) further replaced the naive self-bilinear pooing as the covariance pooling. Moreover, they designed a covariance metric to generate relation scores. However, self-bilinear pooling (Wei et al. 2019*a*, Li, Xu, Huo, Wang, Gao & Luo 2019) cannot extract comparative features between pairs of images, and the dimension of pooled features is usually large. How to design a better feature extraction framework of FGFS is still an open problem. Moreover, none of the current FGFS models (Wei et al. 2019*a*, Li, Xu, Huo, Wang, Gao & Luo 2019, Zhang & Koniusz 2019) explicitly solve the RPSI defect recognition.

Another research issue of FS RPSI defect recognition is the Semi-Supervised Few-Shot Learning (SSFSL). Compared with collecting labeled defect data, it is much easier to obtain abundant unlabeled data. Therefore, SSFSL (Ren, Triantafillou, Ravi, Snell, Swersky, Tenenbaum, Larochelle & Zemel 2018, Liu, Lee, Park, Kim, Yang, Hwang & Yang 2018, Li, Sun, Liu, Zhou, Zheng, Chua & Schiele 2019, Yu, Chen, Cheng & Luo 2020) is proposed to mine the ancillary knowledge from both labeled and extra unlabeled data to boost few-shot learners. The core challenge in SSFSL is how to explore the auxiliary information from these unlabeled thoroughly. Previous SSFSL works indicate that graph-based models (Liu et al. 2018, Ziko, Dolz, Granger & Ayed 2020) can learn a better classifier than inductive ones (Ren et al. 2018, Li, Sun, Liu, Zhou, Zheng, Chua & Schiele 2019, Yu et al. 2020) since these methods directly model the relationship between the labeled and unlabeled samples during the inference. However, current graph-based models adopt Laplace

learning (Zhu, Ghahramani & Lafferty 2003) as the classifier, and recent research (Calder, Cook, Thorpe & Slepcev 2020) indicates that these classifiers suffer from the underdeveloped message-passing capacity for the labels. On the other hand, most SSFSL methods adapt the pre-trained feature embedding on the training set as the testing embedding. This may lead to the embedding degeneration problem.

1.2.3 Summary

In short, there exist three research issues in this thesis: 1) How to design fine-grained deep neural networks for RPSI defect identification, 2) How to design a fine-grained few-shot framework that can generate more robust feature representation for few-shot PRSI defect recognition, 3) How to design a model to mine the value as much as possible from the abundant unlabeled data to boost the few-shot learners for few-shot RPSI defect recognition. The last two issues belong to the limited labeled learning for RPSI defect recognition, and the first issue belongs to the fine-grained RPSI defect recognition.

1.3 Research Contributions

After analyzing the above challenges and issues, the corresponding solutions are developed in this thesis. The main contributions are summarized as follows:

- To solve the fine-grained defects recognition problem, this thesis is the first to apply the deep fine-grained model to rail infrastructure defects detection. We define the defect recognition task as a two-class fine-grained problem:"defect" or "not-defect." Moreover, cooperated with Sydney Trains, we constructed a Railway Power Supply Infrastructure (RPSI) defects dataset; (Chapter 3)

- This thesis proposes a new bilinear convolutional neural network named Spatial Transformer And Bilinear Low-Rank (STABLR) model. To

solve the high variation within the class, we adopt the Spatial Transformer Network. To achieve more effective performance, we present a Low-Rank Bilinear model. Through experiments, the proposed methods achieve the best performance on the RPSI dataset compared with hand-craft machine learning based methods and classical deep frameworks. (Chapter 3)

- To solve the Fine-Grained Few-Shot (FGFS) defects recognition problem, a two-stage meta-learning-based framework is proposed, which contains a feature alignment module and a high-order pairwise relation extraction module. Three models are instantiated under this framework. In the first FGFS model, we propose a new pairwise bilinear pooling operation to capture the subtle differences between the base and query images. In order to acquire the accurate pairwise bilinear features, we adopt the alignment losses to regularize the embedding features; (Chapter 4)

- In the second FGFS model, a more advanced pairwise pooling operation with a low-rank constraint is proposed. We propose to learn multiple transformations for fusing the input image features. By applying these transformations, the proposed model generates more compact and discriminative features than previous ones. Moreover, a novel alignment mechanism is introduced to encourage the input feature pairs of the bilinear operation to be matched. Instead of solely relying on the alignment losses, we incorporate a feature position re-arrangement layer with the alignment loss to boost the matching performance. (Chapter 4)

- In the third FGFS model, we propose a Target-Oriented Matching Mechanism (TOMM) to learn explicit feature transformations to reduce the biases caused by the intra-class variance. By adopting a cross-correction attention mechanism, the target-oriented matching transfers the support image features to align with the query ones spatially.

Moreover, we propose to aggregate the regional representations into pairwise bilinear pooling through the convolutional channel grouping (GPBP), which devises the second-order features from both global and local views. To our best knowledge, this is the first attempt to adopt group bilinear pooling in FGFS. Comprehensive experiments on four fine-grained benchmark datasets and the RPSI dataset are conducted to investigate the effectiveness of the proposed model, and our model achieves the state-of-the-art performance. (Chapter 4)

- To solve the Semi-Supervised Few-Shot Learning (SSFSL) for the RPSI defects recognition problem, we propose a Poisson Transfer Network (PTN) to improve the capacity of mining the relations between the labeled and unlabeled data for graph-based SSFSL. Moreover, we propose to adapt contrastive learning in the representation learning with extra unlabeled data to improve the generalization of the pre-trained base-class embedding for novel-class recognition. We conduct extensive experiments on two widely-used datasets and the RPSI dataset to investigate the effectiveness of PTN, and PTN achieves state-of-the-art performance. (Chapter 5)

.

1.4 Thesis Structure

The rest of the thesis is structured as follow:

In Chapter 2, the related works for automatic industrial defect detection and recognition are reviewed first. Then we introduce the fine-grained image classification, general few-shot learning, fine-grained few-shot learning, and semi-supervised few-shot learning related to the research issues.

In Chapter 3, the proposed bilinear CNNs model for RPSI defect-recognition and the construction of the RPSI dataset are introduced. Experimental results are presented in this chapter.

In Chapter 4, the two-stage meta-learning framework for Few-Shot Fine-Grained (FGFS) learning is firstly introduced. Then we introduce three instantiated FGFS models. The first FGFS model contains a novel pairwise bilinear pooling and two simple but effective feature alignment losses. The second FGFS model is introduced with a low-rank feature representation and novel alignment module. In the last, we develop the third FGFS model: TOAN. TOAN achieves the state-of-the-art performance on the RPSI dataset and several generic fine-grained datasets with a better feature alignment mechanism and powerful high-order relation extraction.

In Chapter 5, we present the solution of the semi-supervised few-shot learning for both RPSI and general image recognition. We conduct experiments on RPSI and two general datasets to validate the proposed model.

In Chapter 6, we conclude the thesis and outline the possible future directions and some potential solutions.

Chapter 2

Literature Review

This chapter reviews some studies related to automatic defect detection through computer vision and machine learning methods. As discussed in Section 1.2, this thesis mainly focuses on two issues: fine-grained defect recognition and few-shot learning for defect recognition. Therefore, we introduce the existing automatic industrial defect-recognition methods first. Then, we review the general fine-grained image classification methods. These two sections are related to the fine-grained defect-recognition issue. For few-shot learning methods, we investigate the mainstream methods from three aspects: general few-shot learning, fine-grained few-shot learning, and semi-supervised few-shot learning.

2.1 Automatic Industrial Defect Recognition

As we study the automatic defect recognition methods for RPSI data, we review the existing automatic industrial defect detection and recognition first. According to the pre-processing of feature extraction and post-processing of decision making, there mainly exist three types of methods: naive-vision-based methods (Otsu 1979, Ng 2006, Chan & Pang 2000, Lowe et al. 1998, Kumar & Pang 2002), traditional machine-learning-based frameworks (Marani et al. 2016, Jia et al. 2004, Huang et al. 2017, Liu & Li 2021), and deep-

learning-based methods (Jia et al. 2016, Lu et al. 2017, Faghih-Roohi et al. 2016, Li, Zhang, Liang & Wei 2019).

Naive-vision-based industrial inspection methods use the raw information from images as the feature, such as color, gray value, etc. For example, Otsu *et al.* (Otsu 1979) proposed a threshold defects detection method that uses a gray level histogram of bi-modal distribution to inspect candidate images. Ng *et al.* (Ng 2006) improved the performance of Otsu's algorithms by picking up optimal thresholds for both bi-modal and uni-modal distribution of gray histogram from images. Besides these gray value features, Chan *et al.* (Chan & Pang 2000) developed a fabric defects detection method based on the Fourier transform. Lowe *et al.* (Lowe et al. 1998) designed a pipe flaw detection framework using Guided waves. Moreover, Kumar *et al.* (Kumar & Pang 2002) applied Gabor filters to inspect the defects in textile products. Naive-vision-based inspection methods are fast and efficient for some common industrial processing tasks. However, it works lousily when the images are cluttered as well as with complex backgrounds.

To improve the naive-vision-based defect inspection methods, some researchers have attempted to solve the defect detection problems from the viewpoint of machine learning with hand-craft features. For instance, Marani *et al.* (Marani et al. 2016) used clustering techniques, such as K-Means, K-Medoids, and hierarchical clustering, to automatically detect subsurface flaws in composite materials. They adopted thermography images as the input features. Jia *et al.* (Jia et al. 2004) proposed an inspection system using Support Vector Machine (SVM) (Vapnik 2013). Moreover, Huang *et al.* (Huang et al. 2017) designed a real-time mobile phone work-piece surface detects detection framework by using Naive Bayesian (Norvig & Intelligence 2002) and SVM (Vapnik 2013) with Histogram Of Gradient (HOG) (McConnell 1986) and Local Binary Pattern (LBP) (Wang & He 1990) features. Most recently, Liu *et al.* (Liu & Li 2021) proposed a low-rank decomposition model with structural constraints for fabric defect detection. By fusing the original image and an energy image as the input feature, it is reported that the

proposed low-rank model achieves excellent performance on a fabric defect dataset. Nevertheless, with the advent of the Big Data era, large amounts of data need to be processed more precisely and robustly, which is difficult for traditional machine learning methods.

Currently, deep learning models are widely applied to computer vision and other areas due to their strong feature extraction and excellent performance in different applications (Huang, Wu, Xu, Zhong & Zhang 2021), such as AlexNet (Krizhevsky 2014), VGGNet (Simonyan & Zisserman 2014), and ResNet (He et al. 2016) for image classification, Mask R-CNN (He et al. 2017) and YOLO (Redmon et al. 2016) for object detection. There exist some works that design deep neural networks to deal with industrial defects inspection such as (Jia et al. 2016, Lu et al. 2017, Faghih-Roohi et al. 2016, Li, Zhang, Liang & Wei 2019). For example, in (Jia et al. 2016, Lu et al. 2017), the authors used deep models to deal with rolling element bearings and planetary gearboxes data which are text logs about the equipment. However, these data are relatively simple compared to images and videos. Faghih *et al.* (Faghih-Roohi et al. 2016) proposed a deep convolutional neural network (DCNN) for rail surface defects detection, which consists of three convolutional layers and three fully-connected layers. The proposed DCNN model achieves good performance on the rail tracks dataset. However, this deep architecture is comparatively shallow, which cannot capture complex higher-level semantic features of images like fine-grained defect features in cluttered backgrounds.

As discussed in Section 1.2.1, for RPSI defects recognition, damaged or worn equipment defects are usually found in some small parts compared to the whole equipment. Even though deep learning models have achieved significant progress in defects recognition, considering that RPSI defects recognition is a fine-grained problem, it should be better to use a specific fine-grained model to deal with it. Therefore, different from the above deep learning models, in this thesis, we develop deeper neural networks with fine-grained models to capture the complex, subtle features and inference the defectiveness for an input image.

2.2 Fine-grained Image Classification

We claim that RPSI defect recognition is a fine-grained image classification problem. Therefore, fine-grained image classification is closely related to the research of RPSI defect recognition, and we review the fine-grained image classification in this section.

Fine-grained image classification has been a trending topic in computer vision for years, and most traditional fine-grained approaches adopt hand-crafted features as image representations (Xie, Tian, Wang & Zhang 2014, Gao, Tsang & Ma 2014, Zhang, Xiong, Zhou & Tian 2016). However, due to the limited representative capacity of hand-crafted features, the performance of this type of method is moderate. In recent years, deep neural networks have developed advanced abilities in the feature extraction and function approximation (Xu, Jagadeesh & Manjunath 2014, He et al. 2016, Gal & Ghahramani 2016, Yao, Shen, Zhang, Liu, Tang & Shao 2019, Zhang, Wu, Shen, Zhang & Lu 2018*b*, Qiao, Liu, Shen & Yuille 2018, Zhang, Wu, Shen, Zhang & Lu 2018*a*, Zhang, Wang, Wang, Jiang, Xu & Zhao 2021, Li, Liu, Yang, Peng & Zhou 2021), bringing significant progress in fine-grained image classification task (Huang, Li, Xie, Wu & Luo 2016, Xu, Tao, Huang & Zhang 2017, Zhang, Wei, Wu, Cai, Lu, Nguyen & Do 2016, Zhao, Wu, Feng, Peng & Yan 2017, Huang, Li, Xie, Wu & Luo 2016, Yao, Zhang, Zhang, Li & Tian 2016, Peng, He & Zhao 2018*b*, Zhang, Yang, Wang, Hong, Nie & Li 2016, Iscen, Tolias, Gosselin & Jégou 2015, Zhang et al. 2014, Fu, Zheng & Mei 2017, Lin, RoyChowdhury & Maji 2015*a*, Gao, Beijbom, Zhang & Darrell 2016, Kong & Fowlkes 2017, Cui, Zhou, Wang, Liu, Lin & Belongie 2017*b*, Li, Xie, Wang & Gao 2018, Lin, RoyChowdhury & Maji 2018, Suh, Wang, Tang, Mei & Mu Lee 2018, Yu, Zhao, Zheng, Zhang & You 2018, Tan, Yuan, Yu, Wang & Gu 2022).

Deep fine-grained classification approaches can be roughly divided into two groups: regional feature-based methods (Huang, Li, Xie, Wu & Luo 2016, Xu et al. 2017, Zhang, Wei, Wu, Cai, Lu, Nguyen & Do 2016, Zhao et al. 2017, Huang, Li, Xie, Wu & Luo 2016, Yao et al. 2016, Peng et al.

2018*b*, Zhang, Yang, Wang, Hong, Nie & Li 2016, Iscen et al. 2015, Zhang et al. 2014, Fu et al. 2017) and global feature-based methods (Lin, Roy-Chowdhury & Maji 2015*a*, Gao et al. 2016, Kong & Fowlkes 2017, Cui et al. 2017*b*, Li et al. 2018, Lin et al. 2018, Suh et al. 2018, Yu et al. 2018). In fine-grained image classification, the most informative information generally lies in the discriminate parts of an object. Therefore, regional feature-based approaches tend to detect such parts first and then fuse them to form a robust representation of the object. For instance, Zhang *et al.* (Zhang, Wei, Wu, Cai, Lu, Nguyen & Do 2016) firstly combined the R-CNN (Girshick, Donahue, Darrell & Malik 2014) into the fine-grained classifier with a geometric prior, in which the modified R-CNN generates thousands of proposals. The most discriminate ones are then selected for object classification. In (Peng et al. 2018*b*), Peng *et al.* adopted two attention modules to localize objects and choose the discriminate parts simultaneously. A spectral clustering method is then employed to align the parts with the same semantic meaning for the prediction. However, the classification performance of these models relies heavily on the parts localization step. Getting a well-trained part detector needs the input of a large amount of subtle annotated samples, which is infeasible to obtain. Moreover, the sophisticated regional feature fusion mechanism leads to the increasing complexity of the fine-grained classifier.

On the contrary, global feature-based fine-grained methods (Lin, Roy-Chowdhury & Maji 2015*a*, Gao et al. 2016, Kong & Fowlkes 2017, Cui et al. 2017*b*, Li et al. 2018, Lin et al. 2018, Suh et al. 2018, Yu et al. 2018) extract the feature of an entire image without explicitly localizing the object parts. Bilinear CNN model (BCNN) (Lin, RoyChowdhury & Maji 2015*a*) is the first work that adopts matrix outer product operation on the original embedded features to generate a second-order representation for fine-grained classification. Li *et al.* (Li et al. 2018) (iSQRT-COV) further improved the naive bilinear model by using covariance matrices over the last convolutional features as fine-grained features. iSQRT-COV obtained the state-of-the-art performance on both generic and fine-grained datasets.

However, the feature dimensions of the second-order models are the square fold of the naive ones. To reduce the computation complexity, Gao *et al.* (Gao et al. 2016) proposed a compact bilinear pooling operation, which applies Tensor Sketch (Pham & Pagh 2013) to reduce the dimensions. Kong *et al.* (Kong & Fowlkes 2017) introduced a low-rank co-decomposition of the covariance matrix that fatherly decreases the complexity, while Kim *et al.* (Kim, On, Lim, Kim, Ha & Zhang 2017*a*) adopted the Hadamard product to redefine the bilinear matrix outer product and proposes a factorized low-rank bilinear pooling for multimodal learning. Furthermore, Gao *et al.* (Yu et al. 2018) devised a hierarchical approach for fine-grained classification using a cross-layer factorized bilinear pooling operation. Inspired by the flexibility and effectiveness of the Hadamard product for extracting the second-order features between visual features and textual features in VQA tasks (Kim et al. 2017*a*), in this thesis, we propose to adopt different second-order extraction methods for both fine-grained and fine-grained few-shot RPSI defect recognition, including matrix-outer-product-based bilinear pooling and Hadmard-product-based factorized bilinear pooling. We further design a new bilinear pooling operation with semantic grouping for fine-grained few-shot recognition.

2.3 Generic Few-shot Learning

We review the generic few-shot learning and then introduce the fine-grained few-shot learning, as the second research issue of this thesis is RPSI defects recognition under the few-shot setting.

As a representative of the learning methods with limited samples, *e.g.,* weakly supervised learning (Lan, Yuen & Chellappa 2017, Zhang, Wei, Feng, Yang & Huang 2018), active learning (Huang, Zhang, Hu & Zhu 2016, Zhao, Shi, Zhang, Chen & Gu 2019, Gu, Zhai, Deng & Huang 2020, Zhao, Qiu & Sun 2022), and semi-supervised learning (Zhu et al. 2003, Calder & Slepčev 2019), Li *et al.* (Fei-Fei, Fergus & Pietro 2006) firstly introduced few-shot

learning based on the Bayesian theory. Recently, due to the excellent performance of deep neural networks, machine few-shot learning (Vinyals et al. 2016, Snell et al. 2017, Sung et al. 2018*b*, Liu, Lee, Park, Kim, Yang, Hwang & Yang 2019*b*, Simon, Koniusz & Harandi 2022) revives again and achieves significant improvements against previous methods. Previous works of the generic Few-Shot (FS) learning are conducted from various perspectives, such as learning with memory (Munkhdalai & Yu 2017, Santoro, Bartunov, Botvinick, Wierstra & Lillicrap 2016), which leverages recurrent neural networks to store the historical information; learning from fine-tuning (Chen, Liu, Kira, Wang & Huang 2019, Finn, Abbeel & Levine 2017*a*, Rajeswaran, Finn, Kakade & Levine 2019, Ravi & Larochelle 2017), which designs a meta-learning framework to obtain well initial weights for the neural network; learning to compare (Li, Wang, Xu, Huo, Gao & Luo 2019, Li, Xu, Huo, Wang, Gao & Luo 2019, Snell et al. 2017, Sung et al. 2018*b*, Vinyals et al. 2016, Zhang, Li & Cheng 2019), *etc.*

Among these, learning to compare is the most widely used (Gidaris & Komodakis 2018, Hao, He, Cheng, Wang, Cao & Tao 2019, Li, Wang, Xu, Huo, Gao & Luo 2019, Li, Xu, Huo, Wang, Gao & Luo 2019, Snell et al. 2017, Sung et al. 2018*b*, Vinyals et al. 2016, Wertheimer & Hariharan 2019, Wu, Li, Guo & Jia 2019, Zhang & Koniusz 2019, Jiang, Huang, Geng & Deng 2020). In general, learning to compare methods can be divided into two modules: feature embedding and similarity measurement. By adopting the episode training mechanism (Vinyals et al. 2016), these approaches optimize the transferable embedding of both auxiliary data and target data. Then, the query images can be identified by the distance-based classifiers (Hao et al. 2019, Li, Wang, Xu, Huo, Gao & Luo 2019, Liu et al. 2019*a*, Snell et al. 2017, Sung et al. 2018*b*, Vinyals et al. 2016). Currently, (Hao et al. 2019, Li, Wang, Xu, Huo, Gao & Luo 2019, Wu et al. 2019) focused on exploring regional information for an accurate similarity comparison.

Different from learning to compare methods that separate the auxiliary data (meta-training data) into a set of few-shot tasks, some research

works (Qiao et al. 2018, Gidaris & Komodakis 2018, Chen, Liu, Kira, Wang & Huang 2018, Qi, Brown & Lowe 2018) utilize all auxiliary classes to pre-train the few-shot model, which is then adapted to novel-class recognition. For example, Tian *et al.* (Tian, Wang, Krishnan, Tenenbaum & Isola 2020) decoupled the learning procedure into the base-class embedding pre-training and novel-class classifier learning. By adopting multivariate logistic regression and knowledge distillation, the proposed model outperforms many few-shot approaches. We denote these methods (Qiao et al. 2018, Gidaris & Komodakis 2018, Chen et al. 2018, Qi et al. 2018) as transfer-learning-based few-shot models.

However, generic FS methods are not designed to address the high intra-class yet low inter-class variance issue in the fine-grained few-shot problem. In Chapter 4, we aim at tackling the fine-grained defect/image classification from the class variance perspective, *i.e.,* we propose a two-stage framework to capture a more robust representation of images by simultaneously eliminating the intra-class variations through feature alignment and enhancing the inter-class discrimination by adopting the group pair-wise second-order feature extraction. Thus the proposed methods outperform the generic few-shot models. Based on our analysis, the proposed two-stage framework can also be extended to other weakly supervised tasks, such as objection detection (Zhang, Han, Zhao & Zhao 2020, Zhang, Han, Guo & Zhao 2018), localization (Oquab, Bottou, Laptev & Sivic 2015, Peyre, Sivic, Laptev & Schmid 2017), and segmentation (Zhang, Han, Yang & Xu 2018), *etc.*, where only image-level supervision is available. The intra-class and inter-class variances in these tasks can be modeled by the proposed target-oriented matching mechanism and global pair-wise bilinear pooling operation, respectively. In Chapter 5, we further research the semi-supervised few-shot defect/image classification. Inspired by the transfer-learning-based few-shot framework, we adapt this framework to semi-supervised few-shot learning by exploring both unlabeled novel-class data and base-class data to boost the performance of few-shot tasks.

2.4 Fine-grained Few-shot Learning

In this section, we review the fine-grained few-shot learning for image recognition. Wei *et al.* (Wei et al. 2019*a*) proposed a Piecewise Classifier Mappings (PCM) framework for fine-grained image categorization under the few-shot setting. PCM injects the bilinear feature (Lin, RoyChowdhury & Maji 2015*a*) into a group of mapping networks to reduce the dimensionality of the features. A deep distance classifier is then appended to generate the final prediction. SoSN (Zhang & Koniusz 2019) adopts the power normalizing second-order pooling to generate the fine-grained features, and a pair-wise mechanism is then proposed to capture the correlation of support-query pairs. Li *et al.* (Li, Xu, Huo, Wang, Gao & Luo 2019) replaced the bilinear pooling with a covariance pooling operation, and a covariance metric is proposed as the distance classifier. Moreover, (Wertheimer & Hariharan 2019) designs a localization network to generate the foreground and background features for an input image with external bounding box annotations. The bilinear-pooled foreground and background features are concatenated and fed into the classifier. In (Li, Xu, Huo, Wang, Gao & Luo 2019, Wei et al. 2019*a*, Wertheimer & Hariharan 2019, Zhang & Koniusz 2019, Koniusz & Zhang 2021, Zhang, Li & Koniusz 2022), the authors adopted the second-order pooling on the input image itself (noted as self-bilinear pooling) to capture the fine-grained representation.

In our research, three published models are proposed to deal with the fine-grained few-shot problem (Chapter 4). To further leverage the second-order pairwise relationship between support and query images, we propose the pairwise bilinear pooling (Huang, Zhang, Zhang, Wu & Xu 2019, Huang, Zhang, Zhang, Xu & Wu 2021), of which, (Huang et al. 2019) adopts the matrix-outer-product pooling to model pairwise relationships, and (Huang, Zhang, Zhang, Xu & Wu 2021) proposes a factorized Hadamard-product low-rank bilinear operation. However, the high intra-class variance issue is not explicitly addressed in these works (Wei et al. 2019*a*, Zhang & Koniusz 2019, Li, Xu, Huo, Wang, Gao & Luo 2019, Wertheimer & Hariharan 2019, Huang

et al. 2019). The second model (Huang, Zhang, Zhang, Xu & Wu 2021) presents a feature position re-arrangement module for feature alignment with a global MSE loss to boost the discrimination of the fine-grained features. With such feature arrangement module, the model can alleviate the intra-class variance. Different from (Huang, Zhang, Zhang, Xu & Wu 2021), in the third model (Huang, Zhang, Yu, Zhang, Wu & Xu 2021), we explicitly make full use of the spatial dependencies between the support and query pairs. We propose to generate the attention map based on the pair-wise similarities and reformulate the support image spatial features without external supervision. The third model achieves superior performances over (Huang, Zhang, Zhang, Xu & Wu 2021). Moreover, to address the low inter-class variance challenge in fine-grained images, existing models (Wei et al. 2019*a*, Zhang & Koniusz 2019, Li, Xu, Huo, Wang, Gao & Luo 2019, Wertheimer & Hariharan 2019, Huang et al. 2019, Huang, Zhang, Zhang, Xu & Wu 2021) usually adopt second-order feature extraction. Different from prior works, we propose to integrate the local compositional concept representations into global pair-wise bilinear pooling operation in the third model.

Besides the bilinear-based works, generative models (Pahde, Nabi, Klein & Jähnichen 2018, Tsutsui, Fu & Crandall 2019, He & Peng 2018) are also used to synthesize more samples for the support classes. MAML-based model (Zhu, Liu & Jiang 2020) adopts a meta-learning strategy to learn good initial FGFS learners. In (Haney & Lavin 2020), the authors revised the hyper-spherical prototype network (Mettes, van der Pol & Snoek 2019) by maximally separating the classes while incorporating domain knowledge as informative prior. Thanks to the prior knowledge, (Haney & Lavin 2020) achieves good performance in FGFS classification. In (Ruan, Lin, Long & Lu 2021), the authors proposed a Spatial Attentive Comparison Network (SCAN) to fuse the support-query features based on selective comparison.

2.5 Semi-Supervised Few-shot Learning

The last research issue in this thesis is the Semi-Supervised Few-Shot Learning (SSFSL) for RPSI defect recognition. We review SSFSL methods in this section. SSFSL aims to leverage the extra unlabeled novel-class data to improve the few-shot learning. For example, Ren *et al.* (Ren et al. 2018) proposed the first meta-learning-based SSFSL framework by extending the prototypical network (Snell et al. 2017) with unlabeled data to refine class prototypes. LST model (Li, Sun, Liu, Zhou, Zheng, Chua & Schiele 2019) re-trains the base model using the unlabeled data with generated pseudo labels. During the evaluation, it dynamically adds the unlabeled sample with high prediction confidence into testing. In (Yu et al. 2020), TransMatch was proposed to initialize the novel-class classifier with the pre-trained feature imprinting, and then employ MixMatch (Berthelot, Carlini, Goodfellow, Papernot, Oliver & Raffel 2019) to fine-tune the whole model with both labeled and unlabeled data. As closely related research to SSFSL, the transductive few-shot approaches (Liu et al. 2018, Kim, Kim, Kim & Yoo 2019, Ziko et al. 2020, Lazarou, Stathaki & Avrithis 2021) also attempt to utilize unlabeled data to improve the performance of the few-shot learning. These methods adopt the entire query set as the unlabeled data and perform inference on all query samples together. For instance, TPN (Liu et al. 2018) employs graph-based transductive inference to address the few-shot problem, and a semi-supervised extension model is also presented in their work.

Different from TransMatch and meta-learnig based SSFSL models, in Chapter 5, we decouple the SSFSL learning progress into feature embedding fine-tuning and classifier learning. We propose to use unsupervised embedding fine-tuning to transfer the base-class knowledge to novel-class by Contrastive training (He, Fan, Wu, Xie & Girshick 2020, Chen, Kornblith, Norouzi & Hinton 2020). Moreover, a powerful Poisson graph model (Calder et al. 2020) is adopted during the final evaluation stage with both the labeled and unlabeled data. In this way, we can make full use of the unlabeled information.

Chapter 3

RPSI Defect Recognition Using Fine-grained Deep Convolutional Neural Networks

3.1 Introduction

As discussed in Section 1.2.1, Railway Power Supply Infrastructure (RPSI) is an important component for rail transportation. Therefore, RPSI defects recognition plays a vital role in the railway maintenance system. For RPSI defects inspection, defects of damaged or worn equipment are usually found in some small parts compared to the whole object. How to find these subtle defects is a typical fine-grained recognition problem.

Despite the fact that deep neural models (Jia et al. 2016, Lu et al. 2017, Faghih-Roohi et al. 2016, Jia et al. 2016, Lu et al. 2017) have made great progress in some industrial defect detection tasks, considering that RPSI defects recognition is a fine-grained problem, it should be better to use a specific fine-grained model to deal with it. For example, in animal species classification, vehicle type discrimination, and food recognition, where classes in these datasets have small inter-class variations but large intra-class gaps. It is reported that fine-grained methods achieve better performance than

classical deep models (Zhang et al. 2014, Lin, Shen, Lu & Jia 2015*a*, Peng et al. 2018*a*, Lin, RoyChowdhury & Maji 2015*a*, Jaderberg et al. 2015, Chen et al. 2017).

Inspired by the great success of deep fine-grained models in dealing with fine-grained image classification, this paper presents an end-to-end deep network to solve the RPSI defects recognition problem. To our best knowledge, we are the first to apply the deep fine-grained model to railway infrastructure defects recognition. We deal with the challenge that complexes noisy background as well as subtle variations of objects in a fine-grained way. We define defect recognition as a two-class fine-grained problem: "defect" or "not-defect". Following Lin's methodology (Lin, RoyChowdhury & Maji 2015*a*, Lin et al. 2018), we further improve this algorithm using a combination of Spatial Transform and Low-rank operation. We propose a new bilinear deep network named Spatial Transformer And Bilinear Low-Rank (STABLR) model and apply it to the RPSI defects recognition. More specifically, in order to solve the high variation within a class, we adopt the Spatial Transformer Network, and to achieve more effective performance, we present a Low-Rank Bilinear model. Moreover, cooperated with Sydney Trains, we constructed the first RPSI defects dataset. The experimental results demonstrate that the proposed method outperforms both hand-craft features based machine learning methods and classic deep neural network methods.

3.2 RPSI Defect Dataset

The data used in this thesis was collected from Sydney Train Maintenance Center. Instead of performing defect detection in the full resolution images directly, we break the task into two steps: general object (no matter it is defective or not) detection and fine-grained object classification on given cropped objects. This thesis focuses on the second part. We selected the five most common defects on specific equipment as the objective data, which contains 1546 images. In each type of object like Splice, defect objects are

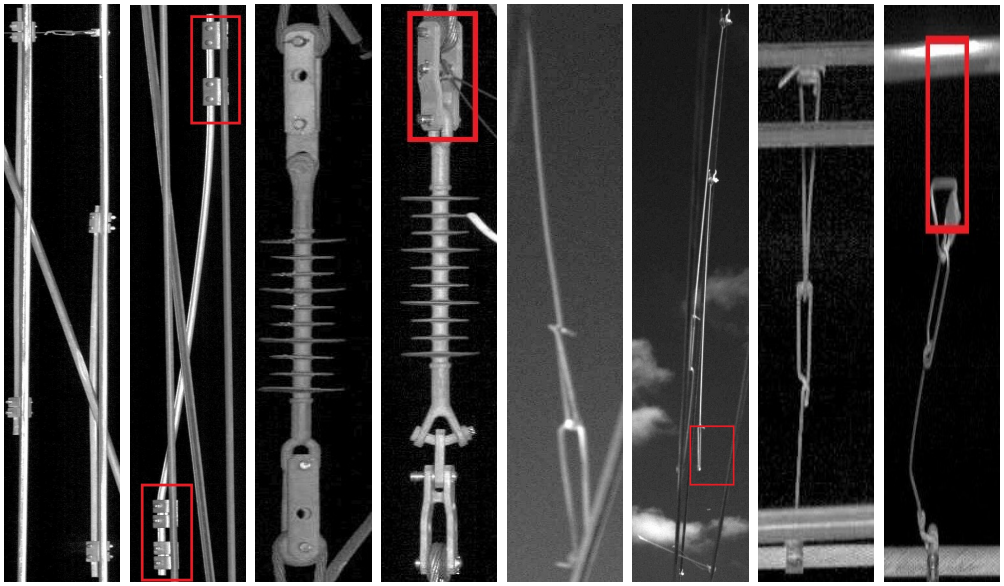
classified into a new class and result in 10 classes (defect and non-defect objects). Therefore our dataset contains both low inter-class variation and high intra-class nonconformity due to the different posture and angle of cameras, illumination variations etc.

More specifically, we picked up 2336 frames from the original video data, which contains power supply equipment and defects. The maximal resolutions of collected images are 2048 × 5400 pixels and 3792 × 2730 pixels, respectively. According to maintenance logs, we manually label the bounding boxes and crop five types of object from each original image and finally produce 1546 images. In summary, we define ten categories including *Splice Standard, Splice V-wear, Knuckle Standard, Knuckle Misinstalled, Kline Standard, Kline Twisted, Dropper1 Standard, Dropper1 Defects, Dropper2 Standard, Dropper2 Broken.* The detailed constituents of this dataset as Table 3.1 shows.

Table 3.1: Railway Power Supply Infrastructure (RPSI) Defects Dataset.

Equipment Name	Class Name	Image Number
Splice	Splice Standard	219
	Splice V-wear	116
Knuckle	Knuckle Standard	332
	Knuckle Misinstalled	36
K-line Insulator	Kline Standard	84
	Kline Twisted	79
Dropper1	Dropper1 Standard	91
	Dropper1 Defects	138
Dropper2	Dropper2 Standard	315
	Dropper2 Broken	136
Summary	10 classes	1546

In Figure 1.2, we give examples of Splice as well as defects samples. For an intuitive understanding of the data set, we list examples of the remaining categories in Figure 3.1.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 3.1: An example of railway infrastructure defects categories. (a) and (b) represent Knuckle, (c) and (d) represent Kline, (e) and (f) represent Dropper1, and (g) as well as (h) represent Dropper2. The detailed parts to distinguish defects are labeled with red boxes.

3.3 Methodology

3.3.1 STABLR Model

Our STABLR model has two parts: Spatial Transformer Network (STN) and Low-Rank Bilinear Convolutional Neural Network (Low-Rank BCNN).

Spatial Transformer Network

One of the challenges of fine-grained classification is the large variation within class due to the pose and location difference of the target objects in the images. Jaberberg *et al.* (Jaderberg et al. 2015) proposed a Spatial Transformer Network (STN in Figure 3.2) that can automatically learn the invariant representation of the original images and locate the target object in the images

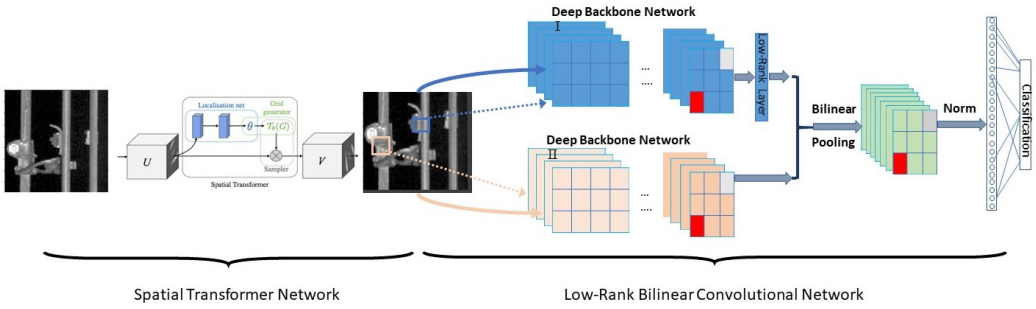


Figure 3.2: The complete architecture of the STABLR model. STABLR can be divided into two parts: STN and Low-Rank BCNN. STN can learn the invariant representation of the dataset and Low-Rank BCNN can capture the fine-grained features of the input images.

at the same time. In our model, we adopt the affine transformer networks as the original paper presented (Jaderberg et al. 2015).

Low-Rank BCNN

Lin *et al.* (Lin, RoyChowdhury & Maji 2015*a*, Lin et al. 2018) proposed a sample and effective architecture for fine-grained classification. It applied outer product operation on deep feature maps to obtain second-order feature descriptors for final classification. A classic Bilinear CNNs model for images recognition can be defined as a quadruple:

$$\mathfrak{B} = (\mathfrak{E}_1, \mathfrak{E}_2, \mathfrak{P}_b, \mathcal{C}), \tag{3.1}$$

where $\mathfrak{B}$ is a bilinear CNNs model, $\mathfrak{E}_1$ and $\mathfrak{E}_2$ represent feature extractor functions, which are specific deep convolutional neural networks like InceptionV3 and InceptionV4 in (Chen et al. 2017). $\mathfrak{P}_b$ is the second-order pooling function and $\mathcal{C}$ represents a classifier. A feature extractor is defined as below:

$$\mathfrak{E} : \mathcal{S} \longrightarrow \mathcal{X} \in \mathcal{R}^{c \times hw}, \tag{3.2}$$

where $\mathcal{S} \in \mathcal{R}^{H \times W \times C}$ represent the images with H height, W width and C color channels. Through function $\mathfrak{E}$, an image is transformed into a tensor

$\mathcal{M} \in \mathcal{R}^{h \times w \times c}$ with c feature channels and h, w indicate the hight and width of the feature map. Then $\mathcal{M}$ is squeezed to a feature matrix $\mathcal{X}$ with $c \times hw$ dimensions. Given two specific functions $\mathfrak{E}_1 : \mathcal{S} \longrightarrow \mathcal{X}_\alpha \in \mathcal{R}^{c_1 \times hw}$ and $\mathfrak{E}_2 : \mathcal{S} \longrightarrow \mathcal{X}_\beta \in \mathcal{R}^{c_2 \times hw}$. Bilinear pooling operator can be described by the following formula:

$$\begin{aligned} \mathfrak{P}_b(\mathcal{S}, \mathfrak{E}_1, \mathfrak{E}_2) &= AVG(\mathcal{X}_\alpha \mathcal{X}_\beta^T), \\ AVG(\mathcal{X}_\alpha \mathcal{X}_\beta^T) &= \frac{1}{hw} \sum_{i=1}^{hw} f_{\alpha,i} f_{\beta,i}^T, \end{aligned} \tag{3.3}$$

where $f_{\alpha,i} \in \mathcal{R}^{c_1}$ and $f_{\beta,i} \in \mathcal{R}^{c_2}$ denote feature vectors at specific location in each feature matrix $\mathcal{X}_\alpha$ and $\mathcal{X}_\beta$ with $i \in [1, hw]$. The pooled feature is a $c_1 \times c_2$ vector. Using matrix outer product, bilinear pooling produces a confertus representation of distinct features from different deep extractors at each location of feature maps in a second-order way.

Notice that an image passed through bilinear pooling in Equation (3.3) will become a $c_1 \times c_2$ vector. Using VGG-D and VGG-M, $c_1 = c_2 = 512$ that the final length of feature is 262,144 (262K). Using these high dimensional features will result in big overhead for time and storage. To address this problem, some researchers adopted low-rank approximation methods to replace the original features (Gao et al. 2016, Kong & Fowlkes 2017, Lin et al. 2018).

Unlike (Gao et al. 2016, Kong & Fowlkes 2017) using complex matrix dimension reduction methods, (Lin et al. 2018) applied PCA (Jolliffe 2011) to activation features before bilinear pooling and achieved as consistent performance as (Gao et al. 2016, Kong & Fowlkes 2017). Following this idea, we propose a simple but effective way to reduce the dimension of activation features:

$$\begin{aligned} & f_{\alpha_{low}} = \theta(f_{\alpha_{high}}), \\ & f_{\alpha_{low}} \in \mathcal{R}^{C_{low}}, \ f_{\alpha_{high}} \in \mathcal{R}^{C_{high}}, \\ & C_{high} > C_{low}. \end{aligned} \tag{3.4}$$

$\theta(\cdot)$ is a convolutional layer with 1×1 kernel size and 1 stride. $f_{\alpha_{high}}$ and

$f_{\alpha_{low}}$ denote original bilinear and low-rank bilinear features, respectively.And C_{high}, C_{low} are the input feature channels and output feature channels of the convolutional layer. With this layer, the proposed bilinear model can automatically learn the dimension reduction rules of the features in an end-to-end way.

According to Lin (Lin, RoyChowdhury & Maji 2015*a*), normalization operation after bilinear pooling could enhance the performance significantly. Without using normalization in (Lin, RoyChowdhury & Maji 2015*a*), we adopt a more robust normalization on bilinear feature vector f in (Lin & Maji 2017):

$$\frac{sign(sqrt(f))\sqrt{|sqrt(f)|}}{\|sign(sqrt(f))\sqrt{|sqrt(f)|}\|_2}. \tag{3.5}$$

At last, f will pass through a fully connected layer as the classifier and get final prediction results. The complete network architecture is in Figure 3.2.

3.3.2 CNN Feature Acquisition

In a bilinear CNNs model (BCNN), how to define proper deep backbone networks is decisive where different deep neural network structure extracts different image features, thus determines the latter classification performance of the B-CNN. In Lin's previous works (Lin, RoyChowdhury & Maji 2015*a*, Lin et al. 2018), BCNN used VGG-M (Chatfield, Simonyan, Vedaldi & Zisserman 2014) and VGG-D (Simonyan & Zisserman 2014), which removed the fully connected classification layer as feature extractors. In (Chen et al. 2017), they adopted InceptionV3 (Szegedy, Vanhoucke, Ioffe, Shlens & Wojna 2016) and InceptionV4 (Szegedy, Ioffe, Vanhoucke & Alemi 2017) as backbone networks.

In Section 3.3.1, an image passed through bilinear pooling in Equation (3.3) will become a $c_1 \times c_2$ vector. Using VGG-D and VGG-M, $c_1 = c_2 = 512$ that the final length of feature is 262,144 (200K), where using InceptionV3 and InceptionV4, $c_1 = 2048$ and $c_2 = 1536$ which result in a 3,145,728 (3000K) feature vector. So the Inception based B-CNN model will have

Table 3.2: CNN features extractors for Railway Infrastructure Defects.

Low-Rank Bilinear Features Extractors			
VGG16		VGG19	
Layer Type	Kernel; Out_dim	Layer Type	Kernel; Out_dim
Conv1_1	3×3; 64	Conv1_1	3×3; 64
Conv1_2	3×3; 64	Conv1_1	3×3; 64
Pool_1	2×2; MaxPooling	Pool_1	2×2; MaxPooling
Conv2_1	3×3; 128	Conv2_1	3×3; 128
Conv2_2	3×3; 128	Conv2_2	3×3; 128
Pool_2	2×2; MaxPooling	Pool_2	2×2; MaxPooling
Conv3_1	3×3; 256	Conv3_1	3×3; 256
Conv3_2	3×3; 256	Conv3_2	3×3; 256
Conv3_3	3×3; 256	Conv3_3	3×3; 256
		Conv3_4	3×3; 256
Pool_3	2×2; MaxPooling	Pool_3	2×2; MaxPooling
Conv4_1	3×3; 512	Conv4_1	3×3; 512
Conv4_2	3×3; 512	Conv4_2	3×3; 512
Conv4_3	3×3; 512	Conv4_3	3×3; 512
		Conv4_4	3×3; 512
Pool_4	2×2; MaxPooling	Pool_4	2×2; MaxPooling
Conv5_1	3×3; 512	Conv5_1	3×3; 512
Conv5_2	3×3; 512	Conv5_2	3×3; 512
Conv5_3	3×3; 512	Conv5_3	3×3; 512
Low-Rank	1×1; 64	Conv5_4	3×3; 512
Feature Maps		Feature Maps	
Bilinear Pooling layer; Output_dim: 64×512			

about 10 times more parameters need to be trained compared to VGG based model, and thus slower than VGG based BCNN model. For our railway infrastructure defects detection task, a real-time and high-accuracy inspection is required. Along these lines, in this thesis, we choose VGG16 (Simonyan & Zisserman 2014) and VGG19 (Simonyan & Zisserman 2014) as the feature extractors, which are more powerful than VGG-M and have fewer parameters than Inception networks. In addition, after passing through a Low-Rank layer, the final dimension of bilinear pooled features is 64×512 (32K). Unlike that (Lin, RoyChowdhury & Maji 2015*a*) resizes the images as 448×448 and (Chen et al. 2017) resizes input images as 229×229, we resize the input images in a 224×224 pixels size since our dataset is relatively simple, and using this way can reduce training time while speeding up detection.

We remove the last fully connected layers of both VGG16 and VGG19 networks as image descriptor extractors. The detailed features extractor architecture is illustrated in Table 3.2.

3.4 Experiments

In this section, we empirically evaluate the STABLR model on the new RPSI dataset. Firstly, we will introduce our experiment in detail. Then we will analyze the experiment results of both hand-craft machine learning based and classic deep neural networks methods.

3.4.1 Experiment Setup

For RPSI video frames, we first resize all image frames into a uniform size as 224 x 224. Then we split the whole dataset into two parts: the train set and test set that 773 images are for training, and the rest are for testing. During the training stage, we randomly crop, rotate, and randomly horizontal flip train images to augment our train sets.

In our experiment, we compare three kinds of methods: hand-craft machine learning based classification, classic deep CNNs model, and deep bilin-

ear methods.

For the hand-craft machine learning based algorithm, we use HOG (Dalal & Triggs 2005) as images feature and SVM (Cortes & Vapnik 1995) as the classifier. In HOG extraction, we set cell size as 8×8 pixels and block size as 2×2 pixels, which are the same as default values in the original paper. By changing the size of the input image, we get 1764 and 8100 dimensions HOG features. We build linear SVM detectors with LIBSVM (Chang & Lin 2011) on MATLAB R2014b platform. And we choose the optimal parameters c, g for SVM using a traversal way.

For classical deep CNNs methods, we use VGG16 (Simonyan & Zisserman 2014) and VGG19 (Simonyan & Zisserman 2014) networks as candidate models. It is impossible to train such deep architectures with our small dataset from scratch. Thus we use the pre-trained VGG16 and VGG19 models in the ImageNet (Russakovsky, Deng, Su, Krause, Satheesh, Ma, Huang, Karpathy, Khosla, Bernstein, Berg & Fei-Fei 2015). We replace the last layer of both VGG16 and VGG19 networks with our classifier layer for defects recognition. We only fine-tune the last layer.

For a more comprehensive analysis of our algorithm, besides original BCNN models (Lin et al. 2018), we design a Spatial Transformer Network for VGG networks (STNVGG16 and STNVGG19) referred to (Jaderberg et al. 2015) that add a STN in front of the VGG networks. In addition, we compare our proposed method with other bilinear models: B-LR models that add our proposed Low-Rank layer to original BCNNs, STNBM models that add STN network in front of BCNN models. In each of above bilinear models, we adopt three types of feature extractors: (16-16) represents two independent VGG16 structures as feature extractors, (19-19) represents two independent VGG19 structures as feature extractors and (16-19) that use a VGG19 and a VGG16 as feature extractors. In summary, we compare 15 methods with our STABLR models.

Both classical deep networks and bilinear models are implemented using PyTorch. For all deep models, we use Adam optimizer to update networks

with Adam' default initial parameters like learning rate and weight decay. Both the training and the testing batch size of all deep models is 8. We then train these models 300 epochs. For the sake of fairness, we freeze the feature extract structures for all deep models and fine-tune the remaining parts of the networks. Deep models experiments of our paper are based on two NVIDIA P4000 GPUs and one NVIDIA P5000 GPU.

3.4.2 Experiments Results Analysis

In this section, we show the results about detection accuracy, average detect time per image, precision, and recall (Powers 2011) of above models for RPSI defect dataset.

In Table 3.3, Avg_Time means average classification time per image for both methods, all models have been run at least five times to obtain average performance. The training is relatively stable. Compared with hand-craft machine learning based methods, deep learning based methods achieve significant improvement in classification accuracy. For example, the classification accuracy of the classic VGG network increased from 78% to 85%, an increase of about 7 percentage points. It indicates that the deep networks induced features have a better discriminative capacity than traditional hand-craft features. We also observe that simply adding the dimensions of the hand-craft features does not significantly improve the classification accuracy, but may reduce the accuracy as Table 3.3 shows. In addition, BCNNs models outperform classical DCNN methods in test accuracy, improving from 87.13% for VGG16 to 92.24% for BCNN(16-19), which indicates that B-CNN models can obtain more subtle features than classical DCNNs. For BCNN models, the fusion of heterogeneous models seems better than the fusion of homogeneous models. This may be caused by that heterogeneous networks can capture more different image features than homogeneous networks. Therefore the bilinear pooled features have stronger fine-grained classification ability.

It can be observed that the Low-Rank layer will make a small reduction of classification performance that the accuracy of B-LR models is approximately

Table 3.3: Detection Results for Railway Supply Power Infrastructure (RPSI) Defects Dataset.

Methods	Feature Dim	Accuracy	Avg_Time(ms)
HOG_SVM	1764	78.33%	2
HOG_SVM	8100	76.39%	10
VGG16	4096	85.81%	39
VGG19	4096	84.51%	40
STNVGG16	4096	87.13%	40
STNVGG19	4096	87.40%	42
BCNN(16-16)	512×512	91.04%	64
BCNN(19-19)	512×512	90.67%	63
BCNN(16-19)	512×512	92.24%	65
B-LR(16-16)	64×512	90.74%	**40**
B-LR(19-19)	64×512	90.40%	**42**
B-LR(16-19)	64×512	91.41%	**41**
STNBM(16-16)	512×512	91.88%	66
STNBM(19-19)	512×512	**91.59%**	68
STNBM(16-19)	512×512	**94.09%**	69
STABLR(16-16)	64×512	**92.67%**	50
STABLR(19-19)	64×512	91.18%	54
STABLR(16-19)	64×512	92.14%	52

one percent lower than the BCNN models. After the STN is integrated, the performance of entire networks is improved that STNVGG models outperform VGG models and STNBM models outperform BCNN models. From Table 3.3, STNBM(16-19) achieves the best recognition performance among all (16-19) bilinear models, STNBM(19-19) reaches the best performance among all (19-19) bilinear models and so is to STABLR(16-16) model. We found that our STABLR can achieve a similar performance as STNBM with 8 times fewer feature dimensions.

Compared with the recognition time per image, HOG_SVM is the fastest

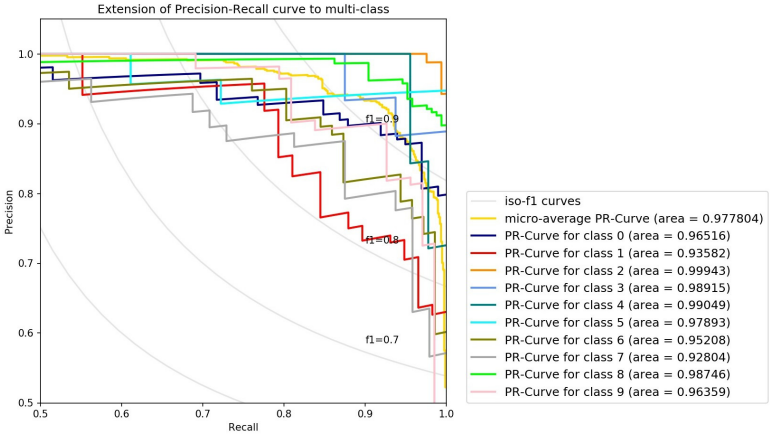
method with 10 ms per image or less due to the relatively low complexity of SVM models in contrast to deep models. VGG models are faster than most bilinear models except for B-LR models. B-LR model is 20 ms faster than BCNN and STNBM models. STNBM models are the slowest models among all methods owing to the huge network parameters. Our STABLR model is 10 ms faster than STNBM and BCNN models with a good classification performance. It indicates that STABLR is the most effective model with good classification performance as well as high recognition speed.

Two important indicators for evaluating the performance of defect detection models are precision and recall. We first use micro-average precision and recall scores (Van Asch 2013) to validate our STABLR models for multi-class classification. As shown in Figure 3.3, micro-average precision scores of STABLR models are 0.977804, 0.964940, 0.970082 for VGG16-VGG16, VGG19-VGG19, and VGG16-VGG19, respectively, which validates the effectiveness of our methods.

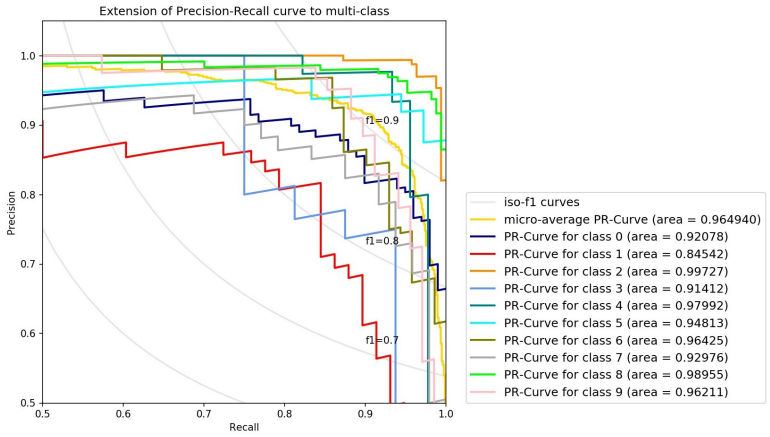
More detailed, we display Precision-Recall (P-R) curves of each class in railway dataset using 1-vs-all strategy in Figure 3.3. It can be observed that in each class, precision and recall scores are close to 1.0, thus indicating the effectiveness of our STABLR models. We also notice that class 0 with the dark blue line and class 1 with the red line which corresponds to Splice equipment, are the hardest class to discriminate, as we mentioned before. This type of flaw has a subtle variation compared to standard equipment. Notice that the iso-f1 curve presents all possible standard F1 scores, which a higher score of F1 means a better model. All STABLR models have F1 scores above 0.8, and the VGG16-VGG16 model gets the highest F1 score.

3.5 Summary

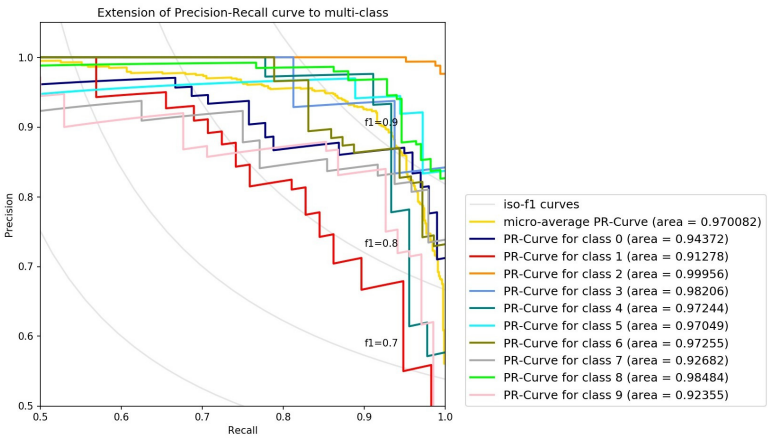
In this chapter, we present a novel railway power supply infrastructure defects detection method STABLR. We use Spatial Transformer Network to learn the invariant representation of the dataset and adopt a simple but ef-



(a) STABLR_VGG16-VGG16



(b) STABLR_VGG19-VGG19



(c) STABLR_VGG16-VGG19

Figure 3.3: P-R curves for all classes in STABLR models.

fective low-rank approximation method to reduce the dimension of original BCNN' activation features. We convert detection into a recognition task and construct a railway infrastructure defects dataset. More importantly, this paper is the first work that applies the fine-grained bilinear CNNs model to the railway infrastructure defects detection problem. Experimental results have shown the effectiveness and high performance of the proposed method. Since STABLR adopts an affine transformer network as the STN module, it is hard to learn a perfect invariant transformation of original images, as affine transformation can not capture complex variations. Therefore, how to design an advanced feature transformation is an open problem. Moreover, STABLR can be applied to other industrial defect-recognition tasks in the future.

Chapter 4

Fine-grained Few-shot RPSI Defect Recognition using Aligned Pairwise Bilinear Framework

4.1 Introduction

In Chapter 3, we investigate RPSI defects recognition using deep models. We claim that the RPSI defects recognition is a typical fine-grained image classification task and therefore designed a fine-grained model to tackle it. However, as discussed in Chapter 1.2.2, a key challenge for RPSI defects recognition is the limited labeled samples. To this end, in this chapter, we study the RPSI defects recognition with limited labeled samples. More specifically, we try to solve this problem under a classic few-shot setting, *i.e.,* Fine-grained Few-shot RPSI defects recognition. We aim to design a unified framework to deal with the Fine-grained Few-shot tasks for both generic image identification and RPSI defects recognition.

Fine-Grained (FG) image recognition aims to identify different sub-categories which belong to the same entry-level category, such as animal identifica-

tion (Khosla, Jayadevaprakash, Yao & Li 2011, Wah et al. 2011) and vehicle recognition (Krause, Stark, Deng & Fei-Fei 2013). Existing FG models (Cui et al. 2017*b*, Krause, Jin, Yang & Li 2015, Lin, RoyChowdhury & Maji 2015*a*, Yu et al. 2018, Zhang et al. 2014, Zhang, Tang & Jia 2018, Wang, Hu, Zhu, Li, Lu, Garibaldi & Li 2019) utilize large-scale and fully-annotated training sets to 'understand' and 'memorize' the subtle differences among classes, thus achieving satisfactory performances in identifying new samples from the same label space. However, in many practical scenarios, it is hard to obtain abundant labeled data for fine-grained classification. For example, in the RPSI defects detection, most defects exist only in a few common categories, while most other categories only contain a small portion of defects. Moreover, annotating a large-scale fine-grained dataset is labor-intensive, which requires high expertise in many fields. Thus, how to obtain an effective model with a small number of labeled samples remains an open problem.

Human beings can learn novel generic concepts with only one or a few samples easily. To simulate this intelligent ability, few-shot machine learning was initially identified by Li *et al.* (Fei-Fei et al. 2006). They propose to utilize probabilistic models to represent object categories and update them with a few training examples. Most recently, inspired by the advanced representation learning ability of deep neural networks, few-shot deep machine learning (Vinyals et al. 2016, Snell et al. 2017, Sung et al. 2018*b*, Liu et al. 2019*b*, Li, Xu, Huo, Wang, Gao & Luo 2019, Li, Wang, Xu, Huo, Gao & Luo 2019) revives and achieves significant improvements against previous methods. However, considering the cognitive process of human beings, preschool students can easily distinguish the difference between generic concepts like the 'Cat' and 'Dog' after seeing a few exemplary images of these animals, but they may be confused about fine-grained dog categories such as the 'Husky' and 'Alaskan' with limited samples. The undeveloped classification ability of children in processing information compared to adults (Brown 1975, John & Cole 1986) indicates that generic few-shot methods cannot cope with the few-shot fine-grained classification task admirably. To this

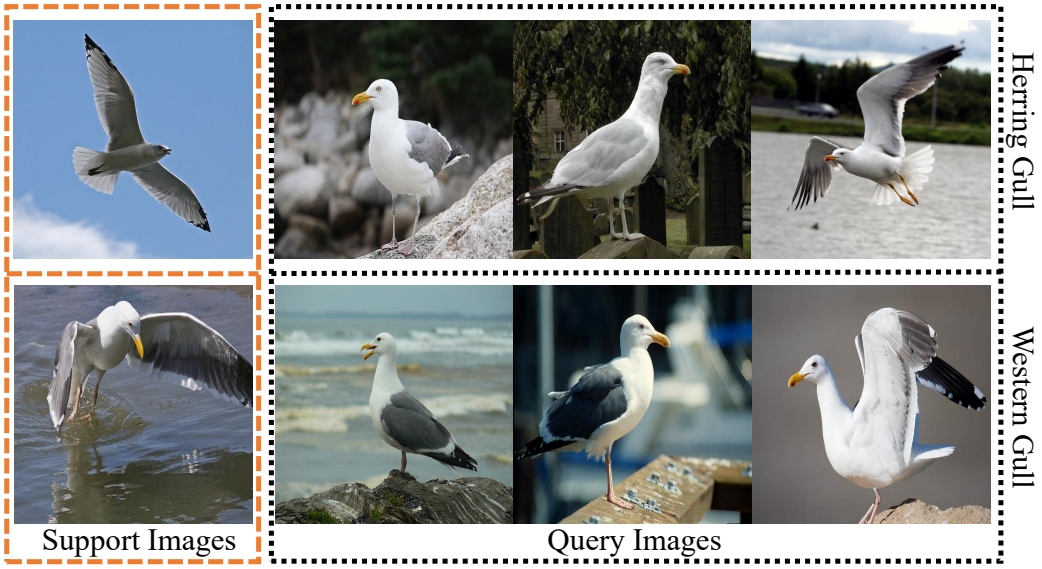


Figure 4.1: The high inter-class visual similarity and significant intra-class variations in FGFS tasks are more rigorous than general FG tasks. Some Herring gull and western gull images have similar visual appearances, which indicates the subtle inter-class variance. However, in each class, gulls present different postures with different backgrounds, which brings significant intra-class variance.

end, in this chapter, we focus on one of the limited sample learning methods for FG tasks, *i.e.,* Fine-Grained image classification under Few-Shot settings (FGFS). We present several solutions for both generic and RPSI image recognition with the limited labeled in a 'developed' way.

The core challenges of the FG problem are the high intra-class variance and low inter-class fluctuations within the datasets (Fu et al. 2017, Lin, Roy-Chowdhury & Maji 2015*a*). The high intra-class variance is mainly caused by different viewpoints, spatial poses, motions, and lighting conditions of different samples in each class. On the other hand, the subtle inter-class variance reflects the taxonomy definition that different fine-grained categories belong to the same entry-level category. The ideal data distribution in a classification problem should posses the low intra-class variance with the high

inter-class variance. If the low inter-class variance is accompanied by high intra-class variance, it can easily lead to inaccurate classification boundaries. With large-scale and fully-annotated datasets available, the high intra-class variance can be somehow relieved through supervised training to obtain a robust representation of each class. However, for FGFS, each class only contains limited labeled samples. As seen from Figure 4.1, in the one-shot bird classification scenario, if the single support (labeled) sample shows a diving gesture, while query (unlabeled) ones are standing. The query-support pairs are not spatially aligned, which can be 'confusing' for classifiers to distinguish them. Therefore, the large intra-class differences bring significant impacts on the representation learning in FGFS. Unfortunately, current FGFS models (Li, Xu, Huo, Wang, Gao & Luo 2019, Wei et al. 2019*a*, Wertheimer & Hariharan 2019, Zhang & Koniusz 2019) rarely focus on this issue.

An effective way to deal with the low inter-class variation in FGFS is to acquire subtle and discriminative image features. Wei *et al.* (Wei, Wang, Liu, Shen & Wu 2019*b*) poposed a deep model named Piece-wise Classifier Mapping (PCM), in which the authors adopt the naive self-bilinear pooling to extract image representations, which widely used in the state-of-the-art FG object classification (Lin, RoyChowdhury & Maji 2015*a*, Cui et al. 2017*b*, Lin et al. 2018). Besides, Li *et al.* (Li, Xu, Huo, Wang, Gao & Luo 2019) proposed a covariance pooling (Li et al. 2018) to learn the image representation of each category. These matrix-outer-product based bilinear pooling operations (Li, Xu, Huo, Wang, Gao & Luo 2019, Wei et al. 2019*b*) could extract the second-order image features and contains more information than traditional first-order features (Lin et al. 2018), and thus achieve better performance on FGFS tasks than generic ones. Both (Li, Xu, Huo, Wang, Gao & Luo 2019) and (Wei et al. 2019*b*) employ bilinear pooling on the input image itself to enhance the information of original features, which noted as the self-bilinear pooling operation. However, when a human identifies the similar objects, she/he tends to compare them thoroughly in a pairwise way, *e.g.,* comparing the heads of two birds first, then the wings and feet last. Therefore, it

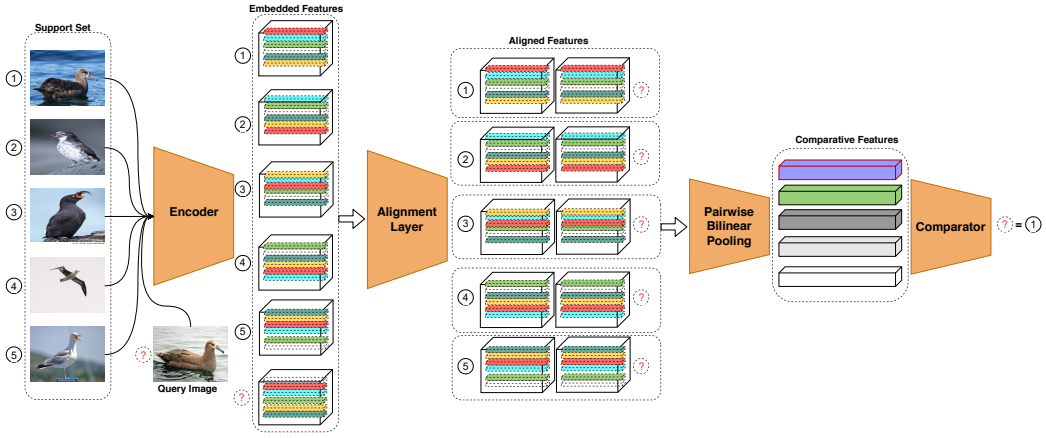


Figure 4.2: The proposed Aligned Pairwsie Bilinear Framework (APBF) in the five-way-one-shot fine-grained image classification. The support set contains five labeled samples for each category (marked with numbers) and the query image labeled with a question mark. The APBF can be divided into four components: Encoder, Alignment Layer, Pairwise Bilinear Pooling, and Comparator. The Encoder extracts coarse features from raw images. Alignment Layer matches the pairs of support and query. Pairwise Bilinear Pooling acts as a fine-grained extractor that captures the subtle features. The Comparator generates the final results.

is natural to enhance the information during the comparing process when dealing with FGFS classification tasks.

To sum up, the high intra-class variation and low inter-class variation are not well addressed in current FGFS models. Therefore, in this chapter, we propose to develop a framework to tackle these two challenges simultaneously. We name the whole framework as **Aligned Pairwise Bilinear Framework** (APBF). By jointly employing the feature alignment transformation to reduce the high intra-class variance and the second-order comparative feature extraction to enlarge the low inter-class discrimination, we explore robust fine-grained relations between each support-query pair. More specifically, the APBF consists of two stages: encoded feature alignment and second-order comparative feature extraction, as indicated in Figure 4.2.

To address the high intra-class variance with limited supervision, we design an embedded feature alignment layer to match the query samples (unlabeled) and support samples (labeled) in the embedded feature space. Therefore, within each class, the sizeable intra-class variance is suppress using the alignment mechanism. To address the low inter-class variance in the FGFS task, we propose a novel pairwise bilinear pooling operation on the aligned support and query samples to extract the comparative second-order images features. Based on the explicit elicitation of correlative information of pair samples, the proposed operation can extract more discriminate features than existing approaches(Wei et al. 2019*b*, Sung et al. 2018*b*, Li, Xu, Huo, Wang, Gao & Luo 2019, Zhang & Koniusz 2019)

Based on the proposed framework APBF, we further develop three models to progressively solve the FGFS image (generic image and RPSI image) recognition. The first model is the **Pairwise Alignment Bilinear Network** (PABN), which is the first work to uncover the fine-grained relations between different support (labeled) and query unlabeled) image pairs. We propose a novel pairwise bilinear pooling operation that adopts a matrix-outer-product operation to extract the second-order comparative features from support-query pairs. Moreover, we instantiate the alignment layer by using two alignment losses to regularize the embedding features. The second model is the **Low-Rank Pairwise Alignment Bilinear Network** (LRPABN), a more advanced pairwise pooling operation with a low-rank constraint is proposed. Instead of directly operating the matrix-outer-product as the PABN model, we propose to learn multiple transformations for fusing the input image features. By applying these transformations, LRPABN generates more compact and discriminative bilinear features than previous ones. Moreover, we introduce a low-rank approximation of the new bilinear model as our final model to further reduce the computation complexity. To improve the feature alignment, a novel alignment mechanism is introduced, we incorporate a feature position re-arrangement layer with the alignment losses. In the third model, we develop a **Target-Oriented Alignment**

Network (TOAN) to tackle the FGFS problem. To reduce the intra-class variance, we propose a target-oriented matching mechanism to reformulate the spatial features of each support image to match the query ones in the embedding space. To enhance the inter-class discrimination, we devise discriminative fine-grained features by integrating local compositional concept representations with the global second-order pooling. Comprehensive experimental results analysis and ablation studies are conducted on four public fine-grained and RPSI datasets, and the proposed TOAN model achieves the state-of-the-art performance compared against PABN, LRPABN, and other compared models.

Next, we will introduce the problem definition of the FGFS task first. Then the three proposed models will be presented separately. The experimental results and analysis will be given in the end.

4.2 Problem Definition

Given a Fine-Grained target dataset $\mathcal{T}$:

$$\begin{aligned} &\mathcal{T} = \left\{\mathcal{B} = \{(\overline{x}_b, \overline{y}_b)\}_{b=1}^{K\times\tilde{C}}\right\} \cup \left\{\mathcal{N} = \{(\overline{x}_v)\}_{v=1}^{V}\right\}, \\ &\overline{y}_b \in \{1, \tilde{C}\}, \overline{x} \in \mathcal{R}^N, \mathcal{B} \cap \mathcal{N} = \emptyset, V \gg K \times \tilde{C}. \end{aligned} \tag{4.1}$$

For the FGFS task, the target data set $\mathcal{T}$ contains two parts: the labeled subset $\mathcal{B}$ and the unlabeled subset $\mathcal{N}$, where samples from each subset are fine-grained images. The model needs to classify the unlabeled data $\overline{x}_v$ from $\mathcal{N}$ according to a few labeled samples from $\mathcal{B}$, where $\overline{y}_b$ is the ground-truth label of sample $\overline{x}_b$. If the labeled data in the target data set contains K labeled images for each of $\tilde{C}$ different categories, the problem is noted as $\tilde{C}$-way-K-shot.

It is far from obtaining an ideal classifier with the limited annotated $\mathcal{B}$. Therefore, FGFS models usually utilize a fully annotated dataset, which has the similar data distribution but disjoint label space with $\mathcal{T}$ as an auxiliary

dataset $\mathcal{A}$:

$$\begin{aligned} &\mathcal{A} = \left\{\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{L}\right\} \cup \left\{\mathcal{Q} = \{(x_j, y_j)\}_{j=1}^{J}\right\}, \\ &y_i, y_j \in \{1, C\}, x \in \mathcal{R}^N, \mathcal{S} \cap \mathcal{Q} = \emptyset, \mathcal{A} \cap \mathcal{T} = \emptyset, \end{aligned} \tag{4.2}$$

where x_i/y_i and x_j/y_j represent images and their corresponding labels.

To make full use of the auxiliary set, we follow the widely used episode training strategy (Vinyals et al. 2016) as our meta-training strategy. Specifically, in each round of training, the auxiliary data set $\mathcal{A}$ is randomly separated into two parts: support data set $\mathcal{S}$, and query data set $\mathcal{Q}$. With setting $L = K \times \tilde{C}$, we can mimic the composition of the target data set in each iteration. Then $\mathcal{A}$ is employed to learn a meta-learner $\mathfrak{F}$, which can transfer the knowledge from $\mathcal{A}$ to target data $\mathcal{T}$. Once the meta-learner is obtained, it can be fine-tuned with labeled target data set $\mathcal{B}$, and finally, classify the samples from $\mathcal{N}$ into their corresponding categories (Vinyals et al. 2016, Sung et al. 2018*b*, Wei et al. 2019*b*, Liu et al. 2019*b*, Huang et al. 2019, Li, Wang, Xu, Huo, Gao & Luo 2019, Li, Xu, Huo, Wang, Gao & Luo 2019).

4.3 Pairwise Alignment Bilinear Network

The first FGFS model under APBF is Pairwise Alignment Bilinear Network (PABN), and the pipeline of PABN is shown in Figure 4.3. Different from traditional few-shot embedding structures (Vinyals et al. 2016, Snell et al. 2017, Sung et al. 2018*b*), we add the fine-grained image feature extractors as shown in the dotted line box, which is our main contribution. In addition, we modify the non-linear comparator (Sung et al. 2018*b*) and apply it to our fine-grained task. Fine-grained feature extractor can be divided into two components: alignment loss regularization and pairwise bilinear pooling layer. The former aims to match the features of the same position in the embedded image features. For example, the features of the bird's head in the target dataset $\mathcal{B}$ should match the query bird's head features from $\mathcal{Q}$. This alignment can reduce the intra-class variance between the query and

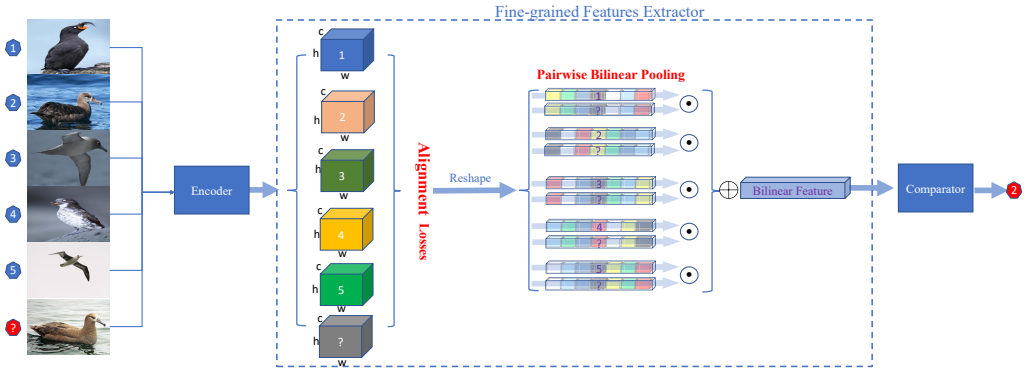


Figure 4.3: The pipeline of PABN under the one-shot fine-grained image recognition setting. There are three parts of PABN: Encoder, Fine-grained Features Extractor, and Comparator. Encoder extracts coarse features from raw images. Fine-grained Extractor captures the subtle features further. Comparator produces the final classification results.

each support-class. The latter pairwise bilinear pooling layer is designed to extract the second-order comparative features from pairs of base images (like samples from $\mathcal{B}$) and query images (like samples from $\mathcal{N}$), which can enlarge the low inter-class variance among different classes.

The pairwise bilinear pooling layer is the core component of the PABN model, which captures the nuanced comparative features of image pairs and therefore decides the relations between base and query images which is crucial to the classifier. However, if the image pairs are not well-matched, these pairwise bilinear pooled features cannot result in the maximum classification performance gain. Thus we propose two feature alignment losses to guarantee the registration between pairs of images. In the next section, we will firstly introduce the pairwise bilinear pooling layer. Then we will present the feature alignment regularization with two alignment losses.

4.3.1 Pairwise Bilinear Pooling

Original Bilinear CNN (Lin, RoyChowdhury & Maji 2015*a*) for the fine-grained image recognition can be defined as a quadruple:

$$\begin{aligned} &\textit{B-CNNs} = (\mathfrak{E}_1, \mathfrak{E}_2, \mathfrak{f}_b, \mathcal{C}), \\ &\mathfrak{E} : \mathcal{I} \longrightarrow \mathcal{X} \in \mathcal{R}^{c \times h \times w}, \\ &\mathfrak{f}_b(\mathcal{I}, \mathfrak{E}_1, \mathfrak{E}_2) = \frac{1}{hw} \sum_{i=1}^{hw} f_{\alpha,i} f_{\beta,i}^T. \end{aligned} \tag{4.3}$$

$\mathfrak{E}_1$ and $\mathfrak{E}_2$ are two encoders. $\mathfrak{f}_b$ is the self-bilinear pooling and $\mathcal{C}$ represents a classifier. $\mathcal{I} \in \mathcal{R}^{H \times W \times C}$ is an image that has H height, W width and C color channels. Through encoder $\mathfrak{E}$, the input image is transformed into a tensor $\mathcal{M} \in \mathcal{R}^{h \times w \times c}$, which has c feature channels and h, w indicate the hight and width of the embedded feature map. Given two specific functions $\mathfrak{E}_1 : \mathcal{S} \longrightarrow \mathcal{X}_\alpha \in \mathcal{R}^{c_1 \times h \times w}$ and $\mathfrak{E}_2 : \mathcal{S} \longrightarrow \mathcal{X}_\beta \in \mathcal{R}^{c_2 \times h \times w}$. $f_{\alpha,i} \in \mathcal{R}^{c_1 \times 1}$ and $f_{\beta,i} \in \mathcal{R}^{c_2 \times 1}$ denote feature vectors at specific location in each feature matrix $\mathcal{X}_\alpha$ and $\mathcal{X}_\beta$ with $i \in [1, hw]$. The pooled feature is a $c_1 \times c_2$ vector. $\mathcal{C}$ is a fully-connected layer with the cross-entropy training loss between self-bilinear feature and image label.

The self-bilinear operates on pairs of embedded features from the same image. However, in our ***pairwise bilinear pooling***, given a pair of image $\mathcal{I}_\mathcal{A}$ (e.g., $\mathcal{I}_\mathcal{A} \in \mathcal{S}$) and image $\mathcal{I}_\mathcal{B}$ (e.g., $\mathcal{I}_\mathcal{B} \in \mathcal{Q}$), an encoder $\tilde{\mathfrak{E}}$, pairwise bilinear pooling $\mathfrak{f}_{pb}$ can be defined as:

$$\begin{aligned} &\mathfrak{f}_{pb}(\mathcal{I}_\mathcal{A}, \mathcal{I}_\mathcal{B}, \tilde{\mathfrak{E}}) = \tilde{\mathfrak{E}}(\mathcal{I}_\mathcal{A}) \tilde{\mathfrak{E}}(\mathcal{I}_\mathcal{B})^T, \\ &\tilde{\mathfrak{E}} : \mathcal{I} \longrightarrow \mathcal{X} \in \mathcal{R}^{c \times hw}. \end{aligned} \tag{4.4}$$

After obtaining these pairwise bilinear vectors, a sigmoid activation is used to generate the relation scores of the compared pairs. The relation scores are then passed to the final comparator.

Note that in our pairwise bilinear pooling, we only have one shared embedding function $\tilde{\mathfrak{E}}$. Different from the self-bilinear pooling that operates on the same input image, pairwise bilinear pooling uses matrix outer product on two disparate samples. The training loss in our bilinear comparator

is Mean Square Error (MSE) loss which regresses the relation score to the images' label similarity as discussed in (Sung et al. 2018*b*). In this way, we can capture the fine-grained second-order comparative features in a pairwise manner.

4.3.2 Feature Alignment Loss

In Equation (4.3), self-bilinear pooling operates on the same image, which means in any location of the embedded features map, the operates features should be aligned. However, our proposed pairwise bilinear pooling conducts on different samples. Thus the encoded features may not always be matched. To overcome this problem, we design two feature alignment losses as follows:

$$Align_{loss_1}(\mathcal{I}_{\mathcal{A}}, \mathcal{I}_{\mathcal{B}}, \tilde{\mathfrak{E}}) = MSE(\tilde{\mathfrak{E}}(\mathcal{I}_{\mathcal{A}}), \tilde{\mathfrak{E}}(\mathcal{I}_{\mathcal{B}})). \tag{4.5}$$

The first $Align_{loss_1}$ loss is a coarse approximation of two embedded image descriptors, which minimizes the Euclidean distances of two transformed features.

$$\begin{aligned} &Align_{loss_2}(\mathcal{I}_{\mathcal{A}}, \mathcal{I}_{\mathcal{B}}, \mathfrak{S}) = MSE(\mathfrak{S}(\mathcal{I}_{\mathcal{A}}), \mathfrak{S}(\mathcal{I}_{\mathcal{B}})), \\ &MSE(\mathfrak{S}(\mathcal{I}_{\mathcal{A}}), \mathfrak{S}(\mathcal{I}_{\mathcal{B}})) = \sum_{1}^{hw} (\mathfrak{S}(\mathcal{I}_{\mathcal{A}}) - \mathfrak{S}(\mathcal{I}_{\mathcal{B}}))^2, \\ &\mathfrak{S}(\mathcal{I}) = \sum_{1}^{c} \tilde{\mathfrak{E}}(\mathcal{I}), \tilde{\mathfrak{E}} : \mathcal{I} \longrightarrow \mathcal{X} \in \mathcal{R}^{c \times hw}. \end{aligned} \tag{4.6}$$

The second $Align_{loss_2}$ loss is a more concise feature alignment loss , where we sum all the raw features along with the third-channel first and then measures the MSE of summed features as Equation (4.6) indicates.

By training with the proposed alignment losses, we encourage the network to learn the matching features automatically. Therefore, the intra-class variance can be somewhat reduced. Moreover, the well-matched support-query pairs can generate a better pairwise bilinear feature, which fatherly enlarges the inter-class variance among different classes.

4.4 Low-Rank Pairwise Alignment Bilinear Network

We introduce the second FGFS model Low-Rank Pairwise Alignment Bilinear Network (LRPABN) in this section. The pipeline of LRPABN is similar to PABN, as illustrated in Figure 4.2 and Figure 4.3, which consists of Encoder, Alignment Layer, Pairwise Bilienar Pooling, and Comparator. Given the support set consisting of five classes with one image per class, an Encoder that is trained with the auxiliary data $\mathcal{A}$ can extract the first-order image features from the raw images, then the Alignment Layer coordinates the embedded feature in support set with the query image feature in pairs. Next, the Pairwise Bilinear Pooling is used to generate the comparative second-order image representation from the embedded feature pairs. Finally, the Comparator assigns the optimal label to the query from support labels in consonance with the similarity between the query and different support classes. LRPABN improves the PABN model from two aspects: **an advanced low-rank pairwise bilinear pooling** to reduce the feature dimension and model complexity, and **a novel alignment layer** consists of a Multi-Layer Perceptron (MLP) feature alignment losses to guarantee the registration of the pairs.

4.4.1 Low-Rank Pairwise Bilinear Pooling

As presented in Section 4.3.1, PABN adopted matrix-outer-product as the pairwise bilinear pooling, which is defined in Equation (4.4). However, the pooled pairwise feature is a $c_1 \times c_2$ vector, which results in a square growth of the original feature dimension. For example, with an embedding network AlexNet (Krizhevsky, Sutskever & Hinton 2012), $c_1 = c_2 = 512$, the pairwise bilinear pooling generates a $512 \times 512 = 262{,}144$-d representation. As reported in (Gao et al. 2016), in such a high-dimensional feature space, less than 5% of dimensions are informative. Moreover, recent research (Gao, Wu, Zhang, Dai, Jia & Harandi 2020) also indicates that the matrix-outer-

product-based bilinear pooling suffers from redundancy and burstiness issues because of the rank-one property of bilinear features. The dimensionality of matrix-outer-product-based bilinear features incites the heavy computational loads as well as burstiness phenomenons.

To overcome this shortcoming of the previous proposed pairwise bilinear pooling, inspired by the Factorized Bilinear Pooling (Kim et al. 2017*a*) applied in the visual-question-answer task, we further propose a low-rank pairwise bilinear pooling operation. For the given $\mathcal{X}_{\mathcal{A}} = \left[\mathbf{x}_1^A, \mathbf{x}_2^A, \cdots, \mathbf{x}_{hw}^A\right]$ and $\mathcal{X}_{\mathcal{B}} = \left[\mathbf{x}_1^B, \mathbf{x}_2^B, \cdots, \mathbf{x}_{hw}^B\right]$ from Equation (4.4), where $\mathbf{x}_j \in \mathcal{R}^{c\times 1}$ stands for any spatial feature vector in $\mathcal{X}$, $j \in [1, hw]$. The low-rank pairwise bilinear can be formulated as:

$$z_j = \left(\mathbf{x}_j^A\right)^T W_i \mathbf{x}_j^B, \tag{4.7}$$

where $W_i \in \mathcal{R}^{c\times c}$ is a projection matrix, $\mathbf{x}_j^A$ and $\mathbf{x}_j^B$ are the feature vectors from $\mathcal{X}_{\mathcal{A}}$ and $\mathcal{X}_{\mathcal{B}}$ in the same position j, separately. Equation (4.7) fuses these feature vectors into a common scalar z_j. Given a set of projection matrices $\mathcal{W} = [W_1, W_2, \cdots, W_n] \in \mathcal{R}^{c\times c\times n}$, the redefined bilinear feature at any position j is $\mathbf{z}_j = [z_1, z_2, \cdots, z_n]^T$, where n is the dimension of this bilinear feature. Then the comparative bilinear representation for the original pairs can be represented as $\mathcal{Z} = [\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_{hw}]$. It is worth noticing that Equation (4.7) is different from Equation (4.4), which adopts projection matrix W_i in learning the bilinear feature. Moreover, in Equation (4.7), the dimension of comparative bilinear feature is n that can be far smaller than $c \times c$ in Equation (4.4). In this way, the model gets a low-rank approximation for the original comparative bilinear feature.

In Equation (4.7), the learned projection $\mathcal{W}$ requires $c\times c\times n$ parameters, where c = 64 and n = 512 in our implementation, *i.e.,* 2,097,152 parameters in total, which requires a large amount of memory footprint, inference time, and computational complexity. To solve this problem, we present a low-rank

approximation of W_i:

$$\begin{aligned} z_j &= \left(\mathbf{x}_j^A\right)^T W_i \mathbf{x}_j^B \\ &= \left(\mathbf{x}_j^A\right)^T U_i V_i^T \mathbf{x}_j^B \\ &= U_i^T \mathbf{x}_j^A \circ V_i^T \mathbf{x}_j^B, \end{aligned} \tag{4.8}$$

where $U_i \in \mathcal{R}^{c\times 1}$ and $V_i \in \mathcal{R}^{c\times 1}$, $\circ$ denotes the Hadamard product. Equation (4.8) is the final form of the proposed low-rank pairwise bilinear pooling, which applies projection matrix and matrix factorization to approximate a full low-rank bilinear model (Equation (4.7)). In Equation (4.8), it needs $2nc$ parameters to generate the pairwise bilinear feature. Therefore, the spatial complexity of the required parameters is reduced from $\mathcal{O}(nc^2)$ to $\mathcal{O}(nc)$. It is worth noting that there are two low-rank approximations applied in the final form of LRPABN. One is to tackle the information redundancy and burstiness issue of the matrix-outer-product-based bilinear pooling (Equation (4.4) to (4.7)), the other is to apply the low-rank matrix factorization to approximate the learned transformations (Equation (4.7) to (4.8)). The proposed LRPABN is different from (Kim et al. 2017*a*, Yu et al. 2018), where (Kim et al. 2017*a*) adopts the factorized bilinear pooling to fuse the multi-modal features, and (Yu et al. 2018) operates on convolutional features of the same image. Our method conducts on pairs of support and query images. To our best knowledge, LRPABN is the first work that extracts the low-rank bilinear feature from pairs of distinct images for FGFS tasks.

Theoretically, the previous proposed model (Huang et al. 2019) belongs to the category of matrix-outer-product bilinear pooling, which has been proved as a similarity-based coding-pooling (Riesenhuber & Poggio 1999, Gao et al. 2020). As (Gao et al. 2020) (Corollary 2) indicates that such bilinear pooling has the unstable dictionary, which is determined by the input pairs, therefore it is inconsistent for all data. This local dictionary can not capture the global geometry of the whole data space, which results in burstiness issues. However, the newly proposed low-rank pairwise bilinear model (4.8) is a type of factorized bilinear coding (Equation (24) in (Gao et al. 2020)),

which can learn a global dictionary from the entire data space in a scalable way, thus achieves better performance than the previous one.

4.4.2 Feature Alignment Layer

The self-bilinear pooling operates on the same image, which means the operating features are entirely aligned in any spatial location of the embedded feature pairs. However, since the proposed pairwise bilinear pooling operates on different inputs, the encoded features may not always be matched. To overcome this obstacle, in the first PABN model, we devise two alignment losses to match the input pairs in the embedding space simultaneously during the training stage, which aims at encouraging the embedding network to generate well-matched features in the testing stage. However, it may be hard to obtain the desired embedding network that fully aligns feature pairs by merely adopting the alignment losses.

Therefore, we design a new feature alignment mechanism inspired by the PointNet (Qi, Su, Mo & Guibas 2017). Given a position transformation $\mathbf{T}$ and the encoded feature $\mathcal{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{hw}]$, the transformed feature can be computed as follows:

$$\begin{aligned} &\mathcal{X}' = \mathcal{X}\mathbf{T}, \\ &s.t.\ \mathbf{T}\mathbf{T}^T = \mathbf{I}, \end{aligned} \tag{4.9}$$

where $\mathbf{T} \in \mathcal{R}^{hw \times hw}$, and $\mathbf{I}$ is an identity matrix. The transformed feature is noted as $\mathcal{X}' = [\mathbf{x}'_1, \mathbf{x}'_2, \cdots, \mathbf{x}'_{hw}]$, in which only the positions of the original feature vectors are rearranged. The transform matrix can be learned with a shallow neural network. Moreover, to ensure the effectiveness of the alignment, we further design two feature alignment losses as follows:

$$Align_{loss_1}(\mathcal{I}_\mathcal{A}, \mathcal{I}_\mathcal{B}, \tilde{\mathfrak{E}}) = MSE(\tilde{\mathfrak{E}}(\mathcal{I}_\mathcal{A}), \tilde{\mathfrak{E}}(\mathcal{I}_\mathcal{B})\mathbf{T}), \tag{4.10}$$

where $\tilde{\mathfrak{E}}$ is the feature encoder. The first $Align_{loss_1}$ loss is a coarse approximation of two embedded image descriptors, which minimizes the Euclidean distances of two transformed features.

$$
\begin{aligned}
&Align_{loss_2}(\mathcal{I}_{\mathcal{A}}, \mathcal{I}_{\mathcal{B}}, \mathfrak{O}) = MSE(\mathfrak{O}(\mathcal{I}_{\mathcal{A}}), \mathfrak{O}(\mathcal{I}_{\mathcal{B}})\mathbf{T}), \\
&\mathfrak{O}(\mathcal{I}) = \sum_{1}^{c} \tilde{\mathfrak{E}}(\mathcal{I}), \tilde{\mathfrak{E}} : \mathcal{I} \longrightarrow \mathcal{X} \in \mathcal{R}^{c \times hw}.
\end{aligned} \tag{4.11}
$$

The second $Align_{loss_2}$ loss is a more concise feature alignment loss. Inspired by the pooling operation, we sum all the embedded features ($\mathcal{X} \in \mathcal{R}^{c \times hw}$) along with the channel dimension ($\mathcal{R}^{c}$) first. And then, we measure the MSE of summed features. By training with the proposed alignment losses, we encourage the model to automatically learn the matching features to generate a better pairwise bilinear feature. It is worth noting that the alignment mechanism utilizes feature position rearrangement matrix $\mathbf{T}$ on one image features ($\tilde{\mathfrak{E}}(\mathcal{I}_{\mathcal{B}})$) to match the target feature ($\tilde{\mathfrak{E}}(\mathcal{I}_{\mathcal{A}})$). $\mathcal{I}_{\mathcal{B}}$ can be either the support or query image, and in our implementation, we choose the support image as $\mathcal{I}_{\mathcal{B}}$. Under the supervision of alignment losses, the model can generate more compactly matched feature pairs compared to the previous method.

In theory, the transformation $\mathbf{T}$ is proposed to project the given feature into a different embedding space. The main purpose of the alignment operation is to match the query and gallery within a common embedding space, and this common space can be either the gallery feature space, the query feature space, or even a new feature space. Our proposed strategy is to project the query-gallery feature pairs into the gallery feature space, since for each C-way-K-shot task, the gallery samples are fixed, transformations only need to be applied to query features.

4.4.3 Comparator

As indicated in Figure 4.3 and Figure 4.2, after passing through the above layers, the pairwise comparative bilinear features are sent to a comparator. This module aims to learn the relations between the query images and support classes. In the one-way-K-shot setting, the support classes are represented

by a single image, where for $\tilde{C}$-way-K-shot setting, the support classes are computed as the sum value of embedded features of K images in each class, *i.e.*, for each query image, the model generates $\tilde{C}$ comparative bilinear features corresponding to each class. For a pair of query image i and support class j, the comparative bilinear feature can be represented as $\mathcal{Z}_{i,j}$, where $\mathcal{Z} = [\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_{hw}]$. The relation score of i and j is computed as:

$$\begin{aligned} &r_{i,j} = \mathcal{C}(\mathcal{Z}_{i,j}), \\ &j = 1, 2, \ldots W; \quad i = 1, 2, \ldots, K \times \tilde{C}, \end{aligned} \tag{4.12}$$

where $\mathcal{C}$ is the comparator, and $r_{i,j}$ is the relation score of query i and class j.

4.4.4 Model Training

The training loss $\mathcal{L}$ in our bilinear comparator is the MSE loss, which regresses the relation score to the images' label similarity. At a certain iteration during the episodic training, there exists m query features and n support class features in total, $\mathcal{L}$ is thus defined as:

$$\mathcal{L} = \sum_{i=1}^{m} \sum_{j=1}^{n} \left(r_{i,j} - \delta\left(y_i = y_j\right)\right)^2, \tag{4.13}$$

where $\delta\left(y_i = y_j\right)$ is the indicator, it equals to one when $y_i = y_j$ and zeroes otherwise. The LRPABN has two optional alignment losses $Align_{loss_1}$ and $Align_{loss_2}$. We back-propagate the gradients when the alignment losses are computed immediately. That is, during the training stage, the model will be updated twice in one iteration.

4.4.5 Network Architecture

The detailed network architecture is shown in Figure 4.4. It consists of three parts: Embedding Network, Low-rank Bilinear Pooling Layer, and Comparator Network.

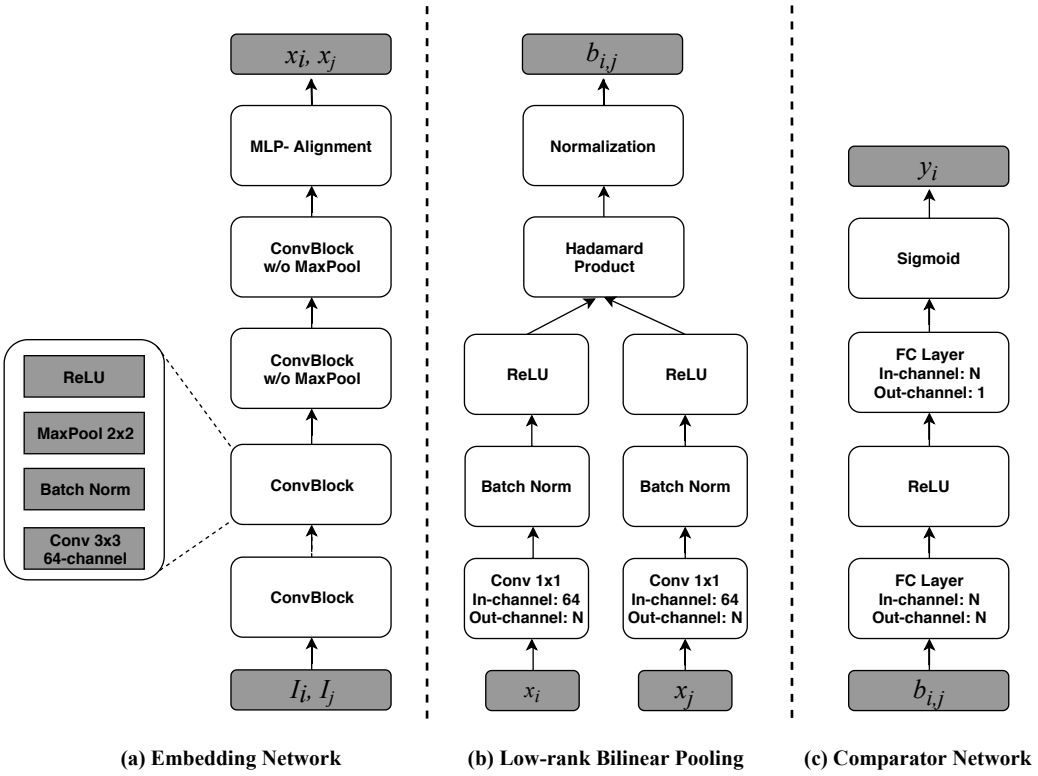


Figure 4.4: Detailed network architectures used in LRPABN. (a) The Embedding network with Alignment Layer. (b) Low-Rank Pairwise Bilinear Pooling Layer. (c) The Comparator. I_i represents the query image, while I_j is the support image, x_i, x_j are the embedded image features and $b_{i,j}$ represents the comparative bilinear feature. y_i is the predicted label by the comparator.

Embedding Network: For a fair comparison with the state-of-the-art generic few-shot and FGFS approaches, we adopt the same encoder structure in (Vinyals et al. 2016, Snell et al. 2017, Sung et al. 2018*b*, Liu et al. 2019*b*, Li, Xu, Huo, Wang, Gao & Luo 2019). It consists of four convolutional blocks, where each block contains a 2D convolutional layer with a 3×3 kernel and 64 filters, a batch normalization layer, and a ReLU layer. Moreover, for the first two convolutional blocks, a 2×2 max-pooling layer is added. For simplicity, we integrate the feature alignment layer into the embedding network as the first-order feature extractor, indicated in Figure 4.4.(a). Unlike the

alignment mechanism used in (Qi et al. 2017, Peng et al. 2018*b*), we devise a simple two layers MLP with the Regulation (4.9). As our alignment mechanism is inspired by PointNet (Qi et al. 2017), which originally adopts a deeper network to learn the transformation matrix $\mathbf{T}$. However, in FGFS, we find that a shallow MLP network $M(\cdot)$ is more efficient in learning a good transformation $\mathbf{T}$. Besides, two optional alignment losses (4.10), (4.11) are applied in the alignment layer to generate the well-matched pairwise features.

Low-rank Bilinear Pooling Layer: For the Low-Rank Pairwise Bilinear Pooling layer in Figure 4.4.(b), we use a convolutional layer with 1×1 kernel followed by the batch normalization and a ReLU layer. The Hadamard product and normalization layers are appended to generate the comparative bilinear features.

Comparator Network: The comparator is made up of two Fully Connected (FC) layers. A ReLU, as well as a Sigmoid nonlinearity layer, are applied to generate the final relation scores, as Figure 4.4.(c) shows.

4.5 Target-Oriented Alignment Network

The third FGFS model is presented in this section. We name this model as Target-Oriented Alignment Network (TOAN). To address the high intra-class variance with limited supervision, we propose a Target-Oriented Matching Mechanism (TOMM), which is inspired by the classical template-based fine-grained methods. As a decent solution to alleviate the high intra-class variance in traditional fine-grained classification, template-based fine-grained methods (Lin, Shen, Lu & Jia 2015*b*, Farrell, Oza, Zhang, Morariu, Darrell & Davis 2011, Yang, Bo, Wang & Shapiro 2012, Yao, Khosla & Fei-Fei 2011) utilize the templates (the closest samples or parts to the class centroid) to align the samples in each class. However, in FGFS, the labeled samples in each class are extremely limited. It is hard to select a good template to represent each category. Therefore, we set each query (testing) sample as the template for all the support (labeled) samples and then adopt

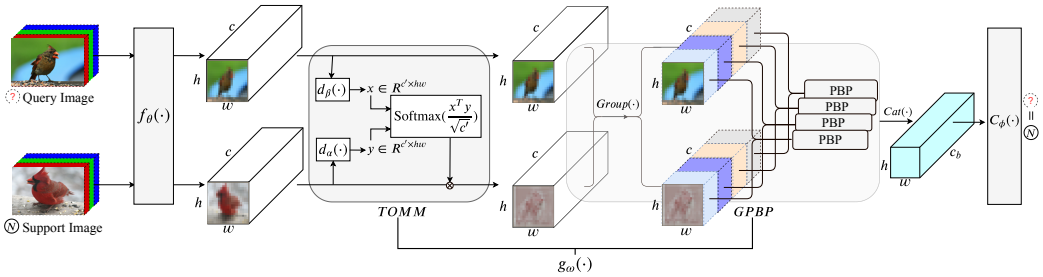


Figure 4.5: The overview of proposed TOAN in the N-way-one-shot FGFS task, other support samples are omitted (replaced by N). The model consists of three parts: the feature embedding f_θ learns the convolved features, the fine-grained relation extractor g_ω gnerates bilinear features, and the comparator C_ϕ maps the query to its ground-truth class. g_ω contains target-orientated matching mechanism (TOMM) and group pairwise bilinear pooling (GPBP), TOMM aims at reformulating the features of support image to match the query image feature in the embedding space through the cross-correction attention mechanism, while GPBP is designed to extract discriminative second-order features by incorporating the channel grouping. With TOMM and GPBP, g_ω learns to generate robust bilinear features from support-query pairs. PBP stands for the pairwise Bilinear Pooling, and we use different colors to indicate the feature maps in GPBP.

the cross-correction attention to transform support features to match query ones. TOMM reformulates the convolutional representations of support images by comparing them with the spatial features of the target query. Different from the conventional self-attention mechanism (Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser & Polosukhin 2017) that operates on the input feature itself, we propose to generate the attention weights in a target-oriented fashion. That is, the similarities of the convolutional features between the support-query pairs are computed first and then converted into a soft-attention map, also noted as the cross-correction attention (Hou, Chang, Ma, Shan & Chen 2019). The spatial features of the support image

are then recomputed as the weighted sum of the whole feature map to reduce the possible variance compared to the query.

To address the low inter-class variance, different from existing works focusing on devising high order features from the global view, we propose to mine the concept local compositionality representation of the bilinear features to enhances their discriminability. Compositionality helps humans learn new concepts from limited samples since it can convert concepts to knowing primitive (Biederman 1987, Hoffman & Richards 1984, Marr & Nishihara 1978). For a convolutional neural network, the channels of convolutional feature usually correspond to different sets of visual patterns (Simon & Rodner 2015, Zhang, Xiong, Zhou, Lin & Tian 2016, Zheng, Fu, Mei & Luo 2017). Therefore, inspired by (Hu, Sun, Saenko & Sclaroff 2019, Zhang, Qi, Xiao & Wang 2017, Zheng et al. 2017, Zheng, Fu, Zha & Luo 2019), we incorporate the compositional concepts into the fine-grained feature extraction by combining the channel grouping operation with the pairwise bilinear pooling, noted as Group Pairwise Bilinear Pooling (GPBP).

It is worth noting that the LRPABN model contains a feature position re-arrangement module for feature alignment with a global MSE loss to boost the discrimination of the fine-grained features. With such feature arrangement module, the model can alleviate the intra-class variance. Different from LRPABN, in this work, we explicitly make full use of the spatial dependencies between the support and query pairs. We propose to generate the attention map based on the pairwise similarities and reformulate the support image spatial features without external supervision. Moreover, to address the low inter-class variance challenge in FG images, existing models (Wei et al. 2019*a*, Zhang & Koniusz 2019, Li, Xu, Huo, Wang, Gao & Luo 2019, Wertheimer & Hariharan 2019, Huang et al. 2019, Huang, Zhang, Zhang, Xu & Wu 2021) usually adopt second-order feature extraction. Different from prior works, we propose to integrate the local compositional concept representations into global pairwise bilinear pooling operation, which fatherly improve the low-rank pairwise bilienar pooling. The pipeline of the proposed

TOAN is shown in Figure 4.5.

Given a support image set $S = \{(x_s, y_s)\}_{s=1}^{K\times C} = \left\{\{x_{1..K}^{(1)}\}, \cdots, \{x_{1..K}^{(C)}\}\right\}$[1], where $x_K^{(t)}$ is the K-th sample in class t, and a query image set $Q = \{x_q\}_{q=1}^{P}$, the learning-to-compare FS model generally consists of two parts: feature embedding module f_θ and the comparator C_ϕ, which can be described as:

$$\mathrm{FS}(S, x_q) = C_\phi \circ f_\theta(S, x_q), \tag{4.14}$$

where $\circ$ denotes the operator of the function composition, f_θ aims to learn the feature embedding of raw images, and C_ϕ is the classifier. However, this framework cannot capture the subtle difference in FG data. Accordingly, FGFS models (Li, Xu, Huo, Wang, Gao & Luo 2019, Wei et al. 2019*a*, Wertheimer & Hariharan 2019, Zhang & Koniusz 2019) incorporate a high-order feature generation module to address the low inter-class variance. However, the large intra-class variance issue is less considered in these methods.

To this end, we propose the Target-Oriented Alignment Network (TOAN) to jointly tackle these issues through a deep fine-grained relation extractor g_ω. Fig. 4.5 illustrates the workflow of the proposed model:

$$\mathrm{TOAN}(S, x_q) = C_\phi \circ g_\omega \circ f_\theta(S, x_q) = C_\phi(Z_{S,q}), \tag{4.15}$$

where the comparator C_ϕ assigns each x_q to its nearest category in S according to the fine-grained relation $Z_{S,q}$, which is generated by applying g_ω on the embedded features $f_\theta(S)$ and $f_\theta(x_q)$ as:

$$\begin{aligned} Z_{S,q} &= g_\omega(f_\theta(S, x_q)) \\ &= \mathrm{GPBP} \circ \mathrm{TOMM}(f_\theta(S, x_q)) \\ &= \mathrm{GPBP}(\{A_1, \cdots, A_C\}, B), \end{aligned} \tag{4.16}$$

where g_ω is composed of two parts, TOMM and GPBP. TOMM is designed to generate the query image feature B and a set of support class prototypes $\{A_1, \cdots, A_t, \cdots, A_C\}$, (*e.g.,* A_t represents the prototype of class t), which are spatially matched in the embedding space. GPBP focuses on extracting the

[1]For a clear understanding, we group the support set S into C subsets.

second-order features from the aligned support class prototypes and query features.

4.5.1 TOMM (Target-Oriented Matching Mechanism)

As discussed previously, it is hard to select a support sample as the template to fully represent its category. Moreover, traditional template-based methods (Lin, Shen, Lu & Jia 2015*b*, Farrell et al. 2011, Yang et al. 2012, Yao et al. 2011) often require a large amount of labeled data. To address the intra-class variance issue with limited training data, we choose each query sample (target) as the template to orient all support samples. Since different sub-classes belong to the same entry-level class, samples from those classes often share similar appearances. The similarities of same parts among sub-classes are higher than those in different parts from the same class. Therefore, we instantiate TOMM using the cross-correction attention. That is, for a pair of support image $x_s^{(t)}$ and query image x_q, TOMM is expressed as:

$$\begin{aligned} &A_s^{(t)}, B = f_\theta(x_s^{(t)}, x_q) \in \mathbb{R}^{c \times hw}, \\ &d_\alpha : A_s^{(t)} \longrightarrow \mathbb{R}^{c' \times hw}, d_\beta : B \longrightarrow \mathbb{R}^{c' \times hw}, \end{aligned} \tag{4.17}$$

where c indicates the channel number of the convolutional feature map and h, w denote the size of the feature map. $A_s^{(t)}$ and B are the embedded support and query features. d_α and d_β are two convolutional sub-networks that capture the task-agnostic similarity between two features ($c' \leq c$). The aligned support feature $A_s^{\prime(t)}$ and the support class prototype A_t is computed as follows:

$$\begin{aligned} (A_s^{\prime(t)})^T &= \text{Softmax}(\frac{d_\beta(B)^T d_\alpha(A_s^{(t)})}{\sqrt{c'}})(A_s^{(t)})^T, \\ A_t &= \frac{1}{K}\sum_{s=1}^{K} A_s^{\prime(t)}, \end{aligned} \tag{4.18}$$

where the Softmax$(\cdot)$ operates in a row-wise way. $A_s^{(t)}$ is transformed to $A_s^{\prime(t)}$, where the similarity of each spatial position between $A_s^{\prime(t)}$ and B reaches maximal. By averaging all aligned features in the given support class, TOMM obtains spatially matched support-class prototypes and query features. Consequently, the intra-class variance in each class is reduced. As is shown in Figure 4.5, the red support bird's embedded features are reformulated according to the query support bird through our TOMM module. It explicitly transforms the 'posture' of support image to match the query ones.

It is worth noting that, for generic FS tasks, since the inter-class variance is relatively large, the cross-correction attention is used to locate the closest features to classify different classes (Hou et al. 2019, Wu et al. 2019, Hao et al. 2019). However, in FG classification, the inter-class variance is relatively subtle, yet much higher intra-class variance exists. Those closest features between query-gallery pairs often perform poorly compared with generic FS. Therefore, we propose to use the cross-correction mechanism to align the feature pairs instead of finding the closest features. Specifically, we explicitly transfer the support image features to match the query ones spatially.

4.5.2 GPBP (Group pairwise Bilinear Pooling)

Semantic compositional information plays an important role in FG tasks, as the discriminative information always exists in some small parts. However, current FGFS models (Wei et al. 2019*a*, Zhang & Koniusz 2019, Wertheimer & Hariharan 2019, Huang, Zhang, Zhang, Xu & Wu 2021) focus on learning the FG features from the global view. Moreover, studies show that high-level convolutional channels represent specific semantic patterns (Zhang et al. 2017, Zheng et al. 2019, Zheng et al. 2017). To this end, we propose to combine compositional concept representations into the second-order feature extraction to generate more discriminative features for FGFS.

GPBP is composed of the convolutional channel grouping operation followed by the pairwise bilinear feature extraction. Given a pair of support class feature $A_t \in \mathbb{R}^{c\times hw}$ and query image feature $B \in \mathbb{R}^{c\times hw}$, we define the

semantic grouping operation as follows:

$$\begin{aligned} &\hat{A} = \text{Group}(A_t), \hat{B} = \text{Group}(B), \\ &\text{Group}(\cdot) : I \longrightarrow [i_1; \cdots; i_k; \cdots; i_N], \\ &I \in \mathbb{R}^{c \times hw}, i_k \in \mathbb{R}^{\frac{c}{N} \times hw}, \end{aligned} \tag{4.19}$$

where $\text{Group}(\cdot)$ converts the original feature into N different groups along the channel dimension, each of these feature groups contains $\frac{c}{N}$ channels, which corresponds to a semantic subspace (Hu et al. 2019). For $\hat{A} = [a_1; \cdots; a_k; \cdots; a_N]$ and $\hat{B} = [b_1; \cdots; b_k; \cdots; b_N]$, we define a bilinear feature z_p of a_k and b_k as:

$$\begin{aligned} z_p &= \text{Bilinear}(a_k, b_k, W_{kp}) \in \mathbb{R}^{1 \times hw} \\ &= [(a_k^1)^T W_{kp} b_k^1, \cdots, (a_k^{hw})^T W_{kp} b_k^{hw}], \end{aligned} \tag{4.20}$$

where $a_k^i, b_k^i \in \mathbb{R}^{\frac{c}{N} \times 1}$ represent the spatial features of a_k and b_k in the given position i. $W_{kp} \in \mathbb{R}^{\frac{c}{N} \times \frac{c}{N}}$ is a projection matrix that fuses a_k^i and b_k^i into a scalar. By adopting W_{kp} on each spatial position of feature pairs, a bilinear feature $z_p \in \mathbb{R}^{1 \times hw}$ is obtained. For each channel group k, GPBP learns $\frac{M}{N}$ projection matrices (M is the dimension of the final bilinear feature), and then we concatenate these scalars to generate a fine-grained relation:

$$Z_k = [z_1; \cdots; z_p; \cdots; z_{\frac{M}{N}}] \in \mathbb{R}^{\frac{M}{N} \times hw}. \tag{4.21}$$

After obtaining the fine-grained relations of each group, we combine them into the final relation Z [2] as:

$$Z = [Z_1; \cdots; Z_k; \cdots; Z_N] \in \mathbb{R}^{M \times hw}, \tag{4.22}$$

where M is the final dimension of Z. Similar to (Kim, On, Lim, Kim, Ha & Zhang 2017*b*, Yu et al. 2018), we adopt a low-rank approximation of W_{kp} to

[2] For brevity, we omit the subscript of $Z_{\hat{A},\hat{B}}$

reduce the number of parameters for regularization:

$$\begin{aligned} z_p &= \text{Bilinear}(a_k, b_k, W_{kp}) \\ &= [(a_k^1)^T W_{kp} b_k^1, \cdots, (a_k^{hw})^T W_{kp} b_k^{hw}] \\ &= [(a_k^1)^T U_{kp} V_{kp}^T b_k^1, \cdots, (a_k^{hw})^T U_{kp} V_{kp}^T b_k^{hw}] \\ &= [U_{kp}^T a_k^1 \odot V_{kp}^T b_k^1, \cdots, U_{kp}^T a_k^{hw} \odot V_{kp}^T b_k^{hw}] \\ &= (U_{kp}^T [a_k^1, \cdots, a_k^{hw}]) \odot (V_{kp}^T [b_k^1, \cdots, b_k^{hw}]) \\ &= (U_{kp}^T a_k) \odot (V_{kp}^T b_k), \end{aligned} \tag{4.23}$$

where $U_{kp} \in \mathbb{R}^{\frac{c}{N} \times 1}$, $V_{kp} \in \mathbb{R}^{\frac{c}{N} \times 1}$, and $\odot$ denotes the Hadamard product.

4.5.3 Comparator

After capturing the comparative bilinear feature of query image i and support class j, the comparator is defined as:

$$\begin{aligned} &C_\phi(\cdot) : Z_{i,j} \in \mathbb{R}^{M \times hw} \longrightarrow \mathbb{R}^1, \\ &j \in \{1, 2, \cdots, C\}, i \in \{1, 2, \cdots, P\}, \end{aligned} \tag{4.24}$$

where C_ϕ learns the distance between the support class j and query image i, that is, for each query i, the comparator generates similarities from C support categories. The query image is assigned to the nearest category. Same as (Sung et al. 2018*b*, Huang et al. 2019), we use the MSE loss as our training loss to regress the predicted label to the ground-truth.

4.5.4 Network Architecture

Feature Embedding Module: In FGFS and FS tasks, f_θ can be any proper convolutional neural network such as ConvNet-64 (Chen et al. 2019, Sung et al. 2018*b*, Vinyals et al. 2016).

Fine-grained Relation Extractor: We show the architecture details of the fine-grained relation extraction module in Figure 4.6. *TOMM*: To construct d_α and d_β, we use a convolutional layer with a 1×1 kernel followed by

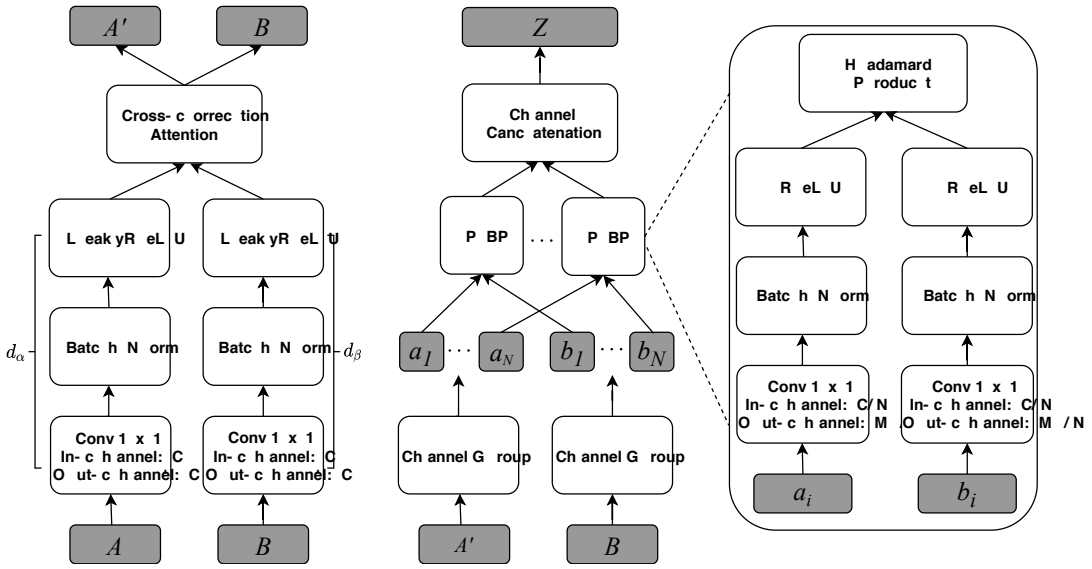


Figure 4.6: The architecture of fine-grained relation extractor, the left figure denotes TOMM, and the right one represents the GPBP operation. A and B indicate the embedded support sample and query sample respectively, Z is the fine-grained relation.

the batch normalization and a LeakyReLU layer. The Cross-correction Attention is implemented using Eq. (4.18). *GPBP*: For the channel grouping, we split the embedded feature map into N groups along the channel dimension. Pairwise bilinear pooling (PBP) consists of a convolutional layer with a 1×1 kernel followed by the batch normalization and a ReLU layer. Then the Hadamard product operation is applied to generate the final bilinear features.

Comparator: The comparator consists of two convolutional blocks and two fully-connected layers. Each block contains a 3×3 convolution, a batch normalization, and a ReLU layer. The activation function of the first fully connected layer is ReLU, where the Sigmoid function is added after the last fully connected layer to generate similarities of input pairs.

4.6 Experiment

In this section, we evaluate the three proposed APBF-based models on the RPSI defect dataset and four widely used fine-grained datasets. First, we give a brief introduction to these datasets. Then we describe the experimental setup in detail. Finally, we analyze the experimental results of the proposed models and compare them with other few-shot learning approaches. For a fair comparison, we conduct two groups of experiments on these data sets. For the first group, we follow the setting, which Wei *et al.* (Wei et al. 2019*b*) used, while for the second group, we follow the newest settings in the recent few-shot methods (Li, Xu, Huo, Wang, Gao & Luo 2019, Li, Wang, Xu, Huo, Gao & Luo 2019).

4.6.1 Datasets

There are five datasets used to investigate the proposed models:

- RPSI Defects is the dataset presented in Section 3.2.Instead of adopting the selected ten classes in Section 3.2, we use the whole 39 categories[3] in the experiment, with 2,336 images. The images per category vary from 10 to 332.
- CUB Birds (Wah et al. 2011) has 200 classes of birds and 11,788 images.
- DOGS (Khosla et al. 2011) has 120 classes of dogs and 20,580 images.
- CARS (Krause et al. 2013) has 196 classes of cars and 16,185 images.
- NABirds (Horn, Branson, Farrell, Haber, Barry, Ipeirotis, Perona & Belongie 2015) has 555 classes of north American birds and 48,562 images.

We designed the first group of experiments to validate the effectiveness of proposed models on generic fine-grained datasets. We then designed the

[3]Due to the IP policy, we omit details about each of these categories.

Table 4.1: The category partion for the four fine-grained datasets, which is the same as PCM (Wei et al. 2019*b*). C_{total} is the original number of categories in the data sets, $C_{\mathcal{A}}$ is the number of categories in separated auxiliary data sets and $C_{\mathcal{T}}$ is the number of categories in target data sets.

dataset	CUB Birds	DOGS	CARS	NABirds
C_{total}	200	120	196	555
$C_{\mathcal{A}}$	150	90	147	416
$C_{\mathcal{T}}$	50	30	49	139

Table 4.2: The class split of five datasets which is the same as (Li, Xu, Huo, Wang, Gao & Luo 2019, Li, Wang, Xu, Huo, Gao & Luo 2019). C_{total} is the original number of categories in the data sets, $C_{\mathcal{A}.Train}$ is the number of training data categories in the auxiliary data sets, $C_{\mathcal{A}.Val}$ is the number of validation data categories in separated auxiliary data sets and $C_{\mathcal{T}}$ is the number of categories in target data sets.

dataset	CUB Birds	DOGS	CARS	NABirds	RPSI Defects
C_{total}	200	120	196	555	39
$C_{\mathcal{A}.Train}$	120	70	130	350	20
$C_{\mathcal{A}.Val}$	30	20	17	66	9
$C_{\mathcal{T}}$	50	30	49	139	10

second group of experiments to further study our models on the RPSI defect dataset and four generic fine-grained datasets. For the first group of experiments, we use the splits of PCM (Wei et al. 2019*b*), as shown in Table 4.1. For the second group, we adopt the dataset splits of Li's (Li, Xu, Huo, Wang, Gao & Luo 2019, Li, Wang, Xu, Huo, Gao & Luo 2019), as indicated in Table 4.2. Both of these methods do not use the RPSI Defects and the NABirds datasets. Thus, for these two datasets only, we do our splits.

4.6.2 Experimental Setup

In each round of training and testing, for the one-shot image classification setting, the support sample number in each class equals 1 (in both $\mathcal{B}$ and $\mathcal{S}$, $K = 1$). Therefore, we use the embedded features of these support samples as the class features, *i.e.*, $\tilde{\mathfrak{E}}(\mathcal{I}_{\mathcal{B}})$. For the few-shot setting, we extract the class features by summing all the embedded support features in each category. In our experiments, we compare the following generic FS models and FGFS methods:

Baselines MatchingNet (Vinyals et al. 2016), ProtoNet (Snell et al. 2017), and RelationNet (Sung et al. 2018*b*) are three exemplary few-shot learning methods. For fair comparisons, we re-implemented these methods by referring to the source codes with our experimental settings. We conducted verification experiments on the MiniImageNet dataset (Vinyals et al. 2016) with the ConvNet-64 backbone (Sung et al. 2018*b*) to validate the correctness of our re-implementations of these three baselines. The classification accuracies of our re-implementations possess no more than 2% fluctuations. These minor margins are mostly caused by the differences in the experimental settings, as (Chen et al. 2019) investigated.

FGFS models Since we aim at tackling the fine-grained classification under the few-shot setting, we selected the most related FGFS methods as main comparisons, including the first FGFS model PCM (Wei et al. 2019*a*). The state-of-the-art models SoSN (Zhang & Koniusz 2019), and FGFS models PABN+ as well as LRPABN$_{cpt}$ for comparison. For PCM, PABN+, and LRPABN$_{cpt}$, we quote the reported results. For other models, the results on the four benchmarks are obtained from their open-sourced models.

Generic FS models It is worth noting that generic FS models can still be applied to fine-grained data. By referring to the first FGFS method (Wei et al. 2019*a*), we selected the most representative ones for comparisons.

That is, we compare our model against DN4 (Li, Wang, Xu, Huo, Gao & Luo 2019) and CovaMNet (Li, Xu, Huo, Wang, Gao & Luo 2019). Results on CUB and NABirds are obtained from their open-sourced models, while others are quoted from reported results.

PABN Family The first FGFS model proposed in this chapter is PABN that uses pairwise bilinear pooling (4.4) without feature alignment transform function (4.9): PABN$_{w/o}$, this model does not use alignment loss on embedded pair features. PABN$_{niv}$ and PABN$_{cpt}$ are the models that adopt the alignment loss $Align_{loss_1}$ and $Align_{loss_2}$ for feature alignment, separately. As Section 4.4.2 discussed, $Align_{loss_1}$ loss is a naive alignment loss where $Align_{loss_2}$ is a more compact loss.

PABN+ models, these models apply the proposed alignment layer into PABN models, which aims to investigate the effectiveness of the proposed feature alignment transform function (4.9): PABN+$_{niv}$ and PABN+$_{cpt}$ are the models that adopt the alignment loss $Align_{loss_1}$ and $Align_{loss_2}$ in the alignment layer (4.9). PABN+$_{cons}$ adopts Cosine loss on the embedded features in the alignment layer (4.9).

LRPABN Family We replaced the naive pairwise bilinear pooling (4.4) with the proposed low-rank bilinear pooling (4.8), and apply the proposed novel feature alignment layer (4.9) into the LRPABN models: LRPABN$_{niv}$ and LRPABN$_{cpt}$, which use the alignment loss $Align_{loss_1}$ and the loss $Align_{loss_2}$ in the alignment layer, respectively.

TOAN Family First of all, we added the proposed matching mechanism TOMM to FS baseline models to investigate its effectiveness for FGFS tasks, noted as FS+TOMM, where FS can be any one of the baselines. Similarly, GPBP is also plugged into the RelationNet, noted as RelationNet+GPBP. To investigate the grouping function, we replaced the proposed function by a 1×1 convolutional layer with a group parameter (Zhang et al. 2017), noted as TOAN-GP*. Moreover, we removed the task-agnostic transformation $d(\cdot)$ in

TOMM, noted as TOAN-$w/o\ d(\cdot)$, where pairwise similarities are computed directly based on the embedded support feature A_t and query feature B. We also replaced the backbone ConvNet-64 with the deeper ResNet-256(Li, Wang, Xu, Huo, Gao & Luo 2019) to study the influence of different backbones, noted as TOAN:ResNet. Finally, we used a larger 224×224 input image size with different backbones to study the effects of the input size for TOAN, noted as TOAN_224 and TOAN:ResNet_224, respectively.

In the first experiment, the LRPABN models are compared with RelationNet, PCM, and our previous proposed PABN models. We follow the data splits (Table 4.1) of PCM and PABN. All of these approaches do not contain the validation data set.

In the second experiment, besides the RelationNet, PABN+ models, and the proposed LRPABN models, we compare the newest state-of-the-art few-shot method DN4 and the newest FGFS approach CovaMNet. To fair compare, we use the same data splits (Table 4.2) and the training strategy of DN4 and CovaMNet.

In the third experiment, we mainly analyze the proposed TOAN model with the same data splits (Table 4.2). We compare the TOAN models against PCM, PABN+, LRPABN$_{cpt}$, SoSN, MatchingNet, ProtoNet, RelationNet, CovaMNet, and DN4.

For all the comparing methods, we conducted both five-way-one-shot and five-way-five-shot classification experiments. In the training stage of the first group of experiments, both five-way-one-shot and five-way-five-shot experiments have 15 query images, which means there are $15 \times 5 + 1 \times 5 = 80$ images and $15 \times 5 + 5 \times 5 = 100$ images in each mini-batch, respectively. For the testing stage, we followed the RelationNet (Sung et al. 2018*b*) that has one query for five-way-one-shot and five queries for five-way-five-shot in each mini-batch. In both the training and testing stages of the second group of experiments (experiment 2 and 3), we randomly select 15 and 10 queries from each category for the five-way-one-shot and five-way-five-shot settings, which is the same setting with (Li, Xu, Huo, Wang, Gao & Luo 2019, Li,

Wang, Xu, Huo, Gao & Luo 2019).

For fair comparisons, we select the optimal models using the same validation strategies as (Sung et al. 2018*b*) for the first group of experiments and (Li, Wang, Xu, Huo, Gao & Luo 2019, Li, Xu, Huo, Wang, Gao & Luo 2019) for the second group of experiments, separately. In the first group, we randomly sample and construct 100,000 episodes to train the LRPABN and PABN+ models. In each episode, there only contains one learning task, while in the second group, we randomly select 10,000 episodes for training, and in each episode, 100 tasks are randomly batched to train the models. For LRPABN models, we set the dimension of the pairwise bilinear feature as 512, where the feature dimension of PABN and PABN+ is $64 \times 64 = 4096$. In training, the learning rate of parameters is decayed by 0.5 every 10,000 epochs using the StepLR schedule in PyTorch. We resize all the input images from all data sets to 84×84. All experiments use Adam optimize method with an initial learning rate of 0.001, and all models are trained end-to-end from scratch. More experimental details can be referred to in our published papers (Huang et al. 2019, Huang, Zhang, Zhang, Xu & Wu 2021, Huang, Zhang, Yu, Zhang, Wu & Xu 2021).

4.6.3 Experimental Results for PABN and LRPABN on Generic Fine-grained Datasets

In the first group of experiments, we compute both one-shot and five-shot classification accuracies on the four public fine-grained data sets by averaging 10,000 episodes in testing. We show the experimental results of 10 compared models in Table 4.3. As the table shows, the proposed LRPABN and PABN models achieve significant improvements on both one-shot and five-shot classification tasks on all data sets compared to the state-of-the-art FGFS methods and generic few-shot methods, which indicates the effectiveness of the proposed APBF framework.

More specifically, the LRPABN, PABN+, and PABN models both obtain around 10% to 30% higher in classification accuracy than PCM (Wei et al.

Table 4.3: Few-shot classification accuracy (%) comparisons on four fine-grained data sets. The second-highest-accuracy methods are highlighted in blue color. The highest-accuracy methods are labeled with the red color. '-' denotes not reported. All results are with 95% confidence intervals where reported.

Methods	CUB Birds		CARS		DOGS		NABirds	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
PCM (Wei et al. 2019*b*)	42.10±1.96	62.48±1.21	29.63±2.38	52.28±1.46	28.78±2.33	46.92±2.00	-	-
RelationNet	63.77±1.37	74.92±0.69	56.28±0.45	68.39±0.21	51.95±0.46	64.91±0.24	65.17±0.47	78.35±0.21
PABN$_{w/o}$	65.99±1.35	76.90±0.21	55.65±0.42	67.29±0.23	54.77±0.44	65.92±0.23	67.23±0.42	79.25±0.20
PABN$_{niv}$	65.04±0.44	76.46±0.22	55.89±0.42	68.53±0.23	54.06±0.45	65.93±0.24	66.62±0.44	79.31±0.22
PABN$_{cpt}$	**66.71±0.43**	76.81±0.21	56.80±0.45	68.78±0.22	**55.47±0.46**	66.65±0.23	67.02±0.43	79.02±0.21
PABN+$_{niv}$	66.68±0.42	76.83±0.22	55.35±0.44	67.67±0.22	54.51±0.45	66.60±0.23	66.60±0.44	**81.07±0.20**
PABN+$_{cpt}$	65.44±0.43	77.19±0.22	57.36±0.45	69.30±0.22	54.66±0.45	**66.74±0.22**	67.39±0.43	79.95±0.21
PABN+$_{cos}$	66.45±0.42	**78.34±0.21**	57.44±0.45	68.59±0.22	54.18±0.44	65.70±0.23	66.74±0.44	80.58±0.20
LRPABN$_{niv}$	64.62±0.43	**78.26±0.22**	**59.57±0.46**	**74.66±0.22**	**54.82±0.46**	66.62±0.23	**68.40±0.44**	80.17±0.21
LRPABN$_{cpt}$	**67.97±0.44**	78.04±0.22	**63.11±0.46**	**72.63±0.22**	54.52±0.47	**67.12±0.23**	**68.04±0.44**	**80.85±0.20**

2019*b*), which demonstrates that the comparative pairwise bilinear feature outperforms the self-bilinear feature on FGFS tasks. Besides, the pairwise bilinear feature-based approaches achieve better classification performances than RelationNet (Sung et al. 2018*b*), which validates the proposed second-order image descriptors surpasses the naive concatenation of feature pairs used in RelationNet for FGFS problems.

From Table 4.3, compared to PABN models, PABN+ and LRPABN models obtain a definite classification performance boost. For instance, the PABN+$_{niv}$ gains 1.64% and 0.37% improvements over PABN$_{niv}$ in one-shot and five-shot settings on CUB Birds data, while LRPABN$_{cpt}$ achieves 1.26% and 1.23% improvements over PABN$_{cpt}$ in one-shot and five-shot setting on the CUB Birds dataset.

These results demonstrate that the effectiveness of the proposed feature alignment layer. It can be observed from Table 4.3 that LRPABN models achieve the best or second-best classification performance on nearly all data sets compared to other methods under various experimental settings. For CARS data, the LRPABN$_{cpt}$ obtains 5.67%, 6.31%, 6.83% significant improvements over PABN+$_{cos}$, PABN$_{cpt}$, and RelationNet on one-shot-five-way task, while achieves 5.36%, 5.88%, 6.27% improvements against PABN+$_{cpt}$, PABN$_{cpt}$, and RelationNet on the five-shot-five-way setting, which validates the effectiveness of our low-rank pairwise bilinear pooling. It is worth noting that the dimension of the pairwise bilinear feature in LRPABN is 512, where the corresponding feature dimension of PABN and PABN+ is 4096.

4.6.4 Experimental Results for PABN and LRPABN on the RPSI Defect Dataset

For a further analysis of our models, we conduct an advanced experiment on the RPSI Defects and the above fine-grained datasets comparing the LRPABN models with DN4 and CovaMNet. In this experiment, we also compare the PABN+ models. Moreover, we use the same setting to rerun the RelationNet on five data sets as the baseline method. We follow the same data

Table 4.4: Few-shot classification accuracy (%) comparisons on four fine-grained and RPSI Defect data sets. The highest-accuracy and second-highest-accuracy methods are highlighted in red and blue, respectively. All results are with 95% confidence intervals where reported.

Methods	CUB Birds		CARS		DOGS		NABirds		RPSI Defect	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
RelationNet	59.82±0.77	71.83±0.61	56.02±0.74	66.93±0.63	44.75±0.70	58.36±0.66	64.34±0.81	77.52±0.60	57.60±0.87	66.43±0.72
CovaMNet	58.51±0.94	71.15±0.80	56.65±0.86	71.33±0.62	**49.10±0.76**	**63.04±0.65**	60.03±0.98	75.63±0.79	59.62±0.92	70.21±0.80
DN4	55.60±0.89	**77.64±0.68**	**59.84±0.80**	**88.65±0.44**	45.41±0.76	**63.51±0.62**	51.81±0.91	**83.38±0.60**	60.43±0.88	**76.70±0.77**
PABN+$_{niv}$	**63.56±0.79**	75.23±0.59	53.39±0.72	66.56±0.64	45.64±0.74	58.97±0.63	**66.96±0.81**	80.73±0.57	61.79±0.89	69.53±0.80
PABN+$_{cpt}$	63.36±0.80	74.71±0.60	54.44±0.71	67.36±0.61	45.65±0.71	61.24±0.62	66.94±0.82	79.66±0.62	62.24±0.90	70.73±0.81
PABN+$_{cos}$	62.02±0.75	75.35±0.58	53.62±0.73	67.15±0.60	45.18±0.68	59.48±0.65	66.34±0.76	80.49±0.59	62.08±0.92	70.56±0.79
LRPABN$_{niv}$	62.70±0.79	75.10±0.61	56.31±0.73	70.23±0.59	**46.17±0.73**	59.11±0.67	66.42±0.83	80.60±0.59	**63.46±0.90**	76.61±0.80
LRPABN$_{cpt}$	**63.63±0.77**	**76.06±0.58**	**60.28±0.76**	**73.29±0.58**	45.72±0.75	60.94±0.66	**67.73±0.81**	**81.62±0.58**	**64.51±0.87**	**77.21±0.79**

set split with DN4 and CovaMNet. The original papers of these two papers do not report the results on CUB Birds (CUB-2011) (Wah et al. 2011) and NABirds (Horn et al. 2015), so we use the open released codes of DN4[4] and CovaMNet[5] to get the results. During the test, 600 episodes are randomly selected from the data.

Table 4.4 presents the average accuracies of different models on the novel classes of the RPSI Defects dataset as well as the fine-grained datasets. Both the one-shot and five-shot classification results are reported. Both the PABN and LRPABN models achieve superior performance over compared FGFS and FS approaches for different experimental settings. Moreover, as the table shows, the proposed LRPABN models get steadily and notably improvements on the RPSI Defects dataset and almost all fine-grained datasets. More detailed, compared with CovaMNet, our proposed models achieve plainly growth performances on the RPSI defect dataset, CUB Birds, CARS, and NABirds data sets on both one-shot and five-shot settings. Especially for NABirds data, the LRPABN$_{cpt}$ obtains 7.70% and 5.99% gain over CovaMNet for one-shot and five-shot settings, respectively.

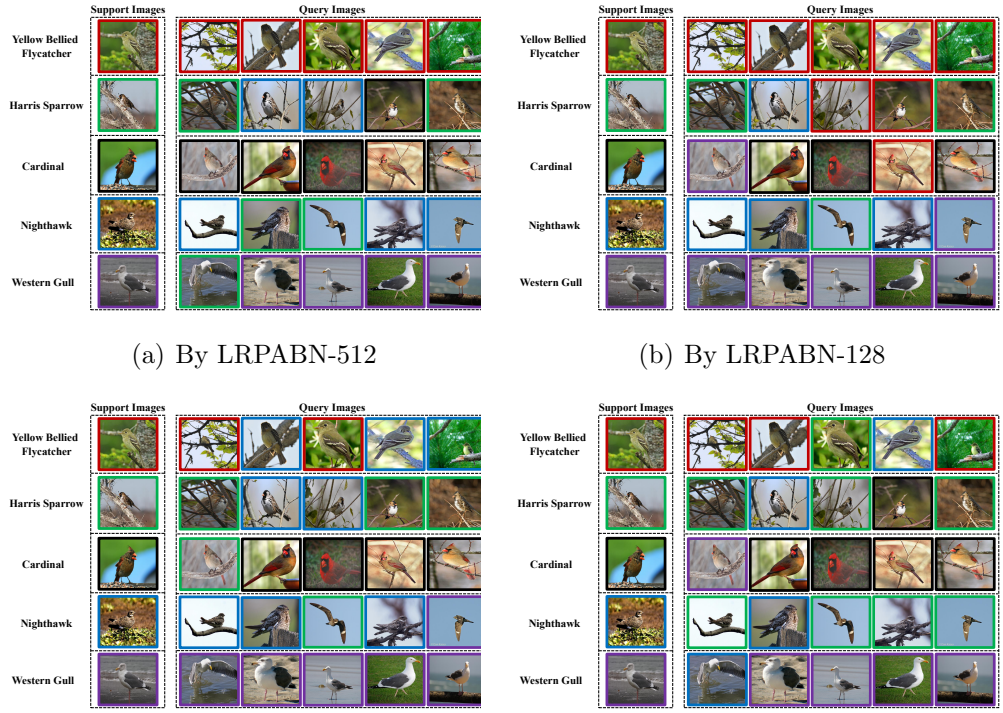
These results again firmly prove that the proposed pairwise bilinear pooling is superior compared to the self-bilinear pooling operation. Meanwhile, the feature alignment layer further boosts the final performance.

For the comparisons against the DN4 method, from Table 4.4, LRPABN models obtain the highest accuracy on one-shot setting on RPSI Defects, CUB Birds, CARS, NABirds data sets, and get second best results on DOGS data, where DN4 performs poorly in one-shot tasks on almost all data sets. For five-shot RPSI Defects recognition, the LRPABN$_{cpt}$ model also achieves the highest performance. Therefore, the proposed LRPABN model is tailored for RPSI Defects recognition, which validates the effectiveness of the proposed two-stage defects recognition framework. DN4 achieves the highest classification accuracy on four fine-grained data sets, while LRPABN$_{cpt}$

[4] `https://github.com/WenbinLee/DN4`

[5] `https://github.com/WenbinLee/CovaMNet`

achieves the second-highest performance on the CUB Birds, CARS, and NABirds.



(a) By LRPABN-512 (b) By LRPABN-128

(c) By PABN+ (d) By RelationNet

Figure 4.7: Sample visual classification results of comparing methods over CUB dataset. All the approaches use the same data batch under the five-way-one-shot setting, and for each class, we randomly select five query images as the testing data. We adopt five colors to label the support classes separately. As to the query images, we label the images with the color corresponding to the class label predicted by different models.

The reason for this is that DN4 uses a deep nearest neighbor neural network to search the optimal local features in the support set as the support classes' feature for a given query image. For the target query features (*e.g.,* a set of local features), the algorithm selects the top k nearest local features in the whole support data set according to the cosine similarity between query local features and support local features. That is, the more image in the

support classes, the better the class feature will be generated. Thus, for five-shot classification, the DN4 outperforms LRPABN, where under the one-shot setting, DN4 has smaller support features to extract a good representation of the class feature. More importantly, our model is more efficient than DN4. Specifically, under the C-way-K-shot setting, in the inference stage, for each query image, DN4 has $h^2 \times w^2 \times K \times C \times O_{cos}$ computations to predict its label, while LRPABN only needs $h \times w \times C \times O_{comp}$ computations. h and w denote the height and width of the feature map, O_{cos} means the cosine similarity computation used in DN4, and O_{comp} represents the comparator computation in LRPABN. Since $h \times w \times K \times O_{cos} \gg O_{comp}$, DN4 is much slower than LRBPAN during both training and testing, as seen from Table 4.6, DN4 costs 15.20 $\times 10^{-3}$ s for each query, while LRPABN only needs 2.23 $\times 10^{-3}$ s, which is approximately seven times faster. Moreover, without considering the computation load, our initial low-rank pairwise bilinear model PABN$_{new}$ (Equation (4.7)) can also achieve the comparable performance against DN4 under both one-shot and five-shot setting, *i.e.,* 78.87% for PABN$_{new}$ compared to 79.64% for DN4 under the five-shot settings. On the other hand, in many practical scenarios, such as endangered species protection, we may only get a one-labeled sample. With higher accuracy under the one-shot setting, our method can achieve more reliable performances compared to DN4 under such circumstances. It indicates the practical value of our models.

The classification examples of LRPABN, PABN+, and RelationNet models are shown in Figure 4.7. We select LRPABN$_{cpt}$ and PABN+$_{cpt}$ as the representative of LRPABN and PABN+ approaches. To investigate the low-rank approximation, we set low-rank comparative feature dimensions as 512 and 128 for LRPABN-512 and LRPABN-128 models separately. By sending a fixed testing batch through the model, which consists of one support sample and five query samples for each of five classes, the prediction of LRPABN-512 only contains six mislabels in the entire 25 queries, while the prediction of LRPABN-128, PABN+, and RelationNet have 7, 8 and 10 wrong labels separately. That validates the effectiveness of the LRPABN models. We

also find that in some classes like Nighthawk and Harris Sparrow, the high intra-variance and low inter-variance confuse all the models.

4.6.5 Experimental Results for TOAN on RPSI Defects and Fine-grained Datasets

We compare our third FGFS model TOAN against other FGFS and FS methods on both RPSI Defects and four fine-grained datasets.

Comparison against existing FGFS methods The comparisons between TOAN and other state-of-the-art FGFS methods are shown in the upper part of Table 4.5. We conclude that our method compares favorably over existing FGFS approaches on five datasets. Specifically, under the five-way-one-shot setting, the classification accuracies are 66.61% vs. 63.75% (SoSN), 65.34% vs. 63.95% (SoSN), 65.90% vs. 29.63% (PCM), 49.30% vs. 49.10% (CovaMNet), and 70.02% vs. 67.73% (LRPABN) on RPSI Defects, CUB, CARS, DOGS, and NABirds, respectively. Moreover, by replacing the ConvNet-64 with the deeper ResNet-256 model (Li, Wang, Xu, Huo, Gao & Luo 2019), the accuracy of TOAN:ResNet gets further improvements, *e.g.* under five-way-five-shot setting, TOAN:ResNet achieves 82.07%, 82.09%, 89.57%, 69.83%, and 90.21% compared with 80.43%, 84.24%, 67.16%, and 85.52% of TOAN model on five datasets.

Comparison against Generic FS As our experiments were conducted under the few-shot setting, we also investigate how generic few-shot learning methods perform for the fine-grained classification. We report the results of representative generic FS models in the lower part of Table 4.5. It can be observed that the proposed TOAN models outperform most of these methods by large margins, which is expected since our models are designed to address both intra- and inter-class variance issues in the fine-grained classification.

More specifically, compared with DN4 (Li, Wang, Xu, Huo, Gao & Luo 2019), the proposed TOAN achieves the highest accuracy on RPSI Defects,

Table 4.5: Fine-grained Few-shot classification accuracy (%) comparisons on RPSI Defect and four FG benchmarks. All results are with 95% confidence intervals where reported. We highlight the best and second-best methods.

Methods	Type	Backbone	CUB		CARS		DOGS		NABirds		RPSI Defects	
			1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
MatchingNet	FS	ConvNet-64	57.59±0.74	70.57±0.62	48.03±0.60	64.22±0.59	45.05±0.66	60.60±0.62	60.70±0.78	76.23±0.62	50.47±0.90	65.33±0.74
ProtoNet	FS	ConvNet-256	53.88±0.72	70.85±0.63	45.27±0.61	64.24±0.61	42.58±0.63	59.49±0.65	55.85±0.78	75.34±0.63	47.54±0.89	64.97±0.76
RelationNet	FS	ConvNet-64	59.82±0.77	71.83±0.61	56.02±0.74	66.93±0.63	44.75±0.70	58.36±0.66	64.34±0.81	77.52±0.60	57.60±0.87	66.43±0.72
CovaMNet	FS	ConvNet-64	58.51±0.94	71.15±0.80	56.65±0.86	71.33±0.62	49.10±0.76	63.04±0.65	60.03±0.98	75.63±0.79	59.62±0.92	70.21±0.80
DN4	FS	ConvNet-64	55.60±0.89	77.64±0.68	59.84±0.80	**88.65±0.44**	45.41±0.76	63.51±0.62	51.81±0.91	83.38±0.60	60.43±0.88	76.70±0.77
PCM	FGFS	AlexNet	42.10±1.96	62.48±1.21	29.63±2.38	52.28±1.46	28.78±2.33	46.92±2.00	-	-	-	-
PABN+$_{cpt}$	FGFS	ConvNet-64	63.36±0.80	74.71±0.60	54.44±0.71	67.36±0.61	45.65±0.71	61.24±0.62	66.94±0.82	79.66±0.62	62.24±0.90	70.73±0.81
LRPABN$_{cpt}$	FGFS	ConvNet-64	63.63±0.77	76.06±0.58	60.28±0.76	73.29±0.58	45.72±0.75	60.94±0.66	67.73±0.81	81.62±0.58	64.51±0.87	77.21±0.79
SoSN	FGFS	ConvNet-64	63.95±0.72	78.79±0.60	62.84±0.68	75.75±0.52	48.01±0.76	64.95±0.64	69.53±0.77	83.87±0.51	63.75±0.91	75.43±0.78
TOAN	FGFS	ConvNet-64	**65.34±0.75**	**80.43±0.60**	**65.90±0.72**	84.24±0.48	**49.30±0.77**	**67.16±0.49**	**70.02±0.80**	**85.52±0.50**	**66.61±0.90**	**79.92±0.80**
TOAN:ResNet	FGFS	ResNet-256	**67.17±0.81**	**82.09±0.56**	**76.62±0.70**	**89.57±0.40**	**51.83±0.80**	**69.83±0.66**	**76.14±0.75**	**90.21±0.40**	**68.79±0.85**	**82.07±0.78**

CUB, DOGS, and NABirds datasets, and achieves the highest performance under the five-way-one-shot setting on the CARS dataset. DN4 achieves the best performance under the five-way-five-shot setting on the CARS dataset, and our TOAN model ranks the second best. In general, the proposed TOAN method holds obvious advantages against DN4 for one-shot tasks and most five-shot tasks. It is worth noting that DN4 employs a deep nearest neighbor network to search the optimal local features in the support set as the class prototypes for a given query image. For the query features, DN4 selects the top k-nearest local features in the whole support set based on the cosine similarities between the local features of query and support images. Therefore, with more images from support classes (under five-shot setting on CARS), DN4 tends to generate relatively accurate prototypes.

In the fine-grained classification, different categories share similar appearances. The similarities between any two samples are always high, which means the top k-nearest local features sorted by DN4 in different support classes are also similar. This leads to the degeneracy problem for DN4 in dealing with fine-grained classification. However, the proposed TOAN aligns the support-query pairs by TOMM. Thus the corresponding positions between two samples achieve the highest spatial similarity. Then a GPBP module is adopted to compare the nuanced differences between the pairs using high-order feature extraction. Therefore, TOAN can learn a more robust representation and generally achieves better performances than DN4.

4.6.6 Ablation Studies

To further investigate the proposed APBF framework, we present some ablation studies of the proposed LRPABN and TOAN models in this section.

Analysis of LRPABN

Following the data split used in (Wei et al. 2019*b*, Huang et al. 2019), we conduct several experiments to investigate the different components of the

Table 4.6: Ablation study of LRPABN with different components. The results are reported with 95% confidence intervals. Model size indicates the number of parameters for each model, the Test Time is the testing time for each input query image, and the Bilinear Dim represents the bilinear feature dimension of the each model.

Methods	CUB data set				
	1-shot (%)	5-shot (%)	Model Size	Test Time (ms)	Bilinear Dim
PABN$_{cpt}$	66.71±0.43	76.81±0.21	375,361	8.65	4096
PABN+$_{cpt}$	65.44±0.43	77.19±0.22	505,682	8.94	4096
PABN$_{new}$	67.39±0.43	78.87±0.21	2,373,819	78.40	512
LRPABN	66.56±0.43	77.60±0.22	213,930	2.23	512
LRPABN$_{only_cpt}$	66.72±0.44	77.98±0.21	213,930	2.23	512
LRPABN$_{cpt}$	67.97±0.44	78.04±0.22	344,251	2.53	512
DN4	60.02±0.85	79.64±0.67	112,832	15.20	-

LRPABN model, and the experimental results are shown in Table 4.6. We analyze our methods from various aspects:

Low-Rank Pairwise Bilinear Pooling: First, we replace previous pairwise bilinear pooling (Equation (4.4)) with Equation (4.7) as PABN$_{new}$. As seen in Table 4.6, PABN$_{new}$ outperforms PABN$_{cpt}$ on both one-shot and five-shot tasks with a lower dimension, which indicates the effectiveness of our proposed initial Low-Rank pairwise pooling (Equation (4.7)). However, using Equation (4.7), the model needs to learn a $n \times c \times c$ transformation tensor $\mathcal{W}$ (discussed in Section 4.4.1), which significantly increases the model size and inference time. Thus, we employ Equation (4.8) to approximate the transformation tensor as LRPABN. We observe that this approximation achieves superior performance against our previous PABN$_{cpt}$ with a reduced model size as well as a shorter bilinear feature dimension. Specifically, as observed in Table 4.6, the proposed LRPABN costs 2.23 $\times 10^{-3}$ s to identify a query image with a 213K model size, while the previous ICME model PABN$_{cpt}$ requires 8.65 $\times 10^{-3}$ s and 375K parameters. Moreover, the inference time

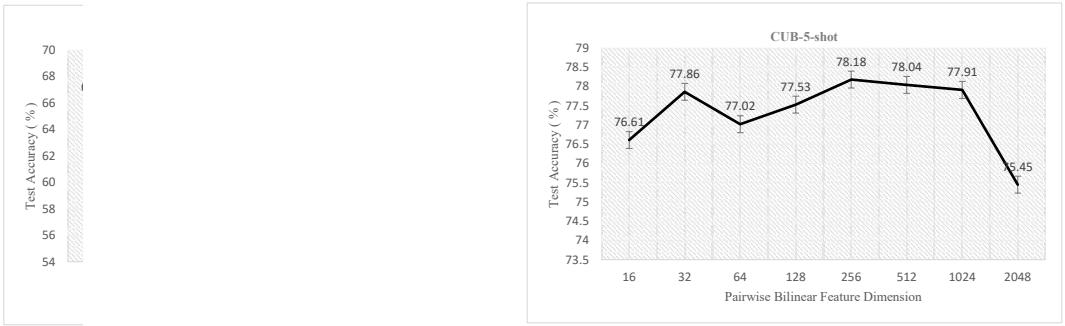
Table 4.7: Impact of input image size on FSFG.

Methods	CUB data set		
	1-shot (%)	5-shot (%)	Image Size
PCM:AlexNet (Wei et al. 2019*b*)	42.10±1.96	62.48±1.21	224 × 224
LRPABN$_{cpt}$:AlexNet	59.34±0.48	69.08±0.24	224 × 224
LRPABN$_{cpt}$:AlexNet	66.19±0.46	75.05±0.23	448 × 448
LRPABN$_{cpt}$:Conv4	67.97±0.44	78.04±0.22	84 × 84
DN4:Conv4 (Li, Wang, Xu, Huo, Gao & Luo 2019)	60.02±0.85	79.64±0.67	84 × 84

of LRPABN is 2.23×10^{-3} s, while PABN$_{new}$ costs 78.40 ×10^{-3} s for each query image. That is, our final low-rank pairwise pooling model LRPABN is more advanced than previous PABN models and much more efficient than PABN$_{new}$ model.

Alignment Mechanism: To investigate the effectiveness of the proposed alignment mechanism. We compare PABN$_{cpt}$ and PABN+$_{cpt}$. Besides, we adopt the proposed alignment loss $Align_{loss_2}$ in Equation (4.11) into LRPABN as LRPABN$_{only_cpt}$. As seen from Table 4.6, cooperating with the position transform function $\mathbf{T}$, PABN+$_{cpt}$ and LRPABN$_{cpt}$ outperform PABN$_{cpt}$ and LRPABN$_{only_cpt}$, respectively. For instance, under the five-shot setting, the classification accuracy of PABN+$_{cpt}$ is 77.19% compared to 76.81% of PABN$_{cpt}$. PABN+$_{cpt}$ obtains less improvements on one-shot setting compared to LRPABN$_{cpt}$ model. This is expected as the proposed LRPABN model is designed to extract more discriminative feature than PABN. With more powerful feature, the alignment mechanism can perform better.

Input Image Size: It is reported that a higher resolution of the input image can capture a more discriminative feature for Fine-grained classification (Lin, RoyChowdhury & Maji 2015*a*, Cui et al. 2017*b*, Li et al. 2018). However, few-shot learning models (Vinyals et al. 2016, Snell et al. 2017, Sung et al. 2018*b*) usually adopt a low input resolution, e.g., 84 × 84. For a fairness comparison with generic few-shot learning approaches, in Section 4.6.2, we set the input image size to 84 × 84. To further investigate the effects

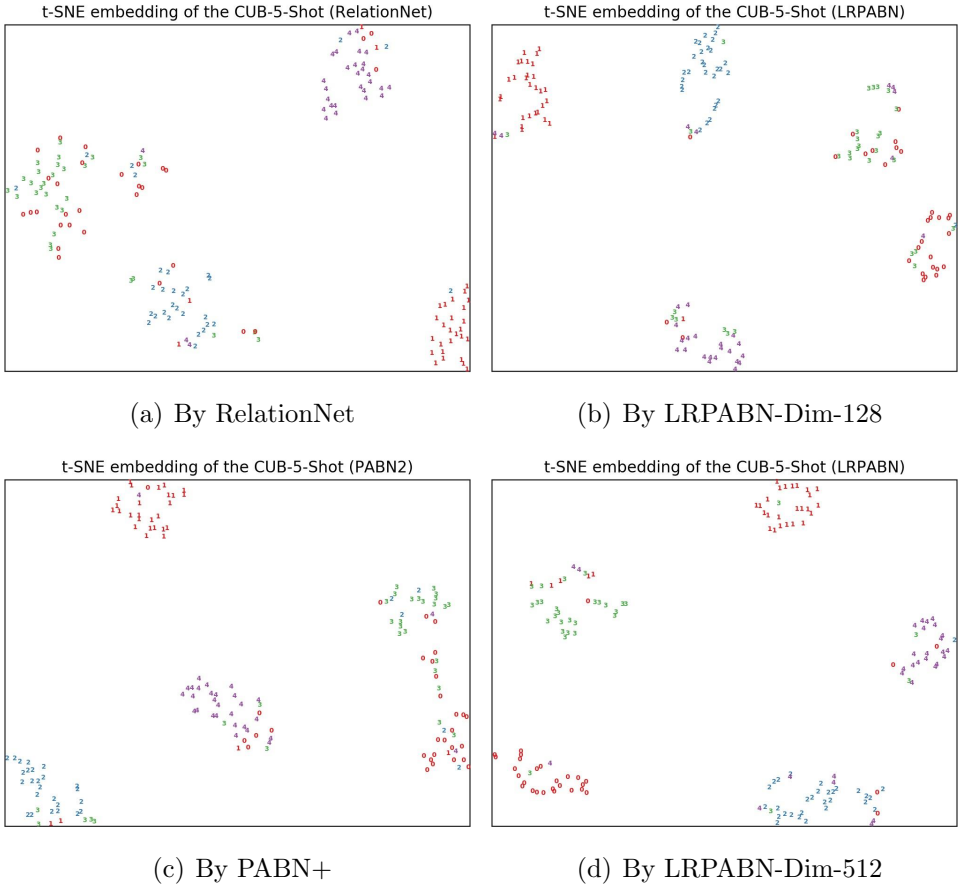


(b)

Figure 4.8: The pairwise bilinear feature dimension selection experiment. In each sub-figure, the horizontal axis denotes the dimension of the pairwise bilinear feature and the vertical axis represents the test accuracy rate. 4.8(a) is the one-shot experiment and 4.8(b) is the five-shot experiment on CUB.

of input size, we follow (Wei et al. 2019*b*) to replace the shallow embedding network Conv4 (Vinyals et al. 2016, Snell et al. 2017, Sung et al. 2018*b*) with AlexNet (Krizhevsky et al. 2012) as LRPABN$_{cpt}$:AlexNet. Moreover, we choose two resolutions for the input images, which are widely used in Fine-grained classification. As Table 4.7 shows, with AlexNet, a higher resolution 448 $\times$ 448 brings a significant performance boost compared to lower input size 224 $\times$ 224, which validates that a higher input resolution can generate a more subtle comparative feature for FSFG. We also observe that the accuracy of the AlexNet-based methods performs worse than Conv4-based methods. A high input resolution always accompanied by a deep embedding network like AlexNet to extract the informative feature. However, training a deeper embedding network with limited labeled samples is easier to lead the over-fitting problem.

Bilinear Feature Dim: For the feature dimension selection, we change the number of dimensions as 16, 32, 64, 128, 256, 512, 1024, and 2048 for both one-shot and five-shot classification tasks on CUB Birds data. The model we used for this experiment is LRPABN$_{cpt}$. The results are shown in Figure 4.8, it can be observed that as the feature dimension gets large, the test accuracy gradually improves to a peak first, then it goes through a



(a) By RelationNet (b) By LRPABN-Dim-128

(c) By PABN+ (d) By LRPABN-Dim-512

Figure 4.9: Visualization of the comparative feature generated by different fusion mechanism in 2D space using t-SNE. Each dot represents a query image that is numeric and marked with different colors according to the real labels. For each class, we randomly select thirty query images to conduct this experiment. The visualization is based on the CUB data set under the five-way-five-shot setting. (a) shows results conducted by RelationNet, (b) shows the result conducted by LRPABN$_{cpt}$, and the dimension of the comparative bilinear feature is 128, denoted as LRPABN-Dim-128, (c) shows the result conducted by PABN+$_{cpt}$ model and (d) shows the result conducted by LRPABN$_{cpt}$, and the dimension of the comparative bilinear feature is 512, denoted as LRPABN-Dim-512.

drastic drop. For the one-shot setting, the performance changes smoothly when the dimension is below 1024. For the five-shot task, the variation of performance is relatively oscillatory, yet it can grow fast and steadily, with the dimension increasing. Moreover, we find that even with a very compact low-rank approximation (*i.e.,* the dimension is 16), the model can still achieve a decent classification performance, which fatherly verifies the stability of the proposed method. When the dimension goes too large, the model performs poorly, and this may be caused by the increased complexity of the framework can not model the data distribution well with few training samples. As (Gao et al. 2016) discussed, for self-bilinear features, less than 5% of dimensions are informative. For FSFG, the best feature dimensions for LRPABN are 256 and 512 in the experiments, which are around 5% to 10% of the entire self-bilinear feature dimension.

t-SNE visualization: The t-SNE (Maaten & Hinton 2008) visualization for different comparative features is presented in Figure 4.9. We randomly select five support images and thirty query images per category from CUB Birds data to conduct the five-way-five-shot tasks. The original comparative feature dimension of RelationNet is $128 \times 3 \times 3$. We use the convolved feature before the first fully-connected layer in classifier as the final comparative feature with dimension size 576. The comparative feature of PABN+ is $64 \times 64 = 4096$, and we choose LRPABN_{cpt} with comparative dimension 128 and 512 separately (denoted as LRPABN-Dim-128 and LRPABN-Dim-512) for comparison. As the figure shows, the learned LRPABN-Dim-512 feature, which can be grouped into five classes correctly, outperforms others, the discriminative performance of LRPABN-Dim-128 and PABN+ are similar, which outperform RelationNet' feature. The intuitive visualization results among the above methods again validate the superior capacity of the proposed low-rank pairwise bilinear features for FSFG tasks.

Table 4.8: The ablation study on TOMM and GPBP. The upper and lower parts of the table show the ablation study on TOMM and GPBP, separately. We incorporate each framework with TOMM and GPBP, we observe definite improvements (%). We also show the results of the whole model TOAN.

Methods	CUB		CARS		DOGS		NABirds	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
MatchingNet (Vinyals et al. 2016)	57.59±0.74	70.57±0.62	48.03±0.60	64.22±0.59	45.05±0.66	60.60±0.62	60.70±0.78	76.23±0.62
MatchingNet+TOMM	**60.87±0.78**	**75.12±0.61**	**53.79±0.72**	**72.67±0.55**	**47.06±0.74**	**63.22±0.62**	**65.83±0.75**	**80.73±0.57**
	+3.28	+4.55	+5.76	+8.45	+2.01	+2.62	+5.13	+4.50
ProtoNet (Snell et al. 2017)	53.88±0.72	70.85±0.63	45.27±0.61	64.24±0.61	42.58±0.63	59.49±0.65	55.85±0.78	75.34±0.63
ProtoNet+TOMM	**61.60±0.76**	**75.09±0.61**	**52.50±0.69**	**68.13±0.58**	**46.36±0.73**	**61.56±0.65**	**64.77±0.79**	**80.84±0.56**
	+7.72	+4.24	+7.23	+3.89	+3.78	+2.07	+9.92	+5.50
RelationNet (Sung et al. 2018*b*)	59.82±0.77	71.83±0.61	56.02±0.74	66.93±0.63	44.75±0.70	58.36±0.66	64.34±0.81	77.52±0.60
RelationNet+TOMM	**64.84±0.77**	**79.75±0.54**	**62.35±0.77**	**81.57±0.51**	**47.24±0.78**	**65.23±0.66**	**69.55±0.77**	**85.01±0.51**
	+5.02	+7.92	+6.33	+14.64	+2.49	+6.87	+5.21	+7.49

Methods	CUB		CARS		DOGS		NABirds	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
RelationNet (Sung et al. 2018*b*)	59.82±0.77	71.83±0.61	56.02±0.74	66.93±0.63	44.75±0.70	58.36±0.66	64.34±0.81	77.52±0.60
RelationNet+GPBP	**60.00±0.74**	**74.01±0.60**	**58.35±0.73**	**73.49±0.59**	**46.45±0.70**	**61.70±0.65**	**65.43±0.81**	**80.13±0.58**
	+0.18	+2.18	+2.33	+6.56	+1.70	+3.34	+1.09	+2.61
RelationNet (Sung et al. 2018*b*)	59.82±0.77	71.83±0.61	56.02±0.74	66.93±0.63	44.75±0.70	58.36±0.66	64.34±0.81	77.52±0.60
TOAN [a]	**65.34±0.75**	**80.43±0.60**	**65.90±0.72**	**84.24±0.48**	**49.30±0.77**	**67.16±0.49**	**70.02±0.80**	**85.52±0.50**
	+5.52	+8.60	+9.88	+17.31	+4.55	+8.80	+5.68	+8.00

[a]TOAN consists of RelationNet, TOMM, and GPBP together.

Table 4.9: Ablation study of TOAN for other choices. Few-shot classification results (%) on four FG datasets. The lower parts of the table is the different backbone choices of TOAN.

Methods	CUB		CARS		DOGS		NABirds	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
TOAN-GP*	65.80±0.78	79.37±0.61	65.88±0.74	82.69±0.50	50.10±0.79	65.90±0.68	69.48±0.75	85.48±0.53
TOAN-w/o $d(\cdot)$	64.48±0.76	78.82±0.59	60.02±0.73	81.65±0.49	47.27±0.72	63.98±0.65	68.70±0.79	83.70±0.53
TOAN	65.34±0.75	80.43±0.60	65.90±0.72	84.24±0.48	49.30±0.77	67.16±0.49	70.02±0.80	85.52±0.50
TOAN_224	69.03±0.79	83.19±0.56	69.48±0.74	87.38±0.45	53.67±0.80	69.77±0.70	75.17±0.76	88.77±0.46
TOAN:ResNet_224	**69.91±0.82**	**84.86±0.57**	**77.25±0.73**	**91.19±0.40**	**55.77±0.79**	**72.16±0.72**	**77.32±0.70**	**91.39±0.41**
Methods	CUB		CARS		DOGS		NABirds	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
ConvNet-64	65.34±0.75	80.43±0.60	65.90±0.72	84.24±0.48	49.30±0.77	67.16±0.49	70.02±0.80	85.52±0.50
ConvNet-128	64.56±0.78	80.02±0.59	69.20±0.72	86.39±0.44	50.26±0.77	66.96±0.66	70.90±0.77	85.63±0.49
ConvNet-256	66.16±0.80	80.72±0.58	68.89±0.74	85.29±0.46	49.68±0.75	67.52±0.66	71.26±0.76	86.42±0.47
ConvNet-512	66.44±0.77	81.46±0.54	69.59±0.73	86.27±0.45	49.20±0.74	66.75±0.66	72.74±0.76	86.91±0.50
ResNet-64	69.25±0.81	81.90±0.61	74.64±0.76	90.20±0.41	53.33±0.82	69.96±0.70	75.98±0.72	89.55±0.44
ResNet-128	68.95±0.78	83.40±0.58	75.14±0.72	90.95±0.36	52.69±0.81	69.95±0.71	76.14±0.75	90.51±0.38
ResNet-256	67.17±0.81	82.09±0.56	76.62±0.70	89.57±0.40	51.83±0.80	69.83±0.66	76.14±0.75	90.21±0.40
ResNet-512	66.10±0.86	82.27±0.60	75.28±0.72	87.45±0.48	49.77±0.86	69.29±0.70	76.24±0.77	89.88±0.43

Table 4.10: Ablation study of TOAN for the output channel size of d_α, d_β. The table shows five-way few-shot recognition results (%) on the CUB dataset.

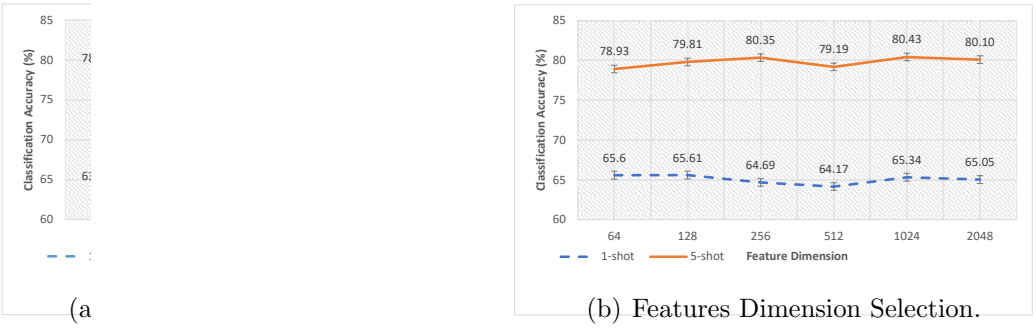
Num_Channel $d(c')$	CUB data set	
	5-way-1-shot	5-way-5-shot
32	65.47±0.77	80.51±0.60
64	67.17±0.81	82.09±0.56
128	68.73±0.80	83.88±0.61
256	**69.40±0.81**	**84.01±0.59**

(a) CUB Dataset (b) CARS Dataset (c) DOGS Dataset (d) NABirds Dataset

Figure 4.10: TOMM Visualization, the first image in each row (except for the first row) represents the support image, and the remaining images in the row are the aligned results of the support image, which are matched to each query image (in each column from the first row).

Analysis of TOAN

Target-Oriented Matching Mechanism (TOMM): First of all, we investigate the effectiveness of TOMM for FGFS tasks. As Table 4.8 shows, there is an approximate 5% averagely increase after adopting TOMM in three FS baselines, among which, when incorporating TOMM into the RelationNet, the model achieves superior performances over other compared approaches. For instance, under the five-way-five-shot setting, the accuracy of RelationNet+TOMM is 79.75% vs. 78.79% (Zhang & Koniusz 2019), 65.23% vs. 63.04% (Li, Xu, Huo, Wang, Gao & Luo 2019), and 85.01% vs. 83.38% (Li,



(a) (b) Features Dimension Selection.

Figure 4.11: Ablation studies about the proposed GPBP, including semantic channel grouping validation 4.11(a) and feature dimension selection 4.11(b). For each group of validation experiments, we show the 1-shot and 5-shot results.

Wang, Xu, Huo, Gao & Luo 2019) on CUB, DOGS, and NABirds. This verifies that the significant intra-class variance is a crucial issue in FGFS tasks, and the TOMM is an effective mechanism to tackle such problems. Furthermore, since the proposed task-agnostic transformation $d(\cdot)$ can better capture the similarities of input pairs, TOAN outperforms TOAN-w/o $d(\cdot)$, as shown in Table 4.9. To fully investigate the influence of the output channel size (c') of $d(\cdot)$ (d_α and d_β) in Eq. (4.17), we employ ResNet-256 (Li, Wang, Xu, Huo, Gao & Luo 2019, Li, Xu, Huo, Wang, Gao & Luo 2019) as the backbone and experiment with different output channel sizes of the TOMM (the input channel size is fixed as 256). As is reported in Table 4.10, the larger output channel size of $d(\cdot)$ generally achieves a better performance.

In Fig. 4.10, we give the visualization of the TOMM. We utilized the original images to get vivid descriptions of the proposed feature alignment. More specifically, we first resized the original images to the same size as the target-oriented attention map (19×19). Then we multiplied the image with the corresponding attention map to generate the aligned features as Equation (4.18). We observe that for the support image (each row in Fig. 4.10), TOMM transforms its features to match each query (top column images). For instance, in the fourth column in Fig. 4.10(b), the postures of five sup-

Table 4.11: Investigation of model complexity. Model size indicates the number of parameters for each model, and the Test Time is the testing time for each input query image.

Methods	CUB data set				
	1-shot (%)	5-shot (%)	Model Size	Test Time (ms)	Feature Dim
ProtoNet	53.88	70.85	113,088	0.69	64
MatchingNet	57.59	70.57	113,088	0.68	64
RelationNet	59.82	71.83	228,686	1.14	128
DN4	55.60	77.64	112,832	15.20	64
PABN_{cpt}	63.36	74.71	375,361	8.65	4096
LRPABN_{cpt}	63.63	76.06	344,251	2.53	512
TOAN	65.60	78.93	198,417	0.66	64
TOAN	65.61	79.81	237,585	0.87	128
TOAN	64.69	80.35	315,921	1.04	256
TOAN	64.17	79.19	472,593	1.23	512
TOAN	65.34	80.43	785,937	2.34	1024

port cars are reshaped to the same posture of the red query car in the top row.

Group pairwise Bilinear Pooling (GPBP): The distance-based frameworks such as MatchingNet (Vinyals et al. 2016) and ProtoNet (Snell et al. 2017) use the l_2 or cosine distance of support-query pairs to conduct the classification, while GPBP aims to learn the distance of support-query pairs with a convolutional network, a classifier is then applied to generate the relation confidences, which is the reason why it is not compatible with MatchingNet or ProtoNet. On the other hand, RelationNet (Sung et al. 2018*b*) proposes to use a comparator network to classify queries according to the distance of support-query pairs. Therefore, we combine GPBP with RelationNet to study its capability. RelationNet+GPBP brings certain performance gains over RelationNet, as is shown in Table 4.8 (lower parts). As is expected, after combining TOMM and GPBP together, the complete model TOAN achieves significant improvements over the ablation models, indicating that

Table 4.12: Investigation of model scalability. Model size indicates the number of parameters for each model.

Methods	CUB data set				
	10-way-1-shot(%)	10-way-5-shot(%)	Model Size	Time(ms)	Dim
ProtoNet	37.50±0.48	57.46±0.56	113,088	0.74	64
MatchingNet	40.85±0.50	58.07±0.55	113,088	0.76	64
RelationNet	42.69±0.52	59.37±0.58	228,686	1.52	128
TOAN	50.95±0.57	68.82±0.57	198,417	1.05	64

the TOMM and GPBP can benefit from each other. From the second results row in Table 4.9, the grouping (Zhang et al. 2017) model TOAN-GP* achieves analogous performances as TOAN under the one-shot setting. However, its performance is lower than TOAN under the five-shot, which verifies the effectiveness of our grouping operation.

We conducted two additional experiments to furtherly investigate the hyper-parameters of GPBP. First of all, we evaluate the semantic grouping in Fig. 4.8(b). We observe that when the grouping number is less than or equal to eight, a larger group size generally results in higher performances, *e.g.,* the accuracy reaches the highest (80.69%) when the grouping size equals eight, under five-shot settings. This indicates the effectiveness of semantic grouping on boosting the discrimination of the bilinear features. When the grouping number is greater than eight, the performances tend to be stable with small fluctuations.

Next, we conducted the selection of feature dimensions, as shown in Fig. 4.11(b). It is observed that a higher dimension brings a slight improvement under the five-shot setting. For example, the performance is 78.93% vs. 80.10%, when the length of the bilinear feature is 64 vs. 2048 under the five-way-five-shot setting on CUB, and the model works relatively stable for one-shot experiments.

Input Image Size for TOAN: In high-order-based FG methods (Cui et al. 2017*b*, Li et al. 2018, Lin, RoyChowdhury & Maji 2015*a*), a higher resolution of the input image usually results in a more fine-grained feature, which is

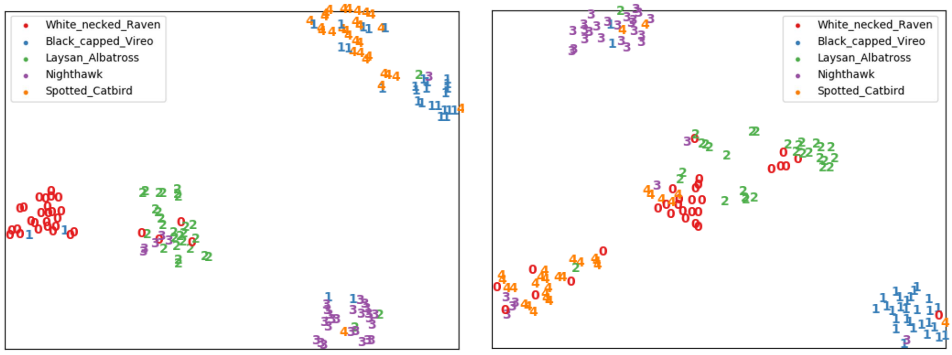
consistent with the results of current FGFS models (Huang, Zhang, Zhang, Xu & Wu 2021, Zhang & Koniusz 2019). Therefore, we conducted experiments to investigate the effects of input resolution for TOAN. As can be seen from Table 4.9, TOAN_224 and TOAN:ResNet_224 achieve further improvements with a larger 224×224 input size compared with the smaller 84×84 resolution.

Different Backbones for TOAN: We selected different backbones to investigate our proposed model. First, we adopted the ConvNet-512 (Gidaris, Bursuc, Komodakis, Pérez & Cord 2019) as the embedding network, which is derived from ConvNet-64 by increasing the width across layers to 512 channels, and we further revise ConvNet-64 to ConvNet-128, ConvNet-256. Similarly, we designed the ResNet-64, ResNet-128 and ResNet-512. From Table 4.9 (lower parts), we observe that a wider ConvNet-based backbone can result in higher performances in FSFG classification. On the other hand, deeper backbones can achieve further improvements compare to shallow ones. For instance, ResNet-64 outperforms ConvNet-512 on both one-shot and five-shot experiments. Under the five-shot setting, TOAN achieves relatively stable performances when the width of ResNet changes.

Feature Visualization Fig. 4.12(a) visualizes the feature distribution of the learned fine-grained features using t-SNE (Maaten & Hinton 2008). The features are generated under the five-way-five-shot setting on the CUB. We used 30 query images per class. As can be observed, the learned features by our TOAN have more compact and separable clusters than RelationNet (Fig. 4.12(b)).

Model Complexity and Scalability The main complexity of our model is the TOMM operation, which has $O((hw)^2)$, where $h \times w$ represents the size of the convolutional map. In our implementation, $h = w = 19$. In general, a deeper convolutional network usually results in a smaller feature map before feeding to the classifier. Therefore, the TOMM operation is more efficient with deeper backbones. We conducted additional experiments to investigate the model size and inference time of TOAN compared with previous works (Snell

et al. 2017, Vinyals et al. 2016, Sung et al. 2018*b*, Li, Wang, Xu, Huo, Gao & Luo 2019, Huang et al. 2019, Huang, Zhang, Zhang, Xu & Wu 2021). As is shown in Table 4.11, using the same feature dimension, the proposed TOAN model achieves the best performance compared with other models with a small model size as well as a short time. While using a larger dimension, the classification performance can be further improved.



(a) TOAN, 80.67% accuracy. (b) RelationNet, 76.67% accuracy.

Figure 4.12: t-SNE visualization of the features learned by TOAN, five classes are randomly selected, and in each class, 30 query images are randomly chosen. Different colored numbers are used to denote different classes, *i.e.*, red zero represents the white-necked raven, blue one denotes the blacked capped vireo, green two represents the Laysan albatross, purple three denotes the nighthawk, and yellow four denotes the spotted catbird. Moreover, each sample is represented by its corresponding number. 4.9(a) and 4.9(b) show the t-SNE results of TOAN and RelationNet, respectively.

Most current models (Wei et al. 2019*a*, Zhang & Koniusz 2019, Huang, Zhang, Zhang, Xu & Wu 2021, Li, Xu, Huo, Wang, Gao & Luo 2019) are based on a relatively smaller size of the category (five-way) when dealing with FGFS. To further investigate the scalability of the proposed TOAN, we conducted a larger number of categories experiments, which is referred to (Chen et al. 2019, Liu et al. 2019*a*). As is shown in Table 4.12, with the same feature dimension, the proposed TOAN outperforms other baseline models

with the comparable model size and inference speed.

4.7 Summary

In this chapter, we present the Aligned Pairwise Bilinear Framework (APBF) for PRSI defects and generic fine-grained image recognition. The key components of APBF are the feature alignment layer and pairwise bilinear pooling. Specifically, the alignment layer can eliminate the biases brought by the intra-class variance in fine-grained datasets, which is a crucial issue but less considered in current studies. Moreover, pairwise bilinear pooling is adopted to extracts the second-order comparative features for the pair of support images and query images, which can enlarge the low inter-class variance in FGFS tasks. Using this framework, we proposed three FGFS models: PABN, LRPABN, and TOAN, progressively. We have validated the effectiveness of the proposed models on RPSI Defects and four generic fine-grained datasets, which achieves state-of-the-art performance. Since TOAN applies a cross-attention mechanism to align the support-query feature pairs, it may cost a large computation load given a large input size and more feature pairs, as cross-attention is a matrix product operation. Therefore, designing a more effective and computationally friendly alignment is an interesting topic for the APBF framework. In addition, fusing the alignment module with the high-order feature extraction module is also an open problem.

Chapter 5

Poisson Transfer Network for Semi-supervised Few-shot RPSI Defect Recognition

5.1 Introduction

In Chapter 4, we studied the RPSI defects recognition under fine-grained few-shot settings and proposed a meta-learning-based framework to tackle this challenge. As discussed in Chapter 1.2.2, the last research issue in this thesis is to study the semi-supervised few-shot learning model that uses the unlabeled samples to boost the few-shot learners. Therefore, in this chapter, we propose to study the semi-supervised few-shot learning model to solve the RSPI defects recognition and generic image recognition.

Generic few-shot learning (Miller, Matsakis & Viola 2000, Fei-Fei et al. 2006, Vinyals et al. 2016) aims to learn a model that generalizes well with a few instances of each novel class. In general, a few-shot learner is firstly trained on a substantial annotated dataset, also noted as the base-class set, and then adapted to unseen novel classes with a few labeled instances. This research topic has been proved immensely appealing in the past few years, as a large number of few-shot learning methods are proposed from various

perspectives. Mainstream methods can be roughly grouped into two categories. The first one is learning from episodes (Vinyals et al. 2016), also known as meta-learning, which adopts the base-class data to create a set of episodes. Each episode is a few-shot learning task, with support and query samples that simulate the evaluation procedure. The second type is the transfer-learning-based method, which focuses on learning a decent classifier by transferring the domain knowledge from a model pre-trained on the large base-class set (Chen et al. 2018, Qiao et al. 2018). This paradigm decouples the few-shot learning progress into representation learning and classification, and has shown favorable performance against meta-learning methods in recent works (Tian et al. 2020, Ziko et al. 2020). Our method shares somewhat similar motivation with transfer-learning-based methods and proposes to utilize the extra unlabeled novel-class data and a pre-trained embedding to tackle the few-shot problem.

Compared with collecting labeled novel-class data, it is much easier to obtain abundant unlabeled data from these classes. Therefore, semi-supervised few-shot learning (SSFSL) (Ren et al. 2018, Liu et al. 2018, Li, Sun, Liu, Zhou, Zheng, Chua & Schiele 2019, Yu et al. 2020) is proposed to mine the knowledge from both labeled and extra unlabeled data to boost few-shot learners. The core challenge in SSFSL is how to explore the auxiliary information from these unlabeled thoroughly. Previous SSFSL works indicate that graph-based models (Liu et al. 2018, Ziko et al. 2020) can learn a better classifier than inductive ones (Ren et al. 2018, Li, Sun, Liu, Zhou, Zheng, Chua & Schiele 2019, Yu et al. 2020) since these methods directly model the relationship between the labeled and unlabeled samples during the inference. However, current graph-based models adopt the Laplace learning (Zhu et al. 2003) to conduct label propagation. The solutions of Laplace learning generate restricted spikes near the labeled. Still, they are essentially constant faraway from labeled samples, *i.e.,* the label propagation of these models is not robust and accurate, especially with few labeled examples. Therefore, these models suffer from the underdeveloped message-passing capacity for

the labels. On the other hand, most SSFSL methods adapt the feature embedding pre-trained on base-class data (meta- or transfer- pre-trained) as the novel-class embedding. This may lead to the embedding degeneration problem, as the pre-trained feature encoder is designed for base-class recognition. It tends to learn the embedding that represents only base-class information and lose information that might be useful outside base classes.

To address the above issues, we propose a novel transfer-learning-based SSFSL method named Poisson Transfer Network (PTN). Specifically, ***to improve the capacity of graph-based SSFSL models in message passing***, we propose to revise the Poisson model tailored for few-shot problems by incorporating the query feature calibration and the Poisson MBO model. Poisson learning (Calder et al. 2020) has been provably more stable and informative than traditional Laplace learning in low label rate semi-supervised problems. However, directly employing Poisson MBO for SSFSL may suffer from the cross-class bias due to the data distribution drift between the support and query data. Therefore, we improve the Poisson MBO model by explicitly eliminating the cross-class bias before label inference. ***To tackle the novel-class embedding degeneration problem***, we propose to transfer the pre-trained base-class embedding to the novel-class embedding by adopting unsupervised contrastive training (He et al. 2020, Chen et al. 2020) on the extra unlabeled novel-class data. Constraining the distances between the augmented positive pairs while pushing the negative ones distant, the proposed transfer scheme captures the novel-class distribution implicitly. This strategy effectively avoids the possible overfitting of retraining feature embedding on the few labeled instances.

By integrating the Poisson learning and the novel-class-specific embedding, the proposed PTN model can fully explore the auxiliary information of extra unlabeled data for SSFSL tasks. The contributions are summarized as follows:

- We propose a Poisson learning based method to improve the capacity of mining the relations between the labeled and unlabeled data for

graph-based SSFSL.

- We propose to adopt unsupervised contrastive learning in the representation learning with extra unlabeled data to improve the generality of the pre-trained base-class embedding for novel-class recognition.
- Comprehensive experimental analyses are conducted on RPSI and two generic image datasets to investigate the effectiveness of PTN, and PTN achieves state-of-the-art performance.

5.2 Methodology

5.2.1 Problem Definition

In the standard few-shot learning, there exists a labeled support set S of C different classes, $S = \{(x_s, y_s)\}_{s=1}^{K\times C}$, where x_s is the labeled sample and y_s denote its label. We use the standard basis vector $\mathbf{e}_i \in \mathbb{R}^C$ to represent the i-th class, *i.e.*, $y_s \in \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_C\}$. Given an unlabeled query sample x_q from the query set $Q = \{x_q\}_{q=1}^{V}$, the goal is to assign the query to one of the C support classes. The labeled support set and unlabeled query set share the same label space, and the novel-class dataset $\mathcal{D}_{novel}$ is thus defined as $\mathcal{D}_{novel} = S \cup Q$. If S contains K labeled samples for each of C categories, the task is noted as a C-way-K-shot problem. It is far from obtaining an ideal classifier with the limited annotated S. Therefore, few-shot models usually utilize a fully annotated dataset, which has similar data distribution but disjoint label space with $\mathcal{D}_{novel}$ as an auxiliary dataset $\mathcal{D}_{base}$ noted as the base-class set.

For the semi-supervised few-shot learning (SSFSL), we have an extra unlabeled support set $U = \{x_u\}_{u=1}^{N}$. These additional N unlabeled samples are usually from each of the C support classes in standard-setting or other novel-class under distractor classification settings. Then the new novel-class dataset $\mathcal{D}_{novel}$ is defined as $\mathcal{D}_{novel} = S \cup Q \cup U$. The goal of SSFSL is

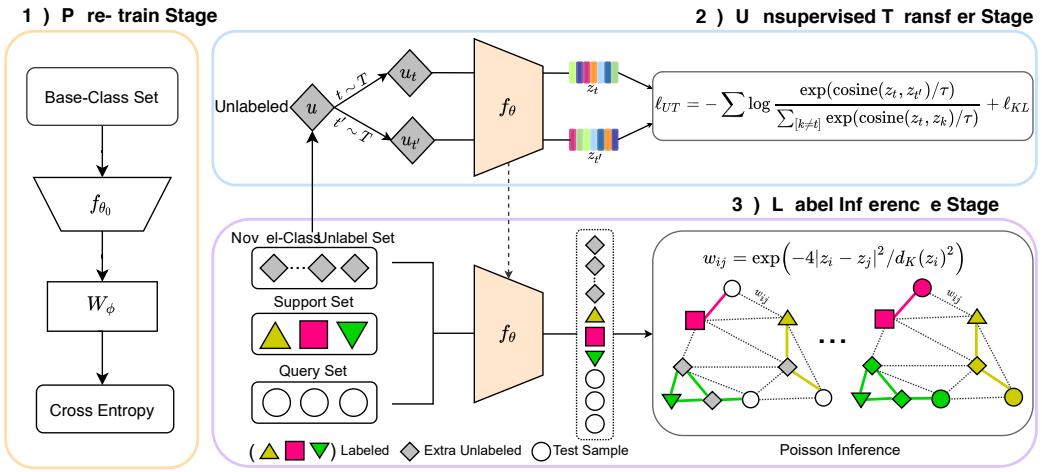


Figure 5.1: The overview of the proposed PTN. A feature embedding f_{θ_0} is pre-trained from the base-class set using standard cross-entropy loss first. This embedding is then fine-tuned with the external novel-class unlabeled data by adopting unsupervised transferring loss ℓ_{UT} to generate f_θ. Finally, we revise a graph model named PoissonMBO to conduct the query label inference. We also denote the Novel-Class Unlabel Set (U), Support Set (S), and Query Set (Q) with different colors and shapes.

to maximize the value of the extra unlabeled data to improve the few-shot methods.

For a clear understanding, the details of the proposed PTN are introduced as follows: we first introduce the proposed Representation Learning, and then we illustrate the proposed Poisson learning model for label inference.

5.2.2 Representation Leaning

The representation learning aims to learn a well-generalized novel-class embedding through Feature Embedding Pre-training and Unsupervised Embedding Transfer.

Feature Embedding Pre-training

On the left part of Figure 5.1, the first step of PTN is the feature embedding pre-training. By employing the cross-entropy loss to train the encoder with $\mathcal{D}_{base}$, we obtain the base encoder f_{θ_0} in a fully-supervised way, which is the same as (Chen et al. 2018, Yu et al. 2020, Tian et al. 2020). This stage can generate powerful embedding for the downstream few-shot learner.

Unsupervised Embedding Transfer

Directly employing the pre-trained base-class embedding for the novel-class may suffer from the degeneration problem. However, retraining the base-class embedding with the limited labeled instances is easy to lead to overfitting. How can we train a novel-class embedding to represent things beyond labels when our only supervision is the limited labels? Our solution is unsupervised contrastive learning. Unsupervised learning, especially Contrastive learning (He et al. 2020, Chen et al. 2020), recently has shown great potential in representation learning for various downstream vision tasks, and most of these works training a model from scratch. However, unsupervised pre-trained models perform worse than fully-supervised pre-trained models. Unlike previous works, we propose to adopt contrastive learning to retrain the pre-trained embedding with the unlabeled novel data. In this way, we can learn a decent novel-class embedding by integrating the fully-supervised pre-trained scheme with unsupervised contrastive fine-tuning.

Specifically, for a minibatch of n examples from the unlabeled novel-class subset $U_i = \{x_u\}_{u=1}^{n}$, randomly sampling two data augmentation operators $t, t' \in T$, we can generate a new feature set $Z = \{Z_t = \{f_{\theta_0} \circ t(x_u)\}_{u=1}^{n}\} \cup \{Z_{t'} = \{f_{\theta_0} \circ t'(x_u)\}_{u=1}^{n}\}$, resulting in n pairs of feature points. We treat each feature pair from the same raw data input as positive pair and the other $2(n-1)$ feature points as negative samples. Then the contrastive loss for the minibatch is defined as

$$\ell_{cont} = -\sum_{i,j=1}^{n} \log \frac{\exp\left(\text{cosine}\left(z_i, z_j\right)/\tau\right)}{\sum_{k \neq i} \exp\left(\text{cosine}\left(z_i, z_k\right)/\tau\right)}, \tag{5.1}$$

where z_i, z_j denote a positive feature pair from Z, τ is a temperature parameter, and cosine$(\cdot)$ represents the cosine similarity. Then, we adopt a Kullback-Leibler divergence (ℓ_{KL}) between two feature subsets Z_t and $Z_{t'}$ as the regulation term. Therefore, the final unsupervised embedding transfer loss ℓ_{UT} is defined as

$$\ell_{UT} = \ell_{cont} + \lambda \ell_{KL}(Z_t \parallel Z_{t'}). \tag{5.2}$$

By training the extra unlabeled data with this loss, we can learn a robust novel-class embedding f_θ from f_{θ_0}.

5.2.3 Poisson Label Inference

Previous studies (Zhu et al. 2003, Zhou, Bousquet, Lal, Weston & Schölkopf 2004, Zhu, Lafferty & Rosenfeld 2005, Liu et al. 2018, Ziko et al. 2020) indicate that the graph-based few-shot classifier has shown superior performance against inductive ones. Therefore, we propose constructing the classifier with a graph-based Poisson model, which adopts a different optimizing strategy with representation learning. Poisson model (Calder et al. 2020) has been proved superior over traditional Laplace-based graph models (Zhu et al. 2003, Zhou et al. 2004) both theoretically and experimentally, especially for the low label rate semi-supervised problem. However, directly applying this model to the few-shot task will suffer from a cross-class bias challenge caused by the data distribution bias between support data (including labeled support and unlabeled support data) and query data.

Therefore, we revise this powerful model by eliminating the support-query bias as the classifier. We explicitly propose a query feature calibration strategy before the final Poisson label inference. It is worth noticing that the proposed graph-based classifier can be directly appended to the pre-trained embedding without adopting the unsupervised embedding transfer training. We dob this baseline model as *Decoupled Poisson Network* (*DPN*).

Query Feature Calibration

The support-query data distribution bias, also referred to as the cross-class bias (Liu, Song & Qin 2020), is one of the reasons for the degeneration of the few-shot learner. In this chapter, we design a simple but effective mechanism to eliminate this distribution bias for Poisson graph inference. For a SSFSL task, we fuse the labeled support set S and the extra unlabeled set U as the final support set $B = S \cup U$. We denote the normalized embedded support feature set and query feature set as $Z_b = \{z_b\}$ and $Z_q = \{z_q\}$, and the cross-class bias is defined as

$$\begin{aligned} \Delta_{\text{cross}} &= \mathbb{E}_{z_b \sim p_{\mathcal{B}}}\left[z_b\right] - \mathbb{E}_{z_q \sim p_{\mathcal{Q}}}\left[z_q\right] \\ &= \frac{1}{|\mathcal{B}|}\sum_{b=1}^{|\mathcal{B}|} z_b - \frac{1}{|\mathcal{Q}|}\sum_{q=1}^{|\mathcal{Q}|} z_q. \end{aligned} \tag{5.3}$$

We then add the bias Δ_{cross} to query features. To such a degree, support-query bias is somewhat eliminated. After that, a Poisson MBO model is adopted to infer the query label.

The Poisson Merriman–Bence–Osher Model

We denote the embedded feature set as $Z_{novel} = Z_b \cup Z_q = \{z_1, z_2, \ldots, z_m\}$ ($m = K \times C + N + V$), where the first $K \times C$ feature points belong to the labeled support set, the last V feature points belong to the query set, and the remaining N points denote the unlabeled support set. We build a graph with the feature points as the vertices, and the edge weight w_{ij} is the similarity between feature point z_i and z_j, defined as $w_{ij} = \exp\left(-4\left|z_i - z_j\right|^2 / d_K\left(z_i\right)^2\right)$, where $d_K\left(z_i\right)^2$ is the distance between z_i and its K-th nearest neighbor. We set $w_{ij} \geq 0$ and $w_{ij} = w_{ji}$. Correspondingly, we define the weight matrix as $W = [w_{ij}]$, the degree matrix as $D = \operatorname{diag}([d_i = \sum_{j=1}^{m} w_{ij}])$, and the unnormalized Laplacian as $L = D - W$. As the first $K \times C$ feature points have the ground-truth label, we use $\bar{y} = \frac{1}{K \times C}\sum_{s=1}^{K \times C} y_s$ to denote the average label vector, and we let indicator $\mathbb{I}_{ij} = 1$ if $i = j$, else $\mathbb{I}_{ij} = 0$. The goal of this

model is to learn a classifier $g : z \to \mathbb{R}^C$. By solving the Poisson equation:

$$Lg\left(z_i\right) = \sum_{j=1}^{K\times C} \left(y_j - \bar{y}\right) \mathbb{I}_{ij} \quad \text{for } i = 1, \ldots, m, \tag{5.4}$$

satisfying $\sum_{i=1}^{m} \sum_{k=1}^{m} w_{ik} g\left(z_i\right) = 0$, we can then result in the label prediction function $g(z_i) = (g_1(z_i), g_2(z_i), \ldots, g_C(z_i))$. The predict label $\hat{y}_i$ of vertex z_i is then determined as $\hat{y}_i = \arg\max_{j\in\{1,\ldots,C\}} \{g_j(x_i)\}$. Let G denote the set of $m \times C$ matrix, which is the prediction label matrix of the all data. We concatenate the support label to form a label matrix $Y = [y_s] \in \mathbb{R}^{C\times(K\times C)}$. Let $A = [Y - \bar{y}, \mathbf{0}^{C\times(m-K\times C)}]$ denotes the initial label of all the data, in which all unlabeled data's label is zero. The query label of Eq. (5.4) can be determined by:

$$G^{tp+1} = G^{tp} + D^{-1}(A^T - LG^{tp}), \tag{5.5}$$

where G^{tp} denotes the predicted labels of all data at the timestamp tp. We can get a stable classifier g with a certain number of iteration using Eq. (5.5). After that, we adopt a graph-cut method to improve the inference performance by incrementally adjusting the classifier's decision boundary. The graph-cut problem is defined as

$$\min_{\substack{g:Z\to H \\ (g)_z = o}} \left\{ g^T Lg - \mu \sum_{i=1}^{K\times C} \left(y_i - \bar{y}\right) \cdot g\left(z_i\right) \right\}, \tag{5.6}$$

where $H = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_C\}$ denotes the annotated samples' label set, $(g)_z = \frac{1}{m}\sum_{i=1}^{m} g(z_i)$ is the fraction of vertices to each of C classes, and $o = [o_1, o_2, \ldots, o_C]^T \in \mathbb{R}^C$ is the piror knowledge of the class size distribution that o_i is the fraction of data belonging to class i. With the constraint $(g)_z = o$, we can encode the prior knowledge into the Poisson Model. $g^T Lg = \frac{1}{2}\sum_{i,j=1}^{m} w_{ij}(g(i) - g(j))^2$, this term is the graph-cut energy of the classification given by $g = [g(z_1), g(z_2), \ldots, g(z_m)]^T$, widely used in semi-supervised graph models (Zhu et al. 2003, Zhu et al. 2005, Zhou et al. 2004).

In Eq.(5.6), the solution will get discrete values, which is hard to solve. To relax this problem, we use the Merriman-Bence-Osher (MBO) scheme (Garcia-Cardona, Merkurjev, Bertozzi, Flenner & Percus 2014) by adopting the Ginzburg-Landau approximation:

$$\begin{aligned} &\min_{\substack{g\in \mathrm{SP}\{Z\to\mathbb{R}^C\}\\ (g)_z=o}} \left\{ \mathrm{GL}_{\tau'}(g) - \mu \sum_{i=1}^{K\times C} (y_i - \bar{y}) \cdot g(z_i) \right\}, \\ &\mathrm{GL}_{\tau'}(g) = g^T L g + \frac{1}{\tau'} \sum_{i=1}^{m} \prod_{j=1}^{C} |g(z_i) - \mathbf{e}_j|^2 . \end{aligned} \tag{5.7}$$

In Eq.(5.7), $\mathrm{SP}\{Z \to \mathbb{R}^C\}$ represents the space of projections $g : Z \to \mathbb{R}^C$, which allow the classifier g to generate any real values, instead of the discrete value from H in Eq.(5.6). More importantly, this leads to a more efficiently computation of the Poisson model. The Eq.(5.7) can be efficiently solved with alternates gradient decent strategy, as shown in lines 9-20 of Algorithm 5.1.

5.2.4 Proposed Algorithm

The overall proposed algorithm is summarized in Algorithm 5.1. Given the base-class set $\mathcal{D}_{base}$, novel-class set $\mathcal{D}_{novel}$, prior classes' distribution o, and other parameters, PTN will predict the query samples' label $G \in \mathbb{R}^{V\times C}$. The query label $\hat{y}_q$ is then determined as $\hat{y}_q = \arg\max_{1\le j\le C} G_{qj}$. More specifically, once the encoder f_θ is learned using the base set $\mathcal{D}_{base}$, we employ the proposed unsupervised embedding transfer method in Step 2 in Algorithm 5.1. After that, we build the graph with the feature set Z_{novel} and compute the related matrices W, D, L, A in Step 3-5. In the label inference stage in Steps 6-20, we first apply Poisson model to robust propagate the labels in Step 7, and then solve the graph-cut problem by using MBO scheme in a certain Steps of gradient-descent to boost the performance. The stop condition in Step 7 follow the constraint: $\left\|\mathbf{sp}_{tp} - W\mathbf{1}/\left(\mathbf{1}^T W \mathbf{1}\right)\right\|_\infty \le 1/m$, where $\mathbf{1}$ is a all-ones column vector, $\mathbf{sp}_{tp} = WD^{-1}\mathbf{sp}_{tp-1}$, $\mathbf{sp_0}$ is a m-column vector with ones in the first $K \times C$ positions and zeros elsewhere. Steps 9-19 are aimed to solve the graph-cut problem in Eq.(5.7), To solve the problem, we first divide the Eq.(5.7) into $E_1 = g^T L g - \mu \sum_{i=1}^{K\times C} (y_i - \bar{y}) \cdot g(z_i)$ and $E_2 = \frac{1}{\tau'} \sum_{i=1}^{m} \prod_{j=1}^{C} |g(z_i) - \mathbf{e}_j|^2$, and then employing the gradient decent

Algorithm 5.1 PTN for SSFSL

Input : $\mathcal{D}_{base}$, $\mathcal{D}_{novel} = S \cup U \cup Q$,
o, μ, M_1, M_2, M_3

Output: Query samples' label prediction G

Train a base model $\mathbf{W}_\phi \circ f_{\theta_0}(x)$ with all samples and labels from $\mathcal{D}_{base}$;
Apply unsupervised embedding transfer method to fine-tune the f_{θ_0} with novel unlabeled data U by using ℓ_{UT} in Eq. (5.2), and result in f_θ;
Apply f_θ to extract features on D_{novel} as Z_{novel};
Apply query feature calibration using Eq. (5.3);
Compute W, D, L, A according to Z_{novel}, $G \leftarrow \mathbf{0}^{m \times C}$
PoissonMBO
    Update G uisng Eq. (5.5) with given steps
    $\mathrm{d}_{mx} \leftarrow 1/\max_{1 \le i \le m} D_{ii}$, $G \leftarrow \mu G$
    for $i = 1$ **to** M_1 **do**
        for $j = 1$ **to** M_2 **do**
            $G \leftarrow G - \mathrm{d}_{mx}\left(LG - \mu A^T\right)$
        end
        $r \leftarrow \mathbf{ones}(1, C)$
        for $j = 1$ **to** M_3 **do**
            $\hat{o} \leftarrow \frac{1}{n}\mathbf{1}^T\mathbf{Proj}_H(G \cdot \mathrm{diag}(r))$
            $r \leftarrow \max\left(\min\left(r + \varphi \cdot (o - \hat{o}), \upsilon_\alpha\right), \upsilon_\sigma\right)$
        end
        $G \leftarrow \mathbf{Proj}_H(G \cdot \mathrm{diag}(r))$
    end
$G \leftarrow G[m - V : m, :]$;

alternative on these two energy functions. Steps 10-12 are used to optimize E_1. We optimize E_2 in Steps 14-17, $\mathbf{Proj}_H : \mathbb{R}^C \to H$ is the closet point projection, $r = [r_1, \ldots, r_C]^T$ $(r_i > 0)$, φ is the time step, and $\upsilon_\alpha, \upsilon_\sigma$ are the clipping values, By adopting the gradient descent scheme in Steps 14-17, the vector r is generated that also satisfies the constraint $(g)_z = o$ in Eq.(5.7). After obtaining the PoissonMBO's solution G, the query samples' label prediction matrix is resolved by Step 20.

The main inference complexity of PTN is $\mathcal{O}(M_1 M_2 E)$ (Steps 9-19 in Algorithm 5.1) , where E is the number of edges in the graph. As a graphed-based model, PTN's inference complexity is heavier than inductive models. However, some studies (Liu et al. 2018, Calder et al. 2020) indicate that this complexity is affordable for few-shot tasks since the data scale is not very big. Moreover, we do not claim that our model is the final solution for SSFSL. We aim to design a novel framework to well use the extra unlabeled information. We report inference time comparison experiments in Table 5.8. The average inference time of PTN is 13.68s.

5.3 Experiments

5.3.1 Datasets

We first validate the PTN model on the RPSI Defects dataset, as presented in Chapter 3.2 and 4.6.1, which contains 2336 images with 39 categories. Then we evaluate the PTN model on two generic few-shot benchmarks: miniImageNet and tieredImageNet. The miniImageNet dataset (Vinyals et al. 2016) is a subset of the ImageNet, consisting of 100 classes, and each class contains 600 images. The image size of miniImageNet is 84×84. We follow the standard split in (Vinyals et al. 2016, Tian et al. 2020) that divide the dataset into 64 base classes, 16 validation classes, and 20 test classes. The tieredImageNet (Ren et al. 2018) is another subset of ImageNet which contains 608 classes instead. We also adopt the standard split in (Ren et al. 2018, Liu et al. 2018) that divides the dataset into 351 base classes, 97 validation classes, and 160

test classes. We resize the images from RPSI Defects and tieredImageNet to 84×84 pixels, and randomly select C classes from the novel class to construct the few-shot task. Within each class, K examples are selected as the labeled data, and V examples from the rest as queries. The extra N unlabeled samples are selected from the C classes or rest novel classes. We set $C = 5, K = \{1, 5\}, V = 15$ and study different sizes of N. We run 600 few-shot tasks and report the average accuracy using the 95% confidence interval (Neyman 1937).

5.3.2 Implementation Details

Similar to previous works (Rusu, Rao, Sygnowski, Vinyals, Pascanu, Osindero & Hadsell 2018, Dhillon, Chaudhari, Ravichandran & Soatto 2019, Liu et al. 2020, Tian et al. 2020, Yu et al. 2020), we adopt the wide residual network (WRN-28-10) (Zagoruyko & Komodakis 2016) as the backbone of our base model $W_\phi \circ f_{\theta_0}$. We follow the protocols in (Tian et al. 2020, Yu et al. 2020) fusing the base and validation classes to train the base model from scratch. The batch size is set to 64 with SGD learning rate as 0.05 and weight decay as $5e^{-4}$. The learning rate is reduced by 0.1 after 60 and 80 epochs. And the base model is trained with 100 epochs.

In unsupervised embedding transfer, the data augmentation T is defined the same as (Lee, Maji, Ravichandran & Soatto 2019, Tian et al. 2020). For fair comparisons against TransMatch (Yu et al. 2020), we conduct data argumentation for each labeled image ten times by random transformations and generate the prototypes of each class as labeled samples. We adopt the SGD optimizer with a momentum of 0.9. The learning rate is initialized as $1e^{-3}$, and the cosine learning rate scheduler is used for ten epochs. The batch size is set to 80 with $\lambda = 1$ in Eq. (5.2). For Poisson inference, we build the graph adopting K-nearest neighbors with Gaussian weights to each sample. We set $K = 30$, and the weight matrix W is summarized with $w_{ii} = 0$, which accelerates the convergence of the iteration in Algorithm 5.1 without changing the solution of Equation 5.4. We set the max $tp = 100$ in Step 7 of

Algorithm 5.1 by referring to the stop constraint discussed in the Proposed Algorithm section. We set hyper-parameters $\mu = 1.5, M_1 = 20, M_2 = 40$ and $M_3 = 100$ empirically. Moreover, we set $\varphi = 10, \upsilon_\alpha = 0.5, \upsilon_\sigma = 1.0$.

5.3.3 Experimental Results

Experimental Results on RPSI Defects Dataset

First of all, we compare the PTN with other state-of-the-art few-shot models on the RPSI Defects dataset. During our experiments, we group the compared methods into five categories, and the experimental results on the two sets are summarized in Table 5.1. Due to the comparatively small size of the RPSI Defects, we choose 50 unlabeled samples in each class. As indicated in the table, PTN gets the best performance compared with both generic few-shot learning methods and fine-grained few-shot learning methods with a large margin, which indicates the proposed models effectively utilize the unlabeled information for few-shot RPSI Defects recognition. More specifically, under the five-way-one-shot setting, the classification accuracy of PTN is 74.11% vs. 68.79% TOAN:ResNet; under the five-way-five-shot setting, the classification accuracy of PTN is 86.64% vs. 80.94% (Tian et al. 2020). Compared with semi-supervised few-shot methods (TPN, LaplacianShot, Masked Soft k-Means, and TPN-semi), PTN also achieved the best performance. For example, in the 5-way-5-shot setting, the classification accuracy of PTN is 86.64%, while the state-of-the-art LaplacianShot (Ziko et al. 2020) only achieved 82.36%. This validates the effectiveness of the proposed Poisson-based PTN models on the RPSI Defects recognition.

Experimental Results on Two Generic Datasets

In the second experiment, we conduct a further comparison of PTN and DPN with other approaches on two benchmark datasets, and the experimental comparisons are shown in Table 5.2. With the auxiliary unlabeled data available, our proposed PTN outperforms the metric-based and optimization-

Methods	Type	Backbone	RPSI Defect	
			1-shot	5-shot
Matching Network (Vinyals et al. 2016)	Metric, Meta	ConvNet-64	50.47±0.90	65.33±0.74
Prototypical-Net (Snell et al. 2017)	Metric, Meta	ConvNet-256	47.54±0.89	64.97±0.76
Relation Network (Sung, Yang, Zhang, Xiang, Torr & Hospedales 2018*a*)	Metric, Meta	ConvNet-64	57.60±0.87	66.43±0.72
PABN+$_{cpt}$	Metric, Meta	ConvNet-64	62.24±0.90	70.73±0.81
LRPABN$_{cpt}$	Metric, Meta	ConvNet-64	64.51±0.87	77.21±0.79
TOAN	Metric, Meta	ConvNet-64	66.61±0.90	79.92±0.80
TOAN:ResNet	Metric, Meta	ResNet-12	68.79±0.85	82.07±0.78
RFS (Tian et al. 2020)	Metric, Transfer	ResNet-12	68.42±0.80	80.94±0.75
MAML (Finn, Abbeel & Levine 2017*b*)	Optimization, Meta	ConvNet-64	46.90±1.32	62.44±0.98
LEO (Rusu et al. 2018)	Optimization, Meta	WRN-28-10	63.27±0.21	75.70±0.30
MetaOptNet (Lee et al. 2019)	Optimization, Meta	ResNet-12	64.35±0.75	76.12±0.62
TPN (Liu et al. 2018)	Transductive, Meta	ConvNet-64	61.17±0.92	68.52±0.74
LaplacianShot (Ziko et al. 2020)	Transductive, Transfer	ResNet-12	69.18±0.22	82.36±0.09
Masked Soft k-Means (Ren et al. 2018)	Semi, Meta	ConvNet-128	46.77±0.43	62.89±0.21
TPN-semi (Liu et al. 2018)	Semi, Meta	ConvNet-64	49.54±0.18	66.62±0.15
DPN (Ours)	Semi, Transfer	WRN-28-10	**72.21±1.24**	**84.53±0.90**
PTN (Ours)	Semi, Transfer	WRN-28-10	**74.11±1.10**	**86.64±0.78**

Table 5.1: The five-way-one-shot and five-way-five-shot image classification accuracy (%) on the RPSI Defect dataset with 95% confidence interval.

based few-shot models by large margins, indicating that the proposed PTN model effectively utilizes the unlabeled information for assisting few-shot recognition. By integrating the unsupervised embedding transfer and PoissonMBO classifier, PTN achieves superior performance over both transductive and existing SSFSL approaches. Specifically, under the five-way-one-shot setting, the classification accuracies are 81.57% vs. 63.02% TransMatch (Yu et al. 2020), 84.70% vs. 80.30% LaplacianShot (Ziko et al. 2020) on miniImageNet and tieredImageNet, respectively; under the five-way-five-shot setting, the classification accuracies are 88.43% vs. 78.70% LST (Li, Sun, Liu, Zhou, Zheng, Chua & Schiele 2019), 89.14% vs. 81.89% BD-CSPN (Liu et al. 2020) on miniImageNet and tieredImageNet, respectively. These results demonstrate the superiority of PTN for SSFSL tasks.

Different Extra Unlabeled Samples

We show the results of selecting different numbers of extra unlabeled instances from the minImageNet dataset, as shown in Table 5.3. For Num_U = 0, PTN* can be viewed as the transductive model without extra unlabeled data, where we treat query samples as the unlabeled data, and we do not

Methods	Type	Backbone	miniImageNet	
			1-shot	5-shot
Prototypical-Net (Snell et al. 2017)	Metric, Meta	ConvNet-256	49.42±0.78	68.20±0.66
Relation Network (Sung et al. 2018*a*)	Metric, Meta	ConvNet-64	50.44±0.82	65.32±0.70
TADAM (Oreshkin, López & Lacoste 2018)	Metric, Meta	ResNet-12	58.50±0.30	76.70±0.30
DPGN (Yang, Li, Zhang, Zhou, Zhou & Liu 2020)	Metric, Meta	ResNet-12	67.77±0.32	84.60±0.43
RFS (Tian et al. 2020)	Metric, Transfer	ResNet-12	64.82±0.60	82.14±0.43
MAML (Finn et al. 2017*b*)	Optimization, Meta	ConvNet-64	48.70±1.84	63.11±0.92
SNAIL (Mishra, Rohaninejad, Chen & Abbeel 2018)	Optimization, Meta	ResNet-12	55.71±0.99	68.88±0.92
LEO (Rusu et al. 2018)	Optimization, Meta	WRN-28-10	61.76±0.08	77.59±0.12
MetaOptNet (Lee et al. 2019)	Optimization, Meta	ResNet-12	64.09±0.62	80.00±0.45
TPN (Liu et al. 2018)	Transductive, Meta	ConvNet-64	55.51±0.86	69.86±0.65
BD-CSPN (Liu et al. 2020)	Transductive, Meta	WRN-28-10	70.31±0.93	81.89±0.60
Transductive Fine-tuning (Dhillon et al. 2019)	Transductive, Transfer	WRN-28-10	65.73±0.68	78.40±0.52
LaplacianShot (Ziko et al. 2020)	Transductive, Transfer	DenseNet	75.57±0.19	84.72±0.13
Masked Soft k-Means (Ren et al. 2018)	Semi, Meta	ConvNet-128	50.41±0.31	64.39±0.24
TPN-semi (Liu et al. 2018)	Semi, Meta	ConvNet-64	52.78±0.27	66.42±0.21
LST (Li, Sun, Liu, Zhou, Zheng, Chua & Schiele 2019)	Semi, Meta	ResNet-12	70.10±1.90	78.70±0.80
TransMatch (Yu et al. 2020)	Semi, Transfer	WRN-28-10	62.93±1.11	82.24±0.59
DPN (Ours)	Semi, Transfer	WRN-28-10	79.67±1.06	86.30±0.95
PTN (Ours)	Semi, Transfer	WRN-28-10	**82.66±0.97**	**88.43±0.67**
Methods	Type	Backbone	tieredImageNet	
			1-shot	5-shot
Prototypical-Net (Snell et al. 2017)	Metric, Meta	ConvNet-256	53.31±0.89	72.69±0.74
Relation Network (Sung et al. 2018*a*)	Metric, Meta	ConvNet-64	54.48±0.93	71.32±0.78
DPGN (Yang et al. 2020)	Metric, Meta	ResNet-12	72.45±0.51	87.24±0.39
RFS (Tian et al. 2020)	Metric, Transfer	ResNet-12	71.52±0.69	86.03±0.49
MAML (Finn et al. 2017*b*)	Optimization, Meta	ConvNet-64	51.67±1.81	70.30±1.75
LEO (Rusu et al. 2018)	Optimization, Meta	WRN-28-10	66.33±0.05	81.44±0.09
MetaOptNet (Lee et al. 2019)	Optimization, Meta	ResNet-12	65.81±0.74	81.75±0.53
TPN (Liu et al. 2018)	Transductive, Meta	ConvNet-64	59.91±0.94	73.30±0.75
BD-CSPN (Liu et al. 2020)	Transductive, Meta	WRN-28-10	78.74±0.95	86.92±0.63
Transductive Fine-tuning (Dhillon et al. 2019)	Transductive, Transfer	WRN-28-10	73.34±0.71	85.50±0.50
LaplacianShot (Ziko et al. 2020)	Transductive, Transfer	DenseNet	80.30±0.22	87.93±0.15
Masked Soft k-Means (Ren et al. 2018)	Semi, Meta	ConvNet-128	52.39±0.44	69.88±0.20
TPN-semi (Liu et al. 2018)	Semi, Meta	ConvNet-64	55.74±0.29	71.01±0.23
LST (Li, Sun, Liu, Zhou, Zheng, Chua & Schiele 2019)	Semi, Meta	ResNet-12	77.70±1.60	85.20±0.80
DPN (Ours)	Semi, Transfer	WRN-28-10	82.18±1.06	88.02±0.72
PTN (Ours)	Semi, Transfer	WRN-28-10	**84.70±1.14**	**89.14±0.71**

Table 5.2: The five-way, one-shot and five-shot recognition accuracy (%) on the two datasets with 95% confidence interval. We mark the best performance in bold. The upper and lower parts of the table show the results on miniImageNet and tieredImageNet, respectively.

Methods	Num_U	1-shot	5-shot
PTN*	0	76.20±0.82	84.25±0.61
PTN	0	77.01±0.94	85.32±0.68
PTN	20	77.20±0.92	85.93±0.82
PTN	50	79.92±1.06	86.09±0.75
PTN	100	81.57±0.94	87.17±0.58
PTN	200	**82.66±0.97**	**88.43±0.76**

Table 5.3: The five-way-one-shot and five-way-five-shot recognition accuracy (%) using various number of extra unlabeled samples on the miniImageNet dataset. PTN* denotes that we adopt PTN as the transductive model without fine-tune embedding. We mark the best results in bold.

fine-tune the embedding with query labels for fair comparisons. Contrary to PTN*, the proposed PTN model utilizes the query samples to fine-tune the embedding when Num_U=0. It can be observed that PTN achieves better performances with more extra unlabeled samples, which indicates the effectiveness of PTN in mining the unlabeled auxiliary information for the few-shot problem.

We conduct further experiments to investigate the current semi-supervised few-shot methods in mining the value of the unlabeled data. All approaches are based on a pre-trained WRN-28-10 (Zagoruyko & Komodakis 2016) model for fair comparisons. As indicated in Table 5.3.3, with more unlabeled samples, all the models achieve higher classification performances. Nevertheless, the proposed PTN model gets the highest performance among the compared approaches, which validates the superior capacity of the proposed model in using the extra unlabeled information for boosting few-shot methods.

	*mini*ImageNet 5-way-1-shot				
	0	20	50	100	200
TransMatch (Yu et al. 2020)	-	58.43±0.93	61.21±1.03	63.02±1.07	62.93±1.11
Label Propagation (Zhou et al. 2004)	69.74±0.72	71.80±1.02	72.97±1.06	73.35±1.05	74.04±1.00
PoissonMBO (Calder et al. 2020)	74.79±1.06	76.01±0.99	76.67±1.02	78.28±1.02	79.67±1.02
DPN (Ours)	75.85±0.97	76.10±1.06	77.01±0.92	79.55±1.13	80.00±0.83
PTN (Ours)	**77.01±0.94**	**77.20±0.92**	**79.92±1.06**	**81.57±0.94**	**82.66±0.97**
	*mini*ImageNet 5-way-5-shot				
	0	20	50	100	200
TransMatch	-	76.43±0.61	79.30±0.59	81.19±0.59	82.24±0.59
Label Propagation	75.50±0.60	78.47±0.60	80.40±0.61	81.65±0.59	82.60±0.68
PoissonMBO	83.89±0.66	84.43±0.67	84.94±0.82	85.51±0.81	86.30±0.65
DPN (Ours)	84.74±0.63	85.04±0.66	85.36±0.60	86.09±0.63	87.17±0.51
PTN (Ours)	**85.32±0.68**	**85.93±0.82**	**86.09±0.75**	**87.17±0.58**	**88.43±0.76**

Table 5.4: Accuracy with various extra unlabeled samples for different semi-supervised few-shot methods on the *mini*ImageNet dataset. All results are averaged over 600 episodes. We mark the best results in bold.

Results with Distractor Classes

Inspired by (Ren et al. 2018, Liu et al. 2018, Yu et al. 2020), we further investigate the influence of distractor classes, where the extra unlabeled data are collected from classes with no overlaps to labeled support samples. We follow the settings in (Ren et al. 2018, Liu et al. 2018). As shown in Figure 5.2, even with distractor class data, the proposed PTN still outperforms other SSFSL approaches by a significant margin, which indicates the robustness of the proposed PTN while tackling distracted unlabeled data. More specifically, we present the experimental results of PTN on both *mini*ImageNet (Vinyals et al. 2016) and *tiered*ImageneNet (Ren et al. 2018) datasets under different settings in Table 5.5 and Table 5.6. Since PTN propagates the labels across all samples according to the graph edges, distractor items could be harmful and interfere with label propagation, as edges were built based on the similarity between samples. Therefore, the performance degraded under the distractor setting, as indicated in Table 5.5 and Table 5.6.

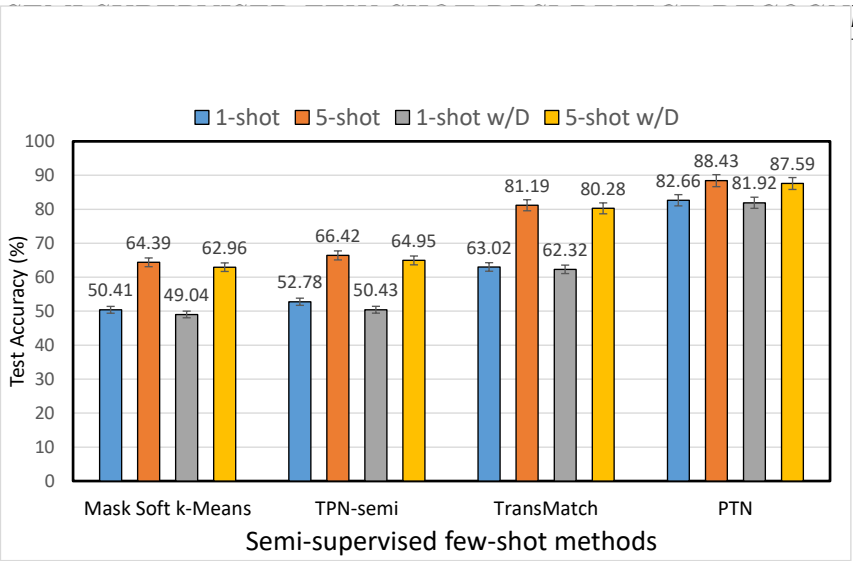


Figure 5.2: The five-way-one-shot and five-way-five-shot classification accuracy (%) using different number of extra unlabeled samples on the miniImageNet dataset. w/D means with distractor classes.

The data distribution of distractor classes

Table 5.5: Distraction comparison on the *mini*ImageNet dataset.

Methods	1-shot	5-shot	1-shot w/D	5-shot w/D
Soft K-Means (Ren et al. 2018)	50.09±0.45	64.59±0.28	48.70±0.32	63.55±0.28
Soft K-Means+Cluster (Ren et al. 2018)	49.03±0.24	63.08±0.18	48.86±0.32	61.27±0.24
Masked Soft k-Means (Ren et al. 2018)	50.41±0.31	64.39±0.24	49.04±0.31	62.96±0.14
TPN-semi (Liu et al. 2018)	52.78±0.27	66.42±0.21	50.43±0.84	64.95±0.73
TransMatch (Yu et al. 2020)	63.02±1.07	81.19±0.59	62.32±1.04	80.28±0.62
PTN (Ours)	**82.66±0.97**	**88.43±0.67**	**81.92±1.02**	**87.59±0.61**

* "w/D" means with distraction classification. In this setting, many extra unlabeled samples are from the distraction classes, which is different from the support labeled classes. All results are averaged over 600 episodes. We mark the best results in bold.

5.3.4 Ablation Study

We analyze different components of PTN and summarize the results in Table 5.7. All compared approaches are based on the pre-trained WRN-28-10

Table 5.6: Distraction comparison on the *tiered*ImageNet dataset.

Methods	1-shot	5-shot	1-shot w/D	5-shot w/D
Soft K-Means (Ren et al. 2018)	51.52±0.36	70.25±0.31	49.88±0.52	68.32±0.22
Soft K-Means+Cluster (Ren et al. 2018)	51.85±0.25	69.42±0.17	51.36±0.31	67.56±0.10
Masked Soft k-Means (Ren et al. 2018)	52.39±0.44	69.88±0.20	51.38±0.38	69.08±0.25
TPN-semi (Liu et al. 2018)	55.74±0.29	71.01±0.23	53.45±0.93	69.93±0.80
PTN (Ours)	**84.70±1.14**	**89.14±0.71**	**83.84±1.07**	**88.06±0.62**

* "w/D" means with distraction classification. In this setting, many extra unlabeled samples are from the distraction classes, which is different from the support labeled classes. All results are averaged over 600 episodes. We mark the best results in bold.

embedding.

First of all, we investigate the graph propagation component (classifier). It can be observed that graph-based models such as Label Propagation (Zhou et al. 2004) and PoissonMBO (Calder et al. 2020) outperform the inductive model TransMatch (Yu et al. 2020), which is consistent with previous researches (Zhu et al. 2005, Liu et al. 2018, Ziko et al. 2020). Compared to directly applying PoissonMBO on few-shot tasks, the proposed DPN ***(without Unsupervised Embedding Transfer)*** achieves better performance, which indicates it is necessary to perform the feature calibration to eliminate the cross-class biases between support and query data distributions before label inference.

For investigating the proposed unsupervised embedding transfer in representation learning, we observe that all the graph-based models achieve clear improvement after incorporating the proposed transfer module. For instance, the Label Propagation obtains 1.61%, 1.86% performance gains on five-way-one-shot and five-way-five-shot minImageNet identification. These results indicate the effectiveness of the proposed unsupervised embedding transfer. Finally, by integrating the unsupervised embedding transfer and graph propagation classifier, the PTN model achieves the best performances compared against all other approaches in Table 5.7.

Methods	1-shot	5-shot
TransMatch	62.93±1.11	82.24±0.59
Label Propagation (LP)	74.04±1.00	82.60±0.68
PoissonMBO	79.67±1.02	86.30±0.65
DPN	80.00±0.83	87.17±0.51
Unsup Trans+LP [a]	75.65±1.06	84.46±0.68
Unsup Trans+PoissonMBO	80.73±1.11	87.41±0.63
Unsup Trans+PTN [b]	**82.66±0.97**	**88.43±0.76**

[a]Unsup Trans means Unsupervised Embedding Transfer.
[b]PTN consists of Unsup Trans and DPN.

Table 5.7: Ablation studies about the proposed PTN, all methods are based on a pretrained embedding with 200 extra unlabeled samples each class on miniImageNet for five-way-one-shot and five-way-five-shot classification (%). Best results are in bold.

5.3.5 Inference Time

In this subsection, we present inference time experiments to investigate the computation efficiency of PTN on the *mini*ImageneNet (Vinyals et al. 2016) dataset. Same as (Ziko et al. 2020), we compute the mean inference time for each five-shot task. The results are summarized in Table 5.8. Compared to inductive methods, the proposed PTN costs more time due to the graph-based Poisson inference. However, our model achieves better classification performance than inductive ones and other transductive models, with affordable inference time.

5.4 Summary

In chapter 5, we studied the semi-supervised few-shot RPSI Defect and generic image classification. We propose a Poisson Transfer Network (PTN) to tackle the semi-supervised few-shot problems, aiming to explore the value

Table 5.8: Mean inference time for the five-shot tasks on *mini*ImageneNet dataset.

Methods	Inference Time (s)
SimpleShot (Wang, Chao, Weinberger & van der Maaten 2019)	0.009
LaplacianShot (Ziko et al. 2020)	0.012
Transductive fine-tune (Dhillon et al. 2019)	20.7
PTN(Ours)	13.68

of unlabeled novel-class data from two aspects. We propose to employ the Poisson learning model to capture the relations from the few labeled and unlabeled data, which generates a more stable and informative classifier than previous semi-supervised few-shot models. Moreover, we propose to adopt unsupervised contrastive learning to improve the generality of the embedding on novel classes, which avoids the possible over-fitting problem when training with few labeled samples. Integrating the two modules, the proposed PTN can well explore the unlabeled auxiliary information boosting the performance of few-shot learning. Extensive experiments indicate that PTN outperforms state-of-the-art few-shot and semi-supervised few-shot methods on both RPSI Defects and two generic benchmark datasets. PTN is a three-step model for SSFSL tasks. How to improve it as an end-to-end model is an interesting problem. Moreover, how to reduce the computation complexity of Poisson inference is also a challengeable research issue.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This thesis summarizes several automatic deep-learning-based methods for Railway Power Supply Infrastructure (RPSI) defects detection. We divided the RPSI defects detection into two stages: general object localization and object defects recognition, and this thesis focus on object defects recognition. Two challenges are abstracted from RPSI defects recognition: fine-grained defects identification and few-shot model training. Moreover, we further studied three research issues: using deep fine-grained models to deal with RPSI defects recognition, fine-grained few-shot RPSI defects classification, and semi-supervised few-shot RPSI defects identification.

Chapter 3 proposed a fine-grained model named Spatial Transformer And Bilinear Low-Rank (STABLR) model and applied it to the RPSI defects recognition. To solve the high variation within the class, we adopted the Spatial Transformer Network. To achieve more effective performance, we presented a Low-Rank Bilinear model. Moreover, cooperated with Sydney Trains, we constructed a novel RPSI defects dataset. The experimental results validate that the STABLR model outperforms both hand-craft features-based machine learning methods and classic deep neural network methods. This chapter is supported by the conference publication at DICTA18 (Huang,

Xu, Zhang, Wu & Kirsch 2018).

Chapter 4 proposed an Aligned Pairwise Bilinear Framework (APBF) to deal with the fine-grained few-shot RPSI defects classification. By adopting a meta-learning training strategy, APBF can simultaneously learn to reduce the high intra-class variance and enlarge the inter-class discrimination of the fine-grained few-shot model. We then designed three models using APBF: PABN, LRPABN, and TOAN. PABN is the first work to adopt pairwise bilinear pooling for fine-grained few-shot tasks. Moreover, two simple yet effective alignment losses are presented in the PABN model. LRPABN model is an advanced PABN model with low-rank pairwise bilinear pooling and a better alignment layer. TOAN is the last APBF-based model that achieves state-of-the-art performance on both RPSI defects and generic fine-grained few-shot image identification. A novel cross-attention alignment layer and group pairwise bilinear pooling are embedded. This chapter is supported by the conference publication at ICME19 (Huang et al. 2019), the journal publications at T-MM (Huang, Zhang, Zhang, Xu & Wu 2021) and T-CSVT (Huang, Zhang, Yu, Zhang, Wu & Xu 2021).

Chapter 5 proposed a Poisson Transfer Network (PTN) to tackle the semi-supervised few-shot RPSI defects classification problem. Different from Chapter 4, PTN aims to explore the value of unlabeled testing data to boost the few-shot models. We proposed to employ the Poisson learning model to capture the relations between the few labeled and unlabeled data, which results in a more stable and informative classifier than previous semi-supervised few-shot models. Moreover, we proposed to adopt contrastive learning to improve the generality of the embedding on novel classes, which avoids the possible over-fitting problem when training with few labeled samples. Extensive experiments indicate that PTN outperforms state-of-the-art few-shot and semi-supervised few-shot methods on both RPSI defects and generic few-shot image classification. This chapter is supported by the conference publication at AAAI21 (Huang, Zhang, Zhang, Wu & Xu 2021).

6.2 Future Work

Most recently, the few-shot detection methods (Kang, Liu, Wang, Yu, Feng & Darrell 2019, Fan, Zhuo, Tang & Tai 2020) have been introduced to learn a detector with limited label video frames. However, these approaches have not been studied in automatic industrial defect detection. How to design a decent few-shot detection model for RPSI defects inspection is an open problem and worth studying. Moreover, some researchers investigated the cross-domain few-shot learning problems (Tseng, Lee, Huang & Yang 2019, Guo, Codella, Karlinsky, Codella, Smith, Saenko, Rosing & Feris 2020), which aim to transfer the knowledge from different well-labeled domains to the target few-shot domain. Cross-domain few-shot is a more realistic setting for industrial defects recognition and detection tasks since the extensive auxiliary training dataset in the industrial area is hard to obtain. Utilizing the auxiliary data with a large domain gap for the industrial data is a challenging issue. For future works, these issues are expected to be well-addressed.

Bibliography

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A. & Raffel, C. A. (2019), Mixmatch: A holistic approach to semi-supervised learning, *in* 'NeurIPS', pp. 5049–5059.

Biederman, I. (1987), 'Recognition-by-components: a theory of human image understanding.', *Psychological review* **94**(2), 115.

Brown, A. L. (1975), The development of memory: Knowing, knowing about knowing, and knowing how to know, *in* 'Advances in child development and behavior', Vol. 10, Elsevier, pp. 103–152.

Calder, J., Cook, B., Thorpe, M. & Slepcev, D. (2020), Poisson learning: Graph based semi-supervised learning at very low label rates, *in* 'ICML', PMLR, pp. 1306–1316.

Calder, J. & Slepčev, D. (2019), 'Properly-weighted graph laplacian for semi-supervised learning', *Applied Mathematics & Optimization* pp. 1–49.

Chan, C.-h. & Pang, G. K. (2000), 'Fabric defect detection by fourier analysis', *IEEE TIA* **36**(5), 1267–1276.

Chang, C.-C. & Lin, C.-J. (2011), 'Libsvm: a library for support vector machines', *ACM TIST* **2**(3), 27.

Chatfield, K., Simonyan, K., Vedaldi, A. & Zisserman, A. (2014), 'Return of the devil in the details: Delving deep into convolutional nets', *arXiv preprint arXiv:1405.3531* .

Chen, H., Wang, J., Qi, Q., Li, Y. & Sun, H. (2017), Bilinear cnn models for food recognition, *in* 'DICTA', pp. 1–6.

Chen, T., Kornblith, S., Norouzi, M. & Hinton, G. (2020), 'A simple framework for contrastive learning of visual representations', *ICLR* .

Chen, W., Liu, Y., Kira, Z., Wang, Y. F. & Huang, J. (2019), A closer look at few-shot classification, *in* 'ICLR', OpenReview.net.

Chen, W.-Y., Liu, Y.-C., Kira, Z., Wang, Y.-C. F. & Huang, J.-B. (2018), A closer look at few-shot classification, *in* 'ICLR'.

Cortes, C. & Vapnik, V. (1995), 'Support-vector networks', *Machine Learning* **20**(3), 273–297.

Cui, Y., Zhou, F., Wang, J., Liu, X., Lin, Y. & Belongie, S. (2017*a*), Kernel pooling for convolutional neural networks, *in* 'CVPR', Vol. 1, p. 7.

Cui, Y., Zhou, F., Wang, J., Liu, X., Lin, Y. & Belongie, S. J. (2017*b*), Kernel pooling for convolutional neural networks, *in* 'CVPR', IEEE Computer Society, pp. 3049–3058.

Dalal, N. & Triggs, B. (2005), Histograms of oriented gradients for human detection, *in* 'CVPR', Vol. 1, pp. 886–893.

Dhillon, G. S., Chaudhari, P., Ravichandran, A. & Soatto, S. (2019), A baseline for few-shot image classification, *in* 'ICLR'.

Faghih-Roohi, S., Hajizadeh, S., Núñez, A., Babuska, R. & De Schutter, B. (2016), Deep convolutional neural networks for detection of rail surface defects., *in* 'IJCNN', pp. 2584–2589.

Fan, Q., Zhuo, W., Tang, C.-K. & Tai, Y.-W. (2020), Few-shot object detection with attention-rpn and multi-relation detector, *in* 'CVPR', pp. 4013–4022.

Farrell, R., Oza, O., Zhang, N., Morariu, V. I., Darrell, T. & Davis, L. S. (2011), Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance, *in* 'ICCV', IEEE, pp. 161–168.

Fei-Fei, L., Fergus, Rob, P. & Pietro (2006), 'One-shot learning of object categories', *IEEE TPAMI* **28**(4), 594–611.

Finn, C., Abbeel, P. & Levine, S. (2017*a*), Model-agnostic meta-learning for fast adaptation of deep networks, *in* 'ICML', Vol. 70 of *Proceedings of Machine Learning Research*, PMLR, pp. 1126–1135.

Finn, C., Abbeel, P. & Levine, S. (2017*b*), Model-agnostic meta-learning for fast adaptation of deep networks, *in* 'ICML', pp. 1126–1135.

Fu, J., Zheng, H. & Mei, T. (2017), Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition, *in* 'CVPR', IEEE Computer Society, pp. 4476–4484.

Gal, Y. & Ghahramani, Z. (2016), Dropout as a bayesian approximation: Representing model uncertainty in deep learning, *in* 'ICML', pp. 1050–1059.

Gao, S., Tsang, I. W.-H. & Ma, Y. (2014), 'Learning category-specific dictionary and shared dictionary for fine-grained image categorization', *IEEE TIP* **23**(2), 623–634.

Gao, Y., Beijbom, O., Zhang, N. & Darrell, T. (2016), Compact bilinear pooling, *in* 'CVPR', IEEE Computer Society, pp. 317–326.

Gao, Z., Wu, Y., Zhang, X., Dai, J., Jia, Y. & Harandi, M. (2020), Revisiting bilinear pooling: A coding perspective, *in* 'AAAI', AAAI Press, pp. 3954–3961.

Garcia-Cardona, C., Merkurjev, E., Bertozzi, A. L., Flenner, A. & Percus, A. G. (2014), 'Multiclass data segmentation using diffuse interface methods on graphs', *IEEE TPAMI* **36**(8), 1600–1613.

Gidaris, S., Bursuc, A., Komodakis, N., Pérez, P. & Cord, M. (2019), Boosting few-shot visual learning with self-supervision, *in* 'ICCV', IEEE, pp. 8058–8067.

Gidaris, S. & Komodakis, N. (2018), Dynamic few-shot visual learning without forgetting, *in* 'CVPR', IEEE Computer Society, pp. 4367–4375.

Girshick, R., Donahue, J., Darrell, T. & Malik, J. (2014), Rich feature hierarchies for accurate object detection and semantic segmentation, *in* 'CVPR'.

Gu, B., Zhai, Z., Deng, C. & Huang, H. (2020), 'Efficient active learning by querying discriminative and representative samples and fully exploiting unlabeled data', *IEEE TNNLS* **32**(9), 4111–4122.

Guo, Y., Codella, N. C., Karlinsky, L., Codella, J. V., Smith, J. R., Saenko, K., Rosing, T. & Feris, R. (2020), A broader study of cross-domain few-shot learning, *in* 'ECCV', Springer, pp. 124–141.

Haney, B. & Lavin, A. (2020), 'Fine-grain few-shot vision via domain knowledge as hyperspherical priors', *CoRR* **abs/2005.11450**.

Hao, F., He, F., Cheng, J., Wang, L., Cao, J. & Tao, D. (2019), Collect and select: Semantic alignment metric learning for few-shot learning, *in* 'ICCV', IEEE, pp. 8459–8468.

He, K., Fan, H., Wu, Y., Xie, S. & Girshick, R. (2020), Momentum contrast for unsupervised visual representation learning, *in* 'CVPR', pp. 9729–9738.

He, K., Gkioxari, G., Dollár, P. & Girshick, R. (2017), Mask r-cnn, *in* 'ICCV', pp. 2961–2969.

He, K., Zhang, X., Ren, S. & Sun, J. (2016), Deep residual learning for image recognition, *in* 'CVPR', pp. 770–778.

He, X. & Peng, Y. (2018), Only learn one sample: Fine-grained visual categorization with one sample training, *in* 'ACM MM', ACM, pp. 1372–1380.

Hoffman, D. D. & Richards, W. A. (1984), 'Parts of recognition', *Cognition* **18**(1-3), 65–96.

Horn, G. V., Branson, S., Farrell, R., Haber, S., Barry, J., Ipeirotis, P., Perona, P. & Belongie, S. J. (2015), Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection, *in* 'CVPR', IEEE Computer Society, pp. 595–604.

Hou, R., Chang, H., Ma, B., Shan, S. & Chen, X. (2019), Cross attention network for few-shot classification, *in* 'NeurIPS', pp. 4005–4016.

Hu, P., Sun, X., Saenko, K. & Sclaroff, S. (2019), 'Weakly-supervised compositional feature aggregation for few-shot recognition', *arXiv preprint arXiv:1906.04833* .

Huang, C., Li, H., Xie, Y., Wu, Q. & Luo, B. (2016), 'Pbc: Polygon-based classifier for fine-grained categorization', *IEEE TMM* **19**(4), 673–684.

Huang, H., Hu, C., Wang, T., Zhang, L., Li, F. & Guo, P. (2017), Surface defects detection for mobilephone panel workpieces based on machine vision and machine learning, *in* 'ICIA', pp. 370–375.

Huang, H., Xu, J., Zhang, J., Wu, Q. & Kirsch, C. (2018), Railway infrastructure defects recognition using fine-grained deep convolutional neural networks, *in* 'DICTA', IEEE, pp. 1–8.

Huang, H., Zhang, C., Hu, Q. & Zhu, P. (2016), Multi-view representative and informative induced active learning, *in* 'PRICAI', Springer, pp. 139–151.

Huang, H., Zhang, J., Yu, L., Zhang, J., Wu, Q. & Xu, C. (2021), 'Toan: Target-oriented alignment network for fine-grained image categorization with few labeled samples', *IEEE TCSVT* pp. 1–1.

Huang, H., Zhang, J., Zhang, J., Wu, Q. & Xu, C. (2021), Ptn: A poisson transfer network for semi-supervised few-shot learning, *in* 'AAAI', Vol. 35, pp. 1602–1609.

Huang, H., Zhang, J., Zhang, J., Wu, Q. & Xu, J. (2019), Compare more nuanced: Pairwise alignment bilinear network for few-shot fine-grained learning, *in* 'ICME', IEEE, pp. 91–96.

Huang, H., Zhang, J., Zhang, J., Xu, J. & Wu, Q. (2021), 'Low-rank pairwise alignment bilinear network for few-shot fine-grained image classification', *IEEE TMM* **23**, 1666–1680.

Huang, Y., Wu, Q., Xu, J., Zhong, Y. & Zhang, Z. (2021), Clothing status awareness for long-term person re-identification, *in* 'ICCV', pp. 11895–11904.

Iscen, A., Tolias, G., Gosselin, P.-H. & Jégou, H. (2015), 'A comparison of dense region detectors for image search and fine-grained classification', *IEEE TIP* **24**(8), 2369–2381.

Jaderberg, M., Simonyan, K., Zisserman, A. et al. (2015), Spatial transformer networks, *in* 'NeurIPS', pp. 2017–2025.

Jia, F., Lei, Y., Lin, J., Zhou, X. & Lu, N. (2016), 'Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data', *Mechanical Systems and Signal Processing* **72**, 303–315.

Jia, H., Murphey, Y. L., Shi, J. & Chang, T.-S. (2004), An intelligent real-time vision system for surface defect detection, *in* 'ICPR', Vol. 3, pp. 239–242.

Jiang, W., Huang, K., Geng, J. & Deng, X. (2020), 'Multi-scale metric learning for few-shot learning', *IEEE TCSVT* pp. 1–1.

John, D. R. & Cole, C. A. (1986), 'Age differences in information processing: Understanding deficits in young and elderly consumers', *Journal of consumer research* **13**(3), 297–315.

Jolliffe, I. (2011), Principal component analysis, *in* 'International Encyclopedia of Statistical Science', pp. 1094–1096.

Kang, B., Liu, Z., Wang, X., Yu, F., Feng, J. & Darrell, T. (2019), Few-shot object detection via feature reweighting, *in* 'CVPR', pp. 8420–8429.

Khosla, A., Jayadevaprakash, N., Yao, B. & Li, F.-F. (2011), Novel dataset for fine-grained image categorization: Stanford dogs, *in* 'CVPR Workshop', Vol. 2.

Kim, J., Kim, T., Kim, S. & Yoo, C. D. (2019), Edge-labeling graph neural network for few-shot learning, *in* 'CVPR', pp. 11–20.

Kim, J., On, K. W., Lim, W., Kim, J., Ha, J. & Zhang, B. (2017*a*), Hadamard product for low-rank bilinear pooling, *in* 'ICLR', OpenReview.net.
URL: *https://openreview.net/forum?id=r1rhWnZkg*

Kim, J., On, K. W., Lim, W., Kim, J., Ha, J. & Zhang, B. (2017*b*), Hadamard product for low-rank bilinear pooling, *in* 'ICLR', OpenReview.net.

Kong, S. & Fowlkes, C. C. (2017), Low-rank bilinear pooling for fine-grained classification, *in* 'CVPR', IEEE Computer Society, pp. 7025–7034.

Koniusz, P. & Zhang, H. (2021), 'Power normalizations in fine-grained image, few-shot image and graph classification', *IEEE TPAMI* .

Krause, J., Jin, H., Yang, J. & Li, F. (2015), Fine-grained recognition without part annotations, *in* 'CVPR', IEEE Computer Society, pp. 5546–5555.

Krause, J., Stark, M., Deng, J. & Fei-Fei, L. (2013), 3d object representations for fine-grained categorization, *in* 'ICCV', IEEE Computer Society, pp. 554–561.

Krizhevsky, A. (2014), 'One weird trick for parallelizing convolutional neural networks', *arXiv preprint arXiv:1404.5997* .

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012), Imagenet classification with deep convolutional neural networks, *in* 'NeurIPS', pp. 1106–1114.

Kumar, A. & Pang, G. K. (2002), 'Defect detection in textured materials using gabor filters', *IEEE TIA* **38**(2), 425–440.

Lan, X., Yuen, P. C. & Chellappa, R. (2017), Robust mil-based feature template learning for object tracking, *in* 'AAAI', pp. 4118–4125.

Lazarou, M., Stathaki, T. & Avrithis, Y. (2021), Iterative label cleaning for transductive and semi-supervised few-shot learning, *in* 'ICCV', pp. 8751–8760.

Lee, K., Maji, S., Ravichandran, A. & Soatto, S. (2019), Meta-learning with differentiable convex optimization, *in* 'CVPR', pp. 10657–10665.

Li, B., Zhang, X., Liang, C. & Wei, T. (2019), Deep learning: Excellent method at surface defect detection of industrial products, *in* 'IMCEC', IEEE, pp. 712–716.

Li, P., Xie, J., Wang, Q. & Gao, Z. (2018), Towards faster training of global covariance pooling networks by iterative matrix square root normalization, *in* 'CVPR', IEEE Computer Society, pp. 947–955.

Li, W., Wang, L., Xu, J., Huo, J., Gao, Y. & Luo, J. (2019), Revisiting local descriptor based image-to-class measure for few-shot learning, *in* 'CVPR', Computer Vision Foundation / IEEE, pp. 7260–7268.

Li, W., Xu, J., Huo, J., Wang, L., Gao, Y. & Luo, J. (2019), Distribution consistency based covariance metric networks for few-shot learning, *in* 'AAAI', AAAI Press, pp. 8642–8649.

Li, X., Sun, Q., Liu, Y., Zhou, Q., Zheng, S., Chua, T.-S. & Schiele, B. (2019), Learning to self-train for semi-supervised few-shot classification, *in* 'NeurIPS', pp. 10276–10286.

Li, Z., Liu, F., Yang, W., Peng, S. & Zhou, J. (2021), 'A survey of convolutional neural networks: analysis, applications, and prospects', *IEEE TNNLS* .

Lin, D., Shen, X., Lu, C. & Jia, J. (2015*a*), Deep lac: Deep localization, alignment and classification for fine-grained recognition, *in* 'CVPR', pp. 1666–1674.

Lin, D., Shen, X., Lu, C. & Jia, J. (2015*b*), Deep lac: Deep localization, alignment and classification for fine-grained recognition, *in* 'CVPR', pp. 1666–1674.

Lin, T., RoyChowdhury, A. & Maji, S. (2015*a*), Bilinear CNN models for fine-grained visual recognition, *in* 'ICCV', IEEE Computer Society, pp. 1449–1457.

Lin, T.-Y. & Maji, S. (2017), 'Improved bilinear pooling with cnns', *arXiv preprint arXiv:1707.06772* .

Lin, T.-Y., RoyChowdhury, A. & Maji, S. (2015*b*), Bilinear cnn models for fine-grained visual recognition, *in* 'ICCV', pp. 1449–1457.

Lin, T.-Y., RoyChowdhury, A. & Maji, S. (2018), 'Bilinear convolutional neural networks for fine-grained visual recognition', *IEEE TPAMI* **40**(6), 1309–1322.

Liu, G. & Li, F. (2021), 'Fabric defect detection based on low-rank decomposition with structural constraints', *The Visual Computer* pp. 1–15.

Liu, J., Song, L. & Qin, Y. (2020), Prototype rectification for few-shot learning, *in* 'ECCV', pp. 741–756.

Liu, Y., Lee, J., Park, M., Kim, S., Yang, E., Hwang, S. J. & Yang, Y. (2018), Learning to propagate labels: Transductive propagation network for few-shot learning, *in* 'ICLR'.

Liu, Y., Lee, J., Park, M., Kim, S., Yang, E., Hwang, S. J. & Yang, Y. (2019*a*), Learning to propagate labels: Transductive propagation network for few-shot learning, *in* 'ICLR', OpenReview.net.

Liu, Y., Lee, J., Park, M., Kim, S., Yang, E., Hwang, S. & Yang, Y. (2019*b*), Learning to propagate labels: Transductive propagation network for few-shot learning, *in* 'ICLR'.

Lowe, M. J., Alleyne, D. N. & Cawley, P. (1998), 'Defect detection in pipes using guided waves', *Ultrasonics* **36**(1-5), 147–154.

Lu, C., Wang, Z.-Y., Qin, W.-L. & Ma, J. (2017), 'Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification', *Signal Processing* **130**, 377–388.

Maaten, L. v. d. & Hinton, G. (2008), 'Visualizing data using t-sne', *JMLR* **9**(Nov), 2579–2605.

Marani, R., Palumbo, D., Galietti, U., Stella, E. & D'Orazio, T. (2016), Automatic detection of subsurface defects in composite materials using thermography and unsupervised machine learning, *in* 'IEEE ICIS', pp. 516–521.

Marr, D. & Nishihara, H. K. (1978), 'Representation and recognition of the spatial organization of three-dimensional shapes', *Proceedings of the Royal Society of London. Series B. Biological Sciences* **200**(1140), 269–294.

McConnell, R. K. (1986), Method of and apparatus for pattern recognition, Technical report.

Mettes, P., van der Pol, E. & Snoek, C. (2019), Hyperspherical prototype networks, *in* 'NeurIPS', pp. 1485–1495.

Miller, E. G., Matsakis, N. E. & Viola, P. A. (2000), Learning from one example through shared densities on transforms, *in* 'CVPR', Vol. 1, pp. 464–471.

Mishra, N., Rohaninejad, M., Chen, X. & Abbeel, P. (2018), A simple neural attentive meta-learner, *in* 'ICLR'.

Munkhdalai, T. & Yu, H. (2017), Meta networks, *in* 'ICML', Vol. 70 of *Proceedings of Machine Learning Research*, PMLR, pp. 2554–2563.

Neyman, J. (1937), 'Outline of a theory of statistical estimation based on the classical theory of probability', *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences* **236**(767), 333–380.

Ng, H.-F. (2006), 'Automatic thresholding for defect detection', *Pattern Recognition Letters* **27**(14), 1644–1649.

Norvig, P. R. & Intelligence, S. A. (2002), *A modern approach*, Prentice Hall Upper Saddle River, NJ, USA:.

Oquab, M., Bottou, L., Laptev, I. & Sivic, J. (2015), Is object localization for free?-weakly-supervised learning with convolutional neural networks, *in* 'CVPR', pp. 685–694.

Oreshkin, B., López, P. R. & Lacoste, A. (2018), Tadam: Task dependent adaptive metric for improved few-shot learning, *in* 'NeurIPS', pp. 721–731.

Otsu, N. (1979), 'A threshold selection method from gray-level histograms', *IEEE TSMC* **9**(1), 62–66.

Pahde, F., Nabi, M., Klein, T. & Jähnichen, P. (2018), Discriminative hallucination for multi-modal few-shot learning, *in* 'ICIP', IEEE, pp. 156–160.

Peng, Y., He, X. & Zhao, J. (2018*a*), 'Object-part attention model for fine-grained image classification', *IEEE TIP* **27**(3), 1487–1500.

Peng, Y., He, X. & Zhao, J. (2018*b*), 'Object-part attention model for fine-grained image classification', *IEEE TIP* **27**(3), 1487–1500.

Peyre, J., Sivic, J., Laptev, I. & Schmid, C. (2017), Weakly-supervised learning of visual relations, *in* 'ICCV', pp. 5179–5188.

Pham, N. & Pagh, R. (2013), Fast and scalable polynomial kernels via explicit feature maps, *in* 'ACM SIGKDD', ACM, pp. 239–247.

Powers, D. M. (2011), 'Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation'.

Qi, C. R., Su, H., Mo, K. & Guibas, L. J. (2017), Pointnet: Deep learning on point sets for 3d classification and segmentation, *in* 'CVPR'.

Qi, H., Brown, M. & Lowe, D. G. (2018), Low-shot learning with imprinted weights, *in* 'CVPR', pp. 5822–5830.

Qiao, S., Liu, C., Shen, W. & Yuille, A. L. (2018), Few-shot image recognition by predicting parameters from activations, *in* 'CVPR', pp. 7229–7238.

Rajeswaran, A., Finn, C., Kakade, S. M. & Levine, S. (2019), Meta-learning with implicit gradients, *in* 'NeurIPS', pp. 113–124.

Ravi, S. & Larochelle, H. (2017), Optimization as a model for few-shot learning, *in* 'ICLR', OpenReview.net.

Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. (2016), You only look once: Unified, real-time object detection, *in* 'CVPR', pp. 779–788.

Ren, M., Triantafillou, E., Ravi, S., Snell, J., Swersky, K., Tenenbaum, J. B., Larochelle, H. & Zemel, R. S. (2018), Meta-learning for semi-supervised few-shot classification, *in* 'ICLR'.

Ren, S., He, K., Girshick, R. & Sun, J. (2015), Faster r-cnn: Towards real-time object detection with region proposal networks, *in* 'NeurIPS', pp. 91–99.

Riesenhuber, M. & Poggio, T. (1999), 'Hierarchical models of object recognition in cortex', *Nature neuroscience* **2**(11), 1019.

Ruan, X., Lin, G., Long, C. & Lu, S. (2021), 'Few-shot fine-grained classification with spatial attentive comparison', *Knowledge-Based Systems* **218**, 106840.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C. & Fei-Fei, L. (2015), 'ImageNet Large Scale Visual Recognition Challenge', *IJCV* **115**(3), 211–252.

Rusu, A. A., Rao, D., Sygnowski, J., Vinyals, O., Pascanu, R., Osindero, S. & Hadsell, R. (2018), Meta-learning with latent embedding optimization, *in* 'ICLR'.

Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D. & Lillicrap, T. P. (2016), Meta-learning with memory-augmented neural networks, *in* 'ICML', Vol. 48 of *JMLR Workshop and Conference Proceedings*, JMLR.org, pp. 1842–1850.

Simon, C., Koniusz, P. & Harandi, M. (2022), Meta-learning for multi-label few-shot classification, *in* 'WACV', pp. 3951–3960.

Simon, M. & Rodner, E. (2015), Neural activation constellations: Unsupervised part model discovery with convolutional networks, *in* 'ICCV', IEEE Computer Society, pp. 1143–1151.

Simonyan, K. & Zisserman, A. (2014), 'Very deep convolutional networks for large-scale image recognition', *arXiv preprint arXiv:1409.1556* .

Snell, J., Swersky, K. & Zemel, R. S. (2017), Prototypical networks for few-shot learning, *in* 'NeurIPS', pp. 4077–4087.

Suh, Y., Wang, J., Tang, S., Mei, T. & Mu Lee, K. (2018), Part-aligned bilinear representations for person re-identification, *in* 'ECCV'.

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H. & Hospedales, T. M. (2018*a*), Learning to compare: Relation network for few-shot learning, *in* 'CVPR', pp. 1199–1208.

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H. S. & Hospedales, T. M. (2018*b*), Learning to compare: Relation network for few-shot learning, *in* 'CVPR', IEEE Computer Society, pp. 1199–1208.

Szegedy, C., Ioffe, S., Vanhoucke, V. & Alemi, A. A. (2017), Inception-v4, inception-resnet and the impact of residual connections on learning., *in* 'AAAI', Vol. 4, p. 12.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. (2016), Rethinking the inception architecture for computer vision, *in* 'CVPR', pp. 2818–2826.

Tan, M., Yuan, F., Yu, J., Wang, G. & Gu, X. (2022), 'Fine-grained image classification via multi-scale selective hierarchical biquadratic pooling', *ACM TOMM* **18**(1s), 1–23.

Tian, Y., Wang, Y., Krishnan, D., Tenenbaum, J. B. & Isola, P. (2020), Rethinking few-shot image classification: a good embedding is all you need?, *in* 'ECCV', pp. 266–282.

Tseng, H.-Y., Lee, H.-Y., Huang, J.-B. & Yang, M.-H. (2019), Cross-domain few-shot classification via learned feature-wise transformation, *in* 'ICLR'.

Tsutsui, S., Fu, Y. & Crandall, D. J. (2019), Meta-reinforced synthetic data for one-shot fine-grained visual recognition, *in* 'NeurIPS', pp. 3057–3066.

Van Asch, V. (2013), 'Macro-and micro-averaged evaluation measures [[basic draft]]', *Belgium: CLiPS* .

Vapnik, V. (2013), *The nature of statistical learning theory*, Springer science & business media.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. & Polosukhin, I. (2017), Attention is all you need, *in* 'NeurIPS', pp. 5998–6008.

Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K. & Wierstra, D. (2016), Matching networks for one shot learning, *in* 'NeurIPS', pp. 3630–3638.

Wah, C., Branson, S., Welinder, P., Perona, P. & Belongie, S. (2011), The Caltech-UCSD Birds-200-2011 Dataset, Technical Report CNS-TR-2011-001, California Institute of Technology.

Wang, L. & He, D.-C. (1990), 'Texture classification using texture spectrum', *Pattern recognition* **23**(8), 905–910.

Wang, Y., Chao, W.-L., Weinberger, K. Q. & van der Maaten, L. (2019), 'Simpleshot: Revisiting nearest-neighbor classification for few-shot learning', *arXiv preprint arXiv:1911.04623* .

Wang, Y., Hu, Q., Zhu, P., Li, L., Lu, B., Garibaldi, J. M. & Li, X. (2019), 'Deep fuzzy tree for large-scale hierarchical visual classification', *IEEE TFS* **28**(7), 1395–1406.

Wei, X.-S., Wang, P., Liu, L., Shen, C. & Wu, J. (2019*a*), 'Piecewise classifier mappings: Learning fine-grained learners for novel categories with few examples', *IEEE TIP* **28**(12), 6116–6125.

Wei, X.-S., Wang, P., Liu, L., Shen, C. & Wu, J. (2019*b*), 'Piecewise classifier mappings: Learning fine-grained learners for novel categories with few examples', *IEEE TIP* **28**(12), 6116–6125.

Wertheimer, D. & Hariharan, B. (2019), Few-shot learning with localization in realistic settings, *in* 'CVPR', Computer Vision Foundation / IEEE, pp. 6558–6567.

Wu, Z., Li, Y., Guo, L. & Jia, K. (2019), PARN: position-aware relation networks for few-shot learning, *in* 'ICCV', IEEE, pp. 6658–6666.

Xie, L., Tian, Q., Wang, M. & Zhang, B. (2014), 'Spatial pooling of heterogeneous features for image classification', *IEEE TIP* **23**(5), 1994–2008.

Xu, J., Jagadeesh, V. & Manjunath, B. (2014), 'Multi-label learning with fused multimodal bi-relational graph', *IEEE TMM* **16**(2), 403–412.

Xu, Z., Tao, D., Huang, S. & Zhang, Y. (2017), 'Friend or foe: Fine-grained categorization with weak supervision', *IEEE TIP* **26**(1), 135–146.

Yang, L., Li, L., Zhang, Z., Zhou, X., Zhou, E. & Liu, Y. (2020), Dpgn: Distribution propagation graph network for few-shot learning, *in* 'CVPR', pp. 13390–13399.

Yang, S., Bo, L., Wang, J. & Shapiro, L. G. (2012), Unsupervised template learning for fine-grained object recognition, *in* 'NeurIPS', pp. 3122–3130.

Yao, B., Khosla, A. & Fei-Fei, L. (2011), Combining randomization and discrimination for fine-grained image categorization, *in* 'CVPR', IEEE, pp. 1577–1584.

Yao, H., Zhang, S., Zhang, Y., Li, J. & Tian, Q. (2016), 'Coarse-to-fine description for fine-grained visual categorization', *IEEE TIP* **25**(10), 4858–4872.

Yao, Y., Shen, F., Zhang, J., Liu, L., Tang, Z. & Shao, L. (2019), 'Extracting multiple visual senses for web learning', *IEEE TMM* **21**(1), 184–196.

Yu, C., Zhao, X., Zheng, Q., Zhang, P. & You, X. (2018), Hierarchical bilinear pooling for fine-grained visual recognition, *in* 'ECCV', Vol. 11220 of *Lecture Notes in Computer Science*, Springer, pp. 595–610.

Yu, Z., Chen, L., Cheng, Z. & Luo, J. (2020), Transmatch: A transfer-learning scheme for semi-supervised few-shot learning, *in* 'CVPR', pp. 12856–12864.

Zagoruyko, S. & Komodakis, N. (2016), Wide residual networks, *in* 'BMVC'.

Zhang, C., Li, C. & Cheng, J. (2019), 'Few-shot visual classification using image pairs with binary transformation', *IEEE TCSVT* pp. 1–1.

Zhang, D., Han, J., Guo, G. & Zhao, L. (2018), 'Learning object detectors with semi-annotated weak labels', *IEEE TCSVT* **29**(12), 3622–3635.

Zhang, D., Han, J., Yang, L. & Xu, D. (2018), 'Spftn: a joint learning framework for localizing and segmenting objects in weakly labeled videos', *IEEE TPAMI* **42**(2), 475–489.

Zhang, D., Han, J., Zhao, L. & Zhao, T. (2020), 'From discriminant to complete: Reinforcement searching-agent learning for weakly supervised object detection', *IEEE TNNLS* **31**(12), 5549–5560.

Zhang, H. & Koniusz, P. (2019), Power normalizing second-order similarity network for few-shot learning, *in* 'WACV', IEEE, pp. 1185–1193.

Zhang, H., Li, H. & Koniusz, P. (2022), 'Multi-level second-order few-shot learning', *IEEE TMM* .

Zhang, J., Wu, Q., Shen, C., Zhang, J. & Lu, J. (2018*a*), 'Mind your neighbours: Image annotation with metadata neighbourhood graph co-attention networks', *CVPR* .

Zhang, J., Wu, Q., Shen, C., Zhang, J. & Lu, J. (2018*b*), 'Multilabel image classification with regional latent semantic dependencies', *IEEE TMM* **20**(10), 2801–2813.

Zhang, L., Yang, Y., Wang, M., Hong, R., Nie, L. & Li, X. (2016), 'Detecting densely distributed graph patterns for fine-grained image categorization', *IEEE TIP* **25**(2), 553–565.

Zhang, N., Donahue, J., Girshick, R. & Darrell, T. (2014), Part-based r-cnns for fine-grained category detection, *in* 'ECCV', pp. 834–849.

Zhang, T., Qi, G., Xiao, B. & Wang, J. (2017), Interleaved group convolutions, *in* 'ICCV', IEEE Computer Society, pp. 4383–4392.

Zhang, X., Wang, J., Wang, T., Jiang, R., Xu, J. & Zhao, L. (2021), 'Robust feature learning for adversarial defense via hierarchical feature alignment', *Information Sciences* **560**, 256–270.

Zhang, X., Wei, Y., Feng, J., Yang, Y. & Huang, T. (2018), Adversarial complementary learning for weakly supervised object localization, *in* 'CVPR', pp. 1325–1334.

Zhang, X., Xiong, H., Zhou, W., Lin, W. & Tian, Q. (2016), Picking deep filter responses for fine-grained image recognition, *in* 'CVPR', IEEE Computer Society, pp. 1134–1142.

Zhang, X., Xiong, H., Zhou, W. & Tian, Q. (2016), 'Fused one-vs-all features with semantic alignments for fine-grained visual categorization', *IEEE TIP* **25**(2), 878–892.

Zhang, Y., Tang, H. & Jia, K. (2018), Fine-grained visual categorization using meta-learning optimization with sample selection of auxiliary data, *in* 'ECCV', Vol. 11212, Springer, pp. 241–256.

Zhang, Y., Wei, X.-S., Wu, J., Cai, J., Lu, J., Nguyen, V.-A. & Do, M. N. (2016), 'Weakly supervised fine-grained categorization with part-based image representation', *IEEE TIP* **25**(4), 1713–1725.

Zhao, B., Wu, X., Feng, J., Peng, Q. & Yan, S. (2017), 'Diversified visual attention networks for fine-grained object classification', *IEEE TMM* **19**(6), 1245–1256.

Zhao, J., Qiu, Z. & Sun, S. (2022), 'Multi-view multi-label active learning with conditional bernoulli mixtures', *IJMLC* pp. 1–13.

Zhao, Y., Shi, Z., Zhang, J., Chen, D. & Gu, L. (2019), 'A novel active learning framework for classification: using weighted rank aggregation to achieve multiple query criteria', *Pattern Recognition* **93**, 581–602.

Zheng, H., Fu, J., Mei, T. & Luo, J. (2017), Learning multi-attention convolutional neural network for fine-grained image recognition, *in* 'ICCV', IEEE Computer Society, pp. 5219–5227.

Zheng, H., Fu, J., Zha, Z. & Luo, J. (2019), Learning deep bilinear transformation for fine-grained image representation, *in* 'NeurIPS', pp. 4279–4288.

Zhou, D., Bousquet, O., Lal, T. N., Weston, J. & Schölkopf, B. (2004), Learning with local and global consistency, *in* 'NeurIPS', pp. 321–328.

Zhu, X., Ghahramani, Z. & Lafferty, J. D. (2003), Semi-supervised learning using gaussian fields and harmonic functions, *in* 'ICML', pp. 912–919.

Zhu, X., Lafferty, J. & Rosenfeld, R. (2005), Semi-supervised learning with graphs, PhD thesis, Carnegie Mellon University, language technologies institute, school of computer science.

Zhu, Y., Liu, C. & Jiang, S. (2020), Multi-attention meta learning for few-shot fine-grained image recognition, *in* 'AAAI', pp. 1090–1096.

Ziko, I. M., Dolz, J., Granger, E. & Ayed, I. B. (2020), Laplacian regularized few-shot learning, *in* 'ICML', pp. 11660–11670.