UNIVERSITY OF NOTTINGHAM

SCHOOL OF COMPUTER SCIENCE

Novel Meta-Learning Approaches for Few-Shot Image Classification

Author:
Heda Song

Supervisors:
Dr. Isaac Triguero
Prof. Ender Özcan

Doctoral dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

April 8, 2022

Declaration

I confirm that this thesis dissertation presented for the degree of Doctor of Philosophy at the School of Computer Science, University of Nottingham, has

- been composed entirely by myself,
- been solely the result of my own work except stated otherwise,
- not been submitted in whole or in part for any other degree or qualification in any other academic institution.

Heda Song
April 2022

Abstract

In recent years, there has been rapid progress in computing performance and communication techniques, leading to a surging interest in artificial intelligence. Artificial intelligence aims to achieve human intelligence by making a machine think. However, current machine learning and optimisation techniques are far from fully accomplishing this, suffering from several limitations. For example, humans can learn a new concept quickly from very few examples, while artificial intelligence algorithms usually require a large number of examples to extract useful patterns. To tackle this issue, the computer science community has recently delved into the challenge of learning from very limited data, also known as few-shot learning.

Few-shot image classification is the most studied research field of few-shot learning, which attempts to learn a new visual concept from limited labelled images. The conventional deep learning techniques cannot be simply applied to solve the problem, hindered by two core issues of few-shot learning, namely lack of information and intrinsic uncertainties. Lack of information is related to the insufficient visual patterns in limited training data, and intrinsic uncertainties are reflected by unrepresentative examples and background clutters. To tackle the problems, recent approaches mostly incorporate meta-learning methods which learn the general knowledge about how to make a few-shot learning process easier and quicker from a collection of learning tasks. However, existing meta-learning approaches mostly focus on either of the two key problems of few-shot image classification. Very few existing works consider both of them at the same time. Therefore, there is a need for developing novel meta-learning approaches that take into account both problems simultaneously for few-shot image classification.

The thesis focuses on developing novel strategies of meta-learning approaches for few-shot image classification through three progressive stages, with the

goal of addressing the aforementioned two core issues concurrently from different perspectives. In the first stage, we tackle the two main problems from the viewpoint of maximising the use of limited training data. Concretely, we propose learning to aggregate embeddings based on a channel-wise attention module. In this stage, we assume the embeddings after feature extraction consist of sufficient useful features. However, a feature extraction process could also lose relevant features. Hence, in the second stage, we target making sure as many useful features as possible can be extracted during a feature extraction process. Specifically, we design a spatial attention-based adaptive pooling module, in which a learnable pooling weight generation block is trained to assign different pooling weights to the features at different spatial locations. To further improve the classification performance, in the third stage, we leverage auxiliary information, such as saliency maps which can highlight the target object in an image, to compensate for the lack of information and mitigate background clutters. A comprehensive exploration of the suitable auxiliary information and how to effectively use it is provided.

In summary, the research presented here introduces novel strategies of meta-learning approaches for few-shot image classification, addressing its two core issues from three different perspectives. The conducted works provide insights and solutions about how to effectively overcome the lack of information and intrinsic uncertainties on few-shot image classification. Our proposed methods lead to competitive results on various few-shot learning benchmarks with respect to the state-of-the-art. Besides, they contribute new meta-learning strategies that deal with the two main problems of few-shot image classification simultaneously to the few-shot learning research community.

Acknowledgements

First and foremost, I would like to express my gratitude to my supervisor, Dr. Isaac Triguero and Prof. Ender Özcan, for their consistent guidance, support, patience. I want to thank them for giving me the opportunity to do this PhD under their supervision.

Along with my supervisors, I would also like to acknowledge the support from the members in Computational Optimisation and Learning (COL) Lab. I had a great time working with them. It is my honour to be one of the members in COL Lab.

Finally, I want to thank my parents and my girlfriend for their endless love and support.

Contents

List of Figures

List of Tables

List of Acronyms

AI	**A**rtificial **I**ntelligence
ML	**M**achine **L**earning
CNNs	**C**onvolutional **N**eural **N**etwork**s**
SOD	**S**aliency **O**bject **D**etection
L2AE-D	**L**earning to **A**ggregate **E**mbeddings with Meta-level **D**ropout
Ada-P	**Ada**ptive **P**ooling
FC	**F**ully **C**onnected
MAML	**M**odel-**A**gnostic **M**eta-**L**earning
ProtoNet	**Proto**typical **Net**works
SVMs	**S**upport **V**ector **M**achine**s**
NNs	**N**eural **N**etwork**s**
DTs	**D**ecision **T**ree**s**
k-NN	**k**-**N**earest **N**eighbours
RL	**R**einforcement **L**earning

Chapter 1

Introduction

Artificial Intelligence (AI) aims to achieve human intelligence by making a machine think (Goodfellow, Bengio, and Courville, 2016). Due to the advance progress of computing performance and communication techniques, AI has drawn widespread attention and has been applied in many real-world problems, such as robotics (Thrun et al., 2006), natural language processing (Bahdanau, Cho, and Bengio, 2015), computer vision (Lake, Salakhutdinov, and Tenenbaum, 2015) or scheduling problems (Shahvari and Logendran, 2017). Machine Learning (ML), serving as one of the core research fields in AI, has been developed dramatically. However, current ML techniques suffer from a few main limitations to make AI really intelligent. For example, existing algorithms are still designed by experts that manually choose suitable algorithm components and hyper-parameters in order to achieve good performance. Besides, existing algorithms are normally tailored to specific tasks and lack the generalisation ability to perform well on an unseen task. In addition, existing ML algorithms usually require a large amount of data to extract patterns in a time-consuming manner.

To address the aforementioned main limitations, ML has been interacting with itself by adding a new layer that applies an ML algorithm to assist another on problem-solving, which is also known as meta-learning (Vilalta and Drissi, 2002; Song, Triguero, and Özcan, 2019). Meta-learning uses high-level ML algorithms to extract general knowledge about how to make an ML process easier and quicker from a range of ML tasks (Vilalta and Drissi, 2002). This is similar to human learning that accumulates experience on how to learn efficiently from diverse learning processes. It has been applied in various fields of ML, such as AutoML (Hutter, Kotthoff, and Vanschoren, 2019), recommender systems (Lu, Fang, and Shi, 2020), and few-shot learning (Finn, Abbeel, and Levine, 2017).

Meta-learning has been targeting various limitations of current ML algorithms. For example, to reduce human involvement on the design of an ML algorithm, ML methods have been used to learn the mapping between algorithm configuration and performance, so that, the best-performing algorithm configuration can be selected for an ML (Hutter, Kotthoff, and Vanschoren, 2019). To improve the generalisation ability of an ML algorithm on unseen tasks, high-level ML methods have been introduced to extract general knowledge about how to select task-specific algorithm components for an ML algorithm (Oreshkin, López, and Lacoste, 2018). The work presented in this thesis focuses on the aforementioned interactions between ML and ML.

Recently, the limitation that existing ML algorithms usually require a large amount of data has drawn widespread attention in the meta-learning research field. This is driven by the fact that humans can quickly learn a new concept from few samples while ML fails to do so. Note that this limitation is also related to generalisation ability, since an ML model trained on limited samples usually cannot generalise well to new test samples. To fill the gap, the relatively new research field of few-shot learning has emerged, targeted at learning quickly only from a limited number of labelled samples (Fei-Fei, Fergus, and Perona, 2006; Lake, Salakhutdinov, and Tenenbaum, 2015). In this scenario, standard ML algorithms cannot perform well due to the inadequate available information, so that, meta-learning is usually introduced to learn general knowledge about how to effectively learn from a limited number of training samples to generalise well on an unseen task. Few-shot learning has been studied in multiple fields, such as computer vision (Fei-Fei, Fergus, and Perona, 2006), natural language processing (Han et al., 2018) and robotics (Xie et al., 2018). The research presented in this dissertation focuses on a specific few-shot learning problem, few-shot image classification. Overall, the goal of this thesis is to develop novel strategies for meta-learning approaches on few-shot image classification.

Having defined the scope of this Ph.D, the remainder of this chapter will present the basic concepts and structure of the conducted research. The fundamentals of meta-learning and few-shot learning are given in Section 1.1. Then, the motivation for our research along with the research questions are discussed in Section 1.2. The contributions of the thesis are outlined in Section 1.3. Finally, the structure of the whole thesis is summarised in Section 1.4.

1.1 Meta-Learning for Few-Shot Image Classification

Image classification is one of the core tasks of computer vision that classifies an image into one of the target categories. It has been widely applied in many real-world problems, such as fingerprint identification (Peralta et al., 2015) or face recognition (Hu et al., 2015). Traditional methods for image classification generally include two stages, feature extraction and classification (O'Mahony et al., 2019). In the feature extraction stage, a raw image is transformed into informative low-dimensional numerical features. In the classification phase, a classifier is learned to make predictions based on the extracted features. Recently, deep learning approaches have combined the two stages into a single one via training convolutional neural networks (CNNs) (LeCun, Bengio, et al., 1995) in an end-to-end manner and achieved state-of-the-art performance. However, there still remain challenging problems, such as multi-label (Wang et al., 2016), imbalanced (Huang et al., 2016) or few-shot image classification (Fei-Fei, Fergus, and Perona, 2006) which is our research focus.

Few-shot learning is the problem of making predictions based on a limited number of training samples (Fei-Fei, Fergus, and Perona, 2006; Lake, Salakhutdinov, and Tenenbaum, 2015; Vinyals et al., 2016). The underlying idea of few-shot learning is driven by the gap between human learning and ML. Humans have the ability to quickly learn new knowledge from very few samples, while ML approaches fail to do so, often requiring large numbers of samples to extract useful patterns. Besides, in many real-world applications, such as classification of rare species or recommending items for newly registered users in a recommender system, it is impractical to collect many samples of rare species or much history data of new users, so that, an ML approach being able to learn from few samples without overfitting is needed. This also facilitates the development of few-shot learning. Another application scenario of few-shot learning is labelling tons of unlabelled data only based on limited manually labelled data. Compared to standard learning, few-shot learning only relies on limited manually labelled data to learn to make a prediction, which reduces a great deal of the labour costs. Few-shot learning has been widely studied in many research fields, including computer vision (Fei-Fei, Fergus, and Perona, 2006; Lake, Salakhutdinov, and Tenenbaum, 2015), natural language processing (Sun et al., 2019b; Han et al., 2018), audio signal processing (Wang et al., 2020), robotics (Xie et al., 2018) and medicine (Tian et al., 2020a). The research

interests of this thesis lie in few-shot image classification problems.

An example of a few-shot image classification task is shown in Figure 1.1. To learn a new visual concept from very few labelled images, conventional ML algorithms need to train repeatedly on the few available labelled samples. Since the number of available training samples is so small, normally ranging from one to ten, the few training samples usually cannot provide enough visual patterns for training a model to generalise well to test samples. To address the issue, it is natural to think about introducing data augmentation techniques (Shorten and Khoshgoftaar, 2019) to enrich the visual patterns of few training data by modifying the original images through geometric transformations or colour jitter. However, these augmentations still rely on the training samples and the expanded visual patterns would not be diverse enough from the original ones. Therefore, when the training data is very limited, the expanded visual patterns in the augmented data are still not enough to train a model to generalise well.

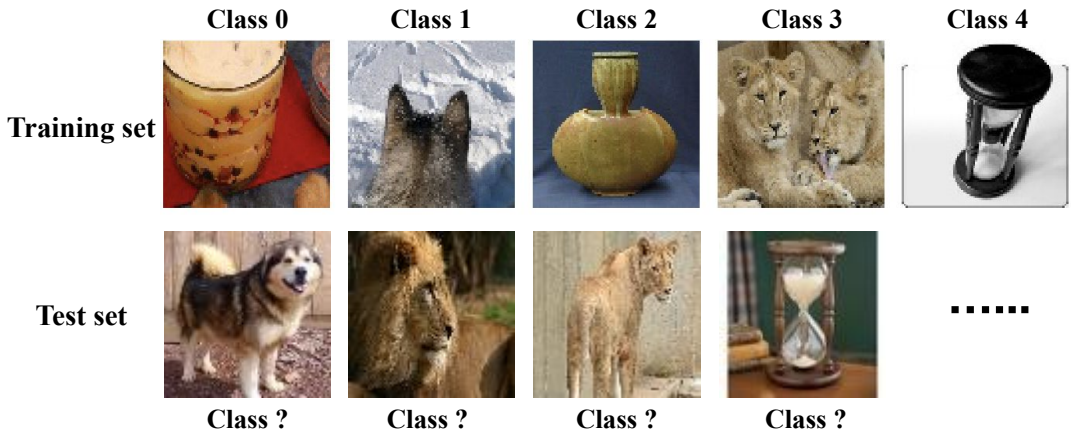


FIGURE 1.1: An example of a few-shot image classification task. There are five classes of images in the training set, namely, trifle, malamute, vase, lion and hourglass from left to right. Only one training sample is provided for each of the five classes. The target is to correctly classify each sample in the test set into its real class.

Humans can learn a new visual concept quickly from a few examples by adapting previous experiences to a new learning task. Inspired by this, most of the existing approaches tackle few-shot learning problems by incorporating pre-learned knowledge from some other learning tasks. To compensate for the lack of visual patterns in few training samples, a feature extractor is usually pre-trained on a large extra data set via multi-class classification. Then the pre-trained feature extractor having a good knowledge of extracting various

patterns can be transferred to a new few-shot learning task, in which the categories have not been seen during the pre-training process. However, when fine-tuning the pre-trained feature extractor on a new few-shot learning task, the limited training samples could still cause overfitting problems. Therefore, meta-learning is normally introduced to improve the generalisation ability. Meta-learning (Vilalta and Drissi, 2002), also known as learning to learn, is inspired by the way of human learning, in which humans can quickly adapt already known knowledge to learn a new concept. As shown in Figure 1.2, generally, meta-learning incorporates an ML algorithm in the meta-level to extract meta-knowledge from a collection of base learning tasks, and then use this meta-knowledge to assist unseen base learning tasks comprising novel categories. The meta-knowledge can be of various types related to a base learning process, such as how to select algorithms (Brazdil, Soares, and Da C., 2003), tune hyper-parameters (Lorraine and Duvenaud, 2017), train a model (Andrychowicz et al., 2016), learn an algorithm component (Sung et al., 2018) or learn to transfer knowledge (Wang, Ramanan, and Hebert, 2017).

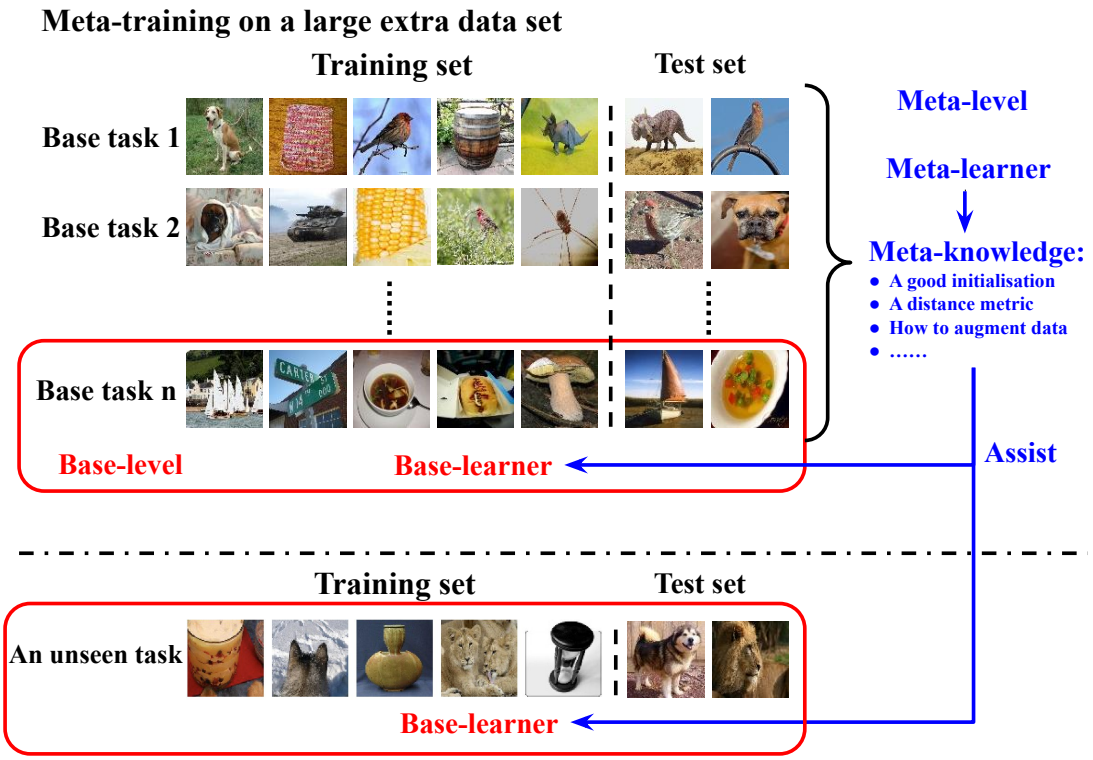


FIGURE 1.2: An illustration of meta-learning framework. During meta-training, a specific type of meta-knowledge is extracted by a meta-learner from a number of base learning tasks. During meta-testing phase, the target is to make correct predictions with the help of the meta-knowledge on an unseen task consisting of novel categories.

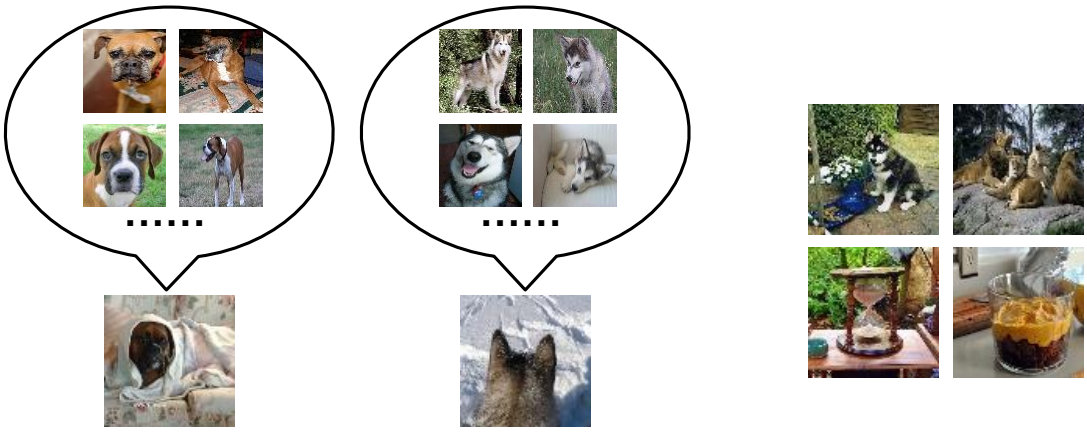
In order to improve the generalisation ability of a classification model on an unseen few-shot learning task, existing meta-learning approaches learn various types of meta-knowledge from a number of few-shot learning tasks, then transfer the meta-knowledge to the unseen few-shot task to prevent overfitting (Sung et al., 2018; Finn, Abbeel, and Levine, 2017). Generally speaking, these approaches can be classified into three main categories: (1) Fast parameterisation-based approaches learn to quickly fine-tune a learner or predict the parameters of a learner for each particular few-shot learning task (Finn, Abbeel, and Levine, 2017); (2) Obtaining more data-based approaches learn the meta-knowledge of how to generate more data using generative models or data augmentation algorithms (Wang et al., 2018b), or how to leverage auxiliary information (Xing et al., 2019); (3) Metric learning-based approaches address few-shot learning by comparing the similarities in a learned metric space (Snell, Swersky, and Zemel, 2017). Specifically, they learn a general feature extractor to transform the training and test examples into embeddings, then assigned a test embedding to its nearest training class based on a distance metric.

1.2 Motivation and Objectives

Few-shot learning is now a classic problem (Fei-Fei, Fergus, and Perona, 2006). The pioneering research works address this by generative models with prior knowledges, such as the pre-trained ML models on some extra categories (Fei-Fei, Fergus, and Perona, 2006) or the strokes of hand-written characters (Lake, Salakhutdinov, and Tenenbaum, 2015). With the success of deep learning techniques, there is a surge of interest in utilising deep learning models to tackle few-shot image classification via a meta-learning approach. However, more and more recent approaches focus more on designing complex meta-learning approaches to improve performance on benchmarks and pay less attention to analyse the essential issues and challenges of few-shot image classification (Chen et al., 2018). In this section, we explain the few-shot image classification problem and existing approaches that confront it from the perspectives of its core issues.

In a specific few-shot image classification task, a classification model could easily fall into overfitting (Snell, Swersky, and Zemel, 2017), because there are only a limited number of training samples available for each category. Therefore, the lack of information about the categories in a few-shot learning task is a core issue for few-shot image classification. In addition, since the few

training samples are collected from a specific distribution of each category in a few-shot learning task, there is a certain probability that some training samples cannot properly represent the large population of its category. For example, as shown in Figure 1.3 (a), a boxer dog is wrapped by a towel and a malamute dog only shows its back. They cannot represent a larger population of their respective categories. Besides, there are sometimes some background clutters in an image (Yan, Zhang, He, et al., 2019) as exhibited in Figure 1.3 (b). These interferences could deteriorate the performance on a few-shot learning task. Therefore, the intrinsic uncertainties reflected by unrepresentative training samples or background clutters in an image belong to another core issue of few-shot image classification. The aforementioned issues may not hinder a good performance on standard learning tasks, because plenty of training data provides adequate information and normally follows a stable data distribution, so that, a well-trained robust model can be obtained. However, when training data is insufficient, lack of information could hinder an ML algorithm to extract adequate patterns and the negative influence of uncertainties will be amplified. Therefore, an approach to tackle few-shot image classification should be able to overcome the lack of information and suppress the influence of intrinsic uncertainties.



(a) unrepresentative samples (b) background clutters

FIGURE 1.3: Examples of two types of intrinsic uncertainties.

Existing methods mostly tackle few-shot image classification from either of the aforementioned core issues. Very few of them take into account both of the core issues at the same time. Since limited training samples could lead to overfitting problems, to overcome a lack of information on a specific task, existing approaches mostly work towards designing meta-learning strategies to prevent overfitting. Some of them enlarge the small training set of a few-shot learning

task by introducing additional data from a data generation approach (Wang et al., 2018b) or a data augmentation method (Chen et al., 2019a). Note that these data generation and augmentation methods do not only rely on the few training samples in a few-shot task, but also leverage data from a large extra data set to make the generated samples more diverse from the original ones. Besides, some other approaches introduce auxiliary information related to a few-shot task, such as semantic features of the categories in a task (Xing et al., 2019), to assist image data in a few-shot learning task. Some other works concentrate on increasing the generalisation ability by learning to design adaptive or general algorithm components, such as task-specific margin loss (Li et al., 2020a) or general initial model parameters for fast adaptation (Finn, Abbeel, and Levine, 2017).

To tackle the issue of intrinsic uncertainties, some existing approaches employ an attention mechanism, which is a technique to tell a machine learner where to focus, to weaken the effect of the unrepresentative samples (Yan, Zhang, He, et al., 2019). Some other methods introduce auxiliary information to help mitigate background clutters (Zhang, Zhang, and Koniusz, 2019). To the best of our knowledge, there are only two prior works considering both of the core issues together (Zhang, Zhang, and Koniusz, 2019; Wertheimer and Hariharan, 2019). However, they only address the issues by introducing auxiliary information to mitigate background clutters. We argue that the two core issues can be tackled from several different perspectives, namely maximising the use of training samples, feature extraction and auxiliary information.

- From the point of view of maximising the use of training samples, an ideal approach for few-shot image classification should be able to reduce the negative influence of unrepresentative samples, at the same time, leverage as much as useful information of limited training samples. When tackling the unrepresentative samples, existing methods generally use an attention mechanism to assign lower weights to them. However, an unrepresentative sample may actually include some useful features, which could help strengthen a part of the class representation. For example, as shown in Figure 1.3 (a), although the unrepresentative boxer dog's body is hidden in a towel, its face is still clearly displayed, which could provide useful information to learn a robust class representation. In this thesis, we hypothesis that it would be better to perform attention in the feature level, so that, more useful information could be preserved. Besides, since samples of different classes may share some similar features,

to maximise the utility of available information in the training samples, it deserves to explore leveraging training information from other classes to strengthen a class representation in multi-class few-shot image classification.

- From the perspective of feature extraction, existing methods mostly tend to ignore that the widely used CNNs may lose useful information during feature extraction on few-shot image classification problems. For example, the commonly used max-pooling and average-pooling operations in CNNs are essentially lossy processes. When labelled data is limited, training samples may not be representative. The trained convolutional block may extract irrelevant features, and therefore the subsequent pooling operation may lose relevant features (max-pooling) or mix up relevant and irrelevant features (average-pooling), which would affect classification performance. In addition, the commonly used pooling techniques perform pooling independently in different channels and ignore the correlation between the features at the same spatial location in different channels, which may lose the information of spatial importance. We hypothesis that it would be better to consider spatial importance during a pooling operation, so that, the features in background clutters would be excluded. Therefore, the CNNs based feature extraction process needs to be promoted to avoid losing useful information and take into account spatial importance to suppress background clutters during feature extraction on few-shot image classification.

- From the viewpoint of auxiliary information, existing approaches usually introduce additional knowledge, such as semantic data (Xing et al., 2019) or manually assigned attributes (Tokmakov, Wang, and Hebert, 2019), to compensate for the lack of information. As far as we know, two previous works attempt to deal with the two key problems of few-shot image classification simultaneously (Zhang, Zhang, and Koniusz, 2019; Wertheimer and Hariharan, 2019). They first introduce auxiliary information, such as annotated bounding boxes (Wertheimer and Hariharan, 2019) or a pretrained Saliency Object Detection (SOD) model (Zhang, Zhang, and Koniusz, 2019), to compensate lack of information, and then use the introduced information to assist a feature extractor to distinguish the target object and background clutter in an image. They both propose a specific strategy to leverage such information. However, a comprehensive exploration of what is suitable auxiliary information and how to effectively

leverage it to mitigate background clutters in few-shot image classification remains unexplored to date. The research presented here addresses this gap.

The above raised research gaps motivate us to develop novel strategies for meta-learning approaches that reduce uncertainties and mitigate the lack of information in a task to improve the performance of few-shot image classification. Based on this, three research questions are presented as follows:

- Are limited training samples, especially unrepresentative samples, sufficiently exploited in few-shot image classification by existing attention mechanism-based approaches?
- Are commonly used pooling operations, such as max-pooling and average-pooling, suitable for few-shot image classification? Is it possible to design a downsampling strategy to avoid losing useful information and meanwhile mitigate background clutters during feature extraction?
- Is it possible to provide a comprehensive exploration on leveraging auxiliary information from computer vision tasks to help a feature extractor focus more on the target object for few-shot image classification?

Based on the motivations and research questions, we summarise the main aim of this thesis as: **To develop novel strategies for meta-learning approaches that can mitigate the intrinsic uncertainties and overcome the lack of information for few-shot image classification.** To achieve this goal, we derive three main research objectives, namely:

- **Objective 1:** To improve few-shot image classification by designing novel strategies that are able to reduce the negative influence of unrepresentative samples and meanwhile leverage as much useful information of training samples as possible.
- **Objective 2:** To improve the lossy pooling operation in few-shot image classification by developing a new downsampling strategy to avoid losing useful information and diminish the impact of background clutters during feature extraction.
- **Objective 3:** To provide a thorough exploration of suitable auxiliary information for compensating the lack of information and beneficial ways to leverage it to lessen the harmful effect of background clutters for few-shot image classification.

1.3 Contributions

The work presented in this dissertation focuses on an exploration of novel strategies of meta-learning approaches for few-shot image classification, which has produced one of the largest and most consolidate bodies of research in few-shot learning. After completing a whole review of the state-of-the-art approaches for few-shot image classification problems, three research stages are completed following a progressive order. The main contributions of this thesis are:

- A novel meta-learning approach that learns to aggregate embedddings (L2AE) for few-shot image classification. The method takes full advantage of the few training samples in a few-shot image classification task. This is accomplished by learning to aggregate useful convolutional features and suppress noisy ones based on a channel-wise attention mechanism, which demonstrates competitive results compared to the state-of-the-art.
- A new adaptive pooling (Ada-P) method for feature extraction on few-shot image classification. The specifically designed pooling operation learns to assign adaptive pooling weights to each embedding, which can avoid discarding useful information or pay more attention to the salient regions in different convolutional layers. This module is lightweight and can be used as a plug-and-play tool to assist future research works on few-shot image classification. The experimental results on three widely used benchmarks demonstrate the effectiveness and superiority of our method.
- A comprehensive exploration of applying diverse methods to leverage various types of auxiliary information on few-shot image classification. Based on this, we identify the most suitable auxiliary information and the appropriate ways to leverage it to assist few-shot image classification, which is named as the saliency object detection (SOD) module. The thorough study could provide guidelines on how to mitigate background clutters for better performance, which has been empirically demonstrated by the improvements made by incorporating the identified most beneficial auxiliary information into several existing methods for few-shot image classification.

The aforementioned contributions are part of or included in the following list of works completed during the Ph.D studies with publicly available code[1]:

- Heda Song, Isaac Triguero, Ender Özcan, A review on the self and dual interactions between ML and optimisation, *Progress in Artificial Intelligence* 8, pp. 143–165, 2019.
  The content of this paper is covered in Chapter 2.
- Heda Song, Mercedes Torres Torres, Ender Özcan, Isaac Triguero, L2AE-D: Learning to Aggregate Embeddings for Few-shot Learning with Meta-level Dropout, *Neurocomputing* 442, pp. 200–208, 2021.
  The content of this paper is covered in Chapter 2 and Chapter 3.
- Heda Song, Bowen Deng, Michael Pound, Ender Özcan, Isaac Triguero, A Fusion Spatial Attention Approach for Few-shot Learning, *Information Fusion* 81, pp. 187–202, 2022.
  The content of this paper is covered in Chapter 2 and Chapter 4 and Chapter 5.

1.4 Structure of the Thesis

The structure of this thesis is outlined as follows:

- **Chapter 1** has presented a broad introduction of the main topic of the research work presented in this thesis. Besides, the motivations and several research objectives of our research works have been provided. In the end, the contributions of the conducted works have been summarised.
- **Chapter 2** has provided the necessary background of the research works conducted across the different stages of this thesis. Concretely, a global overview of the interactions between ML and optimisation, and a comprehensive review on meta-learning approaches for few-shot image classification have been presented. Some fundamental knowledge about ML, optimisation and computer vision have also been introduced in this chapter.
- **Chapter 3** has described the proposed method for maximising the usage of the limited training samples in a few-shot image classification task. A channel-wise attention mechanism has been learned to assign larger weights to useful feature maps and smaller weights to noisy ones.

[1] https://github.com/Heda-Song/PhD-code

- **Chapter 4** has introduced the developed spatial attention based adaptive pooling module for few-shot image classification, in which a learnable pooling weight generation block was trained to assign different pooling weights to the features at different spatial locations for each individual embedding.
- **Chapter 5** has provided a thorough exploration of leveraging different types of auxiliary information to assist few-shot image classification. The found most suitable auxiliary information and the best way to incorporate it into few-shot image classification have been further introduced into various few-shot learning approaches to demonstrate improvements.
- **Chapter 6** has concluded the main contributions of the conducted research works in the thesis reflecting our research objectives. The limitations have been analysed, and based on those, we have pointed out a few potential research directions for future work.

Chapter 2

Background

AI refers to a broad field of science consisting of multiple disciplines. ML is one of the growing fields of AI with an enormous number of computer science applications. The ML techniques aim to learn knowledge from data or experience. To apply them automatically and make AI really intelligent, ML techniques are frequently hybridised, interacting with itself. Such approach is also known as meta-learning or learning to learn. It has been widely applied to tackle one of the most challenging problems of AI, learning from a limited number of training samples. This thesis delves into meta-learning methods for few-shot image classification.

In this chapter, the essential background and related works are provided. First, the basic concepts about ML are presented in Section 2.1. Then, a broad overview of the studies on meta-learning are presented in Section 2.2. Afterwards, Section 2.3 places some fundamentals of computer vision, covering all the computer vision techniques used in the research of this thesis. After that, the few-shot image classification problems are formulated, the common settings related to the experiments for few-shot image classification are introduced, and an overview of existing meta-learning approaches for few-shot image classification is presented in Section 2.4. Finally, we summarise the background knowledge presented in this chapter in Section 2.5.

2.1 Machine Learning Fundamentals

This section explains the concepts of ML at an introductory level. ML is the study of computer algorithms that can improve automatically through experience and by the use of data. Concretely, an ML algorithm learns from experience E related to a task T and its performance is evaluated by a metric P. Its performance at T improves according to P after experience E (Goodfellow,

Bengio, and Courville, 2016). Depending on the available information of an ML problem, ML tasks can be generally classified into four categories:

- **Supervised Learning:** The values of input variables and their corresponding values of output variables are known. The learning process is to automatically find some regularities between input variables and output variables. Depending on different learning tasks, supervised learning can be further categorised into classification and regression. Classical supervised learning algorithms include logistic regression (Hosmer, Lemeshow, and Sturdivant, 2013), neural networks (NNs) (Rumelhart, Hinton, and Williams, 1986), support vector machines (SVMs) (Cortes and Vapnik, 1995), decision trees (DTs) (Quinlan, 1986), k-nearest neighbours (k-NN) (Fix and J., 1952).

- **Unsupervised Learning:** The values of input variables are known while the values of output variables are unknown. The learning task is to find some hidden patterns within the data based on input variables. An example of unsupervised learning is clustering, which learns the distribution of data to gather the samples into different groups. Representative unsupervised learning algorithms are k-means (Lloyd, 1982), autoencoders (Hinton and Salakhutdinov, 2006) or generative adversarial networks (Goodfellow et al., 2014).

- **Semi-supervised Learning** falls between supervised learning and unsupervised learning. The values of input variables are known while a part of the values of output variables are unknown. The unknown values of output variables can normally produce considerable improvements when learned with the known values. Several representative methods include generative models (Kingma et al., 2014) and low-density separation (Chapelle and Zien, 2005).

- **Reinforcement Learning (RL)** aims at choosing the most suitable action at a specific state in an environment to maximise the cumulative reward (Sutton and Barto, 1998). Classic RL algorithms include Q-Learning (Watkins and Dayan, 1992), Monte Carlo RL (Bouzy and Chaslot, 2006), SARSA (Rummery and Niranjan, 1994).

2.2 Meta-Learning: self interactions between ML and itself

ML has been extensively studied and boost the development of AI, however, it still suffers from several limitations, such as choosing the best set of parameter values and algorithm components based on human experience for improved performance, the learning process requiring expensive computation, developed solution becoming tailored only to the specific tasks handled and not generalising well to the unseen problems, and more. To address the limitations, ML has been interacting with itself for better performance as shown in Figure 2.1. Specifically, researchers filled the gap by applying another ML algorithm at a meta-level to learn meta-knowledge from fulfilling base-level ML tasks and then using the learned meta-knowledge to guide unseen ones. Such an approach is called meta-learning, also known as learning to learn. We provide a categorisation of meta-learning approaches based on the type of meta-knowledge as shown in Figure 2.1.

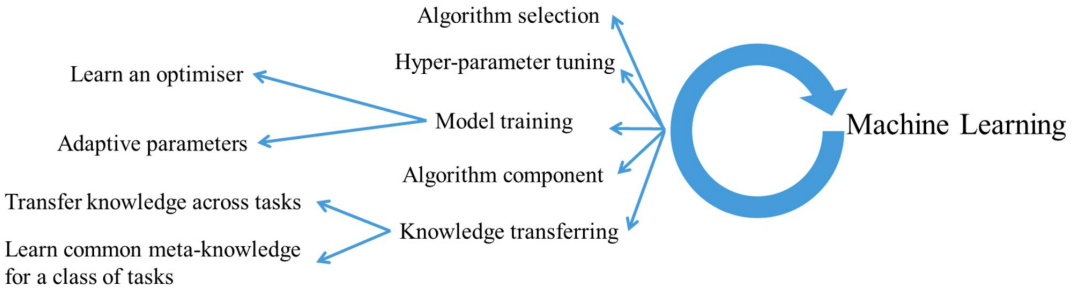


FIGURE 2.1: Interaction between ML and itself.

- **Learn to select algorithms.** To reduce human involvements, some meta-learning approaches use a meta-level ML algorithm to learn how to automatically select appropriate algorithms for an ML task (Gama and Brazdil, 1995; Brazdil, Soares, and Da C., 2003). Generally, these methods learn a classifier that determines a specific ML algorithm or a ranking of ML algorithms based on the characteristics of a task (Rendell and Cho, 1990; Brazdil, Soares, and Da C., 2003), or a regression model that captures the mapping between the characteristics of ML tasks and algorithm performance (Gama and Brazdil, 1995).

- **Learn to tune hyper-parameters.** Some meta-learning approaches learn how to tune hyper-parameters for an ML algorithm by extracting the

mapping between hyper-parameters and algorithm performance (Lorraine and Duvenaud, 2017; Baker et al., 2017). Then hyper-parameter tuning can be conducted efficiently based on the learned mapping.

- **Learn to train a model.** Model training is the core of ML and typically relies on a manually designed optimisation algorithm, such as gradient-based methods. For the sake of faster convergence and better performance, some meta-learning approaches learn how to train an ML model through learning an ML model to update the parameters of a base ML model based on the gradient information (Andrychowicz et al., 2016), or directly generate adaptive model parameters (Ha, Dai, and Le, 2016).
- **Learn an algorithm component.** An algorithm component can be seen as a discrete hyper-parameter of an ML algorithm. Some meta-learning methods substitute a hand-designed algorithm component, such as an activation function in NNs or a splitting criterion in DTs, with an ML model, so that, the learned algorithm component can be tailored to a specific task and contribute to better performance (Vercellino and Wang, 2017; Xiong, Zhang, and Zhu, 2017).
- **Learn to transfer knowledge.** To achieve a high generalisation ability, some works in the meta-learning field focus on how to effectively transfer knowledge across ML tasks. They aim to learn common meta-knowledge that can be shared across different but related tasks (Wang, Ramanan, and Hebert, 2017; Frans et al., 2018). Some of them delve into a specific challenging problem, few-shot image classification, which is also our research focus. A thorough overview of this topic is presented in Section 2.4.6.

2.3 Computer Vision Fundamentals

This section introduces all the computer vision techniques that are thoroughly applied in the experiments developed in the following chapters. Concretely, we first present the materials concerning the basics of image classification in Section 2.3.1. Then, attention mechanisms applied in the computer vision field are investigated and compared with ours in Section 2.3.2. Afterwards, several related detection techniques for images used in our work, namely edge detection, object detection and saliency object detection, are introduced in Section 2.3.3.

2.3.1 Image Classification

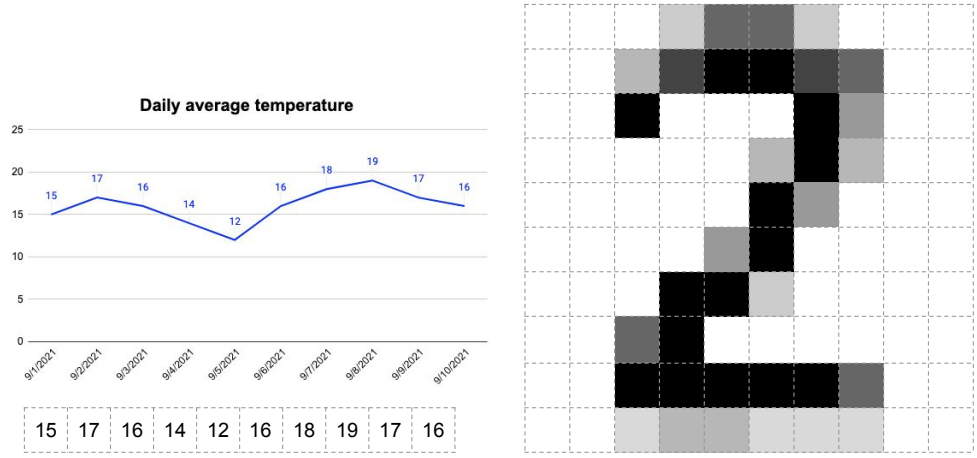
Image classification is one of the main tasks of computer vision, categorising an image into one of the target categories. There are various real-world problems of image classification, such as fingerprint identification (Peralta et al., 2015) or face recognition (Hu et al., 2015). As stated before, traditional methods for image classification generally include two stages, feature extraction and classification (O'Mahony et al., 2019). In the feature extraction stage, a raw image is transformed into informative low-dimensional numerical features. In the classification stage, a classifier is learned to make predictions based on the extracted features.

Recently, due to the progress of computing power, deep learning approaches have developed dramatically and combined the aforementioned two stages into a single one by training a deep model in an end-to-end manner and achieved state-of-the-art performance. Deep-learning methods are representation-learning methods with multiple levels of representation, obtained by composing simple but non-linear modules that each transform the representation at one level (starting with the raw input) into a representation at a higher, slightly more abstract level (LeCun, Bengio, and Hinton, 2015). Convolutional Neural Networks (CNNs) are one of the most popular deep learning architectures and have been widely applied in computer vision problems, such as image classification (LeCun, Bengio, et al., 1995) and object detection (Girshick et al., 2014). In the following subsection, the characteristics of CNNs are examined.

Convolutional Neural Networks

CNNs are a specialised kind of neural networks for processing data that has a known grid-like topology (LeCun, Bengio, et al., 1995), such as time series with 1D grid, and images with a 2D grid of pixels, as shown in Figure 2.2. Normally, there are two main types of layers to build up the architecture of CNNs, namely convolutional layer and pooling layer. We introduce their respective operation and characteristic as follows:

- Convolutional layer: The input of this layer is a set of feature maps. The convolution operation is performed on the input with the use of a set of filters/kernels to obtain a new set of feature maps as the output of this layer. Each new feature map corresponds to an output channel. The filters/kernels are learnable parameters determined by a training process. Compared to fully connected (FC) NNs, in which every neuron in one



(a) A time series with 1D grid

(b) An image with a 2D grid of pixels, each pixel is represented by a real number

FIGURE 2.2: Examples of data with grid-like topologies.

layer is connected to every neuron in the next layer, convolution operation usually leads to much fewer learnable parameters due to its main characteristic of sparse interactions and parameter sharing. The sparse interaction is accomplished by making the size of filters/kernels smaller than the input. The parameter sharing is conducted by learning a set of shared filters/kernels across all the locations on the input, rather than specific filters/kernels for each location.

- Pooling layer: This layer summarises locally extracted features into statistics (Vladimir and Vladimir, 2015). It is an important component in CNNs in order to reduce the number of parameters and computational burden, and improve the translation invariance of the network. The most commonly used pooling techniques are max-pooling and average-pooling, which downsample each sub-region by taking either the max or mean value of that sub-region. These two methods are simple and effective but have their own drawbacks. Max-pooling may lose useful information while average-pooling ignores the importance of relevant and irrelevant features. To address the issues, a few works have been proposed that explore a better way of pooling by theoretical analysis (Boureau, Ponce, and LeCun, 2010), using overcomplete rectangular pooling blocks (Jia, Huang, and Darrell, 2012), learning a linear combination or gated mask of max and average pooling (Lee, Gallagher, and Tu, 2016), considering overlapping between adjacent pooling regions (Krizhevsky, Sutskever, and Hinton, 2012), introducing detail-preserving image downscaling method (Saeedan et al., 2018), etc.

To make a final classification after the last convolutional block, one or few FC layers followed by a softmax function are usually concatenated to produce a probability distribution over all classes. Alternatively, we can compare the similarity between a test sample's embedding (a set of feature maps) and each class of training samples' embeddings from the last convolutional block, and classify the test sample into its nearest class based on a distance function, such as cosine distance, or a learned distance metric. This research field is also known as metric learning (Kulis et al., 2012). A number of research works on few-shot image classification belong to this field.

Since deep CNNs could suffer from the overfitting problem, the dropout technique is usually introduced to address the issue. The key idea is to randomly drop part of the units of neural networks during training and use the whole networks for testing, which can also be seen as a form of model averaging (Srivastava et al., 2014). Although the shared-filter architecture of CNNs decreases the number of model parameters which can reduce the model's capacity to overfit. Still, the experimental results in (Srivastava et al., 2014) show performing dropout in convolutional layers can prevent overfitting and further improve the performance on image recognition tasks. Some later works propose specific dropout techniques for the shared-filter architecture of CNNs by randomly dropping the entire feature maps (Tompson et al., 2015) or continuous regions in a feature map (Ghiasi, Lin, and Le, 2018). In this work, we apply dropout techniques to regularise the CNNs based feature extractor for few-shot image classification.

Recently, various improvements to the classic architecture of CNNs have been proposed, such as AlexNet (Krizhevsky, Sutskever, and Hinton, 2012), VGG (Simonyan and Zisserman, 2014), ResNet (He et al., 2016), DenseNet (Huang et al., 2017), EfficientNet (Tan and Le, 2019), to make CNNs deeper and more scalable to large and complex problems. ResNet is one of the most successful architectures, which utilises skip connections or shortcuts to jump over some layers. It has been widely selected as a backbone for comparisons on different methods in the few-shot image classification field. Following the same setting, we also choose it as a backbone in our experiments.

2.3.2 Attention Mechanisms

An attention mechanism is an technique that tells a machine learner where to focus (Zhu et al., 2019), which is inspired by the human perception system. In the computer vision field, especially when using CNNs, attention mechanisms

can be categorised into channel-wise attention and spatial attention (Woo et al., 2018), as shown in Figure 2.3. Channel-wise attentions concentrate on distinguishing the importance of different output channels (feature maps) (Woo et al., 2018). Spatial attention mechanisms (Zhu et al., 2019) aim to let CNNs focus on the relevant area on the feature maps (Lu et al., 2019; Chen et al., 2017). We leverage both attention mechanisms to develop two new strategies for meta-learning in this thesis.

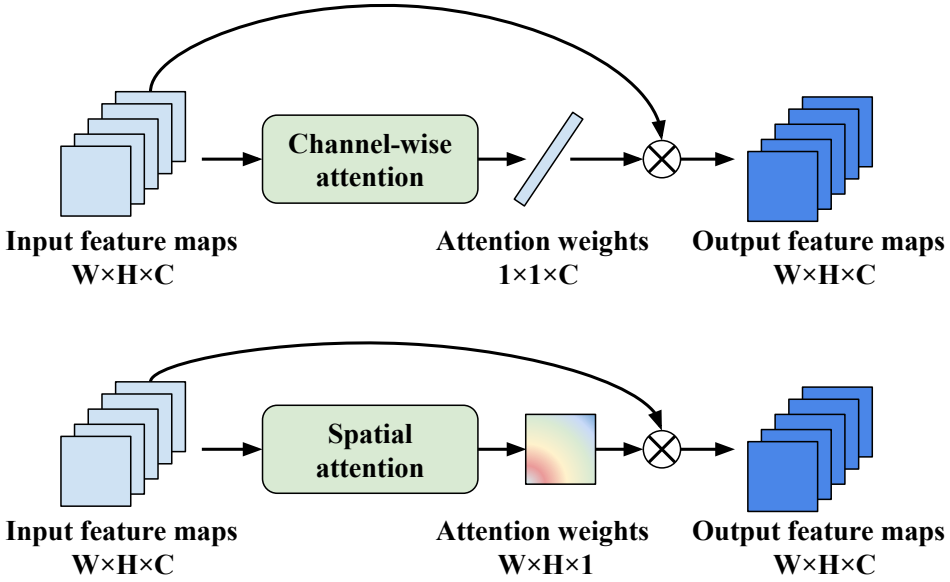


FIGURE 2.3: An illustration of channel-wise attention and spatial attention. The figure is inspired by (Woo et al., 2018). Channel-wise attention generates an attention weight for each convolutional channel. Spatial attention assigns an attention weight to each pixel on a feature map.

2.3.3 Related Detection Techniques for Images

There are various detection techniques in computer vision field, such as edge detection (Dhankhar and Sahu, 2013), corner detection (Dutta, Kar, and Chatterji, 2008), blob detection (Han and Uyyanonvara, 2016), object detection (Zhao et al., 2019b) and saliency object detection (Wang et al., 2021b). Since our objective is to mitigate background clutters with the help of auxiliary information, we aim to select detection techniques that are related to distinguishing foreground and background. Corner detection and blob detection target extracting certain kinds of features or regions in an image, which are widely used to extract informative features for the subsequent computer vision tasks (Dutta, Kar, and Chatterji, 2008; Han and Uyyanonvara, 2016). They treat the

features in foreground and background equally and focus more on specific patterns, therefore, they are not very helpful for mitigating background clutters on few-shot learning. To the best of our knowledge, there are three detection techniques that fulfil our desire, namely edge detection, object detection and saliency object detection, as shown in Figure 2.4. Edges targets detecting the points at which image brightness changes sharply (Dhankhar and Sahu, 2013). Since the brightness at the border between the foreground and background in an image usually changes greatly, we would expect edges could provide information about the outline of a target object. Bounding boxes are rectangles that tightly surround the target object in an image, which would provide information on where the foreground locates (Zhao et al., 2019b). Saliency maps highlight the target object in an image, which would display more exact localisation of the foreground (Wang et al., 2021b). Therefore, we choose these three detection techniques to assist few-shot image classification.

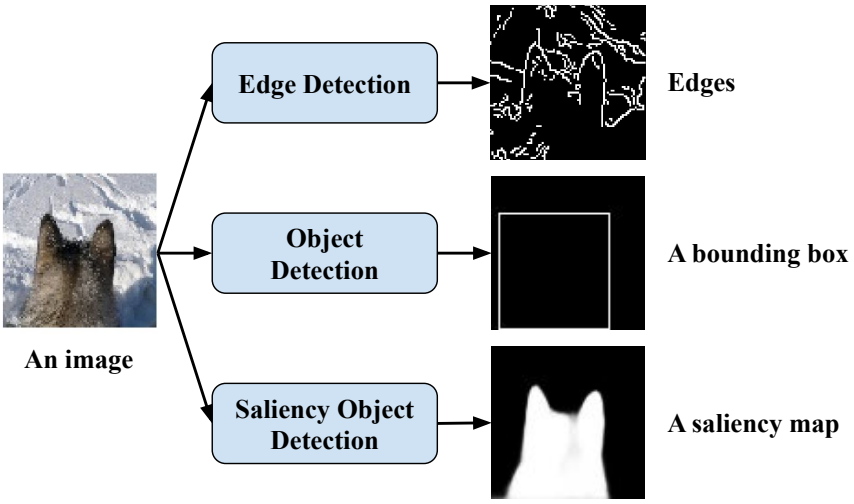


FIGURE 2.4: An illustration of related detection techniques for images.

Edge Detection

Edge detection targets detecting the points at which image brightness changes sharply (Dhankhar and Sahu, 2013). Those points are typically organized into a set of curved line segments, which is termed as edges. Since the brightness of the points between foreground and background in an image usually change sharply, we assume edge detection methods can extract the outline of a target object and provide us with some useful information to help few-shot image classification. Broadly, the existing edge detection methods can be categorised into gradient and Laplacian based approaches (Dhankhar and Sahu,

2013). The canny edge detector is one of the most commonly used image processing tools and it detects edges in a very robust manner (Canny, 1986). We use this technique to extract edges of images as a type of auxiliary information in this thesis, which have not been explored before.

Object Detection

Object detection is a technique that deals with detecting the concepts and locations of semantic objects of a certain class, such as humans or cars, in digital images and videos (Zhao et al., 2019b). A bounding box is normally drawn around the target object to locate it within an image. This technique has applications in many areas of computer vision, such as pedestrian detection (Brunetti et al., 2018), and video surveillance (Pérez-Hernández et al., 2020). Traditional object detection methods can be mainly divided into three stages: 1) selecting informative regions; 2) extracting features of the selected regions; 3) classifying the object in each region into a specific category. Recently, deep learning techniques have led the research field and achieved promising performance. R-CNN is the first work to introduce CNNs to object detection (Girshick et al., 2014), which uses selective search to extract boxes from an image. A few later approaches, such as Fast R-CNN (Girshick, 2015) and Faster R-CNN (Ren et al., 2015), improve it by considering the efficiency when handling a large number of regions. Another branch of works frame object detection as a single regression problem, straight from image pixels to bounding box coordinates and class probabilities (Redmon et al., 2016). There are many versions of this line of research, such as YOLOv1 (Redmon et al., 2016), YOLOv2 (Redmon and Farhadi, 2017), YOLOv3 (Redmon and Farhadi, 2018), etc. In our work, we utilise the most recent best-performing version, YOLOv4 (Bochkovskiy, Wang, and Liao, 2020) to extract bounding boxes in images as another kind of auxiliary information.

Saliency Object Detection

SOD works by predicting the most distinctive objects in an image, which has been widely applied to many object-level applications in various areas such as object recognition (Rutishauser et al., 2004), image retrieval (He et al., 2012), weekly supervised semantic segmentation (Wei et al., 2016; Wang et al., 2018a), image cropping (Wang, Shen, and Ling, 2018) and image captioning (Fang et al., 2015; Das et al., 2017). Normally, SOD includes two steps: 1) detecting the most salient object for a given image; 2) segmenting the salient object in

this image and generating a binary saliency map indicating the locations of salient pixels. Early SOD models mostly rely on low-level features (Zhu et al., 2014; Jiang et al., 2013) or heuristic priors such as contrast (Cheng et al., 2014) and background prior (Wei et al., 2012). However, early SOD models are not robust enough to handle complicated scenarios since it is difficult for them to capture high-level semantic information given the hand-craft features. Recently research into CNNs has significantly stimulated the development of SOD areas and many deep learning based SOD models have emerged. These SOD models are able to generate saliency maps accurately without any prior knowledge such as information on the background. In this work, EGNet (Zhao et al., 2019a), which is a recent approach offering competitive performance on standard SOD benchmarks, is chosen to generate saliency maps as another type of auxiliary information for few-shot image classification.

2.4 Meta-Learning for Few-shot Image Classification

By seeing a few examples, a human can quickly learn a new visual concept, while an ML algorithm needs to extract patterns from a large amount of data. To fill the gap, few-shot image classification, serving as one of the main problems of few-shot learning, classifies an image into its real category based on a limited number of training examples. Traditional ML and data augmentation techniques do not work well on this challenge, because there are very limited visual patterns within the training samples. Since meta-learning is inspired by the way of human learning, in which humans can quickly adapt already known knowledge to a new learning task, this technique has been widely applied to tackle few-shot image classification.

This section provides an overview of the specialised literature concerning the use of meta-learning approaches for few-shot image classification. First, the few-shot image classification problem is defined in Section 2.4.1. Then, we introduce the common benchmarks, training strategies, evaluation and network architectures for few-shot image classification in Section 2.4.2, 2.4.3, 2.4.4 and 2.4.5, respectively. Later on, a thorough review of the existing meta-learning approaches for few-shot image classification is provided in Section 2.4.6. Afterwards, a representative approach for few-shot image classification is introduced in Section 2.4.7. Finally, we summarise the overview on few-shot image classification in Section 2.4.8.

2.4.1 Problem Formulation

In few-shot learning field, a typical classification task is normally called N-way K-shot classification, in which N-way stands for N categories and K-shot represents K training samples in each category. A typical N-way K-shot classification task classifies a test example into one of N unique classes based on $N \times K$ labelled training samples. In an extreme case, K equals to 1 and only one sample is available of each class. Such problems are called one-shot classification. In another extreme case, K could equal to 0, and no training sample can be accessed. In this scenario, some textual information is usually provide to compensate the lack of trainig samples. This problem is named as zero-shot learning, which is not our focus in this thesis. For each N-way K-shot classification task, the training set $D_{train} = \{(x_{i,j}, y_i)|1 \leq i \leq N, 1 \leq j \leq K\}$ includes N classes of K training samples. The test set $D_{test} = \{x_q, y_q\}_{q=1}^{N_q}$ consists of N_q test samples x_q whose labels y_q belong to D_{train}. A machine learner can be trained based on D_{train} to classify test samples, however, the limited training samples often lead to overfitting. Since we focus on few-shot image classification, each training sample $x_{i,j}$ refers to an image in this research.

In a typical meta-learning setting, there are three meta-data sets, meta-training set $\mathfrak{D}_{meta-train}$, meta-validation set $\mathfrak{D}_{meta-validation}$ and meta-testing set $\mathfrak{D}_{meta-test}$. There is no overlapping among their label spaces. An illustration of the split of these three sets on a specific data set is shown in Figure 2.5. A large number of few-shot learning tasks can be constructed based on each of them, such as $\mathfrak{D}_{meta-train} = \left\{\left(D^j_{train}, D^j_{test}\right)\right\}_{j=1}^{N_{train}}$. Meta-learning approaches target to learn some transferable meta-knowledge from a meta-training set $\mathfrak{D}_{meta-train}$ and apply it to tackle unseen few-shot learning tasks in the meta-testing set $\mathfrak{D}_{meta-test}$. The meta-validation set is usually used to select hyper-parameters and the best performing model.

2.4.2 Benchmarks for Few-shot Image Classification

There are a few benchmarks for few-shot image classification, among which Omniglot, miniImageNet, tieredImageNet and CUB are the most widely used data sets in the research field. We choose these four commonly applied data sets to evaluate our methods. Here, we introduce the four benchmarks in detail as follows:

- **Omniglot** consists of 1,623 handwritten characters collected from 50 alphabets. There are 20 examples of each character, which are drawn by

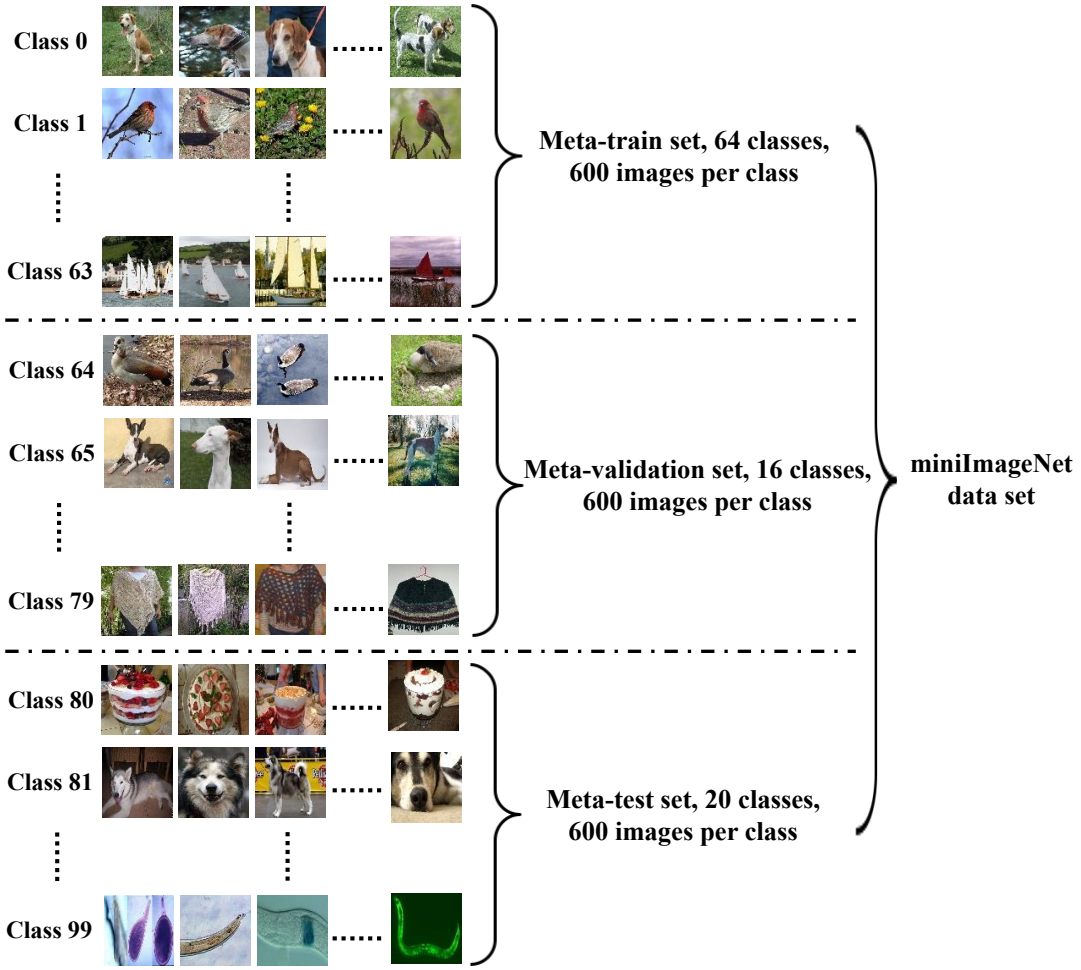


FIGURE 2.5: An illustration of the split of meta-train, meta-validation and meta-test set for few-shot image classification based on miniImageNet data set. There are 100 classes of images, 64 for meta-train, 16 for meta-validation and 20 for meta-test.

different people. We augmented the data sets with rotations with multiple 90 degrees as proposed by (Santoro et al., 2016) to get 6492 classes. Following (Finn, Abbeel, and Levine, 2017), we randomly select 1,200 classes (4,800 classes after augmentation) for meta-train, 100 classes (400 classes after augmentation) for meta-validation, and the remaining 323 (1292 classes after augmentation) for meta-test. Note that there is no specific split of classes for meta-train, meta-val and meta-test in the research field. All the input images are resized to 28×28 as suggested by (Vinyals et al., 2016) to get a suitable sized embedding.

- **miniImageNet** was proposed by (Vinyals et al., 2016) derived from the original ILSVRC-12 data set (Russakovsky et al., 2015). It comprises 100 classes of colour images with 600 of each (60,000 in total). In our experiments, we use the widely used splits proposed by (Ravi and Larochelle,

2017), which divides the 100 classes into 64 for meta-training, 16 for meta-validation and 20 for meta-testing. All the input images are resized to 84×84 as done by most few-shot learning approaches (Ravi and Larochelle, 2017; Snell, Swersky, and Zemel, 2017; Finn, Abbeel, and Levine, 2017). Note that the existing approaches use different tools to resize the images in miniImageNet. We use the library provided by OPENCV (Bradski, 2000) following (Liu et al., 2019b).

- **tieredImageNet** was proposed by (Ren et al., 2018), which is a larger subset of ILSVRC-12 data set (Russakovsky et al., 2015) consisting of 608 classes of colour images (779,165 in total). These classes are grouped into 34 broader categories based on the higher-level nodes in the ImageNet hierarchy, in which 20 of them are used for meta-training, 6 for meta-validation and 8 for meta-testing. Therefore, there are 351, 97 and 160 classes for meta-training, meta-validation and meta-testing in total. The split of classes follows the widely used one in (Ren et al., 2018). All the images are resized to 84×84 following the existing approaches (Ren et al., 2018). This data set is more challenging and realistic compared to miniImageNet, since the meta-training and meta-testing set are less similar in the semantic space.

- **CUB**: CUB-200-2011 proposed by (Wah et al., 2011) is an image data set with photos of 200 bird species. It is comprised by 200 classes of colour images (11,788 in total). Following the split of classes in (Chen et al., 2018), in our experiments, we use 100 of them for meta-training, 50 for meta-validation and 50 for meta-testing, in which each image is resized to 84×84. Since the data set only contains bird species, the few-shot learning tasks on this data set can be seen as fine-grained classification tasks.

The aforementioned four data sets could have different characteristics and challenges when they are chosen to evaluate few-shot learning approaches, since they consist of different types of images. We provide the characteristics and some examples of the four benchmarks in Table 2.1 and Figure 2.6, respectively. It can be seen in Figure 2.6, the images in Ominiglot are relatively simple compared to those in other data sets, since they are grayscale images without background clutters. However, it is still a challenging benchmark because some characters from different alphabets are very similar (eg. the second and third characters in the first row displayed in Figure 2.6) and the handwriting of different people could be in diverse styles. The images in other three

data sets are real-world RGB images. There are lots of intrinsic uncertainties in them. For example, the first image of the second row in Figure 2.6 shows an unrepresentative Alaskan malamute, which only displays its back. Some other images, such as the second and third images in the second row, include complex background clutters. These intrinsic uncertainties of the real-world images could be a real challenge for few-shot image classification. TieredImageNet is more challenging than miniImageNet because it includes much more categories and its meta-training and meta-testing set are less similar in the semantic space. CUB is a fine-grained data set, which only contains bird species as shown in the last row of Figure 2.6. This benchmark is usually used to evaluate approaches on fine-grained few-shot image classification.

TABLE 2.1: The characteristics of few-shot image classification benchmarks. cls stands for the number of classes in each data set. meta-train/meta-val/meta-test represents the split of classes. avg_imgs/cls means the average number of images per class. imgs counts the total number of images in each data set. resolution stands for the image resolution.

Data sets	cls	meta-train/meta-val/meta-test	avg_imgs/cls	imgs	resolution
Omniglot	1623	1200/100/323	20	32,460	28×28
miniImageNet	100	64/16/20	600	60,000	84×84
tieredImageNet	608	351/97/160	1282	779,165	84×84
CUB	200	100/50/50	59	11,788	84×84

FIGURE 2.6: Examples of images in four widely used benchmarks for few-shot image classification.

2.4.3 Training Strategies for Few-shot Image Classification

As mentioned before, the existing approaches of few-shot image classification mostly utilise meta-learning to extract meta-knowledge from a large number of few-shot learning tasks based on a meta-training set. This process is normally called meta-training and the training strategies in the research field can be broadly categorised into two classes, namely episode-based training (Snell, Swersky, and Zemel, 2017; Finn, Abbeel, and Levine, 2017) and large scale training (Qiao et al., 2018; Sun et al., 2019a). Both train on the meta-training set, which is a fair test. The illustration of these two training strategies is shown in Figure 2.7.

1. **Episode-based training** aims to imitate the learning processes in meta-testing during meta-training. In each meta-training iteration, a number of N-way K-shot learning tasks $\left\{\left(D^k_{train}, D^k_{test}\right)\right\}^{N_m}_{k=1}$ are sampled from the meta-training set $\mathfrak{D}_{meta-train}$, N_m is the meta-batch size. The meta-training is performed based on the tasks sampled on the fly, which can be seen as episodes. Figure 2.7 (a) shows an example on miniImageNet. In each meta-training iteration, a 5-way 1-shot learning task is sampled from the meta-training set (the meta-batch size is 1). The meta-knowledge can be accumulated by solving each 5-way 1-shot learning task.

2. **Large scale training** utilises the meta-training set as a whole. Specifically, it adds a fully-connected (FC) layer on the feature extractor and classifies all the classes in the meta-training set simultaneously based on all the available training examples of each class, which is similar to the pre-training on ImageNet for recognition tasks. An example based on miniImageNet data set can be found in Figure 2.7, in which a 64-class classification task is conducted based on all the training samples of the 64 classes in the meta-training set. After the large scale training, the FC layer is removed, the feature extractor can be directly applied to few-shot classification tasks or be fine-tuned for a few epochs based on episode-based training. This large scale training usually provides better performance on a deep model.

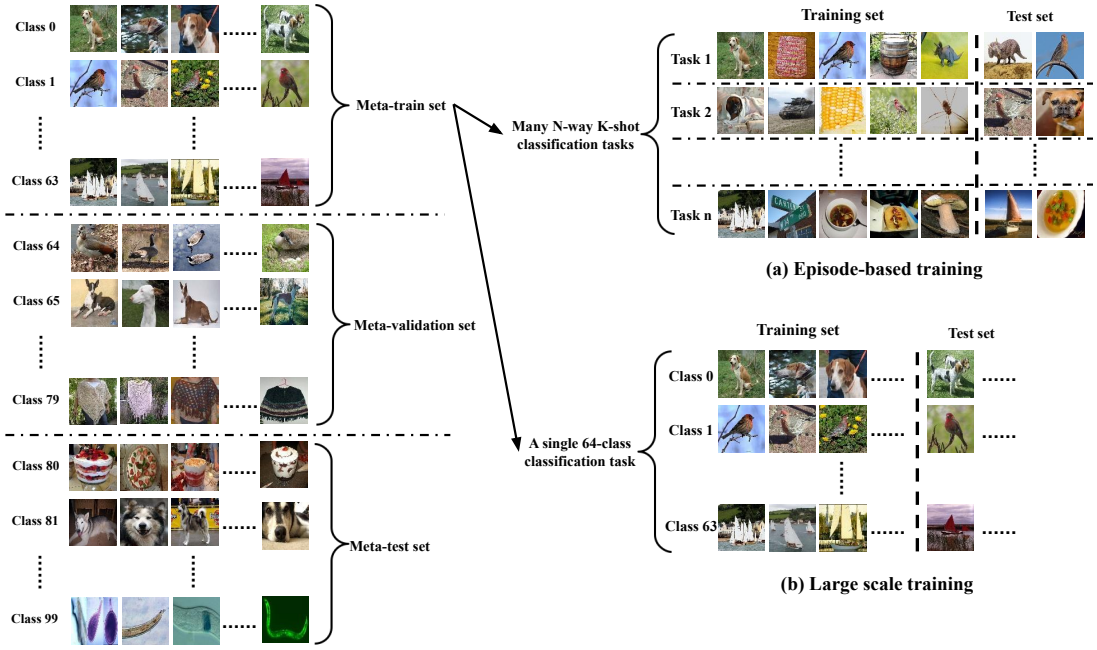


FIGURE 2.7: The illustration of the two training strategies for few-shot image classification on miniImgeNet: (a) episode-based training; (b) large scale training.

2.4.4 Evaluation for Few-shot Image Classification

The evaluation for few-shot image classification is conducted on a meta-testing set as shown in Figure 2.8. This is a common method to evaluate a few-shot learning approach in the research field. During meta-testing, the existing works mostly evaluated their methods on a number of randomly sampled N-way K-shot classification tasks from the meta-testing set. Specifically, on each task, we can obtain a classification model based on the few training samples and the meta-knowledge extracted from the meta-training process. Then, the testing samples in the test set is evaluated based on the classification model and the test accuracy for each specific task, $test\ acc_i$, can be obtained. As shown in Figure 2.8, the overall accuracy for the meta-testing phase is the average of test accuracies of all the tasks,

$$avg\ acc = \sum_{n}^{i=1} test\ acc_i \tag{2.1}$$

Since there are a large number of different tasks in meta-testing, it is possible to sample a large proportion of easy-to-classify or difficult-to-classify tasks using different seeds, which would lead to a result with high variance. To get a more reliable result, we evaluate our approach on 6,000 randomly sampled tasks for all data sets. The average classification accuracy and 95% confidence interval,

$CI_{95\%}$, are reported,

$$CI_{95\%} = avg\,acc \pm 1.96\frac{\sigma}{\sqrt{n}} \tag{2.2}$$

n is the number of meta-testing tasks, σ is the standard deviation of $test\,acc_i$.

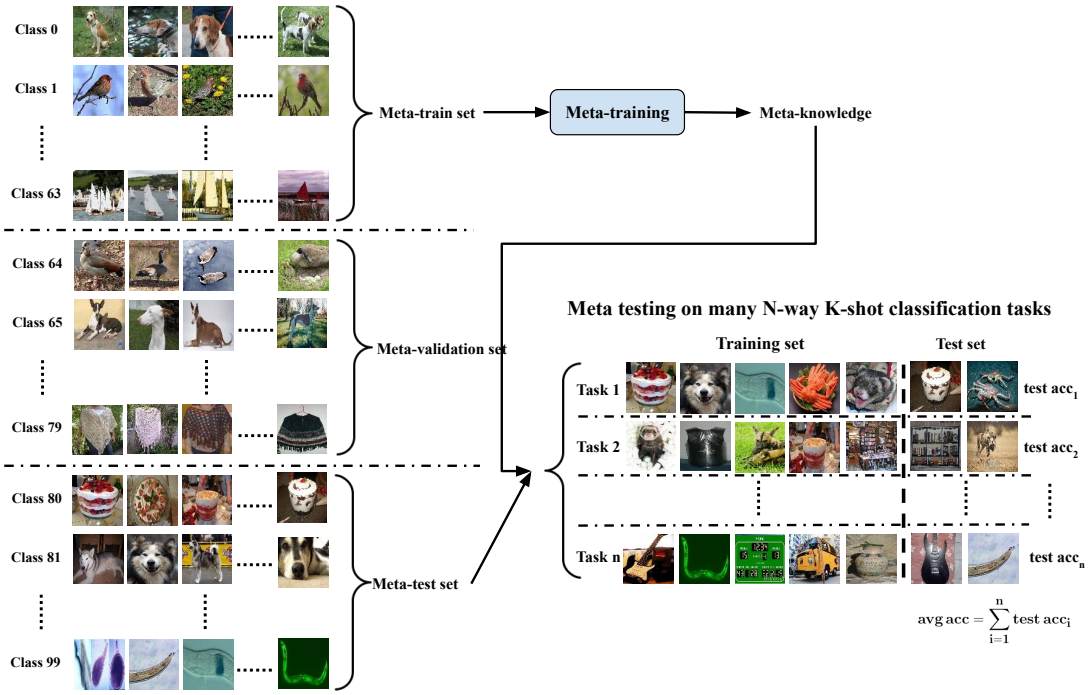


FIGURE 2.8: The illustration of the evaluation on meta-testing set for few-shot image classifciation.

2.4.5 Common Network Architectures for Few-shot Image Classification

In the field of few-shot image classification, a CNNs-based model is commonly used to extract features. The existing few-shot learning approaches apply different CNNs-based architectures, such as plain CNNs (Snell, Swersky, and Zemel, 2017), residual networks (Mishra et al., 2018) or wide residual networks (Rusu et al., 2019), with a certain number of layers. Among them, we choose the network architecture of 4Conv and Res12, as a feature extractor for two reasons. First, they are the most widely used network architectures for the evaluation in the research field. Second, the few-shot learning community focuses more on the effectiveness of the meta-learning strategy rather than the complexity of networks. Therefore, we choose the two most widely used network architectures and compare our approach with other methods based

on the same architecture for fairness. The detailed architecture of 4Conv and Res12 are described as follows:

1. **4Conv** consists of 4 convolutional blocks. Each block is composed of a 3×3 convolution with 64 filters, followed by BN, a ReLU nonlinearity and a 2×2 pooling operation.
2. **Res12** consists of 4 residual blocks, each of them is composed of 3 convolutional layers. An pooling operation is performed after each residual block. Each convolutional layer in the residual block consists of a 3×3 convolution with k filters, followed by BN, a Leaky ReLU (0.1) nonlinearity. k is set to be 64 in the first residual block and is doubled every next block.

2.4.6 A Review of Methods for Few-Shot Image Classification

Few-shot image classification problem was proposed at an early time (Fei-Fei, Fergus, and Perona, 2006), targeting learning with few labelled examples. It is restricted by two core issues, namely the lack of information and intrinsic uncertainty as stated in Chapter 1. The pioneer works use generative models to tackle the problem with the help of some prior knowledge, such as the pre-trained ML models on some extra categories (Fei-Fei, Fergus, and Perona, 2006) or the strokes of hand-written characters (Lake, Salakhutdinov, and Tenenbaum, 2015). Due to the rapid progress of deep learning techniques, more and more recent works incorporate deep learning models to deal with few-shot image classification via a meta-learning approach. Generally speaking, depending on the different methodologies, existing approaches on few-shot learning can be classified into three main categories, as shown in Figure 2.9.

- **Fast parameterisation based approaches** learn a general fast parameterisation strategy that can quickly fine-tune a learner or predict the parameters of a learner for each particular few-shot learning task. This branch of works generally targets addressing the core issue of lack of information, since the limited training samples in a few-shot learning task easily leads to overfitting problems. The fast parameterisation could prevent over-training on the limited training data. A straightforward way is to fine-tune an FC layer based classifier depending on a pre-trained encoder (the first red rectangle in Figure 2.9). A pioneering work shows this direct method does not perform well on few-shot image classification (Ravi and

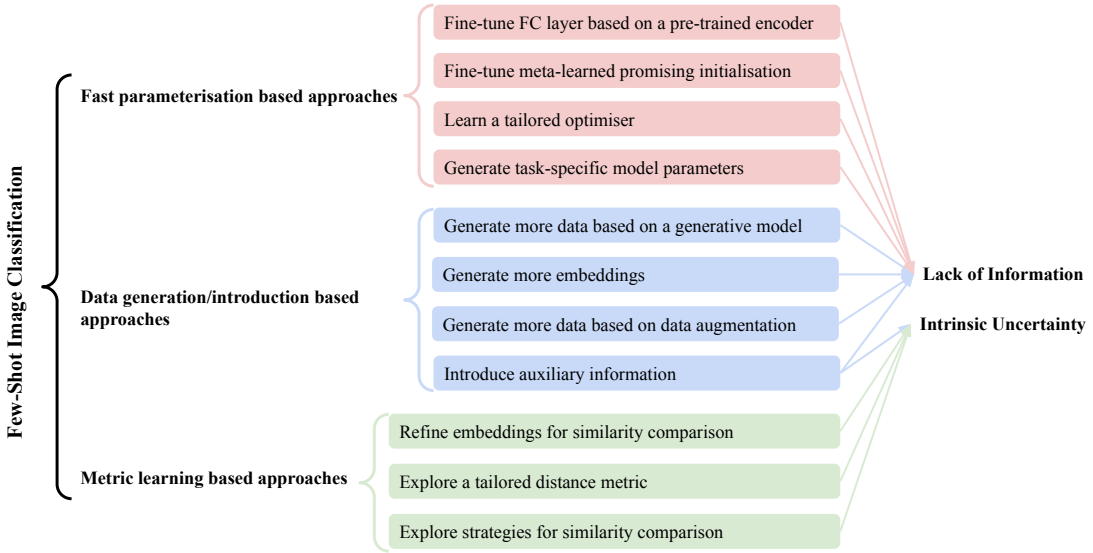


FIGURE 2.9: A taxonomy of approaches for few-shot image classification. The rounded rectangles in red, blue and green represent different research lines in fast parameterisation, data generation/introduction and metric learning based approaches, respectively. These research lines focus on either of the two core issues of few-shot image classification, lack of information and intrinsic uncertainty. Very few of them take into account both of the key problems at the same time.

Larochelle, 2017), potentially due to overfitting problems. To address the issue, some recent works revisit this baseline (Guneet et al., 2020; Tian et al., 2020b) and dramatically improve its performance by introducing several regularization techniques, such as mixup augmentation (Zhang et al., 2018a), label smoothing (Szegedy et al., 2016), and knowledge distillation (Hinton, Vinyals, and Dean, 2014), etc. Another research line learns a promising parameter initialisation that can be quickly fine-tuned to different task-specific parameters (the second red rectangle in Figure 2.9), in which Model-Agnostic Meta-Learning (MAML) for fast adaptation of deep networks (Finn, Abbeel, and Levine, 2017) is the most representative method. Although it sheds new light on how to initialise deep networks for novel tasks, it still has a few weaknesses, such as a high computational burden (Alex, Joshua, and John, 2018), and reusing more features rather than rapid fine-tuning (Jaehoon et al., 2021). To overcome the weaknesses, there are many extensions of MAML having been proposed by taking into account the computational burden (Alex, Joshua, and John, 2018), adaptive learning rate (Li et al., 2017; Kwon et al., 2021), meta-level overfitting problems (Jamal and Qi, 2019), the heterogeneity and homogeneity of learning tasks (Yao et al., 2019; Rusu et al., 2019;

Sun et al., 2019a), the robustness regularisation (Ren et al., 2021), the representation change rather than representation reuse (Jaehoon et al., 2021), adversarial training (Wang et al., 2021a; Kwon et al., 2021), and semi-supervised learning (Li et al., 2019e). Since the fine-tuning process relies on a pre-defined optimisation algorithm, which is not specially designed for few-shot learning problems, some other approaches train a meta-learner to learn the update rule with respect to gradients (the third red rectangle in Figure 2.9), which is tailored to few-shot image classification (Ravi and Larochelle, 2017; Flennerhag et al., 2020). The previous approaches all need fine-tuning for certain steps, which is not quick enough to make a prediction. To further fasten a parameterisation process, some other methods learn a meta-learner to directly predict the task-specific model parameters without fine-tuning (the fourth red rectangle in Figure 2.9). Broadly, most of them learned a general feature extractor and a meta-learner to predict the class-specific or task-specific parameters of the fully connected layer (Qiao et al., 2018; Gidaris and Komodakis, 2018; Qi, Brown, and Lowe, 2018; Gidaris and Komodakis, 2019; Sun et al., 2019a).

- **Data generation/introduction based approaches** tackle few-shot learning by introducing more data either from a generative/augmentation method or an auxiliary data set, aiming to compensate for lack of information mostly. Some approaches apply generative models, such as generative adversarial networks (Goodfellow et al., 2014), to generate more artificial data to assist training (Edwards and Storkey, 2017; Wang et al., 2018b; Gao et al., 2018; Zhang et al., 2018b) (the first blue rectangle in Figure 2.9). Since the original image space may include more uncertainties than the highly summarised embedding space, to reduce such uncertainties, some later works utilise generative models in the embedding space to generate more auxiliary embeddings to assist in representation learning (Chen et al., 2020; Li et al., 2020b) (the second blue rectangle in Figure 2.9). However, the above data generation approach cannot always guarantee the correctness of the generated examples/embeddings. Therefore, a few other methods conduct data augmentation directly on the original images to obtain more samples. Specifically, they introduce various data augmentation techniques to expand the limited training data for few-shot image classification (the third blue rectangle in Figure 2.9),

such as mixing foregrounds and backgrounds (Zhang, Zhang, and Koniusz, 2019), image deformation (Chen et al., 2019b), and jigsaw augmentation (Chen et al., 2019a), etc. The above mentioned methods obtain more data only based on the limited training data in a few-shot image classification task, to use more beneficial information, another research line of works introduce auxiliary information from an extra data source to assist few-shot learning (the fourth blue rectangle in Figure 2.9). The introduced auxiliary information can be of various types strengthening different weaknesses. For example, the pre-trained word embeddings of category names (Li et al., 2019a; Xing et al., 2019) and a category graph (Liu et al., 2019a) could help a feature extractor to learn more distinguishing features between categories. The attribute annotations of categories (Tokmakov, Wang, and Hebert, 2019) is able to provides more detailed features. The saliency maps (Zhang, Zhang, and Koniusz, 2019) would help a feature extractor to focus more on the target object in an image. To leverage these different types of auxiliary information, there are also many ways, like combining the embeddings of an training image and its category name to strengthen the class representation (Xing et al., 2019), or using category names to build category hierarchy and conduct hierarchical classification (Li et al., 2019a), etc. The work in (Zhang, Zhang, and Koniusz, 2019) incorporates saliency maps to mix up the foregrounds and backgrounds of few training samples considering the two core issues simultaneously, which is related to our work in Chapter 5. This method only focuses on saliency maps and develop a specific strategy to leverage such information. However, we provide a comprehensive exploration of what is suitable auxiliary information and how to effectively leverage it to mitigate background clutters in few-shot image classification.

- **Metric learning based approaches** address few-shot learning by comparing the similarities in a learned metric space. Specifically, they learn a general feature extractor to transform the training and test examples into embeddings, then assign a test embedding to its nearest training class according to a distance metric, such as a Euclidean or cosine distance (Snell, Swersky, and Zemel, 2017; Vinyals et al., 2016). These approaches mostly concentrate on the core issue of intrinsic uncertainties, especially unrepresentative samples, and tackle it from three aspects as shown in Figure 2.9. One research line refines the embeddings of few training samples and gain a robust class representation for the subsequent similarity comparison (the first green rectangle in Figure 2.9). The

most representative approach is Prototypical Networks (ProtoNet) (Snell, Swersky, and Zemel, 2017), which simply averages the training embeddings of each category to form a robust class representative. The work in (Tianshi, Marc, and Sanja, 2020) further provides theoretical analysis about how ProtoNet works. A limitation of this method is that it takes the whole embeddings obatained from a feature extractor to the subsequent similarity comparison without taking into account choosing relevant features. To address the issue, some later works improve ProtoNet by finding task-relevant features (Li et al., 2019b), separating the embeddings of foregrounds and backgrounds (Wertheimer and Hariharan, 2019), and transforming training embeddings by various set-to-set adaptation functions (Ye et al., 2020). As stated before, unrepresentative samples may exist in the limited training data, ProtoNet and its extensions simply averages the training embeddings and do not consider the importance of different training samples. To fill the gap, some work introduces attention mechanisms to assist weighted aggregation of training embeddings (Yan, Zhang, He, et al., 2019). Another research line focuses on developing a tailored distance metric for few-shot image classification (the second green rectangle in Figure 2.9). The previous methods mostly utilise a pre-defined distance metric, such as Euclidean or cosine distance, however, these distance metrics may not be suitable for few-shot learning with intrinsic uncertainties. Targeting on the issue, some works delve into developing a tailored distance metric for few-shot image classification by learning to compare the similarity between embeddings with a meta-learner (Sung et al., 2018), or furhter adding the distribution consistency into a distance metric (Li et al., 2019c). Some other works in metric learning approaches explore various strategies for a similarity comparison (the third green rectangle in Figure 2.9), such as using multiple class representatives to compare (Allen et al., 2019), comparing similarity in subspace (Simon et al., 2020), comparing the similarity between each pair of local feature vectors from two embeddings (Li et al., 2019d). Some other approaches in this research line construct graphs based on the extracted embeddings and transformed the metric learning problems into label propagation or edge labelling problems using graph neural networks (Liu et al., 2019b; Kim et al., 2019; Garcia and Bruna, 2018).

2.4.7 ProtoNet: A Representative Approach for Few-Shot Image Classification

In this section, we introduce a representative method for few-shot image classification, ProtoNet (Snell, Swersky, and Zemel, 2017). It has been widely chosen in the literature as a backbone to test various meta-learning algorithms due to its simplicity and effectiveness (Wang et al., 2018b; Sung et al., 2018; Li et al., 2019b). We also apply ProtoNet as a backbone to evaluate our methods in Chapter 4 and 5. Generally, the ProtoNet method includes the following components: an embedding module, a distance module, a loss function and an inference mechanism. We describe each component in detail as follows.

- **Embedding module:** The embedding module (feature extractor) f_ϕ transforms a sample into a high-level representation, which normally consists of a few building blocks, including convolutional layers, BN layers, activation functions and pooling operations.

  Given a training set $D_{train} = \{(x_{i,j}, y_i) | 1 \leq i \leq N, 1 \leq j \leq K\}$, a testing set $D_{test} = \{(x_q, y_q)\}_{q=1}^{N_q}$, the feature extractor f_ϕ transforms a training example $x_{i,j}$ and a test example x_q into an embedding $E_{i,j} = f_\phi(x_{i,j})$ and $E_q = f_\phi(x_q)$. Following the main idea of ProtoNet, the prototype of each class is the mean embedding of the training examples belonging to its class:

$$P_i = \frac{1}{K} \sum_{j=1}^{K} f_\phi(x_{i,j}), P = \{P_i\}_{i=1}^{N} \tag{2.3}$$

- **Distance module:** This module applies a distance metric $d(\cdot)$ to measure the similarity between the embedding of a test example $f_\phi(x_q)$ and the prototype of each class P_i as $d(P_i, f_\phi(x_q))$. The commonly used distance metrics are Euclidean distance, cosine distance, scaled Euclidean or cosine distance.

- **Loss function:** A cross-entropy loss is computed on each task during meta-trainng. First, the softmax function is applied over the negative distance between the test embeddings and the prototypes as follows:

$$p_\phi(y = y_i \mid P, x_q) = \frac{\exp(-d(P_i, f_\phi(x_q)))}{\sum_{i'} \exp(-d(P_{i'}, f_\phi(x_q)))} \tag{2.4}$$

Then, the loss function can be formulated as

$$L(p_\phi(y = y_q \mid P, x_q)) = -\frac{1}{N_q}\sum_{q=1}^{N_q} \log\ p_\phi(y = y_q \mid x_q) \tag{2.5}$$

where N_q is the number of test examples in each training task, y_q is the true label of x_q.

- **Inference:** The inference is conducted on the meta-testing set $\mathfrak{D}_{meta-test}$, whose label space has no overlapping with the meta-training set $\mathfrak{D}_{meta-train}$. The procedure is nearly the same with the meta-training phase through the trained feature extractor, distance module and a softmax function. Then, a test example can be classified into one category by taking its highest probability:

$$\overline{y}_q = \underset{i}{\operatorname{argmax}}\ p_\phi(y = y_i \mid P, x_q) \tag{2.6}$$

where x_q is a test example in a meta-testing task and $\overline{y}_q$ is its predicted label.

2.4.8 Few-Shot Image Classification: Current Gaps and Limitations

Having reviewed the whole research field of few-shot image classification, it can be found out that there are very few works taking into account the two core issues of few-shot image classification at the same time as shown in Figure 2.9. Fast parameterisation based approaches mostly address the overfitting problem caused by lack of information. Obtaining more data based methods often compensate for a lack of information by generating or introducing more data. Most of them ignore the intrinsic uncertainties of few-shot image classification reflected by unrepresentative training samples and background clutters. Instead, metric learning based approaches generally tackle the intrinsic uncertainties by refining training embeddings or developing strategies for similarity comparison, while they pay less attention to the issue of lack of information. To the best of our knowledge, there are only two prior works considering both of the core issues together (Zhang, Zhang, and Koniusz, 2019; Wertheimer and Hariharan, 2019). Nevertheless, they only address the two issues by introducing auxiliary information to mitigate background clutters. Differently, we aim

to confront the two core issues from various perspectives, namely maximising the use of training samples, feature extraction and auxiliary information. From the point of view of auxiliary information, different from the aforementioned two research works, we provide a thorough exploration of using different methods to leverage different types of auxiliary information to assist few-shot image classification.

2.5 Summary

This chapter has provided the necessary background knowledge for the conducted research throughout this thesis. First, a global overview of self and dual interactions between ML and optimisation has presented, in which meta-learning served as the main area of this study. This review has led us to the research field that aims to make AI more intelligent and learn as humans. We focused on a specific challenging problem in this field, few-shot image classification, which targeted learning a new visual concept from a limited number of training samples. The basic concepts about image classification and several computer vision techniques further employed through the experiments have been clarified. These would help to understand the proposed attention mechanisms in Chapter 3 and 4, as well as the introduced auxiliary information in Chapter 5. Finally, a thorough review of few-shot image classification has presented, which made the motivation and research questions raised in Chapter 1 clearer. In the next chapter, as the first stage of our research, we aim to address the first research question, how to sufficiently exploit the limited training samples in few-shot image classification.

Chapter 3

Learning to Aggregate Embeddings for Few-Shot Image Classification with Meta-level Dropout

3.1 Introduction

Few-shot image classification is limited by its two core issues, the lack of information and intrinsic uncertainties. The review of few-shot image classification in the previous chapter has shown that most of the existing approaches focus on either of the two problems, and very few of them take into account both of them simultaneously. Hence, an approach that can address both of key issues at the same time is demanded. According to the previous review, existing methods for few-shot image classification mostly apply a meta-learning framework to extract meta-knowledge from a number of base learning tasks constructed by a large extra data set. Since the final goal is to make correct predictions on an unseen target task, we argue that more attention should be paid to the target task itself, especially to the limited training samples. Therefore, at the first stage of our research, we target tackling the two main problems of few-shot image classification from the perspective of maximising the utility of limited training samples.

As presented in the previous chapters, there is a certain probability that the limited training samples in a few-shot image classification task can include some unrepresentative samples. In this scenario, if we simply use the mean of each class's embeddings as the class representative, as performed in (Snell, Swersky, and Zemel, 2017), the possible unrepresentative samples may force the representative to deviate from the class centre in the embedding space as shown in Figure 3.1 (a). Therefore, it is necessary to appropriately handle the

effect of unrepresentative samples. Existing methods generally use an attention mechanism to assign lower weights to them. However, we argue that an unrepresentative sample may actually contain some useful features, which could help strengthen a part of the class representative. As shown in Figure 3.1 (a), even the outlier sample represented by the blue rounded rectangle with dashed border includes a useful feature (the light blue square). Directly assigning a lower weight to an unrepresentative sample based on attention mechanism may lose some useful features. Therefore, a more proper attention operation that can preserve as much as useful information is needed.

In the extreme case of one-shot tasks, since there is only one training sample per class, we cannot aggregate multiple training samples to obtain each class's representative as we do for K-shot tasks. The existing works normally simply use the only training embedding of each class as its class representative (Snell, Swersky, and Zemel, 2017; Sung et al., 2018). Since we target multi-class classification problems, we argue that there are some beneficial information from other classes that could be utilised to strengthen one class's representation. The assumption is that even the examples of different classes may share some similar features as shown in Figure 3.1 (b). The figure displays a 4-way 1-shot learning task, in which each coloured rounded rectangle represents the only training embedding of each class. Each training embedding consists of a set of feature maps (coloured squares), and the feature maps in the same colour scheme mean that they are similar in the feature space. It can be seen in Figure 3.1 (b) that although the blue and red embeddings belong to different classes, they still share a similar feature represented by the green squares. These two similar features could help each other to strengthen their respective representation, so that, the blue and red classes can be better distinguished from other classes. A concrete illustration can be found in Figure 3.2, which shows a 5-way 1-shot classification task that targets at distinguishing 'goose', 'bird', 'bus', 'crab' and 'jellyfish'. From the given five training examples, we can see that the shape of the heads of 'goose' and 'bird' are similar. Assuming that one of the channels in the last convolutional layer corresponds to the shape of a head, weighted aggregating the feature maps of 'goose' and 'bird' in that channel could help 'goose' and 'bird' to be better distinguished from 'bus', 'crab' or 'jellyfish'.

In addition, meta-learners may also suffer from overfitting. Although meta-learners are trained on different few-shot learning tasks, they may consist of

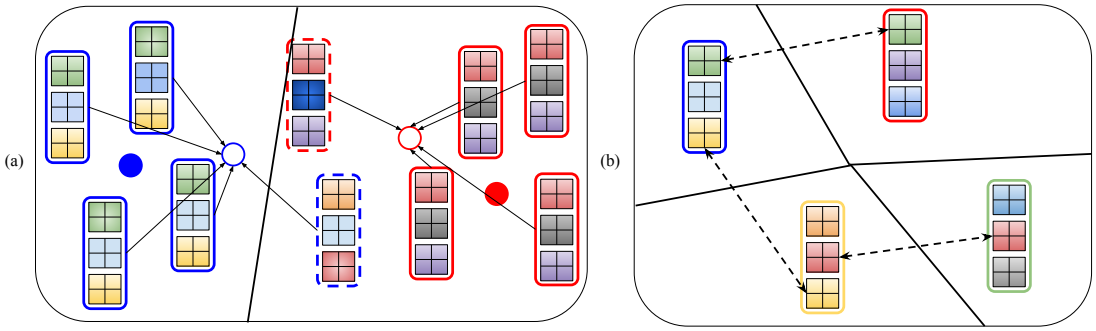


FIGURE 3.1: An illustration of how unrepresentative samples affect embedding learning. Each embedding (rounded rectangle) consists of three feature maps (coloured squares), with unrepresentative samples shown in dashed borders. (a) Binary classification with five training examples per class. We show the real class centres in the embedding space (solid circles) and the mean of each class' embeddings (hollow circle). (b) 4-class classification with one training example per class. Dashed arrows link similar feature maps in the embeddings from different classes.

overlapped classes, because there are a limited number of classes in the meta-training set and some of them are similar. For example, in a well-known data set for few-shot image classification, miniImageNet, there are only 100 classes of objects (Ravi and Larochelle, 2017) and some of them are different breeds of dogs. Thus, meta-learners could be trained to perform well on meta-training tasks and not generalise well on meta-testing tasks comprised of unseen classes.

Having the above motivations in mind, we propose L2AE-D (Learning to Aggregate Embeddings with Meta-level Dropout), a novel meta-learning approach for few-shot image classification that learns to aggregate embeddings with meta-level dropout. L2AE-D learns a CNNs based feature extractor and a channel-wise attention mechanism in an end-to-end manner. The feature extractor is used to transform the input images into discriminative embeddings. The channel-wise attention mechanism is learned to assign larger weights to useful feature maps and smaller weights to noisy ones of different embeddings within the same channel, in order to exploit the limited training embeddings as much as possible. We propose different learning strategies for one-shot and few-shot tasks, since one-shot tasks only include a single training sample per class. We also introduce a meta-level dropout technique into the meta-training process to prevent meta-level overfitting. We test this technique in several representative meta-learning approaches and it significantly improves their performance. We evaluate the proposed method on Omniglot (Lake et al., 2011)

and miniImageNet (Ravi and Larochelle, 2017) data sets, and it achieves competitive performance on various few-shot learning tasks.

This chapter is organised as follows. Section 3.2 introduces our L2AE-D methods for few-shot image classification. Section 3.3 presents the data sets we used and the experimental settings. The experimental results are analysed in Section 3.4. Finally, we summarise the achievement of the first stage of our research in Section 3.5.

3.2 Methodology

This section introduces the proposed L2AE-D strategy for few-shot image classification. First, we describe the L2AE meta-learning strategy in Section 3.2.1. Then, meta-level dropout is explained in Section 3.2.2.

3.2.1 The L2AE Module

L2AE can be divided into three modules: embedding module f_φ, attention module g_ϕ and distance module as shown in Figure 3.2 and 3.3. The attention module is different for the 1-shot and K-shot cases. Figure 3.2 shows our strategy for N-way 1-shot classification and Figure 3.3 depicts our strategy for N-way K-shot classification. Pseudocode for the training process of L2AE-D is provided in Algorithm 1.

- **Embedding module:** This module aims to extract features of each input image and transform it into embeddings. For each input example $x_{i,j}$ belonging to the i-th class, we feed it into the embedding module f_φ to generate an embedding $f_\varphi(x_{i,j}) = E_{i,j} = \left[e^1_{i,j}; e^2_{i,j}; ...; e^C_{i,j}\right]$, which comprises C feature maps $e^c_{i,j} \in R^{l \times l}$ with the size of $l \times l$. Then, the training embeddings are fed into the attention module.

- **Attention module:** This module is used for generating aggregation weights of the feature maps in a channel-wise manner as shown in Figure 3.4. Also, it is shared among different channels. We use two different strategies to do aggregation for 1-shot and K-shot tasks as shown in Figure 3.2 and 3.3, respectively. Generally, in each convolutional channel, we concatenate a set of corresponding feature maps for both strategies, then feed them into a CNNs-based attention weight generation module to obtain the aggregation weight for every feature map. The reason for choosing the concatenation operation to collect the feature maps is that we want

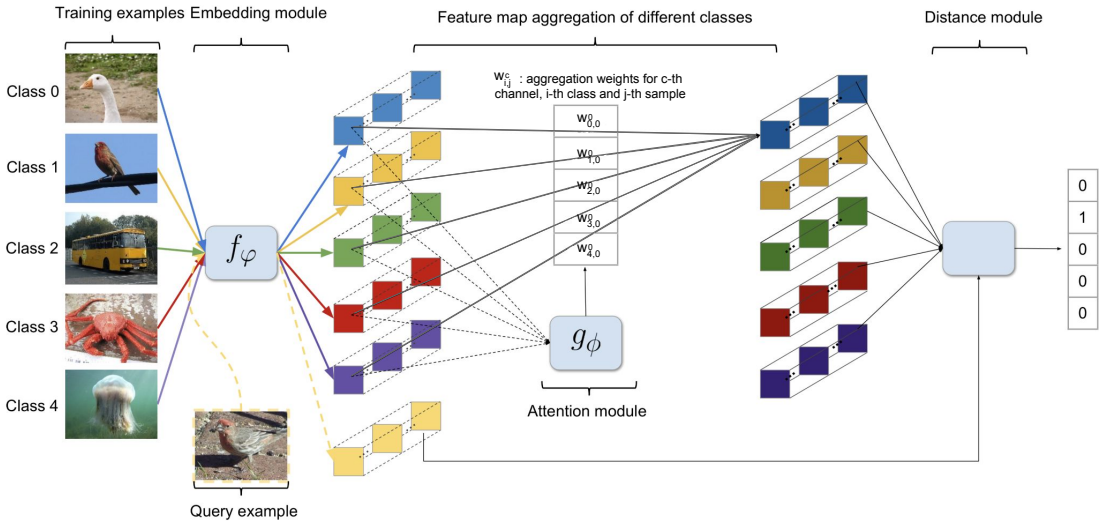


FIGURE 3.2: 5-way 1-shot classification with L2AE-D: (1) Training samples are transformed by f_φ into embeddings (set of feature maps shown in coloured squares, different colours represent different classes); (2) To strengthen the first feature map for class 0, we collect the first feature map of all classes and concatenate them in the channel dimension. Note that we put the feature map of class 0 in the first channel. Then we feed the concatenated 5-channel feature maps into g_ϕ to generate 5 aggregation weights; (3) The 5 feature maps are aggregated based on the generated weights to obtain the new feature map of class 0; (4) Step (2) and (3) are repeated for every feature map; (5) To make predictions, we feed a query into f_φ, then compare its embedding with the aggregated training embeddings in the distance module. This outputs a one-hot vector representing the predicted label of the query.

to apply CNNs to learn the relationships between different feature maps following (Sung et al., 2018). Therefore, we need to arrange the feature maps based on the concatenation in the channel dimension to serve as the input of the CNNs-based attention weight generation module. Compared to FC layers, CNNs are more efficient, therefore, we use CNNs as our attention weight generation module. Concretely, for N-way K-shot tasks, we aggregate the feature maps of K training embeddings in the same class to be the class-representative feature maps. For the c-th channel, we join the corresponding feature maps of K training embeddings in the i-th class as $F_i^c = \left[e_{i,1}^c; e_{i,2}^c; ...; e_{i,K}^c\right]$. For N-way 1-shot tasks, we aggregate the feature maps of N training embeddings from different classes, since there is only one training embedding in each class. In order to generate aggregation weights for the i-th class in the c-th channel, we concatenate the corresponding feature maps of N training embeddings

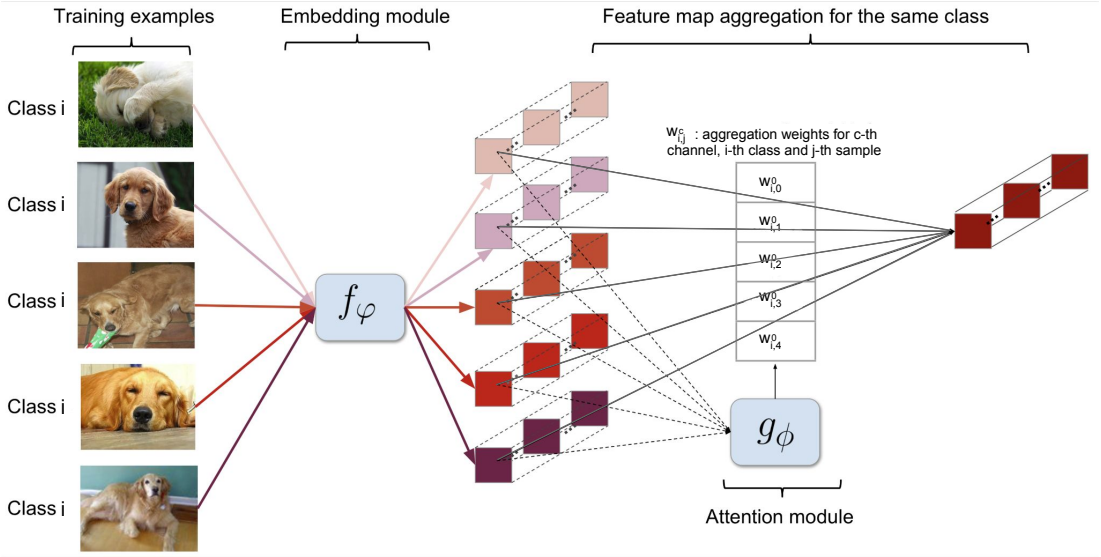


FIGURE 3.3: N-way 5-shot classification with our approach. L2AE-D aggregates embeddings for each class: (1) The training examples are transformed by f_φ into embeddings represented by a set of feature maps (the colours from light red to dark red represent different examples of class i); (2) For each channel, we collect the feature maps and feed them into the attention module; (3) The feature maps are concatenated in depth and fed into g_ϕ to generate aggregation weights; (4) The feature maps are then aggregated based on the generated weights to represent a feature for class i; (5) Step (2) to (4) are repeated for every channel and finally the aggregated representative embedding of class i can be obtained.

from different classes as $F_i^c = \left[e_{i,1}^c; e_{1,1}^c; ...; e_{N,1}^c\right]$. Note that, in this scenario, since we aim to refine each feature map of the i-th class, we always locate the feature map of the i-th class $e_{i,1}^c$ in the first channel as a key and the feature maps of other $N-1$ classes behind randomly in F_i^c as a set of queries. Then the CNNs-based attention mechanism would determine how similar each query and the key are and assign a weight to each query. Then we can aggregate the feature maps based on the attention weights to obtain the refined feature map of the i-th class.

Next, the concatenated feature maps are inputted into CNNs based attention networks g_ϕ, which produce the aggregation weights $w_i^c \in R^K$ for K-shot tasks or $w_i^c \in R^N$ for 1-shot tasks. After that, we can aggregate the feature maps F_i^c based on the weights w_i^c. The aggregated feature map of the i-th class in the c-th channel is calculated as $\widetilde{e}_i^c = \sum w_i^c \cdot F_i^c$, $\widetilde{e}_i^c \in R^{l \times l}$. In the end, we concatenate the aggregated feature maps in all channels and obtain a new embedding $\widetilde{E}_i = \left[\widetilde{e}_i^1; \widetilde{e}_i^2; ...; \widetilde{e}_i^c\right]$ for the i-th class.

The new training embedding set is then represented by $\widetilde{E}_{train} = \left\{\widetilde{E}_i\right\}_{i=1}^{N}$, in which $\widetilde{E}_i$ can be seen as a class representative.

- **Distance module:** This module is used to measure the distance between the embeddings of query examples, $E_q = f_\varphi\left(x_q\right)$, and the aggregated embeddings $\widetilde{E}_{train}$. Following (Snell, Swersky, and Zemel, 2017), we choose the Euclidean distance as the distance function $d() : R^M \times R^M \rightarrow [0, +\infty)$. Thus, the distance between E_q and $\widetilde{E}_i$ is computed by $d(E_q, \widetilde{E}_i)$.

- **Loss function:** We consider cross-entropy loss to train our model. First, the softmax function is applied over the negative distance between the query embeddings and aggregated training embeddings as follows:

$$p_{\varphi,\phi}(y = y_i \mid \widetilde{E}_{train}, x_q) = \frac{exp(-d(E_q, \widetilde{E}_i))}{\sum_{i'} exp(-d(E_q, \widetilde{E}_{i'}))} \tag{3.1}$$

Then the loss function can be formulated as

$$L(p_{\varphi,\phi}(y = y_q \mid \widetilde{E}_{train}, x_q)) = -\frac{1}{N_q}\sum_{q=1}^{N_q} log\, p_{\varphi,\phi}(y = y_q \mid x_q) \tag{3.2}$$

where N_q is the number of query examples in each training epoch.

3.2.2 Meta-Level Dropout

Most meta-learning approaches use multiple layers of CNNs to extract features for few-shot image classification. As discussed before, we incorporate the dropout technique in the meta-level to tackle meta-level overfitting. Specifically, we randomly drop part of units of CNNs for both the training and testing examples in each few-shot learning task during meta-training. During meta-testing, we use the whole trained CNNs to extract features on both training and testing examples. Note that the dropout in the convolutional layers works in a different way from that in the fully connected layers, because the kernel weights are shared with the units at different spatial positions, so that, the weights would still be updated by backpropagation even if part of units are dropped. The actual effect of performing dropout in the convolutional layers is to scale the learning rate (Tompson et al., 2015), which can also help with preventing overfitting. We find this technique can improve several meta-learning approaches significantly according to the experimental results in Section 3.4.

Algorithm 1 The training of L2AE-D

Require: Meta-training set $\mathfrak{D}_{meta-train}$
Require: The number of classes N and the number of training exampels K in each class, the number of the channels C in the last convolutional layer of the Embedding module, the metric-based classifier clf according to Equation 3.1, the loss function $\mathcal{L}$ based on Equation 3.2

randomly initialize φ and ϕ
for each episode **do**
  $D_{train}, D_{test} \leftarrow$ randomly sample from $\mathfrak{D}_{meta-train}$, $D_{train} = \{(x_{i,j}, y_i) | 1 \leq i \leq N, 1 \leq j \leq K\}$, $D_{test} = \{(x_q, y_q)\}_{i=1}^{N_q}$
  for $i = 1$ **to** N **do**
    for $j = 1$ **to** K **do**
      $E_{i,j} = f_\varphi(x_{i,j})$, $x_{i,j}$ from D_{train}, $E_{i,j} = \left[e_{i,j}^1; e_{i,j}^2; ...; e_{i,j}^C\right]$
    end for
  end for
  for $i = 1$ **to** N **do**
    for $c = 1$ **to** C **do**
      if $K == 1$ **then**
        $F_i^c = \left[e_{i,1}^c; e_{1,1}^c; ...; e_{N,1}^c\right]$
      else
        $F_i^c = \left[e_{i,1}^c; e_{i,2}^c; ...; e_{i,K}^c\right]$
      end if
      $w_i^c = g_\phi(F_i^c)$
      $\widetilde{e}_i^c = \sum w_i^c \cdot F_i^c$
    end for
    $\widetilde{E}_i = [\widetilde{e}_i^1, \widetilde{e}_i^2, ..., \widetilde{e}_i^C]$
  end for
  $\widetilde{E}_{train} = \left\{\widetilde{E}_1, \widetilde{E}_2, ..., \widetilde{E}_N\right\}$
  for $q = 1$ **to** N_q **do**
    $E_q = f_\varphi(x_q), x_q \in D_{test}$
  end for
  $\mathcal{L}_{test} = \sum_{q=1}^{N_q} \mathcal{L}(clf(\widetilde{E}_{train}, E_q), y_q)$
  Update φ and ϕ based on $\bigtriangledown_{\varphi,\phi} \mathcal{L}_{test}$
end for

3.3 Experiments

This section introduces the details of the two used data sets and the experimental configurations followed to evaluate the behaviour of L2AE-D against the state-of-the-art. First, we introduce two widely used data sets for few-shot image classification in Section 3.3.1. Then, the network architecture of our L2AE-D model is illustrated in Section 3.3.2. Afterwards, the experimental setting and evaluation are presented in Section 3.3.3.

3.3.1 Data Sets

We evaluate our methods on two widely studied few-shot learning data sets, Omniglot and miniImageNet, which are introduced in Section 2.4.2.

3.3.2 Comparison Algorithms and Network Architecture

In this chapter, we choose 4Conv as our embedding module for two reasons. First, Res12 employs global average pooling (He et al., 2016) in the end to summarise the extracted embedding as a vector, which is not fully compatible with our L2AE method, since our method takes feature maps as the input of our aggregation module. Second, the few-shot learning community focuses on the effectiveness of the meta-learning strategy rather than the complexity of networks. Therefore, we evaluate our L2AE-D method based on 4Conv backbone. The detailed architecture of our L2AE-D method based on 4Conv is described as follows:

- **4Conv** has been described in Section 2.4.5. Specifically, we use a 2×2 max-pooling to conduct the downsampling operation. Besides, we employ a dropblock (Ghiasi, Lin, and Le, 2018) layer after each convolutional block to reduce meta-level over-fitting. The keep rate and block size are set as 0.85 and 3 respectively for all 4 layers on miniImageNet and Omniglot. For the distance metric, we choose scaled Euclidean distance following (Ye et al., 2020), the scale is set 64. For Omniglot, due to the small size of the input images, we feed the embeddings into the CNNs based attention module and we remove the max-pooling layer from the last convolutional block.
- **Attention module.** The architecture of our attention module is showed in 3.4 consisting of 2 convolutional blocks and a fully connected (FC)

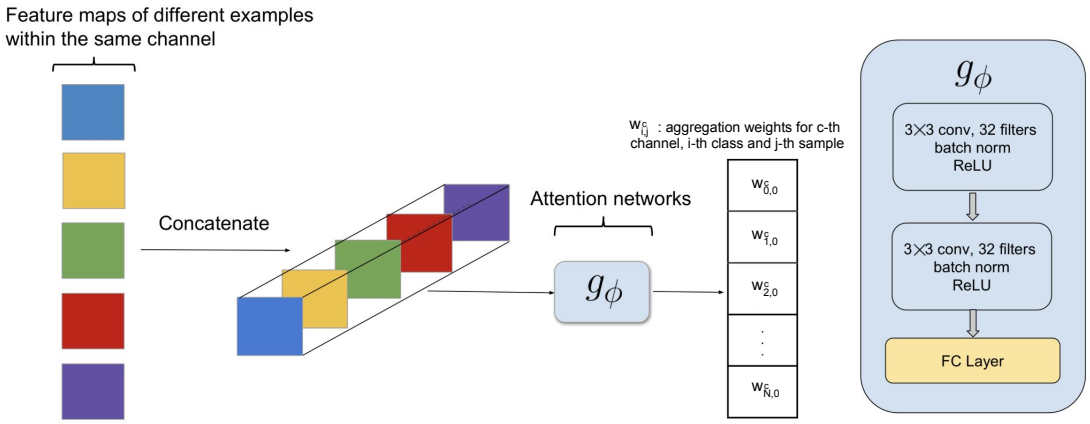


FIGURE 3.4: The architecture of the attention module. The illustrated aggregation weights are for 1-shot tasks and those for 5-shot tasks are the same as shown in Figure 3.3. The squares in different colours represent the feature maps of different examples. The attention networks consist of two convolutional blocks and a FC layer.

layer. Each convolutional block in this module comprises a 3×3 convolution with 32 filters, followed by batch normalisation, a ReLU nonlinearity. The FC layer results in a C-dimensional output, which represent the aggregation weights for the C feature maps. For K-shot tasks, we use the softmax function after the FC layer since we aim to assign positive weights, whose sum is 1, to the C embeddings of the same class. Note that we tune the aforementioned hyper-parameters based their performance on meta-validation set.

On Omniglot data set, we compare our approach against promising methods from each family of few-shot learning approaches that provide experimental results. They are MAML (Finn, Abbeel, and Levine, 2017) from fast-parametrisation based approaches, Neural Statistician (Edwards and Storkey, 2017) and MetaGAN (Zhang et al., 2018b) from generative model based approaches, and Siamese Nets (Koch, Zemel, and Salakhutdinov, 2015), Matching Nets (Vinyals et al., 2016), ProtoNets (Snell, Swersky, and Zemel, 2017), GNN (Garcia and Bruna, 2018) and RN (Sung et al., 2018) as metric learning approaches. On miniImageNet, we compare our method with prior approaches that are based on the same 4Conv backbone for a fair comparison. As before, we choose promising methods from each family that provide experimental results on miniImageNet. They are Meta-learner-LSTM (Ravi and Larochelle, 2017), MAML (Finn, Abbeel, and Levine, 2017) and Activation2Weights (Qiao et al., 2018) from fast-parameterisation based approaches, MetaGAN (Zhang et

al., 2018b) from generative model based approaches, Matching Nets (Vinyals et al., 2016), ProtoNets (Snell, Swersky, and Zemel, 2017), GNN (Garcia and Bruna, 2018), RN (Sung et al., 2018) and TPN (Liu et al., 2019b) from metric learning approaches.

3.3.3 Experimental Setting and Evaluation

To allow for fair comparisons with the current state-of-the-art, we maintain the different experimental setups reported on Omniglot (20-way 5-shot, 20-way 1-shot, 5-way 5-shot, 5-way 1-shot) and miniImageNet (5-way 5-shot, 5-way 1-shot). Similar to most approaches based on 4Conv (Snell, Swersky, and Zemel, 2017; Finn, Abbeel, and Levine, 2017; Liu et al., 2019b), we train our 4Conv based approach in an episodic manner for a fair comparison. All experiments are performed using TensorFlow (Abadi et al., 2016) on a Titan Xp GPU.

- **Episode-based meta-training** has been introduced in Section 2.4.3. Concretely, we use a meta-batch size of 3, which means in each episode we randomly sample 3 N-way K-shot classification tasks to train the model. For each few-shot task, besides the $N \times K$ training examples, we randomly sample 6 query examples per class to compute the loss for Omniglot and miniImageNet. We train our model with Adam (Kingma and Ba, 2015) with an initial learning rate of 0.001 in an end-to-end manner (Snell, Swersky, and Zemel, 2017; Sung et al., 2018). We train the model for 200,000 episodes and cut the learning rate in half every 40,000 episodes to stabilise training, and use the meta-validation set to choose the best-performing model for meta-testing. It is noteworthy that existing methods conduct BN in different ways. As pointed out in (Ravi and Larochelle, 2017), there would be a bad impact on performance if we use the global BN statistics accumulated from the meta-training set to normalise batches of examples in the meta-testing set, since there is no overlap between the classes in these two sets. Thus, we perform BN on each batch of examples following (Finn, Abbeel, and Levine, 2017; Sung et al., 2018). Specifically, for each task during both meta-training and meta-testing, we use each batch's statistics to normalise the training or query examples.

- **Meta-testing:** During meta-testing, the previous works mostly evaluated their methods on 600 or 1,000 randomly sampled N-way K-shot classification tasks from the meta-testing set. Since there are a large number of different tasks in meta-testing, they may sample a large proportion of

easy-to-classify or difficult-to-classify tasks using different seeds, which would lead to a result with high variance. To get a more reliable result, we evaluate our approach on 6,000 randomly sampled tasks for all data sets. The average classification accuracy and 95% confidence interval are reported.

3.4 Analysis of Results

In this section, we analyse the results obtained from different experimental studies. Concretely, our goals are:

- To check the superiority of our L2AE-D for few-shot image classification on two wide used data sets comparing with the state-of-the-art (Section 3.4.1 and 3.4.2).
- To analyse the benefits of the meta-level dropout technique when it is incorporated into several existing representative approaches (Section 3.4.3)
- To clearly show how our L2AE-D works and how our L2AE method influence the representation learning for few-shot image classification (Section 3.4.4).

3.4.1 Analysis of the Results on Omniglot

The reported experimental results of selected comparison algorithms and ours are shown in Table 3.1. In general, all the methods perform worse on 20-way tasks than 5-way tasks, which shows 20-way tasks are more difficult. L2AE-D achieves state-of-the-art performance on 20-way tasks and competitive results on 5-way tasks. Besides, our results are more reliable, since we test on more few-shot learning tasks and the 95% confidence interval is much less than other methods. MetaGAN performs better on 5-way tasks by generating more examples to assist RN while it improves marginally upon RN.

3.4.2 Analysis of the Results on miniImageNet

The experimental results of our method and the comparison with the selected existing methods on miniImageNet are shown in Table 3.2. L2AE-D achieves state-of-the-art performance on 5-way 5-shot classification. On 5-way 1-shot classification, L2AE-D provides the second best result, which is slightly worse than Activations2Weights. However, the feature extractor of Activations2Weights

TABLE 3.1: Few-shot classification results on Omniglot averaged over 6,000 testing tasks. It is noteworthy we only compare our method with prior approaches that are based on the same type of model, 4-layer CNNs. The ± shows 95% confidence over tasks. The best and second best performing results are highlighted in bold and underlined, respectively. All the results are rounded to 1 decimal place other than MetaGAN's that are reported with 2 decimal places.

Method	**5-way Acc.**		**20-way Acc.**	
	1-shot(%)	5-shot(%)	1-shot(%)	5-shot(%)
Siamese Nets	96.7	98.4	88.0	96.5
Matching Nets	98.1	98.9	93.8	98.5
Neural Statistician	98.1	99.5	93.2	98.1
ProtoNets	98.8	99.7	96.0	98.9
GNN	99.2	99.7	97.4	99.0
MAML	98.7±0.4	**99.9±0.1**	95.8±0.3	98.9±0.2
RN	99.6±0.2	99.8±0.1	97.6±0.2	99.1±0.1
MetaGAN + RN	**99.67±0.18**	99.86±0.11	97.64±0.17	**99.21±0.1**
L2AE-D	99.3±0.05	99.8±0.02	**97.8±0.05**	99.2±0.02

is trained with more classes (higher ways) and more queries in each meta-training episode. In contrast, our model is trained on 5-way classification with 15 queries per episode, which is consistent with the setting of most existing approaches. Besides, TPN obtains very competitive results on 1-shot and 5-shot classification. However, TPN is a transductive method that requires unlabelled data to propagate labels and its performance is affected by the number of query examples. Even though we use query batch statistics to normalise the query examples in a transductive way, we can simply modify it into an inductive way by using training batch statistics to normalise the query data without decreasing the performance much.

3.4.3 Analysis of the Effect of Meta-Level Dropout

Since the augmented Omniglot data set includes much more classes (4,800) than miniImageNet (64) in the meta-training set, the meta-learners do not suffer much from meta-level overfitting on Omniglot. Therefore, we focus on miniImagent to analyse the effect of meta-level dropout through 5-way 1-shot tasks. We introduce meta-level dropout into several representative meta-learning approaches, including MAML (Finn, Abbeel, and Levine, 2017), ProtoNets (Snell, Swersky, and Zemel, 2017) and RN (Sung et al., 2018). Specifically, we use their provided code and add an advanced dropout technique,

TABLE 3.2: Few-shot classification results on miniImageNet averaged over 6,000 tests based on 4-layer CNNs. It is noteworthy we only compare our method with prior approaches that are based on the same type of model, 4-layer CNNs. The ± shows 95% confidence over tasks. The best and second best performing results are highlighted in bold and underlined, respectively.

Model	**5-way Acc.**	
	1-shot	5-shot
Matching Nets	43.56 ± 0.84%	55.31 ± 0.73%
Meta-Learner-LSTM	43.44 ± 0.77%	60.60 ± 0.71%
MAML (1 query)	48.70 ± 1.84%	63.11 ± 0.92%
ProtoNets	49.42 ± 0.78%	68.20 ± 0.66%
GNN	50.33 ± 0.36%	66.41 ± 0.63%
RN	50.44 ± 0.82%	65.32 ± 0.70%
MetaGAN +RN	52.71 ± 0.64%	68.63 ± 0.67%
TPN	53.75 ± 0.86%	<u>69.43 ± 0.67%</u>
Activations2Weights	**54.53 ± 0.40%**	67.87 ± 0.70%
L2AE-D	<u>53.85 ± 0.26%</u>	**70.16 ± 0.19%**

dropblock (Ghiasi, Lin, and Le, 2018), in each convolutional layers before max-pooling with the keep rate of 0.85 and block size of 3, respectively. We compare the results with dropout to the reported ones of the chosen methods except for MAML, since it tests on 1 query per class using 32 filters in CNNs. We modified their setting to use 64 filters and test on 15 queries per class in order to be consistent with the settings of other methods. We evaluate these methods on 5-way 1-shot classification in the same way as Section 3.4.2. The experimental results in Table 3.3 show that adding meta-level dropout can significantly improve several promising meta-learning approaches, as well as ours. It can also be seen that, even without dropout, L2AE also outperforms those representative few-shot learning approaches. Since the proposed L2AE algorithm improves upon ProtoNets, Table 3.3 presents an ablation analysis. The only difference between ProtoNets and L2AE is that L2AE adds our proposed attention based aggregation module upon ProtoNets. Both of our results with and without dropout outperform ProtoNets by around 2%, which demonstrates that adding our channel-wise attention based aggregation module is effective for few-shot learning.

3.4.4 Visualisation of the Working of L2AE-D

To further show how our approach works, we visualise the aggregated embeddings for the unseen few-shot classification tasks in the meta-testing set

TABLE 3.3: Few-shot classification results on miniImageNet with or without dropout averaged over 6,000 testing tasks. The ± shows 95% confidence over tasks. * denotes MAML uses 64 filters and tests on 15 queries per class.

Model	**5-way 1-shot Acc.**	
	without dropout	with dropout
MAML*	47.71 ± 0.84%	50.43 ± 0.87%
ProtoNets	49.42 ± 0.78%	52.08 ± 0.81%
RN	50.44 ± 0.82%	52.40 ± 0.85%
L2AE	51.55 ± 0.25%	53.85 ± 0.26% (**L2AE-D**)

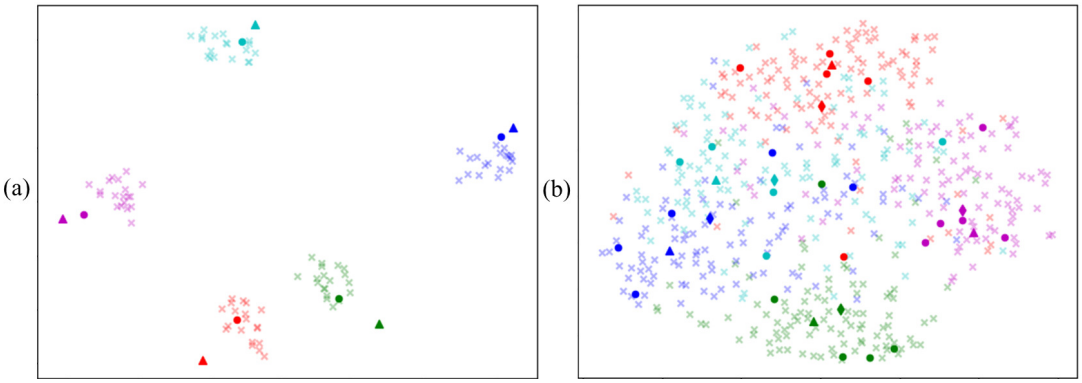


FIGURE 3.5: t-SNE visualisation of the aggregated embeddings of unseen classes for a 5-way 1-shot classification task on Omniglot (a) and a 5-way 5-shot task on miniImagenet (b). The embeddings of training samples are shown as points. Aggregated embeddings are shown as triangles. The embeddings of regular examples are shown as crosses. The Means of training embeddings are shown as diamonds.

based on t-SNE (Maaten and Hinton, 2008). t-SNE is a technique for dimensionality reduction that is particularly well suited for the visualisation of high-dimensional data sets (Maaten and Hinton, 2008). Figure 3.5 (a) shows the visualisation of the aggregated embeddings for an unseen 5-way 1-shot classification task on Omniglot. The embeddings aggregated from different classes tend to move away from their own cluster and be farther from the clusters of other classes.

Figure 3.5 (b) shows the visualisation of the aggregated embeddings for an unseen 5-way 5-shot classification task on miniImagenet. Compared to the embeddings of Omniglot in Figure 3.5 (a), we can see that the embeddings of miniImagenet are much messier and consist of more unrepresentative examples. This indicates the difficulty of few-shot learning on miniImagenet and the necessity to reduce the impact of outliers for a method. When there are unrepresentative examples in the training set, such as the outliers (the bottom

red point) and the example located near the boundary of a cluster (the rightmost blue point), the mean of training embeddings (Snell, Swersky, and Zemel, 2017) deviates from a good position that represents a class in the embedding space. However, our aggregated embeddings stick to a representative position in the embedding space and are much more stable regardless of unrepresentative examples, which can lead to a more robust decision boundary.

3.5 Summary

In this chapter, we have achieved our first research objective by proposing a novel meta-learning approach for aggregating useful convolutional features and suppressing noisy ones based on a channel-wise attention mechanism. We have proposed two different learning strategies for one-shot and few-shot tasks aiming to fully and effectively use the few training examples. Our model does not require any fine-tuning and can be trained in an end-to-end manner. In addition, we have tackled the problem of meta-level overfitting by introducing a meta-level dropout technique. This technique has significantly improved several well-known meta-learning approaches as well as ours. Furthermore, we have achieved competitive performance over a few N-way K-shot classification tasks on both Omniglot and miniImageNet data sets, which has demonstrated the effectiveness and competitiveness of our method. Note that we only used two data sets to evaluate our approach, because other benchmarks mentioned in Section 2.4.2 are not proposed or widely used at the time of proposing our method. In Section 5.5 where we illustrate the overall contribution of this thesis, the results of our L2AE-D for few-shot image classification on two other benchmarks, including larger and fine-grained data sets, are shown. Those can further support our conclusion that the L2AE-D are effective for few-shot image classification on various types of data sets.

To tackle the core issues of few-shot image classification, besides paying attention to the limited training samples in few-shot image classification, we should also guarantee as many useful features as possible can be extracted during a feature extraction process. Therefore, in the next chapter, we plan to address the two key problems of few-shot image classification from the perspective of feature extraction.

Chapter 4

Spatial Attention-based Adaptive Pooling for Few-Shot Image Classification

4.1 Introduction

In the first stage of our research, we performed channel-wise attention to suppress unrepresentative training samples and meanwhile leveraged as much useful information as possible in each final convolutional channel. The proposed solution assumed that the few training embeddings have included sufficient useful information for the subsequent similarity comparisons and focused on how to maximising the use of limited training embeddings. However, we did not take into consideration that the assumption cannot always be guaranteed, because a feature extraction process may lose useful information. Therefore, we believe there is more work that can be done to address the two key issues of few-shot image classification from a different perspective.

A general meta-learning approach for few-shot image classification normally starts from transforming the few training images into discriminative features, then makes predictions based on a distance metric or fully connected layer based classifier (Snell, Swersky, and Zemel, 2017; Finn, Abbeel, and Levine, 2017). To overcome the lack of information, besides paying attention to limited training data, we should also make sure as much as useful features can be extracted during a feature extraction process. In addition, a good feature extractor should also be able to distinguish the foreground and background of an image. To this end, the second stage of our research aims to tackle the two core issues of few-shot image classification from the point of view of feature extraction.

To extract features for few-shot image classification, existing meta-learning approaches mostly employ a regular CNNs block, usually composed of a convolution operation, batch normalisation, an activation function and a downsampling operation. For the downsampling operation, existing methods mainly use the max-pooling or average-pooling and ignore the fact that these pooling methods are essentially lossy processes. They may lose relevant features (max-pooling) or mix up relevant and irrelevant features (average-pooling) during feature extraction. Since the lack of information is one of the key issue of few-shot image classification, a tailored pooling method that can save as much useful features as possible for few-shot image classification is needed. In addition, the commonly used pooling techniques perform pooling independently in different convolutional channels and ignore the correlation between the features at the same spatial location in different channels, which may lose the information of spatial importance. Therefore, a more appropriate downsampling method in a CNNs based feature extraction process is needed to avoid losing useful information and mitigate the intrinsic uncertainties for few-shot image classification.

To address the above issues, we design a spatial attention based adaptive pooling (Ada-P) module to replace the conventional pooling methods for few-shot learning, in which a learnable pooling weight generation block is trained to assign different pooling weights to the features at different spatial locations for each individual embedding. The module performs weighted pooling by taking into account the importance of the features at different spatial locations, which can pay more attention to the salient regions and avoid discarding useful information. Since the sizes of receptive fields in different convolutional layers vary, we learn a specific pooling module for each convolutional layer. To consider the correlations between channels, we use CNNs as a meta-learner to assign pooling weights. Different from regular CNNs, our CNNs-based meta-learner is lightweight (only including one output channel) and can be incorporated into different CNNs-based few-shot learning approaches as a plug-and-play module.

Our Ada-P module is related to a few spatial attention methods introduced in Section 2.3.2 (Woo et al., 2018; Chen et al., 2017; Wang et al., 2017; Jetley et al., 2018; Meng et al., 2019). These methods applied different strategies, such as using CNNs or multilayer perceptron (Woo et al., 2018; Chen et al., 2017; Wang et al., 2017; Meng et al., 2019), performing downsampling, upsampling

operations and residual connections (Wang et al., 2017), measure the compatibility between the final global feature and the features in the intermediate layers (Jetley et al., 2018), on embeddings to assign weights to the features at different spatial locations. They performed attention after each convolutional block while our adaptive pooling module is incorporated in each convolutional block serving as a pooling operator. They mostly introduced much more trainable parameters while our Ada-P module is lightweight containing only one convolutional kernel. Their aim is to refine embeddings whereas our target is to find an appropriate way to downsample embeddings while preserving useful information. In addition, these works all tackled standard learning tasks with sufficient training data, while our focus is on few-shot learning problems. In the experiments, we compare Ada-P with these related spatial attention methods on few-shot image classification.

As in the last chapter, the experiments are completed using the widely evaluated miniImageNet data set. Note that, in this chapter, we discard the Omniglot data set, since it only contains grayscale images of handwritten characters without background clutters. Our spatial attention based Ada-P module can pay more attention to the salient regions and discard background noise, therefore, it is not appropriate to evaluate our method on the data set without any background clutter, such as the Omniglot data set. Additionally, here we introduce two more challenging few-shot learning benchmarks, tieredImageNet (Ren et al., 2018) and CUB (Wah et al., 2011). The meta-training and meta-testing set of tieredImageNet are less similar in the semantic space, therefore, it is more challenging and realistic compared to miniImageNet. The CUB data set results in fine-grained few-shot image classification tasks, which are challenging, because categories can only be distinguished by subtle or local differences. On the above three data set, our method achieves competitive performance on various few-shot learning tasks.

This chapter is structured as follows. Section 4.2 describes the proposed Ada-P module for few-shot image classification. Section 4.3 presents the data sets we used and the experimental settings. Section 4.4 analyse the experimental results. In the end, we summarise the achievement of the second stage of our research in Section 4.5.

4.2 Methodology

This section introduces the proposed adaptive pooling module. First, the details of our Ada-P module is provided in Section 4.2.1. Then, we describe a whole meta-learning pipeline based on our Ada-P module in Section 4.2.2.

4.2.1 Adaptive Pooling Module

The Ada-P module can be placed in every convolutional layer during feature extraction as shown in Figure 4.1. The pooling weight generation block learns to adaptively generate feature fusing weights for each embedding. Then local features in a sliding window can be fused based on the generated weights by taking into account the spatial importance. Let $E \in R^{W\times H\times C}$ be an embedding (a set of feature maps) serving as the input of a convolutional layer, where W, H, C represent the width, height of feature maps and the number of channels, respectively. As shown in Figure 4.1, after a standard convolutional operation followed by BN and an activation function, we obtain the embedding $\widetilde{E} = Conv(E)$, $\widetilde{E} \in R^{W\times H\times C}$. Then, the embedding $\widetilde{E}$ is fed into the pooling weight generation block $g_\psi(\cdot) : R^{W\times H\times C} \rightarrow R^{W\times H\times 1}$ to generate the adaptive pooling weights for the features at different spatial locations, which is displayed as the square in gradient colour in Figure 4.1:

$$w = g_\psi(\widetilde{E}), w \in R^{W\times H\times 1} \tag{4.1}$$

Specifically, our pooling weight generation block is represented by a convolutional layer with a single convolutional kernel, which is much more lightweight compared to the convolutional operation in the feature extractor. We choose convolutions to be the meta-learner because it provides a larger receptive field and considers the features in all channels at the same spatial location. Furthermore, g_ψ also includes a BN layer followed by a sigmoid function. We use a sigmoid function to limit the pooling weight values between 0 and 1.

Then, weighted pooling is conducted over the embedding $\widetilde{E}$ to fuse local features. The generated pooling weights w are shared among different channels. Therefore, for each pooling sub-region Ω_k whose size is determined by the pooling window size, the weighted pooling can be computed as:

$$\widetilde{E}_{pooled}[k] = \frac{1}{|\Omega_k|}\sum_{i\in\Omega_k} w[i]\widetilde{E}[i] \tag{4.2}$$

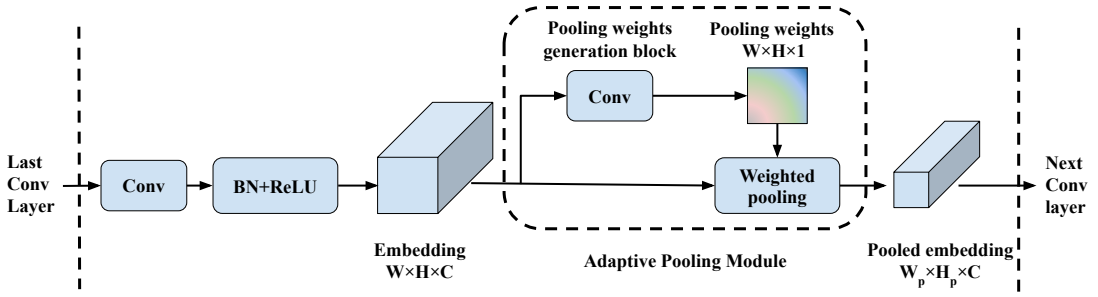


FIGURE 4.1: Workflow of the proposed Ada-P module. Conv represents convolutions. BN stands for batch normalisation (Ioffe and Szegedy, 2015). In the embedding space, a pooling weights generation block is learned to assign a specific pooling weight map ($R^{W\times H\times 1}$) for each embedding ($R^{W\times H\times C}$); then weighted pooling is conducted based on the generated pooling weights per channel.

where $\widetilde{E}_{pooled}[k] \in R^{1\times 1\times C}$ is the pooled embedding for sub-region Ω_k, $w[i] \in R^{1\times 1\times 1}$ and $\widetilde{E}[i] \in R^{1\times 1\times C}$ are the generated pooling weight and the embedding at $i-th$ location in sub-region Ω_k. After performing weighted average pooling for all the sub-regions, the fused embedding $\widetilde{E}_{pooled}$ will feed into the next convolutional layer. Pseudocode for our Ada-P module is provided in Algorithm 2.

Algorithm 2 Ada-P module

Input: Embedding $\widetilde{E} \in R^{W\times H\times C}$, pooling region $\Omega = W \times H$, pooling weights generation block g_ψ, window size t, strides s, padding p

Output: $\widetilde{E}_{pooled}$

1: $w = g_\psi(\widetilde{E}), w \in R^{W\times H\times 1}$ //Obtain adaptive pooling weights
2: **for** Ω_k in Ω **do**
3: $\widetilde{E}_{pooled}[k] = \frac{1}{|\Omega_k|}\sum_{i\in\Omega_k} w[i]\widetilde{E}[i]$, $\widetilde{E}_{pooled}[k] \in R^{1\times 1\times C}$ //Weighted average pooling for each sub-region
4: **end for**

4.2.2 Meta-Learning Pipeline with Ada-P Module

The Ada-P module is light-weight and can be employed as a plug-and-play module for various types of meta-learning approaches for few-shot image classification, such as fast parameterisation and metric learning based approaches. In this section, we choose ProtoNet (Snell, Swersky, and Zemel, 2017) as an

example to illustrate how our Ada-P module work with a meta-learning approach. We choose ProtoNet as our backbone for several reasons. First, compared to fast parameterisation and data generation based approaches, ProtoNet is computationally efficient. It does not require any inner optimisation or data generation process. Second, ProtoNet is simple and effective and has been used as a backbone for a few advanced few-shot learning approaches (Wang et al., 2018b; Sung et al., 2018; Song et al., 2021). This makes it a fair to use as a backbone when comparing with other state-of-the-art methods. Third, from our experiments, ProtoNet can always achieve a relatively good and stable performance on different data sets. The meta-learning pipeline includes the following components: an embedding module, a distance module, a loss function and an inference mechanism. The whole training procedure of our few-shot learning pipeline is presented in Algorithm 3.

- **Embedding module:** The embedding module (feature extractor) $f_{\{\phi,\psi\}}$ aims to transform a sample into a high-level representation, which consists of two kinds of building blocks. One is convolutional block $Conv_{\phi}$ comprised by a convolution operation, a BN layer and an activation function, ϕ represents the trainable parameters. The other is our Ada-P module Ada-P$_{\psi}$, ψ stands for the trainable parameters. These two types of building blocks are connected one after another as shown in Figure 4.1, and a number of them are stacked in the embedding module.

  Given a training set $D_{train} = \{(x_{i,j}, y_i)|1 \leq i \leq N, 1 \leq j \leq K\}$, a testing set $D_{test} = \{(x_q, y_q)\}_{q=1}^{N_q}$, the feature extractor $f_{\{\phi,\psi\}}$ transforms a training example $x_{i,j}$ and a test example x_q into an embedding $E_{i,j} = f_{\{\phi,\psi\}}(x_{i,j})$ and $E_q = f_{\{\phi,\psi\}}(x_q)$. Following the idea of ProtoNet, the prototype of each class is the mean embedding of the training examples belonging to its class:

$$P_i = \frac{1}{K}\sum_{j=1}^{K} f_{\{\phi,\psi\}}(x_{i,j}), P = \{P_i\}_{i=1}^{N} \tag{4.3}$$

- **Distance module:** This module applies a distance metric $d(\cdot)$ to measure the similarity between the embedding of a test example $f_{\{\phi,\psi\}}(x_q)$ and the prototype of each class P_i as $d(P_i, f_{\{\phi,\psi\}}(x_q))$. The commonly used distance metrics are Euclidean distance, cosine distance, scaled Euclidean or cosine distance or a learned distance metric. In our approach, we

use scaled Euclidean and cosine distance for shallow and deeper CNNs-based feature extractors, respectively. These settings are widely used in previous works (Oreshkin, López, and Lacoste, 2018; Gidaris and Komodakis, 2018).

- **Loss function:** We choose cross-entropy loss to train each task. First, the softmax function is applied over the negative distance between the test embeddings and the prototypes as follows:

$$p_{\{\phi,\psi\}}(y = y_i \mid P, x_q) = \frac{\exp(-d(P_i, f_{\{\phi,\psi\}}(x_q)))}{\sum_{i'} \exp(-d(P_{i'}, f_{\{\phi,\psi\}}(x_q)))} \tag{4.4}$$

Then, the loss function can be formulated as

$$L(p_{\{\phi,\psi\}}(y = y_q \mid P, x_q)) = -\frac{1}{N_q} \sum_{q=1}^{N_q} \log\, p_{\{\phi,\psi\}}(y = y_q \mid x_q) \tag{4.5}$$

where N_q is the number of test examples in each training task, y_q is the true label of x_q.

- **Inference:** The inference is conducted on the meta-testing set $\mathfrak{D}_{meta-test}$, whose label space has no overlapping with the meta-training set $\mathfrak{D}_{meta-train}$. The procedure is nearly the same as the meta-training phase through the trained feature extractor, distance module and a softmax function. Then, a test example can be classified into one class by taking its highest probability:

$$\overline{y}_q = \underset{i}{\operatorname{argmax}}\, p_{\{\phi,\psi\}}(y = y_i \mid P, x_q) \tag{4.6}$$

where x_q is a test example in a meta-testing task and $\overline{y}_q$ is its predicted label.

4.3 Experiments

This section introduces the details of the three used data sets and the experimental settings. First, we introduce three widely used data sets for few-shot image classification in Section 4.3.1. Then, the network architecture of our L2AE-D model is illustrated and selected comparison methods are presented in Section 4.3.2. Afterwards, the experimental setting and evaluation are provided in Section 4.3.3.

Algorithm 3 The training procedure of our few-shot learning pipeline

Input: Meta-training set $\mathfrak{D}_{meta-train}$

Input: The number of classes N, the number of training exampels K and the number of test samples N_q in each task, the number of convolutional layers M, the ProtoNet-based classifier clf based on Equation 4.3 and 4.4, the loss function L according to Equation 4.5.

randomly initialize φ and ϕ
for each episode **do**
  Randomly sample a batch of *Tasks* from $\mathfrak{D}_{meta-train}$
  $\mathcal{L} = 0$ //Initialise loss
  for all $Task_k$ in *Tasks* **do**
    Randomly sample D^k_{train}, D^k_{test} for $Task_k$, $D^k_{train} = \{(x_{i,j}, y_i) | 1 \leq i \leq N, 1 \leq j \leq K\}$, $D^k_{test} = \{(x_q, y_q)\}_{q=1}^{N_q}$
    for all $x_{i,j}$ in D^k_{train} **do**
      $E_{i,j} = x_{i,j}$ //set $x_{i,j}$ as the input to Embedding module
      for $m = 1$ **to** M **do**
        $\widetilde{E}_{i,j} = Conv_{\phi_m}(E_{i,j})$ //Feed forward each Conv block
        $\widetilde{E}_{i,j,pooled} = \text{Ada-P}_{\psi_m}(\widetilde{E}_{i,j})$ //Feed forward Ada-P module
        $E_{i,j} = \widetilde{E}_{i,j,pooled}$
      end for
    end for
    $E^k_{train} = \{E_{i,j} | 1 \leq i \leq N, 1 \leq j \leq K\}$ //Training embeddings
    for all x_q in D^k_{test} **do**
      $E_q = x_q$ //set x_q as the input to Embedding module
      for $m = 1$ **to** M **do**
        $\widetilde{E}_q = Conv_{\phi_m}(E_q)$ //Feed forward each Conv block
        $\widetilde{E}_{q,pooled} = \text{Ada-P}_{\psi_m}(\widetilde{E}_q)$ //Feed forward Ada-P module
        $E_q = \widetilde{E}_{q,pooled}$
      end for
    end for
    $E^k_{test} = \{E_q\}_{q=1}^{N_q}$ //Testing embeddings
    $\mathcal{L}^k_{test} = \sum_{q=1}^{N_q} L(clf(E^k_{train}, E_q), y_q), y_q \in D^k_{test}$ //Loss for each task
    $\mathcal{L} = \mathcal{L} + \mathcal{L}^k_{test}$ //Accumulate losses for a batch of tasks
  end for
  Update ϕ and ψ based on $\bigtriangledown_{\phi,\psi}\mathcal{L}$ //Update the parameters of feature extractor and Ada-P module
end for

4.3.1 Data Sets

We evaluate our Ada-P module on three widely studied few-shot learning data sets, miniImageNet, tieredImageNet and CUB. Note that, we discard the Omniglot data set used in the last chapter, since it only contains grayscale images of handwritten characters without background clutters, which is not suitable to evaluate our Ada-P module. In addition, we introduce two more challenging few-shot learning benchmarks, tieredImageNet and CUB. Note that these two data sets were not included in the previous chapter, because at the time of the work in Chapter 3 was done, these two data sets were not widely used as a few-shot image classification benchmark. To further show how our L2AE-D performs on them, we display the experimental results in Chapter 5, in which we combine our three research works together to demonstrate our contributions. The details of the data sets can be found in Section 2.4.2.

4.3.2 Comparison Algorithms and Network Architecture

In this chapter, we choose two widely used network architectures, 4Conv and Res12, as a feature extractor and select representative state-of-the-art methods based on the same network architecture for comparison. We also include several recent methods using deeper networks, which can further demonstrate the superiority of our Ada-P module.

Concretely, for 4Conv based feature extractor, we choose MAML (Finn, Abbeel, and Levine, 2017), A2P (Qiao et al., 2018), MetaOptNet (Lee et al., 2019), R2D2 (Bertinetto et al., 2019) from fast parameterisation based approaches, Meta-GAN+RN (Zhang et al., 2018b) from data generation based approaches, ProtoNet (Snell, Swersky, and Zemel, 2017), RN (Sung et al., 2018), L2AE-D (Song et al., 2021), TPN (Liu et al., 2019b), GNN (Garcia and Bruna, 2018) from metric learning approaches. For Res12 based feature extractor, we compare with LEO (Rusu et al., 2019), A2P (Qiao et al., 2018), MetaOptNet (Lee et al., 2019), MTL (Sun et al., 2019a) from fast parameterisation based approaches, SNAIL (Mishra et al., 2018), TADAM (Oreshkin, López, and Lacoste, 2018), DC (Lifchitz et al., 2019), CTM (Li et al., 2019b), ProtoNet (Snell, Swersky, and Zemel, 2017) from metric learning approaches. Note that we only compare against methods that provide results on a given data set, therefore the selected comparison algorithms on different data sets can be different.

To demonstrate that Ada-P is particularly suitable for few-shot learning, we

compare our module with several widely used and advanced pooling methods, namely max-pooling (Boureau, Ponce, and LeCun, 2010), average-pooling (Boureau, Ponce, and LeCun, 2010), overlapping-pooling (Krizhevsky, Sutskever, and Hinton, 2012), stochastic-pooling (Zeiler and Fergus, 2013), mixed-pooling (Lee, Gallagher, and Tu, 2016) and gated-pooling (Lee, Gallagher, and Tu, 2016).

Since our Ada-P can also be seen as a spatial attention mechanism, to illustrate the novelty of our Ada-P module for few-shot image classification, we further compare our method with several advanced spatial attention methods, namely SCA (Chen et al., 2017), CBAM (Woo et al., 2018), Residual-AT (Wang et al., 2017), Interpret-SA (Meng et al., 2019), L2-pay-AT (Jetley et al., 2018) from the computer vision field.

The detailed architecture of 4Conv and Res12 are described as follows:

1. **4Conv** backbone has been introduced in Section 2.4.5. In this approach, we replace commonly used max-pooling in 4Conv with our Ada-P module, in which the pooling weight generation block contains a 3×3 convolution with 1 filter, followed by BN and a sigmoid function. The pooling window size is 2×2. Following (Lee et al., 2019), we also include a dropblock (Ghiasi, Lin, and Le, 2018) layer after each convolutional block to reduce meta-level over-fitting. The keep rate and block size are set as 0.85 and 3 respectively for all 4 layers on miniImageNet, tieredImageNet and CUB. For the distance metric, we choose scaled Euclidean distance following (Ye et al., 2020), the scale is set 64.

2. **Res12** has been described in Section 2.4.5. The pooling operation is chosen as our Ada-P module in this approach. the pooling weight generation block contains a 3×3 convolution with 1 filter, followed by BN and a sigmoid function. The number of convolutional kernels k is set to be 64 in the first residual block and is doubled every next block. A dropblock layer is added after each residual block following (Lee et al., 2019) to reduce meta-level overfitting. The keep rate and block size are the same as those for 4Conv model. Moreover, we choose cosine distance as distance metric following (Gidaris and Komodakis, 2018), since we found that cosine distance works better with Res12 model empirically.

4.3.3 Experimental Setting and Evaluation

We evaluate our method on 5-way 1-shot, 5-way 5-shot learning tasks on all three data sets. Like most approaches based on 4Conv (Snell, Swersky, and Zemel, 2017; Finn, Abbeel, and Levine, 2017; Liu et al., 2019b), we train our 4Conv based approach in an episodic manner for a fair comparison. Note that we add a deep backbone, Res12, for evaluation here to show the effectiveness of the Ada-P module on deep networks. As discussed before, our L2AE-D method is not fully compatible with a deep architecture, thus, we did not include it in the last chapter. Specifically, for training Res12 based approaches, we adopted a large scale training strategy following (Sun et al., 2019a; Ye et al., 2020; Rusu et al., 2019; Li et al., 2019b) for better performance. During meta-testing, as before, we evaluate our approach on 6,000 randomly sampled tasks for all three data sets. The average classification accuracy and 95% confidence interval are reported. All experiments are performed using TensorFlow (Abadi et al., 2016) on a Titan Xp GPU.

1. **Episode-based meta-training:** During meta-training, the meta-batch size is set as 3 for both 1-shot and 5-shot tasks. Note that a larger meta-batch size could contribute to faster convergence while we set our meta-batch size taking into account the limitation of GPU memory. For each few-shot task, besides the $N \times K$ training examples, we randomly sample 6 test examples per class to compute the meta-training loss. Following (Snell, Swersky, and Zemel, 2017; Finn, Abbeel, and Levine, 2017; Liu et al., 2019b), we train our model with Adam (Kingma and Ba, 2015) with an initial learning rate of 0.001. On miniImageNet and tieredImageNet, we train 300,000 episodes for 1-shot tasks and 200,000 episodes for 5-shot tasks. The learning rate is cut in half every 60,000 and 40,000 episodes for 1-shot and 5-shot tasks, respectively. On CUB data set, we train 100,000 episodes for both 1-shot and 5-shot tasks and cut the learning rate in half every 20,000 episodes.

2. **Large scale meta-training:** Following (Sun et al., 2019a; Ye et al., 2020; Li et al., 2019b), we perform large scale training on Res12 using the whole meta-training set. A FC layer is added to the Res12 based feature extractor. Then the meta-training process is transformed into a 64, 351 and 100 classes classification problem for miniImageNet, tieredImageNet and CUB, respectively. Following (Sun et al., 2019a; Lee et al., 2019), we train our model with stochastic gradient descent with Nesterov momentum

of 0.9 with an initial learning rate of 0.1 for 30,000 episodes on miniImageNet and CUB, 100,000 episodes on tieredImageNet. The learning rate is divided by 10 every 10,000 episodes for miniImageNet and CUB, 20,000 episodes for tieredImageNet. The batch size is set as 128. The weight decay is set to be 0.0005. We adopt random horizontal flip and random crop data augmentations as in (Lee et al., 2019; Li et al., 2019b; Sun et al., 2019a). After the large scale training, the feature extractor can be used in few-shot classification tasks without any fine-tuning.

4.4 Analysis of Results

In this section, we analyse the results obtained from different experimental studies. Specifically, our aims are:

- To compare the performance of our Ada-P module with several widely used and advanced pooling methods in few-shot learning problems (Section 4.4.1).
- To perform ablation studies that test whether pooling operations are really needed in few-shot learning, verifies the working of Ada-P is not caused by adding more parameters (Section 4.4.2).
- To analyse the benefits and flexibility of our Ada-P module when they are incorporated into existing few-shot learning approaches (Section 4.4.3).
- To check whether the superiority of our few-shot learning approach is maintained on various data sets (small, large, fine-grained data sets) based on both shallow and deep models (Section 4.4.4).
- To illustrate the novelty of designing effective spatial attention methods for few-shot learning problems, comparing the performance of our Ada-P module against advanced spatial attention methods (Section 4.4.5).
- To analyse how much computational burden the Ada-P module introduces to an few-shot learning approach. (Section 4.4.6).

4.4.1 Comparisons with Pooling Baselines

To demonstrate the superiority of Ada-P, we first compare our Ada-P with a few pooling baselines on few-shot learning problems in Table 4.1. The same as before, we choose ProtoNet as the baseline. We compare Ada-P with a few pooling baselines mentioned in Section 4.3.2. From Table 4.1, we can see that

our approach outperforms all these baselines, which verifies Ada-P is more suitable for few-shot learning problems.

TABLE 4.1: Comparisons with several pooling baselines. **Archt.** represents the architecture of the feature extractor. The last number of **Archt.** stands for the number of filters in each convolutional layer. The average accuracy (%) with 95% confidence intervals are reported. The best and second best performing results are highlighted in bold and underlined, respectively. $^+$ represents an enhanced version of ProtoNet.

		miniImageNet 5-way	
Methods	**Archt.**	**1-shot(%)**	**5-shot(%)**
ProtoNet$^+$ w/ Max-pooling	4Conv-64	52.34 ± 0.26	69.90 ± 0.18
ProtoNet$^+$ w/ Avg-pooling	4Conv-64	53.01 ± 0.26	70.12 ± 0.18
ProtoNet$^+$ w/ Max-pooling-overlap	4Conv-64	$\underline{53.23 \pm 0.26}$	70.14 ± 0.18
ProtoNet$^+$ w/ Avg-pooling-overlap	4Conv-64	52.59 ± 0.26	70.43 ± 0.18
ProtoNet$^+$ w/ Stochastic-pooling	4Conv-64	51.98 ± 0.26	69.92 ± 0.20
ProtoNet$^+$ w/ Mixed-pooling	4Conv-64	52.88 ± 0.26	$\underline{70.87 \pm 0.18}$
ProtoNet$^+$ w/ Gated-pooling	4Conv-64	53.16 ± 0.26	70.77 ± 0.18
ProtoNet$^+$ w/ Ada-P	4Conv-64	$\mathbf{54.75 \pm 0.26}$	$\mathbf{71.63 \pm 0.20}$

To further show how our adaptive pooling works in comparison to commonly used pooling methods, we visualise the embeddings after the pooling operation in some convolutional layers of a few samples. Specifically, we use two ways to visual convolutional feature maps. First, we simply compress the multiple channels into a single channel by the mean operation as shown in Figure 4.2. Second, we apply a famous CNNs based visualisation technique (Zeiler and Fergus, 2014) by mapping the convolutional features to the input pixel space as shown in 4.3. Figures 4.2 and 4.3 show the visualisation of the training embeddings in the first three convolutional layers using different pooling methods from a 5-way 1-shot task. The categories are trifle, Alaskan malamute, vase, lion and hourglass from left to right. We can see that in the first layer, the embeddings of max-pooling and avg-pooling look similar. These two pooling methods seize a few relevant features, such as the shape of the Alaskan malamute and the vase, the eyes and noses of the lions, while our Ada-P module preserves more details, such as the hair features of the Alaskan malamute and the lions. For few-shot learning, we do not want to lose any useful information from the beginning when the training data is very small. Therefore, the end-to-end trained Ada-P module in the first layer learns to preserve more details, which is more suitable for few-shot learning. In the second and third layers, the feature maps are downsampled to a low-resolution space and the blurred mean feature maps in Figure 4.2 cannot provide much

insight. However, taking advantage of the visualisation technique in (Zeiler and Fergus, 2014), we can map the low-resolution feature maps back to the input pixel space. In Figure 4.3, we can see that the embeddings of Ada-P in the third layer focus more on the target objects, while the other pooling methods include more background noise. We can easily recognise the target object from the embeddings of Ada-P in the third layer compared to the original images. However, the embeddings of max-pooling and avg-pooling look blurred and it is difficult to distinguish the target object and the background. Therefore, we can conclude the Ada-P in the higher layers learns to perform spatial attention and suppresses background noise, which is aligned with our motivation.

4.4.2 Ablation Studies on Ada-P

We further conduct two ablation studies shown in Table 4.2 to show the actual working of our Ada-P. Specifically, we first remove the pooling operations in ProtoNet and set the convolutional stride as 2 to perform downsampling. It can be seen that all other pooling methods including ours outperform convolutional stride based downsampling, which shows that pooling is necessary for few-shot learning. Besides, we test whether our improvements are caused by adding more learnable parameters in CNNs. Since our meta-learner only includes one convolutional kernel, we add one more kernel in each convolutional layer in ProtoNet with max-pooling and average-pooling for a fair comparison. The results show our method outperforms ProtoNet with additional kernels significantly, which demonstrates that our improvements are not caused by adding more parameters but learning to assign adaptive pooling weights.

TABLE 4.2: Ablation study. **Archt.** represents the architecture of the feature extractor. The last number of **Archt.** stands for the number of filters in each convolutional layer. The average accuracy (%) with 95% confidence intervals are reported. $^+$ represents an enhanced version of ProtoNet.

		miniImageNet 5-way	
Methods	**Archt.**	**1-shot(%)**	**5-shot(%)**
ProtoNet$^+$ w/ Conv-stride(2)	4Conv-64	49.98 ± 0.25	66.91 ± 0.17
ProtoNet$^+$ w/ Max-pooling	4Conv-65	52.59 ± 0.26	70.37 ± 0.18
ProtoNet$^+$ w/ Avg-pooling	4Conv-65	53.13 ± 0.26	70.67 ± 0.18
ProtoNet$^+$ w/ Ada-P	4Conv-64	$\mathbf{54.75 \pm 0.26}$	$\mathbf{71.63 \pm 0.20}$

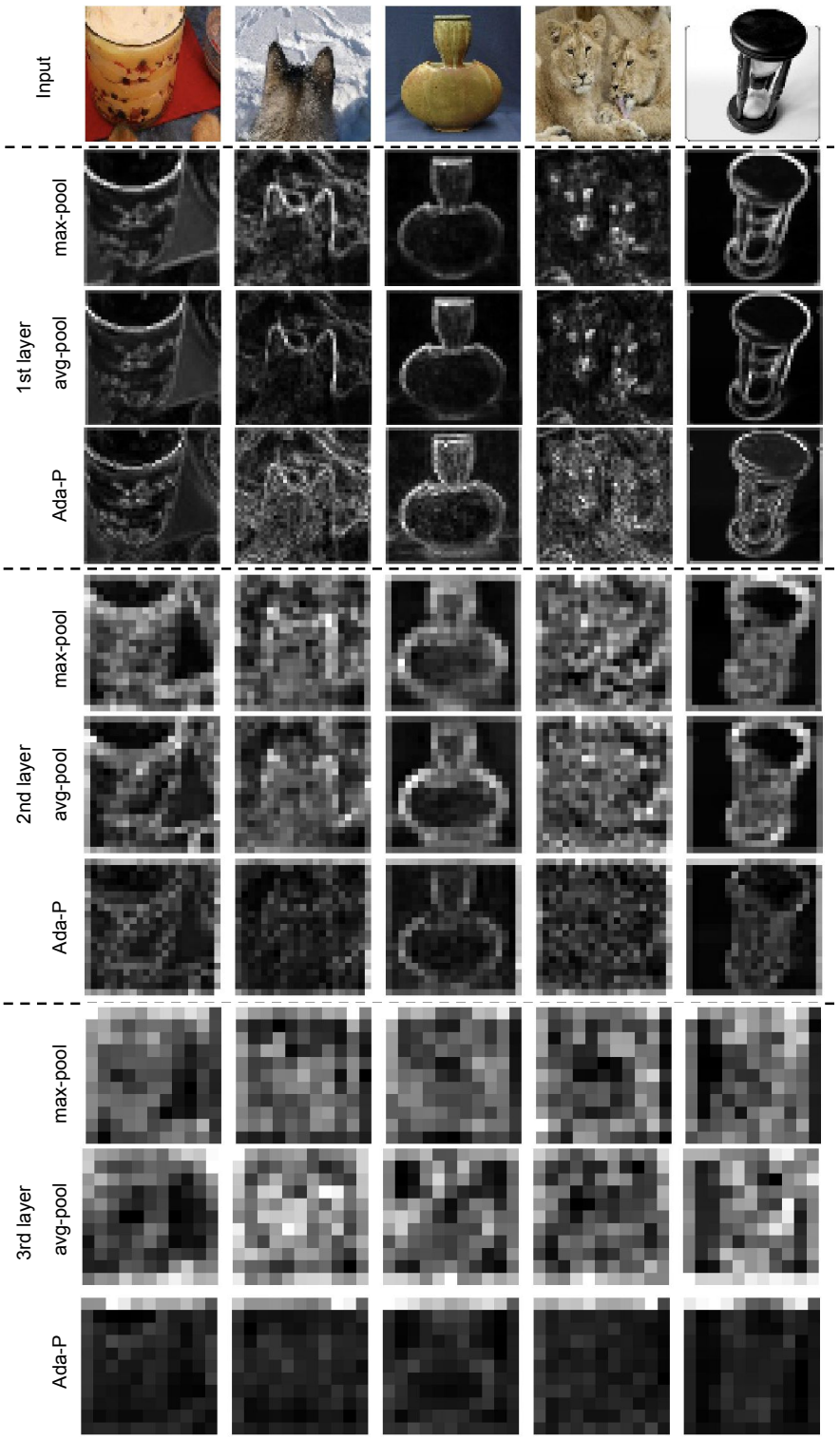


FIGURE 4.2: Visualisation of feature maps by compressing multiple feature maps into a mean one. The first row shows five input images. The Figures between dash lines represent the embeddings of different pooling operations in different layers.

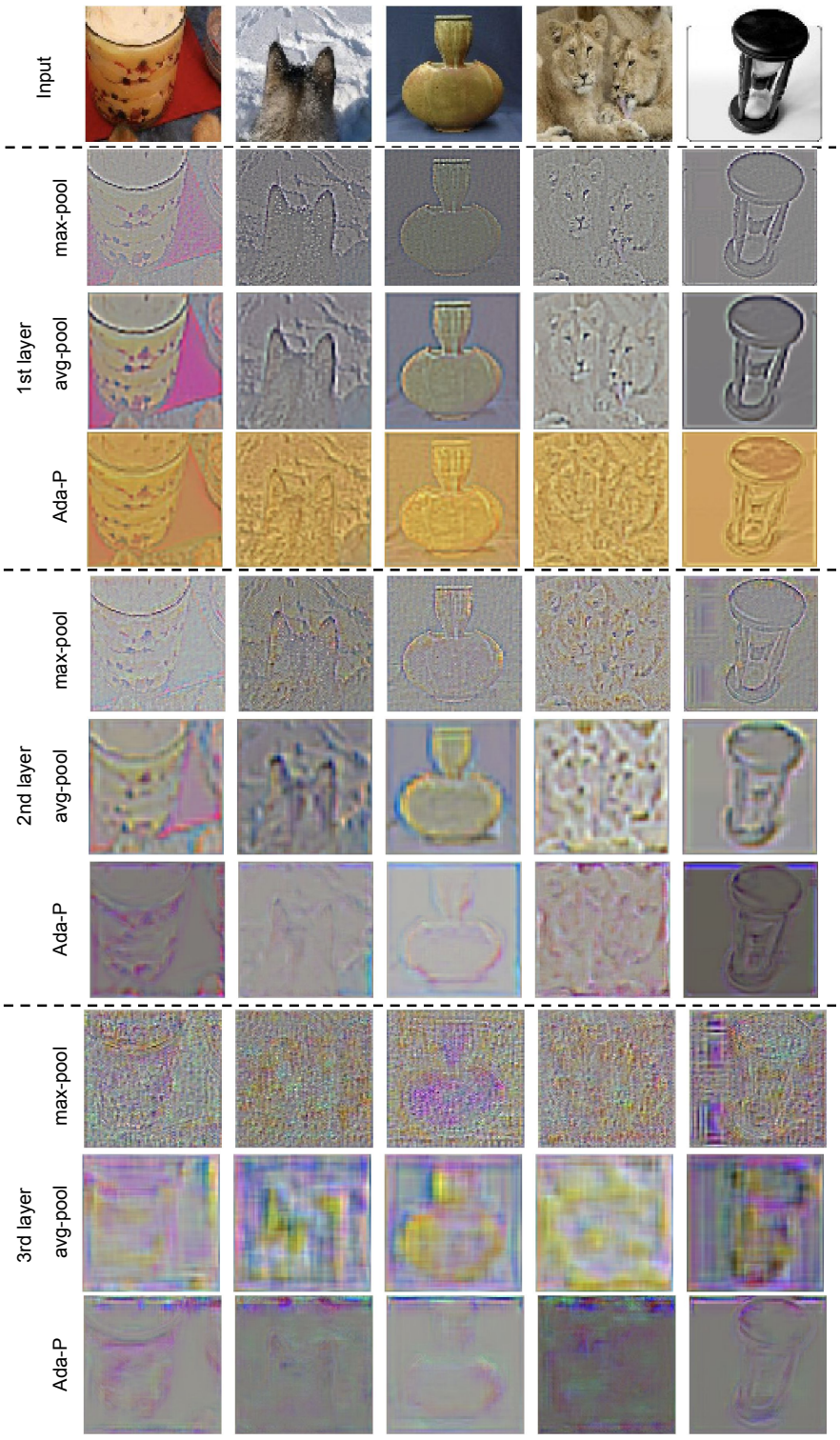


FIGURE 4.3: Visualisation of feature maps by Deconvolution (Zeiler and Fergus, 2014). The first row shows five input images. The Figures between dash lines represent the embeddings of different pooling operations in different layers.

4.4.3 Incorporating Ada-P into Representative Approaches.

To verify the effectiveness and compatibility of our Ada-P module, we incorporate either and both of them into a few existing few-shot learning approaches based on a simple backbone, 4Conv. We choose MAML (Finn, Abbeel, and Levine, 2017), RN (Sung et al., 2018), ProtoNet (Snell, Swersky, and Zemel, 2017), L2AE-D (Song et al., 2021) and TPN (Liu et al., 2019b), because they are representative approaches from different few-shot learning branches and also include transductive and inductive methods. We perform a comparison on miniImageNet using the 4Conv model. For ProtoNet, we use an enhanced version with dropblock and meta-batch training strategy. For the other methods, we do not add any training strategies described in Section 4.3.1, such as dropblock and augmentation. We reuse their released code and incorporate our Ada-P module, strictly following their respective experimental setting. Table 4.3 shows the comparison results. We can see our Ada-P module improves all the methods on 1-shot and 5-shot tasks, especially for MAML on 5-shot tasks it improves significantly by around 4%. When the Ada-P module is introduced individually, we can see all the approaches are improved. Note that it improves RN marginally, since RN only uses 2 pooling layers while the other methods all use 4 layers. As for the other methods, our pooling module mostly improves by around $1.5 - 2.5\%$. Overall, this experiment verifies our Ada-P module is beneficial for and compatible with different types of few-shot learning frameworks, including fast parameterisation and metric learning approaches or inductive and transductive methods.

4.4.4 Comparisons with the State-of-the-Art Approaches

In this section, we compare our method with state-of-the-art approaches on 5-way 1-shot and 5-shot classification tasks on various data sets using both 4Conv and Res12 models. The results and analysis are presented as follows:

Results on miniImageNet: The comparisons on miniImageNet using 4Conv and Res12 models are shown in Table 4.4. We choose comparable state-of-the-art methods from different branches for comparison. Based on the 4Conv feature extractor, our approach achieves state-of-the-art performance on 5-shot tasks and a comparable result to the best performance on 1-shot tasks. Based on the 4Conv feature extractor, our approach achieves state-of-the-art performance on both 1-shot and 5-shot tasks. Note that the second best performing method on 1-shot tasks, A2P (Qiao et al., 2018), adopted a large scale pre-training strategy, which is not used in all other methods. However, our method

TABLE 4.3: Results after incorporating Ada-P into several existing approaches on miniImageNet. **Trans** represents if a method is a transductive method. BN represents a BN based transductive method. The last number of **Archt.** stands for the number of filters in each convolutional layer. The average accuracy (%) with 95% confidence intervals is reported. ↑ shows the improvements after incorporating Ada-P. $^+$ represents an enhanced version of ProtoNet.

			miniImageNet 5-way	
Methods	**Trans**	**Archt.**	**1-shot(%)**	**5-shot(%)**
MAML	BN	4Conv-32	48.70 ± 1.84	63.74 ± 0.92
MAML w/ Ada-P	BN	4Conv-32	$50.74 \pm 1.82 \uparrow 2.04$	$66.85 \pm 0.87 \uparrow 3.11$
RN	BN	4Conv-64	50.44 ± 0.82	65.32 ± 0.70
RN w/ Ada-P	BN	4Conv-64	$50.95 \pm 0.86 \uparrow 0.51$	$66.46 \pm 0.69 \uparrow 1.14$
ProtoNet$^+$	N	4Conv-64	52.34 ± 0.26	69.90 ± 0.18
ProtoNet$^+$ w/ Ada-P	N	4Conv-64	$54.75 \pm 0.26 \uparrow 2.41$	$71.63 \pm 0.20 \uparrow 1.73$
TPN	Y	4Conv-64	53.75 ± 0.86	69.43 ± 0.68
TPN w/ Ada-P	Y	4Conv-64	$55.19 \pm 0.86 \uparrow 1.44$	$70.90 \pm 0.69 \uparrow 1.47$

achieves a better performance without pre-training.

Based on the Res12 feature extractor, our approach achieves the second best performance on both 5-way 1-shot and 5-shot tasks. It is noteworthy that these methods are not strictly comparable since their network architectures are not exactly the same. For example, the best performing method, CTM, use ResNet-18 backbone, which represents a deeper architecture (11 million parameters) with around 37% more parameters than our Res12 model (8 million parameters). We apply the architecture that the majority of previous methods used and achieve competitive performance on both 1-shot and 5-shot tasks compared to the state-of-the-art results based on this backbone. Our method also improves upon ProtoNet$^+$ by around 1.0% on both 1-shot and 5-shot tasks. Since our approach is built upon ProtoNet$^+$ under the same experimental setting, these improvements also verify the effectiveness of our few-shot learning approach when using a deeper model.

Results on tieredImageNet: In Table 4.4, we also compare our approach with recent state-of-the-art methods that provide evaluations on 5-way 1-shot and 5-shot classification tasks on tieredImageNet using a 4Conv and Res12 based model. Note that the missing values in Table 4.4 indicate the methods are not tested on tiredImageNet. TieredImageNet is a more challenging benchmark compared to miniImageNet as discussed before, however, our approach still achieves a promising performance. Based on the 4Conv model, our approach achieves the best performance on 5-shot tasks and the third best on

TABLE 4.4: Results on miniImageNet and tieredImageNet. The average accuracy (%) with 95% confidence intervals is reported. The best and second best performing results are highlighted in bold and underlined, respectively. Res12 represents Res12 models and the behind number stands for the number of filters in the last residual block. WRN represents wide residual networks. Res18 represents ResNet-18 models. † uses the results of MetaOptNet with SVM trained only on the meta-training set. $^{+}$ represents an enhanced version of ProtoNet.

		miniImageNet 5-way		**tieredImageNet 5-way**	
Methods	**Archt.**	**1-shot(%)**	**5-shot(%)**	**1-shot(%)**	**5-shot(%)**
MAML	4Conv-32	48.70±1.84	63.74±0.92	51.76±1.81	70.30±1.75
RN	4Conv-64	50.44±0.82	65.32±0.70	54.48±0.93	71.32±0.78
MetaGAN+RN	4Conv-64	52.71±0.64	68.63±0.67		
L2AE-D	4Conv-64	53.85±0.85	70.16±0.65	55.81±0.85	71.71±0.65
TPN	4Conv-64	53.75±0.86	69.43±0.68	**57.53±0.96**	72.85±0.74
A2P	4Conv-64	54.53±0.40	67.87±0.20		
ProtoNet^{+}	4Conv-64	52.34±0.26	69.90±0.18	52.44±0.27	70.91±0.23
ProtoNet^{+} w/ Ada-P	4Conv-64	**54.75±0.26**	**71.63±0.20**	55.73±0.28	**73.54±0.23**
SNAIL	Res12-256	55.71±0.99	68.88±0.92		
TADAM	Res12-512	58.50±0.30	76.70±0.30		
A2P	WRN28	59.60±0.41	73.74±0.19		
LEO	WRN28	61.76±0.08	77.59±0.12	66.33±0.05	81.44±0.09
MTL	Res12-512	61.20±1.80	75.50±0.80	62.83±1.80	74.50±0.92
DC	Res12-512	62.53±0.19	79.77±0.19		
MetaOptNet†	Res12-640	62.64±0.61	78.64±0.46	65.99±0.72	81.56±0.53
CTM	Res18	**64.12±0.82**	**80.51±0.13**	68.41±0.39	84.28±1.73
ProtoNet^{+}	Res12-512	61.27±0.26	77.79±0.18	67.95±0.30	83.30±0.21
ProtoNet^{+} w/ Ada-P	Res12-512	62.69±0.27	78.75±0.19	**69.04±0.30**	**84.51±0.21**

1-shot tasks. The best performing method, TPN, is a transductive method that uses a few unlabelled examples to assist learning, while our method is an inductive approach that only uses labelled training examples to predict test examples. Based on the Res12 model, Ada-P achieves state-of-the-art performance on both 1-shot and 5-shot tasks, even compared to other approaches with a deeper model or more kernels. In addition, our method improves upon our baseline, ProtoNet^{+}, using both 4Conv and Res12 model.

Results on CUB: Finally, we test our approach on fine-grained few-shot classification tasks on CUB. Table 4.5 summarises the comparison of our and other few-shot learning approaches using both 4Conv and Res12 backbones. Note that some recent approaches achieve much higher performance on CUB based on a much deeper backbone, such as Res18 or WRN28. We do not include them

TABLE 4.5: Results on CUB. The average accuracy (%) with 95% confidence intervals is reported. The best and second best performing results are highlighted in bold and underlined, respectively. $^{+}$ represents an enhanced version of ProtoNet. — indicates the method does not provide a confidence interval. * represents results from (Chen et al., 2018).

		CUB 5-way	
Methods	**Archt.**	**1-shot(%)**	**5-shot(%)**
MatchingNet*	4Conv-64	60.52 ± 0.88	75.29 ± 0.75
MAML*	4Conv-64	54.73 ± 0.97	75.75 ± 0.76
RN*	4Conv-64	$\underline{62.34 \pm 0.94}$	$\mathbf{77.84 \pm 0.68}$
ProtoNet^{+}	4Conv-64	57.61 ± 0.29	74.51 ± 0.18
ProtoNet^{+} w/ Ada-P	4Conv-64	$\mathbf{62.40 \pm 0.30}$	$\underline{77.65 \pm 0.18}$
MatchingNet*	Res10-512	$\mathbf{71.29 \pm 0.87}$	83.47 ± 0.58
MAML*	Res10-512	70.32 ± 0.99	80.93 ± 0.71
RN*	Res10-512	70.47 ± 0.99	83.70 ± 0.55
ProtoNet^{+}	Res12-512	68.60 ± 0.26	$\underline{85.51 \pm 0.20}$
ProtoNet^{+} w/ Ada-P	Res12-512	$\underline{70.79 \pm 0.26}$	$\mathbf{85.81 \pm 0.20}$

in our comparison for fairness. From Table 4.5, we can observe that the Ada-P module achieves very competitive performance on both shallow and deep backbones on both 1-shot and 5-shot tasks. Besides, it improves upon our baseline, ProtoNet^{+}, using both 4Conv and Res12 model by a large margin, which demonstrates the Ada-P module is effective for fine-grained few-shot classification tasks.

4.4.5 Comparisons with Spatial Attention Methods

Since our Ada-P module can be seen as spatial attention mechanisms, to illustrate the novelty of our Ada-P module for few-shot learning, we integrate a few recent representative spatial attention approaches into the few-shot learning framework and compare with our method. Our aim is to show simply applying off-the-shelf spatial attention methods does not work satisfactorily for few-shot learning and we need to specifically design an effective approach. The selected advanced spatial attention methods are SCA (Chen et al., 2017), CBAM (Woo et al., 2018), Residual-AT (Wang et al., 2017), Interpret-SA (Meng et al., 2019), L2-pay-AT (Jetley et al., 2018) from the computer vision field. It is noteworthy that these methods are not specifically designed for few-shot learning, some of them need to be tweaked a bit to be compatible with the few-shot learning framework. The same as before, we choose ProtoNet as a

TABLE 4.6: Comparisons with several spatial attention methods on few-shot learning problems. **Archt.** represents the architecture of the feature extractor. The last number of **Archt.** stands for the number of filters in each convolutional layer. The best and second best performing results are highlighted in bold and underlined, respectively. The average accuracy (%) with 95% confidence intervals are reported. $^+$ represents an enhanced version of ProtoNet.

		miniImageNet 5-way	
Methods	**Archt.**	**1-shot(%)**	**5-shot(%)**
ProtoNet$^+$	4Conv-64	52.34 ± 0.26	69.90 ± 0.18
ProtoNet$^+$ w/ SCA	4Conv-64	51.45 ± 0.26	68.83 ± 0.18
ProtoNet$^+$ w/ CBAM	4Conv-64	$\underline{53.54 \pm 0.26}$	70.19 ± 0.18
ProtoNet$^+$ w/ Residual-AT	4Conv-64	51.91 ± 0.26	70.35 ± 0.18
ProtoNet$^+$ w/ Interpret-SA	4Conv-64	53.70 ± 0.26	$\underline{71.18 \pm 0.18}$
ProtoNet$^+$ w/ L2-pay-AT	4Conv-64	52.52 ± 0.26	69.97 ± 0.18
ProtoNet$^+$ w/ Ada-P	4Conv-64	$\mathbf{54.75 \pm 0.26}$	$\mathbf{71.63 \pm 0.20}$

baseline and incorporate each spatial attention method into it respectively. To compare the meta-testing performance, we train a 4Conv backbone on miniImageNet data set in an episodic manner. The results are shown in Table 4.6, where we can observe that simply incorporating the selected spatial attention methods into few-shot image classification does not improve too much. SCA and Residual-AT even degrade the performance. They work well on standard learning tasks, while they may not be a good choice for few-shot learning tasks. This demonstrates designing an effective spatial attention method for few-shot image classification is a nontrivial task. We can not simply introduce off-the-shelf spatial attention methods to improve few-shot learning. From Table 7, we can also see that our Ada-P module outperform other spatial attention methods, respectively, on both 1-shot module is especially effective for few-shot image classification.

To better show how these spatial attention methods influence the feature extraction, we visualise the embedding space in the first two convolutional layers of a few samples as shown in Figure 4.4 and 4.5. The same as before, we compress the multiple channels into a single one by mean operation. From Figure 4, we can see that in the first layer the Ada-P module preserves more details of the object. This is helpful for few-shot image classification, since we do not want to lose any useful information at the beginning. Other spatial attention methods either include more background noise or focus too much on a specific

feature on the object, such as the eyes of lions. In the second layer, as shown in Figure 5, our Ada-P module suppresses the background noise and concentrates more on the target object. Compared to other spatial attention methods, our Ada-P module assists the feature extractor to focus more on the objects.

Based on the above analysis, we can conclude that simply incorporating other spatial attention methods into few-shot image classification improves marginally, while our Ada-P module is carefully designed and especially effective for few-shot image classification problems, which demonstrates our contributions and novelty.

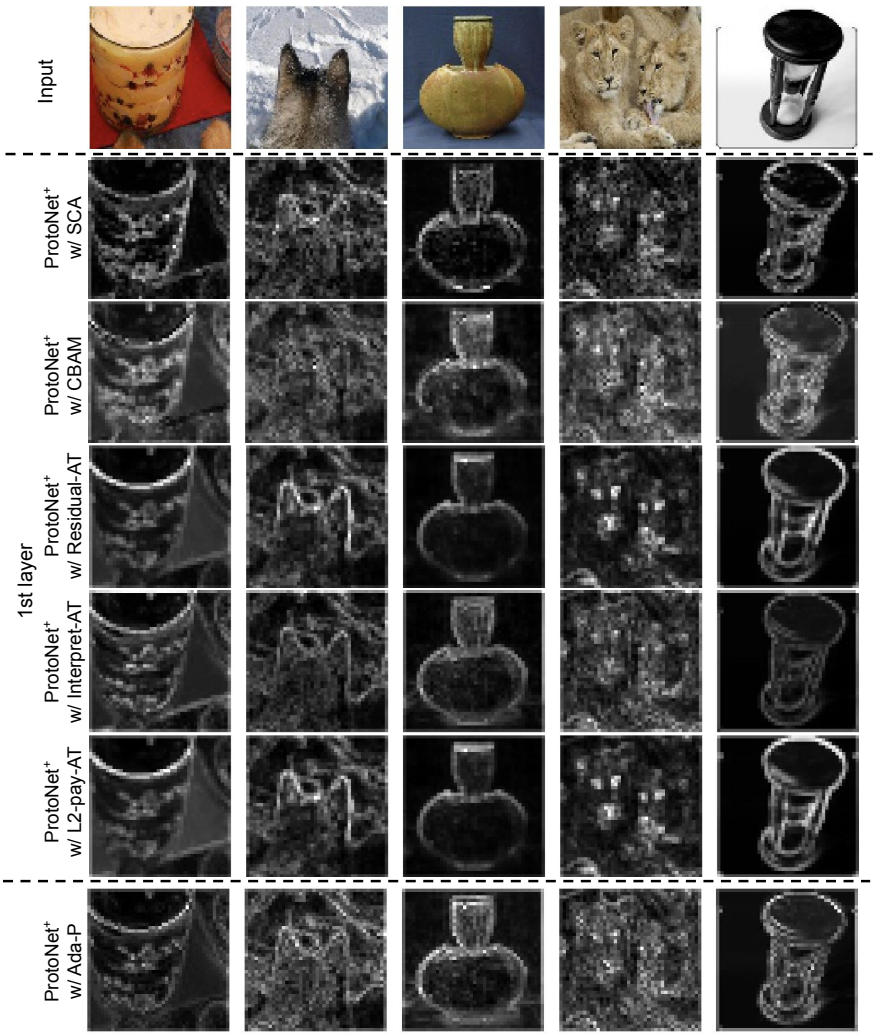


FIGURE 4.4: Visualisation of feature maps by compressing multiple feature maps into a mean one. The first row shows five input images. The Figures between dash lines represent the embeddings of different spatial attention methods in the first layer.

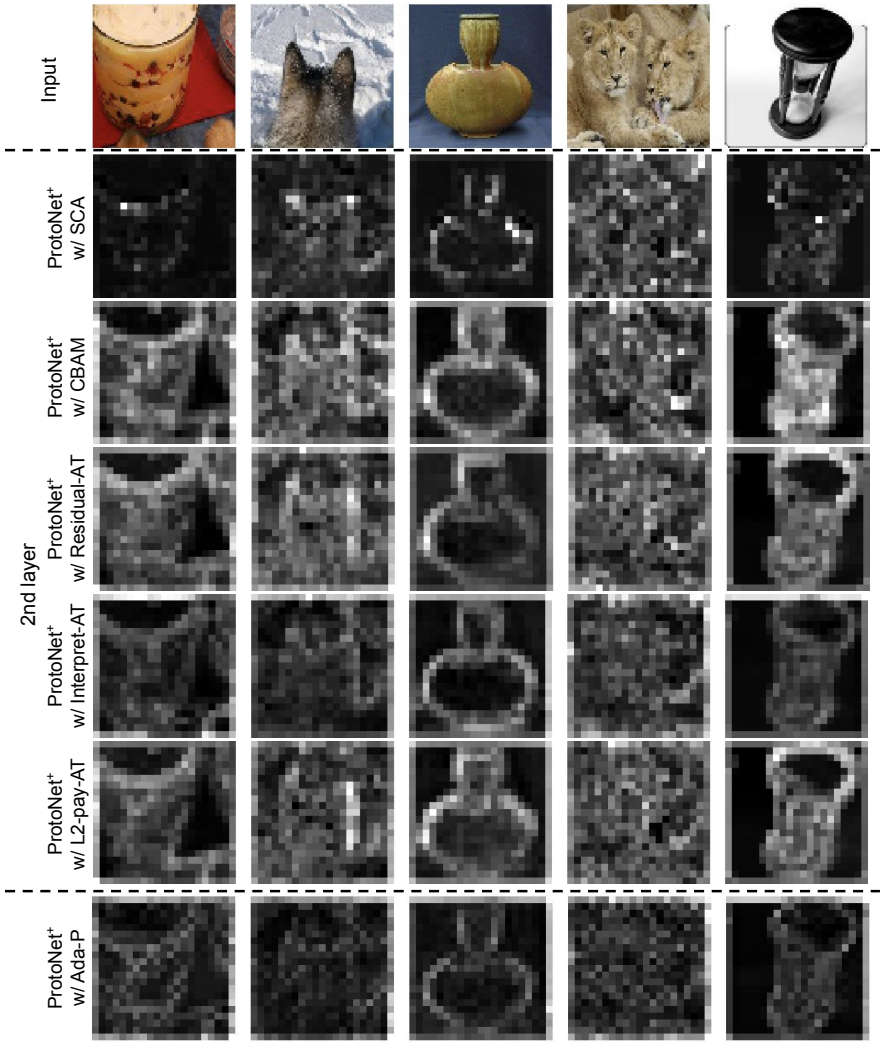


FIGURE 4.5: Visualisation of feature maps by compressing multiple feature maps into a mean one. The first row shows five input images. The Figures between dash lines represent the embeddings of different spatial attention methods in the second layer.

4.4.6 Runtime Analysis

To analyse how much computational burden the Ada-P introduces, we have calculated the wall-clock time for training and testing on 1,000 few-shot learning tasks on miniImageNet before and after adding our Ada-P modules, respectively, which are shown in table 4.7. We can see that our Ada-P module leads to an increase of runtime compared to ProtoNet baseline based on both shallow and deep backbones. On meta-testing phase, the runtime based on 4Conv and Res12 is increased by 15% and 10%, respectively. However, we think these increases are acceptable considering the improvement of the performance on few-shot image classification.

TABLE 4.7: Runtime Analysis. Each number represents the wall-clock time (s) of training/testing on 1,000 few-shot image classification tasks on miniImageNet.

		Training runtime(s)		Testing runtime(s)	
		5-way		5-way	
Methods	**Archt.**	**1-shot**	**5-shot**	**1-shot**	**5-shot**
ProtoNet$^+$	4Conv-64	19.11	23.89	9.89	10.91
ProtoNet$^+$ w/ Ada-P	4Conv-64	30.52	36.62	11.05	12.86
ProtoNet$^+$	Res12-512	132.10	167.64	35.48	44.85
ProtoNet$^+$ w/ Ada-P	Res12-512	145.30	184.24	39.01	49.19

4.4.7 Summary of Results

Based on the above results and analysis, we can conclude the following remarks, which also reflect the aims of different experiments at the beginning of this section.

1. Our Ada-P module performs better than a few pooling baselines in few-shot image classification problems, such as max-pooling, average-pooling, overlapping-pooling, mixed-pooling and gated-pooling. This demonstrates our Ada-P is more suitable for few-shot learning.

2. The ablation studies show that pooling is a necessary component of a feature extractor for few-shot image classification and the working of Ada-P is not simply caused by adding more learnable parameters.

3. Our Ada-P module is compatible with and beneficial for most existing few-shot learning approaches. The results in Table 4.3 show significant improvements when incorporating Ada-P into representative few-shot learning methods.

4. The performance of our approach based on both shallow (4Conv) and deep (Res12) backbones on various data sets (small, large and fine-grained data sets) remains superior compared to the state-of-the-art approaches, which illustrates the effectiveness and robustness of our modules.

5. Our Ada-P outperform advanced spatial attention methods on few-shot learning problems. The visualisation of the embedding space further shows our Ada-P can avoid losing useful information and help focus on target objects. These demonstrate our method are more effective for few-shot image classification than other spatial attention methods.

4.5 Summary

In this chapter, we have achieved research objective 2 by designing an adaptive pooling method for few-shot image classification, which reduced information loss of conventional pooling operations and meanwhile performed spatial attention on each embedding to mitigate background clutters. Our Ada-P module has learned a meta-learner to assign adaptive pooling weights to each individual embedding, which took into account the importance of the features at different spatial locations and lost less useful information. Our Ada-P module can be used as a plug-and-play module and applied in various existing few-shot learning approaches trained in an end-to-end manner, which has been demonstrated in experiments and achieved significant improvements. We have evaluated our approach on three widely used benchmarks, miniImangeNet, tieredImageNet, CUB and achieved very competitive performance compared to the state-of-the-art.

In the first two stages of our research, we concentrated on a few-shot image classification task itself, addressing the two core issues by maximising the use of training data and improving the feature extraction process. Since the available information of a few-shot learning task is very limited, adhering to a few-shot image classification task itself could provide marginal performance improvements. Consequently, it is natural to think of introducing auxiliary information to assist few-shot image classification for further performance enhancements. In the following chapter, we aim to tackle the two core issues from the perspective of leveraging auxiliary information.

Chapter 5

Leveraging Auxiliary Information for Few-Shot Image Classification

5.1 Introduction

In the first two stages of our research, we have addressed the two core issues of few-shot image classification from the viewpoint of maximising the use of training data and feature extraction. Even though we have increased the utility of the limited training samples in a few-shot learning task, the performance improvement is still limited by the insufficient available information in the training data. To further improve the final performance, one solution could be to introduce auxiliary information to assist the learning process of a few-shot image classification task. A follow-up question is what type of auxiliary information is helpful. Some existing works introduce semantic data (Xing et al., 2019) or manually assigned attributes (Tokmakov, Wang, and Hebert, 2019), to compensate lack of information. The added new features strengthen the learned representations, however, they ignore the intrinsic uncertainty which is one of the core issues of few-shot learning. Following the same principles of the previous contribution that target tackling the two core issues of few-shot image classification at the same time, we argue that addressing both of them is necessary and can lead to better performance.

To the best of our knowledge, there are only two previous works that introduce auxiliary information to tackle the intrinsic uncertainties. The method in (Wertheimer and Hariharan, 2019) introduced annotated bounding boxes as auxiliary information to distinguish the foreground and background of an image, then extracted and concatenated their respective embeddings to form a final class representation. One limitation of this approach is that it relies on manually annotated bounding boxes, which demands labour costs. Thus, a

more automatic way to obtain auxiliary information is needed. The other work used a pre-trained SOD model to extract the saliency map for each sample as auxiliary information, which is more automatic (Zhang, Zhang, and Koniusz, 2019). It proposed an with a complex architecture to mix the foregrounds and backgrounds of samples in a task to expand the training set. However, it skipped to explore some simpler methods to leverage auxiliary information, such as adding an additional channel. In our experiments, we find that this simple algorithm achieves even better results on few-shot image classification without using a complex architecture. Besides, the common weakness of the above two approaches is that they only considered a single type of auxiliary information and focused a specific strategy to leverage it. However, a comprehensive exploration of what is suitable auxiliary information and how to effectively leverage it to mitigate background clutters in few-shot image classification remains unexplored to date.

Having the above motivations in mind, in the third stage of our research, we provide a thorough exploration of leveraging suitable auxiliary information in few-shot image classification, considering its two cores issues simultaneously. Concretely, we test three types of auxiliary information related to identifying the target object in an image, namely edges, bounding boxes and saliency maps. These kinds of auxiliary information are extracted based on classic or promising edge detection (Canny, 1986), object detection (Bochkovskiy, Wang, and Liao, 2020) and saliency object detection techniques (Zhao et al., 2019a).

Taking the above auxiliary information in hand, we further design several methods to apply it in a few-shot image classification pipeline. We test the three types of auxiliary information using different methods to assist few-shot image classification on three widely used benchmarks, miniImageNet, tieredImageNet and CUB. The most suitable auxiliary information found and the best-performing method are then further incorporated into existing meta-learning approaches including our L2AE-D to demonstrate the improvements.

This chapter is structured as follows. First, we introduce how we obtain different types of auxiliary information and various avenues to utilise them to assist few-shot image classification in Section 5.2. Then we describe the used data sets and the experimental settings in Section 5.3. Section 5.4 analyses the experimental results. Section 5.5 illustrates the overall contribution of the three of our methods together. Finally, we summarise the achievement of the third stage of our research in Section 5.6.

5.2 Methodology

This section introduces our exploration of using auxiliary information to assist few-shot image classification. Since the auxiliary information is not provided in the original few-shot image classification data sets, we first explain how we extract auxiliary information for few-shot image classification benchmarks in Section 5.2.1. Then, we introduce a few designed avenues to leverage the auxiliary information in Section 5.2.2.

5.2.1 Auxiliary Information Extraction

Since our aim is to mitigate the background clutters in an image, in this research, we test three types of auxiliary information related to identifying the target object in an image, namely edges, bounding boxes and saliency maps. Edges targets detecting the points at which image brightness changes sharply. Since the brightness at the border between the foreground and background in an image usually changes greatly, we would expect edges could provide information about the outline of an target object. Bounding boxes are rectangles that tightly surround the target object in an image, which would provide information on where the foreground locates. Saliency maps highlight the target object in an image, which would display more exact localisation of the foreground. The above three types of auxiliary information can be obtained by manually annotation, a pre-defined algorithm or a pre-trained model. To incorporate them into few-shot image classification in an automated way, we choose a pre-defined algorithm for edges and pre-trained models for bounding boxes and saliency maps, so that, the auxiliary information for each sample can be extracted during both meta-training and meta-testing phase. An illustration of these three types of information for a few images in miniImageNet data set extracted by our selected detection methods is shown in Figure 5.1. The details of how we extract the auxiliary information for few-shot image classification data set are explained as follows:

Edges

One of the most commonly used edge detection tools is Canny edge detection, which uses a multi-stage algorithm, including noise reduction, finding intensity gradient, non-maximum suppression and hysteresis thresholding, to detect a wide range of edges in images (Dharampal, 2015). Even though it was a classic method proposed in the early days of the computer vision field, it still

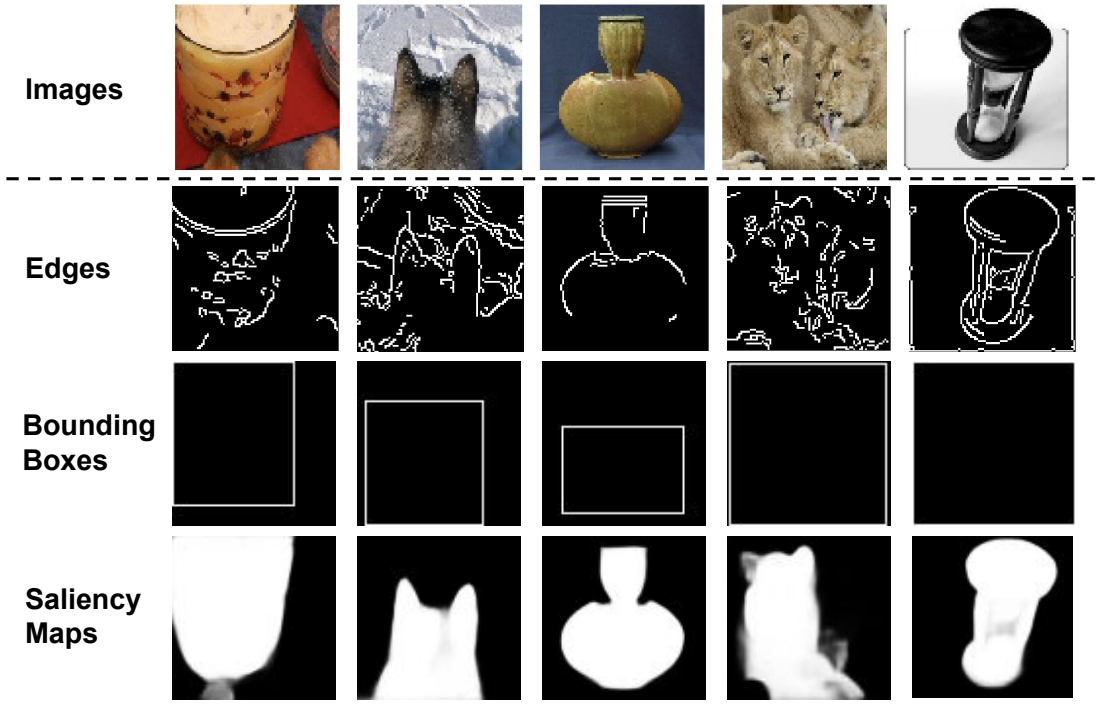


FIGURE 5.1: An illustration of three types of auxiliary information for a few images in miniImageNet data set.

provides a good performance. We select this technique to extract edge information of the images in few-shot image classification tasks. Since our target is to extract the edges between the foreground and background of an image, we need to carefully set two threshold values, maxVal and minVal, in the hysteresis thresholding stage. These two values control how many detected edges can be kept or discarded. To avoid too many noisy edges in background clutters, we empirically set the two threshold values, maxVal and minVal, as 300 and 200, respectively. Note that tuning these hyper-parameters can be a limitation of our approach, we leave exploring more automatic method for edge detection as a future work. Some detected edges based on our selected threshold values can be seen in Figure 5.1.

Bounding Boxes

A bounding box is a rectangle that tightly surrounds the target object in an image. There are a large number of works on detecting bounding boxes for images. Taking into account efficiency and accuracy, we choose a pre-trained YOLO (Bochkovskiy, Wang, and Liao, 2020) model as our bounding box detector, serving as prior knowledge for few-shot image classification. There are many versions of YOLO, such as YOLOv1 (Redmon et al., 2016), YOLOv2 (Redmon and Farhadi, 2017), YOLOv3 (Redmon and Farhadi, 2018) and YOLOv4

(Bochkovskiy, Wang, and Liao, 2020). In our work, we select YOLOv4 to extract bounding boxes on few-shot image classification data sets for two reasons. First, compared to previous versions, YOLOv4 achieves the best performance on object detection and can generate bounding boxes at a real-time speed. Second, since the images in widely used few-shot image classification benchmarks are low resolution (84×84), we need an object detector that can handle such images. The YOLO variants are all trained on higher resolution images, so that, they cannot work well with low-resolution images other than YOLOv4. Compared to previous versions, YOLOv4 incorporates a few training techniques, such as data augmentation and self adversarial training, to make an object detector more robust and effective, so that, it can identify relatively good bounding boxes for low-resolution images. Therefore, in our work, we directly feed the images of few-shot image classification data sets into a pre-trained YOLOv4 model to obtain bounding boxes as auxiliary information.

Saliency Maps

A saliency map is a grey-scale image that highlights the region of target objects, on which people's eyes focus. In this paper, EGNet (Zhao et al., 2019a) is chosen to generate saliency maps as a recent approach offering competitive performance on standard SOD benchmarks. Similar to extracting bounding boxes, EGNet trained on standard SOD benchmarks cannot work well with low resolution images either. To address the issue, we pre-train EGNet based on a resized SOD data set, DUTS (Wang, Ramanan, and Hebert, 2017), in which all the images are resized to 84×84 to keep them the same size as those in data sets for few-shot image classification. Note that there are a few overlappings, 245 images, between the meta-testing set of miniImageNet and the training set of DUTS. To strictly follow the meta-learning setting that the meta-testing samples should be unseen, we exclude those overlapping images from DUTS when pre-training EGNet for miniImageNet. After pre-training, we feed the images of few-shot image classification data sets into the pre-trained EGNet model to obtain saliency maps as auxiliary information.

5.2.2 Approaches to Leverage Auxiliary Information for Few-shot Image Classification

This section introduces few different methods to leverage the three types of auxiliary information, respectively, in few-shot image classification, namely

(a) Additional Channel, (b) Removing Background, (c) Using Respective Feature extractor and (d) Multi-Task Learning based on U-Net. These methods are inspired by previous research works that leveraged auxiliary information for image classification (Murabito et al., 2018; Zhuang et al., 2019; Figueroa-Flores et al., 2021). Specifically, to help image classification, the auxiliary information can be used as an additional input signal to tell a feature extractor when to focus (Murabito et al., 2018), to refine input images or even embeddings (Figueroa-Flores et al., 2021), and to serve as a supervisory signal to teach a feature extractor how to concentrate on a target object (Zhuang et al., 2019). Based on different application scenarios, these methods can be classified into two categories. Approaches (a), (b) and (c) all use images and auxiliary information during both meta-training and meta-testing phases. In this scenario, we need a pre-trained object detection or saliency object detection model to extract relevant auxiliary information for each sample during meta-testing, which is not very efficient. Method (d) leverages images and auxiliary information during meta-training, but only utilises images to conduct a few-shot image classification task during meta-testing. This would save more time comparing to the previous scenario, because there is no need to use a pre-trained model to extract auxiliary information on an unseen task during meta-testing. The details of all the above methods are described as follows:

(a) Additional Channel

Our selected three types of auxiliary information are all represented in the form of grayscale images with the same size as original images. Following (Murabito et al., 2018), we first apply a straightforward method to use auxiliary information, which is to add each of them, respectively, as an additional channel alongside the original RGB channels of an image as shown in Figure 5.2. The auxiliary information serving as an additional input signal would tell a feature extractor where the foreground locates in an image. Note that, when using bounding boxes, we set the values in the additional channel within a bounding box as 1 and set those outside a bounding box as 0. In this method, we do not need to modify much the network architecture or the training of a meta-learning approach other than increasing one input channel of CNNs.

(b) Removing Background

When incorporating auxiliary information, one goal is to mitigate the influence of background clutters in an image. Note that edge is not suitable for this, since

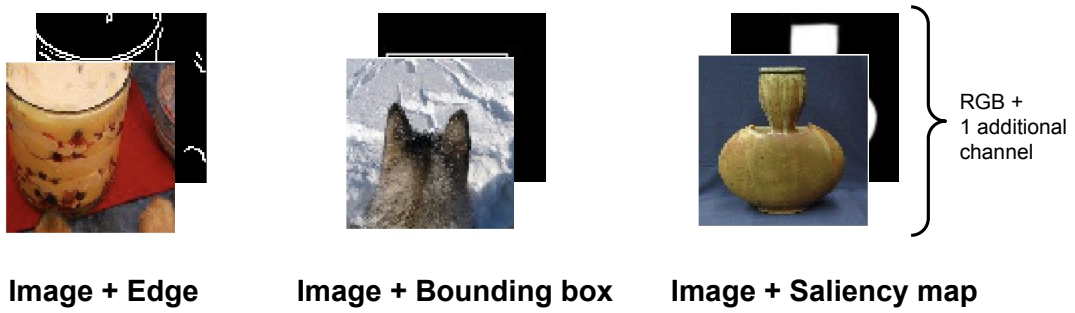


FIGURE 5.2: An illustration of adding three types of auxiliary information as an additional channel alongside the original RGB channels of an image.

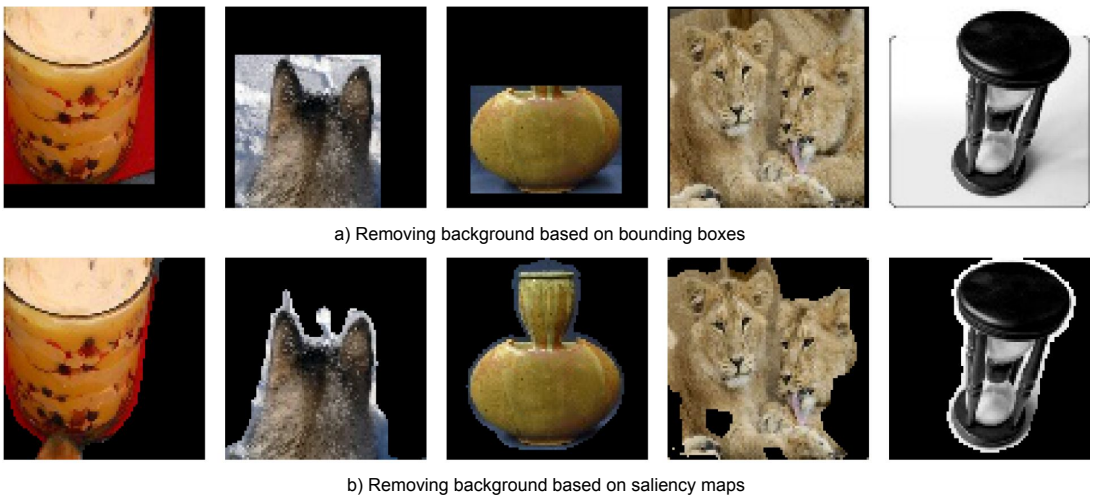


FIGURE 5.3: An illustration of removing the background of an image based on bounding boxes and saliency maps.

the outline of a target object is not guaranteed to be fully extracted by an edge detector, as shown in Figure 5.1, while we can use bounding boxes and saliency maps to remove the background. We expect that eliminating the background from an image could let a feature extractor focus more on a target object, so that, more useful features can be obtained for few-shot image classification. Concretely, when using bounding boxes, we keep the values of RGB channels within a bounding box and set those outside a bounding box as 0, then, the background of processed input images will turn black. Some examples are shown in Figure 5.3. As for using saliency maps, we normalise a saliency map between 0 and 1, then removing the background by multiplying the saliency map with each of the RGB channels of an image. Some processed examples are shown in Figure 5.3. In this approach, we do not need to modify the network architecture or the training of a meta-learning approach.

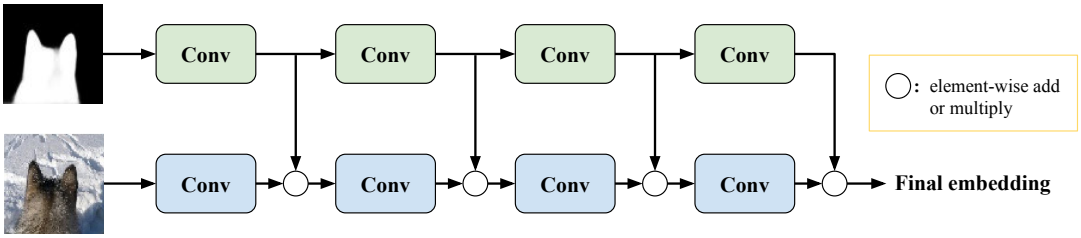


FIGURE 5.4: The workflow of using the respective feature extractor to extract the embedding for an original image and its corresponding auxiliary information. In this example, the auxiliary information is represented by a saliency map. Each rounded rectangle represents a convolutional layer consisting of a convoutional operation, a BN layer, an activation function and a pooling operation.

(c) Using Respective Feature Extractor

The previous two methods both leverage auxiliary information in the original image space. The information may be forgotten after passing several convolutional layers during feature extraction. To keep utilising auxiliary information to assist few-shot image classification during feature extraction, inspired by (Figueroa-Flores et al., 2021), we can also use two separate CNNs based feature extractors to learn the respective embedding of an original image and its auxiliary information, and let the embeddings interact during feature extraction. We assume that the embedding of an image can keep being refined by auxiliary information in each convolutional layer, so that, more target object related features can be focused. Specifically, in each convolutional layer, we add or multiply the embedding of an original image with that of corresponding auxiliary information as displayed in Figure 5.4. It is noteworthy that compared to additional channel and removing background, this method would increase the complexity of network architecture and introduce much more trainable parameters. However, it deserves to be explored if this method could provide more improvements.

(d) Multi-Task Learning based on U-Net

The previous methods all treat auxiliary information as an input of few-shot image classification, and leverage it as an additional input signal or to refine an input/embedding. Different from them, we could also utilise auxiliary information as a supervisory signal to teach a feature extractor how to concentrate on foregrounds. This can be achieved by a multi-task learning framework to perform few-shot image classification and edge/bounding box/saliency map

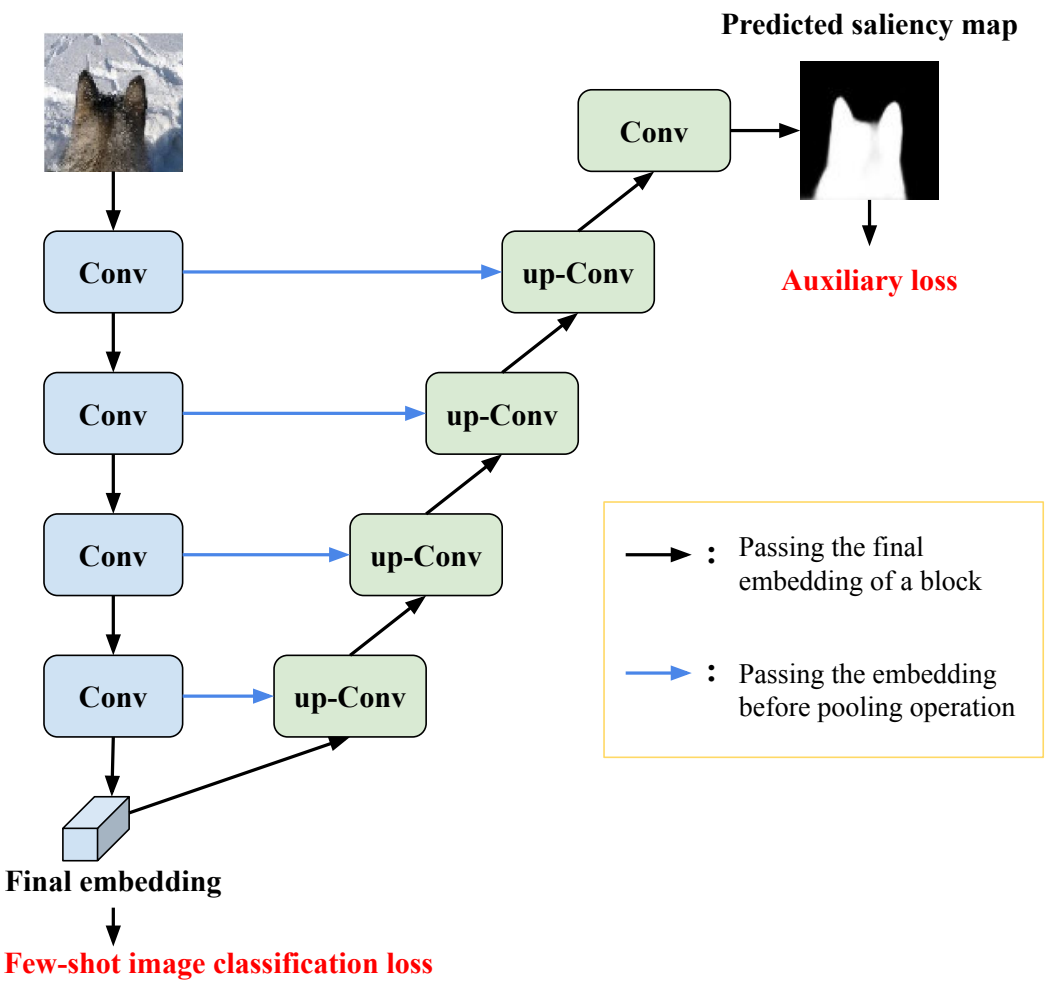


FIGURE 5.5: The framework of our U-Net based multi-task learning. In this example, the auxiliary information is represented by a saliency map. Conv represents a convolutional layer. up-Conv denotes an up-Conv block illustrated in Figure 5.6. The pipeline (shown in blue) on the left side conducts few-shot image classification, while the pipeline (shown in green) on the right side performs the task of saliency map prediction.

prediction at the same time. The approach is inspired by the work in (Zhuang et al., 2019), which conduct image classification and bounding box prediction simultaneously. The underlying assumption of this method is that the additional task of predicting auxiliary information (the green pipeline on right in Figure 5.5) could force a feature extractor (the blue Conv blocks on left in Figure 5.5) to preserve more information of localising an target object, which may potentially help to extract more relevant features for few-shot image classification. Compared to the previous methods, this approach only utilises images

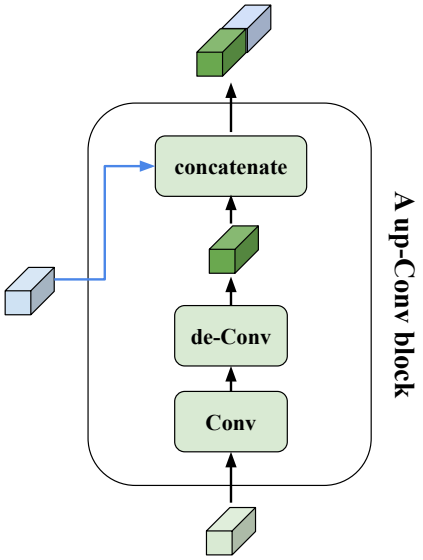


FIGURE 5.6: The pipeline of an up-Conv block in Figure 5.5. Each up-Conv block takes the embedding from its previous block as an input and performs a convolution operation on it. Then upsampling is conducted based on a deconvolution operation. Finally, the upsampled embedding and the corresponding embedding from the feature extraction process are concatenated, serving as the output of each up-Conv block.

to conduct a few-shot image classification task and do not need to extract auxiliary information during meta-testing , which would save more time. Specifically, during meta-training, we treat auxiliary information as an additional training signal to assist classification training signals, so that, the total training loss can be computed as follows,

$$L_{total} = L_{cls} + \alpha * L_{aux} \tag{5.1}$$

L_{cls} and L_{aux} are the training losses of few-shot image classification and auxiliary information prediction, respectively, α is a hyper-parameter controlling the influence of auxiliary information prediction. The workflow of this method is shown in Figure 5.5. It can be seen that we employ a U-Net-like architecture to learn to predict auxiliary information, such as a saliency map. U-Net is a popular and effective architecture in image segmentation (Ronneberger, Fischer, and Brox, 2015) and image generation field (Esser, Sutter, and Ommer, 2018), etc. We choose it as our auxiliary information predictor and modify its architecture to be compatible with our backbones used in few-shot image classification. As shown in Figure 5.5, the left side of the architecture is our 4Conv backbone used for image classification and the right side focuses on auxiliary

prediction tasks. During meta-testing, we only use the left side backbone to extract features from original images and perform few-shot image classification based on a meta-learning method.

5.3 Experiments

This section presents the experimental settings for this chapter. First, three widely used data sets are introduced in Section 5.3.1. Then, the network architecture is illustrated and selected comparison methods are presented in Section 5.3.2. Finally, we describe the experimental setting and evaluation in Section 5.3.3.

5.3.1 Data Sets

As before, we evaluate our method on three widely studied few-shot learning data sets, miniImageNet, tieredImageNet and CUB. The details of these data sets can be found in Section 2.4.2. Since the EGNet is retrained based on resized low-resolution images, we provided details of the used DUTS data set as follows,

- **DUTS**: DUTS (Wang, Ramanan, and Hebert, 2017) is the largest salient object detection benchmark, which includes 10,553 training images and 5,019 testing images. Most images are challenging with various locations and scales. We pre-train EGNet based on this data set. All the images are resized to 84×84 to keep them the same size as those in data sets for few-shot image classification. Note that there are a few overlappings, 245 images, between the meta-testing set of miniImageNet and the training set of DUTS. To strictly follow the meta-learning setting that the meta-testing samples should be unseen, we exclude those overlapping images from DUTS when pre-training EGNet for miniImageNet.

5.3.2 Comparison Algorithms and Network Architecture

To compare the performance of the different types of auxiliary information based on different avenues, we choose the same network architecture as last chapter, 4Conv and Res12. More details of these two backbones can be found in Section 2.4.5 and 4.3.2.

After obtaining the most suitable auxiliary information and best way to leverage it, we further incorporate them into existing few-shot learning approaches

to demonstrate improvements and compare with state-of-the-art. Similar to the last chapter, we select representative state-of-the-art methods based on the same network architecture for comparison. We also include several recent methods using deeper networks, which can further demonstrate the superiority of our Ada-P module. Concretely, for 4Conv based feature extractor, we choose MAML (Finn, Abbeel, and Levine, 2017), A2P (Qiao et al., 2018), MetaOptNet (Lee et al., 2019), R2D2 (Bertinetto et al., 2019) from fast parameterisation based approaches, MetaGAN+RN (Zhang et al., 2018b), SalNet (Zhang, Zhang, and Koniusz, 2019) from data generation based approaches, ProtoNet (Snell, Swersky, and Zemel, 2017), RN (Sung et al., 2018), L2AE-D (Song et al., 2021), TPN (Liu et al., 2019b), GNN (Garcia and Bruna, 2018) from metric learning approaches. For Res12 based feature extractor, we compare with LEO (Rusu et al., 2019), A2P (Qiao et al., 2018), MetaOptNet (Lee et al., 2019), MTL (Sun et al., 2019a) from fast parameterisation based approaches, SNAIL (Mishra et al., 2018), TADAM (Oreshkin, López, and Lacoste, 2018), DC (Lifchitz et al., 2019), CTM (Li et al., 2019b), ProtoNet (Snell, Swersky, and Zemel, 2017) from metric learning approaches. Note that we only compare against methods that provide results on a given data set, therefore the selected comparison algorithms on different data sets can be different.

Since our goal of using auxiliary information for few-shot image classification is to mitigate its intrinsic uncertainty of background clutters, our approach can also be seen as a spatial attention strategy. Similar to the last chapter, we compare our method with several advanced spatial attention methods, namely SCA (Chen et al., 2017), CBAM (Woo et al., 2018), Residual-AT (Wang et al., 2017), Interpret-SA (Meng et al., 2019), L2-pay-AT (Jetley et al., 2018) from the computer vision field.

5.3.3 Experimental Setting and Evaluation

The experimental settings are nearly the same as those in the last chapter. We evaluate our method on 5-way 1-shot, 5-way 5-shot learning tasks on all three data sets. Like most approaches based on 4Conv (Snell, Swersky, and Zemel, 2017; Finn, Abbeel, and Levine, 2017; Liu et al., 2019b), we train our 4Conv based approach in episodic manner for a fair comparison as introduced in Section 2.4.3. For training Res12 based approaches, we adopted large scale training strategy described in Section 2.4.3 following (Sun et al., 2019a; Ye et al., 2020; Rusu et al., 2019; Li et al., 2019b) for better performance. During meta-testing, we evaluate our approach on 6,000 randomly sampled tasks for all

three data sets. The average classification accuracy and 95% confidence interval are reported. All experiments are performed using TensorFlow (Abadi et al., 2016) on a Titan Xp GPU. More details regarding to model training can be found in Section 4.3.3.

5.4 Analysis of Results

In this section, we analyse the results obtained from different experimental studies. Specifically, our aims are:

- To provide a thorough exploration of using different methods to leverage different types of auxiliary information to assist few-shot image classification based on various data sets (Section 5.4.1).
- To demonstrate the improvements and flexibility of incorporating the most suitable auxiliary information and the best method to leverage it into a few representative approaches for few-shot image classification, and to test the compatibility of this approach with our Ada-P module (Section 5.4.2).
- To check whether the superiority of our few-shot learning approach that leverages auxiliary information with Ada-P module is maintained on various data sets based on both shallow and deep models (Section 5.4.3).
- To illustrate the necessity of leveraging auxiliary information to perform effective spatial attention for few-shot image classification, comparing the performance of our approach against a few advanced spatial attention methods (Section 5.4.4).

5.4.1 Leveraging Different Types of Auxiliary Information

The experiment presented here provides a thorough exploration of utilising different types of auxiliary information for few-shot image classification. The same as before, we choose ProtoNet as the baseline. We test three types of auxiliary information, namely, edges, bounding boxes and saliency maps. To incorporate these types of information into a few-shot image classification approach, we evaluate four different methods, (a) adding additional channel, (b) removing background, (c) using respective feature extractors, (d) multi-task learning based on U-Net, as presented in Section 5.2.2. Note that, the interaction between the embeddings from two feature extractors can be element-wise

addition or multiplication. Here, we denote method (c) with element-wise addition as (c) w/ add, and method (c) with element-wise multiplication as (c) w/ mul. We compare these methods combined with different types of auxiliary information on 5-way 1-shot and 5-shot classification tasks on various data sets using 4Conv backbone. The experimental results are shown in Table 5.1, reflecting the following observations:

TABLE 5.1: Results of leveraging various types of auxiliary information for few-shot image classification on miniImageNet, tieredImageNet and CUB data sets. The average accuracy (%) with 95% confidence intervals is reported. **Aux** denotes the type of auxiliary information. Bbox and sal represent bounding box and saliency map, respectively. The best and second best performing results of each task are highlighted in bold and underlined, respectively.

		miniImageNet 5-way		**tieredImageNet 5-way**		**CUB 5-way**	
Methods	**Aux**	**1-shot**	**5-shot**	**1-shot**	**5-shot**	**1-shot**	**5-shot**
ProtoNet$^+$	none	52.34±0.26	69.90±0.20	52.85±0.27	71.22±0.23	57.12±0.30	74.51±0.18
ProtoNet$^+$	edge	52.03±0.25	69.73±0.20	52.79±0.27	71.32±0.23	56.94±0.30	74.63±0.18
w/ (a)	bbox	52.71±0.26	69.97±0.20	53.12±0.27	71.74±0.23	57.06±0.30	75.11±0.18
	sal	**55.30±0.26**	72.77±0.20	56.47±0.27	74.83±0.23	66.41±0.30	81.86±0.18
ProtoNet$^+$	bbox	44.00±0.24	60.98±0.21	43.44±0.27	62.83±0.23	52.85±0.30	70.47±0.18
w/ (b)	sal	43.49±0.24	67.65±0.20	49.93±0.27	68.93±0.23	62.81±0.30	79.33±0.18
ProtoNet$^+$	edge	51.69±0.25	68.82±0.20	53.53±0.27	71.79±0.23	53.56±0.30	70.48±0.18
w/ (c) add	bbox	52.62±0.26	69.98±0.20	53.08±0.27	72.20±0.23	57.73±0.30	75.02±0.18
	sal	55.00±0.25	**72.77±0.19**	**56.69±0.27**	**74.84±0.23**	**66.77±0.30**	**82.78±0.18**
ProtoNet$^+$	edge	50.55±0.26	69.90±0.21	50.66±0.27	69.51±0.23	52.32 ±0.30	69.92 ±0.18
w/ (c) mul	bbox	49.05±0.25	66.17±0.20	48.42±0.27	67.01±0.23	50.12 ±0.30	68.22 ±0.18
	sal	53.68±0.26	71.85±0.20	55.18±0.28	73.56±0.23	64.85±0.30	82.21±0.18
ProtoNet$^+$	edge	52.45±0.26	69.68±0.20	53.10±0.27	71.63±0.23	57.42±0.30	75.18±0.18
w/ (d)	bbox	52.65±0.26	69.92±0.20	52.65±0.27	71.34±0.23	57.47±0.30	74.68±0.18
	sal	52.89±0.26	70.31±0.20	53.15±0.27	71.62±0.23	58.24±0.30	75.12±0.18

- Comparing the performance of different types of auxiliary information in each method, we can see that saliency map achieves the best performance on most of the few-shot image classification tasks when using different methods to incorporate auxiliary information. It helps the ProtoNet baseline to be improved by most of methods other than (b). Especially based on method (a) and (c) add, saliency maps improve the baseline by a large margin, around 3% on miniImageNet and tieredImageNet, and 8% on CUB. Compared to saliency maps, edges and bounding boxes do not display clear positive effects on few-shot image classification tasks. They improve the baseline on some tasks. For example, using bounding boxes via (c) add improve upon ProtoNet by around 1% on 5-shot tasks of tieredImageNet, and utilising edges via (d) enhance the baseline by

around 0.5% on both 1-shot and 5-shot tasks on CUB. However, these improvements are not fully consistent on different data sets. Based on the above analysis, we conclude that the most beneficial auxiliary information for few-shot image classification in our study is the saliency map.

- Comparing the performance of using different methods to incorporate auxiliary information, we can find that (a) adding additional channel and (c) using respective feature extractor with addition operation outperform other methods on most of tasks, especially when utilising the most suitable auxiliary information, saliency maps, the performance of (a) and (c) increases upon the ProtoNet baseline by a large margin. Method (b) is not a good choice for few-shot image classification, since it produces clear marked drops in performance on most of tasks and data sets, likely due to the fact that the feature extractor focuses more on the outline rather than the texture of a target object. Method (d) provides slight improvements when using different types of auxiliary information for both 1-shot and 5-shot tasks on all three data sets. However, these improvements are so limited compared to those obtained by methods (a) and (c) add leveraging the auxiliary information of saliency maps. To this end, we select methods (a) and (c) add as the most beneficial avenues to leverage auxiliary information for few-shot image classification.

Based on the above observations and analysis, we have found the most beneficial auxiliary information and the best ways to leverage it for few-shot image classification. As a next step, we aim to incorporate them into existing few-shot learning approaches to demonstrate improvements and compare with state-of-the-art. Note that methods (a) and (c) add achieve very similar performance of both 1-shot and 5-shot tasks on all data sets. Since method (c) add introduces much more trainable parameters than method (a), we choose the method (a) as our method to incorporate auxiliary information in the next stage. To apply our findings above into existing few-shot image classification approaches, we define a Saliency Object Detection (SOD) module, as shown in Figure 5.7. This module employs a pre-trained SOD model (e.g. EGNet) to extract the saliency map $s_{i,j}$ of an image $x_{i,j}$. The extracted saliency map serves as an additional channel in the original image space alongside the RGB channels, which will be fed into the feature extractor as a whole $[x_{i,j}; s_{i,j}]$, $[;]$ representing concatenation along the last dimension.

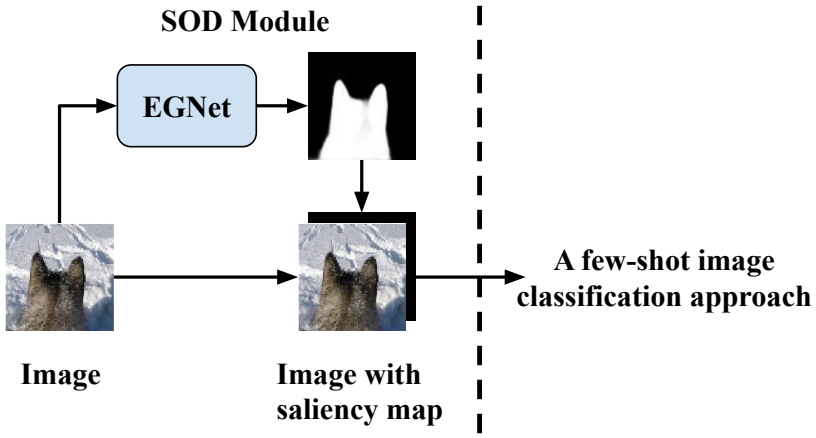


FIGURE 5.7: The workflow of our SOD module.

5.4.2 Incorporating Auxiliary information into Representative Approaches

To verify the effectiveness and flexibility of leveraging auxiliary information for few-shot image classification, we incorporate previously defined SOD module into a few representative few-shot learning approaches. In order to test the compatibility of our SOD and Ada-P modules, we further introduce both of them into existing methods to see if the combination would lead to a further improvement. Concretely, we choose MAML (Finn, Abbeel, and Levine, 2017), RN (Sung et al., 2018), ProtoNet (Snell, Swersky, and Zemel, 2017), TPN (Liu et al., 2019b), because they are representative approaches from different few-shot learning branches and also include transductive and inductive methods. We perform a comparison on miniImageNet using the 4Conv backbone. For ProtoNet, we use an enhanced version with dropblock and meta-batch training strategy. For the other methods, we do not add any training strategies described in Section 5.3.2, such as dropblock and augmentation. We reuse their released code and incorporate our SOD with and without Ada-P module, strictly following their respective experimental setting. Table 5.2 shows the comparison results.

It can be seen that our SOD module improves all the methods on 1-shot and 5-shot tasks, especially for MAML, which improves significantly by around 6% on 1-shot tasks and 4% on 1-shot tasks. This experiment verifies our SOD module is beneficial for and compatible with various types of few-shot learning frameworks. It is noteworthy that compared to merely adding Ada-P module, our SOD module improves all methods more than Ada-P module. This is probably because the SOD module leverages auxiliary information to guide

few-shot image classification, while Ada-P is only trained from a few-shot learning data set. The auxiliary information can be seen as prior knowledge that provides more spatial information of an object, therefore, the SOD module generally performs better than Ada-P. When our SOD and Ada-P modules are incorporated together, we see that most of the tasks demonstrate further improvements compared to solely adding one of them. Since they can both be seen as a spatial attention mechanism, their positive influence on performance may overlap a bit. However, the SOD module is placed before the feature extractor to provide prior knowledge and Ada-P module is located in each convolutional layer to refine embeddings. Besides, the SOD module is pre-trained based on a SOD data set and Ada-P is trained with the feature extractor based on few-shot learning data set in an end-to-end manner. They should play a different role in few-shot learning, which is reflected by the visualisation in Figure 5.8 and 5.9, and we will provide more discussions in Section 5.4.4.

TABLE 5.2: Results after incorporating Ada-P and/or SOD into several existing approaches on miniImageNet. The last number of **Archt.** stands for the number of filters in each convolutional layer. The average accuracy (%) with 95% confidence intervals is reported. ↑ shows the improvements after incorporating Ada-P and/or SOD. $^+$ represents an enhanced version of ProtoNet.

		miniImageNet 5-way	
Methods	**Archt.**	**1-shot(%)**	**5-shot(%)**
MAML	4Conv-32	48.70 ± 1.84	63.74 ± 0.92
MAML w/ Ada-P	4Conv-32	50.74 ± 1.82 ↑ 2.04	66.85 ± 0.87 ↑ 3.11
MAML w/ SOD	4Conv-32	54.56 ± 1.82 ↑ 5.86	68.12 ± 0.87 ↑ 4.38
MAML w/ SOD & Ada-P	4Conv-32	55.76 ± 1.82 ↑ 7.06	70.11 ± 0.87 ↑ 6.37
RN	4Conv-64	50.44 ± 0.82	65.32 ± 0.70
RN w/ Ada-P	4Conv-64	50.95 ± 0.86 ↑ 0.51	66.46 ± 0.69 ↑ 1.14
RN w/ SOD	4Conv-64	54.02 ± 0.86 ↑ 3.58	67.81 ± 0.69 ↑ 2.49
RN w/ SOD & Ada-P	4Conv-64	53.95 ± 0.86 ↑ 3.51	68.59 ± 0.69 ↑ 3.27
ProtoNet$^+$	4Conv-64	52.34 ± 0.26	69.90 ± 0.18
ProtoNet$^+$ w/ Ada-P	4Conv-64	54.75 ± 0.26 ↑ 2.41	71.63 ± 0.20 ↑ 1.73
ProtoNet$^+$ w/ SOD	4Conv-64	55.00 ± 0.26 ↑ 2.66	72.77 ± 0.20 ↑ 2.87
ProtoNet$^+$ w/ SOD & Ada-P	4Conv-64	57.29 ± 0.26 ↑ 4.95	74.60 ± 0.20 ↑ 4.70
TPN	4Conv-64	53.75 ± 0.86	69.43 ± 0.68
TPN w/ Ada-P	4Conv-64	55.19 ± 0.86 ↑ 1.44	70.90 ± 0.69 ↑ 1.47
TPN w/ SOD	4Conv-64	56.74 ± 0.86 ↑ 2.99	72.53 ± 0.64 ↑ 3.10
TPN w/ SOD & Ada-P	4Conv-64	57.08 ± 0.86 ↑ 3.33	73.68 ± 0.65 ↑ 4.25

5.4.3 Comparisons with the State-of-the-Art Approaches

This section presents comparisons with the state-of-the-art approaches based on our SOD module. The evaluation is performed on 5-way 1-shot and 5-shot classification tasks based on various data sets using both 4Conv and Res12 models. As before, we choose ProtoNet as a backbone to incorporate the SOD module for its simplicity and effectiveness. The results and analysis are presented as follows:

Results on miniImageNet: We compare our approach with the state-of-the-art methods from different branches. The results on miniImageNet using both shallow and deep backbones are shown in Table 5.3. We can see that, based on the 4Conv feature extractor, ProtoNet$^+$ w/ SOD surpasses all the selected state-of-the-art methods on 5-shot tasks. When further incorporating our Ada-P module, our approach achieves state-of-the-art performance on 5-shot tasks and a comparable result to the best performance on 1-shot tasks. Note that SalNet (Zhang, Zhang, and Koniusz, 2019) also utilised SOD in their approach, however, they incorporated a more complex network to mix foregrounds and backgrounds, introducing more convolutional layers, increasing the complexity when incorporating SOD beyond our approach. Instead, we perform a simpler method that adds a saliency map as an additional channel in image space and incorporates lightweight Ada-P modules, achieving similar performance on 1-shot tasks and a better result on 5-shot tasks with much less learnable parameters and computational complexity. Based on the Res12 feature extractor, our SOD module improves upon ProtoNet$^+$ by around 1.0% on both 1-shot and 5-shot tasks. Since our approach is built upon ProtoNet$^+$ under the same experimental setting, these improvements verify the effectiveness of using auxiliary information for few-shot image classification when using a deeper model. Besides, when combining our SOD and Ada-P module, our approach achieves the second best performance on both 5-way 1-shot and 5-shot tasks. It is noteworthy that these methods are not strictly comparable since their network architectures are not exactly the same. For example, the best performing method, CTM, use ResNet-18 backbone, which represents a deeper architecture (11 million parameters) with around 37% more parameters than our Res12 model (8 million parameters). We apply the architecture that the majority of previous methods used and achieve competitive performance on both 1-shot and 5-shot tasks compared to the state-of-the-art results based on this backbone.

Results on tieredImageNet: As shown in Table 5.3, we compare our method

with state-of-the-art methods that conduct few-shot evaluations on tieredImageNet. Note that the missing values in Table 5.3 indicate the methods did not provide testing results on tiredImageNet. Based on both 4Conv and Res12 model, our SOD module improves upon the ProtoNet$^+$ baseline, which demonstrate it is beneficial for few-shot image classification. Also, it achieves the best performance on 5-shot tasks using 4Conv model and 1-shot tasks using Res12 backbone, comparing to the selected promising approaches. When further incorporating our Ada-P module, our approach achieves the state-of-the art performance on most of the tasks using either shallow or deep backbones, except 1-shot tasks based on 4Conv model. In this scenario, the best performing method is TPN, which is a transductive method using a few unlabelled examples to assist learning, while our method is an inductive approach that only uses labelled training examples to predict test examples.

Results on CUB: Table 5.4 summarises the comparison of our method and other selected few-shot learning approaches using both 4Conv and Res12 backbones on CUB data set. As before, we do not include the recent methods that builds on a much deeper backbone, such as Res18 or WRN28, in our comparison for fairness. From Table 5.4, we can see that ProtoNet$^+$ w/ SOD surpasses all the other methods by a large margin based on both shallow and deep backbones on both 1-shot and 5-shot tasks. Our SOD and Ada-P modules achieve further improvements when collaborating with each other, which demonstrates the Ada-P and SOD are effective for fine-grained few-shot classification tasks.

5.4.4 Comparisons with Spatial Attention Methods

The same as Ada-P module in the last chapter, our SOD module can also be seen as a spatial attention mechanism. Therefore, we compare our SOD module with a few recent representative spatial attention methods on few-shot image classification problems to illustrate the novelty of our SOD module, as we do in Section 4.4.5. The selected advanced spatial attention methods are SCA (Chen et al., 2017), CBAM (Woo et al., 2018), Residual-AT (Wang et al., 2017), Interpret-SA (Meng et al., 2019), L2-pay-AT (Jetley et al., 2018) from the computer vision field. It is noteworthy that these methods are not specifically designed for few-shot learning, some of them need to be tweaked a bit to be compatible with the few-shot learning framework. Since both of our Ada-P and SOD modules can be seen as spatial attention mechanisms, here, we further compare SOD with our Ada-P module and combine them

TABLE 5.3: Results on miniImageNet and tieredImageNet. The average accuracy (%) with 95% confidence intervals is reported. The best and second best performing results are highlighted in bold and underlined, respectively. Res12 represents Res12 models and the behind number stands for the number of filters in the last residual block. WRN represents wide residual networks. Res18 represents ResNet-18 models. † uses the results of MetaOptNet with SVM trained only on the meta-training set. $^+$ represents an enhanced version of ProtoNet.

		miniImageNet 5-way		**tieredImageNet 5-way**	
Methods	**Archt.**	**1-shot(%)**	**5-shot(%)**	**1-shot(%)**	**5-shot(%)**
MAML	4Conv-32	48.70±1.84	63.74±0.92	51.76±1.81	70.30±1.75
RN	4Conv-64	50.44±0.82	65.32±0.70	54.48±0.93	71.32±0.78
MetaGAN+RN	4Conv-64	52.71±0.64	68.63±0.67		
L2AE-D	4Conv-64	53.85±0.85	70.16±0.65	55.81±0.85	71.71±0.65
TPN	4Conv-64	53.75±0.86	69.43±0.68	57.53±0.96	72.85±0.74
A2P	4Conv-64	54.53±0.40	67.87±0.20		
SalNet	4Conv-64	**57.45±0.88**	72.01±0.67		
ProtoNet$^+$	4Conv-64	52.34±0.26	69.90±0.18	52.44±0.27	70.91±0.23
ProtoNet$^+$ w/ SOD	4Conv-64	55.00±0.26	72.77±0.20	56.69±0.28	74.65±0.23
ProtoNet$^+$ /w Ada-P & SOD	4Conv-64	57.29±0.26	**74.60±0.20**	**58.40±0.28**	**76.06±0.23**
SNAIL	Res12-256	55.71±0.99	68.88±0.92		
TADAM	Res12-512	58.50±0.30	76.70±0.30		
A2P	WRN28	59.60±0.41	73.74±0.19		
LEO	WRN28	61.76±0.08	77.59±0.12	66.33±0.05	81.44±0.09
MTL	Res12-512	61.20±1.80	75.50±0.80	62.83±1.80	74.50±0.92
DC	Res12-512	62.53±0.19	79.77±0.19		
MetaOptNet†	Res12-640	62.64±0.61	78.64±0.46	65.99±0.72	81.56±0.53
CTM	Res18	**64.12±0.82**	**80.51±0.13**	68.41±0.39	84.28±1.73
ProtoNet$^+$	Res12-512	61.27±0.26	77.79±0.18	67.95±0.30	83.30±0.21
ProtoNet$^+$ w/ SOD	Res12-512	62.79±0.27	79.37±0.19	68.95±0.30	84.23±0.21
ProtoNet$^+$ w/ Ada-P & SOD	Res12-512	63.29±0.27	80.10±0.19	**70.28±0.30**	**84.92±0.21**

together to see if a target object can be paid more attentions. The same as before, we choose ProtoNet as a baseline and incorporate each spatial attention method into it respectively. To compare the meta-testing performance, we train a 4Conv backbone on miniImageNet data set in an episodic manner. The results are shown in Table 5.5, where we can observe that simply incorporating the selected spatial attention methods into few-shot learning does not improve too much. This demonstrates designing an effective spatial attention method for few-shot learning is a nontrivial task. From Table 5.5, we can also see that our SOD module outperforms other spatial attention methods, including our Ada-P module, on both 1-shot and 5-shot tasks. This is likely because the SOD module leverages auxiliary information that provides more spatial

TABLE 5.4: Results on CUB. The average accuracy (%) with 95% confidence intervals is reported. The best and second best performing results are highlighted in bold and underlined, respectively. $^{+}$ represents an enhanced version of ProtoNet. * represents results from (Chen et al., 2018).

		CUB 5-way	
Methods	**Archt.**	**1-shot(%)**	**5-shot(%)**
MatchingNet*	4Conv-64	60.52 ± 0.88	75.29 ± 0.75
MAML*	4Conv-64	54.73 ± 0.97	75.75 ± 0.76
RN*	4Conv-64	62.34 ± 0.94	77.84 ± 0.68
ProtoNet^{+}	4Conv-64	57.61 ± 0.29	74.51 ± 0.18
ProtoNet^{+} w/ SOD	4Conv-64	$\underline{66.41 \pm 0.30}$	$\underline{81.86 \pm 0.18}$
ProtoNet^{+} w/ Ada-P & SOD	4Conv-64	$\mathbf{69.29 \pm 0.30}$	$\mathbf{83.86 \pm 0.18}$
MatchingNet*	Res10-512	71.29 ± 0.87	83.47 ± 0.58
MAML*	Res10-512	70.32 ± 0.99	80.93 ± 0.71
RN*	Res10-512	70.47 ± 0.99	83.70 ± 0.55
ProtoNet^{+}	Res12-512	68.60 ± 0.26	85.51 ± 0.20
ProtoNet^{+} w/ SOD	Res12-512	$\underline{72.47 \pm 0.26}$	$\underline{88.03 \pm 0.20}$
ProtoNet^{+} w/ Ada-P & SOD	Res12-512	$\mathbf{73.65 \pm 0.27}$	$\mathbf{88.39 \pm 0.20}$

information, while our Ada-P and other methods are only trained from a few-shot learning data set. It can also be seen that combining both of our Ada-P and SOD modules surpasses other methods and themselves by a large margin. This illustrates our Ada-P and SOD modules are compatible with each other and especially effective for few-shot image classification.

To better show how these spatial attention methods influence the feature extraction, we visualise the embedding space in the first two convolutional layers of a few samples as shown in Figure 5.8 and 5.9. The same as before, we compress the multiple channels into a single one by mean operation. From Figure 5.8, we can see that in the first layer the SOD module helps to outline the object from the background, while the Ada-P module preserves more details of the object. When incorporating both of them, the embeddings well preserve the shape and texture of the object and suppress the background noise. Other spatial attention methods either include more background noise or focus too much on a specific feature on the object, such as the eyes of lions. In the second layer, as shown in Figure 5.9, our Ada-P module suppresses the background noise and the SOD module still helps to emphasise the outline of the objects. Compared to other spatial attention methods, our Ada-P and SOD modules assist the feature extractor to focus more on the objects. We can clearly recognise the target object in the mean feature maps using our Ada-P and SOD modules while the embeddings of other methods look more blurred.

TABLE 5.5: Comparisons with several spatial attention methods on few-shot learning problems. **Archt.** represents the architecture of the feature extractor. The last number of **Archt.** stands for the number of filters in each convolutional layer. The best and second best performing results are highlighted in bold and underlined, respectively. The average accuracy (%) with 95% confidence intervals are reported. $^+$ represents an enhanced version of ProtoNet.

		miniImageNet 5-way	
Methods	**Archt.**	**1-shot(%)**	**5-shot(%)**
ProtoNet$^+$	4Conv-64	52.34 $\pm$ 0.26	69.90 $\pm$ 0.18
ProtoNet$^+$ w/ SCA	4Conv-64	51.45 $\pm$ 0.25	68.83 $\pm$ 0.18
ProtoNet$^+$ w/ CBAM	4Conv-64	53.54 $\pm$ 0.26	70.19 $\pm$ 0.18
ProtoNet$^+$ w/ Residual-AT	4Conv-64	51.91 $\pm$ 0.26	70.35 $\pm$ 0.18
ProtoNet$^+$ w/ Interpret-SA	4Conv-64	53.70 $\pm$ 0.26	71.18 $\pm$ 0.19
ProtoNet$^+$ w/ L2-pay-AT	4Conv-64	52.52 $\pm$ 0.26	69.97 $\pm$ 0.18
ProtoNet$^+$ w/ Ada-P	4Conv-64	54.75 $\pm$ 0.26	71.63 $\pm$ 0.20
ProtoNet$^+$ w/ SOD	4Conv-64	<u>55.00 $\pm$ 0.26</u>	<u>72.77 $\pm$ 0.20</u>
ProtoNet$^+$ w/ SOD & Ada-P	4Conv-64	**57.29 $\pm$ 0.26**	**74.60 $\pm$ 0.20**

Based on the above analysis, we can conclude that directly incorporating other spatial attention methods into few-shot image classification may not be a good choice, while our SOD module is especially effective for few-shot image classification and can provide further improvements when combined with our Ada-P module, which demonstrates our contributions and novelty.

5.5 Incorporating SOD and Ada-P into L2AE-D

To illustrate the overall contribution of this thesis, we combine the three of our research works together and compare with the state-of-the arts on few-shot image classification. As before, we test all methods on various data sets. Note that we only evaluate these methods based on the 4Conv backbone here, since the L2AE-D approach is not suitable for the Res12 model that uses a global average pooling in the end to summarise the embeddings. The comparison results are shown in Table 5.6 and 5.7. To make a fairer comparison and further demonstrate that our Ada-P and SOD module can be beneficial for few-shot image classification on various data sets, we incorporate the Ada-P and SOD modules into a few promising methods and compare the combination of our three research works with them. It can be seen that the comparison results are divided into three parts by dashed lines in both Table 5.6 and 5.7. The upper,

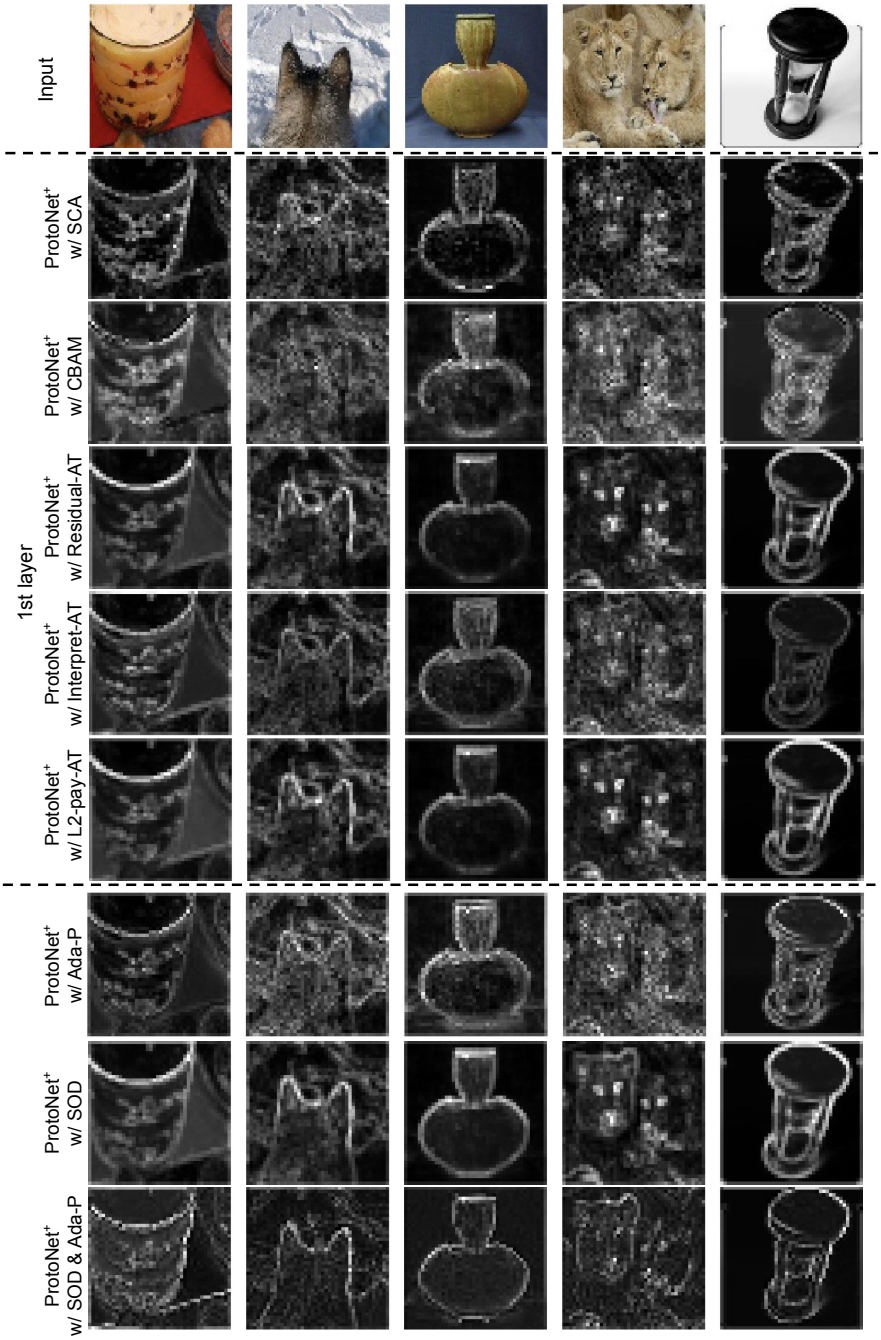


FIGURE 5.8: Visualisation of feature maps by compressing multiple feature maps into a mean one. The first row shows five input images. The Figures between dash lines represent the embeddings of different spatial attention methods in the first convolutional layer.

middle and below parts of the tables show the results of a few promising approaches, selected methods incorporated with the Ada-P and SOD modules and the combinations of our three methods, on few-shot image classification, respectively. It can be seen that the combination of our three works achieves

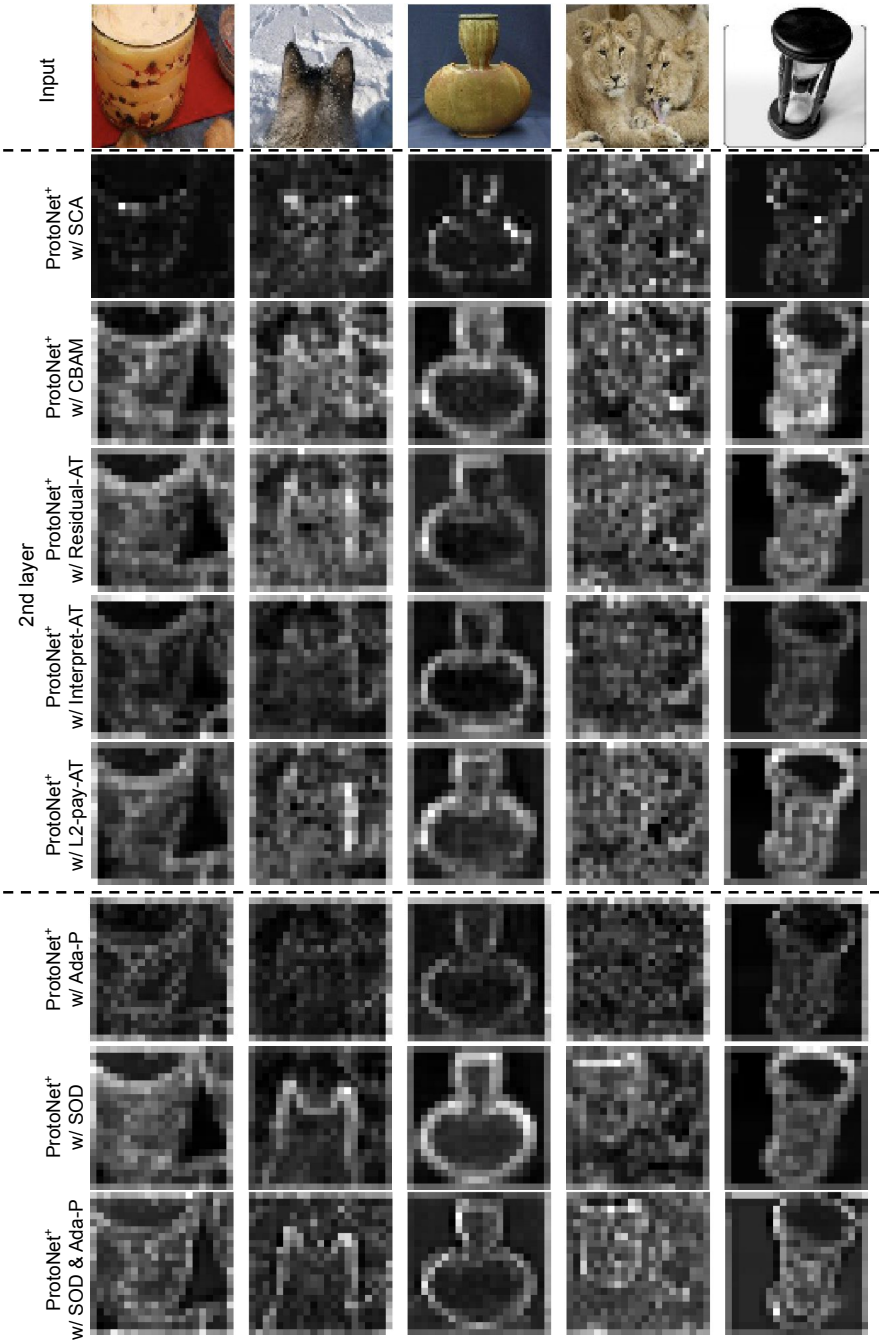


FIGURE 5.9: Visualisation of feature maps by compressing multiple feature maps into a mean one. The first row shows five input images. The Figures between dash lines represent the embeddings of different spatial attention methods in the second convolutional layer.

the state-of-the-art performance on most of the 5-way 1-shot and 5-shot tasks across three different data sets. This demonstrates the superiority of our research works in few-shot image classification field. In addition, looking at the

TABLE 5.6: Results on miniImageNet and tieredImageNet. The average accuracy (%) with 95% confidence intervals is reported. The best and second best performing results are highlighted in bold and underlined, respectively. **Archt.** represents the architecture of the feature extractor. The last number of **Archt.** stands for the number of filters in each convolutional layer. $^{+}$ represents an enhanced version of ProtoNet.

		miniImageNet 5-way		**tieredImageNet 5-way**	
Methods	**Archt.**	**1-shot(%)**	**5-shot(%)**	**1-shot(%)**	**5-shot(%)**
MAML	4Conv-32	48.70±1.84	63.74±0.92	51.76±1.81	70.30±1.75
RN	4Conv-64	50.44±0.82	65.32±0.70	54.48±0.93	71.32±0.78
MetaGAN+RN	4Conv-64	52.71±0.64	68.63±0.67		
TPN	4Conv-64	53.75±0.86	69.43±0.68	57.53±0.96	72.85±0.74
A2P	4Conv-64	54.53±0.40	67.87±0.20		
SalNet	4Conv-64	<u>57.45±0.88</u>	72.01±0.67		
ProtoNet^{+}	4Conv-64	52.34±0.26	69.90±0.18	52.44±0.27	70.91±0.23
MAML w/ Ada-P & SOD	4Conv-32	55.76±1.82	70.11±0.87	56.53±1.88	73.31±0.92
RN w/ Ada-P & SOD	4Conv-64	53.95±0.86	68.59±0.69	58.24±0.88	74.48±0.68
TPN w/ Ada-P & SOD	4Conv-64	57.08±0.86	<u>73.68±0.65</u>	<u>59.93±0.95</u>	75.76±0.76
ProtoNet^{+} w/ Ada-P & SOD	4Conv-64	57.29±0.26	**74.60±0.20**	58.40±0.28	**76.06±0.23**
L2AE-D	4Conv-64	53.85±0.85	70.16±0.65	55.81±0.85	71.71±0.65
L2AE-D w/ Ada-P	4Conv-64	55.65±0.85	70.91±0.65	57.55±0.85	72.76±0.65
L2AE-D w/ SOD	4Conv-64	56.40±0.85	72.16±0.65	59.05±0.85	74.72±0.65
L2AE-D w/ Ada-P & SOD	4Conv-64	**57.48±0.26**	73.55±0.20	**60.03±0.26**	<u>75.98±0.20</u>

TABLE 5.7: Results on CUB. The average accuracy (%) with 95% confidence intervals is reported. The best and second best performing results are highlighted in bold and underlined, respectively. $^{+}$ represents an enhanced version of ProtoNet. * represents results from (Chen et al., 2018).

		CUB 5-way	
Methods	**Archt.**	**1-shot(%)**	**5-shot(%)**
MatchingNet*	4Conv-64	60.52 ± 0.88	75.29 ± 0.75
MAML*	4Conv-64	54.73 ± 0.97	75.75 ± 0.76
RN	4Conv-64	59.40 ± 0.94	72.15 ± 0.68
ProtoNet^{+}	4Conv-64	57.61 ± 0.29	74.51 ± 0.18
MAML w/ Ada-P & SOD	4Conv-64	64.10 ± 1.77	78.08 ± 0.83
RN w/ Ada-P & SOD	4Conv-64	66.11 ± 0.92	77.66 ± 0.65
ProtoNet^{+} w/ Ada-P & SOD	4Conv-64	69.29 ± 0.30	83.86 ± 0.18
L2AE-D	4Conv-64	63.47 ± 0.30	76.51 ± 0.18
L2AE-D w/ Ada-P	4Conv-64	65.14 ± 0.30	78.58 ± 0.18
L2AE-D w/ SOD	4Conv-64	<u>70.46 ± 0.30</u>	<u>82.65 ± 0.18</u>
L2AE-D w/ SOD & SOD	4Conv-64	**73.36 ± 0.30**	**84.09 ± 0.18**

results of incorporating the Ada-P and SOD modules into L2AE-D, we can observe that either of them improves L2AE-D obviously, and introducing both of

them leads to dramatic improvements, on both 1-shot and 5-shot tasks across all evaluated data sets. This illustrates the compatibility of our three research works. From the middle part of Table 5.6 and 5.7, we can also see that the Ada-P and SOD modules improve several selected promising few-shot learning methods by a large margin on all data sets. This further illustrates that the Ada-P and SOD modules can be used as a plug-and-play module in existing methods for better performance on few-shot image classification. We further compare our L2AE-D w/ Ada-P & SOD to those selected promising methods combined with the Ada-P and SOD modules. The results show, after adding the same Ada-P and SOD modules, our L2AE-D still outperforms other representative methods on 1-shot tasks on all data sets and achieves very competitive performance on 5-shot tasks. This further demonstrates that the Ada-P and SOD modules work well with L2AE-D.

5.6 Summary

This chapter has provided a thorough exploration of suitable auxiliary information and beneficial methods to leverage it to mitigate background clutters for few-shot image classification, which has realised our research objective 3. According to our experimental results on three widely used benchmarks, we have found the saliency map is the most suitable auxiliary information and the best way to utilise it is to add it as an additional channel alongside the RGB channels of an image. Based on these, we have designed a SOD module tailored to few-shot image classification, which could be used as a plug-and-play module for various existing few-shot learning approaches. The experimental results have demonstrated the SOD module could improve a few meta-learning approaches and achieve competitive performance compared to the state-of-the-arts. We have also empirically demonstrated that using existing spatial attention methods do not work well in few-shot image classification, and our solution has been shown to address that problem effectively.

To conclude this thesis, in the next chapter we outline the final conclusions for the thesis after the main body of research conducted and summarise the contributions. In addition, a few limitations of the research and some future works are also presented.

Chapter 6

Conclusions

This final chapter summarises the main contributions and several limitations of the research works carried out in this thesis. First, an overview of the contributions of our research works is presented in Section 6.1. Then, we identify a few limitations that are found after summarising our contributions and point out several potential research directions of future works in Section 6.2.

6.1 Summary of Contributions

AI is a cutting-edge research field mainly on the basis of ML and optimisation. Although AI develops quickly, we still have a long way to go before fully realising it. Currently, it suffers from a few limitations, such as relying on human knowledge to tune an algorithm and lack of generalisation ability. To address the issues, ML and optimisation have interacted with each other and themselves frequently. One of the current main limitations of AI, especially for ML techniques, is that existing ML algorithms normally rely on a lot of data to extract knowledge in a time-consuming manner, while humans can quickly learn a new concept from few examples. To fill the gap, researchers delve into a new research field, few-shot learning, aiming to learn effective knowledge from limited training data.

Few-shot image classification is the one of the key problem of few-shot learning, working towards classifying an image into its category based on few examples. The performance of this problems is limited by two main issues, lack of information and intrinsic uncertainties reflected by unrepresentative sample and background clutters. These two problems are usually tackled separately by existing methods. Very few of them consider the two main issues at the same time. This context demands research efforts to develop new methods

that tackle the two core issues simultaneously and achieve promising performance on few-shot image classification tasks.

Since few-shot image classification cannot be simply accomplished by common image recognition algorithms, normally requiring cooperation between methods, first, we have contributed to a thorough global overview of the self and dual interactions between ML and optimisation. Based on this, we delved into few-shot image classification, which is the main body of the research developed in this dissertation. We has contributed to the improvements of meta-learning strategies for few-shot image classification. First, a comprehensive review of the state-of-the-art about few-shot image classification has been completed. Then, three research stages have been developed following a progressive order, in which early findings take part in the final research outcomes. The main contributions achieved through the three stages are revisited in the following paragraphs, where they are also linked to the research objectives established at the beginning of this dissertation (Section 1.2).

- The first contribution has addressed the development of a novel method that sufficiently exploits the limited training data in few-shot image classification (Chapter 3). As far as we know, there is no existing approach that addresses the two core issues of few-shot image classification from the perspective of maximising the utility of limited training samples. We accomplish this by proposing L2AE-D that learns to aggregate embeddings with meta-level dropout. A specially designed channel-wise attention mechanism is learned to assign larger weights to useful feature maps and smaller weights to noisy ones of different embeddings within the same channel, which can sufficiently exploit the limited training embeddings. We also propose a special learning strategy for the extreme cases, one-shot learning tasks, to maximising the use of training data. In addition, a meta-level dropout technique is introduced into several representative few-shot learning approaches including ours and demonstrates improvements. These results have covered the Objective 1 of the thesis: a novel meta-learning approach is developed that is able to reduce the negative influence of unrepresentative samples and meanwhile leverage as much useful information of training samples as possible for few-shot image classification.

- Besides maximising the utility of limited training samples, a second contribution has encompassed the development of a novel strategy that addressed the two key problems of few-shot image classification from the

angle of feature extraction (Chapter 4). Specifically, we designed an adaptive pooling method for few-shot image classification, which uses a meta-learner to learn a proper pooling operator that reduces the loss of useful information when available data is limited. This module is lightweight and can be integrated into varied few-shot learning approaches as a plug-and-play module for further improvements. Besides, this module takes into account the importance of the features at different spatial locations, which can pay more attention to the salient regions and mitigate background clutters. We have empirically demonstrated that directly introducing off-the-shelf spatial attention methods into few-shot image classification is not very helpful, and our Ada-P module is carefully designed and especially effective for few-shot image classification. These results have covered Objective 2 of this thesis: a new strategy has been developed to avoid losing useful information and meanwhile suppress background clutters during feature extraction.

- The previous two research works are limited to only accessing the available information in the limited training data. For further improvements, the third contribution has addressed the exploration of leveraging auxiliary information to assist few-shot image classification (Chapter 5). To the best of our knowledge, there is no existing work presenting a thorough investigation on this. To fill the gap, we carried out a detailed analysis to figure out which is the most suitable auxiliary information to mitigate background clutters for few-shot image classification. We also designed and explored a few ways to incorporate auxiliary information into a few-shot learning pipeline, and found out the most beneficial method. The thorough exploration could provide guidelines on how to mitigate background clutters for few-shot image classification. The found most suitable auxiliary information and best way to leverage it has been empirically demonstrated improvements when incorporating them into various existing approaches for few-shot image classification. These results have covered Objective 3 of this thesis: a comprehensive investigation of suitable auxiliary information and beneficial ways to leverage it to mitigate background clutters for few-shot image classification has been carried out.

6.2 Limitations and Future Work

By summarising the contributions above, we have found several limitations related to the conducted research works in this thesis that need to be mentioned. In this section, we point out a few limitations that need to be enhanced and meanwhile raise some potential directions of future works that would overcome the limitations identified.

- Our first contribution has developed a new method to maximise the usage of the few training data in a few-shot image classification task. Since each testing sample could also provide some useful information to strengthen the representation of a category, some recent approaches propose to further use each testing sample to compose class embeddings (Garcia and Bruna, 2018; Yan, Zhang, He, et al., 2019). In our first contribution, we only consider using the limited training samples as much as possible without taking into account utilising each testing sample. When training data is limited, it would be beneficial to leverage more useful information from each testing sample alongside training data. Consequently, the extension of our L2AE method that makes use of both limited training samples and each testing sample to form robust class embeddings offers an opportunity for future work.

  Another potential limitation of our first work is that the L2AE method is not fully compatible with ResNet-like deep models, which normally apply global average pooling to summarise the final embedding into a vector. Since our L2AE module performs convolution operations to generate aggregation weights on the concatenated feature maps in each convolutional channel, it does not work well with the final vector obtained by a ResNet model. As future work, we plan to adjust the network architecture and tune the corresponding hyper-parameters of our L2AE module to make it work well with ResNet-like deep models as a future work.

- Few-shot image classification is a special problem that cannot be simply tackled by regular deep learning techniques, usually requiring particularly designed or learned new algorithm components. In the second contribution, we learned a specialised pooling operation for few-shot image classification. The Ada-P module learns a meta-learner to generate adaptive pooling weights and performs weighted pooling with a fixed pooling size for each individual embedding. To utilise as much as useful information and focus more on the target object, we think that

an adaptive-shaped pooling window could introduce more adaptability and be beneficial for few-shot image classification. The potential reason is that an adaptive-shaped window could include more related features when summarising each local region, while a fixed window is limited to a pre-defined fixed region. Similar to the Ada-P, an adaptive pooling window size can be generated by a meta-learner trained on various tasks.

Besides a pooling operation, convolution is also a key component of feature extraction. When addressing research objective 2 that targets tackling the two main issues from the angle of feature extraction, we focused on designing a more appropriate pooling operation for few-shot image classification. We think that an adaptive convolution operation that can automatically adjust the weights of its convolutional kernel according to different tasks, could extract task-specific features and offer further improvements on few-shot image classification. Therefore, this could be another direction of future works.

- In the third contribution, we leverage different types of auxiliary information and demonstrate their benefits for few-shot image classification. Since they are extracted by a pre-defined algorithm or a pre-trained model, there is no guarantee that they are always correct. For example, some extracted bounding boxes or saliency maps do not correctly cover the target objects or include some background noises in images. The potential reason could be some visual patterns in our few-shot image classification data set are unseen during the pre-training process of an object detection or SOD model, so that, the pre-trained model cannot generalise well to the new images. In fact, when manually examining the extracted different types of auxiliary information, we can indeed find some inappropriate bounding boxes or saliency maps. Besides, since there are many different algorithms for auxiliary information extraction, it is not guaranteed that our selected methods are most beneficial to few-shot image classification. These observations provide a new future work direction that targets improving the correctness of the extracted auxiliary information for better performance on few-shot image classification. We think this can be accomplished from two perspectives. First, we can improve the generalisation ability of a pre-trained model by introducing more visual patterns. Normally, the data sets for object detection or SOD generally include a number of categories of images. For example, the widely used object detection data set, COCO data set (Lin et al., 2014),

consists of 80 classes of images. We argue that these data sets may not provide sufficient visual patterns to let a pre-trained model generalise well to the unseen classes of images in a few-shot image classification data set. It would be beneficial for few-shot image classification to incorporate an object detection or SOD model pre-trained on a large-scale data set comprised by a large number of categories, such as ImageNet (Deng et al., 2009) consisting of thousands of categories. Second, we can explore various promising edge detection, object detection or SOD algorithms and find out the best-performing one for few-shot image classification.

- A common limitation of the three research works is that the proposed modules can introduce more computational burden. There is usually a trade-off between efficiency and effectiveness in algorithm design. We focus more on the effectiveness in this thesis. For future works, we will take into account both efficiency and effectiveness at the same time when developing new approaches for few-shot image classification.

- Few-shot learning is becoming a hot research topic in the ML field in recent years. This thesis focused on few-shot image classification. As we mentioned in Chapter 1, few-shot learning has been widely applied in various fields, such as computer vision (Fei-Fei, Fergus, and Perona, 2006; Lake, Salakhutdinov, and Tenenbaum, 2015), natural language processing (Sun et al., 2019b; Han et al., 2018), audio signal processing (Wang et al., 2020), robotics (Xie et al., 2018), medicine (Tian et al., 2020a), etc. As a future work direction, we may explore different types of few-shot learning problems and even design a general method applicable to various research fields. Since the main difference of few-shot learning algorithms between different fields is different embedding modules, it is possible to design a flexible and general algorithm to learn from limited data on various application fields.

  In addition to extending to other application fields, there are also some more challenging few-shot learning problems that need to be explored, such as few-shot imbalanced classification, few-shot multi-label classification, or on-line few-shot learning, etc. In this thesis, all the methods are evaluated on balanced single-label classification tasks. However, in real-world problems, we cannot guarantee the few training samples of different classes are always balanced, and we may need to assign multiple labels to a sample or learn in an on-line manner. In these scenarios,

the extension of few-shot learning approaches to embrace more complex problems poses a great challenge for future work.

Besides, the conducted research works in this thesis evaluated our methods on the widely used benchmarks following the standard N-way K-shot setting. As a future work, we plan to apply our methods into more realistic scenarios to test their performance. For example, we can explore some real-world problems that lack labelled data, such as rare disease diagnosis. Alternatively, we can use a large data set including more categories from various domains to test the generalisation ability of our methods.

Currently, the performance of few-shot image classification is worse than that of standard image classification. Even thought the final goal of few-shot learning is to perform as well as learning from plenty of data, at current stage, it deserves to be explored how much few-shot learning methods can help standard ML tasks to relieve the amount of labelled training data. For example, we plan to figure out how few training samples a few-shot learning method can use to achieve a comparable performance to that obtained by learning from the whole training set of a standard learning task. This will also provide us insights on how much the performance of a few-shot learning approach will change based on different proportions of training data.

Bibliography

Abadi, M. et al. (2016). "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems". In: *arXiv preprint arXiv:1603.04467*.

Alex, N., A. Joshua, and S. John (2018). "On First-Order Meta-Learning Algorithms". In: *arXiv preprint arXiv:1803.02999*.

Allen, K. et al. (2019). "Infinite mixture prototypes for few-shot learning". In: *International Conference on Machine Learning*, pp. 232–241.

Andrychowicz, M. et al. (2016). "Learning to learn by gradient descent by gradient descent". In: *Advances in Neural Information Processing Systems*. Barcelona, Spain, pp. 3981–3989.

Bahdanau, D., K. H. Cho, and Y. Bengio (2015). "Neural machine translation by jointly learning to align and translate". In: *International Conference on Learning Representations*.

Baker, B. et al. (2017). "Accelerating neural architecture search using performance prediction". In: *Advances in Neural Information Processing Systems, Workshop on Meta-learning*.

Bertinetto, L. et al. (2019). "Meta-learning with differentiable closed-form solvers". In: *International Conference on Learning Representations*.

Bochkovskiy, A., C. Wang, and H. Mark Liao (2020). "YOLOv4: Optimal speed and accuracy of object detection". In: *arXiv preprint arXiv:2004.10934*.

Boureau, Y., J. Ponce, and Y. LeCun (2010). "A theoretical analysis of feature pooling in visual recognition". In: *International Conference on Machine Learning*, pp. 111–118.

Bouzy, B. and G. Chaslot (2006). "Monte-Carlo Go reinforcement learning experiments". In: *2006 IEEE Symposium on Computational Intelligence and Games*, pp. 187–194.

Bradski, G. (2000). "The OpenCV Library". In: *Dr. Dobb's Journal of Software Tools*.

Brazdil, P. B., C. Soares, and J. P. Da C. (2003). "Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results". In: *Machine Learning* 50.3, pp. 251–277.

Brunetti, A. et al. (2018). "Computer vision and deep learning techniques for pedestrian detection and tracking: A survey". In: *Neurocomputing* 300, pp. 17–33.

Canny, J. (1986). "A computational approach to edge detection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6, pp. 679–698.

Chapelle, O. and A. Zien (2005). "Semi-supervised classification by low density separation". In: *International Workshop on Artificial Intelligence and Statistics*, pp. 57–64.

Chen, L. et al. (2017). "SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5659–5667.

Chen, M. et al. (2020). "Diversity Transfer Network for Few-Shot Learning." In: *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 10559–10566.

Chen, W. et al. (2018). "A Closer Look at Few-shot Classification". In: *International Conference on Learning Representations*.

Chen, Z. et al. (2019a). "Image block augmentation for one-shot learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33, pp. 3379–3386.

Chen, Z. et al. (2019b). "Image deformation meta-networks for one-shot learning". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8680–8689.

Cheng, M. et al. (2014). "Global contrast based salient region detection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.3, pp. 569–582.

Cortes, C. and V. Vapnik (1995). "Support-vector networks". In: *Machine learning* 20.3, pp. 273–297.

Das, A. et al. (2017). "Human attention in visual question answering: Do humans and deep networks look at the same regions?" In: *Computer Vision and Image Understanding* 163, pp. 90–100.

Deng, J. et al. (2009). "Imagenet: A large-scale hierarchical image database". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255.

Dhankhar, P. and N. Sahu (2013). "A review and research of edge detection techniques for image segmentation". In: *International Journal of Computer Science and Mobile Computing* 2.7, pp. 86–92.

Dharampal, V. M. (2015). "Methods of image edge detection: A review". In: *Journal of Electrical & Electronic Systems* 4.2, p. 5.

Dutta, A., A. Kar, and B. Chatterji (2008). "Corner detection algorithms for digital images in last three decades". In: *IETE Technical Review* 25.3, pp. 123–133.

Edwards, H. and A. Storkey (2017). "Towards a neural statistician". In: *International Conference on Learning Representations*.

Esser, P., E. Sutter, and B. Ommer (2018). "A variational u-net for conditional appearance and shape generation". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8857–8866.

Fang, H. et al. (2015). "From captions to visual concepts and back". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1473–1482.

Fei-Fei, L., R. Fergus, and P. Perona (2006). "One-shot learning of object categories". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.4, pp. 594–611.

Figueroa-Flores, C. et al. (2021). "Hallucinating Saliency Maps for Fine-Grained Image Classification for Limited Data Domains". In: *International Conference on Computer Vision Theory and Applications*.

Finn, C., P. Abbeel, and S. Levine (2017). "Model-agnostic meta-learning for fast adaptation of deep networks". In: *International Conference on Machine Learning*, pp. 1126–1135.

Fix, E. and Hodges J. (1952). *Discriminatory analysis-nonparametric discrimination: Small sample performance*. Tech. rep. California Univ Berkeley.

Flennerhag, S. et al. (2020). "Meta-Learning with Warped Gradient Descent". In: *International Conference on Learning Representations*.

Frans, K. et al. (2018). "Meta Learning Shared Hierarchies". In: *International Conference on Learning Representations*. Vancouver, Canada.

Gama, J. and P. Brazdil (1995). "Characterization of classification algorithms". In: *Portuguese Conference on Artificial Intelligence*. Springer, pp. 189–200.

Gao, H. et al. (2018). "Low-shot learning via covariance-preserving adversarial augmentation networks". In: *Advances in Neural Information Processing Systems*, pp. 975–985.

Garcia, V. and J. Bruna (2018). "Few-shot learning with graph neural networks". In: *International Conference on Learning Representations*.

Ghiasi, G., T. Lin, and Q. V. Le (2018). "Dropblock: A regularization method for convolutional networks". In: *Advances in Neural Information Processing Systems*, pp. 10727–10737.

Gidaris, S. and N. Komodakis (2018). "Dynamic few-shot visual learning without forgetting". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4367–4375.

Gidaris, S. and N. Komodakis (2019). "Generating classification weights with GNN denoising autoencoders for few-shot learning". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 21–30.

Girshick, R. (2015). "Fast R-CNN". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448.

Girshick, R. et al. (2014). "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587.

Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep Learning*. MIT Press.

Goodfellow, I. et al. (2014). "Generative adversarial nets". In: *Advances in Neural Information Processing Systems*, pp. 2672–2680.

Guneet, S. et al. (2020). "A Baseline for Few-Shot Image Classification". In: *International Conference on Learning Representations*.

Ha, D., A. Dai, and Q. V. Le (2016). "Hypernetworks". In: *International Conference on Learning Representations*.

Han, K. and B. Uyyanonvara (2016). "A survey of blob detection algorithms for biomedical images". In: *International Conference of Information and Communication Technology for Embedded Systems*. IEEE, pp. 57–60.

Han, X. et al. (2018). "FewRel: A Large-Scale Supervised Few-Shot Relation Classification Dataset with State-of-the-Art Evaluation". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 4803–4809.

He, J. et al. (2012). "Mobile product search with bag of hash bits and boundary reranking". In: *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 3005–3012.

He, K. et al. (2016). "Identity mappings in deep residual networks". In: *Proceedings of the European Conference on Computer Vision*, pp. 630–645.

Hinton, G., O. Vinyals, and J. Dean (2014). "Distilling the knowledge in a neural network". In: *Advances in Neural Information Processing Systems, Workshop on deep learning*.

Hinton, G. E and R. Salakhutdinov (2006). "Reducing the dimensionality of data with neural networks". In: *Science* 313.5786, pp. 504–507.

Hosmer, J., S. Lemeshow, and R. Sturdivant (2013). *Applied logistic regression*. Vol. 398. John Wiley & Sons.

Hu, G. et al. (2015). "When face recognition meets with deep learning: an evaluation of convolutional neural networks for face recognition". In: *Proceedings of the IEEE International Conference on Computer Vision workshops*, pp. 142–150.

Huang, C. et al. (2016). "Learning deep representation for imbalanced classification". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5375–5384.

Huang, G. et al. (2017). "Densely connected convolutional networks". In: *IEEE conference on computer vision and pattern recognition*, pp. 4700–4708.

Hutter, F., L. Kotthoff, and J. Vanschoren (2019). *Automated machine learning: methods, systems, challenges*.

Ioffe, S. and C. Szegedy (2015). "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *International Conference on Machine Learning*, pp. 448–456.

Jaehoon, O. et al. (2021). "BOIL: Towards Representation Change for Few-shot Learning". In: *International Conference on Learning Representations*.

Jamal, M. A. and G. Qi (2019). "Task agnostic meta-learning for few-shot learning". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11719–11727.

Jetley, S. et al. (2018). "Learn to Pay Attention". In: *International Conference on Learning Representations*.

Jia, Y., C. Huang, and T. Darrell (2012). "Beyond spatial pyramids: Receptive field learning for pooled image features". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3370–3377.

Jiang, H. et al. (2013). "Salient object detection: A discriminative regional feature integration approach". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2083–2090.

Kim, J. et al. (2019). "Edge-labeling graph neural network for few-shot learning". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11–20.

Kingma, D. P. and J. Ba (2015). "ADAM: A method for stochastic optimization". In: *International Conference on Learning Representations*.

Kingma, D. P. et al. (2014). "Semi-supervised learning with deep generative models". In: *Advances in Neural Information Processing Systems*, pp. 3581–3589.

Koch, G., R. Zemel, and R. Salakhutdinov (2015). "Siamese neural networks for one-shot image recognition". In: *International Conference on Machine Learning(Deep Learning Workshop)*.

Krizhevsky, A., I. Sutskever, and G. E. Hinton (2012). "Imagenet classification with deep convolutional neural networks". In: *Advances in Neural Information Processing Systems*, pp. 1097–1105.

Kulis, B. et al. (2012). "Metric learning: A survey". In: *Foundations and Trends in Machine Learning* 5.4, pp. 287–364.

Kwon, N. et al. (2021). "Repurposing Pretrained Models for Robust Out-of-domain Few-Shot Learning". In: *International Conference on Learning Representations*.

Lake, B. et al. (2011). "One shot learning of simple visual concepts". In: *Proceedings of the Annual Meeting of the Cognitive Science Society*. Vol. 33. 33.

Lake, B. M., R. Salakhutdinov, and J. B. Tenenbaum (2015). "Human-level concept learning through probabilistic program induction". In: *Science* 350.6266, pp. 1332–1338.

LeCun, Y., Y. Bengio, and G. Hinton (2015). "Deep learning". In: *nature* 521.7553, pp. 436–444.

LeCun, Y., Y. Bengio, et al. (1995). "Convolutional networks for images, speech, and time series". In: *The handbook of brain theory and neural networks* 3361.10, p. 1995.

Lee, C., P. W. Gallagher, and Z. Tu (2016). "Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree". In: *International Conference on Artificial Intelligence and Statistics*, pp. 464–472.

Lee, K. et al. (2019). "Meta-learning with differentiable convex optimization". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10657–10665.

Li, A. et al. (2019a). "Large-scale few-shot learning: Knowledge transfer with class hierarchy". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7212–7220.

Li, A. et al. (2020a). "Boosting Few-Shot Learning With Adaptive Margin Loss". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12576–12584.

Li, H. et al. (2019b). "Finding task-relevant features for few-shot learning by category traversal". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–10.

Li, K. et al. (2020b). "Adversarial Feature Hallucination Networks for Few-Shot Learning". In: *IEEE Conference on Computer Vision and Pattern Recognition*.

Li, W. et al. (2019c). "Distribution consistency based covariance metric networks for few-shot learning". In: *AAAI Conference on Artificial Intelligence*. Vol. 33. 01, pp. 8642–8649.

Li, W. et al. (2019d). "Revisiting local descriptor based image-to-class measure for few-shot learning". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7260–7268.

Li, X. et al. (2019e). "Learning to self-train for semi-supervised few-shot classification". In: *Advances in Neural Information Processing Systems*, pp. 10276–10286.

Li, Z. et al. (2017). "Meta-SGD: Learning to learn quickly for few shot learning". In: *arXiv preprint arXiv:1707.09835*.

Lifchitz, Y. et al. (2019). "Dense classification and implanting for few-shot learning". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9258–9267.

Lin, T. et al. (2014). "Microsoft COCO: Common objects in context". In: *European Conference on Computer Vision*, pp. 740–755.

Liu, L. et al. (2019a). "Prototype Propagation Networks (PPN) for weakly-supervised few-shot learning on category graph". In: *International Joint Conference on Artificial Intelligence*.

Liu, Y. et al. (2019b). "Learning to propagate labels: Transductive propagation network for few-shot learning". In: *International Conference on Learning Representations*.

Lloyd, S. (1982). "Least squares quantization in PCM". In: *IEEE Transactions on Information Theory* 28.2, pp. 129–137.

Lorraine, J. and D. Duvenaud (2017). "Hyperparameter Optimization with Hypernets". In: *Advances in Neural Information Processing Systems, Workshop on Meta-learning*.

Lu, Y., Y. Fang, and C. Shi (2020). "Meta-learning on heterogeneous information networks for cold-start recommendation". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1563–1573.

Lu, Z. et al. (2019). "Cross-modality interactive attention network for multispectral pedestrian detection". In: *Information Fusion* 50, pp. 20–29.

Maaten, L. Van der and G. Hinton (2008). "Visualizing data using t-SNE." In: *Journal of machine learning research* 9.11.

Meng, L. et al. (2019). "Interpretable spatio-temporal attention for video action recognition". In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*.

Mishra, N. et al. (2018). "A simple neural attentive meta-learner". In: *International Conference on Learning Representations*.

Murabito, F. et al. (2018). "Top-down saliency detection driven by visual classification". In: *Computer Vision and Image Understanding* 172, pp. 67–76.

Oreshkin, B., P. R. López, and A. Lacoste (2018). "TADAM: Task dependent adaptive metric for improved few-shot learning". In: *Advances in Neural Information Processing Systems*, pp. 721–731.

O'Mahony, N. et al. (2019). "Deep learning vs. traditional computer vision". In: *Science and Information Conference*, pp. 128–144.

Peralta, D. et al. (2015). "A survey on fingerprint minutiae-based local matching for verification and identification: Taxonomy and experimental evaluation". In: *Information Sciences* 315, pp. 67–87.

Pérez-Hernández, F. et al. (2020). "Object detection binary classifiers methodology based on deep learning to identify small objects handled similarly: Application in video surveillance". In: *Knowledge-Based Systems* 194, p. 105590.

Qi, H., M. Brown, and D. G. Lowe (2018). "Low-shot learning with imprinted weights". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5822–5830.

Qiao, S. et al. (2018). "Few-shot image recognition by predicting parameters from activations". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7229–7238.

Quinlan, J. Ross (1986). "Induction of decision trees". In: *Machine Learning* 1.1, pp. 81–106.

Ravi, S. and H. Larochelle (2017). "Optimization as a model for few shot learning". In: *International Conference on Learning Representations*.

Redmon, J. and A. Farhadi (2017). "YOLO9000: better, faster, stronger". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7263–7271.

– (2018). "YOLOv3: An incremental improvement". In: *arXiv preprint arXiv:1804.02767*.

Redmon, J. et al. (2016). "You only look once: Unified, real-time object detection". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788.

Ren, M. et al. (2018). "Meta-learning for semi-supervised few-shot classification". In: *International Conference on Learning Representations*.

Ren, S. et al. (2015). "Faster R-CNN: Towards real-time object detection with region proposal networks". In: *Advances in Neural Information Processing Systems* 28, pp. 91–99.

Ren, W. et al. (2021). "On Fast Adversarial Robustness Adaptation in Model-Agnostic Meta-Learning". In: *International Conference on Learning Representations*.

Rendell, L. and H. Cho (1990). "Empirical learning as a function of concept character". In: *Machine Learning* 5.3, pp. 267–298.

Ronneberger, O., P. Fischer, and T. Brox (2015). "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241.

Rumelhart, D., G. Hinton, and R. Williams (1986). "Learning representations by back-propagating errors". In: *nature* 323.6088, pp. 533–536.

Rummery, G. and M. Niranjan (1994). *On-line Q-learning using connectionist systems*. Vol. 37. Citeseer.

Russakovsky, O. et al. (2015). "Imagenet large scale visual recognition challenge". In: *International Journal of Computer Vision* 115.3, pp. 211–252.

Rusu, A. A. et al. (2019). "Meta-learning with latent embedding optimization". In: *International Conference on Learning Representations*.

Rutishauser, U. et al. (2004). "Is bottom-up attention useful for object recognition?" In: *IEEE Conference on Computer Vision and Pattern Recognition*. Vol. 2.

Saeedan, F. et al. (2018). "Detail-preserving pooling in deep networks". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9108–9116.

Santoro, A. et al. (2016). "Meta-learning with memory-augmented neural networks". In: *International Conference on Machine Learning*, pp. 1842–1850.

Shahvari, O. and R. Logendran (2017). "An enhanced tabu search algorithm to minimize a bi-criteria objective in batching and scheduling problems on unrelated-parallel machines with desired lower bounds on batch sizes". In: *Computers & Operations Research* 77, pp. 154–176.

Shorten, C. and T. M. Khoshgoftaar (2019). "A survey on image data augmentation for deep learning". In: *Journal of Big Data* 6.1, pp. 1–48.

Simon, C. et al. (2020). "Adaptive Subspaces for Few-Shot Learning". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4136–4145.

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

Snell, J., K. Swersky, and R. Zemel (2017). "Prototypical networks for few-shot learning". In: *Advances in Neural Information Processing Systems*, pp. 4077–4087.

Song, H., I. Triguero, and E. Özcan (2019). "A review on the self and dual interactions between machine learning and optimisation". In: *Progress in Artificial Intelligence* 8.2, pp. 143–165.

Song, H. et al. (2021). "L2AE-D: Learning to aggregate embeddings for few-shot learning with meta-level dropout". In: *Neurocomputing* 442, pp. 200–208.

Srivastava, N. et al. (2014). "Dropout: a simple way to prevent neural networks from overfitting". In: *The Journal of Machine Learning Research* 15.1, pp. 1929–1958.

Sun, Q. et al. (2019a). "Meta-transfer learning for few-shot learning". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 403–412.

Sun, S. et al. (2019b). "Hierarchical attention prototypical networks for few-shot text classification". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*, pp. 476–485.

Sung, F. et al. (2018). "Learning to compare: Relation network for few-shot learning". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1199–1208.

Sutton, R. S. and A. G. Barto (1998). *Reinforcement learning: An introduction*. MIT press.

Szegedy, C. et al. (2016). "Rethinking the inception architecture for computer vision". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826.

Tan, M. and Q. Le (2019). "EfficientNet: Rethinking model scaling for convolutional neural networks". In: *International Conference on Machine Learning*. PMLR, pp. 6105–6114.

Thrun, S. et al. (2006). "Stanley: The robot that won the DARPA Grand Challenge". In: *Journal of Field Robotics.* 23.9, pp. 661–692.

Tian, Y. et al. (2020a). "Few-shot anomaly detection for polyp frames from colonoscopy". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 274–284.

Tian, Y. et al. (2020b). "Rethinking few-shot image classification: a good embedding is all you need?" In: *Proceedings of the European Conference on Computer Vision*, pp. 266–282.

Tianshi, C., L. Marc, and F. Sanja (2020). "A Theoretical Analysis of the Number of Shots in Few-Shot Learning". In: *International Conference on Learning Representations*.

Tokmakov, P., Y. Wang, and M. Hebert (2019). "Learning compositional representations for few-shot recognition". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 6372–6381.

Tompson, J. et al. (2015). "Efficient object localization using convolutional networks". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 648–656.

Vercellino, C. J. and W. Y. Wang (2017). "Hyperactivations for Activation Function Exploration". In: *Advances in Neural Information Processing Systems, Workshop on Meta-learning*.

Vilalta, R. and Y. Drissi (2002). "A perspective view and survey of meta-learning". In: *Artificial Intelligence Review* 18.2, pp. 77–95.

Vinyals, O. et al. (2016). "Matching networks for one shot learning". In: *Advances in Neural Information Processing Systems*, pp. 3630–3638.

Vladimir, P. and D. Vladimir (2015). "Focused pooling for image fusion evaluation". In: *Information Fusion* 22, pp. 119–126.

Wah, C. et al. (2011). "The caltech-ucsd birds-200-2011 dataset". In.

Wang, F. et al. (2017). "Residual attention network for image classification". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3156–3164.

Wang, J. et al. (2016). "CNN-RNN: A unified framework for multi-label image classification". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2285–2294.

Wang, R. et al. (2021a). "On Fast Adversarial Robustness Adaptation in Model-Agnostic Meta-Learning". In: *International Conference on Learning Representations*.

Wang, W., J. Shen, and H. Ling (2018). "A deep network solution for attention and aesthetics aware photo cropping". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.7, pp. 1531–1544.

Wang, W. et al. (2021b). "Salient object detection in the deep learning era: An in-depth survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Wang, X. et al. (2018a). "Weakly-supervised semantic segmentation by iteratively mining common object features". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1354–1362.

Wang, Y., D. Ramanan, and M. Hebert (2017). "Learning to Model the Tail". In: *Advances in Neural Information Processing Systems*. Long Beach, USA, pp. 7032–7042.

Wang, Y. et al. (2018b). "Low-shot learning from imaginary data". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7278–7286.

Wang, Y. et al. (2020). "Few-shot sound event detection". In: *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 81–85.

Watkins, C. and P. Dayan (1992). "Q-learning". In: *Machine Learning* 8.3-4, pp. 279–292.

Wei, Y. et al. (2012). "Geodesic saliency using background priors". In: *Proceedings of the European Conference on Computer Vision*. Springer, pp. 29–42.

Wei, Y. et al. (2016). "Stc: A simple to complex framework for weakly-supervised semantic segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.11, pp. 2314–2320.

Wertheimer, D. and B. Hariharan (2019). "Few-shot learning with localization in realistic settings". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6558–6567.

Woo, S. et al. (2018). "CBAM: Convolutional block attention module". In: *Proceedings of the European Conference on Computer Vision*, pp. 3–19.

Xie, A. et al. (2018). "Few-shot goal inference for visuomotor learning and planning". In: *Conference on Robot Learning*, pp. 40–52.

Xing, Chen et al. (2019). "Adaptive cross-modal few-shot learning". In: *Advances in Neural Information Processing Systems*, pp. 4848–4858.

Xiong, Z., W. Zhang, and W. Zhu (2017). "Learning Decision Trees with Reinforcement Learning". In: *Advances in Neural Information Processing Systems, Workshop on Meta-learning*. Long Beach, USA.

Yan, S., S. Zhang, X. He, et al. (2019). "A dual attention network with semantic embedding for few-shot learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33, pp. 9079–9086.

Yao, H. et al. (2019). "Hierarchically Structured Meta-learning". In: *International Conference on Machine Learning*, pp. 7045–7054.

Ye, H. et al. (2020). "Few-Shot Learning via Embedding Adaptation with Set-to-Set Functions". In: *IEEE Conference on Computer Vision and Pattern Recognition*.

Zeiler, M. D. and R. Fergus (2013). "Stochastic pooling for regularization of deep convolutional neural networks". In: *International Conference on Learning Representations*.

– (2014). "Visualizing and understanding convolutional networks". In: *Proceedings of the European Conference on Computer Vision*, pp. 818–833.

Zhang, H., J. Zhang, and P. Koniusz (2019). "Few-shot learning via saliency-guided hallucination of samples". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2770–2779.

Zhang, H. et al. (2018a). "mixup: Beyond Empirical Risk Minimization". In: *International Conference on Learning Representations*.

Zhang, R. et al. (2018b). "MetaGAN: An adversarial approach to few-shot learning". In: *Advances in Neural Information Processing Systems*, pp. 2365–2374.

Zhao, J. et al. (2019a). "EGNet: Edge guidance network for salient object detection". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 8779–8788.

Zhao, Z. et al. (2019b). "Object detection with deep learning: A review". In: *IEEE Transactions on Neural Networks and Learning Systems* 30.11, pp. 3212–3232.

Zhu, W. et al. (2014). "Saliency optimization from robust background detection". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2814–2821.

Zhu, X. et al. (2019). "An empirical study of spatial attention mechanisms in deep networks". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 6688–6697.

Zhuang, J. et al. (2019). "CARE: Class attention to regions of lesion for classification on imbalanced data". In: *International Conference on Medical Imaging with Deep Learning*, pp. 588–597.