FEW-SHOT SEGMENTATION BY ENHANCED ENSEMBLE OF BASE AND META PREDICTIONS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ALPER KAYABAŞI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

APRIL 2023

Approval of the thesis:

FEW-SHOT SEGMENTATION BY ENHANCED ENSEMBLE OF BASE AND META PREDICTIONS

submitted by **ALPER KAYABAŞI** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. İlkay Ulusoy
Head of Department, **Electrical and Electronics Engineering** _____

Prof. Dr. İlkay Ulusoy
Supervisor, **Electrical and Electronics Engineering, METU** _____

Examining Committee Members:

Prof. Dr. Abdullah Aydın Alatan
Electrical and Electronics Engineering, METU _____

Prof. Dr. İlkay Ulusoy
Electrical and Electronics Engineering, METU _____

Prof. Dr. Klaus Werner Schmidt
Electrical and Electronics Engineering, METU _____

Assoc. Prof. Dr. Sinan Kalkan
Computer Engineering, METU _____

Assoc. Prof. Dr. Nazlı İkizler Cinbiş
Computer Engineering, Hacettepe University _____

Date:28.04.2023

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Alper Kayabaşı

Signature :

ABSTRACT

FEW-SHOT SEGMENTATION BY ENHANCED ENSEMBLE OF BASE AND META PREDICTIONS

Kayabaşı, Alper
M.S., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. İlkay Ulusoy

April 2023, 72 pages

Supervised learning approaches assume that there is a large amount of data available with labels. Since data annotation is a costly and time-consuming task, this assumption loses its validity when there are financial or time constraints. In addition, only a small number of samples may be available for specific classes, such as images of endangered animals. To address these issues, the concept of few-shot learning has been developed to recognize patterns from novel tasks with limited supervision. As a sub-task of few-shot learning, few-shot segmentation aims to create a generalizing model that can segment query images from unseen classes during training, using a few support images whose class matches that of the query image. Previous research has identified two specific problems in this domain: spatial inconsistency and a bias toward seen classes. To address the issue of spatial inconsistency, the proposed method in this thesis compares the support feature map to the query feature map at multiple scales, making it scale-agnostic. To address the bias towards seen classes, a supervised model called the base learner is trained on available classes to identify pixels belonging to seen classes accurately. The meta learner then uses an ensemble learning model to coordinate with the base learner and discard areas belonging to seen classes.

Our method in this thesis is the first to address these two crucial problems simultaneously and achieves state-of-the-art performance on both PASCAL-5^i and COCO-20^i datasets.

Keywords: few-shot segmentation, ensemble learning

ÖZ

META VE BAZ TAHMİNLERİ KULLANAN GELİŞTİRİLMİŞ KOLEKTİF ÖĞRENMEYLE BİRKAÇ ÖRNEKLİ BÖLÜTLEME

Kayabaşı, Alper
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. İlkay Ulusoy

Nisan 2023 , 72 sayfa

Denetimli öğrenme paradigması, etiketiyle birlikte bol miktarda verinin mevcut olduğunu varsaymaktadır. Veri etiketleme maliyetli ve zaman alıcı bir iş olduğundan, finansal ve ya zamansal kısıtlar olduğunda bu varsayım geçerliliğini yitirir. Buna ek olarak, bazı sınıflar için örnek görüntülerin sayısı az olabilir (Örneğin soyu tükenmekte olan hayvanlara ait görüntüler). Bu durumlara değinmek için, kısıtlı denetim altında yeni görevlerdeki örüntülerin tanınması amacıyla birkaç adımda öğrenme konsepti geliştirilmiştir. Birkaç adımlı öğrenmenin alt başlığı olan birkaç adımlı bölütleme, sınıfı sorgu görüntüsündeki sınıfla eşleşen birkaç destek görüntüsünün rehberliğinde eğitim sırasında görünmeyen sınıfları içeren sorgu görüntülerini bölütleyen genelleyici bir model geliştirmeyi amaçlar. Daha önceki çalışmalarda belirtilen, boyutsal tutarsızlık ve görülen sınıflara yönelik önyargı olmak üzere alana ait iki problem vardır. Boyutsal tutarsızlık sorununu çözmek için, bu tezde önerilen yöntem, destek özellik haritasını çeşitli ölçeklerde sorgu özellik haritasıyla karşılaştırır, bu nedenle ölçeğe bağımlılık ortadan kalkmaktadır. Görülen sınıflara yönelik eğilimi ele almak için, temel öğrenen adı verilen denetimli bir model görülen sınıflara ait piksel-

leri doğru bir şekilde tanımlamak için mevcut sınıflar üzerinde eğitilir. Meta öğrenici daha sonra temel öğrenenle koordinasyon sağlamak ve görülen sınıflara ait alanları göz ardı etmek için bir topluluk öğrenme modeli kullanır. Bu tez, bu iki hayati sorunu ilk kez aynı anda ele alıyor ve hem PASCAL-5^i hem de COCO-20^i veri kümelerinde en iyi performansları elde ediyor.

Anahtar Kelimeler: Birkaç örnekle bölütleme, kollektif öğrenme

To my beloved family

ACKNOWLEDGMENTS

I consider myself fortunate to have had the opportunity to be under the supervision of Prof. Dr. İlkay Ulusoy. Over the course of three wonderful years, we made contributions to the exciting world of few-shot learning, which I believe is shaping the future of deep learning. Her unwavering support and mentorship have been invaluable, and I am deeply grateful for her contributions to my academic growth.

I am also grateful for the intellectual conversations I had with my colleague, Gülin Tüfekci, regarding the few-shot learning. Our exchanges of ideas were enjoyable and enlightening. Gülin's technical expertise and positive attitude towards challenges have been a constant source of motivation for me. I extend my sincere thanks to her for her support.

I would also like to express my appreciation to my company, ASELSAN, and my leader, Dr. Veysel Yücesoy, for providing full support during the development of my thesis. With their help, I was able to present our findings at a prestigious computer vision conference, which was an invaluable experience for me.

Last but certainly not least, my family has been my rock throughout my academic journey. My father, Uğur Kayabaşı, my mother, Melahat Kayabaşı, and my sister, Zeynep Kayabaşı, have provided me with support and instilled in me the spirit of perseverance. Their love and unwavering encouragement have been instrumental in my success, and for that, I am truly grateful.

TABLE OF CONTENTS

ABSTRACT v

ÖZ vii

ACKNOWLEDGMENTS x

TABLE OF CONTENTS xi

LIST OF TABLES xiv

LIST OF FIGURES xv

LIST OF ABBREVIATIONS xix

CHAPTERS

1 INTRODUCTION 1

1.1 From Segmentation to Few-Shot Segmentation 1

1.2 Problem Statement 2

1.3 Motivation 2

1.4 Contributions 4

1.5 The Outline of the Thesis 5

2 OVERVIEW OF FEW-SHOT SEGMENTATION 7

2.1 Few-Shot Segmentation Problems 10

2.1.1 Misinterpretation of Background 10

2.1.2 Imbalance in Details 11

2.1.3 Inter-Class Gap 20
2.1.4 Spatial Inconsistency 23
2.1.5 Scalability with Number of Support Data 25
2.1.6 Correlation Reliability 28
2.1.7 Thin Object Issue 30
2.2 Connection between prompt learning and few-shot segmentation . . . 32
3 PROPOSED METHOD: BASE AND META LEARNER++ 33
3.1 Background Methods 34
3.1.1 Revisit Feature Enrichment Module 34
3.1.1.1 Inter-Source Enrichment Module 34
3.1.1.2 Prior Mask 35
3.1.1.3 Inter-Source Enrichment Module 37
3.1.1.4 Information Concentration 39
3.1.1.5 K-Shot Configuration 40
3.1.2 Revisit Base and Meta Learner 41
3.1.2.1 Episodic Learning 41
3.1.2.2 Bias and Its Solution 42
3.1.2.3 Base Learner 43
3.1.2.4 Meta Learner 44
3.1.2.5 K-Shot Configuration 45
3.1.2.6 Ensemble Learner 48
3.2 Proposed Method 49
3.2.1 Range of Ensembles 51

4 EXPERIMENTAL EVALUATION 55
4.1 Experimental Setup 55
4.1.1 Dataset 55
4.1.2 Implementation Details 55
4.1.3 Performance Metrics 56
4.2 Experimental Results 57
4.2.1 Compared Methods 57
4.2.2 Quantitative Results 58
4.2.3 Generalized few-shot segmentation results 59
4.2.4 Multi-scale few-shot segmentation results 60
4.2.5 Model Complexity 61
4.2.6 Qualitative Results 61
4.2.7 Weakness of the proposed method 62
4.2.8 Ablation Study 63
5 CONCLUSIONS 65
5.1 Summary 65
5.2 Conclusions 65
5.3 Limitations and Future Work 66
REFERENCES 67

LIST OF TABLES

TABLES

Table 4.1 1-shot and 5-shot class mIoU results on PASCAL-5^i dataset for VGG-16 and ResNet-50 as backbone, provided for 4 folds and the average. The best results are given in **boldface**. The underlined results show the best performance excluding our method. 58

Table 4.2 1-shot and 5-shot class mIoU results on COCO-20^i dataset for ResNet-50 as backbone, provided for 4 folds and the average. The best results are given in **boldface**. The underlined results show the best performance excluding our method. 58

Table 4.3 1-shot and 5-shot FB-IoU results on PASCAL-5^i dataset for VGG-16 and ResNet-50 as backbone, provided as the average. The best results are given in **boldface**. The underlined results show the best performance excluding our method. 59

Table 4.4 Generalized few-shot segmentation results on PASCAL-5^i dataset for VGG-16 and ResNet-50 as backbone. The best results are given in **boldface**. 60

Table 4.5 Multi-scale few-shot segmentation results on PASCAL-5^i dataset for ResNet-50 as backbone. The results presented in this table are obtained by averaging the results from fold-0 and fold-1. The x in the table corresponds to the total number of pixels in evaluated images. 60

Table 4.6 Ablation studies on inner losses for the multi-scale predictions regarding the ensembling with the base map under 1-shot setting for PASCAL-5^i. Results show the averaged mIoU over 4 folds. 63

LIST OF FIGURES

FIGURES

Figure 1.1 (a) Overview of BAM [1]. Support and query features are used by meta learner to extract support feature map while base learner provides guidance for the base classes and leads the meta learner to focus on novel regions via ensembling. (b) Our proposed method. The decoder for meta learner is improved such that query feature map is obtained at multi-scale. Support feature map is compared with query feature maps at multi-scale to obtain enriched query features. Query predictions obtained from enriched query features at each scale are ensembled with the base map as well as the prediction obtained from the fusion of them. Inner losses are computed at different scale levels and the final prediction is obtained from the ensemble of the base map with the predictions from the fused query feature maps. (Best viewed in zoom) 3

Figure 2.1 Taxonomy of Few-Shot Segmentation 9

Figure 2.2 Novel class in background, namely person, is treated as background, leading to reduction of its distinguishability [2]. 10

Figure 2.3 Despite selecting the support image identical to the query image, the model fails to segment certain regions of the query. This indicates a loss of information resulting from the averaging operation [3]. 13

Figure 2.4 Foreground of support is clustered into regions, and each query pixel only attends most similar partitioned region in support [4]. 15

Figure 2.5 The sample does not conform to cycle-consistency paradigm [5]. 17

Figure 2.6 Inter-class gap problem is illustrated with help of t-SNE graphs. Left-hand side figure portrays embedding space before application of [6] while right-hand figure represents embedding space after its application. 19

Figure 2.7 Optimizing part of model can overcome overfitting problem. Left-hand side figure completely freezes backbone, while [7] selects particular weights cleverly, after which it fine-tunes them only. 21

Figure 2.8 Discrepancy between sizes of object in query and support image leads to spatial inconsistency problem. 23

Figure 2.9 Most up-to-date methods do not show comparable performance in 5-shot case relative to that in 1-shot [8]. 26

Figure 2.10 Correlation maps are filtered with 4-D convolution to eliminate misinterpreted correspondences [9]. 28

Figure 2.11 Left-hand side figure reveals oversegmentation of the leaves of the plant, which is a weakness of the most few-shot segmentation methods. Right-hand side figure shows that [10] alleviates this problem effectively. 30

Figure 3.1 This figure presents an overview of the feature enrichment module proposed in [1]. The blue, orange, and gray rectangles correspond to the query feature map at different resolutions, the expanded support prototype, and the resized prior mask, respectively. These three components are concatenated and passed through a 1x1 convolutional layer denoted by the pink rectangle. The inter-source enrichment module is responsible for the aforementioned operations up to this point. The outputs of these blocks are then provided to block M, which is named the inter-scale merging module. 35

Figure 3.2 This figure highlights that the regions related to the target class have higher activation compared to the background. 36

Figure 3.3 Overview of Inter-Scale Merging Module [11]. In this illustration, C, α, and β correspond to the concatenation operation, a 1-by-1 convolution, and a 3-by-3 set of convolutions, respectively. 38

Figure 3.4 The left-hand side of the figure represents a convolution layer that is designed to enhance the feature representation. On the other hand, the right-hand side of the figure shows a classification block, where the final layer reduces the channel dimension to 2. 40

Figure 3.5 Episodic Learning comprises of two stages: meta-training and meta-testing. During meta-testing, the classes presented are completely distinct from the ones encountered during meta-training. The term 'task' can be used interchangeably with 'episode'. 41

Figure 3.6 Overview of Base and Meta Learner. 42

Figure 3.7 Visualization of Computation of Gram Matrix. In this figure, n_c, n_h, and n_w corresponds to C_{low}, H_{low}, and W_{low} respectively. 45

Figure 3.8 Visualization of the Computation of the Adjustment Factor is presented in this illustration. U and T represent the unfold and transpose operations, respectively. The cross inside the circle denotes matrix multiplication, while the minus sign inside the circle denotes element-wise subtraction. 46

Figure 3.9 Detailed architecture of the multi-scale ensemble module. Features at multi-scale and the fusion of them are obtained at the end of the improved decoder as $\mathbf{X}_q^{s_i}$ and $\mathbf{X}_q^{fused}$ respectively, which are used by the corresponding auxiliary classifiers. The resultant enriched query feature maps are ensembled with the base map to obtain query predictions at multi-scale, which are denoted by $\mathbf{p}_{f,s_i}$ and $\mathbf{p}_{f,fused}$ respectively. Inner losses are computed from probability maps at intermediate scales $\left(\mathbf{p}_{m,s_i}\right)$ and predictions at intermediate scales $\left(\mathbf{p}_{f,s_i}\right)$ while fused losses are computed from fused probability maps $\left(\mathbf{p}_{m,fused}\right)$ and fused predictions $\left(\mathbf{p}_{f,fused}\right)$. (Best viewed in color) 52

Figure 4.1 Green coloured box represents ground truth while red coloured box represents prediction. 56

Figure 4.2 Qualitative 1-shot results on PASCAL-5^i dataset for ResNet-50 backbone. Results for one novel class from each fold are provided in rows. First two columns contain image and mask for support while the following two columns contain image and ground truth for query. Fifth column shows the probability map for query obtained from base learner. Predictions are provided for BAM [1] and our method for comparison in the last two columns. (Best viewed in color) 62

Figure 4.3 Misguidance of base map 63

LIST OF ABBREVIATIONS

FCN	Fully Convolutional Neural Network
ASPP	Atrous Spatial Pyramid Pooling
PSP	Pyramid Spatial Pooling
BAM	Base and Meta Learner
MLC	Mining Latent Class
FPTrans	Feature Proxy Transformer
PANet	Prototypical Alignment Network
DANet	Democratic Attention Network
SCL	Self-guided and Cross-guided Learning
CWT	Classifier Weight Transformer
ASGNet	Adaptive Superpixel-guided Network
MASK-SLIC	Masked Simple Linear Iterative Clustering
IPMT	Intermediate Prototype Mining Transformer
CANet	Class Agnostic Segmentation Network
RePRI	Region Proportion Regularized Inference
KL	Kullback-Leibler
PGNet	Pyramid Graph Network
SAGNN	Scale-Aware Graph Neural Network
PFENet	Prior Guided Feature Enrichment Network
ConvGRU	Convolutional Gated Recurrent Unit
CMN	Cyclic Memory Network
HS	Hyper-correlation Squeeze
IoU	Intersection Over Union
mIoU	Mean Intersection Over Union

CyCTR	Cycle-Consistent Transformer
ASR	Anti-aliasing Semantic Reconstruction
MMNet	Meta-class Memory Network
DGP	Dense Gaussian Process
	DPCN Dynamic Prototype Convolution Network
ASNet	Attentive Squeeze Network

CHAPTER 1

INTRODUCTION

1.1 From Segmentation to Few-Shot Segmentation

Semantic segmentation is a crucial task that classifies each pixel of an image to make sense of the scene with application areas such as autonomous driving [12], and medical imaging [13]. Deep learning pervades semantic segmentation like other tasks of computer vision [14, 15]. Fully convolutional neural network (FCN), which is the pioneering work in semantic segmentation field, formulates semantic segmentation as a pixel-wise classification task [15]. In FCN, all fully connected layers at the end of a model are transformed into convolution layers so that the network accepts arbitrary input sizes. Success of FCN accelerates the field and results in outstanding architectures such as UNet [13], PSPNet [16], and Deeplab [14, 17]. PSPNet combines average pooled feature maps at different scales to contain not only global but also local context [16]. Deeplab introduces ASPP module [14] equipped with dilated convolution that increases the receptive field of the network without a decrease in resolution by inserting holes between filter weights. Supervised segmentation models are required to employ abundant annotated data belonging to each class in the training set since the generalization capacity of supervised models decreases with scarce labeled data. Therefore, adapting the model to work on unseen classes requires dense annotation of myriad data from novel classes. Shaban *et al.* [18] proposed few-shot segmentation to remove the labeling effort and increase the generalization capacity of a model given few data for the first time.

1.2 Problem Statement

Few-shot segmentation task utilizes a base dataset containing adequate images with their annotations whose classes, $\mathcal{C}_{base}$, are disjoint from novel classes, $\mathcal{C}_{novel}$, in which dense predictions are fulfilled with few data and their annotations. K number of available data and their annotations belonging to the novel classes constitute support set $\mathcal{S}$ for testing which are expected to guide a model M to make predictions for query image $\mathcal{I}_q$, which is dubbed as K-shot segmentation. The support set is formally represented as $\mathcal{S} = \{\mathcal{I}_{s_i}, \mathcal{M}_{s_i}\}_{i=1}^{K}$, where $\mathcal{I}_{s_i}$, and $\mathcal{M}_{s_i}$ correspond to i$^{\text{th}}$ support image and its dense ground truth mask. On the side of training, support set for training is sampled from base dataset along with query set which consists of the query image and its ground truth, sharing its class with the chosen support set. The aforementioned classes are treated as a novel class during training in order to perform episodic training, where pixels belonging to chosen class are assigned as foreground while pixels from all other classes are considered as background. Query set is formally represented as $\mathcal{Q} = \{\mathcal{I}_q, \mathcal{M}_q\}$, where $\mathcal{I}_q$ and $\mathcal{M}_q$ correspond to the query image and its dense ground truth mask. The model, M, is trained by backpropagating binary cross entropy loss between prediction $\hat{\mathcal{M}}_q$ for query and ground truth $\mathcal{M}_q$ over tasks, named as episodes involving the selected support set from the base dataset with the accompanying query set.

1.3 Motivation

Few-shot segmentation addresses the problem of making pixel-wise predictions for a target image, called a query, from an unseen class with the guidance of a support image from the same category. Inspired by the few-shot classification task [19], most methods utilized the episodic training strategy in which the gradients are averaged over tasks named as an episode. Each episode is sampled from a dataset whose classes are disjoint from a test case where only a few data are available. These episodes are used to imitate the test case during training to prevent overfitting. Despite this intention, a model trained with this strategy tends to mistake segments from seen training classes, referred to as *base classes*, as *novel classes* because of constantly

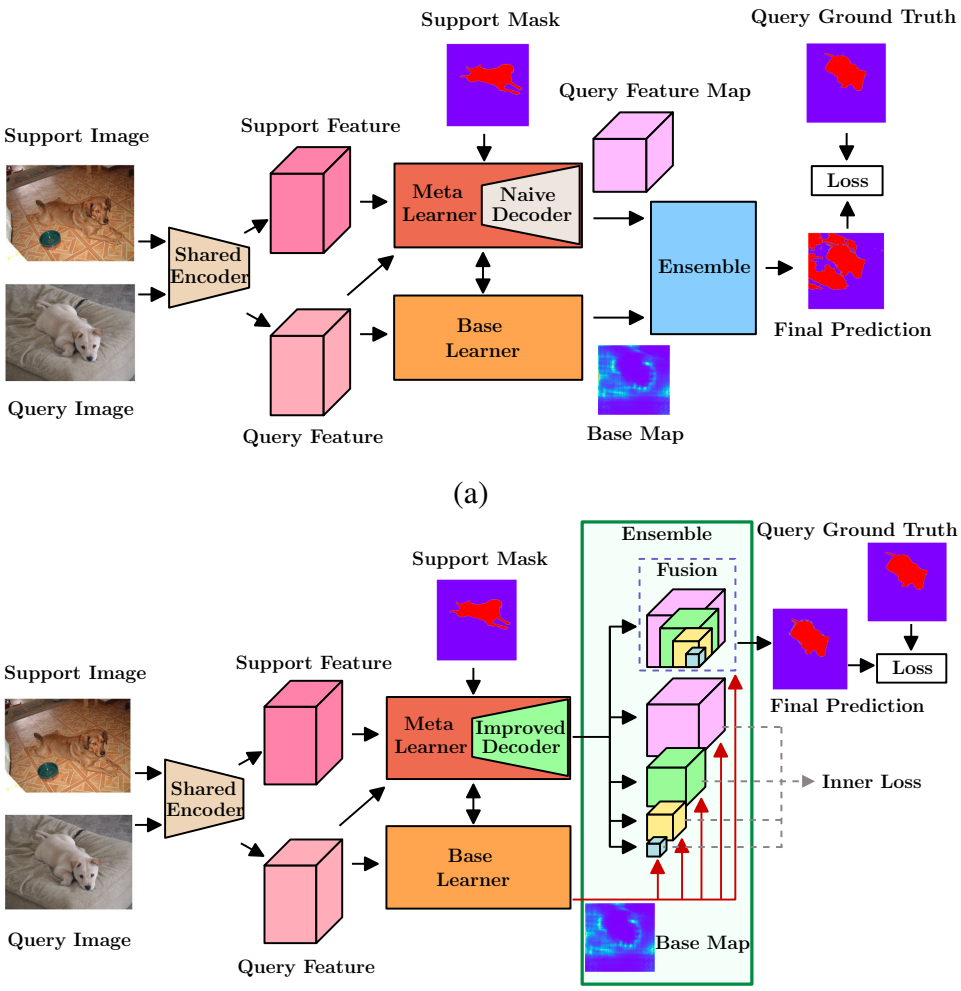

Figure 1.1: (a) Overview of BAM [1]. Support and query features are used by meta learner to extract support feature map while base learner provides guidance for the base classes and leads the meta learner to focus on novel regions via ensembling. (b) Our proposed method. The decoder for meta learner is improved such that query feature map is obtained at multi-scale. Support feature map is compared with query feature maps at multi-scale to obtain enriched query features. Query predictions obtained from enriched query features at each scale are ensembled with the base map as well as the prediction obtained from the fusion of them. Inner losses are computed at different scale levels and the final prediction is obtained from the ensemble of the base map with the predictions from the fused query feature maps. (Best viewed in zoom)

experiencing the same set of classes during training. Hence, the co-occurrence of novel and base classes in the same scene causes entanglement between features of pixels that are part of the novel and base categories.

To prevent this entanglement, prediction of the supervised model, which is trained on base classes, guides the meta learner, which is responsible for detecting novel areas. Meta learner is directed to areas not occupied by the base classes so that contradiction between the base learner and the meta learner is avoided via the ensemble model entailing both base and meta predictions [1]. On the side of meta learner, candidate objects in query image might not cover as same area as those in support images, so the model should compare support feature map with query feature map at different resolutions to disentangle adjacent regions around novel segments [11]. As shown in Fig. 1.1, the ensemble of base and meta learner without improved decoder fails to distinguish background from foreground since naive decoder, which is designed for the supervised scheme, lacks to combine features at different resolutions in favor of complete query prediction. Hence, we transform the naive decoder into an improved decoder such that not only does it correlate the support image with the query image at multi-resolution but also it benefits from merits of base learner at multi-resolution. In this regard, we hypothesize that there are cases where it is not enough that base learner discourages meta learner from base regions at single-scale. Our experiments verify that the improved decoder and ensembling the predictions at multi-scale outperform the decoder equipped with ensembling the prediction at single-scale.

1.4 Contributions

Our contributions in this thesis are two-fold:

- We alleviate the spatial inconsistency and the bias problems together with the assistance of our proposed decoder that seeks to remove bias at multi-resolution.
- Our proposed method achieves new state-of-the art performance on both PASCAL-5^i (mIoU @ 1-shot: 68.59%, mIoU @ 5-shot: 72.05%) and COCO-20^i (mIoU @ 1-shot: 47.16%, mIoU @ 5-shot: 52.50%) datasets for few-shot segmentation task.

1.5 The Outline of the Thesis

Chapter 1 presents an introduction to the need for few-shot segmentation, followed by the formulation of the problem at hand. Additionally, we provide a detailed discussion of the motivations and contributions of our proposed methodology in this dissertation.

Chapter 2 provides an overview of the relevant literature on few-shot segmentation, categorized based on the problem they aim to solve. Notable works from each category are presented and summarized in their respective subsections.

Chapter 3 provides the necessary background information for comprehending our novel approach, followed by a thorough discussion of our proposed method.

Chapter 4 provides an introduction to the benchmark dataset and the evaluation metrics that are employed to measure the performance of the methods. The experimental configuration, including hyperparameters, is also outlined in this chapter. In this chapter, our proposed method is evaluated quantitatively and qualitatively to demonstrate its effectiveness. Additionally, this chapter contains ablation studies ascertaining the individual contributions of each component used in our methodology.

Chapter 5 provides a concise summary of our proposed method, highlighting its strengths and limitations.

CHAPTER 2

OVERVIEW OF FEW-SHOT SEGMENTATION

Few-shot segmentation is a task of predicting pixel-wise labels for new object classes in query images, given only a few annotated support images. It extends the few-shot learning paradigm to dense prediction tasks and mostly relies on specialized architectures that perform meta-training on a base set followed by meta-testing on a disjoint set. Meta-training is performed on classes from dataset which consists of ample images with their annotations, which treats those classes as novel to match training scenario with test one. On the other hand, meta-testing is performed on classes, which is disjoint from rich and accessible dataset used during meta-training.

Shaban *et al.* [18] proposed dual-branch network whose support branch undertakes to generate classifier weights for query branch. Inspiring by prototype concept in few-shot classification [20], support image with its mask passes through prototype learner model, which is followed by global average pooling to acquire prototype depicting category specific information [21]. Extraction of prototype by averaging lead to lose of details, and it is not obligatory that each support pixel is beneficial for segmentation of each query. In addition, appearance of support can differ from query such that only mining features from support is not sufficient to close intra-class gap, and excavation of query is required to bridge the gap. We name this problem as imbalance in details, which majority of few-shot segmentation struggle to solve.

Another finding is that freezed backbone and inductive inference can lead to reduction of model generalizability. It was demonstrated that optimizing part of model during meta-training [7] and transductive [22] inference mitigate overfitting. We name this problem as inter-class gap. Most methods take precautions to deal with this problem while proposed method in thesis acknowledges overfitting, detect such regions, and

discourage model from predicting those regions as novel.

Other common problem is notorious for vision tasks, called spatial inconsistency [11]. It is not possible that representative template from support captures appearance of category at different scales. Therefore, content in support should be compared with query at different scales. Proposed method in this thesis addresses this problem by establishing such mechanism.

Episodic learning paradigm perceives whole background as if it comes from same distribution. However, hidden background can consist of objects from novel categories in which their discriminability is hurted since they are regarded as single class [2]. We name this problem as misinterpretation of background, which [2], and [23] address this problem.

Another bunch of works [9, 24] believes that processing of correspondences is more suitable than processing features for learning-to-learn paradigm since matching patterns tends to be more class agnostic compared to the comparison of features. Since their processing lead to increase reliability of correspondences, we name this problem as correspondence reliability.

Other problem overlooked by most methods is scalability of performance according to number of supports used. Joakim *et al.* [8] demonstrated that up-to-date methods show comparable performance at 1-shot case compared to them while they fail in 5-shot case compared to their method. We name this problem as scalability with number of support data.

Last problem is that most methods are not capable of segmenting thin objects, which manifest itself on supervised segmentation as well. We name this problem as thin object issue.

In next subsections, methods addressing to each problem is discussed in detail manner. Fig. 2.1 demonstrates which problems each method address with taxonomy.

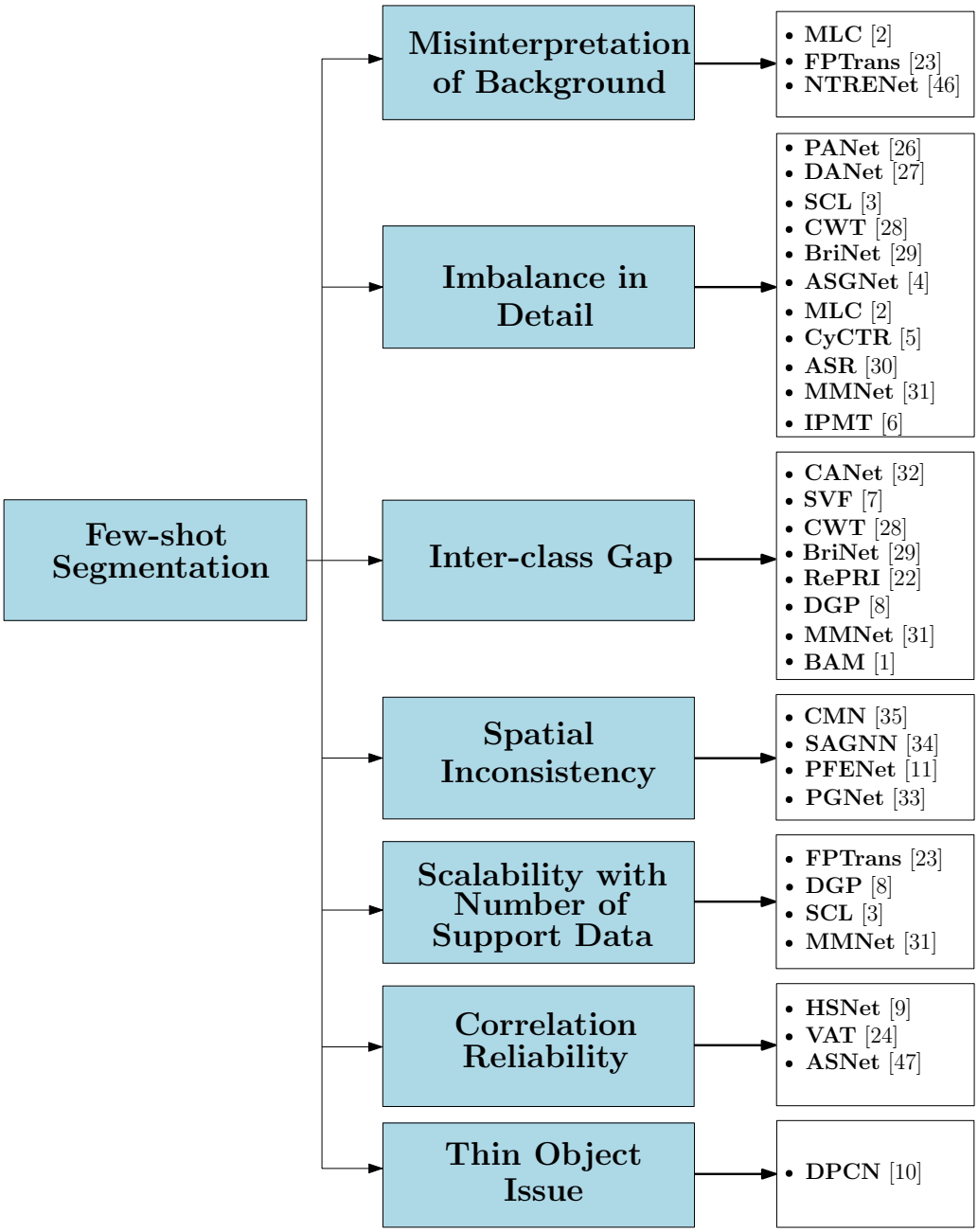

Figure 2.1: Taxonomy of Few-Shot Segmentation

2.1 Few-Shot Segmentation Problems

2.1.1 Misinterpretation of Background

Many existing few-shot learning techniques conceptualize the background as a single entity; however, it can encompass multiple objects that belong to both the training and testing classes. When a unified label is assigned to the entire background, the model is trained to map features from various classes to that single label. Mining Latent Class (MLC) [2] refers to this problem as feature undermining. The background component in query images often comprises objects belonging to both novel and base classes, leading to incorrect categorization of these objects as background as shown in Fig. 2.2. This results in a decrease in discriminability not just for the novel classes, but also for the base classes, as their embeddings are treated as background by the models. To address the problem of feature undermining, [2] presents a method for obtaining prototypes for each class, including the background, by utilizing masked average pooling. The foreground prototypes are collected in one set, while background prototypes are separated into another set. The authors posit that latent classes share commonalities with base classes, so they apply k-means clustering on the foreground set to uncover these commonalities. Embeddings in the background set are basically averaged to get a global background embedding. Each image is then labeled with one of these K+1 commonalities by nearest neighbor classification. During classical meta-training, they apply multi-class segmentation for extra images with their pseudo-masks. This solution prevents latent classes from being treated as the same as background.

Figure 2.2: Novel class in background, namely person, is treated as background, leading to reduction of its distinguishability [2].

Different from MLC [2], the Feature Proxy Transformer (FPTrans) [23] employs a strategy to determine K points exhibiting the greatest interpoint spatial separation in background, which serve as the centers in the Voronoi Clustering methodology. Subsequently, each point in the background is assigned to the closest cluster center, and the regions associated with each cluster center are transformed into binary masks. FPTrans employs a prompting strategy to adjust a transformer to the requirements of the task at hand. The prompts are derived from the feature maps obtained through the Vision Transformer [25]. The foreground prompt is represented by the average of the pixel-level features within the foreground region, while the background prompts are generated through the computation of the average of the pixel-level features within the binary masks. The prompts are then subjected to processing by the transformer to produce background proxies, which serve as weights for the background classifiers. The cosine similarity between each query pixel and the background classifiers is calculated, and the maximum similarity score is utilized as the background logit. In this manner, the model is expected to effectively distinguish relevant background regions from novel regions, thereby alleviating the problem of misinterpretation by avoiding the prediction of novel regions as background. Additionally, FPTrans facilitates a mechanism that attracts the features of the foreground support pixels towards the features of the foreground query pixels while simultaneously repelling the foreground and background pairs between the support and query.

2.1.2 Imbalance in Details

As single support embedding can not reflect all details that belong to a query image, there might be details that do not co-exist in both query and corresponding support image, so inconsistent regions between support and query should be determined and eliminated on support side to prevent redundant details or noise adaptively. If each query pixel attends relevant parts of support image where relevancy is generally quantified by similarity metrics such as cosine similarity, noise is removed. Although there are proposed methods to alleviate discrepancy between support and query, this problem continues its importance, and better treatment might lead to performance improvement. *What about details that are found in query image but not in support image*? It is an open question to diagnose part of query differs from support and per-

form information exchange depending on lacking content on support side to segment query. In absence of such a mechanism, few support image is augmented without any exception in order to closing intra class gap. In summary, details found in support but not in query results with noisy segmentation, while details found in query but not in support results with incomplete segmentation. In other words, an imbalanced detail level between support and query prevent the models from making correct segmentation.

Prototypical Alignment Network (PANet) [26] adopts a Siamese encoder instead of a two-branch network as it posits that both support and query features should be extracted from the same encoder to facilitate metric learning. This approach not only reduces the number of parameters but also allows for a better comparison of features. Similarly, few-shot segmentation techniques also use support images to extract feature vectors for comparison with query feature maps. However, they incorporate a regularization mechanism to ensure that the network successfully segments the support image from the query image. Specifically, they interchange the roles of the support and query images to derive consistent prototypes for each class. Finally, they evaluate the distance between the support and query prototypes for both regularized and non-regularized models and demonstrate through empirical analysis that feature consistency is improved when regularization is applied.

The salient features of an object compete with each other to capture attention, and some features are more salient than others. For instance, while searching for a coffee cup, our eyes may be drawn to a table. Similarly, in the context of few-shot segmentation, the connection between the query and support graphs becomes strong only in a narrow and specific region of support nodes. Therefore, only this limited region has the ability to transfer knowledge, and occlusion of this region can lead to a failure in the segmentation task. To address this issue, Democratic Attention Network (DANet) [27] proposes a regularization technique that encourages a larger area of support to participate in the decision process, thereby increasing the diversity of connections and widening the region of interest in the support image. While low-level features have been shown to provide benefits in segmentation, many studies have focused on high-level semantic guidance rather than comprehensive multi-scale guidance. DANet applies this regularization to features extracted from different lay-

ers of the model hierarchy and fuses the processed features to transfer multi-scale information.

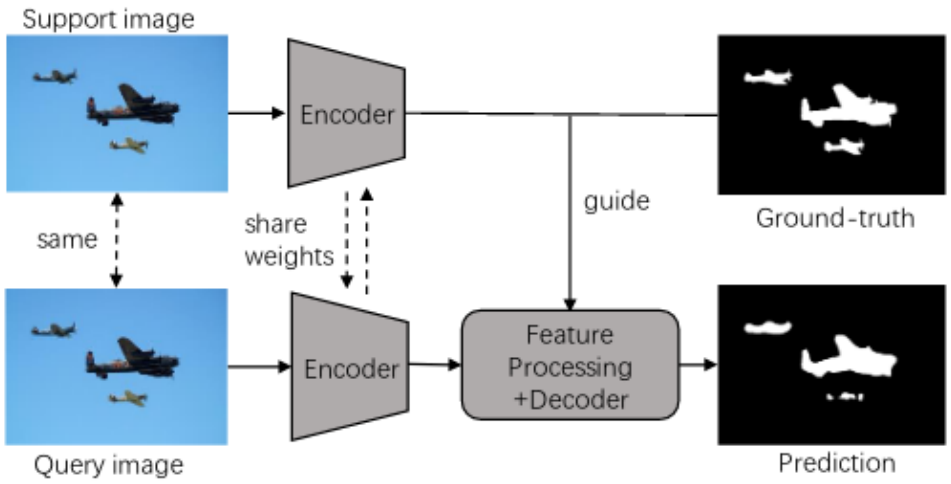

Figure 2.3: Despite selecting the support image identical to the query image, the model fails to segment certain regions of the query. This indicates a loss of information resulting from the averaging operation [3].

In previous few-shot segmentation approaches, the support feature vector is obtained through an averaging operation, which can lead to the loss of necessary information for the segmentation of the query. This issue persists even when the support and query images are identical as shown in Fig. 2.3. Due to the restricted data, averaging fails to capture the general expectation of the support set. To overcome this limitation, Self-gudided and Cross-guided Learning (SCL) approach [3] proposes to predict a mask of the support image with its prototype and identify the false negative regions of the binary mask to locate where the lost information originates. The lost information is then modeled to predict the query segmentation. However, since SCL presumes that the lost information can be accurately represented by a single prototype, its solutions are suboptimal. Despite this drawback, SCL confers an advantage over the simple averaging approach. It is noteworthy that these efforts were aimed at addressing the deficiencies of the averaging operation, which has been demonstrated to be insufficient in preserving the required information for successful few-shot segmentation.

In the Classifier Weight Transformer [28] approach, it was deemed unrealistic to con-

tinually adapt the parameters of the encoder, decoder, and classifier to each new task due to the excessive number of parameters in the encoder and decoder. As a result, the feature encoder and decoder were pre-trained on the meta-training set and then frozen prior to meta-training the classifier component. This strategy was based on the assumption that the features extracted from the meta-training set would possess sufficient generalizability to transfer to novel tasks, which serves as a justification for the aforementioned approach. In the CWT approach, the classifier is fine-tuned on the support set from the new task. However, the query images may differ visually from the support images. To address this issue, the fine-tuned classifier weights are adapted to the query features using a transformer with the assumption that the support classifier attends to relevant parts in the query. The transformer extracts an attention map between the classifier weights and the query feature map, after which the query pixels are weighted averaged using these coefficients to perform the adaptation. This approach enables not only the revelation of co-occurrent details but also the dynamic capture of query-intrinsic details in the adapted classifier weights.

BriNet [29] stated that two instances of the same class do not have to share the same properties. They propose to transfer features found in only query to support and vice versa. Collaboration between support and query forms feature representation that ideally has no intra-class gap. For example, embeddings from person with glass can be combined with that of person without glasses. In addition to masked global average prototypes, local averaging with determined sizes is performed in multi-scale correlation module to obtain prototypes that retain high detail levels compared to masked global average pooling. In multi-scale correlation module, image is divided into s parts in both transversal and longitudinal direction. Average pooling is applied on each part. In addition to global average pooled vector, obtained weights with size of $c \times 1 \times s$ and $c \times s \times 1$ are also convolved with query, and obtained responses are fused.

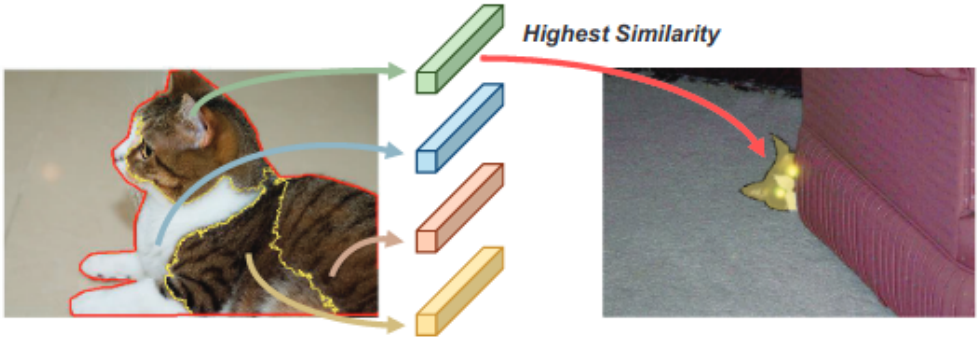

Figure 2.4: Foreground of support is clustered into regions, and each query pixel only attends most similar partitioned region in support [4].

Adaptive Superpixel-guided Network (ASGNet) [4] aims to address the issue of losing spatial-semantic information in representation of an entire object with a single prototype. ASGNet proposes a method that adaptively chooses the number of prototypes and their spatial extent based on image content, such that each prototype represents parts of the object with similar characteristics. For example, prototypes are allocated to the head, body, and leg regions in a human image, while prototypes are allocated to the wheel, hood, and wing mirror in a car image. This is achieved by using a trainable super-pixel sampling strategy that separates the feature map into representative groups. ASGNet notes that the scale of the image affects the number of prototypes required, with larger images requiring multiple prototypes to represent excessive details, while small-scale objects can be represented with just a few prototypes. The method involves two main components: super-pixel guided clustering and guided prototype allocation. The super-pixel guided clustering initializes the seed using the MASK-SLIC algorithm and updates the super-pixel centroids using a weighted average of support pixel features. The guided prototype allocation computes the cosine similarity between all prototypes and the query features and selects the prototype with the highest similarity to the query pixel as shown in Fig. 2.4. The network also includes a feature enrichment module for multi-scale feature aggregation. The super-pixel number is determined by dividing the total number of pixels in the mask by the average number of pixels in the seeds, with an empirical limit of 100 pixels. The authors observe that adaptive prototype selection slightly improves performance compared to fixed selection. The network utilizes a spatial regularization

by summing the cosine similarities between each prototype and the query pixel. The part-based matching approach provides robustness to segmentation and the network only utilizes the most similar prototype for comparison, preventing other representative areas from deciding the prediction of the query pixel's class.

MLC approach [2] claims that it is difficult for a few support images to mimic real class-wise statistics, making it sub-optimal to merely utilize the current supports for prototype estimation. This is known as prototype bias. To address the problem of prototype bias, they propose to update support prototypes during episodic training and testing. The global background prototypes are moving averaged with each current background prototype during episodic training or testing. On the other hand, foreground prototype rectification is only performed during inference. In their work, support prototypes are consolidated by transferring most similar group of pixels from each of retrieved N-nearest image in base training set, where each group corresponds to pseudo-labeled regions acquired with K cluster center of foreground support prototypes in meta-training set. Region embeddings are weighted averaged according to similarity to the foreground support prototype, and the resulting feature is multiplied with predefined constant and added to support foreground prototype to update it. This approach aims to obtain different aspects of a class with the help of pseudo-labeled samples. Although it is not proved, it is believed that the groups contain transferable knowledge. Therefore, pseudo labeled meta-training set augments support set with hope of closing intra-class gap. In summary, the Cycle-Consistent Transformer comprises two critical modules that leverage deformable attention and cycle consistency to enhance the performance of few-shot segmentation. The cross-alignment block samples a subset of foreground and background pixels to reduce the computational complexity while retaining only the cycle-consistent pixels in attention ensures that only the relevant concepts are preserved.

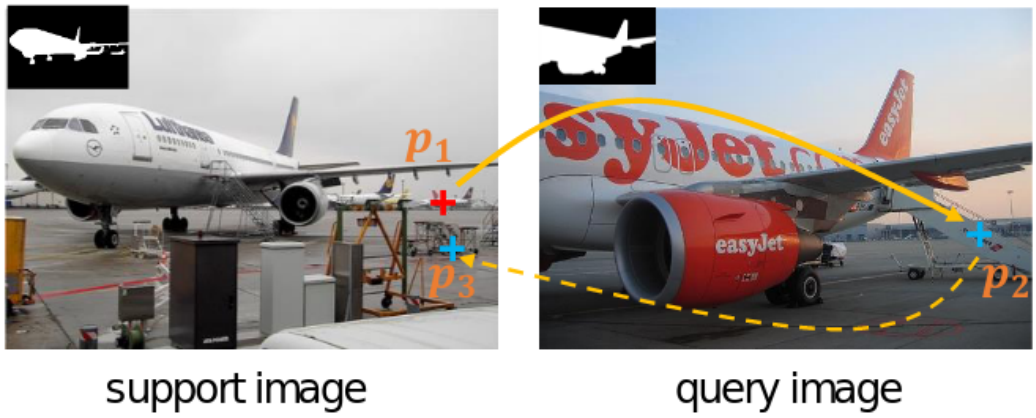

Figure 2.5: The sample does not conform to cycle-consistency paradigm [5].

The Cycle-Consistent Transformer [5] comprises two critical modules, namely the self-alignment and cross-alignment blocks. The former employs a self-attention mechanism to query the feature map, whereby the features of each pixel are treated as individual tokens. However, it is challenging to establish connections between pixels that originate and terminate in the same region since the foreground and background of the query are unknown. To overcome this limitation, the authors proposed the utilization of deformable attention, which predicts a predefined number of offsets for the pixel to be attended to, as well as the corresponding attention weights of the pixels. In their ablation study, they validated that the replacement of simple self-attention with deformable attention leads to a significant performance boost. On the other hand, the cross-alignment block leverages the support feature map as the key and value for self-attention, while the transformer's query is derived from the query feature map. Here, the background pixels of the support are retained in the attention since they can offer context-specific information on the category in question. To address the computational complexity associated with a large number of tokens, particularly in the K-shot setting, the authors uniformly sample a predetermined number of foreground and background pixels in equal proportions. Furthermore, the authors propose retaining only cycle-consistent pixels in attention. Cycle consistency is determined by identifying the most similar query pixel to its corresponding support pixel, followed by the most similar support pixel to that query. A pixel is deemed cycle consistent if its starting and ending labels match based on the support mask. Fig. 2.5 illustrates

an example in which point p_1 in the foreground of the support set matches the most similar query pixel p_2 to itself. Subsequently, point p_2 is matched to the most similar support pixel p_3 to itself. Since p_1 is in the foreground while p_3 is in the background, p_1 is not cycle consistent. This mechanism preserves only the concepts that occur simultaneously in the query and support and addresses the imbalance in details.

The Anti-Aliasing Semantic Reconstruction [30] introduces the notion of semantic aliasing, whereby common features can express two distinct semantic classes that share similar content. This phenomenon results in the association of semantically dissimilar objects, such as a dog and a cat sharing the feature of fur. To address this challenge, the authors propose optimizing a space constructed by orthogonal basis vectors, where each vector is associated with a distinct base class. They then reconstruct the support and query feature maps using these basis vectors, and empirical evaluations demonstrate that this approach eliminates the problem of semantic aliasing. To obtain the reconstruction weights, the authors employ a softmax layer to derive the magnitude of each basis vector, which in turn corresponds to the reconstruction weight. They design a loss function to penalize basis vectors from different classes that are not orthogonal while ensuring that basis vectors from the same class are colinear. During meta-training, the authors enforce high reconstruction weights for the corresponding class to decouple the basis vectors from one another. Lastly, they perform semantic filtering by only utilizing query features that align with the reconstructed support vectors while treating the remaining features as noise, such as background cluttering. This filtering mechanism enables the disambiguation of semantically confusing classes. In summary, the Anti-Aliasing Semantic Reconstruction paper proposes a novel approach to address the issue of semantic aliasing, which involves optimizing a space constructed by orthogonal basis vectors, reconstructing the support and query feature maps, and employing a loss function to penalize non-orthogonal basis vectors while ensuring colinearity within the same class. The proposed method also employs semantic filtering to disambiguate semantically confusing classes.

The activation propagation module of the meta-class memory network [31] tackles the challenge of imbalance in detail by initially computing the cosine similarity between the activation map of each query and support pixel. Activation maps are extracted in

meta-class memory module and explained in next section since it addresses another problem. This similarity matrix captures the relationship between each query pixel and all foreground support pixels. However, to counteract the effect of background support pixels, the similarity values corresponding to them are replaced with negative infinity. Subsequently, a softmax operation is performed along the row dimension of the matrix to obtain the attention weights. For each query pixel, the corresponding support activations are multiplied by their respective attention weights and then aggregated. The resulting vector is then multiplied with the query activations to regulate the activity of the features, such that only common features are encouraged to maintain their existence. This effective approach effectively addresses the issue of imbalance in detail.

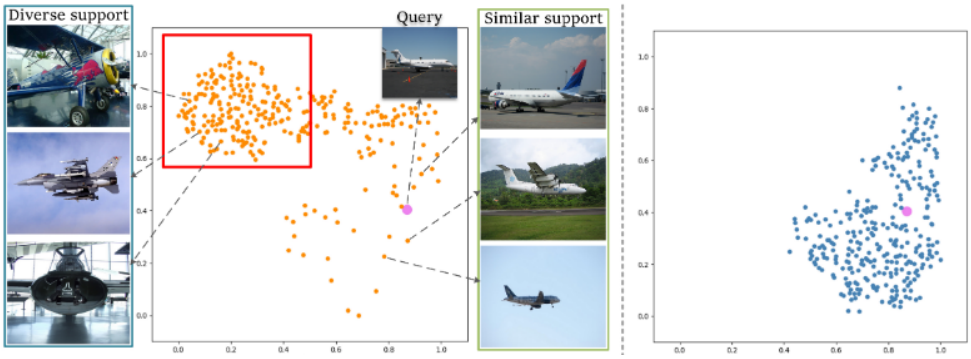

Figure 2.6: Inter-class gap problem is illustrated with help of t-SNE graphs. Left-hand side figure portrays embedding space before application of [6] while right-hand figure represents embedding space after its application.

The Intermediate Prototype Mining Transformer (IPMT) [6] aims to reduce the intra-class gap between the support and query images in an iterative manner. Fig. 2.6 illustrates the severity of the intra-class gap problem through a t-SNE graph. The left-hand side of the figure displays the situation before IPMT is applied, while the right-hand side shows the situation after its application. The results demonstrate that IPMT effectively reduces the intra-class gap, as depicted in Fig. 2.6. To achieve this, the IPMT employs a learnable prototype that is used to mine prototypes from both the query and support images via masked cross-attention. Since the mask for the query is not available, the IPMT predicts an initial mask for the query from the initial query feature map. Once the masked cross-attention is applied to the query and support,

two prototypes are obtained that are adapted from the query and support, respectively. The average of these two prototypes is then added to the learnable prototype at the current step. IPMT then computes the dot product of the linearly transformed updated prototype with the query and support images, respectively. Segmentation loss is applied to both the query and support images to guide the network in bridging the class gap. Subsequently, the updated prototype activates the query image to suppress the background region. The refined query map, support map, latest query prediction, and support feature map are then used in the next iteration for the mining of a new prototype. As the initial mask for the query is erroneous, the iterative mechanism progressively improves both the refined map and predictions, leading to the reduction of the intra-class gap between the support and query images.

2.1.3 Inter-Class Gap

Approaches assume that transferable knowledge exists in base set and works to segment images from unseen classes during training. This strong assumption loses its validity in proportion to discrepancy between base and novel dataset. However, there exists risk of memorization of the few data that discourages the adaptation of novel classes regardless of amount of the shift, so few-shot community generally ignores optimizing classifier with few data. Recent studies show that the severity of overfitting is exaggerated in few-shot segmentation, and adapting novel classes improves segmentation performance.

CANet [32] argued that middle-level features might serve as a common denominator that underlies semantically similar object classes. For instance, the feature of a "wheel" could be regarded as a mid-level feature, which is likely to be more beneficial in predicting novel classes like "bus" or "truck" than high-level features. Hence, the transferable knowledge in the middle layers of a pre-trained network can be utilized to infer the location of novel classes by exploiting the underlying shared concepts. In light of this, CANet proposes to incorporate these predictions as a prior to a class-agnostic module that distinguishes foreground from background in a progressive manner to refine the final segmentation result. Therefore, they devise an iterative optimization module that gradually improves the predictions by refining them at each

iteration.

An alternative perspective posits that the transferability of pre-trained models is hindered by the fact that ImageNet weights are primarily suited for classification tasks and are tailored to ImageNet's specific classes. While overfitting is a genuine concern in few-shot segmentation, it is suggested that pre-trained network weights must be fine-tuned to eliminate the noise caused by this issue. Therefore, the Singular Value Fine-tuning (SVF) method [7] proposes that only a small portion of the model should be updated. This approach entails decomposing the weights into three matrices using Singular Value Decomposition, where the matrices excluding the one that retains the singular values are deemed to carry semantic cues. Altering these matrices is believed to degrade performance. The SVF only fine-tunes the singular value matrix to avoid this issue as shown in right-hand side of Fig. 2.7, thus suppressing redundant semantic cues while preserving generalizable information.

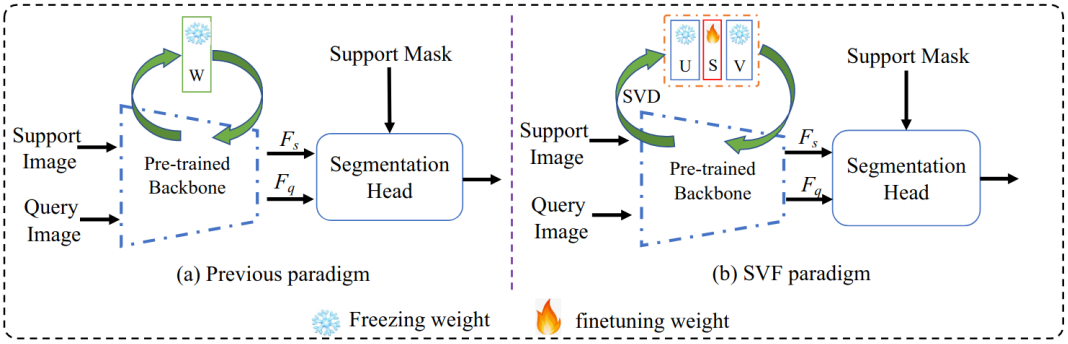

Figure 2.7: Optimizing part of model can overcome overfitting problem. Left-hand side figure completely freezes backbone, while [7] selects particular weights cleverly, after which it fine-tunes them only.

In the CWT [28] approach, the classifier is fine-tuned on the support set of the new task to adapt the classifier to the specific task and thus bridge the inter-class gap.

BriNet, as described in [29], introduced an online refinement module that alternates the roles of query and support. The prediction for the query is treated as a pseudo-mask and is used as support, and the binary cross entropy loss is computed for the support image. This process is repeated until the determined mean Intersection over Union (mIoU) threshold between the support mask and its prediction is surpassed.

This design allows the model to adapt to the current task at hand continuously.

RePRI [22] is a transductive inference-based approach that prioritizes accuracy on test data compared to inductive inference-based one over-generalizing to all inputs. Despite offering improved performance in low-data scenarios, it requires more time to learn new tasks compared to inductive methods. RePRI leverages three loss functions to train the classifier for a target task. The first loss, cross-entropy, is applied between the support set labels and the model's predictions. However, this loss has the tendency to result in overfitting of the support set. The second loss, the Shannon entropy of the query examples, ensures confident predictions on the query examples and shapes the decision boundary through low-density regions of the feature space. The third loss calculates the KL divergence between the query image's background-foreground probability density function and that of the initial classifier, which is defined as the average of the support features.

Dense Gaussian Processes [8] provide not only the mean prediction but also the associated uncertainty for a given query, owing to their ability to learn function distributions where each instance represents a random variable. The presence of uncertainty allows the model to assess the reliability of its predictions and return a prior query feature map when the available support information is deemed insufficient based on the associated uncertainty. Furthermore, the uncertainty enables the model to measure the correlation of each pixel with its neighboring pixels and refine its predictions based on a consensus within the decoder's kernel. Without such conditioning, elimination of noisy regions would not be possible. To address inter-class gaps, Dense Gaussian Processes utilize low-level feature maps that are generally more generalizable, such as boundary regions that are commonly activated in such feature maps. Rather than increasing the adaptability capacity, the model benefits from using a feature map with a lower risk of overfitting.

The Meta-class memory module of Meta-class Memory Network [31] postulates that middle-level features extracted by pre-trained networks may not be readily transferable to novel classes. In response, the module introduces 50 learnable parameters, referred to as meta-class embeddings, which encode shared knowledge across classes. These meta-class embeddings then interact with query and support features through

the computation of their cosine similarity with the memory elements. The resulting similarity values are subsequently transformed by a sigmoid function to generate activation maps. The findings from an ablation study corroborate the assertion that the integration of the meta-class memory module leads to improved performance.

2.1.4 Spatial Inconsistency

In a supervised setting, adaptive average pooled feature maps at different scales are concatenated to contain not only global both also local context. For example, a boat that is zoomed in a picture can be mistaken for a car; however, adding global context signs that a car can't be over water and environment resembles a port rather than a road, so aggregation of feature maps at different scales works to acquire scale-invariant representations with the help of abundant number of training data. Architectures designed for supervised cases fail to provide scale invariance in few-shot scenarios since contextual relationships are not figured out by a handful of data.

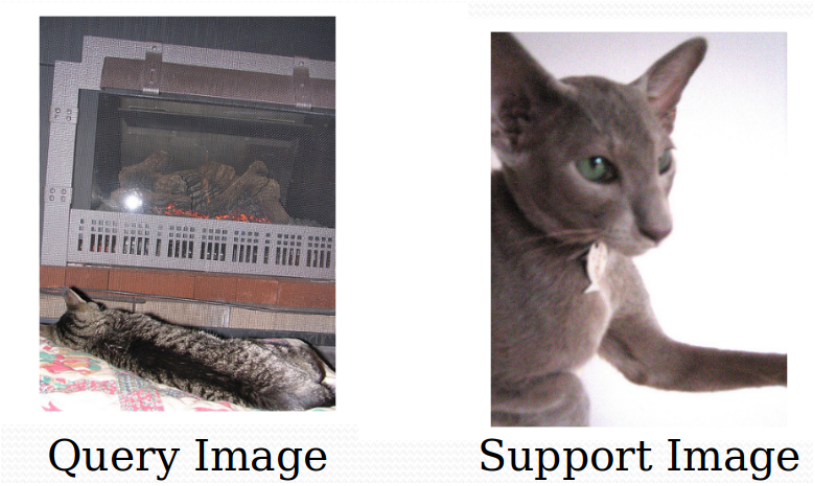

Figure 2.8: Discrepancy between sizes of object in query and support image leads to spatial inconsistency problem.

Interested objects in query image might not cover the same area as those in support image. For instance, as shown in Fig. 2.8, the cat in the query image does not cover the same area as the corresponding cat in the support image. As solution, model can separately compares support feature map with query feature map at different scales

to become scale-agnostic. On the other hand, controlled information flow across different scales are satisfied to gather discriminative cues from different resolution, where control mechanism allows favorable information exchange. Methods design control mechanism, multi-scale processing, and fusion scheme to overcome the scale problem.

The prototype-based methods in computer vision often rely on making the feature vector at each pixel of a query image as similar as possible to the global descriptor of a support image, despite the fact that not all parts of the support image may align with the query image. This discrepancy can introduce noise into the network. To address this issue, the Pyramid Graph Network (PGNet) model [33] proposes a solution in which each query feature vector gives attention to relevant parts of the support image. PGNet applies this attention mechanism in a pyramid structure, which allows for correspondences at different scales to be captured. The model generates a pyramid of query graphs by modeling predefined sub-regions of the image as nodes in the query graph using adaptive pooling. PGNet is inspired by graph attention networks, which collect messages from neighboring nodes based on attention mechanisms such as self-attention. The goal of this type of network is to adaptively propagate label information from the support graph to the query graph. Specifically, PGNet combines the support and query graphs as a bipartite graph, with each node of the query graph connected to the foreground part of the support graph. The model then computes the correlation between the foreground support nodes and each query node. The correlation-weighted average of the foreground support features is then concatenated to the query graph as guidance.

Other methods in computer vision model different scales of the query feature map, but they may encounter issues where certain scales dominate over others, particularly in the presence of objects of varying sizes. To address this, a SAGNN [34] that facilitates favorable information exchange between scales can be utilized to remove noise effectively. This network is distinguishable from others by its ability to provide controlled high-order information flow through the use of a graph neural network. Unlike the PFENet model, which fuses multi-resolution feature maps in a top-down hierarchy, this design allows for information flow in all directions. The nodes of the graph correspond to multi-scale fused feature maps, while the edges represent the

directed relation between two different resolutions. The multi-scale query features are formed by concatenating the average pooled query feature map with the global average support embedding and the maximum support-foreground response. The maximum support-foreground response serves a similar purpose as the prior mask in the PFENet model. Additionally, it is worth noting that each node, rather than its neighbors, contributes to the message-passing process to include information from its resolution. The network utilizes a ConvGRU structure to determine which part of messages are accepted during updates. After a determined number of message-passing iterations, the updated nodes are conveyed to a read-out module to synthesize information from all resolutions. The synthesized features are then passed through a classifier to predict the query mask.

The Cyclic Memory Network [35] extracts multi-resolution feature maps by combining a support prototype, an adaptive average pooled query feature map, and a prior map similar to the SAGNN. However, unlike SAGNN, each resolution in CMN has key and value maps that are mapped by different convolutional layers. Specifically, the value map at a particular resolution serves as a query, and a similarity matrix between its key map and all other resolutions is computed to obtain attention weights. These attention weights express how each pixel at the query resolution attends to all other pixels. The features at cross resolutions are then multiplied by these attention weights before aggregation. Notably, each resolution takes turns serving as the query, resulting in complex interactions between all resolutions. Furthermore, aggregation is performed using a recursive block for reasoning about which parts of upcoming knowledge are most beneficial and which parts of the query resolution are preserved.

2.1.5 Scalability with Number of Support Data

Performance of majority of models marginally increases, although the number of support examples in its support set increases as shown in Fig. 2.9. It signs that methods lack leveraging whole support set, especially when shot number exceeds 5. This might be caused by fusion type, such as averaging that smooths out features in support set. Therefore, spatial integrity of image should be preserved to exploit each piece of feature corresponding to each pixel.

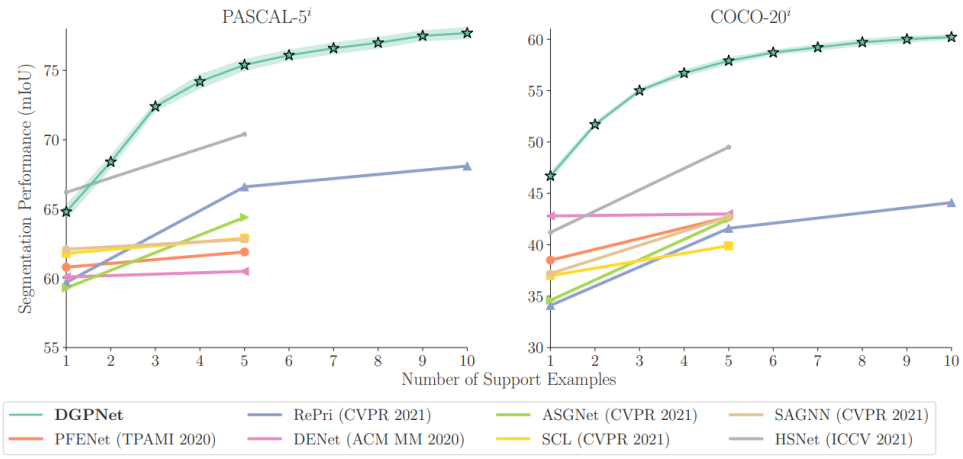

Figure 2.9: Most up-to-date methods do not show comparable performance in 5-shot case relative to that in 1-shot [8].

In FPTrans approach [23] described in Subsection 2.1.1, the prompts serve as a connection between the support images and the query image. To avoid the quadratic complexity of the transformer, the prompts are associated separately with each support image and the query image. These interactions result in distinct hidden prompt states, which are then averaged to reveal the common properties shared by the query and support, referred to as prompt synchronization. The advantage of FPTrans lies in this unique mechanism for facilitating interactions between the query and support at all levels of the model, which differentiates it from other methods.

Dense Gaussian Process [8] utilize Gaussian process function to struggle with scalability problem. Gaussian process functions make observations of support pixel features mapped to output masks as training data and assume that query and support mask values come from the joint Gaussian distribution. The equation giving the mean query pixel respects the correlation between each support pixel with the precision matrix. Although some support pixels do not directly represent the foreground, their connections with all other support pixels might require them to be closer to the foreground. Such complicated linking mechanisms are modeled with a specific kernel function designed to capture the covariance of the support with itself. For instance, the squared exponential kernel enforces similar features to be correlated in the output space. Furthermore, the covariance between the query and support allows us to model

how each support and query co-occur simultaneously. The support pixels are aggregated with weights in the row of this matrix. For example, the first row relates the first query pixel in the raster direction with all support pixels, e.g. the second entry in the first row shows how likely the occurrence of first query pixel requires the occurrence of the second support pixel in the raster direction. This model does not compromise the granularity of each pixel, so it can increase performance when the support set expands. In addition to these benefits, Gaussian processes are highly nonlinear classifiers that can handle cases where linear classifiers are impossible to overcome. Since there is no guarantee of linear separability of novel cases, this property brings about an advantage compared to prototype classifiers.

SCL [3] adopts a methodology for assigning importance scores to each support sample by performing evaluations with the aid of all other support images. The importance scores are proportional to the evaluation scores of the mean intersection over union. The predictions from each prototype are then fused based on their corresponding importance scores. This approach is superior to a naive scheme that assigns equal importance to each prediction, thereby enabling SCL to leverage the support sets more effectively.

The Quality Measurement Module of the Meta-Class Memory Network [31] is responsible for computing activation maps for each support image. Unlike other methodologies, this module assigns distinct weights for the fusion of activation maps at each pixel location, enabling an assessment of the importance of each activation map at a specific location. To determine these weights, the module utilizes similarity matrices used during each activation propagation step. Specifically, a sigmoid function is applied to each element in the matrix, and the resulting values are summed along the row dimension. The resulting vector is then reshaped to the size of the feature map provided as input to the activation propagation module. These computations are repeated for each activation map, and the resulting maps are concatenated along the channel dimension. Finally, a softmax function is applied to the concatenated maps, and each number in the n^{th} channel represents the contribution of the n^{th} activation map at the corresponding position.

2.1.6 Correlation Reliability

Correlation map determines pixel-wise similarity between support and query images. Problems such as background clutter and occlusion render particular similarities noisy, resulting in erroneous comparisons and training based on misinterpreted correspondences. As semantic correspondence literature suggests, neighbor points around key points, whose correspondence in target is reliable, should also map into points in target near the key point match. This principle is called neighborhood consensus or semi-local constraint. Methods are invented to check validity of correspondences based on learnable or engineered criteria, so filtered correlation maps become interpretable. After elimination of deceptive correspondences as shown in Fig. 2.10, all similarities corresponding to each query pixel from support image are summed to obtain activation score that determines level of association of that query pixel with foreground of support. Since there are many aspects to prune correlation maps, there are more than one activation maps that are available to be used for segmentation of query in general.

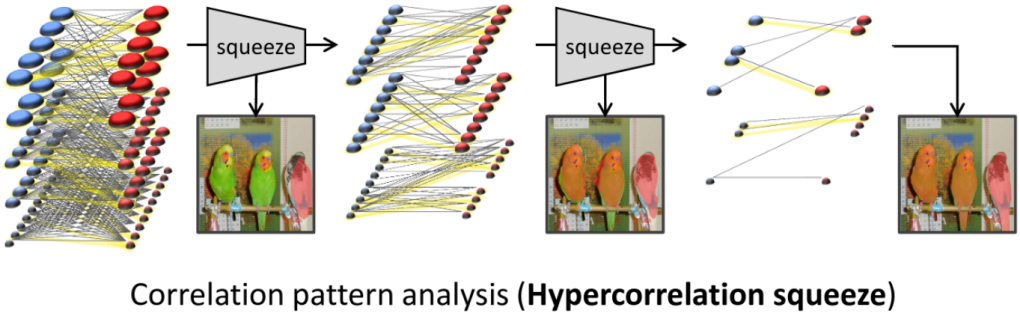

Figure 2.10: Correlation maps are filtered with 4-D convolution to eliminate misinterpreted correspondences [9].

The visual perception of human possesses an exceptional capacity to swiftly and accurately generalize the visual properties of novel objects, even with minimal supervision. This is attributed to its ability to discern consistent correspondences across various instances of a given class. Correspondence methods in computer vision recently have been based on utilizing feature maps from different layers of a network and 4D convolution, whose task is to analyze relational patterns. These relational

patterns correspond to correlation maps between the support and query features at different visual aspects that capture not only semantic but also geometric characteristics. The 4D convolution is inspired by the neighborhood consensus in classical vision, which takes a trustworthy match between two images and checks whether or not matches around the neighborhood occupy the region around the trustworthy match. Before the deep learning era, people designed constraints such as angle conservation, which demands that the angle between three points has to be the same as that of matched points with a given tolerance. The 4D convolution filter works to replace such designed constraints and filter unreliable matches. However, this type of convolution incurs excessive memory and computation time due to a high number of parameters. HSNet [9] solves this problem by weight-sparsification. They only focus on correlation that pivots the center of the support and query images so that they prune unnecessary parameters and it provides development in terms of not only computation but also performance. This simplification paves the way for the usage of a great deal of 4D convolution without inference time problem and overfitting. As a second contribution, they propose a novel architecture that uses these 4D convolutions. The 4D convolution processes hyper correlation that stacks correlation maps in different layers whose size is the same. They take these hyper correlations from early to late layers of the architecture to capture geometric and semantic characteristics.

The Volumetric Aggregation with Transformers (VAT) framework, as described in [24], utilizes the Swin Transformer to aggregate hyper correlation maps between the support and query sets. The high dimensionality of these hyper correlation maps presents a challenge, as treating each entry as a token requires significant computational resources. To address this issue, some approaches employ spatial pooling, though this sacrifices valuable information. An alternative solution is to split the correlation map into non-overlapping chunks and embed them with a linear mapping. However, this approach leads to an increased number of learnable parameters. To overcome these limitations, the VAT framework employs a layer consisting of 4D pooling, convolution, and group normalization, which contains fewer parameters while also providing equivariance properties to the model, which are lacking in the transformer architecture. This layer effectively reduces the token size, leading to computational advantages and more meaningful tokens. They called this module as

Volumetric Embedding Module. Afterward, the Swin Transformer operates by dividing the correlation map into four hypercubes, where self-attention is applied to each one. To further enhance context aggregation, the correlation map is then subjected to a cyclical shift, moving it leftward and upward by half of the hypercube's edge. This shifting mechanism enables the exchange of information between various windows, thereby allowing for the incorporation of large contextual information from different positions. This module is referred to as Volumetric Transformer. Additionally, the processed hyper correlation maps at the coarser level are upsampled and merged with those at the finer level, thereby forming a pyramidal structure. Subsequently, the query feature map from the initial layer is concatenated with the correlation map, which has been averaged along the support direction. This combination serves to eliminate noisy matching scores, and the resulting feature map is processed through the Swin Transformer-based decoder. The output is then upsampled by a factor of two, and the process is repeated until the original image size is reached. Finally, a classification layer is applied to the final feature map, and the binary cross entropy loss is employed to optimize the model, guided by the query ground truth map.

2.1.7 Thin Object Issue

Figure 2.11: Left-hand side figure reveals oversegmentation of the leaves of the plant, which is a weakness of the most few-shot segmentation methods. Right-hand side figure shows that [10] alleviates this problem effectively.

The most few-shot segmentation methods are not capable of segmenting thin objects as shown in left-hand side of Fig. 2.11. One reason for this is that the efficacy of

convolutional kernels with a square shape in detecting thin objects such as poles and sticks is limited. To overcome this limitation, it is advisable to learn both horizontal and vertical kernels in conjunction with the square kernel. However, the shortcoming is not solely attributable to the kernel's geometry. When constructing similarity matrices between query and support images to extract coarse masks, the conventional approach involves comparing the pixels of the two images. However, a pixel-wise comparison may not be ideal for detecting thin objects oriented vertically or horizontally. To address this issue, the Dynamic Prototype Network [10] proposes a regional matching method, which compares 1x5, 5x1, and 3x3 regions of the query and support images. As a result of this regional comparison, the similarity matrix has a channel number equivalent to the region size. To obtain three coarse target masks that account for vertically and horizontally oriented thin objects, as well as homogeneous ones, these similarity matrices are averaged along the channel dimension, and the maximum operation is subsequently applied along the row dimension. Finally, the average of these three target masks is calculated to derive an initial pseudo mask for the query. This approach captures thin objects more effectively than target masks that rely solely on pixel-by-pixel comparisons. The authors refer to the module responsible for this task as the support activation module. To filter out background pixels, the initial pseudo mask is multiplied with the query feature map. A refined pseudo target mask is then generated by combining the filtered query with a support prototype, which is subsequently processed with a 1x1 convolution and sigmoid. The resulting tensor is used to represent the refined pseudo target mask. This refined pseudo target mask is multiplied with the query image, and the resulting product is added to the query with a residual connection to enable residual learning. This operation, performed by the feature filtering module, reveals features from the class of interest by suppressing background ones. The authors further extracted foreground support pixel features and applied 1D pooling with sizes of S and S^2 to these features. Rather than using constant kernel weights, they proposed generating kernel weights by applying a convolutional network to the pooled feature map. The pooled feature map with a size of S is used to produce the horizontal and vertical kernels, while the pooled feature map with a size of S^2 gives rise to the squared kernel. As the kernel weights in this approach are dynamically adjusted based on the pooled features and possess an asymmetric structure in addition to the symmetric square kernel, they are better

equipped to represent subtle details compared to conventional convolutions whose weights remain constant post-training.

2.2 Connection between prompt learning and few-shot segmentation

The technique of prompting was initially utilized in natural language processing to distinguish between various tasks by incorporating a few clue words into the input sentences. In a broader sense, prompting methods can effectively condition the model to different domains or tasks, without making any modifications to model parameters. [36] employ the CLIP vision-language model [37]. CLIP is trained on a large dataset of image-text pairs sourced from the web in a self-supervised manner. The model can encode images and texts into a joint embedding space, which results in a strong correlation between the two modalities. To harness the potential of CLIP, the researchers incorporate the class name into a simple prompt template and then compute the cosine similarity of the output from both the vision and text encoders in CLIP. [38] utilize the CLIP model as a pre-trained backbone and then train a conditional segmentation layer on top. This layer is thin, and serves as the decoder for the model. The authors leverage the joint text-visual embedding space of CLIP to condition their model, allowing them to handle both textual prompts and images. As evidenced by aforementioned studies, prompting has been demonstrated to be highly effective in few-shot segmentation tasks, likely due to the close relationship between the two concepts.

CHAPTER 3

PROPOSED METHOD: BASE AND META LEARNER++

The proposed method presented in this dissertation enhances the BAM [1] model through two modifications. As first, we replace the ASPP [14] in BAM with Feature Enrichment Module [11] since ASPP [14] decoder within the BAM model lacks the capability to transfer information across various resolutions. Furthermore, the ASPP lacks distinct branches that concentrate on segmentation at separate resolutions. The ASPP is developed for a supervised case and is susceptible to a low-data regime. As a result, the utilization of the ASPP can lead to the persistence of spatial inconsistency issues. Proposed method mitigate spatial inconsistency problem by employing FEM as its decoder. Section 3.1.1 is dedicated to a thorough examination of the FEM and all of its subcomponents.

BAM observed that the meta-learned model confuses pixels from base classes with novel ones during meta-testing, where the base classes correspond to the set of classes used in meta-training. This is because the base classes are introduced as foreground in meta-training, resulting in overfitting. To address this issue, they train a model that specializes in base classes via supervised learning, which they call the base learner. The small head, which branches from the backbone, adopts episodic learning and is called the meta learner. During meta-training, the meta-learner interacts with the base learner to prevent confusion between base classes and novel ones. To achieve this, they sum the probabilities of all classes except the selected class during an episode to obtain the base map, which ideally shows the background region relative to the class in the episode. Prediction of the meta-learner is prone to errors, as it has a higher tendency to mistake base classes for novel ones. To compensate for this, they concatenate the base map with the background prediction of the meta-learner, and pass

the resulting tensor through a 1 by 1 convolution. This is called the ensemble mechanism, which compensates for the weaknesses of the meta-learner. As the base maps are a decent estimate for the background due to supervised training, this approach is effective in resolving the confusion issue. As a second modification, the proposed method implements the ensemble mechanism at each segmentation prediction made at the branches of the decoder in addition to the one made at the final prediction. Therefore, the confusion problem is eliminated from the auxiliary predictions. Section 3.1.2 provides a comprehensive examination of the base learner, meta learner, and ensemble mechanisms, while Section 3.2 presents a detailed explanation of the proposed method.

3.1 Background Methods

3.1.1 Revisit Feature Enrichment Module

Multi-scale modules in supervised semantic segmentation generally do not provide mechanism to form independent interaction between masked global average pooled support feature map, called as support prototype $\mathbf{v}_s$, and average pooled query feature maps at different scales.

In these modules, information at different resolutions is generally processed at a single branch, preventing consideration of each resolution separately. For example, conventional multi-scale architecture, PSPNet [16], applies single filtering to the combination of query feature maps at different resolutions and support prototype.

3.1.1.1 Inter-Source Enrichment Module

Different from these approaches, inter-source enrichment module of FEM in Fig. 3.1 separately applies the filtering to the query feature map at each different scale, which is combined with support prototype and prior mask.

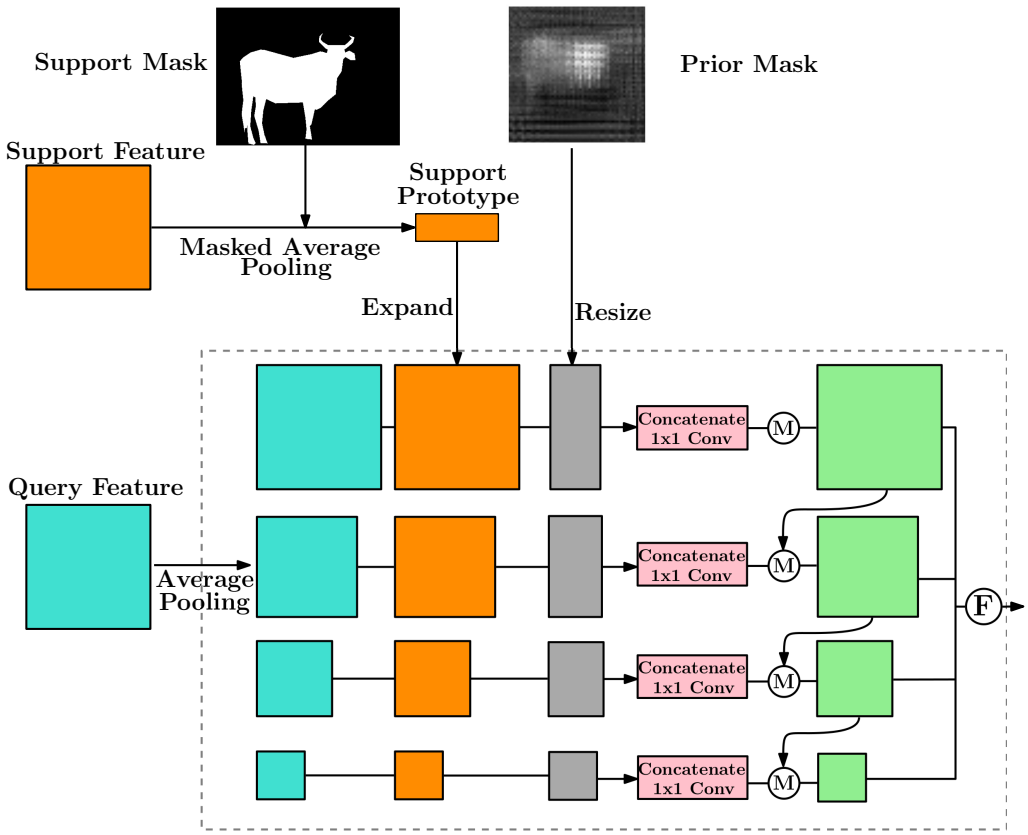

Figure 3.1: This figure presents an overview of the feature enrichment module proposed in [1]. The blue, orange, and gray rectangles correspond to the query feature map at different resolutions, the expanded support prototype, and the resized prior mask, respectively. These three components are concatenated and passed through a 1x1 convolutional layer denoted by the pink rectangle. The inter-source enrichment module is responsible for the aforementioned operations up to this point. The outputs of these blocks are then provided to block M, which is named the inter-scale merging module.

3.1.1.2 Prior Mask

Prior mask describes likelihood of query pixel being related with at least one pixel in foreground of support [11]. To create a prior map in a class-agnostic manner, features after block-4 of ResNet-50 are used, namely $\mathbf{f}_b^q$ and $\mathbf{f}_b^s$ corresponding to query and support respectively. The prior map is created by comparing each pixel in the query image with the pixels in the foreground of the support image using cosine similarity. The maximum similarity between a query pixel and all of the pixels in

the foreground of the support image is assigned as the value for that pixel in the prior map. Mathematically, high-level query and support features are reshaped from $\mathrm{R}^{H\times W\times C}$ to $\mathrm{R}^{HW\times C}$ at first, where H, W, and C correspond to the number of height, width, and channel respectively. After that, row-wise norms for high-level query and support pixel features are computed respectively as in Eq. 3.1 and Eq. 3.2, where $^\circ$ corresponds to Hadamard root while $diag$ outputs diagonal elements of a matrix as a column vector.

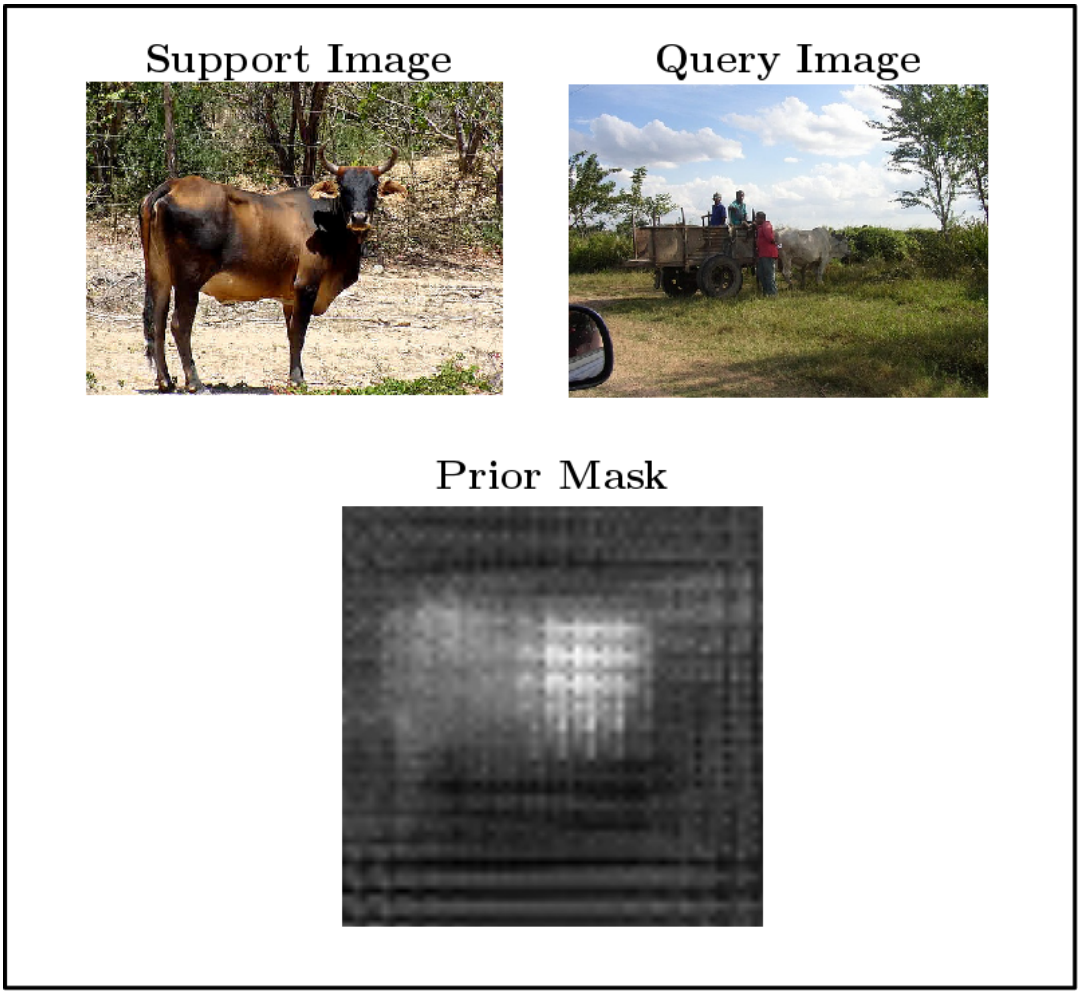

Figure 3.2: This figure highlights that the regions related to the target class have higher activation compared to the background.

$$\|\mathbf{f}_b^q\| = (diag(\mathbf{f}_b^q \times \mathbf{f}_b^{q\intercal}))^{\circ 1/2} \in R^{HW\times 1} \tag{3.1}$$

$$\|\mathbf{f}_b^s\| = (diag(\mathbf{f}_b^s \times \mathbf{f}_b^{s\intercal}))^{\circ 1/2} \in R^{HW\times 1} \tag{3.2}$$

Prior map before normalization is calculated by max pooling the cosine similarity matrix between the high-level query and support pixels along row-wise direction as shown in Eq. 3.3, where $\oslash$ is Hadamard division. Pool operation in Eq. 3.3 replaces similarity values computed for background pixel with negative infinity to neglect background region.

$$\mathbf{C}_q^{un} = \text{pool}((\mathbf{f}_b^q \times \mathbf{f}_b^{s\intercal}) \oslash (\|\mathbf{f}_b^q\| \times \|\mathbf{f}_b^s\|^\intercal)) \ \in R^{H \times W \times 1} \tag{3.3}$$

Before the final version of the prior map is obtained, min-max normalization is applied to it as shown in Eq. 3.4. During the min-max normalization of the prior map, the minimum value within the map is subtracted from all values, and the result is divided by the difference between the maximum and minimum values within the map. This standardizes all values within the prior map to be between 0 and 1. A small ϵ value is included to address the problem of division by zero when the denominator is zero.

$$\mathbf{C}_q = \frac{\mathbf{C}_q^{un} - min\left(\mathbf{C}_q^{un}\right)}{max\left(\mathbf{C}_q^{un}\right) - min\left(\mathbf{C}_q^{un}\right) + \epsilon} \tag{3.4}$$

The prior mask roughly identifies regions that are likely to belong to the foreground region and provides a hint about where subsequent processing should focus its attention. In Fig. 3.2, the segmentation of a cow as a novel class is shown. The region of the image that is next to the people in the query image has a higher response compared to other areas of the image due to the presence of the cow.

3.1.1.3 Inter-Source Enrichment Module

Now that the concept of prior maps as a form of prior knowledge has been understood, we can move on to the details of the inter-source enrichment module. For average pooling, there are N different dimensions shown as $S = \left[S^1, S^2, S^3, \ldots, S^N\right]$, where the dimensions decrease in size as the index increases. Adaptive average pooling is applied to the query features in a way that corresponds to all dimensions within the set S by employing $avg\_pool\_S^i$ function in Eq. 3.5. The support prototypes are

enlarged to match the dimensions in the set S by using $expand\_S^i$ function and then concatenate to the back of the query feature maps. Then, the prior maps are resized appropriately by using $resize\_S^i$ function and merged along the channel dimension. The feature map that is generated is transformed using a 1x1 convolution to produce $f_b^{q_i} \in R^{S^i \times S^i \times C}$ as shown in Eq. 3.5.

$$f_b^{q_i} = \mathcal{F}_{1\times1}\left(avg\_pool\_S^i\left(\mathbf{f}_b^q\right) \oplus expand\_S^i\left(\mathbf{v}_s\right) \oplus resize\_S^i\left(\mathbf{C}_q\right)\right) \tag{3.5}$$

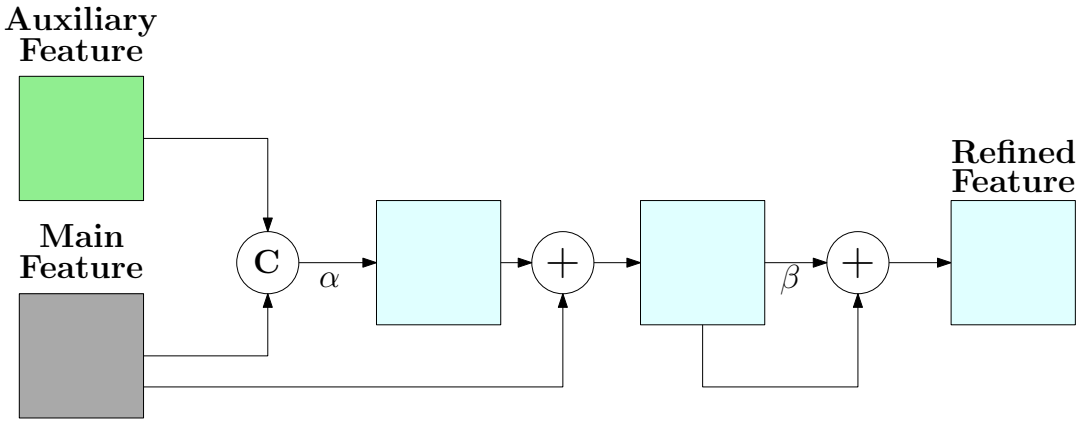

Figure 3.3: Overview of Inter-Scale Merging Module [11]. In this illustration, C, α, and β correspond to the concatenation operation, a 1-by-1 convolution, and a 3-by-3 set of convolutions, respectively.

Furthermore, inter-scale merging module of FEM fulfills the information transfer between two consecutive resolutions in top-down path, where top-down path consists of outputs of inter-source enrichment module ordered from high resolution to low resolution. During information transfer, preservation of hierarchical structure allows gradual accumulation of information from higher resolution to lower resolution. In this module as shown in Fig. 3.3, each resolution has direct connection only to its neighbour in the top-down direction. Therefore, there is no connection between any resolution pairs other than the consecutive ones. Hence, the module has a chance to decide on the scale at which the obtained information is sufficient to make a prediction, and the following scales would bring redundancy.

$$
\begin{aligned}
X_q^{s_1} &= \mathcal{M}\left(f_b^{q_1}(main)\right) \\
X_q^{s_i} &= \mathcal{M}\left(f_b^{q_i}(main), X_q^{s_{i-1}}(auxilary)\right), i \in \{x | x \in \mathbb{Z}, 2 \leq x \leq N\}
\end{aligned}
\tag{3.6}
$$

At the highest resolution, only the main feature map is processed because there is no auxiliary feature map available. The interscale merging module then combines the feature maps from successive resolutions and passes them through a 1x1 α convolution (as shown in Fig. 3.3), enabling the flow of more detailed information from higher to lower resolutions. The main feature maps are also merged using skip connections to facilitate learning through a residual structure. The final version of the feature map is created by applying two 3x3 convolutions with residual connections, and this process continues until the lowest resolution, S^n, is reached. The operations performed with α and β convolutions are represented by the symbol M in Eq. 3.6.

3.1.1.4 Information Concentration

Enriched feature maps with dimensions smaller than S^1 in Eq. 3.5 are upsampled to the size of S^1 through interpolation. Then, all the sequentially concatenated features are processed with a 1x1 convolution to obtain X_q^{fused}. This feature map contains the combined information from all resolutions. All these operations are carried out using the information concentration module within the FEM as described in Eq. 3.7.

$$
X_q^{fused} = \mathcal{F}_{1\times1}\left(X_q^{s_1} \oplus X_q^{s_2} \cdots \oplus X_q^{s_N}\right) \tag{3.7}
$$

The feature maps from different resolutions, which are input to the information concentration module, are passed through convolutions with the architecture shown in Fig. 3.4(b), resulting in predictions for each resolution. Loss functions calculated at those resolutions can help create a hierarchical structure for feature maps. Therefore, these loss functions serve as a guide and can improve the accuracy of the predictions made. The X_q^{fused} feature map is processed through two 3x3 convolutions with a residual connection, as depicted in Fig. 3.4(a), prior to being passed through the classification block composed of a 1x1 convolution and softmax, as illustrated in Fig.

3.4(b). This differs from the processing of the X_q^{si} feature maps, which are passed through the classification block without undergoing the 3x3 convolutions with the residual connection.

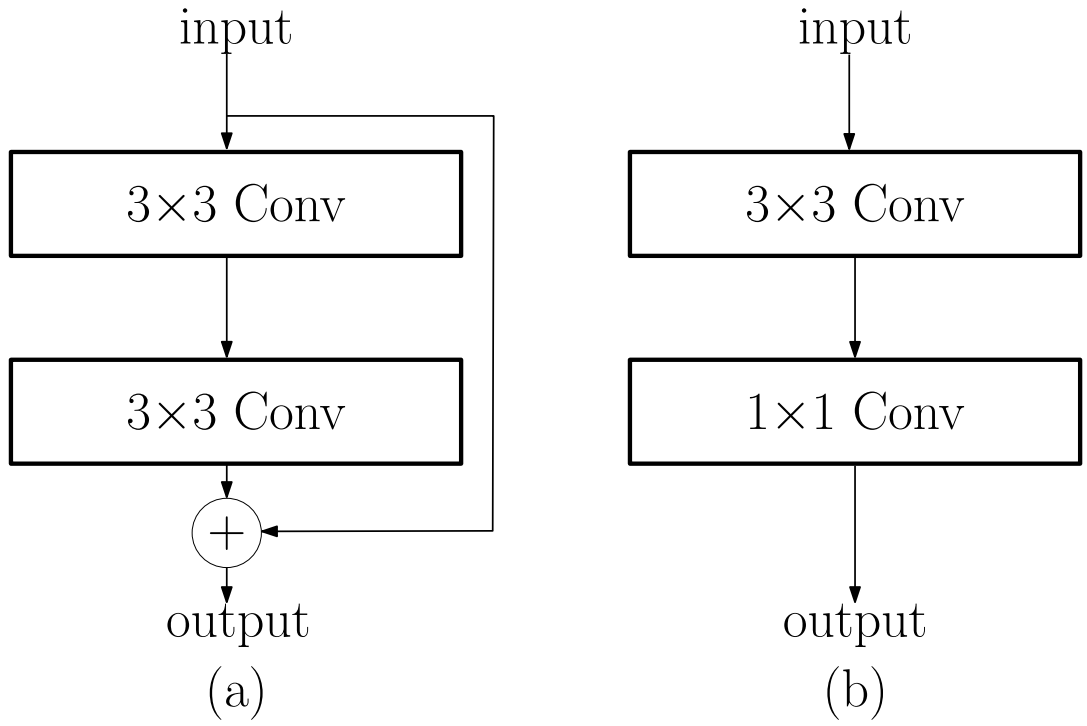

Figure 3.4: The left-hand side of the figure represents a convolution layer that is designed to enhance the feature representation. On the other hand, the right-hand side of the figure shows a classification block, where the final layer reduces the channel dimension to 2.

3.1.1.5 K-Shot Configuration

There are several differences when there are multiple support examples. Firstly, the number of support prototypes is equal to the number of shots because there are multiple support examples. Secondly, there is a prior map for each example in the support set in a similar manner. If we consider the circumstances specifically within the 5-shot context, the effective support prototype is found by taking the average of the 5 support prototypes that are generated. Then, in a similar manner to the 1-shot case, this prototype is concatenated with the query feature map and used in the Inter-Source Enrichment Module. A similar procedure is followed to acquire the effective prior

map. 5 support instances are used to obtain 5 prior maps. The average of the 5 prior maps is calculated to obtain the effective prior map. Similar to the 1-shot situation, the effective prior map is concatenated to the back of the query and support prototypes. In subsequent sections, the module described above will be referred to as the FEM function, which will provide outputs $X_q^{fused}, X_q^{s_1}, \ldots, X_q^{s_N}$.

3.1.2 Revisit Base and Meta Learner

3.1.2.1 Episodic Learning

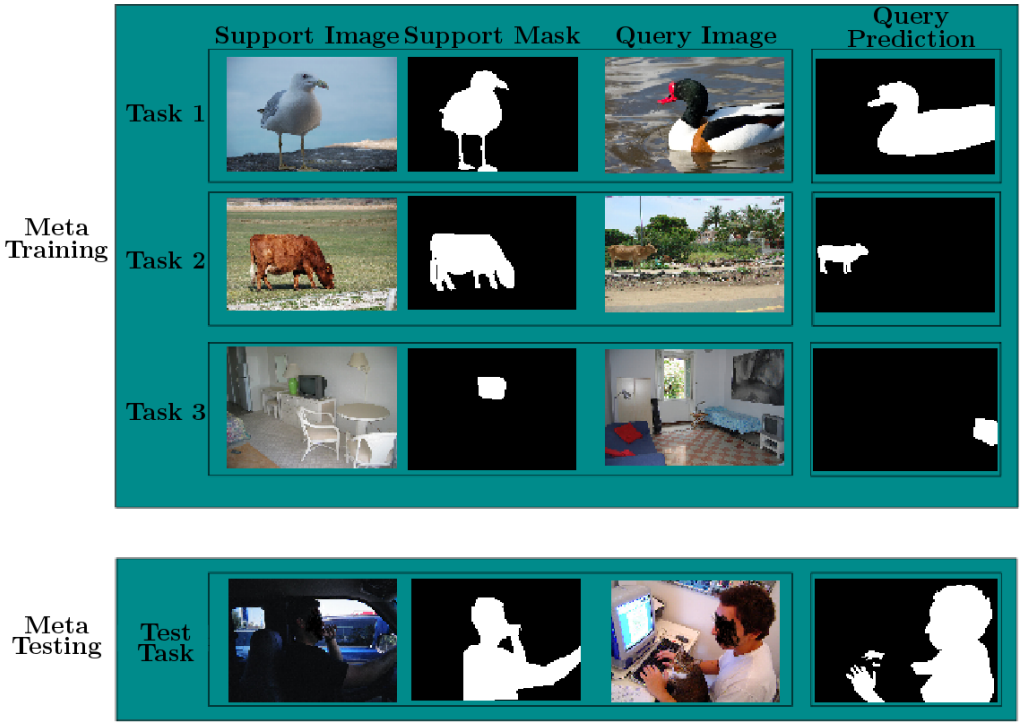

Figure 3.5: Episodic Learning comprises of two stages: meta-training and meta-testing. During meta-testing, the classes presented are completely distinct from the ones encountered during meta-training. The term 'task' can be used interchangeably with 'episode'.

Typical few-shot segmentation approaches use meta-learning approach such that the knowledge gained from training the model on the base classes is utilized to predict the mask of the query image belonging to a novel class given a support image belonging to the same novel class. This process is called as meta-learning since learning tasks

are sampled from the base classes during training in order to simulate the few-shot settings in testing so that the training and testing conditions are matched.

For example, in Fig. 3.5, images and masks belonging to support and query for the bird, cow, and monitor classes found in the base dataset are sampled. The positions of each sampled class in the foreground are treated as if they had not been seen before and labeled with 1, while the remaining parts are labeled with 0. Each row in Fig. 3.5 is referred to as an episode in meta-learning, and serves as the equivalent of a mini-batch in supervised learning. A forward pass is performed over a certain number of episodes, followed by the application of backpropagation. Thus, it is expected that segmentation will be performed independently of the class when the positions of previously unseen classes in the incoming test are labeled with 1 in the support images.

3.1.2.2 Bias and Its Solution

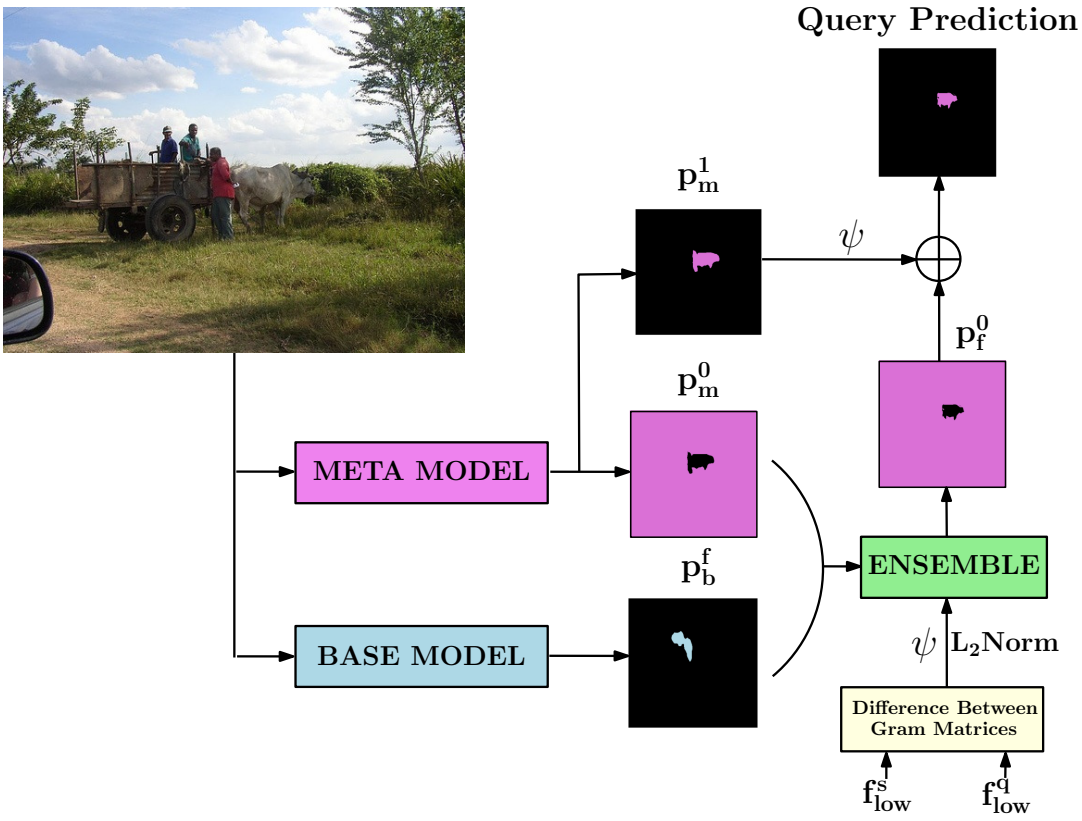

Figure 3.6: Overview of Base and Meta Learner.

As [1] states, training on base classes introduces a bias towards them during testing, preventing the model from working on the novel classes properly. To tackle this bias, BAM is introduced, where a base learner, apart from the meta learner, explicitly works on the known classes. When the information related to known classes is used during inference, the recognition of novel classes is enhanced. For example, in Fig. 3.6, while attempting to segment the target cow, the image is constantly exposed to humans from the base classes during training, causing the parts of the image associated with humans to be wrongly classified as a novel. The meta learner is warned of confusion in the regions incorrectly predicted as cows by the model trained supervised on base classes. As illustrated in Fig. 3.6, predictions from the base effectively eliminate the regions that were wrongly identified as novel. To resolve this confusion, an ensemble learning method that incorporates both meta and base predictions as inputs is implemented.

Training BAM consists of two stages, namely base-training and meta-training. Both learners share the same backbone as feature encoder. To leverage the representations at different levels of abstraction, features are obtained from different layers of the encoder.

3.1.2.3 Base Learner

Base learner is trained in a supervised manner so that the ability to make confident predictions regarding base classes is gained. Query features after block-4 of ResNet-50, $\mathbf{f}_b^q$, are processed by base learner and decoded by Pyramid Scene Parsing Network (PSPNet) [16], which is composed of Pyramid Pooling Module (PPM) and classifier. The operation $\mathcal{D}_b$ upsamples a feature map mixed from pre-defined multi-scales to the original height (H) and width (W) of the query image, then applies a classifier to each pixel. Logits for N_b number of the base classes and background are obtained for each pixel through this process. The logits are transformed into probabilities with the softmax operation.

$$\mathbf{p}_b = \text{softmax}\left(\mathcal{D}_b\left(\mathbf{f}_b^q\right)\right) \in \mathbb{R}^{(N_b+1)\times H\times W} \tag{3.8}$$

In contrast to the typical approach of episodic learning, which is commonly used in few-shot learning, traditional supervised learning strategy is adopted for classifying individual pixels into either base classes or the background category. The amount of discrepancy between the ground truth $\mathcal{M}_q^b$ for the image and the estimated $\mathbf{p}_b$ is determined by the cross-entropy loss function, and the base learner is trained accordingly as shown in Eq. 3.9.

$$\mathcal{L}_{base} = \frac{1}{n_{bs}} \sum_{i=1}^{n_{bs}} \text{CE}\left(p_{b,i}, \mathcal{M}_{q,i}^b\right) \tag{3.9}$$

In Eq. 3.10, probabilities for each base class are aggregated with summation to obtain $\mathbf{p}_b^f$, where $\mathbf{p}_b^i$ corresponds to probability of pixel belonging to i$^{\text{th}}$ class. This step is crucial since the base classes are the background classes for the query image while the novel class is the foreground, which is to be predicted by the meta learner.

$$\mathbf{p}_b^f = \sum_{i=1}^{N_b} \mathbf{p}_b^i \tag{3.10}$$

3.1.2.4 Meta Learner

In meta-training stage, the parameters of the base learner are fixed. The features of support and query images are extracted by the shared encoder, and the features obtained after block-2 and block-3 of ResNet-50 [39] are concatenated and transformed with 1×1 convolution layer, which is denoted by $\mathbf{f}_m^s$ and $\mathbf{f}_m^q$ respectively. The support mask, $\mathbf{m}^s$, is used together with $\mathbf{f}_m^s$ in order to obtain the support prototype, $\mathbf{v}_s$. Then, $\mathbf{v}_s$ are expanded and concatenated with combination of query and prior map by $\mathcal{EC}$, initial letters of expand and concatenation operations, in Eq. 3.11. Resulting feature map is inputted to meta-decoder denoted as $\mathcal{D}_m$ in Eq. 3.11. $\mathcal{D}_m$ performs task of classifying each pixel into class selected in current episode and background.

$$\mathbf{p}_m = \text{softmax}\left(\mathcal{D}_m\left(\mathcal{EC}(\mathbf{v}_s, \mathbf{f}_m^q)\right)\right) \tag{3.11}$$

$$\mathcal{L}_{meta} = \frac{1}{n_e} \sum_{i=1}^{n_e} BCE\left(\mathbf{p}_{m,i}, \mathcal{M}_{q,i}\right) \tag{3.12}$$

The binary cross entropy loss function is utilized to gauge the dissimilarity between the ground truth $\mathcal{M}_q$ and the predicted output $\mathbf{p}_m$ over a set of n_e episodes as shown in Eq. 3.12. The average of the computed errors is then backpropagated to optimize the weights of the meta-learner.

3.1.2.5 K-Shot Configuration

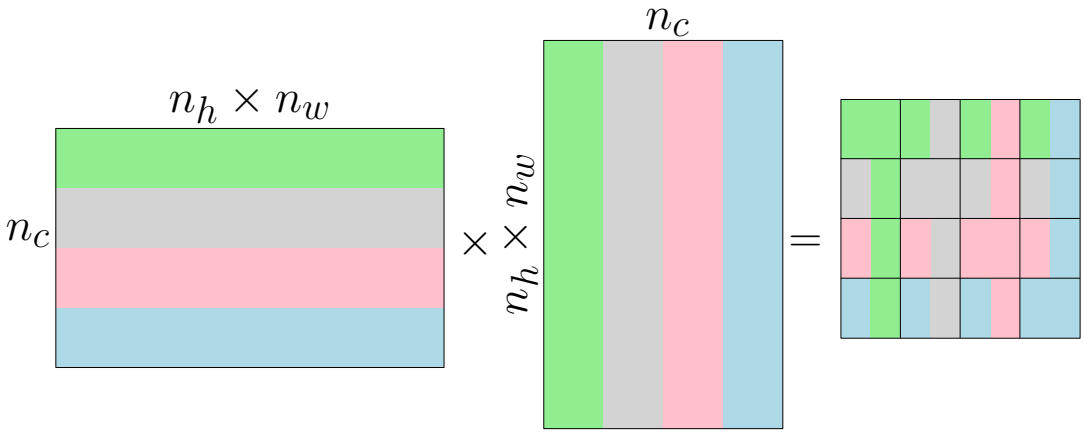

Figure 3.7: Visualization of Computation of Gram Matrix. In this figure, n_c, n_h, and n_w corresponds to C_{low}, H_{low}, and W_{low} respectively.

The adjustment factor, ψ, quantifies the confidence in the prediction made by the meta-learner, and also determines the relative significance of each sample from the support set in the final prediction. In fact, the coefficient, ψ, indicates the degree of stylistic difference between a query image and a support image. When there is a mismatch between the styles of the query and support image, it implicitly leads to a decrease in the confidence in the meta-learner or the support image.

A gram matrix, a mathematical representation of the inner product of a set of vectors, serves as the foundation for the derivation of adjustment factors. The gram matrix in this context holds the correlations between channels in the feature maps in the initial layers of the model for the support and query images, namely $\mathbf{f}^s_{low}$ and $\mathbf{f}^q_{low}$. To compute the Gram matrix, the feature map is first unfolded in the raster direction, and

then the dot product is taken with its transpose as shown in Eq. 3.13. This serves to demonstrate the extent to which certain attributes are present together within the image [40] as shown in Fig. 3.7.

$$\begin{aligned}
\mathbf{V}_s &= \text{unfold}\left(\mathbf{f}^s_{low}\right) \in R^{C_{low} \times H_{low} W_{low}} \\
\mathbf{G}_s &= \mathbf{V}_s \mathbf{V}_s^T \in R^{C_{low} \times C_{low}} \\
\mathbf{V}_q &= \text{unfold}\left(\mathbf{f}^q_{low}\right) \in R^{C_{low} \times H_{low} W_{low}} \\
\mathbf{G}_q &= \mathbf{V}_q \mathbf{V}_q^T \in R^{C_{low} \times C_{low}}
\end{aligned} \tag{3.13}$$

The adjustment factor is determined by using the Frobenius norm of the differences between the gram matrices of support and query image as shown in Eq. 3.14. A graphical representation of the computations is provided in Fig. 3.8 to enhance comprehensibility.

$$\psi = \left\| \mathbf{G}_s - \mathbf{G}_q \right\|_F \tag{3.14}$$

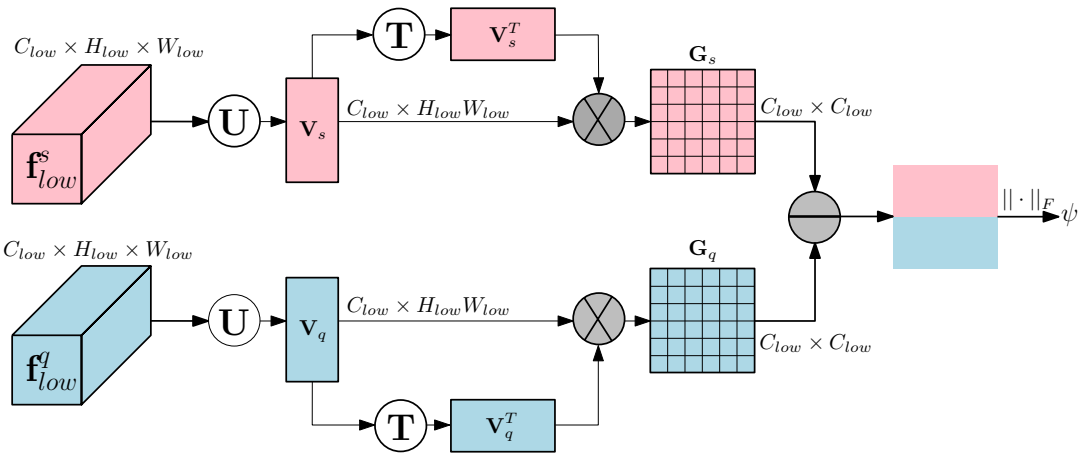

Figure 3.8: Visualization of the Computation of the Adjustment Factor is presented in this illustration. U and T represent the unfold and transpose operations, respectively. The cross inside the circle denotes matrix multiplication, while the minus sign inside the circle denotes element-wise subtraction.

If there are multiple support samples, it would not be correct for each support sample to contribute equally to the prediction of the query. Weighted averaging of support prototypes, rather than uniform averaging, is considered a more appropriate method. This is because, scene-wise, support samples that are distant from the query may contain different concepts and can make the final prediction noisy. Therefore, it is necessary to create a weighting system. The ψ coefficients are included in the creation of this weighting system. The weights of the support samples with smaller ψ coefficients are determined to be higher, achieving the desired goal. The concatenation of all coefficient vectors, denoted as ψ_{all}, is performed to form a single vector. Subsequently, fusion weights are learned by passing the resulting vector through a non-linear layer as shown in Eq. 3.15. This layer consists of two subsequent blocks, including a fully connected layer and a rectified linear unit (ReLU) activation function. These blocks function to map adjustment factors to fusion weights, where smaller adjustment factors result in larger fusion weights. The matrices $\mathbf{w}_1 \in \mathbb{R}^{K \times \frac{K}{f}}$ and $\mathbf{w}_2 \in \mathbb{R}^{\frac{K}{f} \times K}$ represent the weights of a fully connected layer in a neural network, where K denotes the number of shots and f denotes the reduction ratio.

$$\eta = \text{softmax}\left(\mathbf{w}_2^{\intercal}\text{ReLU}\left(\mathbf{w}_1^{\intercal}\psi_{\text{all}}\right)\right) \tag{3.15}$$

The effective support prototype is determined by computing the weighted mean of all support prototypes, utilizing the weighting factor η as outlined in Eq. 3.16. In Eq. 3.17, the effective adjustment factor, which assesses the reliability of the support set for meta-prediction, is derived by means of a weighted average of individual adjustment factors, where fusion weights are employed as the weights. This process is akin to the fusion of support prototypes into a single, effective representation.

$$\mathbf{v}_s = \frac{\sum_{i=1}^{K} \eta_i \cdot \mathbf{v}_{s,i}}{\sum_{i=1}^{K} \eta_i} \tag{3.16}$$

$$\psi = \frac{\sum_{i=1}^{K} \psi_{all,i} \cdot \eta_i}{\sum_{i=1}^{K} \eta_i} \tag{3.17}$$

3.1.2.6 Ensemble Learner

At the end of the meta decoder, output background and foreground probability maps, $\mathbf{p}_m^0$ and $\mathbf{p}_m^1$, are obtained. The probabilities, $\mathbf{p}_m^0$ and $\mathbf{p}_m^1$, are combined with the credibility information provided by ψ through concatenation. A 1x1 convolutional layer is then applied to this combination, allowing for the adjustment of the probabilities based on the ψ. For example, a higher value of ψ can result in the suppression of the contribution of the meta-learner in the ensemble, while a lower value facilitates a more equitable treatment of both the base and meta-learners. In Eq. 3.18, regulated $\mathbf{p}_m^0$ is ensembled with $\mathbf{p}_b^f$ in order to force the pixels belonging to non-novel regions for the query image to be closer to the base map. Ensembling operation applies 1 by 1 convolutional layer to concatenation of regulated $\mathbf{p}_m^0$ and $\mathbf{p}_b^f$. As long as base map is accurately predicted, misclassified base regions can be rectified through this ensemble learning mechanism. In other words, the utilization of ensemble methods serves to mitigate the likelihood of mistaking base regions for the novel class selected during the episode through the guidance provided by $\mathbf{p}_b^f$. Therefore, the regions in the final background map $\mathbf{p}_f^0$ that may have been previously confused are refined. Resultant ensembled information is concatenated with regulated $\mathbf{p}_m^1$ in order to produce final prediction $\mathbf{p}_f$ as shown in Eq. 3.19.

$$\mathbf{p}_f^0 = Ens_\phi \left(\mathbf{p}_b^f \oplus Ens_\psi \left(\mathbf{p}_m^0 \oplus \psi \right) \right) \tag{3.18}$$

$$\mathbf{p}_f = Ens_\psi \left(\mathbf{p}_m^1 \oplus \psi \right) \oplus \mathbf{p}_f^0 \tag{3.19}$$

The parameters of the meta-learner are updated by summing the binary cross-entropy losses of the final prediction, $\mathbf{p}_f$, and the meta prediction, $\mathbf{p}_m$, as shown in Eq. 3.20.

$$\mathcal{L}_{final} = \frac{1}{n_e} \sum_{i=1}^{n_e} BCE\left(\mathbf{p}_{f,i}, \mathcal{M}_{q,i}\right)$$
$$\mathcal{L}_{total} = \mathcal{L}_{final} + \lambda \mathcal{L}_{meta} \tag{3.20}$$

In the context of the Eq. 3.20, the hyperparameter λ is held constant at a value of 1 throughout the entirety of the experiments. Furthermore, the objective function for training the meta-learner, represented by $\mathcal{L}_{meta}$, is the same as the one previously described in Eq. 3.12.

3.2 Proposed Method

As proposed in the CANet paper [32], we employ middle-level features by applying 1x1 convolution to the concatenation of feature maps obtained from both block-2 and block-3 layers of the model. The middle-level features extracted from the support and query images are denoted by Eq. 3.21 and Eq. 3.22 respectively, where the symbol Enc represents the middle-level feature extraction process.

$$\mathbf{f}_m^s = Enc(\mathcal{I}_s) \ \in R^{H \times W \times C} \tag{3.21}$$

$$\mathbf{f}_m^q = Enc(\mathcal{I}_q) \ \in R^{H \times W \times C} \tag{3.22}$$

Masked global average pooling is applied to $\mathbf{f}_m^s$ to extract support prototype, $\mathbf{v}_s$, in Eq. 3.23, where $\mathcal{R}$ downsamples $\mathcal{M}_s$ to the size of $\mathbf{f}_m^s$. The R function also duplicates the downsampled $\mathcal{M}_s$ along the channel dimension a number of times equal to the number of channels present in $\mathbf{f}_s$. Then, the masked average pooling function utilizes an extended mask, which is multiplied with the support feature map, to neutralize features associated with background pixels prior to averaging effectively. This operation allows for the computation of a robust representation of the foreground elements within the feature map, discarding any contributions from background. Through this process, the masked average pooling function selectively aggregates features from

the foreground regions of the feature map. Subsequently, the resulting feature map is divided by the number of foreground pixels to derive the support prototype as shown in Eq. 3.23.

$$\mathbf{v}_s = \text{masked\_avg\_pool}(\mathbf{f}_m^s, \mathcal{R}\left(\mathcal{M}_s\right)) \ \in R^{1\times1\times C}$$

$$z = \text{masked\_avg\_pool}\left(x, y\right) = \frac{\sum_{i=1}^{W}\sum_{j=1}^{H}\mathbf{x}_{i,j} \odot \mathbf{y}_{i,j}}{\frac{1}{C}\sum_{i=1}^{W}\sum_{j=1}^{H}\mathbf{y}_{i,j}} \tag{3.23}$$

The feature enrichment module (FEM) is a computational module that receives as input a support prototype vector $\mathbf{v}_s$, a prior map $\mathbf{C}_q$, and a query feature map $\mathbf{f}_m^q$. The output of the FEM is a set of N+1 enriched query feature maps, which includes N auxiliary feature maps at different scales, as well as a final fused feature map. This is mathematically represented in Eq. 3.24. FEM is equivalent to function mentioned in subsection 3.1.1.5.

$$\mathbf{X}_q^{s_1}, \mathbf{X}_q^{s_2}, ..., \mathbf{X}_q^{s_N}, \mathbf{X}_q^{fused} = FEM(\mathbf{C}_q, \mathbf{f}_m^q, \mathbf{v}_s) \tag{3.24}$$

$$\mathbf{C}^{AUX} = \{\mathbf{C}^{aux,1}, \mathbf{C}^{aux,2}, ..., \mathbf{C}^{aux,N}, \mathbf{C}^{aux,fused}\} \tag{3.25}$$

In Eq. 3.25, $\mathbf{C}^{AUX}$ denotes a set of classifiers. The first N classifiers in this set are auxiliary classifiers, which generate predictions for multi-scale features. The final classifier in this set is responsible for making predictions based on the fused feature. By utilizing these classifiers, we are able to obtain background and foreground logit values for the enhanced query feature maps at each scale and the fused feature map, respectively, as defined in Equations 3.26 and 3.28. The symbol $\oplus$ in these equations represents the concatenation operation.

$$\mathbf{p}^0_{m,s_i}, \mathbf{p}^1_{m,s_i} = \mathbf{C}^{aux,i}(\mathbf{X}^{s_i}_q) \tag{3.26}$$

$$\mathbf{p}_{m,s_i} = \mathbf{p}^0_{m,s_i} \oplus \mathbf{p}^1_{m,s_i} \tag{3.27}$$

$$\mathbf{p}^0_{m,fused}, \mathbf{p}^1_{m,fused} = \mathbf{C}^{aux,fused}(\mathbf{X}^{fused}_q) \tag{3.28}$$

$$\mathbf{p}_{m,fused} = \mathbf{p}^0_{m,fused} \oplus \mathbf{p}^1_{m,fused} \tag{3.29}$$

$BaseLearner$ in Eq. 3.30 takes $\mathbf{f}^m_q$ as input and outputs summation of predicted probabilities for all classes except background. The function of the BaseLearner is equivalent to the sequential application of the operations defined in Eq. 3.8 and Eq. 3.10.

$$\mathbf{p}^f_b = BaseLearner(\mathbf{f}^q_m) \tag{3.30}$$

3.2.1 Range of Ensembles

The current method implements ensemble models, as represented by Equations 3.31, 3.32, and 3.33, similar to those used in BAM [1]. However, it introduces a new aspect by utilizing different ensemble models for each auxiliary prediction at various scales, as depicted in Equations 3.34, 3.35, and 3.36. This allows the meta-model to consider non-novel regions at each scale, which is illustrated in Fig. 3.9 where the pink rectangular boxes enclosed by dashed lines represent the ensemble models.

$$\mathbf{p}^0_{f,fused} = Ens_\phi(\mathbf{p}^f_b, Ens_\psi(\mathbf{p}^0_{m,fused}, \psi)) \tag{3.31}$$

$$\mathbf{p}^1_{f,fused} = Ens_\psi(\mathbf{p}^1_{m,fused}, \psi) \tag{3.32}$$

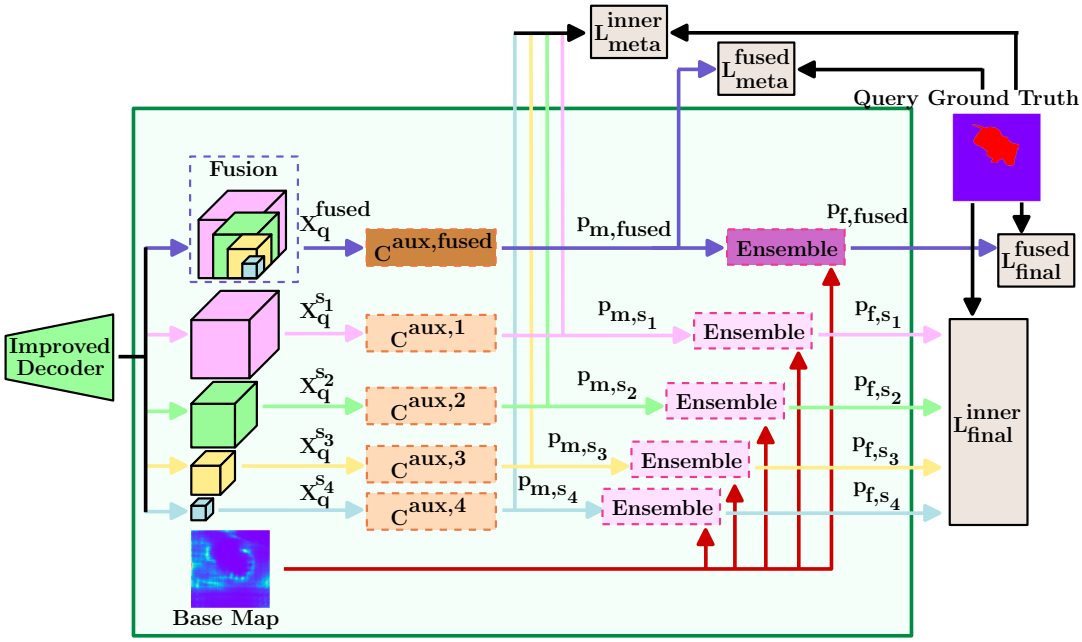

Figure 3.9: Detailed architecture of the multi-scale ensemble module. Features at multi-scale and the fusion of them are obtained at the end of the improved decoder as $\mathbf{X}_q^{s_i}$ and $\mathbf{X}_q^{fused}$ respectively, which are used by the corresponding auxiliary classifiers. The resultant enriched query feature maps are ensembled with the base map to obtain query predictions at multi-scale, which are denoted by $\mathbf{p}_{f,s_i}$ and $\mathbf{p}_{f,fused}$ respectively. Inner losses are computed from probability maps at intermediate scales $\left(\mathbf{p}_{m,s_i}\right)$ and predictions at intermediate scales $\left(\mathbf{p}_{f,s_i}\right)$ while fused losses are computed from fused probability maps $\left(\mathbf{p}_{m,fused}\right)$ and fused predictions $\left(\mathbf{p}_{f,fused}\right)$. (Best viewed in color)

$$\mathbf{p}_{f,fused} = \mathbf{p}^0_{f,fused} \oplus \mathbf{p}^1_{f,fused} \tag{3.33}$$

$$\mathbf{p}^0_{f,s_i} = Ens_{\phi,s_i}(\mathbf{p}^f_b, Ens_{\psi}(\mathbf{p}^0_{m,s_i}, \psi)) \tag{3.34}$$

$$\mathbf{p}^1_{f,s_i} = Ens_{\psi}(\mathbf{p}^1_{m,s_i}, \psi) \tag{3.35}$$

$$\mathbf{p}_{f,s_i} = \mathbf{p}^0_{f,s_i} \oplus \mathbf{p}^1_{f,s_i} \tag{3.36}$$

The proposed method calculates cross-entropy loss for both the auxiliary predictions and the fused prediction, both before and after the ensemble process. Equations 3.37 and 3.38 calculate the cross-entropy loss for the auxiliary predictions and the fused prediction respectively, prior to the ensemble process. Equations 3.39 and 3.40 compute the cross-entropy loss for the auxiliary predictions and the fused prediction respectively, after the ensemble process.

$$\mathcal{L}^{inner}_{meta} = \sum_{i=1}^{N} CE(\mathbf{p}_{m,s_i}, \mathcal{M}_q) \tag{3.37}$$

$$\mathcal{L}^{fused}_{meta} = CE(\mathbf{p}_{m,fused}, \mathcal{M}_q) \tag{3.38}$$

$$\mathcal{L}^{inner}_{final} = \sum_{i=1}^{N} CE(\mathbf{p}_{f,s_i}, \mathcal{M}_q) \tag{3.39}$$

$$\mathcal{L}^{fused}_{final} = CE(\mathbf{p}_{f,fused}, \mathcal{M}_q) \tag{3.40}$$

$$\mathcal{L}^{total}_{final} = \mathcal{L}^{inner}_{meta} + \mathcal{L}^{fused}_{meta} + \mathcal{L}^{inner}_{final} + \mathcal{L}^{fused}_{final} \tag{3.41}$$

The proposed method utilizes a cumulative loss function, as represented in Eq. 3.41, that aggregates all individual losses calculated for the auxiliary predictions and the fused prediction, both before and after the ensemble process. This accumulated loss is used to update the network parameters.

CHAPTER 4

EXPERIMENTAL EVALUATION

4.1 Experimental Setup

4.1.1 Dataset

The model is evaluated on two datasets which are commonly used in few-shot segmentation tasks. PASCAL-5^i [18] is the first dataset, containing 20 classes, and it is a combination of PASCAL VOC 2012 [41] and the extended annotations obtained from [42]. The second dataset is COCO-20^i [43], which is generated from MSCOCO [44]. COCO-20^i is more challenging when compared to PASCAL-5^i as it consists of images belonging to 80 classes. The datasets are split into 4 folds containing equal number of classes in order to perform cross-validation while 1000 support and query pairs are randomly sampled for each fold. One of the folds is selected for evaluating the performance of the model on unseen classes, while the rest of them are used as base classes for training the model. This procedure is repeated for all folds.

4.1.2 Implementation Details

All experiments are conducted on PyTorch framework with NVIDIA RTX 2080Ti GPUs. As suggested in BAM [1], there are two training stages, namely pre-training and meta-training. Pre-training stage is utilized for learning the base classes while ResNet-50 [39] and VGG-16 [45] are used as backbone for PASCAL-5^i and only ResNet-50 [39] is used as backbone for COCO-20^i. For PASCAL-5^i, PSPNet [16] is trained for 100 epochs as base learner with an initial learning rate of 2.5e-3. For the base learner on COCO-20^i, the model shared by the authors of [1] is used. In

meta-training stage, PASCAL-5^i and COCO-20^i are trained for 200 and 50 epochs respectively while the learning rate is set to 5e-2. For both stages, SGD is utilized as optimizer. Random scaling, rotation, horizontal flip, cropping and Gaussian Blur is applied to images. The sizes of the enriched query features at the output of the improved decoder are set to 60, 30, 15, and 8, which makes N = 4 as suggested by [11].

4.1.3 Performance Metrics

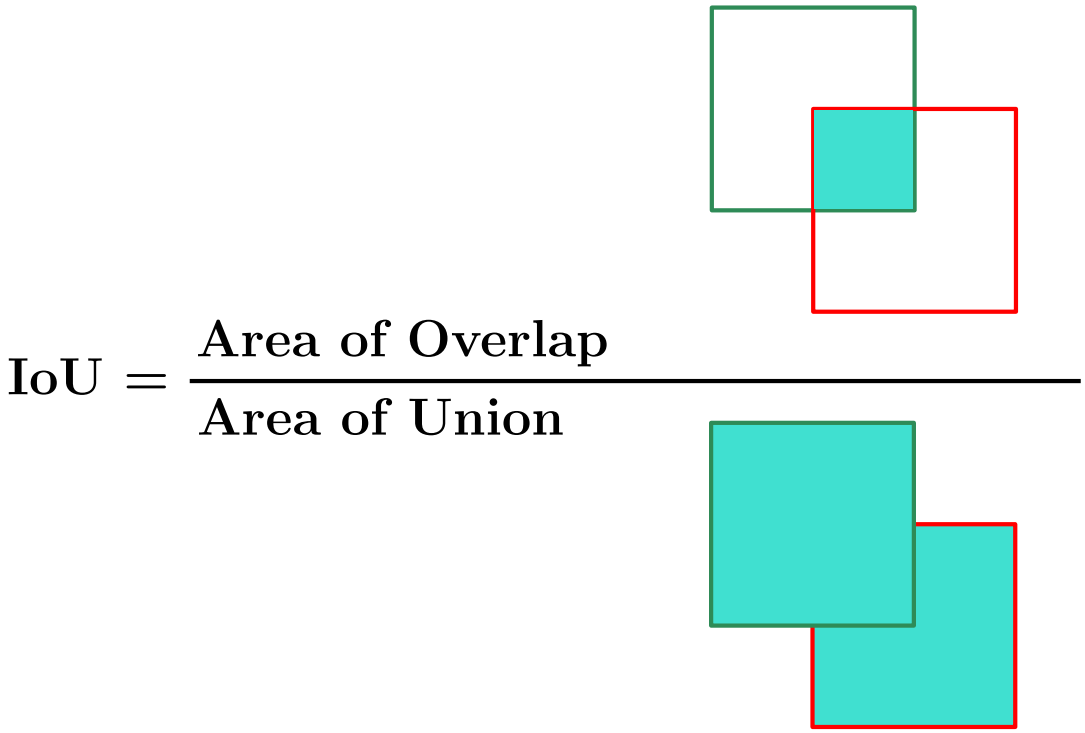

Figure 4.1: Green coloured box represents ground truth while red coloured box represents prediction.

The main metric used in the segmentation task is mean intersection over union (mIoU), which evaluates the degree of overlap between the predicted segmentation mask and the ground truth segmentation mask, as shown in Fig. 4.1. The IoU for each class is computed and then averaged to obtain the mIoU, as shown in Eq. 4.1. To compute the IoU, the confusion matrix should be first extracted. The confusion matrix counts the total number of pixels predicted as the jth class while belonging to the ith class at

the entry of c_{ij} of the confusion matrix. Let N_{cl} represent the total number of classes in the dataset.

The foreground-background IoU (FB-IoU) is also calculated as an additional metric.

$$mIoU = \frac{1}{N_{cl}} \sum_{i=1}^{N_{cl}} \left(\frac{c_{ii}}{\sum_{j=1}^{N_{cl}} c_{ij} + \sum_{j=1}^{N_{cl}} c_{ji} - c_{ii}} \right) \tag{4.1}$$

4.2 Experimental Results

4.2.1 Compared Methods

During our experimentation, we opted to evaluate the performance of proposed method against two other established models, namely, BAM and PFENet. We chose BAM as it represents a baseline version of proposed model that lacks the improved decoder. On the other hand, we selected PFENet as it includes an advanced decoder but does not have any measures in place to mitigate confusion of base classes as novel. Our decision to include these models in the evaluation was based on the desire to determine the efficacy of our modifications on segmentation performance.

We selected NTRENet, ASNet, and DPCN for our evaluation, as these models have exhibited state-of-the-art performance on the relevant benchmarks. It is noteworthy that our benchmark comprises approaches that almost entirely address the issues identified in Section 2. Specifically, NTRENet aims to resolve the challenge of misinterpreting the background by identifying universal background elements, as well as distracting objects, and subsequently eliminating them from the foreground prediction. ASNet, in turn, proposes a self-attention mechanism for cost-aggregation that reduces the size of the correlation matrix, thereby mitigating issues related to hypercorrelation reliability. Moreover, DPCN tackles the problem of thin objects by modifying the kernel and matching geometry. Furthermore, BAM and PFENet address inter-class gaps and spatial inconsistencies, respectively. In addition, we include a milestone study of PGNet to highlight the progress made in few-shot segmentation between 2019 and

2022. We assessed the segmentation performance of all the aforementioned methods based on their original papers, and did not perform any reimplementations.

4.2.2 Quantitative Results

Table 4.1: 1-shot and 5-shot class mIoU results on PASCAL-5^i dataset for VGG-16 and ResNet-50 as backbone, provided for 4 folds and the average. The best results are given in **boldface**. The underlined results show the best performance excluding our method.

Backbone	Method	1-shot (%)					5-shot (%)				
		Fold-0	Fold-1	Fold-2	Fold-3	Average	Fold-0	Fold-1	Fold-2	Fold-3	Average
VGG-16	PFENet (TPAMI'20) [11]	56.90	68.20	54.40	52.40	58.00	59.00	69.10	54.80	52.90	59.00
	NTRENet (CVPR'22) [46]	57.70	67.60	57.10	53.70	59.00	60.30	68.00	55.20	57.10	60.20
	DPCN (CVPR'22) [10]	58.90	69.10	63.20	55.70	61.70	63.40	70.70	68.10	59.00	65.30
	BAM (CVPR'22) [1]	63.18	70.77	66.14	57.53	64.41	67.36	73.05	70.61	64.00	68.76
	BAM++ (ours)	**64.67**	**72.11**	**67.83**	**59.47**	**66.02**	**69.40**	**74.35**	**72.77**	**67.19**	**70.93**
ResNet-50	PGNet (ICCV'19) [33]	56.00	66.90	50.60	50.40	56.00	57.70	68.70	52.90	54.60	58.50
	PFENet (TPAMI'20) [11]	61.70	69.50	55.40	56.30	60.80	63.10	70.70	55.80	57.90	61.90
	NTRENet (CVPR'22) [46]	65.40	72.30	59.40	59.80	64.20	66.20	72.80	61.70	62.20	65.70
	ASNet (CVPR'22) [47]	68.90	71.70	61.10	**62.70**	66.10	**72.60**	74.30	65.30	67.10	70.80
	DPCN (CVPR'22) [10]	65.70	71.60	69.10	60.60	66.70	70.00	73.20	70.90	65.50	69.90
	BAM (CVPR'22) [1]	68.97	73.59	67.55	61.13	67.81	70.59	75.05	70.79	67.20	70.91
	BAM++ (ours)	**69.46**	**74.16**	**69.20**	61.54	**68.59**	70.81	**75.34**	**73.04**	**68.99**	**72.05**

Table 4.2: 1-shot and 5-shot class mIoU results on COCO-20^i dataset for ResNet-50 as backbone, provided for 4 folds and the average. The best results are given in **boldface**. The underlined results show the best performance excluding our method.

Backbone	Method	1-shot (%)					5-shot (%)				
		Fold-0	Fold-1	Fold-2	Fold-3	Average	Fold-0	Fold-1	Fold-2	Fold-3	Average
ResNet-50	NTRENet (CVPR'22) [46]	36.80	42.60	39.90	37.90	39.30	38.20	44.10	40.40	38.40	40.30
	ASNet (CVPR'22) [47]	-	-	-	-	42.20	-	-	-	-	47.90
	DPCN (CVPR'22) [10]	42.00	47.00	43.20	39.70	43.00	46.00	54.90	50.80	47.40	49.80
	BAM (CVPR'22) [1]	43.41	50.59	**47.49**	43.42	46.23	49.26	54.20	**51.63**	**49.55**	51.16
	BAM++ (ours)	**44.43**	**51.98**	47.01	**45.22**	**47.16**	**52.53**	**57.02**	50.97	49.49	**52.50**

Table 4.1 shows the performance comparison between BAM++ and other methods proposed for few-shot segmentation task using ResNet-50 and VGG-16. The mIoU results include 1-shot and 5-shot cases for PASCAL-5^i dataset. BAM++ outperforms the existing methods for both settings. When VGG-16 is utilized as backbone, our

method surpasses the state-of-the-art results by 1.61% and 2.17% for 1-shot and 5-shot settings, respectively. When it comes to the model with ResNet-50 as backbone, 0.78% and 1.14% performance gains are achieved for 1-shot and 5-shot settings. The results on COCO-5^i dataset are provided in Table 4.2 for ResNet-50 as backbone only. BAM++ outperforms the best results by 0.93% and 1.34% under 1-shot and 5-shot settings, respectively.

Table 4.3: 1-shot and 5-shot FB-IoU results on PASCAL-5^i dataset for VGG-16 and ResNet-50 as backbone, provided as the average. The best results are given in **boldface**. The underlined results show the best performance excluding our method.

Backbone	Method	1-shot (%)	5-shot (%)
VGG-16	PFENet (TPAMI'20) [11]	72.00	72.30
	NTRENet (CVPR'22) [46]	73.10	74.20
	DPCN (CVPR'22) [10]	73.70	77.20
	BAM (CVPR'22) [1]	77.26	81.10
	BAM++ (ours)	**78.69**	**82.52**
ResNet-50	PFENet (TPAMI'20) [11]	73.30	73.90
	NTRENet (CVPR'22) [46]	77.00	78.40
	ASNet (CVPR'22) [47]	77.70	80.40
	DPCN (CVPR'22) [10]	78.00	80.70
	BAM (CVPR'22) [1]	**79.71**	82.18
	BAM++ (ours)	79.65	**82.84**

Comparison with state-of-the-art models regarding the FB-IoU scores is provided in Table 4.3 for both backbones on PASCAL-5^i dataset. The results show that our method performs well in 1-shot setting while exceeding the best result by 0.66% in 5-shot setting for ResNet-50. On the other hand, model with VGG-16 outperforms the previous state-of-the-art by 1.43% and 1.42% for 1-shot and 5-shot settings respectively.

4.2.3 Generalized few-shot segmentation results

Our method is also evaluated in generalized few-shot segmentation setting, which is defined by [1], where both pixels belonging to novel and base classes are detected. For this setting, novel pixels are predicted as *novel* if their final foreground probabilities exceed a predefined threshold, while the pixels predicted as *base* should be assigned to one of the base classes. By this way, the pixels belonging to different base classes

are distinguished while the rest of the pixels are classified as *novel* or *background*. This setting requires the calculation of mIoU on base and novel classes and also the combination of them, which are denoted by mIoU_n, mIoU_b and mIoU_a respectively. Our method surpasses BAM [1] in generalized few-shot segmentation setting for both backbones on PASCAL-5^i dataset as shown in Table 4.5. The mIoU results validate the superiority of ensembling at multi-scale for both novel and base predictions.

Table 4.4: Generalized few-shot segmentation results on PASCAL-5^i dataset for VGG-16 and ResNet-50 as backbone. The best results are given in **boldface**.

Backbone	Method	1-shot (%)			5-shot (%)		
		mIoU_n	mIoU_b	mIoU_a	mIoU_n	mIoU_b	mIoU_a
VGG-16	BAM [1]	43.19	67.03	61.07	46.15	67.02	61.80
	BAM++	**43.94**	**67.80**	**61.83**	**47.20**	**67.80**	**62.64**
ResNet-50	BAM [1]	47.93	72.72	66.52	49.17	72.72	66.83
	BAM++	**49.98**	**72.87**	**67.15**	**52.41**	**72.87**	**67.76**

4.2.4 Multi-scale few-shot segmentation results

Table 4.5: Multi-scale few-shot segmentation results on PASCAL-5^i dataset for ResNet-50 as backbone. The results presented in this table are obtained by averaging the results from fold-0 and fold-1. The x in the table corresponds to the total number of pixels in evaluated images.

Method	1-shot (%)			5-shot (%)		
	$x < 32^2$	$32^2 < x < 96^2$	$96^2 < x$	$x < 32^2$	$32^2 < x < 96^2$	$96^2 < x$
BAM	0.38	41.45	74.72	0.95	43.32	76.43
BAM++	0.54	42.36	76.18	1.29	44.35	77.31

Inspired by the COCO-20^i evaluation in object detection [43], we have partitioned masks into three distinct groups based on their size. The small group comprises objects whose foreground area occupies less than 32^2 pixels. The medium group includes objects whose foreground area falls between 32^2 pixels and 96^2 pixels, while the large group encompasses objects whose foreground area covers more than 96^2

pixels. These size categories have been chosen to facilitate the analysis and evaluation of the algorithms in a consistent and standardized manner. The proposed method in this thesis consistently outperforms BAM at three different scales, demonstrating the impact of the improved decoder.

4.2.5 Model Complexity

Compared to BAM, BAM++ has an additional 6 million parameters, bringing its total to 57.63 million parameters. However, a closer analysis of the multiplication and addition operations reveals that BAM++ performs 275.679 Giga Multiply-Accumulate operations, while BAM performs 273.733 Giga Multiply-Accumulate operations. This results in a negligible difference of merely 1.94 Giga Multiply-Accumulate operations. Consequently, during both training and inference, the difference between the two models leads to an insignificant gap of approximately one second per epoch.

4.2.6 Qualitative Results

Qualitative results for PASCAL-5^i dataset under 1-shot setting with ResNet-50 backbone are provided in Fig. 4.2. The differences between our proposed architecture and BAM can be seen when the predicted masks are analyzed. The main advantage of our model is revealed in cases where there is another object adjacent to the novel target object. In such cases, models generally tend to entangle the objects. In Fig. 4.2, it is seen that BAM predicts both the monitor and the computer as novel objects, although there is only monitor in the support image. Since our model analyzes the features at different scales, it distinguishes the neighboring objects from each other well. Moreover, another faulty case is given in the third row, which is consistent with our hypothesis. Even though base learner discourages meta learner from non-novel regions, i.e. sofa, meta learner of BAM predicts these regions as novel. When ensembling the query predictions at different scales is introduced, such incorrect predictions are eliminated. As it can be seen in the predicted map of our method, only the regions belonging to the dog are considered as foreground. We deduce that ensembling at

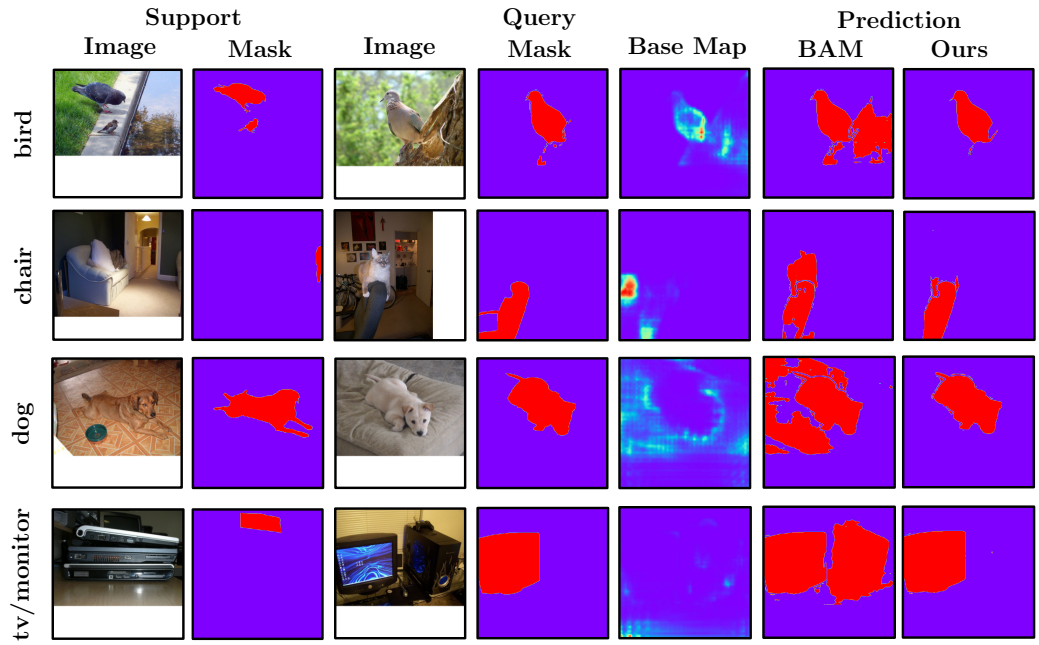

Figure 4.2: Qualitative 1-shot results on PASCAL-5^i dataset for ResNet-50 backbone. Results for one novel class from each fold are provided in rows. First two columns contain image and mask for support while the following two columns contain image and ground truth for query. Fifth column shows the probability map for query obtained from base learner. Predictions are provided for BAM [1] and our method for comparison in the last two columns. (Best viewed in color)

multi-scale ensures the model to focus on non-novel regions rather than the areas belonging to base classes.

4.2.7 Weakness of the proposed method

The base learner can perform well at segmenting objects that belong to the in distribution. However, it tends to make overconfident predictions on objects that belong to the out-of-distribution. This can cause the base learner to mislabel the out-of-distribution objects as belonging to one of the base classes with high confidence, thereby misleading the meta-learner about the base regions. As a result, the meta-learner may fail to predict the out-of-distribution regions as novel due to the misguidance of base map. For instance, in Fig. 4.3, we can see an example where a person is holding a bird over his hand. The person and the bird correspond to objects from the in-distribution and

out-of-distribution, respectively. As shown in Fig. 4.3, the base learner segments the region belonging to the bird as belonging to the in-distribution, failing to distinguish it from the base classes. Therefore, when the task for segmentation is the bird, the overall model overlooks the bird due to the incorrect base map.

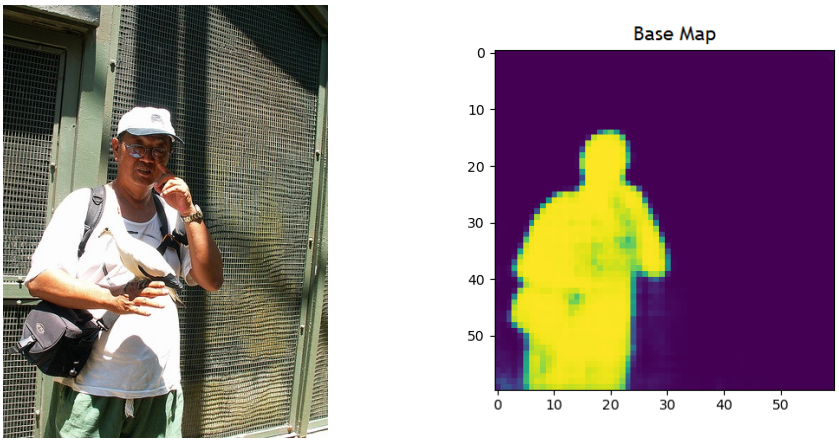

Figure 4.3: Misguidance of base map

4.2.8 Ablation Study

Table 4.6: Ablation studies on inner losses for the multi-scale predictions regarding the ensembling with the base map under 1-shot setting for PASCAL-5^i. Results show the averaged mIoU over 4 folds.

Method	$\mathcal{L}_{inner}^{meta}$	$\mathcal{L}_{inner}^{final}$	mIoU (%)
BAM++	✓	-	68.37
BAM++	-	✓	68.45
BAM++	✓	✓	68.59

Ablation study regarding the decision on how to include the inner losses for the multi-scale predictions is performed by considering the following cases: calculation of inner losses before and after the ensembling, without the ensembling, and after the ensembling only. The contributions of $\mathcal{L}_{meta}^{inner}$ in Eq. 3.37 and $\mathcal{L}_{final}^{inner}$ in Eq. 3.39 on the final mIoU performance are investigated. Thus, we experimented with the cases where

either $\mathcal{L}_{meta}^{inner}$ is inactive, $\mathcal{L}_{final}^{inner}$ is inactive, or both $\mathcal{L}_{meta}^{inner}$ and $\mathcal{L}_{final}^{inner}$ are active for the $\mathcal{L}_{final}^{total}$ calculation in Eq. 3.41. The results are obtained for PASCAL-5^i dataset under 1-shot setting and provided in Table 4.6. Activating only $\mathcal{L}_{meta}^{inner}$ reaches an mIoU performance of 68.37% while including $\mathcal{L}_{final}^{inner}$ alone obtains the performance of 68.45%. The last row in Table 4.6 indicates that when both $\mathcal{L}_{meta}^{inner}$ and $\mathcal{L}_{final}^{inner}$ are used, the highest performance is achieved, which is 68.59%. As consequence, this ablation experiment validates our hypothesis, which emphasizes the weakness of the model implementing ensembling at single scale and the merits of the co-existence of $\mathcal{L}_{meta}^{inner}$ and $\mathcal{L}_{final}^{inner}$.

CHAPTER 5

CONCLUSIONS

5.1 Summary

The present study proposes a novel approach that entails substituting the conventional decoder with an advanced version. Upon incorporating this enhanced decoder into the network architecture, it was observed that the auxiliary predictions were susceptible to bias, while the final predictions were effectively purified. To overcome this challenge, an ensembling mechanism was introduced to the intermediate predictions, which served to mitigate the bias and improve the accuracy of the overall predictions.

5.2 Conclusions

We observed that although ensembling meta prediction with base prediction guides the model by making the meta learner cautious in the regions where objects from base classes exist, meta learner misclassifies non-novel regions by neglecting base learner. This situation arises as a consequence of ensembling the predictions at single-scale. Therefore, we proposed to perform ensembling for predictions at multi-scale as well as the final prediction. By this way, bias existing at non-novel regions is diminished. The experiments on PASCAL-5^i and COCO-20^i verifies our hypothesis and our model achieves new state-of-the-art on few-shot segmentation benchmark.

5.3 Limitations and Future Work

Our method is not able to provide solution to thin object issue mentioned in Subsection 2.1.7. We empirically observe that our model fails to segment such a thin objects. An additional challenge arises with base prediction models, as they often exhibit overconfidence in their predictions when faced with out-of-distribution objects. However, uncertainty estimation techniques can be employed to identify instances of such erroneous predictions. Subsequently, these uncertainty maps can be leveraged to refine the initial base predictions.

REFERENCES

[1] C. Lang, G. Cheng, B. Tu, and J. Han, "Learning what not to segment: A new perspective on few-shot segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8057–8067, 2022.

[2] L. Yang, W. Zhuo, L. Qi, Y. Shi, and Y. Gao, "Mining latent classes for few-shot segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8721–8730, 2021.

[3] B. Zhang, J. Xiao, and T. Qin, "Self-guided and cross-guided learning for few-shot segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8312–8321, 2021.

[4] G. Li, V. Jampani, L. Sevilla-Lara, D. Sun, J. Kim, and J. Kim, "Adaptive prototype learning and allocation for few-shot segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8334–8343, 2021.

[5] G. Zhang, G. Kang, Y. Yang, and Y. Wei, "Few-shot segmentation via cycle-consistent transformer," *Advances in Neural Information Processing Systems*, vol. 34, pp. 21984–21996, 2021.

[6] Y. Liu, N. Liu, X. Yao, and J. Han, "Intermediate prototype mining transformer for few-shot semantic segmentation," *arXiv preprint arXiv:2210.06780*, 2022.

[7] Y. Sun, Q. Chen, X. He, J. Wang, H. Feng, J. Han, E. Ding, J. Cheng, Z. Li, and J. Wang, "Singular value fine-tuning: Few-shot segmentation requires few-parameters fine-tuning," *arXiv preprint arXiv:2206.06122*, 2022.

[8] J. Johnander, J. Edstedt, M. Felsberg, F. S. Khan, and M. Danelljan, "Dense gaussian processes for few-shot segmentation," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIX*, pp. 217–234, Springer, 2022.

[9] J. Min, D. Kang, and M. Cho, "Hypercorrelation squeeze for few-shot segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6941–6952, 2021.

[10] J. Liu, Y. Bao, G.-S. Xie, H. Xiong, J.-J. Sonke, and E. Gavves, "Dynamic prototype convolution network for few-shot semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11553–11562, 2022.

[11] Z. Tian, H. Zhao, M. Shu, Z. Yang, R. Li, and J. Jia, "Prior guided feature enrichment network for few-shot segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[12] J. Fan, F. Wang, H. Chu, X. Hu, Y. Cheng, and B. Gao, "Mlfnet: Multi-level fusion network for real-time semantic segmentation of autonomous driving," *IEEE Transactions on Intelligent Vehicles*, 2022.

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

[14] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[15] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, 2015.

[16] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2881–2890, 2017.

[17] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 801–818, 2018.

[18] A. Shaban, S. Bansal, Z. Liu, I. Essa, and B. Boots, "One-shot learning for semantic segmentation," *arXiv preprint arXiv:1709.03410*, 2017.

[19] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, *et al.*, "Matching networks for one shot learning," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[20] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in neural information processing systems*, vol. 30, 2017.

[21] N. Dong and E. P. Xing, "Few-shot semantic segmentation with prototype learning.," in *BMVC*, vol. 3, 2018.

[22] M. Boudiaf, H. Kervadec, Z. I. Masud, P. Piantanida, I. Ben Ayed, and J. Dolz, "Few-shot segmentation without meta-learning: A good transductive inference is all you need?," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13979–13988, 2021.

[23] J.-W. Zhang, Y. Sun, Y. Yang, and W. Chen, "Feature-proxy transformer for few-shot segmentation," *arXiv preprint arXiv:2210.06908*, 2022.

[24] S. Hong, S. Cho, J. Nam, S. Lin, and S. Kim, "Cost aggregation with 4d convolutional swin transformer for few-shot segmentation," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIX*, pp. 108–126, Springer, 2022.

[25] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[26] K. Wang, J. H. Liew, Y. Zou, D. Zhou, and J. Feng, "Panet: Few-shot image semantic segmentation with prototype alignment," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9197–9206, 2019.

[27] H. Wang, X. Zhang, Y. Hu, Y. Yang, X. Cao, and X. Zhen, "Few-shot semantic segmentation with democratic attention networks," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, pp. 730–746, Springer, 2020.

[28] Z. Lu, S. He, X. Zhu, L. Zhang, Y.-Z. Song, and T. Xiang, "Simpler is better: Few-shot semantic segmentation with classifier weight transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8741–8750, 2021.

[29] X. Yang, B. Wang, K. Chen, X. Zhou, S. Yi, W. Ouyang, and L. Zhou, "Brinet: Towards bridging the intra-class and inter-class gaps in one-shot segmentation," *arXiv preprint arXiv:2008.06226*, 2020.

[30] B. Liu, Y. Ding, J. Jiao, X. Ji, and Q. Ye, "Anti-aliasing semantic reconstruction for few-shot semantic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9747–9756, 2021.

[31] Z. Wu, X. Shi, G. Lin, and J. Cai, "Learning meta-class memory for few-shot semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 517–526, 2021.

[32] C. Zhang, G. Lin, F. Liu, R. Yao, and C. Shen, "Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5217–5226, 2019.

[33] C. Zhang, G. Lin, F. Liu, J. Guo, Q. Wu, and R. Yao, "Pyramid graph networks with connection attentions for region-based one-shot semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9587–9595, 2019.

[34] G.-S. Xie, J. Liu, H. Xiong, and L. Shao, "Scale-aware graph neural network for few-shot semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5475–5484, 2021.

[35] G.-S. Xie, H. Xiong, J. Liu, Y. Yao, and L. Shao, "Few-shot semantic segmentation with cyclic memory network," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7293–7302, 2021.

[36] H. Wang, L. Liu, W. Zhang, J. Zhang, Z. Gan, Y. Wang, C. Wang, and H. Wang, "Iterative few-shot semantic segmentation from image label text," *arXiv preprint arXiv:2303.05646*, 2023.

[37] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, pp. 8748–8763, PMLR, 2021.

[38] T. Lüddecke and A. Ecker, "Image segmentation using text and image prompts," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7086–7096, 2022.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[40] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2414–2423, 2016.

[41] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[42] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik, "Semantic contours from inverse detectors," in *2011 International Conference on Computer Vision*, pp. 991–998, IEEE, 2011.

[43] K. Nguyen and S. Todorovic, "Feature weighting and boosting for few-shot segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 622–631, 2019.

[44] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision*, pp. 740–755, Springer, 2014.

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

[46] Y. Liu, N. Liu, Q. Cao, X. Yao, J. Han, and L. Shao, "Learning non-target knowledge for few-shot semantic segmentation," in *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, pp. 11573–11582, 2022.

[47] D. Kang and M. Cho, "Integrative few-shot learning for classification and segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9979–9990, 2022.