Effective Few-shot Learning Approaches for Image Semantic Segmentation

by **Wenbo Xu**

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of Jian Zhang and Qiang Wu

School of Electrical and Data Engineering

Faculty of Engineering and IT

University of Technology Sydney

September 13, 2024

Certificate of Authorship / Originality

I, Wenbo Xu, declare that this thesis is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Electrical and Data Engineering at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis. This document has not been submitted for qualifications at any other academic institution.

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

Signature: Production Note: Signature removed prior to publication.

Date: September 13, 2024

Abstract

Semantic image segmentation has gained significant attention in computer vision due to its wide range of applications, including visual understanding, medical image analysis, self-driving vehicles, augmented reality, and video surveillance. While modern deep learning models have achieved surprising performance on segmentation tasks, it relies heavily on a massive amount of dense-labelled training data. However, abundant high-quality labeled data are not always available in real-world scenarios due to privacy or ethical concerns and safety issues. This research aims to reduce the reliance on data volume of segmentation tasks by introducing few-shot learning (FSL) technology. This empowers deep learning models to accurately segment unseen classes from only a few labeled images, thereby relieving researchers and engineers from intensive data labeling works.

This research initially addresses the problem of few-shot semantic segmentation (FSS), which requires segmenting the novel class objects in a test image on the condition of a few labeled data. For the challenges of prototype bias and sub-optimal feature representation, we propose the Masked Cross-image Encoding technique. This method captures shared information and mutual dependencies between training data and testing data, enhancing the visual properties of novel classes for improved prototype-feature matching. Then, we re-evaluate the standard binary matching paradigm employed in FSS and identify its association with potential false-matching and under-matching issues, which can significantly degrade segmentation performance. To alleviate this issue, a Multi-Prototype Discrimination scheme is introduced to explicitly assign each pixel-wise query features to a specific class, reducing class matching ambiguity present in conventional FSS methods. Building upon the FSS task, we tackle a more practical and challenging task known as Incremental Few-Shot Semantic Segmentation (iFSS). It requires a deep learning model to continuously learn new classes with scarce annotated examples, while retaining the knowledge learned from previously encountered classes. We consider a meta-

learning-based approach that simulates the incremental learning evaluation protocol during the base training stage. This training task alignment strategy encourages the model to learn how to incrementally adapt to novel classes without forgetting previous ones.

The overall research contributes valuable insights and methodologies to enhance the effectiveness of few-shot learning approaches for semantic image segmentation.

Publications

1. Xu, W., Huang, H., Cheng, M., Yu, L., Wu, Q., & Zhang, J. Masked cross-image encoding for few-shot segmentation. In 2023 IEEE International Conference on Multimedia and Expo (ICME) (pp. 744-749). IEEE.
2. Wenbo Xu, Yanan Wu, Haoran Jiang, Yang Wang, Qiang Wu, Jian Zhang. Task Consistent Learning for Few-shot Incremental Semantic Segmentation. Accepted by IEEE International Conference on Pattern Recognition (ICPR) 2024.
3. Wenbo Xu, Huaxi Huang, Yongshun Gong, Litao Yu, Qiang Wu, Jian Zhang. Multi-Prototype Discrimination: Rethinking Support-Query Matching for Few-Shot Segmentation. Submitted to IEEE Transactions on Multimedia (Revision submitted).

Acknowledgements

Time flies by in the blink of an eye, and it's hard to believe that three and a half years have passed. My Ph.D. journey began during the COVID-19 pandemic in 2020. Due to travel restrictions, I had to commence my studies remotely. The first year was spent at home, facing challenges unique to conducting research in isolation. Limited access to university resources and the inability to engage in face-to-face discussions with peers and supervisors slowed progress significantly.

However, despite these initial hurdles, the second year brought a significant change when I relocated to Australia. It was during this time that everything slowly fell into place with the help of my principal supervisor, Jian Zhang, and Dr. Litao Yu and Dr.Mingzhe Wang. Particularly noteworthy is Jian's indispensable role in guiding my research journey. His guidance has been invaluable, introducing me to research methodologies I had previously overlooked and assisting me in meticulously managing my research progress. Without his stringent guidance, I would not have been able to complete my PhD.

In addition, I am immensely grateful to my co-supervisor, Qiang, whom I had the pleasure of meeting after relocating to Australia. In contrast to Jian's assertiveness, Qiang's characteristics is gentle and patient. Our communication is always straightforward and efficient, devoid of any burdens. Whenever I encounter differences with Jian, I find solace in confiding in Qiang. His academic expertise and humble demeanor consistently alleviate the academic pressures I face.

I am also grateful to the following collaborators: Dr. Huaxi Huang, A/Prof. Yang Wang, Yanan Wu, and Haoran Jiang. Their insights and feedback have been invaluable, not only in guiding my research direction but also in significantly enhancing the quality of my manuscript through their contributions.Additionally, I am incredibly fortunate to have had supportive colleagues such as Yimin Liu, Kai Xu, Ming Cheng, Yike Wu, and other friends at UTS. Their

encouragement has been a source of strength during challenging times.

Furthermore, I would like to extend my heartfelt appreciation to my family for their unwavering love, encouragement, and understanding throughout this academic endeavor. My parents and sister have been a constant source of strength and inspiration, motivating me to persist in the face of challenges and pursue excellence in my academic pursuits.

In closing, I am deeply thankful to all those who have supported me throughout this journey. Your encouragement and guidance have been instrumental in shaping both my academic and personal growth, and for that, I am truly grateful.

Wenbo Xu

September 13, 2024

Sydney, Australia

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Research Background

Semantic image segmentation is a core field in image processing and computer vision, focusing on the task of classifying each pixel in an image into distinct categories. This pixel-level classification is fundamental in a wide array of applications, including but not limited to medical image analysis for tumor identification and chest X-rays [1]–[3], Advanced Driver-Assistance Systems (ADAS) in autonomous vehicles [4], [5], robotics [6], [7], environmental monitoring [8], [9], and precision agriculture [10], [11]. The literature is rich with various algorithms developed for this purpose, ranging from traditional techniques like statistical region merging [12], [13], k-means clustering [14], [15], and the use of conditional and Markov random fields [16], to more advanced deep learning (DL)-based methods. These DL methods encompass Fully Convolutional Networks [17], Encoder-Decoder [18] and Auto-Encoder architectures [19], Dilated Convolutional Models[20], and members of the DeepLab Family [21]. More recently, transformer-based models [22]–[25] have gained prominence in the field of image segmentation, demonstrating significant improvements in performance and achieving near-optimal accuracy levels in tests conducted using publicly accessible datasets.

Nevertheless, fully-supervised DL models exhibit a heavy reliance on data, necessitating a substantial volume of annotated training data to effectively train a model for a specific task. Consequently, a natural inclination arises: can we devise a methodology to adapt a model to novel classes in a more data-efficient manner? In other words, can we extract generalizable features for new classes from a limited number of exemplars? In response to this query, a

novel machine learning paradigm known as Few-shot learning (FSL) [26] has been proposed, mirroring the manner in which human beings acquire knowledge. For example, a child can easily identify an unfamiliar person from a multitude of photos after seeing only a few images of that person.

FSL helps overcome the barrier of collecting data in some complex scenarios where images with densely annotated information are hard or impossible to obtain due to privacy, security or ethical issues. Especially, semantic image segmentation training data are all intensively annotated at the pixel level, which requires a considerable amount of time annotating every single example. FSL is one of the ideal strategies to reduce the data gathering and labeling effort for segmentation tasks.

Driven by the goals of Few-shot Learning (FSL), a variety of machine learning approaches have emerged, including meta-learning [27]–[29], embedding learning [30], [31], and generative modeling [32]. Initially, these methods were primarily applied to solve basic computer vision tasks like image classification, image retrieval, and object detection. It was not until 2017 [33] that Few-shot and One-shot learning began to be explicitly applied to semantic image segmentation. Few-shot segmentation (FSS), where the model is expected to learn to segment objects from a very limited number of densely labeled examples, poses several unique challenges. Generalizing from a few examples is inherently challenging, as the model needs to learn from a small sample of data and make predictions on unseen data, which can be significantly different from the training samples. This requires the model to capture the essence of the class from very few examples, a task that is not trivial.

With a very limited number of examples, models are at a high risk of overfitting. Overfitting occurs when the model learns patterns that are specific to the training set, rather than learning generalizable features, leading to poor performance on unseen data. The quality and representativeness of the few examples are critical; if the selected examples are not representative of the class, the model may learn incorrect or incomplete representations.

Objects belonging to the same class can have significant variations in appearance, pose, size, and context, adding to the complexity of learning. Capturing this intra-class variability and learning robust class representations from very few examples is a challenging task. Furthermore, learning discriminative visual properties from a small number of examples is difficult as the model must extract and leverage the most salient features from the constrained data.

To address FSS, a foundational framework was introduced by Amirreza and Shray [33]. This framework, which builds upon the Siamese Network concept, includes two branches: a support branch that processes an annotated image to generate vectorial parameters (termed as prototypes), and a query branch that uses the support branch's output along with a test image of an unseen class to produce a segmentation mask. This structure, coupled with the 'Learning to learn' philosophy, has significantly influenced the development of FSS.

Dominant strategies in few-shot segmentation [34]–[39] focus heavily on prototype learning within the support-query structure. Most of them apply methods like masked average pooling or feature clustering to learn class vectors that capture essential segmentation cues from support instances. This paradigm naturally faces several challenges from the following aspects: 1) General feature embedding techniques fall short in addressing few-shot challenges, highlighting the need for more sophisticated and resilient feature representation methods. 2) The interaction between guiding maps and the guiding process in current models is not optimal for effective segmentation. Enhancing models to better generalize and logically guide the segmentation process is crucial in few-shot segmentation. This thesis aims to address these limitations, offering a thorough understanding, detailed analysis, and effective solutions for training deep learning models to achieve enhanced generalization and robustness in FSS.

Despite its potential, conventional few-shot segmentation faces significant challenges that limit its applicability in real-world settings. This is because it often operates under the assumption that the entire set of classes is known a priori, which is rarely the case in dynamic and evolving environments.

To take a step further, this thesis tackles incremental Few-shot Segmentation (iFSS) as a more practical and adaptive approach for efficient image semantic segmentation tasks. Unlike the traditional FSS paradigm, which assumes a static set of classes, iFSS acknowledges the dynamic nature of real-world environments where new classes can emerge over time. This task not only leverages the minimal data available for each class but also adapts to the introduction of new classes without forgetting the previously learned ones. This is crucial in scenarios where systems continuously encounter novel categories and must adapt without extensive retraining or manual annotation.

iFSS is still under-explored in the literatures which can be considered as a combination of incremental learning and few-shot learning. It poses unique challenges, primarily the notorious

problem of catastrophic forgetting, where integrating new knowledge erases previous learning. Additionally, the model must maintain robust generalization capabilities, performing well on both old and new classes, despite the scarcity of data.

Insights gleaned from the literature on incremental learning reveal that a direct and effective strategy to address catastrophic forgetting is data replay [32], [40], which involves preserving and revisiting crucial data or features from prior classes, thereby reinforcing and maintaining pre-existing knowledge while assimilating new information. In contrast, space-based methods [41] focus on developing a feature space that can seamlessly incorporate new classes without compromising the representation of existing ones, ensuring a balanced coexistence of old and new knowledge. Alternatively, dynamically adjusting the network's architecture [42], [43]—either by expanding it with new neurons or by selectively activating specific sections of the network based on the task at hand—can endow a more adaptive and resilient learning ability to the model. Nonetheless, the practice of preserving a sample of past data to represent previous classes can be unfeasible in situations involving numerous classes or data with high dimensionality. Maintaining a feature space that is both distinctive and reflective of all classes, both new and old, as the model progresses, poses a significant challenge. Moreover, while dynamic network methods provide adaptability and versatility, they tend to grow in complexity and computational demands as the network expands, presenting scalability issues.

Grounded in the objectives of incremental few-shot segmentation and addressing the inherent limitations of incremental learning, this thesis is dedicated to crafting deep learning models that are efficient in terms of parameters and training process. These models are designed to effectively tackle the iFSS challenge, ensuring they do not overfit to novel classes while maintaining their segmentation prowess on classes encountered previously.

1.2 Research Objectives and Overview

This thesis delves into effective few-shot learning approaches for image semantic segmentation, which disclose the exciting potential of deep learning models to revolutionize image understanding. Training models on massive datasets consumes significant computational resources and energy. Few-shot learning offers a potential solution by achieving comparable accuracy with significantly less data, leading to more efficient and sustainable model development. FSS aims to bridge this gap by enabling accurate segmentation with only a handful of labeled examples.

This opens doors to applying semantic segmentation in diverse fields like medical imaging, autonomous driving, and robotics, where data scarcity is often a hurdle. iFSS techniques empower model to learn and adapt quickly to new visual concepts, without forgetting old ones. This flexibility is crucial for real-world applications where environments and objects can vary significantly.

The specific research objectives of this thesis are:

1. Design novel FSS architectures capable of extracting rich semantic features from a limited pool of labeled data. These architectures should leverage knowledge transfer across related visual concepts to compensate for data scarcity.

2. Introduce strategies for guiding features that adeptly manage the variability within classes and the distinction between different classes, thereby refining the models' ability to generalize and intelligently navigate the segmentation process

3. Develop incremental FSS methods that enable models to continuously adapt to new classes with minimal labeled examples while retaining their segmentation ability on previously encountered classes.

Chapter 2 provides a comprehensive review of literature related to image semantic segmentation. It covers the spectrum of semantic segmentation, from foundational architectures in conventional methods to few-shot and incremental learning. It deeply analyzes few-shot semantic segmentation, discussing innovative strategies like meta-learning and data augmentation for effective generalization from limited data. Additionally, it delves into the intricate task of incremental few-shot segmentation, evaluating how these approaches skillfully balance learning new information with preserving existing knowledge.

Chapter 3 introduces a novel masked cross-image encoding (MCE) approach to identify common visual representations of target objects within support and query images. Utilizing a symmetric cross-attention framework, MCE capitalizes on bidirectional inter-image relations across multiple feature levels. This not only infuses query features with significant details from support object regions but also strengthens the support-query interaction, enhancing the model's feature representation capabilities for Few-Shot Segmentation (FSS).

Chapter 4 presents a comprehensive FSS approach aimed at addressing the inherent undermatching and mismatching issues within the support-query prototype matching framework.

Building upon the robust feature representation capabilities of MCE, as discussed in Chapter 3, this chapter presents a multi-prototype matching approach. This approach aims to narrow the semantic divide between support prototypes and query features, enhancing the differentiation between similar novel and base classes effectively.

Chapter 5 delves into a pragmatic aspect of semantic segmentation, focusing on the incremental learning of new classes from a limited set of examples. This chapter proposes a meta-learning-based strategy specifically designed to optimize the network's capacity for incremental learning in a few-shot context. To mitigate the challenges of catastrophic forgetting and overfitting, a novel prototype space re-distribution mechanism is introduced. This mechanism is responsible for dynamically updating class prototypes throughout each incremental phase, ensuring consistent learning progress.

Chapter 2

Literature Review

As introduced in chapter 1, the labor-intensive nature of acquiring such large annotated datasets has prompted a shift towards more data-efficient learning paradigms for semantic segmentation. This literature review begins by providing a comprehensive overview of conventional semantic segmentation which offers crucial insights into the foundational architectures and learning mechanisms that underpin semantic segmentation. Transitioning from fully-supervised paradigms, the review then explores the realm of few-shot learning and semi-supervised learning methods. Their principles and techniques are critical for understanding the underpinnings of few-shot semantic segmentation.

The core of the review is dedicated to analyzing current methodologies in few-shot semantic segmentation. This section scrutinizes the innovative approaches that enable models to generalize effectively from limited examples, discussing various strategies like meta-learning, metric learning, and data augmentation that are employed to overcome the challenges posed by few-shot settings.

Furthermore, the review ventures into the more challenging task of incremental few-shot segmentation. This section evaluates works on few-shot incremental learning and incremental few-shot segmentation, examining how these methods balance the acquisition of new knowledge with the retention of previously learned information.

2.1 Datasets and Metrics

Commonly used open-access datasets

- PASCAL VOC dataset, referenced as VOC [44], has been a fundamental resource for evaluating object detection and segmentation algorithms in computer vision research. Despite its moderate size, consisting of approximately 20,000 images spanning around 20 object categories, VOC stands out for its well-defined classes and meticulously annotated data. This dataset's emphasis on high-quality annotations has positioned it as a crucial benchmark for swiftly verifying ideas and exploring emerging research areas in the field of computer vision. Its comprehensive evaluation has made it a cornerstone in the development and assessment of segmentation techniques. Moreover, this dataset serves as the primary dataset employed in the research conducted in this thesis.

- Microsoft COCO dataset (Common Objects in Context) [45] is a large-scale, richly annotated dataset designed for evaluating the performance of computer vision algorithms in three key areas: object detection, image segmentation, and image captioning. The dataset encompasses over 80 object categories, ranging from common objects like "dog" and "car" to more specific ones like "fire hydrant" and "baseball bat." For segmentation, it provides pixel-wise annotations that define the exact shape and boundaries of each object, enabling the training of image segmentation models that can classify every pixel in an image according to its corresponding object class.

- ADE20K [46] dataset is a semantic segmentation dataset that contains more than 20K scene-centric images exhaustively annotated with pixel-level objects and object parts labels. There are totally 150 semantic categories, which include stuffs like sky, road, grass, and discrete objects like person, car, bed. The dataset is used as the benchmark for scene parsing and instance segmentation. The effect of synchronized batch normalization is evaluated and it is found that a reasonably large batch size is crucial for the semantic segmentation performance.

- Cityscapes [47] dataset is a semantic segmentation dataset that contains 5,000 high-quality images of 50 different cities captured from a car-mounted camera. The dataset is used for evaluating the performance of semantic segmentation algorithms on urban scenes. The dataset includes pixel-level annotations for 30 classes of objects and stuffs such as road, sidewalk, building, vegetation, person, car, etc. The dataset is widely used in the field of computer vision and machine learning for developing and testing new algorithms for semantic segmentation.

- FSS-1000 [48] is a large-scale dataset specifically designed for training models in few-shot segmentation tasks. Unlike prior datasets that focused on a limited number of well-represented classes, FSS-1000 boasts a whopping 1,000 object classes. This allows models to learn from a wider variety of objects. The dataset incorporates many objects not found in previous segmentation datasets, such as tiny everyday items, merchandise, cartoon characters, and logos. Each class only has 10 images with corresponding segmentation masks. This aspect aligns with the few-shot learning paradigm where models need to learn with minimal training data.

2.2 Evaluation Metrics

There are many evaluation matrics to measure the performance of image segmentation. This section explain the main evaluation matrics most commonly used in measuring semantic segmentation performance:

Pixel Accuracy: Pixel Accuracy is a widely used evaluation metric in the field of image segmentation. It measures the accuracy of pixel-level classification by comparing the predicted segmentation mask with the ground truth mask on a pixel-by-pixel basis. Pixel Accuracy calculates the percentage of correctly classified pixels in the segmentation output. It provides a straightforward assessment of the overall segmentation performance by considering both foreground and background pixels. A pixel is considered correct if its predicted class label matches the ground truth label. And mean Pixel Accuracy (mPA), calculates the average accuracy of each class and provides insights into the segmentation performance.

$$PA = \frac{\sum_{j=1}^{k} n_{jj}}{\sum_{j=1}^{k} t_j}, \quad mPA = \frac{1}{k} \sum_{j=1}^{k} \frac{n_{ij}}{t_j} \tag{2.1}$$

The n_{jj} in formula 2.1 denotes the total number of pixels both classified and labelled as class j.

Intersection over Union (IoU) and mean Intersection over Union (mIoU) are mostly used evaluation metrics in image segmentation tasks.

Intersection over Union (IoU): Intersection over Union measures the overlap between the predicted segmentation mask and the ground truth mask for a specific class. It is calculated by dividing the intersection of the two masks by the union of the two masks. IoU provides

a measure of how well the predicted mask aligns with the ground truth, with a value of 1 indicating a perfect match and lower values indicating less accurate segmentation.

Mean Intersection over Union (mIoU): It calculates the average IoU across all classes in the dataset. It provides an overall measure of the segmentation performance by taking into account the performance across multiple classes. mIoU is particularly useful when dealing with imbalanced datasets where some classes may have more instances than others. The class-averaged IoU, illustrates as:

$$mIoU = \frac{1}{k}\sum_{j=1}^{k}\frac{n_{ij}}{n_{ij} + n_{ji} + n_{ij}}, \quad i \neq j \tag{2.2}$$

Foreground-Background Intersection over Union (FBIoU): This metrics is an evaluation metric used in image segmentation tasks to specifically assess the accuracy of segmenting the foreground (object) and background regions. It evaluates the overlap between the predicted foreground and background segmentation masks with their respective ground truth masks. FB-IoU is particularly relevant in tasks where differentiating between foreground objects and the background is crucial, which is most frequently adopted matric for few-shot segmentation as the predicted mask are binary mask for unseen classes. FB-IoU is often calculated as the average of the Foreground IoU and the Background IoU as:$FB - IoU = \frac{1}{2}\left(\mathrm{IoU_F} + \mathrm{IoU_B}\right)$.

2.3 Supervised semantic segmentation

Early approaches relied on hand-engineered features and classical machine learning algorithms. Traditional image segmentation techniques, such as thresholding [49], [50], region growing [12], [13], and clustering [14], [15], are employed to divide an image into meaningful regions. Thresholding techniques, for instance, were used to separate objects from the background based on pixel intensity values. Region-growing methods involved selecting seed points and then adding neighboring pixels to the segment if they had similar properties. There are two main types of clustering methods: hierarchical and partitional. Hierarchical methods create a tree-like structure of clusters by either merging smaller clusters into larger ones (agglomerative) or splitting larger clusters into smaller ones (divisive). Partitional methods assign each pixel to one of the predefined number of clusters, such as K-means, fuzzy C-means, or meta-heuristic methods. To obtain pixel-level scene labels, various graphical models such as Markov Random Fields (MRF) [16], Conditional Random Fields (CRF) [51], and forest-based methods [52] are com-

monly utilized. These models leverage the relationships between neighboring pixels to infer the scene labels, allowing for the incorporation of spatial dependencies in the segmentation process. While deep learning methods have largely dominated the field of segmentation, graphical models, particularly Conditional Random Fields (CRF), continue to serve a purpose in recent approaches. They are often employed as a post-processing tool to refine and enhance the semantic segmentation results obtained from deep learning models

With the development of powerful computation graphics processing units (GPUs), deep learning has demonstrated remarkable success in addressing computer vision challenges. The evolution of semantic segmentation is closely tied to advancements in neural networks, particularly Convolutional Neural Networks (CNN). In the early stage of deep learning, CNN was considered a powerful feature extractor to replace hand-crafted descriptors. Many researchers were dedicated to exploiting novel deep features for semantic segmentation [53], [54]. However, the fully connected layers in CNN cause overfitting and computation-consuming problems as well as the solid input size for the semantic segmentation tasks.

Fully Convolutional Networks. The concept of eliminating fully connected layers from convolutional neural networks (CNN) led to the development of the Fully Convolutional Network (FCN) by E. Shelhamer and J. Long [17]. This approach is considered a breakthrough in DL-based semantic segmentation. In FCN, the conventional fully connected layers, which discard spatial information, are replaced with convolutional layers. This architectural modification enables classification to be performed at the pixel level. Notably, FCN also introduced deconvolution layers, allowing it to process input images of any size and produce segmentation maps with identical resolution.By incorporating skip connections, the model leverages the fusion of feature maps obtained from both high-level layers, which capture semantic information at a coarse level, and low-level layers, which capture detailed appearance features. This integration facilitates the generation of precise and comprehensive segmentations. Specifically, the feature maps from the deeper layers of the model are up-sampled to the same resolution as prior layer's output and then fused with those feature maps from lower layers, enabling the model to effectively combine semantic understanding with fine-grained visual details.

Encoder-Decoder Architecture. The encoder-decoder architecture is a fundamental framework in the field of semantic segmentation, playing a crucial role in modern deep learning approaches. This architecture is designed to effectively process and analyze visual data by

capturing spatial hierarchies and semantic information. As demonstrated in 2.1, the encoder part of the architecture is responsible for feature extraction. It typically consists of a series of convolutional layers followed by pooling layers. As the input image progresses through the encoder, it undergoes a series of transformations that reduce its spatial dimensions while increasing the depth of feature maps. This process helps in capturing the context and understanding the content of the image at different scales. The encoder effectively captures high-level semantic information; however, due to the pooling layers, there's a loss of spatial resolution, which is crucial for precise localization in segmentation tasks. The decoder's primary function is to transform abstract, high-dimensional feature representations learned by the encoder into meaningful pixel-wise predictions, leading to accurate and detailed segmentation of objects in the input image. It progressively recovers the spatial dimensions of the feature maps through operations such as upsampling, deconvolution, or transposed convolution. The decoder also utilizes skip connections from the encoder to the decoder layers. These connections help in combining the high-level semantic cues and low-level geometric information, essential for accurate segmentation. The final layer of the decoder typically involves a 1x1 convolution followed by a softmax activation function to assign each pixel a predefined class label. Notably, Noh et al. [55] introduced a seminal paper that utilized deconvolution-based architecture for semantic segmentation. Several models based on the encoder-decoder framework have been proposed, with U-Net [19] and SegNet [18] being notable examples. U-Net is particularly renowned in medical image segmentation for its effective use of skip connections, while SegNet is recognized for its efficient use of pooling indices to perform upsampling in the decoder. Variants and improvements to this architecture, such as using atrous (dilated) convolutions [20], [56], feature pyramid networks (FPN) [21], [57], and attention mechanisms [58], [59], have been developed to enhance its performance in semantic segmentation tasks. In parallel, recent endeavors have sought to bridge the gap between neural inductive learning and logic reasoning by integrating rich data and symbolic knowledge, as exemplified by LogicSeg [60]. Another innovative direction involves considering the joint distribution of pixel features and class labels, as seen in GMMSeg [61]. Additionally, pixel-wise contrastive learning techniques, as proposed in [62], have emerged as a valuable tool for semantic segmentation in fully supervised settings.

Spatial Pyramid Pooling (SPP) [63] is a method incorporated into convolutional neural networks to capture contextual features across multiple scales, significantly enhancing segmentation performance. As illustrated in Figure 2.2, it addresses this issue by introducing a special

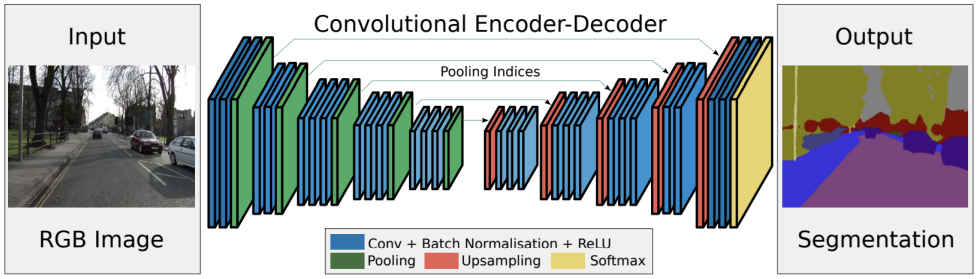


Figure 2.1: Illustration of the Encoder-Decoder architecture. Adapted from [18]

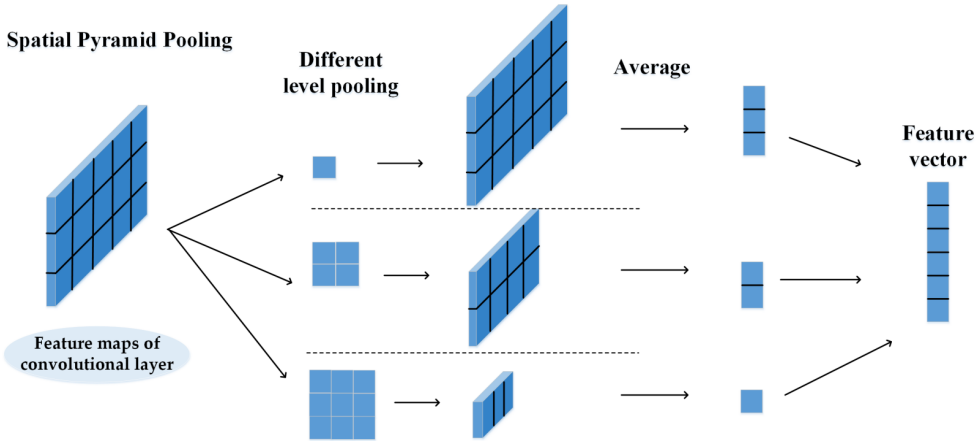


Figure 2.2: The structure of Spatial Pyramid Pooling (SPP)

layer that pools the feature maps from the last convolutional layer into a fixed size, regardless of the input image size. SPP creates a pyramid of levels, each with a different bin size (e.g., 1x1, 2x2, 3x3). The idea is to capture features at various scales. For each level of the pyramid, SPP applies pooling (usually max pooling) on the feature maps. Each bin pools a region of the feature map into a single number, regardless of the region's size. This results in a fixed number of outputs per level. SPP can be integrated into encoder-decoder architectures for segmentation. In such architectures, SPP is usually placed at the end of the encoder or before the decoder. This allows the network to handle inputs of varying sizes and to capture multi-scale features, improving the quality of the segmentation.

However, if the stride values of a pooling layer, positioned immediately before a decision layer, are proportional to the input size, the resultant feature map of that layer will have a fixed dimension. Li et al. [64] introduced a refined variant known as the Pyramid Attention Network (PAN). PAN integrates an SPP layer with global pooling, aiming to enhance feature representation learning by leveraging the combined strengths of both techniques.

Dilated convolution [20], also known as atrous convolution, is another important technique commonly employed in semantic segmentation for capturing contextual information at different scales. Unlike traditional convolution, which uses a fixed-sized kernel, dilated convolution introduces gaps or "holes" within the kernel, thereby enlarging its receptive field without extra unnecessary parameters. By adjusting the dilation rate, the receptive field can be effectively expanded to incorporate larger contextual regions. This allows the model to capture object boundaries and capture context across multiple scales, improving the accuracy and spatial coherence of the segmentation results.

The DeepLab series [20], [21], [56] stands out as a landmark in the realm of semantic segmentation models, introducing a host of groundbreaking innovations. Among its notable contributions is the integration of atrous convolutions and SPP into the feature fusion module known as Atrous Spatial Pyramid Pooling (ASPP) module. By incorporating ASPP, DeepLabv3+ [20] can effectively capture objects of varying sizes and their surrounding context within an image. This leads to more accurate and robust semantic segmentation performance.

Transformer-based segmentation. Propelled by the success of natural language processing (NLP), the Transformer [65] model has been adapted for the field of computer vision. A pivotal adaptation is the Vision Transformer (ViT) [66], which dissects an image into patches, transforms them into vectors, and feeds them into a transformer encoder. The encoder's outputs serve various vision-related tasks, including recognition, segmentation, and image generation. Since then, many variants and extensions [67]–[70] of ViT have been proposed to improve its efficiency, accuracy, and applicability to different vision tasks.

Talking to semantic segmentation task, SETR [71] is the pioneer in substituting the CNN backbone with the ViT backbone, setting new benchmarks on the ADE20k dataset. After SETR, researchers start to design more powerful vision transformers for segmentation task. Typically, Segformer [22] is considered as a classic segmentation framework that combines an efficient hierarchical Transformer encoder with a lightweight MLP decoder. Its encoder, Mix Transformer encoder (MiT), captures multi-scale features through a pyramid structure, while the simple decoder fuses these features to produce high-quality segmentation maps. To improve parameter efficiency, SegViT [25] introduces a new component named Attention-to-Mask (ATM), which creates pixel-level labels from the comparison between class tokens and spatial features. It also uses query-based methods to lower and raise the resolution of ViTs,

making them more efficient. With the success of ViTs, recent representative works [72], [73] merge the global receptive field of transformers with the local feature extraction prowess of U-Net [19]. TransUnet [72] transforms image patches into tokens and constructs the global context. Then it increases the resolution of the encoded features using decoder, and merges them with the detailed CNN features to achieve accurate localization. Swin-Unet [73] integrates Swin Transformer [68] blocks that use shifted windows to capture local and global information, achieving remakable performance on serveral medical image segmentation benchmarks.

Besides using a sophisticated transformer encoder, another approach to enhance the transformer architecture is by focusing on the cross-attention mechanism. Segmenter [23] employs a class query, which is a learnable vector that represents a class, to directly produce class-wise masks, which are binary masks that indicate the presence of a class in each pixel. Segmenter is a pure mask-based approach, which means it does not rely on intermediate representations such as bounding boxes or instance centers. Pure mask-based approaches can generate more accurate masks from high-resolution features. Max-Deeplab [74] is a model unifies semantic segmentation and panoptic segmentation (PS). It is the first model to remove the box head, which is a component that predicts bounding boxes for object instances, and design a pure-mask-based segmenter for panoptic segmentation. It combines a CNN-transformer hybrid encoder and a transformer decoder as an extra path, achieving stronger performance than box-based methods. Meanwhile, MaskFormer [75] focuses on a per-pixel classification strategy, where each pixel is associated with a set of learnable mask embeddings. It efficiently handles both semantic and instance segmentation tasks by predicting masks and their corresponding classes, demonstrating versatility across various segmentation challenges.

In parallel, recent endeavors have sought to bridge the gap between neural inductive learning and logic reasoning by integrating rich data and symbolic knowledge, as exemplified by LogicSeg [60]. Another innovative direction involves considering the joint distribution of pixel features and class labels, as seen in GMMSeg [61]. Additionally, pixel-wise contrastive learning techniques, as proposed in [62], have emerged as a valuable tool for semantic segmentation in fully supervised settings.

Each of these models represents a stride towards more accurate, context-aware, and efficient segmentation, leveraging the transformer's ability to model complex dependencies in visual data. They embody the synergy of global and local processing, setting new standards in the

field of image segmentation. Overall, whether utilizing CNN or Transformer architectures, the encoder-decoder framework serves as a fundamental cornerstone for segmentation tasks. Its capability to extract contextual features and convert them into pixel-wise classifications renders it a pivotal tool for both Few-Shot Semantic Segmentation and incremental Few-Shot Semantic Segmentation tasks in this thesis.

2.4 Few Shot Learning

Traditional machine learning paradigms often necessitate substantial amounts of labeled data to attain satisfactory performance levels. However, the acquisition and annotation of such datasets are frequently resource-intensive, time-consuming, and occasionally unfeasible for certain domains. Few-shot learning (FSL) emerges as a pivotal solution to mitigate these challenges inherent in conventional machine learning methodologies. By focusing on learning from a limited set of labeled instances, few-shot learning endeavors to enhance the data efficiency of models, thereby offering a promising avenue for addressing the constraints associated with extensive data requirements in machine learning research and applications. Given that semantic segmentation can be regarded as a pixel-wise challenge, numerous few-shot semantic segmentation techniques draw inspiration from FSL methodologies.

Normally, the few-shot training set contains many classes, and each class has multiple samples. In the training phase, C categories will be randomly selected from the training set, with K samples for each category (a total of $N * K$ data), and a meta-task will be constructed as the input of the model's support set, then a batch of samples belonging to N classes from the remaining data are picked as the model's prediction object (batch set). That is, the model is required to learn how to distinguish these N categories based on $N * K$ data. Such a task is called a N-way K-shot problem.

Building upon the setting, few-shot learning can be interpreted as an instantiation of meta-learning within the realm of supervised learning. Meta-learning, often referred to as 'learning to learn,' operates by partitioning the dataset into distinct meta-tasks during the meta-training phase, facilitating the acquisition of the model's generalization capabilities amidst class variations. In the subsequent meta-testing phase, when encountering novel, unseen classes, the model can execute classification tasks without necessitating alterations to its existing architecture.

During the training stage, each training task (episode) adopts different meta-tasks, in other words, the training contains different combinations of classes. This mechanism allows the model to learn the common 'experiences' of different meta-tasks, such as how to extract critical features based on the similarity between samples and ignoring the specific task. The model learned through this learning mechanism can also be better classified when facing new and unseen meta-tasks. The subsequent sections delve into three primary strategies of few-shot learning.

2.4.1 Meta-learners for Few-Shot Learning

One type is optimization-based methods. In order to quickly adapt to new tasks, most work focuses on how to learn a good parameter initialization.

Model-agnostic Meta-Learning (MAML) [76] first proposed that the parameters of the model can be generalized to new tasks with only a few gradient descents and a small amount of training data. MAML uses a large amount of data to train a meta-model, and then fine-tunes it based on limited labeled samples on new tasks to obtain the final model. Specifically, the process of training the meta-model is to first randomly sample multiple tasks for training according to the experimental setting of N-way K-shot. Since MAML is based on double gradients, as depicted in Figure 2.3 , the original model will be copied first, and the gradient will be updated based on the loss of each task on the copied model. This is the first gradient update. Then the second gradient update is calculated based on the batch, and this gradient is applied to the original model, thus completing the training process of the meta-model in MAML. When fine-tuning is performed, the new model will directly load the initialization parameters of the meta-model, and then update the model parameters based on the limited new sample to obtain the final model. The idea of MAML is simple but very efficient. It is a representative work in meta-learning based on optimization.

Latent Embedding Optimization (LEO) [31] is an improvement on MAML, which proposes parameter updates in a low-dimensional latent space. LEO uses encoders and relational networks to map data to a low-dimensional latent representation space to obtain latent vectors, and then decodes latent vectors to obtain high-dimensional model parameters for subsequent loss calculations. The inner loop of LEO updates the parameters of the hidden vector, and the outer loop updates the parameters of the encoder, relationship network, and decoder. In

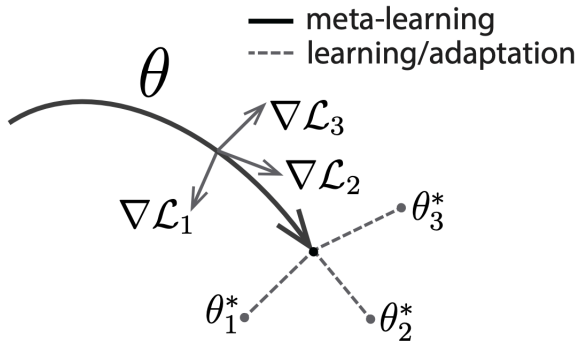


Figure 2.3: Demonstration of how Model-agnostic Meta-Learning determines the direction of gradient updates. Adapted from [76]

addition, the initialization parameters of new tasks in LEO are sampled from the conditional probability distribution related to the task, which is different from the random initialization in MAML. LEO indirectly updates model parameters through low-dimensional space mapping, decoupling gradient calculations from high-dimensional parameters of the model, which can capture uncertainty in the data and perform parameter adaptation more effectively.

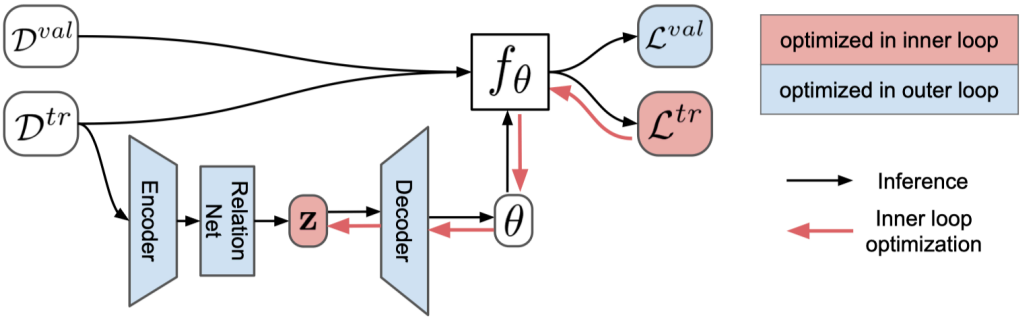


Figure 2.4: The process of Latent Embedding Optimization (LEO). Adapted from [31]

Reptile [77] is a first-order meta-learning method. Specifically, Reptile first initializes the parameters of the model, and then performs N iterations. In each iteration, a task is randomly sampled, and then the weight vector of the corresponding loss of the task is calculated, and then the Adam or SGD optimizer is used to calculate the parameters after K times of gradient descent. Finally, the current parameters are updated to the parameters after k updates and the difference of the previous dimension parameter plus the previous dimension parameter. Unlike MAML, Reptile does not perform second-order derivation and has achieved better results than MAML on multiple data sets.

Meta-Learner LSTM [78] believes that gradient descent will fail when the sample size is small,

so it is proposed to use LSTM to update the parameters of the classifier to simulate gradient descent. In the meta-training phase, the parameters of both the LSTM and the classifier will be updated. They each have a set of meta-parameters. The LSTM is trained to update the parameters of the classifier. In the meta-testing phase, the parameters of the classifier are fine-tuned through LSTM based on a small number of new samples.

MetaOptNet [79] proposes that discriminative linear predictors can provide better generalization capabilities for small sample learning tasks, and its goal is to learn feature embeddings for new classes with good generalization capabilities. MetaOptNet takes advantage of the two properties of the implicit differentiation (KKT condition) of the optimization conditions of convex problems in linear classifiers and the dual formula of the optimization problem. Specifically, MetaOptNet uses linear support vector machines as classifiers instead of nearest neighbor classifiers to solve the optimization problem through differentiable quadratic programming. The network diagram is shown in Figure 2.5, which shows a 3-way 1-shot classification task. The goal of meta-learning is to learn the parameters of the feature extractor that can generalize between different tasks. MetaOptNet found that regularized linear classifiers allow higher-dimensional feature embedding to reduce overfitting, but at the same time have a higher computational cost than nearest neighbor classifiers.

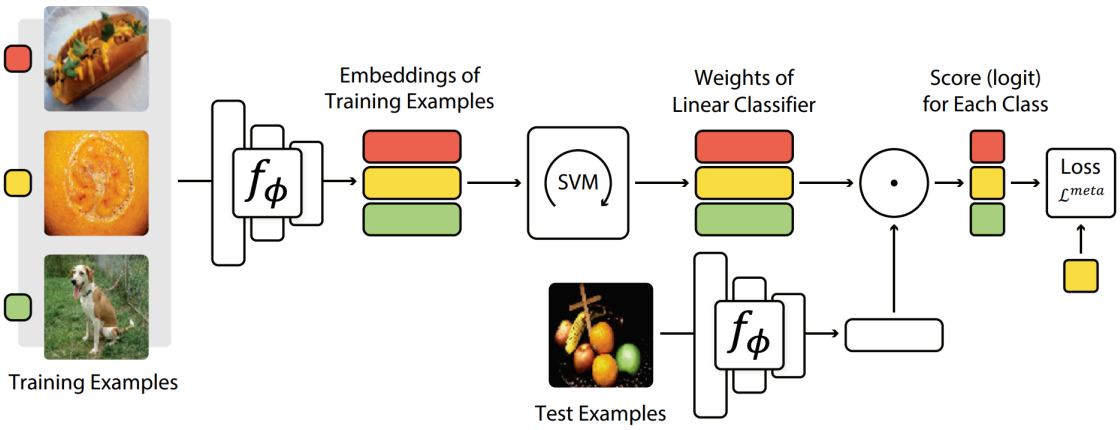


Figure 2.5: MetaOptNet. Adapted from [79]

2.4.2 Deep Metric Learning for Few-Shot Learning

The main idea of deep metric learning is to find a distance or similarity measurement between pairs from support set(base set) and query set(novel set). Early works mainly focus on learning discriminative feature embeddings for each category, which designs a task-agnostic model that

is powerful enough to extract discriminative features and can generalize well to novel classes.

The Siamese Neural Network comprises two identical twin networks, which are jointly trained to learn the relationship between pairs of input data samples. These twin networks share the same weights and network parameters, effectively representing a single embedding network. This embedding network is responsible for learning a compact and informative representation that captures the relationship between pairs of data points. Koch, et al [80] introduced a technique utilizing the siamese neural network for one-shot image classification. Initially, the siamese network is trained on a verification task, determining whether two input images belong to the same class. It outputs the probability of two images sharing the same class. During testing, the siamese network evaluates all pairs of images between a test image and every image in the support set. The ultimate prediction is based on the class of the support image with the highest probability.

In contrast to the Siamese Neural Network, the Relation Network [81] does not rely on a simple L1 distance in the feature space to capture relationships. Instead, it predicts the relationship between pairs of inputs using a CNN classifier. The relation score between a pair of inputs is obtained by concatenating their features and feeding them to a relation module to learn latent relationships. Unlike binary classification, the objective function in Relation Network is Mean Squared Error (MSE) loss. This choice reflects the focus of Relation Network on predicting relation scores, akin to regression, rather than binary classification.

Matching Networks, introduced by Vinyals et al. [82], are a type of few-shot learning model designed to learn from small datasets with limited labeled examples per class. The key idea behind Matching Networks is to utilize a flexible attention mechanism to effectively compare and classify new instances based on their similarity to a support set of labeled examples. A bidirectional Long Short-Term Memory (LSTM) network is used for support set images. LSTMs can process sequences, enabling the network to consider not only an individual image but also its relationship with other images within the support set. This can be particularly beneficial in few-shot learning where understanding the context of the limited data is crucial. For query image, a regular LSTM network with an attention mechanism is used for query images. The attention mechanism allows the model to focus on specific regions of the query image that are most relevant to the support set, aiding in classification.

To effectively recognize an image, the visual characteristics of specific regions typically play a

more significant role in feature representation. Consequently, various approaches aim to enhance feature embedding methods by emphasizing local information within regions of interest. For instance, D2N4 [83] addresses this challenge by combining the strengths of deep learning with a nearest neighbor classification strategy. Instead of using a single, global image representation, D2N4 extracts multiple local descriptors from an image. The classification process operates at the image-to-class level. Local descriptors from support images belonging to the same class are pooled together. For each query image's local descriptor, K-Nearest Neighbors (KNNs) are identified within each class pool. The total distance between the query image and a specific class is then calculated by summing the distances between the query's local descriptors and their corresponding KNNs in that class pool. This method has proven particularly effective on datasets involving fine-grained classification tasks, where distinguishing subtle differences between similar objects is crucial.

Learning generalized class features from only a few images poses a significant challenge for training effective classification models. LGM-Net [84] addresses this challenge by introducing a meta-learning framework that learns to "generate" matching networks suitable for specific few-shot classification tasks. The network comprises two key modules: i) TargetNet: This module serves as the actual classifier for the unseen few-shot task. It takes query and support set images as input and outputs class probabilities. ii) MetaNet: This module plays a crucial role in meta-learning. It takes training data containing multiple few-shot classification episodes as input. Each episode consists of a support set and a query set for a specific class. The MetaNet learns to generate the weights (parameters) for the TargetNet based on the information from these training episodes. Essentially, the MetaNet learns a transferable representation that allows it to adapt the TargetNet to new, unseen few-shot tasks efficiently.

Metric learning-based methods have proven effective in addressing data scarcity challenges. However, these methods can be computationally intensive as early approaches like Matching Networks directly comparing similarities between support and query high dimentional features. To reduce the computational costs, researchers have proposed using class prototypes, which offer compact and representative embeddings of class information, facilitating the classification of new instances with limited training data. researchers have proposed using class prototypes, which offer compact and representative embeddings of class information, facilitating the classification of new instances with limited training data.

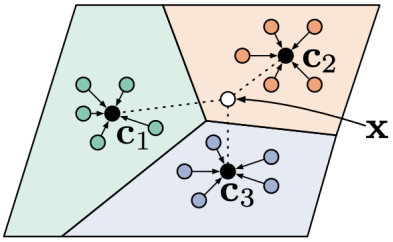


Figure 2.6: Prototypical Network computes mean embeddings of support images as class prototypes. Adapted from [85]

A milestone method, called Prototypical Network [85], provide a powerful and efficient approach for few-shot learning. By leveraging representative prototypes and efficient similarity measures, it enables the model to learn effectively from limited labeled data. For each class, the network averages the feature representations of all the images in its support set. This average embedding serves as the class prototype, encapsulating the key characteristics of that class within the feature space. When presented with a new, unseen image (query image), the network extracts its features using the same feature backbone. The network then calculates the distance between the query image's features and the prototype of each class. The class associated with the closest prototype (smallest distance) is considered the most likely class for the query image.

The Relation Network [81] is the first work of introducing a neural network to model the similarity of feature embeddings in few-shot learning. It consists of an embedding module and a relation module. The embedding module builds on convolutional blocks for mapping original images into an embedding space, and the relation module consists of two convolutional blocks and two fully-connected layers for computing the similarity between each pair of support and query images. The learnable similarity measure enhances the model flexibility

Unlike traditional metric learning, semantic alignment metric learning (SAML) [86] focuses on ensuring that the learned metric reflects the semantic relationships between different data samples. As shown in Figure 2.7, two key modules work in tandem: the feature embedding module and the semantic alignment module. The semantic alignment module is responsible for ensuring that the learned metric aligns well with semantic concepts present in the data. Following the computation of the relation matrix, it is passed through a Multilayer Perceptron (MLP) network. The MLP network processes the relation matrix and produces a similarity score that reflects the semantic similarity between the query instance and the support class. This score represents how well the query instance aligns with the semantic characteristics of

the support class.

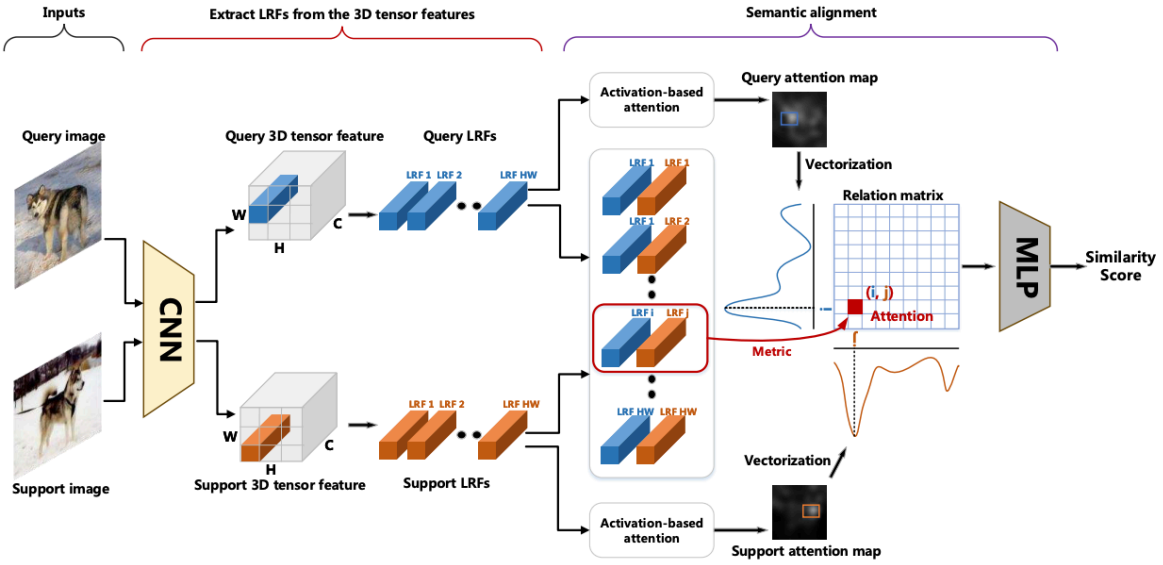


Figure 2.7: Semantic Alignment Metric Learning SAML. Adapted from [86]

More recently, Transformer has shown great potential in FSL. CrossTransformer [87] first uses self-supervised SimCLR [88] technology to enhance the discriminability of learned features, and then builds a Transformer-based network to achieve local feature alignment. Specifically, SimCLR is executed on 50% of the meta-tasks, data enhancement is performed on both the support set and the query set samples, and optimization is performed according to the loss function in SimCLR. Instance-level discriminative information is learned through SimCLR and is robust to transformations such as cropping and color. CrossTransformer is improved based on ProtoNet, but unlike ProtoNet, which calculates the mean value for each category to obtain the category center, CrossTransformer wants to learn a category center specific to the query set sample. Construct (q, k, v) triples to calculate attention, where q is the query set sample used, k and v are support sets, and the attention map is obtained. The attention map is used to weight the feature values of different images in the support set, and the features of similar support set samples are added to obtain the prototype. Finally, after the query set samples are mapped to the same dimensions as the prototypes, the distance between the query set samples and each prototype is directly compared for classification.

2.4.3 Data Augmentation in Few-shot Learning

Data augmentation has been shown effective for few-shot learning, it can avoid catastrophic forgetting by reducing data scarcity. Current approaches can be roughly divided into two categories: image-level and feature-level.

Standard image-level augmentation techniques such as flipping, rotating, transforming, adding Gaussian noise, and so on are limited by the data itself and thus may be tough to obtain diverse generations. MetaGAN [89] is a pioneering work that uses GANs to generate fake data in order to force the classifier to learn sharper decision boundaries. IDeMe-Net [90] proposes to deform images by linearly fusing image patches from probe and gallery images. Specifically, each image is divided into nine patches and a deformation network is used to estimate the fusion weight of each patch. By generating images via patch combination, semantic information can be better preserved while sharpening the decision boundaries of the classifier.

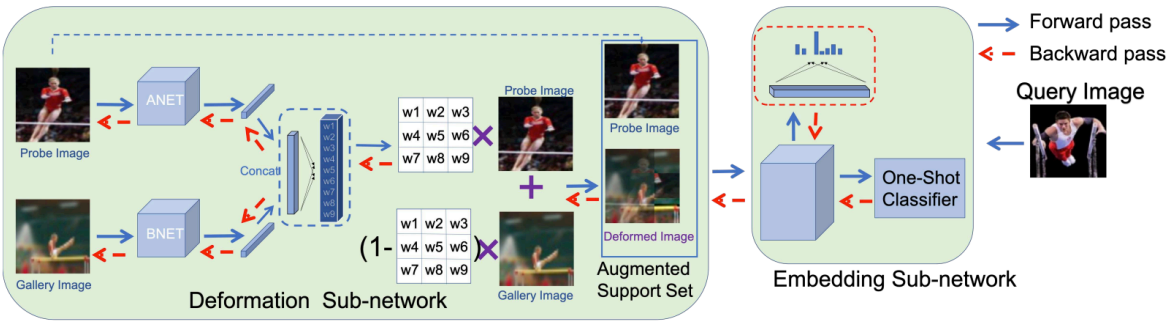


Figure 2.8: Image deformation meta-networks (IDeMe-Net). Adapted from [90]

However, directly generating high-quality new images remains a significant challenge. Denoising diffusion probabilistic models (DDPM) have recently emerged as powerful generators in the few-shot scenario. Based on limited samples, it learns to create synthetic images that share characteristics with the real examples from the limited dataset (support set). These generated images essentially act as artificial data points, expanding the training set for the specific class.

2.5 Few-shot semantic segmentation

The few-shot semantic image segmentation (FSS) develops from few-shot classification, which aims to accurately segment objects in an image given only a limited number of annotated examples per class. The core problem revolves around the model's ability to learn and generalize

from a few examples, typically referred to as "shots", to perform segmentation tasks on unseen images effectively. Unlike traditional machine learning methods that rely on extensive manual annotation of training data, few-shot semantic segmentation tackles the problem of scarce data availability by learning to achieve class-agnostic and generalization from a small set of annotated images to segment previously unseen classes or objects.

As illustrated in Figure 2.9, a FSS dataset is split into two distinct subsets: meta-training and meta-testing. Both subsets are comprised of "support images" and "query images.". Support images are a set of labeled images provided as examples for each class involved in the few-shot learning task. Each support image is accompanied by its corresponding segmentation mask, which precisely indicates the object or region of interest within the image. Support images are used during the training phase to adapt the model to recognize and segment new objects based on the provided examples. Conversely, a query image is an unlabeled image that the model is tasked to segment. Unlike support images, the query image does not come with a corresponding segmentation mask. The challenge for the model is to apply what it has learned from the support images to accurately segment the query image. When the task involves using K support images along with their corresponding masks for learning, this setup is referred to as K-shot semantic segmentation. The remainder of this section goes through the problem setting and typical FSS methods so far.

2.5.1 Datasets setting

PASCAL-5^i Most of the existing works on few-shot semantic image segmentation adopt the partition scheme proposed in OSLSM (One-Shot Learning for Semantic Segmentation) [33]. The dataset used in these works is based on PASCAL-5^i, an extension of the PASCAL VOC dataset combined with the SDS (Semantic Boundaries Dataset). The PASCAL VOC dataset consists of 20 object classes. For the few-shot segmentation task, five classes are sampled and considered as the test label-set and the remaining 15 classes data are utilized as the training set D_{train}. To ensure consistency, the masks in D_{test} are carefully chosen to ensure that classes not belonging to D_{test} are labeled as the background class.

COCO-20^i. This dataset is derived from the MSCOCO [45] dataset, which comprises 80 object classes and ground-truth segmentation masks of relatively lower quality compared to those in PASCAL VOC. Similar to PASCAL-5^i, the COCO-20^i dataset is also divided into four folds for

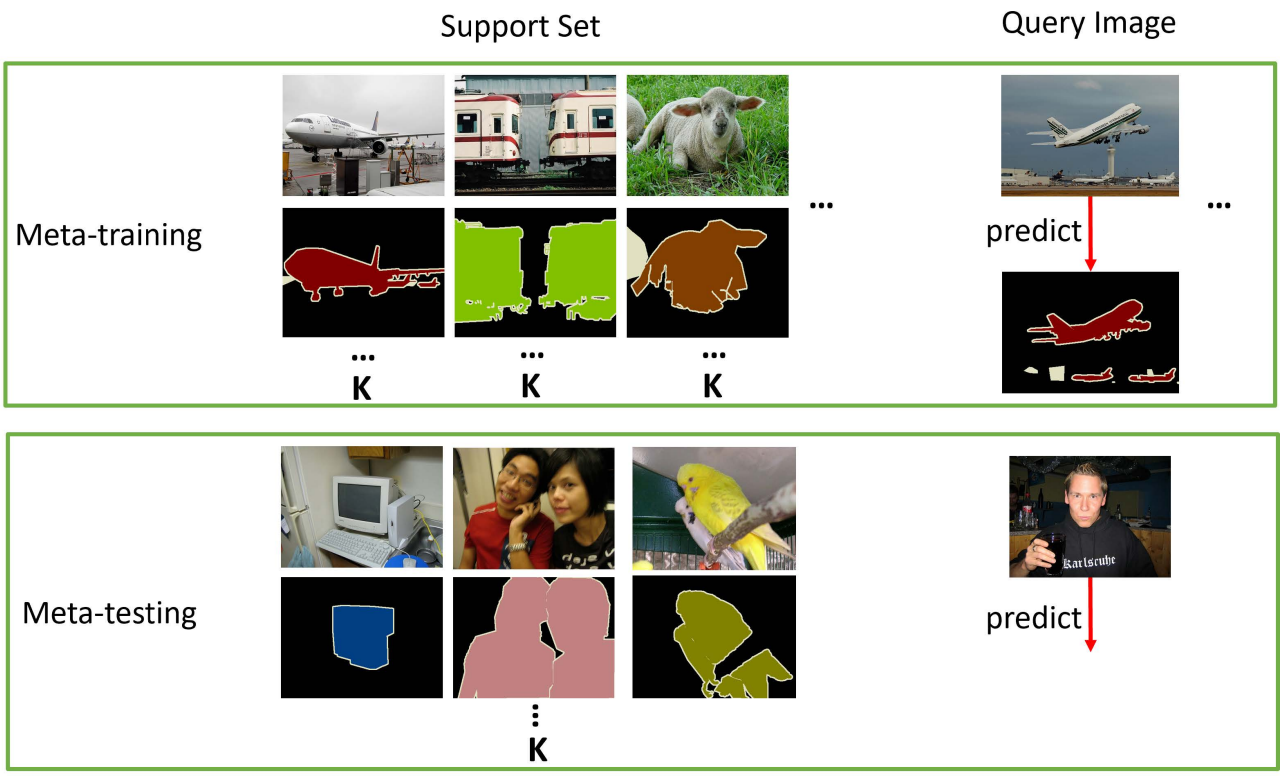


Figure 2.9: Illustration of Few-shot Semantic Segmentation problem

cross-validation purposes. For each fold, denoted as {fold0, fold1, fold2, fold3}, the COCO-20^i dataset samples 20 object classes as test classes from the pool of 80 classes in MSCOCO. The remaining 60 classes are consisted of the training set.

FSS-1000 In contrast to previous general datasets, FSS-1000 is specifically tailored for training models in few-shot segmentation tasks. It comprises 1000 object classes, with each class containing only 10 images. Adhering to the original FSS setting as outlined in [48], there is no need for cross-validation to enhance the diversity of test classes. The dataset is partitioned into train/validation/test sets with a distribution of 520/240/240 classes, respectively, ensuring that all classes are disjoint from one another.

Table 2.1 and Table 2.2 reprots the detailed few shot semantic segmentation splits of PASCAL-VOC and COCO datasets, respectively.

2.5.2 Few-shot semantic segmentation methods

The OSLSM (One-Shot Learning for Semantic Segmentation) method is a pioneering approach that introduces the few-shot semantic segmentation challenge and proposes a base structure

Sub Dataset	Test classes
PASCAL-5^0	aeroplane, bicycle, bird, boat, bottle
PASCAL-5^1	bus, car, cat, chair, cow
PASCAL-5^2	dining table, dog, horse, motorbike, person
PASCAL-5^3	potted plant, sheep, sofa, train, tv/monitor

Table 2.1: Few shot semantic segmentation split of PASCAL dataset

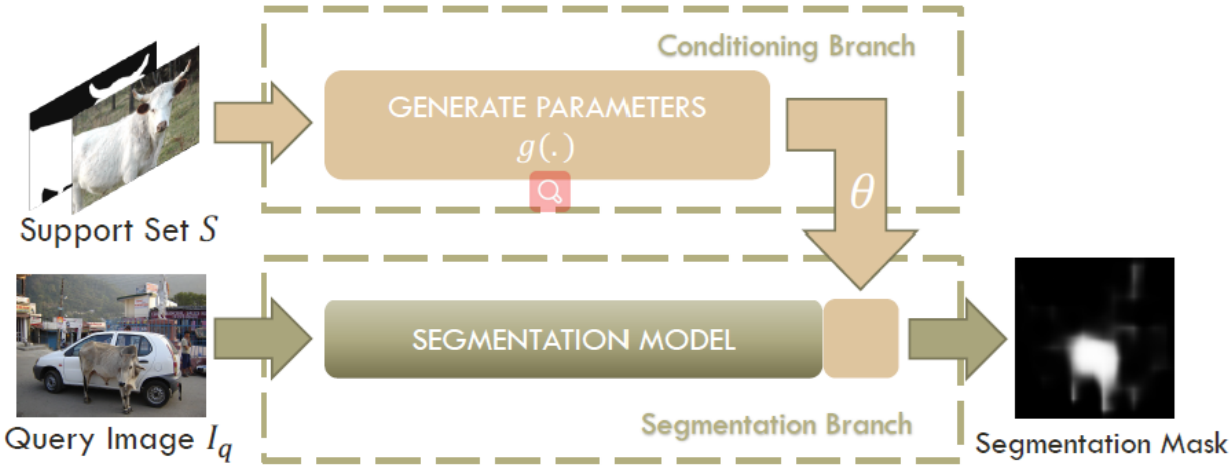


Figure 2.10: Overview of the FSS meta framework

for addressing this task. Drawing inspiration from few-shot learning strategies, the OSLSM method employs a two-branch architecture for support processing and query processing, respectively. The architecture is visually depicted in 2.10. In the OSLSM method, the upper branch, known as the Conditioning branch or Support branch, processes the support label-image pairs to generates a series of conditional features (prototypes) as reference information. These prototypes play a crucial role in guiding the segmentation process for the new class. The button branch, referred to as the Segmentation branch, uses both the computed prototypes from the support branch and an image of novel classes (query image) to produces a segmentation mask for the new class as the prediction. Unlike fine-tuning approaches commonly used in few-shot learning, which often applies multiple iterations of stochastic gradient descent (SGD) to learn optimize the network until convergence, the OSLSM method computes the prototypes of the Conditioning branch in one inference. This brings several advantages to the approach. Firstly, the single forward pass enables fast computation, enhancing efficiency. Secondly, the differentiable nature of the method allows for joint training of the Conditioning branch and the support branch within the model. Lastly, the number of guidance prototypes, denoted as θ, is

COCO-5^0	**COCO-5^1**	**COCO-5^2**	**COCO-5^3**
1. Person	2. Bicycle	3. Car	4. Motorcycle
5. Airplane	6. Bus	7. Train	8. Truck
9. Boat	10. Traffic light	11. Fire hydrant	12. Stop sign
13. Parking meter	14. Bench	15. Bird	16. Cat
17. Dog	18. Horse	19. Sheep	20. Cow
21. Elephant	22. Bear	23. Zebra	24. Giraffe
25. Backpack	26. Umbrella	27. Handbag	28. Tie
29. Suitcase	30. Frisbee	31. Skis	32. Snowboard
33. Sports ball	34. Kite	35. Baseball bat	36. Baseball glove
37. Skateboard	38. Surfboard	39. Tennis racket	40. Bottle
41. Wine glass	42. Cup	43. Fork	44. Knife
45. Spoon	46. Bowl	47. Banana	48. Apple
49. Sandwich	50. Orange	51. Broccoli	52. Carrot
53. Hot dog	54. Pizza	55. Donut	56. Cake
57. Chair	58. Couch	59. Potted plant	60. Bed
61. Dining table	62. Toilet	63. TV	64. Laptop
65. Mouse	66. Remote	67. Keyboard	68. Cell phone
69. Microwave	70. Oven	71. Toaster	72. Sink
73. Refrigerator	74. Book	75. Clock	76. Vase
77. Scissors	78. Teddy bear	79. Hair drier	80. Toothbrush

Table 2.2: Few shot semantic segmentation split of COCO dataset

independent of the image resolution, ensuring scalability and avoiding any issues related to the size of the images. This property allows the method to handle images of varying resolutions efficiently and effectively.

In order to boost the performance of few-shot semantic image segmentation, several optimizations have been proposed that target the architecture components of the method. These components include guidance feature extraction, which aims to extract relevant features from both labeled support images; segmentation feature embedding, which aims to embed the features of the unlabeled query images into a common space; and guidance strategies, which aim to guide

the dense prediction challenge using the extracted and embedded support features. Based on this support-query architecture, FSS methods can be categoried as:

1) Metric Learning-Based Approaches. Metric learning is crucial in addressing FSS challenges, with prototype network-based methods [34]–[36], [39], [91] leading the field. These methods aim to learn a general distance function that assigns higher affinity scores to similar features and lower scores to distinct ones, regardless of the category. Unlike traditional learning methods [24], [92], [93], which generate a class prototype as a rough optimal estimate, few-shot techniques focus on creating class-specific prototypes. These prototypes, even if not optimal, are effective if they convey object information and align query features with similar semantic classes. Recognizing the limitations of representing a category with a single prototype vector, some strategies [37], [94] aim to produce multiple prototypes per class. Furthermore, some methods [95], [96] explore direct element-level dense matching between support and query features as an innovative solution. These varied approaches are outlined below.

Early attempts [34], [35] focused on feature matching using a singular class descriptor. The innovative approach in [33] sparked subsequent studies like [35], which accomplished binary segmentation by measuring cosine distances between query feature vectors and class-specific prototypes. To optimize support data usage, query samples and their predicted masks were treated as additional support, enhancing the segmentation of original support samples.

Diverging from the fixed distance functions in [35], other methods [36], [39] employed learnable neural networks, acting as adaptable distance metrics to gauge support and query feature similarities. These methods merged target class support cues with query features, decoding the amalgamated features for final segmentation. A prevalent fusion technique involved appending query features with scaled prototypes [34], [39] or support feature maps across the channel dimension. This process incorporated multi-scale features [36] and multi-class labels [91] to refine query sample representations.

Beyond simple channel-wise concatenation [34], [39], alternative integration methods like element-level addition [97], attention map re-weighting [98], and similarity guidance [99], [100] were explored to merge support and query features effectively.

In the realm of dense matching, several innovative strategies have been developed to enhance the interplay between support and query features: Pyramid Graph Network [101] introduces a network to meticulously map the dense connections between support and query features across

various scales, ensuring a comprehensive multi-scale correspondence. Democratic Attention Network [95] focuses on object-centric pixels, this network forges a resilient link between support and query images, prioritizing areas with object presence for enhanced correspondence. Cross-Attention Mechanism [102] is also employed to assimilaterelevant pixel-level features from support images into query images, enriching the feature landscape of the query images. Bipartite Graph with Graph Attention [103] Constructs a bipartite graph and integrates a graph attention mechanism and a weight adjustment strategy to draw more target-object pixels into the segmentation process of query images, enhancing the focus on relevant areas. By analysing the dense foreground-background correlations, [104] delves into the dense relationships between foreground and background, addressing the loss of information typically encountered in prototype learning and dense foreground feature matching. Each approach contributes uniquely to refining dense matching, focusing on enhancing the precision and relevance of feature correlations for more accurate segmentation outcomes.

2) Parameter Prediction-based Methods. Unlike metric learning-based methods that focus on learning a powerful predictor transferable across tasks, parameter prediction-based methods aim to create a unique predictor for each task. This is accomplished by devising a parameter generator responsible for predicting the neural weights of the prediction layer.

A notable approach [105] utilize the modified logistic regression layer, pixel-wise semantic labels are derived from the query features. This approach expands beyond support samples, incorporating query images in classifier weight generation. Instead of directly substituting classifier parameters, another method dynamically adds weights to the classifier, enabling the model to proficiently handle both base and novel categories [106]. This approach offers a more flexible and adaptive way to accommodate a diverse range of classes without compromising the model's performance on previously learned categories.

3) Finetune-based methods for few-shot image semantic segmentation focus on utilizing optimization algorithms to adjust the parameters of a pre-trained segmentation network, facilitating the learning of new, unseen categories. The process involves iteratively refining the network by reducing the discrepancy between support predictions and their corresponding masks [107]. This method of parameter refinement significantly mitigates performance drops caused by differences in class characteristics between training (offline) and application (online) phases.

To enhance this process, an architecture combining an embedding network with a differentiable linear classification model was introduced [108]. This configuration allows for more effective updates to the linear classifier's parameters while maintaining the embedding network's ability to generalize across various classes. Contrary to the episodic training used in [107] and [108], a transductive inference approach was employed, leveraging standard supervised learning to develop a feature extractor for base classes [109]. During inference, a linear classifier is fine-tuned by minimizing a loss function that considers both labeled support images and the statistical properties of unlabeled query images.

In addition to adapting across categories, these methods also address the distribution shift between training and inference data, enhancing the model's applicability and robustness in real-world scenarios.

2.6 Few-Shot Class Incremental learning

Few-Shot Class Incremental Learning (FSCIL) is an emerging and challenging area in machine learning, which extends both few-shot learning and classical incremental learning paradigm. The core objective of FSCIL is to enable a model to learn new classes with very few examples while retaining its performance on previously learned classes, a concept also known as "learning without forgetting." This is particularly important in real-world scenarios where data can come in streams, and it is impractical to retain all the data or continually retrain the model from scratch. On top of that, this thesis aims to enhance the applicability of few-shot segmentation methods by tackling the difficult and practical setting of few-shot semantic segmentation.

There are three key characteristics of FSCIL: 1) Class Incrementality, the model must adapt to new classes that weren't part of the original training set without needing to retrain from scratch. 2) Data Scarcity, only a few examples of the new classes are available for training, making it a challenging problem compared to traditional machine learning settings where ample data is typically available. 3) Catastrophic Forgetting, a significant challenge in FSCIL is avoiding catastrophic forgetting of old knowledge when adapting to new classes.

The methods for FSCIL can be summarized into the following categories:

1) Meta Learning-Based Methods. This category involves methods that use meta-learning principles to leverage prior knowledge and learning experiences to facilitate quick adaptation

to new tasks and classes with limited data. Like FSL, one of the widely used technique should be prototype learning, which learns a overall representation for novel classes given a few examples, and then leverage the similarity between the testing query and the examples to perform classification or other visual tasks. However, using conventional prototype-based methods to combine all the learned class prototypes may make some prototypes hard to tell apart. To address this issue, [110] introduced a meta-learning class structure. The core idea is to learn a notion of how classes should be distributed in the embedding space. This is achieved through a meta-learning process where the model is trained on simulated incremental learning tasks. These tasks involve learning to distinguish between a small set of new classes and previously learned ones. The model learns to align each class with the class structure by moving it along the base vectors of the subspace using an alignment kernel. This ensures that learned classes are distinctive from each other within and across different learning sessions.

Beyond existing meta-learning methods, researchers are exploring ways to explicitly construct the meta-learning process itself. This approach focuses on incorporating the trade-off between adapting to new knowledge and retaining knowledge of previously learned classes.

One such method is MetaFSCIL [111], taking inspiration by the multi-task learning approach MAXL [27]. MetaFSCIL formulates the challenge as a meta-objective, directly aiming to balance adapting to new knowledge while retaining knowledge of previously learned classes. It achieves this by mimicking the meta-testing scenario during training, where the model is exposed to a sequence of incremental tasks drawn from the base classes. Additionally, MetaFSCIL introduces a bidirectional guided modulation mechanism that leverages meta-learning to automatically adapt to new knowledge.

Another line of research by Zou et al. [112] focuses on the specific challenge of class-level overfitting that occurs when the model prioritizes easily learned patterns within a class during training, neglecting the need for margins to separate different classes effectively. The method utilizes a margin-based approach to handle the uncertainty associated with class boundaries during incremental learning. It introduces a margin-based loss function that penalizes predictions made with low confidence, aiming to improve the generalization performance of the model across both base and novel classes. During incremental learning, the model leverages a support set of examples from the base classes to adapt to new classes. By enforcing a margin-based constraint on the feature space, the model learns to distinguish between classes effectively while

minimizing the risk of overfitting to specific class instances.

2) Feature Space-Based Methods. Subspace representation is instrumental in improving algorithm efficiency by reducing the dimensionality of data while preserving critical features. In FSCIL, leveraging subspace representation involves mapping new classes from their original space into a low-dimensional space defined by old classes. This facilitates better adaptation to new classes.

In the study of cite[41], a novel method that operates on mixture of subspaces which allows the model to dynamically allocate resources to different subspaces based on their relevance to the current task. It leverages synthesized features to address the challenge of learning new classes incrementally while avoiding catastrophic forgetting. At its core, the method represents each class as a subspace in a high-dimensional feature space. During the incremental learning process, the model synthesizes new features by combining existing ones in a principled manner, enabling it to adapt to new classes without forgetting previously learned ones.

Akyürek et al. [113] presented a subspace regularization approach, prompting the weight representation of novel classes to align with the subspace formed by existing old-class weights. This regularization, user-friendly and simple, also allows for integrating additional prior information. Viewing from the parameter feature space angle, [114] devised WaRP, amalgamating F2M's [115] proficiency in locating flat loss function minimums with FSLL's [116] expertise in parameter fine-tuning, creating a robust model for class adaptation.

3) Replay-Based Methods. These methodologies leverage the principle of revisiting previously acquired knowledge when faced with new tasks. These approaches encompass direct replay [40], [117], which involves storing and reusing real samples from previous tasks, and generative replay [32], [118], where a generative model is utilized to replicate the distribution of data from previous tasks.

[117] proposed a three-part framework where the initial two stages independently train the network on base and novel classes, employing a model parameter constraint technique to maintain memory of old classes. The final stage utilizes a limited collection of retained samples for replay and fine-tuning of the classifier's proficiency across all classes, encompassing both base and novel categories. On the other hand, [40] introduced a semantic-aware knowledge distillation approach, preserving a select set of samples from prior classes. This method leverages word embeddings as supplementary data and translates images into vector space, demonstrating the

effectiveness of knowledge distillation within the context of FSCIL

In FSCIL, storing real past data (old data) might raise privacy concerns. To address this, [32] suggests a method for creating substitute "old samples" without requiring real data. This method relies on a generator network that creates uncertain examples close to decision boundaries. This approach is necessary because traditional data replay techniques used in continual learning are not suitable for FSCIL due to the limited amount of data available.

[118] introduces Few-Shot Incremental Learning Generative Adversarial Network (FSIL-GAN), capable of replicating the real data distribution with a limited amount of data. This is achieved by aligning synthetic visual features, extracted from generated images, with their corresponding semantic representations. The method ensures diversity and separability of the synthetic features to prevent the model from becoming fixated on a limited set of generated examples.

4) Dynamic Network Structure-Based Methods. These approaches involve network architectures that can dynamically adjust during runtime based on input data features. They are designed to possess strong generalization capabilities and mitigate the risks of overfitting. For instance, [42] introduced the Neural Gas (NG) network, which captures the topological layout of the feature space across various categories, enhancing knowledge representation. The framework ensures the stability of the NG's topology, safeguarding against forgetting old categories. It allows the NG network to dynamically expand, accommodating new samples and refining the representation of few-shot new classes.

Subsequently, [43] developed the Learnable Expansion-and-Compression Network (LEC-Net), which enhances feature representation by strategically expanding network nodes while mitigating feature drift through model regularization. Building on this, they introduced the Dynamic Support Network (DSN) [119], a network capable of adaptive expansion. DSN employs compressive network expansion to enhance feature representation with each incremental task and dynamically tailors the feature space according to the old class distribution.

More recently, [120] investigated a masking-based approach within network structures. It employs non-binary masks to forge soft subnetworks from the primary network, striking a balance between mitigating forgetting and preventing overfitting. During the base classes phase, the model learns soft-subnetwork parameters and weight scores. In the incremental learning phase, it updates select parameters of the subnetwork, ensuring a seamless transition and learning continuity.

2.7 Summary

This chapter comprehensively explores the evolving landscape of few-shot semantic segmentation, delving into methods ranging from supervised semantic segmentation to few-shot incremental learning. Supervised semantic segmentation, serving as the foundation, utilizes extensive labeled data to train models for pixel-level classification. Techniques like fully convolutional networks (FCNs) and encoder-decoder architectures have set benchmarks in this domain. However, the dependency on vast annotated datasets limits their applicability in scenarios where labeling is expensive or impractical.

Transitioning to few-shot learning, the review highlights methods that adapt to new tasks or classes with minimal data. Meta-learning, or learning to learn, stands out by training models to quickly adapt to new tasks using small datasets, effectively addressing the data scarcity issue. Few-shot segmentation, a subset of few-shot learning, specifically targets the challenge of segmenting images with few labeled examples. Techniques like prototype learning and metric learning redefine feature space, enabling models to generalize from limited data by comparing new instances with a learned metric or reference points.

The review further scrutinizes few-shot incremental learning (FSCIL), where the model continuously learns new classes without forgetting the previously learned ones. It underscores the complexity of balancing the acquisition of new knowledge with the retention of previously learned information, a phenomenon known as catastrophic forgetting. Strategies like replay, where the model retrains on a mix of old and new data, and meta-learning approaches that optimize the model's ability to learn new tasks while retaining old knowledge, are pivotal. The review also delves into advanced techniques like parameter prediction-based methods, which dynamically adapt model parameters for new classes, and representation learning, which seeks to encode data into a form where classes are inherently separable.

In conclusion, this review encapsulates the trajectory of few-shot semantic segmentation, highlighting the shift from data-intensive supervised methods to innovative few-shot techniques. It underscores the significance of adaptability and knowledge retention in model evolution, laying the foundation and conceptual framework for the thesis work on proficient few-shot learning methods in image semantic segmentation.

Chapter 3

Masked Cross-image Encoding for Few-shot Segmentation

3.1 Introduction

As discussed in Chapter 2, semantic segmentation tackles the challenge of pixel-wise classification, aiming to assign a class label to every pixel within an image. This task traditionally demands a substantial volume of meticulously labeled images for effective supervised learning. To address this challenge and reduce the annotation workload, few-shot semantic segmentation (FSS) methods [33], [37], [94] have been proposed to learn segmenting previously unseen classes with only one or a few labeled training images.

FSS learning methods generally follow the "learning to learn" paradigm, also known as meta-learning [30], [121], to obtain generalized prototypes that can describe the classes of interest. These methods typically apply a two-branch framework [33], [99], [122] with a shared frozen pre-trained backbone, where a support branch is used to learn the object prototypes from very few labeled images (support images) and a query branch is utilized to make predictions for a query image on conditioned of the support prototypes. Under this framework, masked average pooling (MAP) [35], [99], [122] and clustering-based aggregation methods [94], [123], [124] are two popular approaches to learning the representations for support images. While both prototype-based and metric-based approaches have shown promise for few-shot segmentation (FSS), they primarily focus on capturing the overall visual properties of a class and do not effectively incorporate contextual information from a spatial perspective. This can result in

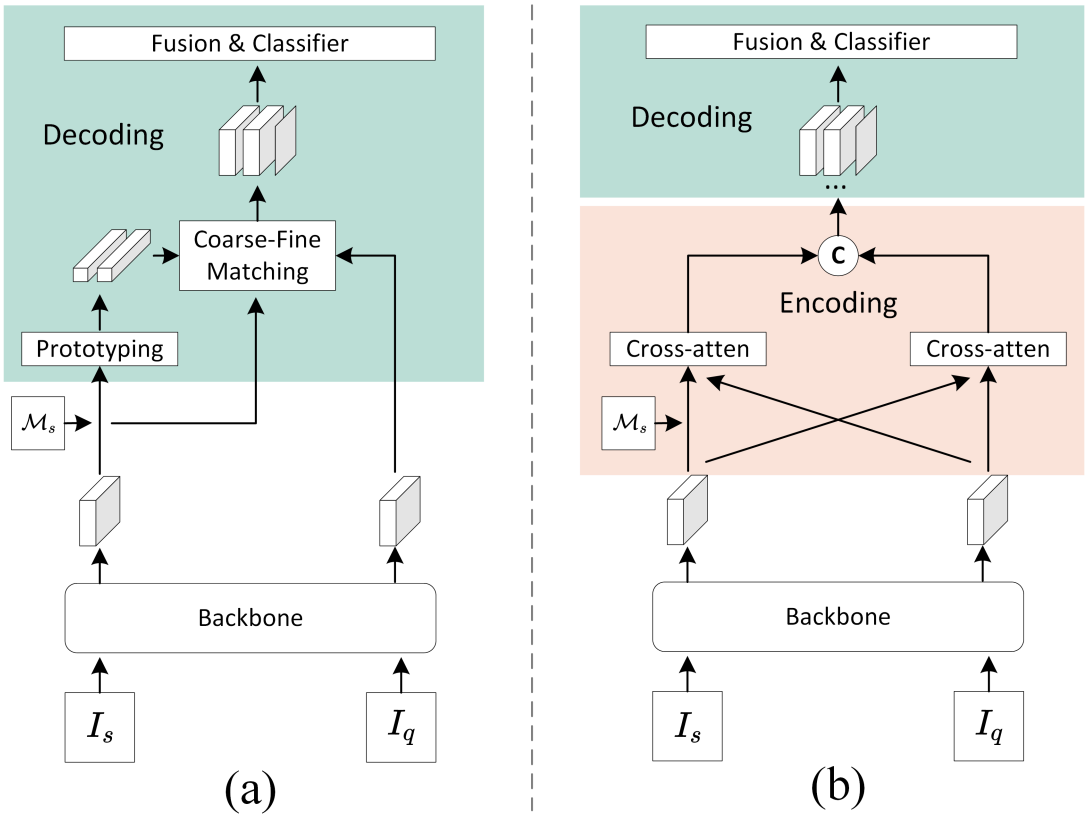


Figure 3.1: Comparison between conventional "fixed feature + decoder optimization" FSS methods and the proposed learnable cross-image feature scheme. (a) Current FSS methods independently learn object prototypes from support features and conduct coarse prototype matching or fine "pixel-wise" feature guidance at the decoding stage. (b) The proposed joint encoding scheme makes it possible to learn extra object context from other images before decoding.

a failure to guide features that correspond to the local context of support images, leading to decreased segmentation performance. To overcome this limitation, alignment schemes for FSS employ "features matching" and "decoder optimization" strategies to learn the metric similarities between support and query features extracted from the backbone. Specifically, methods such as [36], [37], [124] calculate prior maps, or similarity metrics, based on the query and support features to define their correlations. Other methods, including [102], [125], [126], feed support and query features into a decoder (e.g., 4D convolution or transformer) to perform local region matching between query and support images. However, as support and query features are generated independently through a fixed backbone, these alignment methods can still suffer from underdeveloped contextual information mining for support-query pairs that

contain the same objects.

It is widely acknowledged that deep learning features extracted from different images of identical objects share similar representations compared to features of other objects. Humans are good at recognizing new things by comparing specific details (like shapes, textures, or patterns) with things they've seen before. This allows them to pick up on subtle differences and categorize the new object even if they've only seen a few examples from that category. On top of this observation, it is resonable and more effective to encode an image with features from others that contain the same object when only a limited number of samples are provided. This approach enables better exploration of the semantic representation among features across different image sources and preserves more detailed local contexts that aid in identifying subtle targets. Notably, the FSS two-branch framework naturally provides different image sources containing the same object, namely, the support images and the query image, whose visual features were extracted separately in prior methods. Building on this insight, we observe that self-attention in the Vision Transformer (ViT) [66] and cross-attention in [127] can be utilized to capture contextual information of images during token dependency construction. Consequently, we adopt a similar approach to consider joint learning between query and support features for FSS. Specifically, we model cross-image object semantic encoding to identify discriminative local regions, as illustrated in Figure 3.1 (b). CRNet [128] also presented a joint scheme. However, our method is focused on position-wise features, while CRNet encodes overall support-query image-wise representations. CRNet leverages global average pooling to extract overall statistics of the query-support features on a per-image basis and then fuses the resulting branch vectors via element-wise multiplication. In addition, since the module uses image-wise representation, it may miss important details, and element-wise vector multiplication only amplifies common features between the two feature maps. In contrast, our method employs multi-head attention to analyze position-wise features across query and support image features. This approach allows us to capture contextual information with fine-grained details and enhance mutual support-query interaction without any fusion operation. As a result, our method can provide a more comprehensive analysis of the features, leading to more accurate results.

The overall framework of the proposed method is illustrated in Figure 3.2. The method aims to capture the object semantic mutual relations across support and query images. Unlike CyCTR [102] which employs Transformer blocks to pass support features to the query decoder,

our approach emphasizes the importance of consistent mutual relations between query and support features. To achieve this, we propose a symmetric cross-attention structure called Masked Cross-Image Encoding (MCE), which is designed to assemble bidirectional inter-image relations on multi-level features. The MCE incorporates the support segmentation mask to restrict attention within the localized features of the target objects, thus enhancing the ability to distinguish objects from backgrounds. Additionally, we utilize multi-level features of the support-query images to calculate similarity score matrices, which provide a comprehensive understanding of the correspondence between each position of query features and the support object. Thus, these similarity score matrices help refine the pixel-wise classification accuracy in FSS.

We evaluated the meta-learning ability of our model on two public FSS benchmarks, PASCAL-5^i [44] and COCO-20^i [45]. The experimental results show that the proposed masked cross-attention encoding helps enrich query features by attending support object regions mutually and, therefore, obtains a strong meta-learning ability, surpassing the prior counterpart methods. In summary, the main contributions of this work are as follows:

- We propose a masked cross-image encoding method to discover shared visual representations of the target objects in support and query features. By using a symmetric cross-attention structure, MCE can attend to bidirectional inter-image relations on multi-level features, which not only enriches the query features with information from the support object regions but also enhances the support-query interaction, leading to a more favorable meta-learning capability for FSS.
- We performed comprehensive experiments to explore various designs of cross-attention schemes for FSS, aiming to identify the most effective scheme for the encoding module.
- We propose to calculate support-query similarity score matrices that reflect the likelihood of a pixel in query features belonging to the foreground. These matrices are then incorporated into our model along with multi-level cross-image features to facilitate final segmentation.
- Extensive experiments on PASCAL-5^i and COCO-20^i benchmarks under 1-shot and 5-shot settings demonstrate the effectiveness of the proposed MCE and similarity score matrices. The proposed model achieves superior meta-learning performance across all compared state-of-the-art methods.

3.2 Methodology

3.2.1 Problem Definition

We normally follow the definition of few-shot semantic image segmentation in [129]. In this scenario, a semantic segmentation dataset is split into a training set D_{train} and a testing set D_{test}. Unlike traditional semantic segmentation setups, there is no overlap between object classes in the training and testing sets, denoted as $\{C_{train}\} \cap \{C_{test}\} = \emptyset$. Following the two-branch framework, multiple episodic paradigm pairs are sampled from both sets, each comprising a support set $S = \{(I^s, M^s)\}_1^k$ and a query image pair $Q = (I^q, M^q)$ sharing the same class. If there are K annotated images in one support set, the target few-shot problem is called K-shot. Our objective is to learn a mapping $\mathcal{H}_\theta$ on the training set D_{train}, which can precisely predict the query image segmentation mask M^q from combined input (I^s, M^s, I^q). Note that both the support masks $\{M^s\}_1^k$ and query mask M^q are available during the training stage, while only the support masks are provided to perform segmentation during the testing stage.

3.2.2 Model Architecture

We formulate the overall model architecture in Figure 3.2. Initially, multi-level features are extracted from both the query image and support images using a pre-trained backbone network. Specifically, the support features extracted from the intermediate layers of the network, along with their corresponding masks, are used to derive a class-wise prototype vector $\mathbf{V_s}$ by Mask Average Pooling (MAP). The query features and support features extracted from the deep layer are utilized to calculate a similarity score matrix $\mathbf{A}_{sim}$. Simultaneously, all features from the query image and support images are exploited to compute the multi-level cross-image attention map $\boldsymbol{f}_{cross}$ through the Masked Cross-image Encoding (MCE) module. The cross-image encoding feature $\boldsymbol{f}_{cross}$, the prototype vector $\mathbf{V_s}$, and the similarity score matrix mask $\mathbf{A}_{sim}$, along with query feature $\boldsymbol{f}_l^Q$ are concatenated and then fed to the decoder, which is composed of an Atrous Pyramid Pooling (ASPP) module, a 3×3-convolution and a 1×1-convolution for binary pixel-wise classification.

The followings present the main components of our cross-attention-based method. We first illustrate the masked attention designed for support-query image pairs. Then, we elaborate on

how the symmetric cross-image encoding architecture incorporates features from other images. Finally, we introduce a feature enhancement scheme with feature fusion.

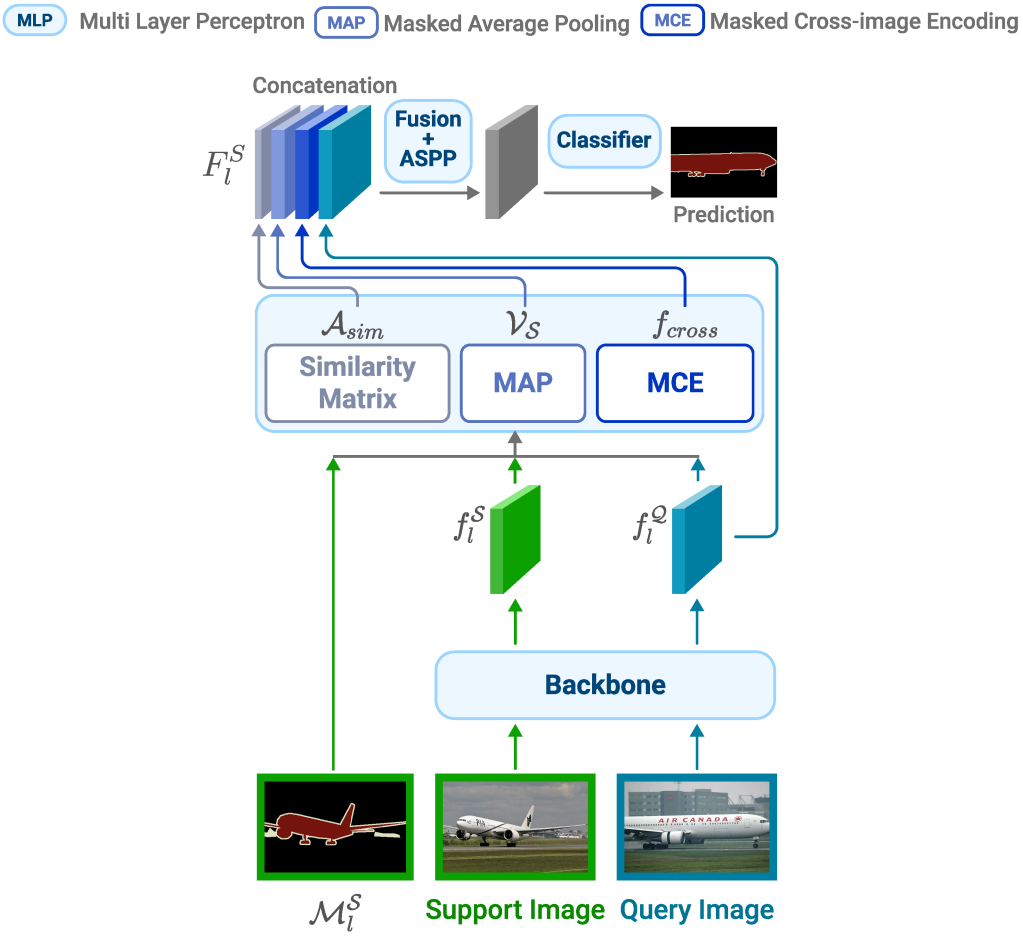


Figure 3.2: The proposed architecture consists of three distinct modules that take multi-level support-query features $\boldsymbol{f}l^S$, $\boldsymbol{f}l^Q$ obtained from the backbone network as input along with the support mask to align features. The similarity matrix calculation module evaluates the pixel-wise feature correspondences between the query and support features to derive a similarity score matrix $\mathbf{A}_{sim}$. The Mask Average Pooling (MAP) computes a class-wise prototype $\mathbf{V_s}$ from the support image and corresponding mask. Finally, the Masked Cross-image Encoding (MCE) module leverages the support segmentation mask to confine attention within the localized features of the target objects, thereby improving the ability to differentiate objects from the background. These module outputs $\boldsymbol{f}_{cross}$, $\mathbf{V_s}$ and $\mathbf{A}_{sim}$, are then concatenated and fused to generate rich features for final prediction.

3.2.3 Masked Attention Encoding

The proposed masked attention encoding is inspired by the attention mechanism of the Vision Transformer (ViT) [66] model, which first converts an image into a sequence of patches and then linearly maps each patch into tokens with positional embedding. A transformer encoder is composed of a sequence of blocks where each block contains multi-head self-attention (MSA) with a multi-layer perceptron (MLP). Specifically, a scaled dot-product attention is formulated as:

$$\text{Attention} = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V}, \tag{3.1}$$

where $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ are the different views of input patch tokens, d is the dimension of each token.

The proposed model adopts a meta-learning scheme with a masked cross-image attention module, which extracts the local features by constraining cross-attention to the foreground region of support features. Specifically, we take multi-level intermediate visual feature representations from support and query images as input. For input feature maps $\boldsymbol{f}_l \in \mathbb{R}^{H_l \times W_l \times C}$, $\mathbf{Q_l}, \mathbf{K_l}, \mathbf{V_l} \in \mathbb{R}^{H_l W_l \times N}$ are query-key-value tokens derived from flattened inputs through a projection head $\mathcal{G}_{proj}(\cdot)$, and H_l and W_l are the spatial resolutions of features to attend. The masked attention matrix is computed by:

$$\text{Attention} = \text{Softmax}\left(\left(\boldsymbol{\mathcal{M}}_l + \mathbf{Q}_l\right)\mathbf{K}_l^{\mathrm{T}}\right)\mathbf{V}_l, \tag{3.2}$$

where the segmentation mask $\mathcal{M}_l \in \mathbb{R}^{H_l W_l \times N}$ at feature location (x, y) is calculated by:

$$\boldsymbol{\mathcal{M}}_l(x,y) = \begin{cases} 0 & \text{if } \mathbf{M}_l(x,y) = 1; \\ -\infty & \text{otherwise.} \end{cases} \tag{3.3}$$

$\mathbf{M}_l \in \{0,1\}^{H_l W_l \times N}$ is a binary support mask that transformed from the original image mask $\mathbf{M}_l \in \{0,1\}^{H \times W}$. It is resized to the exact size of W_l, H_l as the input features by linear interpolation, followed by expansion along channel wise and flattened to $\mathbb{R}^{H_l W_l \times N}$.

3.2.4 A Symmetric Cross-image Feature Encoding Method

The few-shot Siamese segmentation framework [33] consists of a Support branch and a query branch. The former receives support images with annotated mask learning class representation to guide the query image segmentation in the latter branch. The two-branch guidance

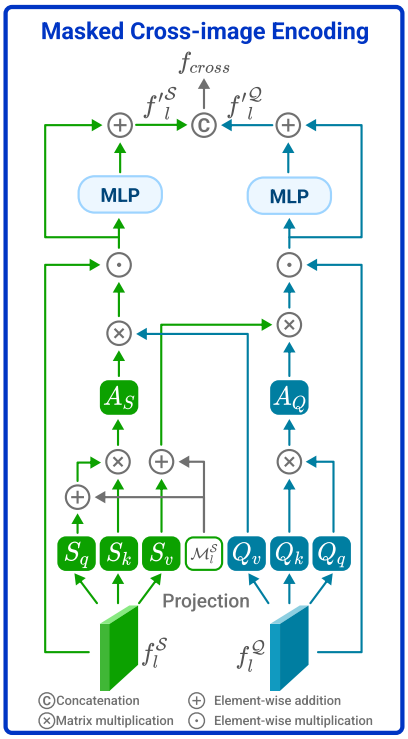


Figure 3.3: The proposed Masked Cross-Image Encoding.

architecture has been a source of inspiration for researchers in utilizing Transformer's cross-attention mechanism to enable query features to attend to informative support features, as demonstrated by CyCTR, a closely related work to our study. In CyCTR, support and query features are typically treated as separate entities. The query features are encoded using standard self-attention to obtain a query image representation, which is then unidirectionally passed to a cross-alignment block for query guidance. In contrast, our proposed method suggests that query and support features containing the same class can be integrated as a uniform feature source for generalizing novel class semantic information. By employing cross-image attention, we aim to capture shared representations and semantic relationships among images sharing the same class, which can enhance the meta-learning ability of the few-shot segmentation model.

As it shown in Figure 3.3, l^{th} level feature maps $\left(\boldsymbol{f}_l^S, \boldsymbol{f}_l^Q\right)$ are flattened to sequences of $H_l W_l$ patches, where each patch has C channels. This tokenization can be formulated by $\boldsymbol{f_l} = \left[\boldsymbol{f}_l^1, \boldsymbol{f}_l^2, \ldots, \boldsymbol{f}_l^{H_l W_l}\right]$, where $\boldsymbol{f}_l^i \in \mathbb{R}^C$. Given a token embedding with mappings $\boldsymbol{W}_S^q, \boldsymbol{W}_S^k, \boldsymbol{W}_S^v$, and $\boldsymbol{W}_Q^q, \boldsymbol{W}_Q^k, \boldsymbol{W}_Q^v$ for a specific scale support sequence $\boldsymbol{f}_S$ and query sequence $\boldsymbol{f}_Q$, the query-

key-value tokens $(\boldsymbol{S}_q, \boldsymbol{S}_k, \boldsymbol{S}_v)$ and $(\boldsymbol{Q}_q, \boldsymbol{Q}_k, \boldsymbol{Q}_v)$ can be calculated by:

$$\begin{cases} \boldsymbol{S}_q = \boldsymbol{W}_S^q \boldsymbol{f}_S \\ \boldsymbol{S}_k = \boldsymbol{W}_S^k \boldsymbol{f}_S \\ \boldsymbol{S}_v = \boldsymbol{W}_S^v \boldsymbol{f}_S \end{cases} \quad \begin{cases} \boldsymbol{Q}_q = \boldsymbol{W}_Q^q \boldsymbol{f}_Q \\ \boldsymbol{Q}_k = \boldsymbol{W}_Q^k \boldsymbol{f}_Q \\ \boldsymbol{Q}_v = \boldsymbol{W}_Q^v \boldsymbol{f}_Q \end{cases} \tag{3.4}$$

Note that we omit multi-head attention and multi-level indicator for a concise presentation. We employ the token embedding to preserve the spatial properties, then implement the cross-image encoding by a symmetric cross-attention in two branches: 1) We obtain the support cross feature embedding using the value vectors $(\boldsymbol{S}_q, \boldsymbol{S}_k, \boldsymbol{Q}_v)$. 2) Similarly, the query cross feature embedding from the query branch is calculated with vectors $(\boldsymbol{Q}_q, \boldsymbol{Q}_k, \boldsymbol{S}_v)$.

In the support branch, the support embedding first performs self-attention between support tokens $(\boldsymbol{S}_q, \boldsymbol{S}_k)$. Then it conducts cross-attention with a query token $\boldsymbol{Q}_v$ to enhance the feature representation of an object. Let $\boldsymbol{A}_S \in R^{HW \times HW}$ denote the matrix of self-attention scores obtained via linear mapping:

$$\boldsymbol{A}_S = (\boldsymbol{S}_q + \boldsymbol{\mathcal{M}})\, \boldsymbol{S}_k^{\mathrm{T}}, \tag{3.5}$$

where $\boldsymbol{S}_q = \left[\boldsymbol{S}_q^1, \boldsymbol{S}_q^2, ..., \boldsymbol{S}_q^{HW}\right] \in R^{HW \times C}$ and $S_k = \left[\boldsymbol{S}_k^1, \boldsymbol{S}_k^2, ..., \boldsymbol{S}_k^{HW}\right] \in R^{C \times HW}$ are token embeddings to perform self-attention within the query features, which can be obtained using Eq. 3.4. After the self-attention, the model conducts cross-image attention with the attention matrix $\boldsymbol{A}_S$ and the query token $\boldsymbol{Q}_v$. Moreover, to perform normalization for masked-attention scores and find out the regional semantical relations from the query branch, the scaled attention is calculated as follows:

$$\boldsymbol{R}_S = \mathrm{softmax}\left(\frac{\boldsymbol{A}_S}{\sqrt{d}}\right) \boldsymbol{Q}_v, \tag{3.6}$$

where $\boldsymbol{Q}_v = \left[\boldsymbol{Q}_v^1, \boldsymbol{Q}_v^2, ..., \boldsymbol{Q}_v^{HW}\right]$ is a token of query image features, and $\boldsymbol{R}_S \in \mathbb{R}^{HW \times C}$ represents the masked cross-image feature maps in support branch.

The query cross attention encodes the local semantic information of the support objects into the query feature in a similar way:

$$\boldsymbol{A}_Q = \boldsymbol{Q}_q \boldsymbol{Q}_k^{\mathrm{T}}, \tag{3.7}$$

$$\boldsymbol{R}_Q = \mathrm{Softmax}\left(\frac{\boldsymbol{A}_Q}{\sqrt{d}}\right)(\boldsymbol{S}_v + \boldsymbol{\mathcal{M}}), \tag{3.8}$$

where the support mask is applied on the support token to remove the background.

The obtained cross-image relation maps $\boldsymbol{R}_Q$ and $\boldsymbol{R}_S$ are then fed to the MLP block to further encode the cross-image common information into local regions as:

$$\begin{cases} \boldsymbol{f}_l'^S = \text{MLP}\left(\text{Norm}\left(\boldsymbol{R}_S \odot f_l^S\right)\right) + \text{Norm}\left(\boldsymbol{R}_S \odot f_l^S\right) \\ \boldsymbol{f}_l'^Q = \text{MLP}\left(\text{Norm}\left(\boldsymbol{R}_Q \odot f_l^Q\right)\right) + \text{Norm}\left(\boldsymbol{R}_Q \odot f_l^Q\right), \end{cases} \tag{3.9}$$

where Norm represents Layernorm and the MLP block contains of two transformation layers with GELU non-linearity. Finally, the reshaped outputs $\boldsymbol{f}\boldsymbol{r}_l^S$ and $\boldsymbol{f}\boldsymbol{r}_l^Q$ are aggregated as the cross encoding feature $\boldsymbol{f}_{cross}$ by concatenation and 1×1-convolution.

3.2.5 Similarity Matrix

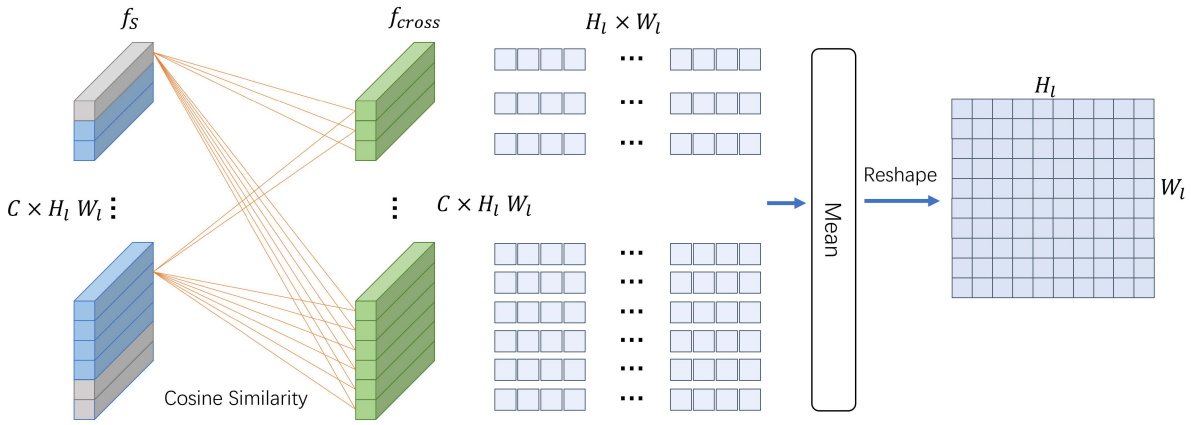


Figure 3.4: The calculation process of similarity matrix

We notice that the prior map of PFENet [36] determines the query pixel class according to the maximum similarity over the support pixels, which may bias towards a unique support pixel and thus fail to consider the whole object features. Therefore, as shown in Figure 3.4, we propose calculating a mean similarity score matrix to reflect the mean semantic correlation between each query feature position and support object positions. To obtain the relation matrix, we first compute the cosine similarity between every position pair $(\boldsymbol{x}_{cross}, \boldsymbol{x}_s)$ from the intermediate masked support feature $\boldsymbol{f}_S$ and the enhanced query feature $\boldsymbol{f}_{cross}$. The similarity for query pixel and background support pixel pair would be zero, as the mask operation sets background features to zero. The similarity scores matrix $\boldsymbol{A}_{sim} \in \mathbb{R}^{H_l \times W_l}$ are the mean relation scores as follows:

$$\begin{gathered} \boldsymbol{A}_{sim}\left(\boldsymbol{x}_{cross}, \boldsymbol{x}_s\right) = \text{mean}\left(\frac{\boldsymbol{x}_{cross}^T \cdot \boldsymbol{x}_s}{\|\boldsymbol{x}_{cross}\| \|\boldsymbol{x}_s\|}\right) \\ cross \in (1, 2, \ldots, H_l W_l), s \in (1, 2, \ldots, H_l W_l), \end{gathered} \tag{3.10}$$

The features $\boldsymbol{f}_Q, \boldsymbol{V}_S, \boldsymbol{f}_{cross}, \boldsymbol{A}_{sim}$ are concatenated as a whole feature, then it is fed into an ASPP module, where a dilated convolution is used to enlarge the receptive field. Finally, we apply a convolution block and a pixel-wise classifier to predict the final segmentation mask $Pred$:

$$Pred = \text{Softmax}\left(\text{CLS}\left(\text{Cat}\left(\text{f}_\text{Q}, \text{V}_\text{S}, \text{f}_\text{cross}, \text{A}_\text{sim}\right)\right)\right), \tag{3.11}$$

Here, CLS represents a combined operation of an ASPP, an 3×3-convolution and a classifier.

3.3 Experiments

3.3.1 Dataset and Evaluation Metric

We evaluated our method on the PASCAL 5^i [33], COCO-20^i [45] and the FSS-100 dataset as introduced in Section 2.5.1. PASCAL 5^i is composed of PASCAL VOC 2012 and extended annotations from SDS [130] datasets with 5,953 and 1,449 images for training and validation, respectively. 20 classes were evenly divided into 4 folds $5^i \in 0, 1, 2, 3$ and each fold contains 5 classes. The dataset COCO-20^i consists of 82,081 training images and 40,137 validation images from 80 object classes divided into 4 folds: $20^i \in 0, 1, 2, 3$. For the four subsets, three of them were selected as the training set, and the rest one was used as the testing set to validate the effectiveness. Note that the training images containing the novel classes on the testing set were removed to prevent information disclosure. For these four subsets, three of them are selected as the training set, and the rest one is used as the test set to validate the effectiveness of the proposed method. In FSS-1000, we adhere to the dataset's original configuration, dividing the 1,000 classes into train/validation/test sets with proportions of 520/240/240 classes, respectively. In the training stage, we randomly select one (one-shot) or five (five-shot) images for each class as the support images, another image as the query image. Following [36], we randomly sample 1,000 query-support pairs for testing.

We adopt mean intersection over union (mIoU) and foregroud-background IoU (FBIoU) as the evaluation metrics of our experiments. IoU for class c is defined as $IoU_c = \frac{TP_c}{TP_c+FP_c+FN_c}$, where TP, FP and FN are the number of true positives, false positives and false negatives of the predicted pixels. The mIoU is the average of all the IoU of different classes in each fold, i.e., $mIoU = \frac{1}{N}\sum_{i=1}^{N} IoU_i$, where $N = 5$ for PASCAL 5^i and $N = 20$ for COCO-20^i. FB-IoU calculates the foreground-background average IoU as $FB - IoU = \frac{1}{2}\left(\text{IoU}_\text{F} + \text{IoU}_\text{B}\right)$. As

FSS-1000 only provide the foreground object labels, we calculate the FB-IoU on this dataset.

3.3.2 Implementation Details

All the experiments are conducted on Pytorch platform, using a server equit with Intel Xeon Gold 6226R CPU and Nvidia Quadro RTX 6000 GPU. We use Stochastic Gradient Descent (SGD) as the optimizer, which we apply the "ploy" learningl rate scheduler with the momentum and weight decay of 0.9 and 10^{-5}, respectively. The model was trained for 300 epochs with a base learning rate of 0.0025 and batch size 16 on PASCAL 5^i. For COCO-20^i, models were trained for 150 epochs with a base learning rate of 0.005 and batch size 8. FSS-1000 is trained for 100 epochs using initial learning rate of 0.01 and batch size 32.

In the training stage, we randomly cropped the input images to 473×473 for PASCAL-5^i and COCO-20^i, 224×224 for FSS-1000. The models are implemented with VGG-16 and ResNet-50 backbone pre-trained on ImageNet, whereas we load the pre-train parameter from the offical Pytorch model zoo for fare comparison with other methods. The intermediate features from the backbone are 1/4, 1/8, 1/16 of the original input size for multi-scale feature fusion. For K-shot setting, the model takes averaged cross-image encoding feature $\{\boldsymbol{f}^i_{cross}\}^K_{i=1}$, prototype vector $\{\boldsymbol{V}^i_s\}^K_{i=1}$ and similarity score matrix $\{\boldsymbol{A}^i_{sim}\}^K_{i=1}$ to concatenate a fusion feature for final prediction. The shared channel dimension c of $\boldsymbol{f}_{cross}$ and V_s is set to 256. We build our baseline model on the source code of [99], where the Masked Average Pooling (MAP) operation is applied to generate a single prototype for guiding query images.

3.3.3 Results Analysis

Recent efforts optimize the few-shot segmentation models from the aspects of prototype construction [35], [94], [133], feature correlation learning [34], [126], [128], and query feature enhancement [36], [132]. Following the nature of few-shot learning, our method focuses on the aspects of feature enhancement and feature correlation learning. We compare the proposed method with other state-of-the-art methods on PASCAL-5^i and COCO-20^i under both 1-shot and 5-shot settings.

Quantitative Results Table 3.1 and Table 3.2 compare the meta-learner performance without filtering background through a base class segmentation network [39]. The tables provide the mean IoU for each fold, representing the average IoU scores across all classes within each fold

Table 3.1: The class mIoU results are reported for each Fold, with MeanIoU(%) representing the average class mIoU and FB-IoU for averaged foreground-background IoU across four folds for 1-shot and 5-shot segmentation on PASCAL-5^i. BAM* presents the performance of the meta-learner.

Backbone	Method	1-shot						5-shot					
		Fold-0	Fold-1	Fold-2	Fold-3	MeanIoU(%)	FB-IoU%	Fold-0	Fold-1	Fold-2	Fold-3	MeanIoU(%)	FB-IoU%
VGG-16	PANet [35]	42.3	58.0	51.1	41.2	48.1	-	51.8	64.60	59.8	46.5	55.7	-
	FWB [131]	47.0	59.6	52.6	48.3	51.9	-	50.9	62.9	56.6	50.1	55.1	-
	CRNet [128]	-	-	-	-	55.2	-	-	-	-	-	58.5	-
	PFENet [36]	56.9	68.2	54.4	52.4	58.0	72.0	59.0	69.10	54.8	52.9	59.0	72.3
	HSNet [126]	59.6	65.7	59.6	54.0	59.7	73.4	64.9	69.0	64.1	58.6	64.1	76.6
	QCLNet [132]	61.3	66.8	58.4	55.8	60.6	-	66.1	68.5	63.2	58.8	64.2	-
	BAM* [39]	59.9	67.5	64.9	55.7	62.0	-	64.0	71.5	69.4	63.6	67.1	-
	Ours	**60.6**	**69.5**	**65.1**	**56.3**	**62.9**	**74.5**	**65.6**	**72.8**	**69.7**	**64.7**	**68.2**	**78.2**
ResNet-50	PANet [35]	44.0	57.5	50.8	44.0	49.1	-	55.3	67.2	61.3	53.2	59.3	-
	CANet [122]	52.5	65.9	51.3	51.9	55.4	-	55.5	67.8	51.9	53.2	57.1	-
	CRNet [128]	-	-	-	-	55.7	-	-	-	-	-	58.8	-
	PPNet [94]	48.6	60.6	55.7	46.5	52.5	69.2	58.9	68.3	66.8	58.0	63.0	75.8
	PFENet [36]	61.7	69.5	55.4	56.3	60.8	73.3	63.1	70.7	55.8	57.9	61.9	73.9
	CyCTR [102]	**67.2**	71.1	57.6	59.0	63.7	-	**71.0**	**75.0**	58.5	65.0	67.4	-
	HSNet [126]	64.3	70.7	60.3	60.5	64.0	76.7	70.3	73.2	67.4	**67.1**	69.5	80.6
	QCLNet [132]	65.2	70.3	60.8	61.0	64.3	-	70.6	73.5	66.7	67.1	69.5	-
	BAM* [39]	65.7	**71.4**	65.6	59.0	65.4	-	67.3	72.4	69.2	66.3	68.8	-
	Ours	65.3	71.2	**66.2**	**61.0**	**65.9**	**78.1**	69.2	73.7	**70.5**	66.8	**70.0**	**81.3**

(5 classes in PASVAL and 20 classes in COCO). The term "Mean" refers to the average of the mIoU scores across the four folds. Additionally, the tables include the both overall mIoU and the FB-IoU for the entire dataset.

In Table 3.1, the results for PASCAL-5^i demonstrate that our proposed method surpasses all compared state-of-the-art techniques in terms of mIoU and FB-IoU when utilizing VGG-16 and ResNet-50 backbones for both 1-shot and 5-shot settings. Particularly noteworthy is our method's superiority in the most challenging 1-shot scenario, where it outperforms BAM by 0.9% and 0.5% with VGG-16 and ResNet-50 backbones, respectively. Furthermore, substantial performance gains are observed when providing 5 support images, with our method achieving a 1.1% (VGG-16) and 1.2% (ResNet-50) mIoU improvement over the SOTA. This underscores the effectiveness and superiority of our proposed model.

Additionally, with the ResNet-50 backbone, although CyCTR obtains marginally higher scores in fold-0 (0.8%) and fold-1 (1.3%), our model significantly outperforms it in challenging folds. Notably, in fold-2, we observe a remarkable 12% improvement, and in fold-3, a 1.8% improve-

Table 3.2: The class mIoU results are reported for each Fold, with MeanIoU(%) representing the average class mIoU across four folds for 1-shot and 5-shot segmentation on COCO-20^i. BAM* presents the performance of the meta-learner.

Backbone	Method	1-shot					5-shot				
		Fold-0	Fold-1	Fold-2	Fold-3	MeanIoU(%)	Fold-0	Fold-1	Fold-2	Fold-3	MeanIoU(%)
VGG-16	PANet [35]	-	-	-	-	20.9	-	-	-	-	29.7
	FWB [131]	18.4	16.7	19.6	25.4	20.0	20.9	19.2	21.9	28.4	22.6
	PRNet [133]	27.5	33.0	26.7	29.0	29.1	31.2	36.5	31.5	32.0	32.8
	PFENet [36]	35.4	38.1	36.8	34.7	36.3	38.2	42.5	41.8	38.9	40.4
	BAM* [39]	38.4	43.8	44.3	39.8	41.6	45.9	48.9	47.9	**47.0**	47.4
	Ours	**39.5**	**44.1**	**45.3**	**41.6**	**42.6**	**46.7**	**51.4**	**48.3**	46.5	**48.2**
ResNet-50	ASGNet [124]	-	-	-	-	34.6	-	-	-	-	42.5
	RePRI [134]	32.0	38.7	32.7	33.1	34.1	39.3	45.4	39.7	41.8	41.6
	PPNet [94]	28.1	30.8	29.5	27.7	29.0	39.0	40.8	37.1	37.3	38.5
	PFENet [36]	36.5	38.6	34.5	33.8	35.8	36.5	43.3	37.8	38.4	39.0
	HSNet [126]	36.3	43.1	38.7	38.7	39.2	43.3	51.3	48.2	45.0	46.9
	CyCTR [102]	38.9	43.0	39.6	39.8	40.3	41.1	48.9	45.2	47.0	45.6
	QCLNet [132]	39.8	45.7	42.5	41.2	42.3	46.4	53.0	52.1	48.6	50.0
	BAM* [39]	41.9	45.6	**43.9**	41.2	43.1	47.0	51.9	49.5	47.8	49.0
	Ours	**42.1**	**48.3**	43.7	**42.8**	**44.2**	**47.8**	**55.2**	**50.8**	**50.3**	**51.0**

ment. These folds encompass challenging classes like "potted plant" and "dining table," often accompanied by other classes, posing difficulties in distinction. Our method, which employs patch-wise mutual correlation encoding to enhance the visual representation of the target class, proves superior in addressing such complex scenarios compared to CyCTR, which solely performs image-level feature propagation.

Similarly, Table 3.2 depicts the comparison of class mean IoU performance on COCO-20^i. With VGG-16 as the backbone, our approach attains state-of-the-art results of 42.6% mIoU under the 1-shot setting and 48.2% under the 5-shot setting. Furthermore, we observe even more precise segmentation performance with the utilization of the more powerful ResNet-50 backbone, achieving 44.2% and 51.0% in the 1-shot and 5-shot settings, respectively.

In comparison to the PASCAL-5^i dataset, the COCO-20^i dataset encompasses a broader array of object categories and poses greater challenges, including variations in object scale and occlusion levels. Previous prototype-based approaches have primarily relied on compressed prototype vectors to match all query feature pixels, disregarding the spatial structures of support features and contextual affinity. Consequently, they struggle to accurately align support and query images, particularly for objects with significant scale disparities and occlusion. In contrast, our proposed method integrates intra-image regional affinity and inter-image semantic relationships

Table 3.3: FB-IoU results on FSS-1000

Methods	Backbone	1-Shot	5-Shot
OSLSM [33]	VGG-16	70.3	73.0
GNet [135]		71.9	74.3
FSS1000 [48]		73.5	80.1
PFENet [36]		81.5	82.7
HSNet [126]		82.3	85.8
Ours		**83.8**	**86.2**
PFENet [36]	ResNet-50	84.6	86.1
HSNet [126]		85.5	87.8
Ours		**86.6**	**88.2**

by embedding contextual correlations that encapsulate many-to-many dependencies. Consequently, our method consistently outperforms the compared methods on the COCO-20^i dataset in comparison to the PASCAL-5^i dataset. In contrast to Pascal VOC and COCO datasets, FSS-1000 encompasses a broader array of object categories, including those not present in previous datasets such as diminutive objects, merchandise, and logos. Nevertheless, the dataset suffers from a dearth of images per class, with each class consisting of merely ten images. This limitation constrains the diversity of available support image combinations and exacerbates the risk of overfitting during model training. Sine the dataset exclusively provides annotations for foreground objects, In Table 3.3, we delineate the FB-IoU performance of models using VGG-16 and ResNet-50 architectures under both 1-shot and 5-shot settings. Our approach attains a pinnacle in performance among the contrasted methodologies, exhibiting a notable improvement over the FSS-1000 [48] baseline by 10.3% using the VGG-16 backbone with a single support image. Remarkably, despite the absence of a complex dense correlation decoder akin to HSNet [126], our method outperforms HSNet by considerable margins of 1.5% and 1.1% when leveraging the VGG-16 and ResNet-50 backbone, respectively. This underscores the compelling effectiveness and computational efficiency of our proposed approach.

Qualitative Results Figure 3.5 illustrates qualitative segmentation results for unseen classes in the 1-shot setting. The first column displays support images with their corresponding masks highlighted in blue, while the second and third columns depict the predictions and ground truth of query images, respectively. Notably, our method produces precise pixel-wise predictions, effectively covering nearly all target areas with just one support image. Particularly remarkable

is the model's performance in segmenting "boat," where it even surpasses the ground truth by labeling the rear part of the boat, not included in the ground truth label. This achievement is attributed to the MCE module, which captures contextual information to identify adjacent pixels sharing common semantics, thus underscoring the benefits of cross-image encoding within our framework.

The visualization results presented in Figure 3.6 showcase the MCE output maps for the PASCAL-5^i dataset under a 1-shot setting. The first two columns exhibit instances of support images with ground truth annotations highlighted in green, followed by query images with labeled masks depicted in red. Subsequent to these, correlation encoded feature maps and prediction outcomes are displayed, illustrating the impact of MCE on segmentation performance. Specifically, we examine the correlation maps generated in the MCE moduel, which provide a succinct overview of the relationship between different image regions. These correlation maps are represented as heatmaps for ease of visualization. Notably, the features of target object regions such as "bottle" and "boat" are notably enhanced through cross-image encoding. Consequently, regions exhibiting higher values are expected to be activated in the final segmentation decoder, contributing to improved segmentation performance.

3.3.4 Ablation Study

We first discussed the choices of output maps in the masked cross-image encoding module and then studied the effectiveness of each fused feature. All the tests were conducted under the 1-shot setting on PASCAL-5^i using ResNet-50 backbone.

Selection of the MCE outputs. Due to the nature of the few-shot segmentation task, which mainly focuses on segmenting the query image based on the support images, support features are typically kept constant and used only as references to enhance the query feature. Recent decoder-oriented transformer model [102] suggests that passing support images that do not correspond to the query mask may negatively impact the self-alignment of the query images. However, it is also worthwhile to enrich the support feature by referencing contextual information from the query feature to reduce underlying inductive bias. To determine the most effective feature in our encoder-oriented symmetric architecture, we separately output the support feature $\boldsymbol{f}\prime_l^Q$ and query feature $\boldsymbol{f}\prime_l^Q$ in Figure 3.3, as well as their fusion feature $\boldsymbol{f}_{cross}$, as the final cross-image encoding feature. Experimental results presented in Table 3.4

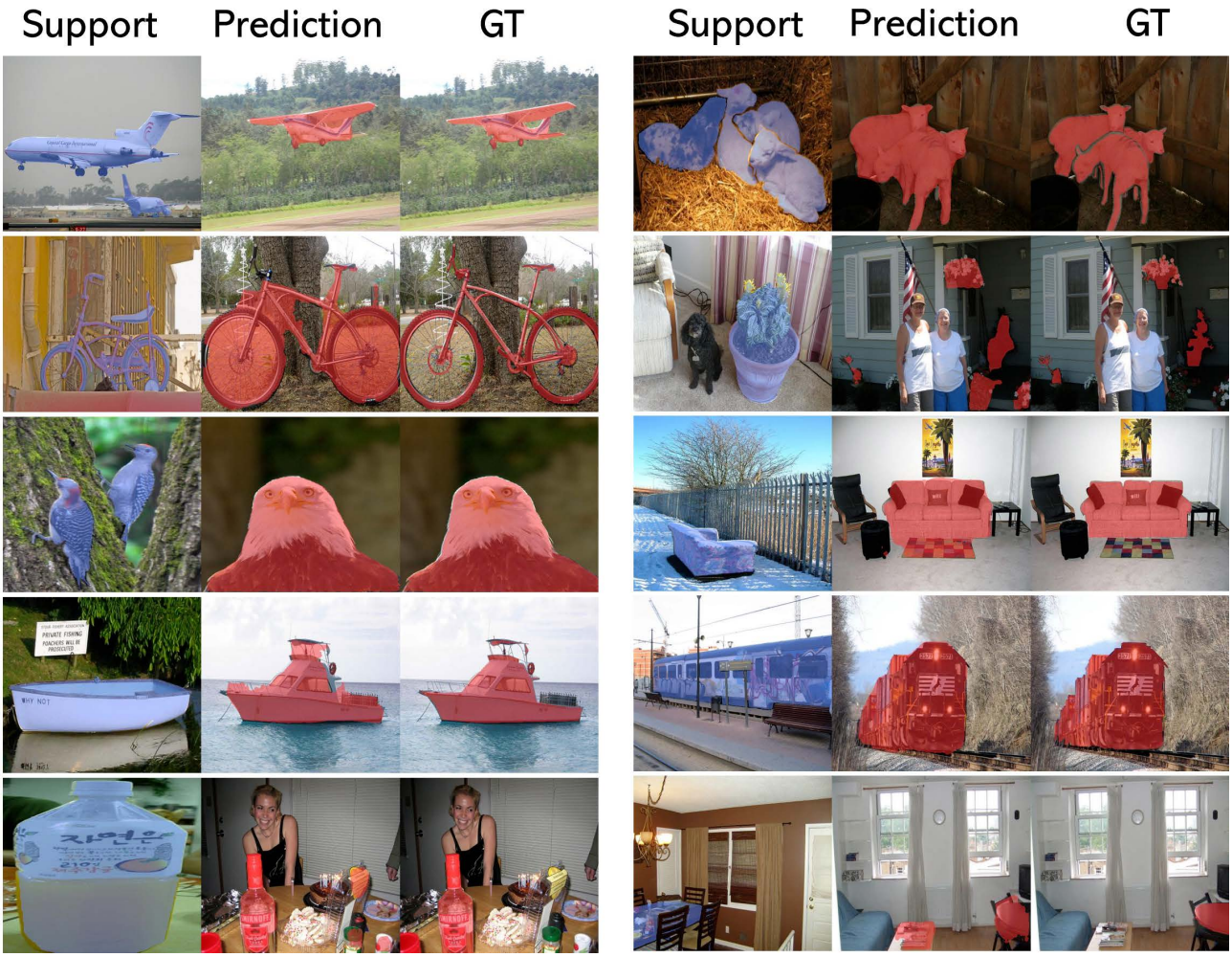


Figure 3.5: Qualitative results on PASCAL-5^i dataset in 1-shot setting

indicate that the fused feature outperforms the other two features, suggesting that symmetric cross-image encoding exploits more mutual dependencies than its asymmetric counterparts and leads to implicit feature guidance at the encoding stage.

Components Ablations. Table 3.5 presents an analysis of the effectiveness of each component in the proposed network. The results indicate that all proposed modules have a positive impact on performance improvement. Specifically, the absence of cross-image encoding, similarity matrix, and multi-level strategy decreases the prediction mIoU by 0.82%, 0.32%, and 1.54%, respectively, compared to the final aggregation performance. The proposed cross-support-query encoding on multi-level features contributes a noticeable performance gain in enhancing few-shot segmentation performance.

Differences between possible cross-attention schemes. Various cross-attention-based variants exist for modeling pixel-level relations between support and query features extracted

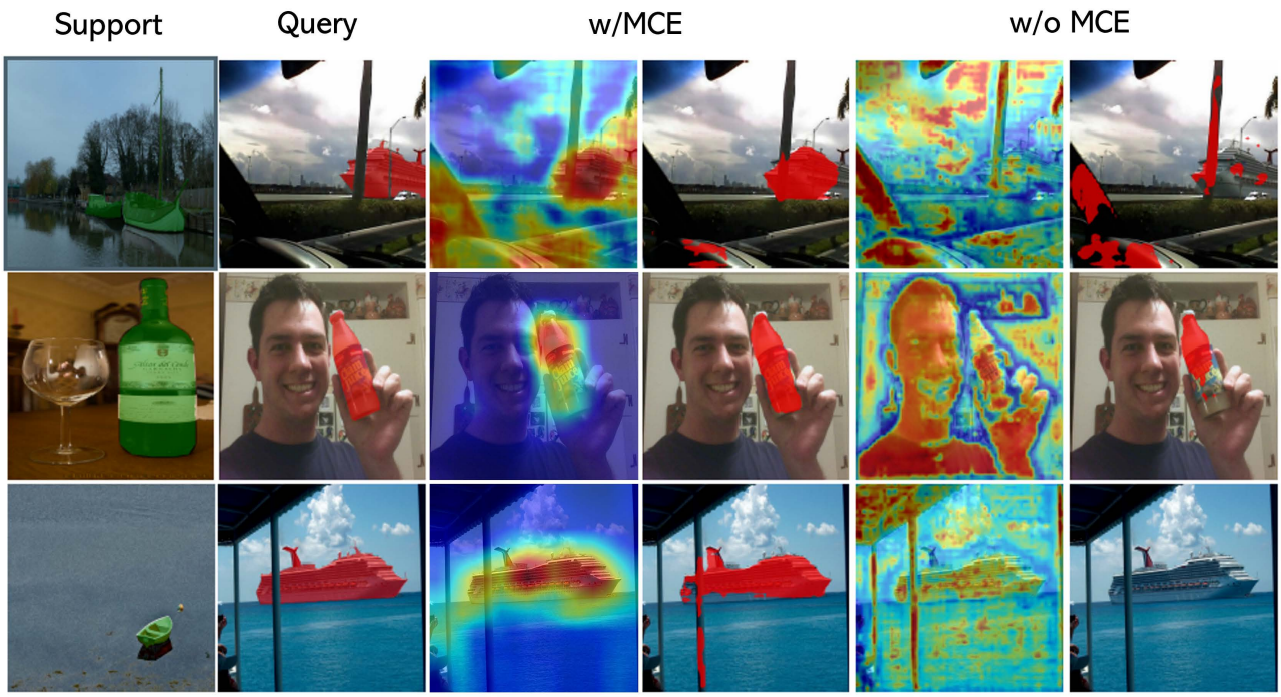


Figure 3.6: Visualization results of the MCE output maps on both PASCAL-5^i under 1-shot setting.

Table 3.4: The performance of optimal output map of the masked cross-image encoding module

$\boldsymbol{f}_l'^Q$	$\boldsymbol{f}_l'^S$	$\boldsymbol{f}_{cross}$	mIoU%
✓			65.24
	✓		63.13
		✓	65.93

from a CNN-based backbone at a high level. In this section, we consider typical forms of these variants illustrated in Figure 3.7 and separately incorporate them into the baseline model. The mIoU results under the 1-shot setting on PASCAL-5^i are presented in Figure 3.7. These variants can be categorized into three distinct classes as follows:

- 1) Unidirectional Query Encoding(UQE): This is the most straightforward approach, which directly adapts vanilla self-attention and cross-attention in the Transformer [65]. In this variant, only the features from the query branch are enhanced through cross-attention from the support branch. Much like the approach described in [102], this method employs self-attention within the support image and subsequently passes the resulting features to the query branch for one-way feature propagation. This unidirectional cross-attention

Table 3.5: The results of module performance

Cross Map	Sim Mat.	multi-level	mIoU%
	✓	✓	64.84
✓		✓	65.34
✓	✓		64.12
✓	✓	✓	65.93

approach yields the lowest segmentation performance across most of the splits and the overall mIoU. The limitation lies in the fact that this scheme only focuses on enhancing the semantics of a single branch, leading to inferior segmentation performance

- 2) Efficient Bidirectional Encoding (EBE). An efficient variant is introduced in [38], where cross-image attention encoding is performed only once to obtain a weighted feature correlation matrix, which is used to enhance both query and support features. This approach effectively mitigates the computational and memory challenges associated with the Transformer architecture. However, it is important to note that this variation lacks self-attention, which could potentially result in the loss of vital intrinsic information. The subpar performance observed on split3 can be attributed to the model's overemphasis on inter-image correspondence while neglecting the contextual intra-image information within both the support and query images themselves.

- 3) Masked Cross-image Encoding (MCE). Our masked cross-encoding encoding scheme, as illustrated in Section 3.2.4, represents a more versatile approach to cross-attention models for FSS. Unlike the aforementioned structures, MCE adopts a symmetric bidirectional architecture that simultaneously considers both intra-image contextual information and inter-image feature mutual correspondences, resulting in fewer inductive biases and a more comprehensive representation of the query image.

From Figure 3.8, it is evident that the MCE architecture attains the highest scores in terms of mean IoU compared to the other two typical cross-attention designs, and it also outperforms them in the majority of splits. Regarding the EBE approach, which employs bidirectional cross-attention mechanisms, it achieves notably higher scores than the one-way image-level feature propagation scheme (UQE). These findings indicate the significant role of mutual correlations

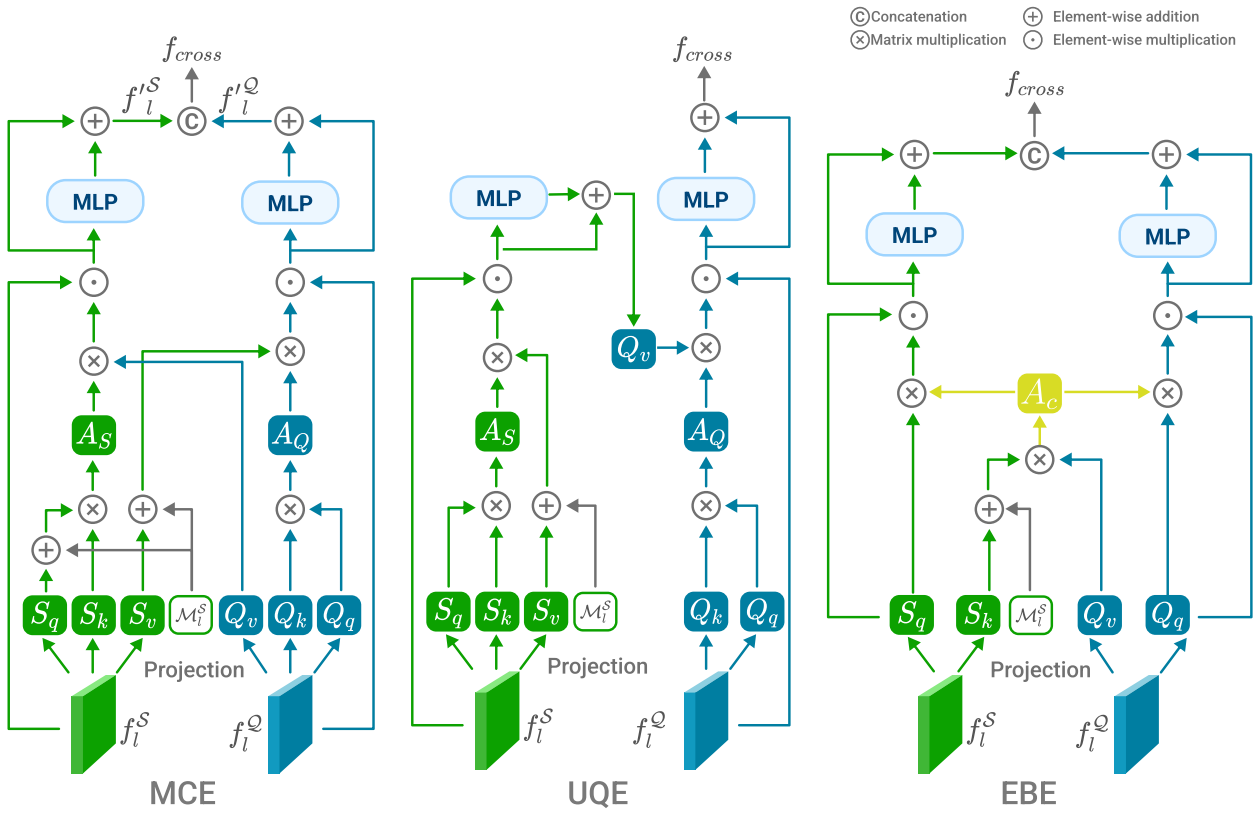


Figure 3.7: Comparison between conventional "fixed feature + decoder optimization" FSS methods and the proposed learnable cross-image feature scheme. (a) Current FSS methods independently learn object prototypes from support features and conduct coarse prototype matching or fine "pixel-wise" feature guidance at the decoding stage. (b) The proposed joint encoding scheme makes it possible to learn extra object context from other images before decoding.

in enhancing feature representations within the FSS framework. Moreover, the inclusion of self-attention operations proves to be indispensable in enriching intra-image target region features.

The visualized output masks of the potential cross-attention schemes, as depicted in Figure 3.9, illustrate that the Multi-Context Encoding (MCE) approach can better differentiate irrelevant regions compared to UQE and EBE. Particularly noteworthy is the observation in the last row, where the majority of the "people" area is included in both UQE and EBE, whereas MCE effectively excludes the main body of the rider.

From the in-depth camparsion of different cross-image schemes, it suggests how to design a cross-attention mechanism under the setting of few-shot segmentation from the following aspects.

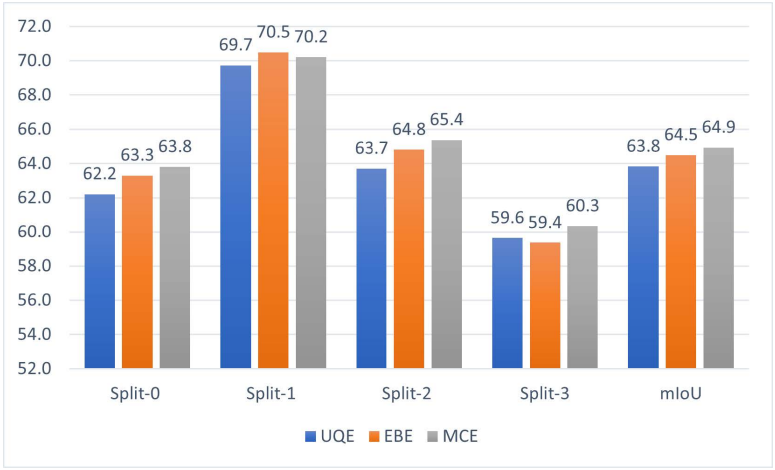


Figure 3.8: Performance of different cross-attention schemes on 4 sub-folds of PASCAL-5^i

1) Mutual consistency is a principle guiding the model to maintain symmetry in considering relationships between support and query images. It asserts that the relationship observed from one perspective should align with the relationship observed from the opposite perspective. For example, when comparing a picture of a cat (query image) to pictures of dogs (support set), the similarity in shape between the cat and the dog should be consistent regardless of which image is considered the reference. This principle ensures fairness in evaluating relationships between query and support images. To implement mutual consistency, the model employs a symmetrical architecture, ensuring that relationships are calculated in a balanced manner from both perspectives. This symmetrical design allows the model to treat both query and support images equally when assessing their relationships.

2) Taking into account local features is crucial for precise dense prediction tasks, such as semantic segmentation. While cross-image relation learning has broad applicability, semantic segmentation demands pixel-level feature affinity. Real-world images present diverse challenges, including objects appearing in different poses, scales, and with varying appearances. Global feature propagation, akin to the UQE scheme, may overlook subtle local semantic cues and introduce dependency inconsistencies. This occurs because relation dependencies are established across different levels of feature representations, potentially impairing the model's capacity to accurately classify objects.

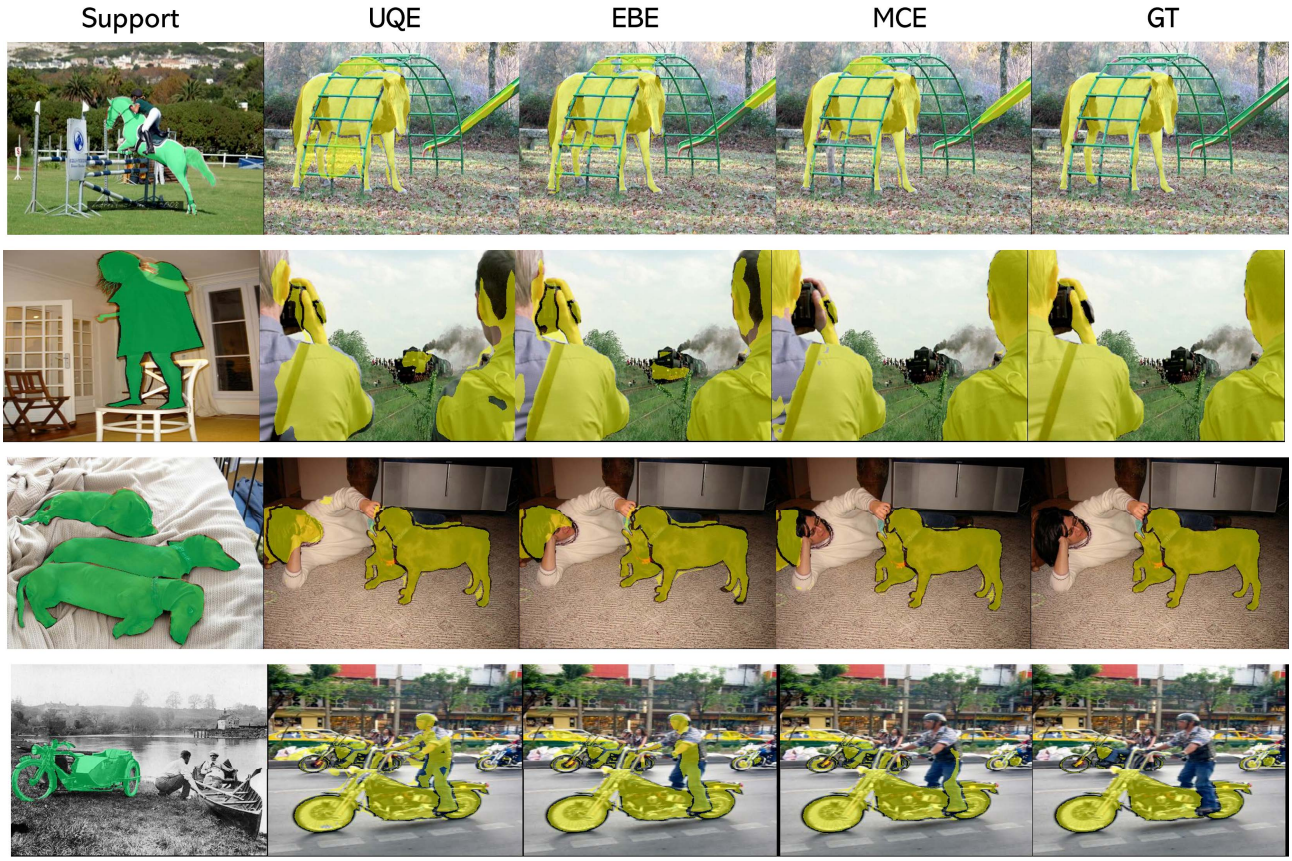


Figure 3.9: Qualitive comparison of the output masks of three alternative cross-attention designs. From the left colunm to the right are support image, Unidirectional Query Encoding(UQE), Efficient Bidirectional Encoding (EBE), Masked Cross-image Encoding (MCE) and ground Truth.

3.4 Chapter Summary

This chapter presents a novel approach to few-shot semantic segmentation by incorporating masked cross-image encoding, mask average pooling, and similarity scores to perform multi-level guidance for FSS. The approach is designed to mine support-query mutual dependencies and introduce a novel approach for jointly encoding shared semantic information and an intuitive scheme for calculating comprehensive pixel-wise relations. The symmetric encoder, utilizing masked cross-image attention, effectively constrains attention within target object regions to enhance support-query feature interaction, which enriches the semantic context of query features and provides implicit guidance for segmentation. Extensive experiments on benchmark datasets demonstrate the effectiveness of our approach, which outperforms compared methods.

Chapter 4

Hierarchical Multi-Prototype Discrimination: Boosting Support-Query Matching for Few-Shot Segmentation

4.1 Introduction

Chapter 3 introduces an efficient way to exploit support and query images from the aspect of feature encoding enhancement. While the proposed MCE module exhibits a plug-and-play advanced encoding ability to improve the support-query FSS framework, the post-encoding process plays a more vital role in the performance boost. This chapter discusses a potential scheme to address FSS from the aspect of feature matching. This is motivated by the fact that mainstream FSS methods adopt a support-query matching paradigm that activates target regions of the query image according to their similarity with a single support class prototype. However, this prototype vector is inclined to overfit the support images, leading to potential under-matching in latent query object regions and incorrect mismatches with base class features in the query image.

FSS is an extension of few-shot learning (FSL) [136]–[139], specifically designed as a dense prediction task. In contrast to semi-supervised methods, which initialize a base model using

a small set of labeled data and then refine the base model with large amounts of unlabeled data, FSS addresses a more challenging setting that only a few labeled examples are provided, without any additional unlabeled data. It implies that semi-supervised strategies, which require considerable unlabeled data to provide pseudo-label [140], contrastive pairs [141] or synthetic samples [142], are not applicable when addressing FSS task. To rapidly adapt to unseen classes, mainstream FSS approaches [35], [124], [128], [143] basically adopt the prototype matching paradigm [28], [29], [81] that aims to learn novel class prototypes from labeled support images and then segments query(test) image by referring to prototype matching scores. Following this paradigm, optimization works primarily revolve around two major aspects, namely, learning class-agnostic visual representations of the target class [94], [124], [144], [145], and devising effective semantic matching mechanisms [126], [146], [147].

However, scarce support images pose a formidable challenge in constructing a versatile prototype to encapsulate all the visual properties of a novel class. This challenge arises from two primary facets: firstly, the inherent variability observed among support-query images can lead to an issue of under-matching. This implies that the prototype matching process fails to activate certain query visual content that is rarely presented within the support images. As illustrated in Figure 4.1, the features of "nose" and "mouth" constitute a relatively small fraction of the "dog" support image. As a result, the prototype incorporates limited information concerning these regions, inevitably missing the "mouth" region in the query image. Secondly, the "dog" shares visually similar attributes with the "cat'' class, which can potentially lead to false positive matches when solely relying on foreground prototypes for feature alignment.

To alleviate the under-matching problem, the key lies in minimizing the distance between support prototypes and the missed query features within a latent space. Many prevalent methods extract foreground prototypes from support images [124], [144] or the entire training set [37], hoping that these prototypes can capture a broader range of novel class properties, thus increasing the likelihood of matching more target pixels within the query image. However, we argue that, in addition to creating optimal prototypes, another viable strategy to increase matching confidence is to bring the potential novel class features of the query image closer to the support prototypes. To end this, we introduce a collaborative learning approach between query and support features to establish a unified representation for novel classes and offer implicit feature guidance. This is achieved through the introduction of Masked Cross-Image Encoding (MCE), as proposed in Chapter 3, which integrates the semantic information of the support novel class

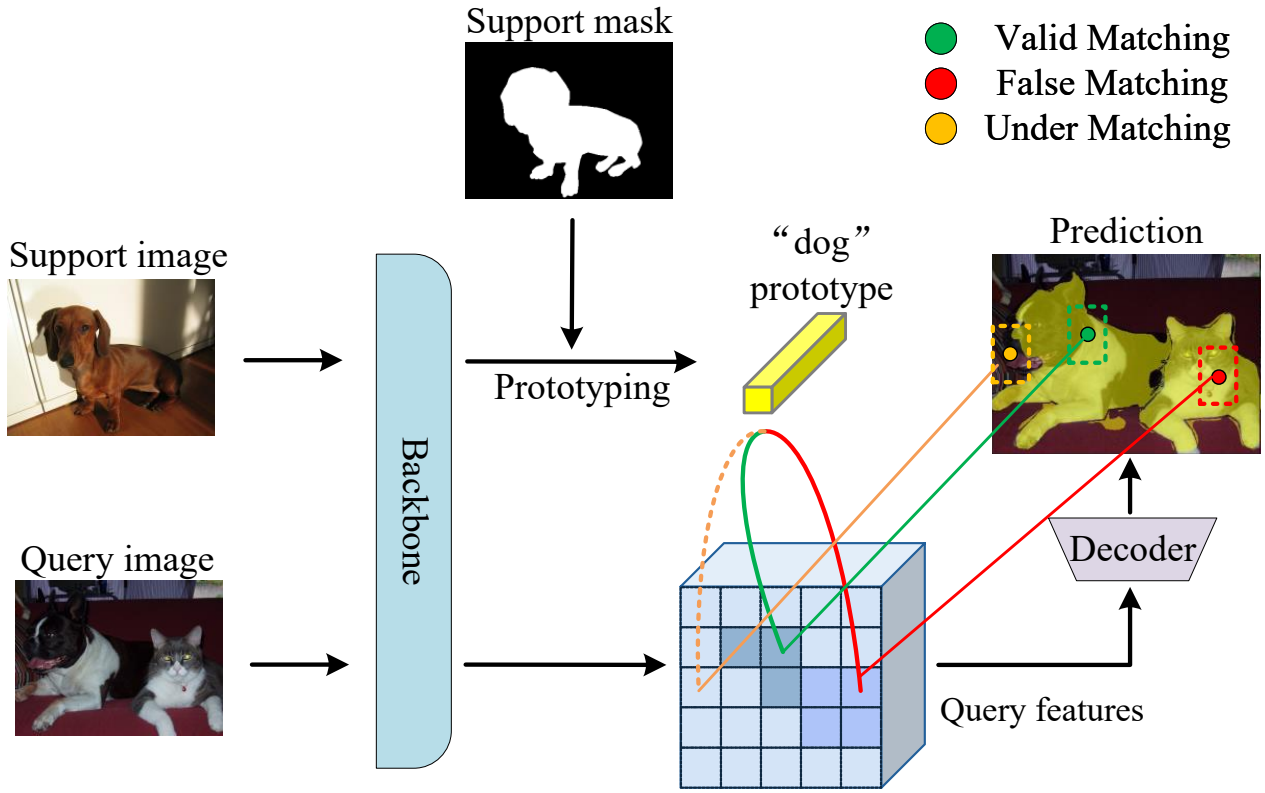


Figure 4.1: Conventional FSS methods learn a single novel class prototype (e.g., "dog ") from support features independently, which results in under-matching problems when the support image lacks a similar part to that in the query image and false matching problems when background features resemble the "dog".

into the query feature. The MCE module adopts a symmetric structure inspired by the cross-attention mechanism in Transformer architectures [65], [66] to establish semantic relationships between objects across support and query images. By leveraging this module, the target query feature is facilitated in becoming more akin to the support features, thereby obtaining a higher similarity score when matched with the support prototype.

In contrast to learning novel class prototypes with limited images, it is easier to capture comprehensive base class visual properties as abundant labeled base images are available in the FSS task. Additionally, it is important to note that base class objects are treated as background when segmenting novel class images. Inspired by these observations, we reevaluate the prototype learning and matching process in existing FSS approaches from a multi-prototype perspective. Unlike employing multiple prototypes as a pixel classifier for label assignment in conventional segmentation [148], our primary aim is to effectively suppress irrelevant ob-

ject regions with the assistance of base class prototypes. Specifically, we introduce a "visual words" dictionary lookup paradigm, where every spatial location of the query image feature is compared to both the base class prototypes and the novel class prototype. In practice, we devise a Semantic Multi-prototype Matching (SMM) module that combines base class prototypes with the current novel class prototype to identify target regions in the query image according to matching scores. Query features exhibit greater similarity to base prototypes are categorized as background, and their matching scores are set to 0. The feature-prototype matching is performed on multi-scale features to obtain corresponding guide maps for the subsequent segmentation guidance. This innovative method effectively mitigates the class-matching ambiguity typically encountered in conventional FSS methods that primarily rely on foreground prototype matching.

We notice that dense feature matching often struggles to identify continuous semantic regions within target objects, we propose to exploit the prior knowledge from the backbone to activate salient regions that might be suppressed in the prototype matching stage. Concretely, we design an adaptive feature activation map called Target-Aware Class Activation Map (TWCAM), derived from the Class Activation Map (CAM) that is commonly used in weakly supervised semantic segmentation to approximate the spatial location and broad semantics of target objects. To enhance the effectiveness of CAM in the context of few-shot segmentation, we incorporate a learnable lightweight network into the FSS meta-task. This network learns how to predict a weight matrix to refine CAM, resulting in a more accurate and precise representation. Leveraging weighted CAM in our model significantly improves the identification and delineation of target segments.

In this model, we incorporate the MCE module introduced in Section 3 to improve the correlation between query and support features. It works as a complementary component with other modules to facilitate the multi-prototype matching in the following ways:

- Enhanced MCE Functionality: In this work, MCE serves not only to enrich the representation of query features but also helps bring target object features closer to the support prototype, mitigating the issue of under-matching.

- Multi-Prototype Alignment Paradigm: We introduce a new multi-prototype alignment approach to tackle the remaining mismatching problem observed in the previous work, which was caused by a single foreground matching approach. By comparing query features

with multiple prototypes derived from both novel and base classes, we effectively suppress regions that show high confidence in base class prototypes.

- Feature Activation Map, TCAM: Instead of relying solely on prototype matching, which neglects spatial relationships within semantics, we introduce a novel feature activation map called TCAM. TCAM complementarily activates salient target regions to enhance segmentation performance.

- Hierarchical Feature Alignment: To address objects of diverse sizes, ranging from intricate details to prominent large elements, we advocate for multi-prototype feature alignment across multiple scales. Through the fusion of guide maps from different scales, the model can leverage rich semantic cues to enhance its capacity to differentiate between objects with similar appearances based on their contextual surroundings.

Leveraging the proposed modules, we introduce a Hierarchical Multi-prototype Matching Network (HMMNet). This network establishes a hierarchical feature guidance scheme based on multi-prototype matching and enhanced features. The multiple prototypes serve as generalized visual representations for all base classes across the entire dataset, aimed at amplifying mutual discrimination among novel and base class regions, thereby mitigating mismatching phenomena in FSS. To address under-matching issues, the network establishes symmetric patch-wise correspondences between the support target region and query feature using the MCE module. This mechanism effectively enhances the query feature, bringing it closer to the support prototype and resulting in more precise matching with increased confidence. Furthermore, to generate semantically consistent masks, we incorporate a novel module that predicts target-aware class activation maps. This module effectively activates target segments that might otherwise be erroneously eliminated due to dense matching guidance. Extensive experiments conducted on the PASCAL 5^i and COCO 20^i benchmarks illustrate that our method outperforms state-of-the-art techniques, offering a fresh perspective on handling prototype-based FSS methods.

4.2 Methodology

The framework of the proposed encoder-decoder-based model is depicted in Figure 4.2. The encoder comprises a convolutional neural network pretrained on ImageNet and the proposed MCE module in Chapter 3. This integrated feature encoder is tasked with extracting a diverse

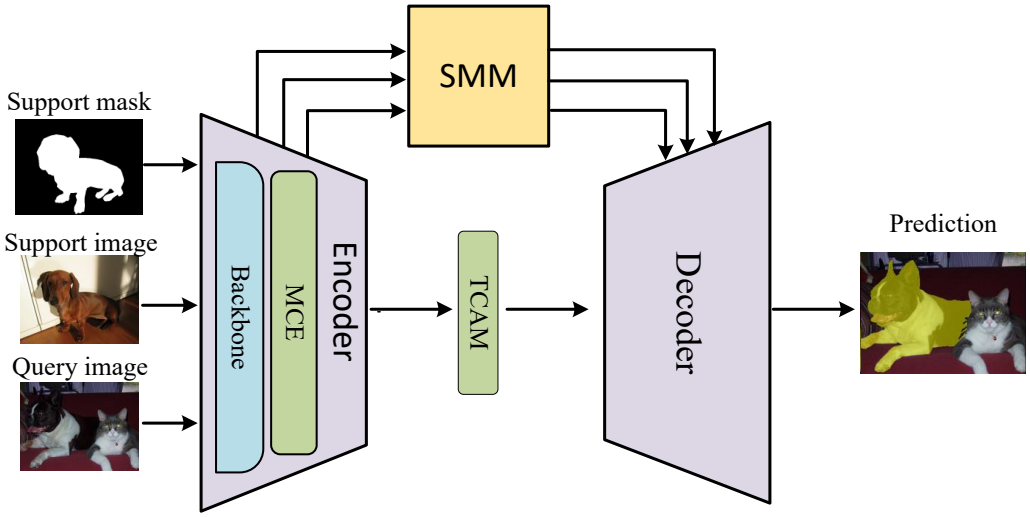


Figure 4.2: The proposed hierarchical multi-prototype matching network (HMMNet). The backbone network, along with the proposed MCE, extracts enriched multi-scale support and query features. These features are then sent to SMM to produce multi-scale matching score maps. Finally, in the decoder, the query features pre-activated by TCAM are gradually guided at corresponding resolution using matching score maps from SMM.

array of enriched intermediate feature maps with varying spatial resolutions from both support and query images. Subsequently, these features are inputted into the SMM to conduct multi-prototype matching and generate multi-scale matching guide maps. These maps guide the pre-activated query feature in the decoder in a coarse-to-fine manner.

In the following, we provide an overview of the model's data flow, followed by a detailed explanation, using single-scale features as an example. This includes an exploration of feature enhancement in MCE, the construction of multiple prototypes using the base dataset, and how object regions are activated by the proposed Target-Aware Class Activation Map (TCAM).

4.2.1 Network Data Flow

This research presents a novel framework for few-shot semantic segmentation, where the support-query matching problem is tackled through a visual properties prototype look-up approach. The proposed semantics alignment scheme architecture consists of several modules, as illustrated in Figure 4.3. The backbone network, which is pre-trained on ImageNet, is used to extract a set of intermediate feature maps from both the support and query images. These feature maps, referred to as $\{(F_l^s, F_l^q)\}_{l=1}^3$, are fed into the Masked Cross-image Encoding (MCE) module,

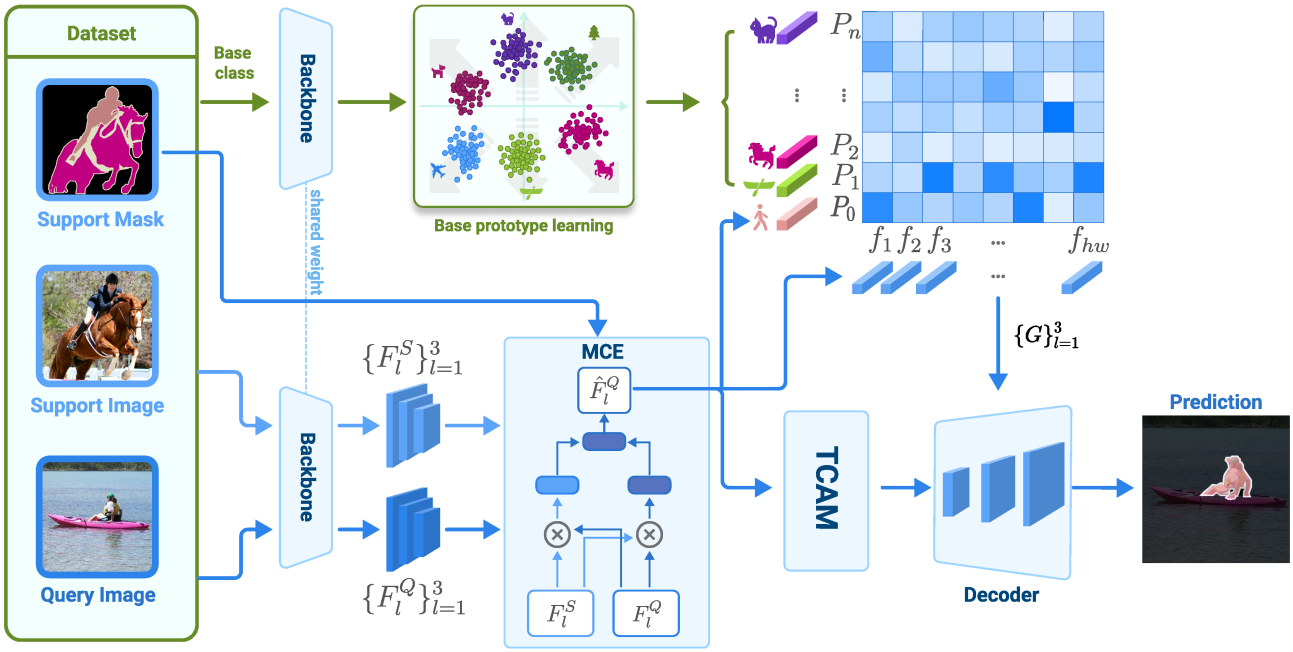


Figure 4.3: Overall architecture of the proposed Hierarchical Multi-prototype Matching Network (HMMNet), which incorporates three novel components of Masked cross-image encoding module(MCE), Target-aware Class Activation Map (TCAM), and Semantic Multi-prototype Matching (SMM). The detailed data flow is elaborated in Section 4.2.

where a masked cross-attention operation is employed to captures cross-image visual information. The MCE builds the support-query mutual correspondence for implicit guidance and reduces the inter-image feature distance between support and query features in a latent space.

The enhanced query feature $\hat{F}_l^Q$ is then fed into the Semantic Multi-prototype Matching (SMM) module, where each location of the query feature is compared with the support novel class prototype and base class prototypes. Those features yielding high confidence with base class prototypes are deemed as background, thus the similarity scores of the corresponding features are set to 0. After the matching process, the SMM module outputs multi-scale class similarity maps $\{G\}_{l=1}^3$, which are used to guide the pyramid segmentation process for the query image. Finally, the hierarchical decoder leverages the class similarity maps $\{G\}_{l=1}^3$ and TCAM to activate the query feature $\hat{F}_l^Q$ at corresponding resolutions in a coarse-to-fine manner.

4.2.2 Cross Feature Enhancement and Novel Prototype Acquisition

In Chapter 3, cross-attention has demonstrated its efficacy in enhancing visual representations of relevant image parts by merging information from different instances. In this work, we

mainly focus on alleviating the problem of prototype overfitting to support images and false matching to the base classes. We argue that the MCE bridges the intra-class semantic gap and the base prototype construction deals with the inter-class discrimination problem to achieve comprehensive feature matching.

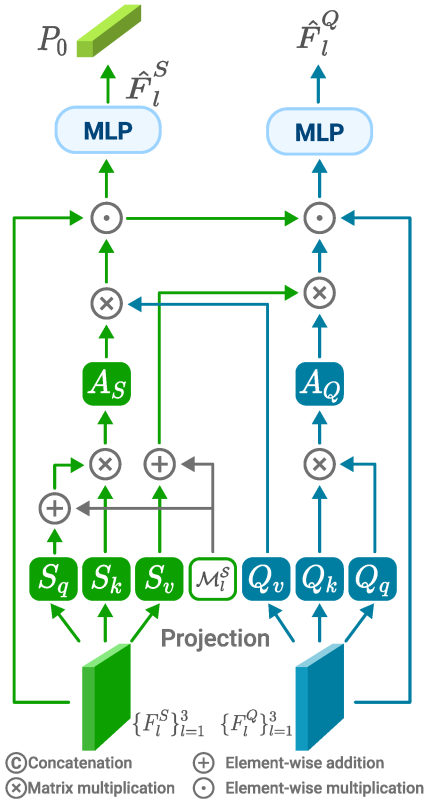


Figure 4.4: The modified Masked Cross-Image Encoding module. It outputs the enhanced query feature $\hat{F}_l^Q$ for subsequent segmentation and the the enhanced support feature $\hat{F}_l^S$ to generate the novel class prototype P_0

Expanding on the Masked Cross-Image Encoding (MCE), the cross-image feature enhancement module should not only allow the model to extract shared knowledge between support images but also draw the query features of target objects closer to the prototype of the support novel class. To end this, we modify the MCE as it shown in Figure 4.4. Similarly, l^{th} scale feature maps $\left(\boldsymbol{F}_l^S, \boldsymbol{F}_l^Q\right)$ are flattened to sequences of $H_l W_l$ patches, and are tokenized to $(\boldsymbol{S}_q, \boldsymbol{S}_k, \boldsymbol{S}_v)$ and $(\boldsymbol{Q}_q, \boldsymbol{Q}_k, \boldsymbol{Q}_v)$. In the support branch, masked self-attention is performed between support tokens $(\boldsymbol{S}_q, \boldsymbol{S}_k)$. Let $\boldsymbol{A}_S \in R^{HW \times HW}$ denote the matrix of masked self-attention scores obtained via:

$$\boldsymbol{A}_S = (\boldsymbol{S}_q + \boldsymbol{\mathcal{M}})\, \boldsymbol{S}_k^{\mathrm{T}}. \tag{4.1}$$

To obtain corss-image relation map $\boldsymbol{R}_S$, it conducts cross-attention with the query token $\boldsymbol{Q}_v$

to enhance the feature representation of the target object. In the query branch, cross-attention encodes the local semantic information of the support objects into the query feature in a similar way:

$$\boldsymbol{A}_Q = \boldsymbol{Q}_q \boldsymbol{Q}_k^{\mathrm{T}}, \tag{4.2}$$

$$\boldsymbol{R}_Q = \mathrm{Softmax}\left(\frac{\boldsymbol{A}_Q}{\sqrt{d}}\right)(\boldsymbol{S}_v + \boldsymbol{\mathcal{M}}), \tag{4.3}$$

where the support mask is applied on the support token to remove the background.

The obtained cross-image relation maps $\boldsymbol{R}_Q$ and $\boldsymbol{R}_S$ are then fed to the MLP block to further encode the cross-image common information into local regions as:

$$\begin{cases} \hat{\boldsymbol{F}}_l^S = \mathrm{MLP}\left(\mathrm{Norm}\left(\boldsymbol{R}_S \odot F_l^S\right)\right) \\ \hat{\boldsymbol{F}}_l^Q = \mathrm{MLP}\left(\mathrm{Norm}\left(\boldsymbol{R}_Q \odot F_l^Q \odot \boldsymbol{R}_S\right)\right), \end{cases} \tag{4.4}$$

where Norm represents Layernorm, $\odot$ denotes the element-wise multiplication and the MLP block contains of two transformation layers with GELU non-linearity. Different from the proposed MCE in Chapter 3, the modified module outputs the enhanced query feature $\hat{F}_l^Q$ by merging the cross-image relation map $\boldsymbol{R}_Q$ and $\boldsymbol{R}_S$ to its input feature F_l^Q for subsequent segmentation. The the enhanced support feature $\hat{F}_l^S$ is used to generate the novel class prototype P_0 by mask average pooling as:

$$P_0 = \frac{\sum_{p,q} \hat{F^S}_{p,q} \cdot M_{p,q}}{\sum_{p,q} M_{p,q}} \tag{4.5}$$

where M represents the binary mask, where 1 indicates regions of interest and 0 indicates regions to be ignored, p and q denote the spatial location of the output feature map.

4.2.3 Multiple Semantic Prototypes Matching

Motivation Currently, most of the FSS methods typically construct a single novel class prototype vector [35], [123], [124] using masked average pooling in the support branch to guide a query feature. Improved versions [124], [144], [149] have attempted to use cluster-based algorithms to create multiple prototype vectors that can capture diverse and fine-grained support object features. However, many of these methods tend to treat the few-shot segmentation problem as a densely 1-way classification task, where the focus is primarily on constructing foreground feature prototypes, and fail to fully consider the relationships between base classes

and the novel class. Despite AGNN [150] constructing a graph network to examine semantic similarities between two data instances, effectively separating objects with high similarity to the target class visual representation remains challenging without prior class semantic knowledge.

The multiple semantic prototypes learning. The idea of utilizing multiple prototypes is examined in APANet [151] to indirectly promote foreground prototype matching. The APANet involves clustering background prototypes from query images to create negative feature-prototype pairs, thereby encouraging query features to exhibit higher confidence towards the foreground prototype. The strategy is then discarded during meta-testing because those background prototypes are unavailable as the query mask is not provided. Conversely, the rationale behind our multi-prototype scheme centers on the suppression of irrelevant class object regions that exhibit greater similarity to base prototypes. Our multi-prototype generation process is designed to obtain discriminative base class prototypes that can be directly utilized to filter background areas during meta-testing. These prototypes offer a broader and more diverse set of class information, thereby enhancing the network's ability to conduct accurate support-query matching.

It is composed of two primary steps: prototype generation and relation loss calculation. Given the base class images and their corresponding masks, the prototype generation step extracts feature representations through a pre-trained network and subsequently utilizes the masked average pooling operation to compute the vectorial representation P_i^c for the c-th class in the i-th image as:

$$\mathbf{P}_i^c = \frac{\sum_1^{HW} \mathbf{F}_i^{x,y} \cdot \mathbb{1}\left[M_i^{x,y} = c\right]}{\sum_1^{HW} \mathbb{1}\left[M_i^{x,y} = c\right]}, \tag{4.6}$$

where $\mathbf{F}_i^{x,y}$ denotes the learned features for the pixel at (x, y) in the i-th image and $M_i^{x,y}$ indicates its segmentation mask of class c. A generalized prototype P^c of base class c is then obtained by computing the average vectorial representations as:$P^c = \frac{1}{N_c} \sum_{N_c}^{i=1} P_i^c$, where N_c is the number of images containing class object c. The multiple class prototypes of both novel class and base class are derived using masked average pooling on the support set and base class training set. These semantic descriptors are then used to perform explicit multi-class matching to densely assign query features to the most likely class.

To enhance the discriminability of the vectorial representations of all base classes, a contrastive learning strategy is employed by utilizing the derived prototypes. This process involves computing the average cosine similarity among all possible pairs of two classes. The average similarity

loss L_p is formulated as:

$$L_p = \frac{\sum_{c_s}^{n} \sum_{c_t}^{n} \mathrm{Sim}\left(P^{c_s}, P^{c_t}\right) \mathbb{1}\left[c_s \neq c_t\right]}{\sum_{c_s}^{n} \sum_{c_t}^{n} \mathbb{1}\left[c_s \neq c_t\right]} \tag{4.7}$$

$$\mathrm{Sim}\left(P^{c_s}, P^{c_t}\right) = \frac{P^{c_s} \cdot P^{c_t}}{\|P^{c_s}\| \cdot \|P^{c_t}\|} \tag{4.8}$$

where c_s and c_t denote the prototype index. In this process, similarity calculations are carried out for all possible combinations of prototype pairs, with the pairs being switched between each other. The resulting prototype similarity values are then averaged to compute the relation loss, denoted as L_p. The primary objective of L_p is to facilitate prototype separation during training. It achieves this by encouraging the network to minimize the similarity between different classes. By doing so, L_p contributes to enhancing the network's performance and prepares it for more accurate matching.

Multi-prototype matching. After obtaining the base class prototypes $\{P^1, P^2, ..., P^n\}$, we freeze the backbone for the following meta-learning stage. In order to precisely activate the query features, we first extract the novel class prototype P^0 by performing masked average pooling on the support image feature. Then, we calculate the cosine similarity between all prototypes $P_a = \{P^0, P^1, P^2, ..., P^n\}$ and the query feature $\hat{F}_l^Q$ at each spatial location (x, y) to get the affinity matrix $A_i^{xy} \in \mathbb{R}^{H_l W_l \times N}$ as:

$$\mathbf{A}_i^{xy} = \cos(\hat{F}_l^Q(x, y), P_a) \tag{4.9}$$

Using the affinity matrix, we determine the activation value for each spatial location by referring to the index with the highest similarity score. The process of generating the prototype similarity map G can be formulated as:

$$idx = \mathrm{argmax}(\mathbf{A}_i^{xy}), \tag{4.10}$$

$$\mathbf{G}^{xy} = \begin{cases} \mathbf{A}_i^{xy} & idx = 0 \\ 0 & \text{otherwise} \end{cases}, \tag{4.11}$$

4.2.4 Target-aware Class Activation Map

The class activation map (CAM) serves as a valuable tool in semantic segmentation tasks by offering insights into the regions of the input image that play a significant role in identifying specific classes. Given that the backbone is pretrained on ImageNet and target class objects

are enhanced in the MCE, generating a CAM from the encoder's output embedding is likely to localize regions in the input image relevant to the target class.

Given this insight, we opt for the Class Activation Map (CAM) as the primary attention map generator to pinpoint prominent object regions in the query image. To be precise, we utilize the class-specific feature maps generated by the final convolutional layer of the encoder to produce attention maps, a process that demands minimal effort owing to the utilization of pre-trained backbone network knowledge.

However, due to the inherent nature of the Few-Shot Segmentation task, attention maps in different training iterations tend to emphasize various regions of the target objects. To address this variability, we propose an online attention-generation strategy that adaptively predicts a weight score map to refine the corresponding class activation map.

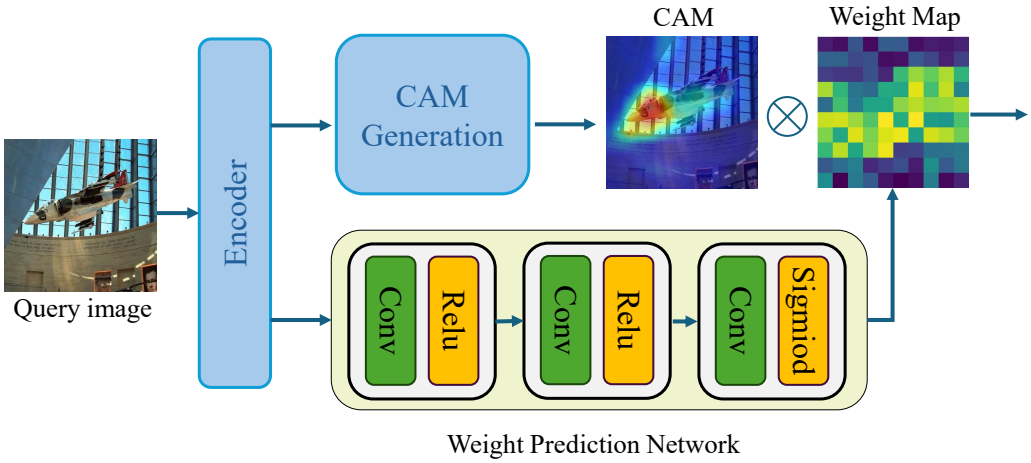


Figure 4.5: Target-aware class activation map (TCAM) learning. The weight prediction network $\phi()$ takes the query image feature extracted from the encoder as input and learns to predict a weight map that can be applied to adaptively enhance or suppress CAM activation values.

In pursuit of this approach, we introduce a lightweight CNN network denoted as $\Phi()$ during the meta-training stage as depicted in Figure 4.5. The purpose of this network is to predict an adaptive CAM weight map, denoted as W_{adp}, which can selectively activate the target object pixel values within CAM. Specifically, the query feature F_l^Q, extracted from the model backbone, is fed into the lightweight network $\Phi()$ to derive the object target weight map. Consequently, the CAM is adaptively adjusted as follows:

$$M_{tcam} = \text{Norm}(M_{cam} \cdot \Phi(f^Q)) \tag{4.12}$$

where $\cdot$ represents element-wise multiplication, $Norm$ donates Min-max normalization.

Although the weight prediction network operates as an auxiliary branch, it does not impose an additional learning objective. Instead, it learns to extract instance-specific information that aids in localizing target object regions conditioned on the segmentation loss function.

4.2.5 Hierarchical Guidance

The hierarchical pyramid strategy proves valuable in image segmentation tasks, especially when confronted with complex scenes and objects of varying scales. Images encompass objects ranging from intricate details to prominent elements, necessitating a progressive refinement process. The hierarchical structure facilitates this refinement, with lower levels capturing coarse information about larger objects. As the resolution increases, the model can then concentrate on finer details and delineate boundaries between objects. Consequently, we advocate for a hierarchical guidance structure that aligns prototypes and features across different scales to refine pixel-wise feature alignment. In other words, pixels that generate high similarity scores to the foreground prototype across multiple scales of features are more likely to be activated in the final segmentation mask.

Initially, the multi-scale intermediate feature maps $\{(F_l^s, F_l^q)\}_{l=1}^3$ from l stages are collected from the encoder. Following this, multiple prototype similarity maps are calculated using multi-prototype matching at corresponding resolutions within the SMM module. Subsequently, the HMMNet incorporates a pyramid decoder to predict the final segmentation mask with the hierachically guided feautres. Concretely, the multi-scale activation map $\{M_{tcam}\}_{l=1}^3$ and the prototype similarity maps $G_{l=1}^3$ are concatenated with the query features $\hat{F}_l^Q$ to provide guide information at a specific scale l. These combined features are subsequently aggregated by a 1×1 convolutional operation, resulting in the generation of three fused query features $\{\tilde{F}_l^Q\}_{l=1}^3$ across multiple scales. Finally, we incorporate the information of all scale features for hierarchical segmentation as:

$$\tilde{\mathbf{F}}'^Q = \mathcal{I}(\mathcal{F}_{1\times1}(\mathcal{I}\left(\tilde{F}_3^Q\right) \oplus \tilde{F}_2^Q)) \oplus \tilde{F}_1^Q \tag{4.13}$$

where $\mathcal{I}(\cdot)$ indicates the interpolation operation, $\mathcal{F}_{1\times1}$ represents the convolution with kernel size 1×1, and $\oplus$ denotes the concatenation along the channel dimension.

4.2.6 Loss Function

We adopt the cross-entropy loss as our main segmentation loss function denoted as L_{seg}. Similar to Eq.4.7, one of the optimization objectives is to ensure that the novel class prototype remains distinct from all base prototypes. This is accomplished by minimizing the average similarity loss $L_p = \frac{\sum_{c=1}^{n} \mathrm{Sim}\left(P^0, P^c\right)}{n}$. The total loss L is a balanced sum of L_{seg} and L_p as:

$$L = L_{seg} + \mu L_p, \tag{4.14}$$

where the weight coefficient μ is assigned a value of 0.3.

4.3 Experiments

4.3.1 Datasets

We evaluate the proposed method on the three most widely adopted few-show semantic segmentation datasets: PASCAL-5^i [129] and COCO-20^i [45]. The PASCAL-5^i is an extension of the PASCAL VOC 2012 dataset [44] and the SDS [130], which has been partitioned into four subsets. It contains 20 object classes, from which five classes are sampled and designated as the test label-set $D_{test} = 4i{+}1, \ldots, 4i{+}5$, where i denotes the fold number. The remaining 15 classes are then used to form the training label-set D_{train}.To facilitate evaluation on a more challenging dataset than PASCAL-5^i, the COCO-20^i dataset has been created. It contains 80 object classes and features ground-truth segmentation masks with lower quality compared to those in PASCAL VOC. Specifically, 60 classes are utilized for training, while the remaining 20 classes are designated as the test classes. In FSS-1000 [48], the 1,000 object classes are separated into distinct training, validation, and testing sets. The split uses a balanced proportion of 520 classes for training, 240 classes for validation, and another 240 classes for testing.

4.3.2 Eavluation Metrics

We adopt mean intersection over union (mIoU) as the evaluation metrics of our experiments. IoU for class c is defined as $IoU_c = \frac{TP_c}{TP_c+FP_c+FN_c}$, where TP, FP and FN are the number of true positives, false positives and false negatives of the predicted pixels. FB-IoU calculates the foreground-background average IoU as $FB - IoU = \frac{1}{2}\left(\mathrm{IoU_F} + \mathrm{IoU_B}\right)$.

Table 4.1: The class mIoU results are reported for each Fold, with MeanIoU(%) representing the average class mIoU and FB-IoU for averaged foreground-background IoU across four folds for 1-shot and 5-shot segmentation on PASCAL-5^i. BAM* presents the performance of the meta-learner.

Backbone	Method	1-shot						5-shot					
		Fold-0	Fold-1	Fold-2	Fold-3	MeanIoU(%)	FB-IoU(%)	Fold-0	Fold-1	Fold-2	Fold-3	MeanIoU(%)	FB-IoU(%)
VGG16	ASR(CVPR'21)[152]	50.2	66.4	54.3	51.8	55.7	72.9	53.7	68.5	55.0	54.8	58.0	74.1
	PFENet(TPAMI'20)[36]	56.9	68.2	54.4	52.4	58.0	72.0	59.0	69.1	54.8	52.9	59.0	72.3
	HSNet(CVPR'21)[126]	59.6	65.7	59.6	54.0	59.7	73.4	64.9	69.0	64.1	58.6	64.1	76.6
	BAM(CVPR'22)[39]	63.2	70.8	66.1	**57.5**	64.7	**77.3**	**67.4**	**73.1**	70.6	64.0	**68.8**	**81.1**
	MCE(ICME'23)[153]	60.6	69.50	65.1	56.3	62.9	74.5	65.6	72.8	69.7	64.7	68.2	78.2
	HMMNet(Ours)	**64.5**	**70.8**	**67.4**	56.1	**64.7**	76.8	66.8	72.8	**70.6**	**64.2**	68.6	80.7
ResNet-50	PFENet(TPAMI'20)[36]	61.7	69.5	55.4	56.3	60.8	73.3	63.1	70.7	55.8	57.9	61.9	73.9
	CyCTR(NIPS'21)[102]	67.2	71.1	57.6	59.0	63.7	-	71.0	75.0	58.5	65.0	67.4	-
	MMNet(ICCV'21)[154]	62.7	70.2	57.3	57.0	61.8	-	62.2	71.5	57.5	62.4	63.4	-
	HSNet(ICCV'21)[126]	64.3	70.7	60.3	60.5	64.0	76.7	70.3	73.2	67.4	67.1	69.5	80.6
	NTRENet(CVPR'22)[155]	65.4	72.3	59.4	59.8	64.2	77.0	66.2	72.8	61.7	62.2	65.7	78.4
	BAM(CVPR'22)[39]	69.0	73.6	**67.6**	61.1	67.8	-	70.6	75.0	70.8	67.2	70.9	-
	ABCNet(CVPR'23) [156]	68.8	73.4	62.3	59.5	66.0	76.0	71.7	74.2	65.4	67.0	69.6	80.0
	FECANet(TMM'23) [38]	**69.2**	72.3	62.4	65.7	67.4	78.7	**72.9**	74.0	65.2	**67.8**	70.0	80.7
	MCE(ICME'23) [153]	65.3	71.2	66.2	61.0	65.9	78.1	69.2	73.7	70.5	66.8	70.0	81.3
	HMMNet(Ours)	68.8	**74.9**	67.0	**61.7**	**68.1**	**80.2**	71.8	**75.6**	**71.3**	67.4	**71.5**	**81.7**
ResNet-101	PFENet(TPAMI'20) [36]	60.5	69.4	54.4	55.9	60.1	72.9	62.8	70.4	54.9	57.6	61.4	73.5
	CyCTR(NIPS'22) [102]	67.2	71.1	57.6	59.0	63.7	-	71.0	75.0	58.5	65.0	67.4	-
	NTRENet(CVPR'22) [155]	65.5	71.8	59.1	58.3	63.7	75.3	67.9	73.2	60.1	66.8	67.0	78.2
	SCCAN(ICCV'23) [157]	**70.9**	73.9	66.8	61.7	68.3	78.5	**73.1**	**76.4**	70.3	66.1	71.5	82.1
	ABCNet(CVPR'23) [156]	65.3	72.9	65.0	59.3	65.6	78.5	71.4	75.0	68.2	63.1	69.4	80.8
	HMMNet(Ours)	70.1	**75.3**	**67.5**	**62.0**	**68.7**	**81.4**	72.6	75.9	**71.8**	**67.8**	**72.0**	**82.6**

4.3.3 Implementation Details

Multiple semantic prototypes learning: The pre-trained (on ImageNet) VGG-16 and ResNet50 are used as backbone networks for feature extraction. The prototypes used in our method consist of the online novel class prototype and offline base class prototypes. The former is obtained from the support images using masked average pooling during the meta-training stage and the latter is generated on images containing the base classes in the prior prototype learning stage. To ensure the consistency of features used for both novel and base prototype construction, the backbone parameters are kept frozen after the base prototype learning stage. The number of base prototypes is 15 on PASCAL-5^i, 60 on COCO-20^i and 540 on FSS-1000, based on the statistics of categories in these datasets.

Baseline. We adopt a modified version of PFENet [36] as our baseline method, wherein the feature enrichment module (FEM) is substituted with atrous spatial pyramid pooling (ASPP) [21] to expedite training

K-shot setting Given a support set $S = \{(I^s, M^s)\}_1^k$ and a query image $Q = (I^q, M^q)$, we take the average vectorial representation of K support image as the novel class prototype P_0,

Table 4.2: The class mIoU results are reported for each Fold, with MeanIoU(%) representing the average class mIoU across four folds for 1-shot and 5-shot segmentation on COCO-20^i. BAM* presents the performance of the meta-learner.

Backbone	Method	1-shot					5-shot				
		Fold-0	Fold-1	Fold-2	Fold-3	MeanIoU(%)	Fold-0	Fold-1	Fold-2	Fold-3	MeanIoU(%)
ResNet-50	PFENet(TPAMI'20)[36]	36.5	38.6	34.5	33.8	35.8	36.5	43.3	37.8	38.4	39.0
	HSNet(ICCV'21)[126]	36.3	43.1	38.7	38.7	39.2	43.3	51.3	48.2	45.0	46.9
	CyCTR(NeurIPS'22)[102]	38.9	43.0	39.6	39.8	40.3	41.1	48.9	45.2	47.0	45.6
	FECANet(TMM'23) [38]	38.5	44.6	42.6	40.7	41.6	44.6	51.5	48.4	45.8	47.6
	BAM(CVPR'22) [39]	**43.4**	50.6	47.5	43.4	46.2	49.3	54.2	51.6	49.6	51.2
	NTRENet(CVPR'22) [155]	36.8	42.6	39.9	37.9	39.3	38.2	44.1	40.4	38.4	40.3
	ABCNet(CVPR'23) [156]	42.3	46.2	46.0	42.0	44.1	45.5	51.7	52.6	46.4	49.1
	MCE(ICME23) [153]	42.1	48.3	43.7	42.8	44.2	47.8	55.2	50.8	50.3	51.0
	HMMNet(Ours)	41.2	**52.7**	**48.5**	**47.4**	**47.5**	**49.8**	57.3	**53.6**	**50.1**	**52.7**
ResNet-101	PFENet(TPAMI'20) [36]	34.3	33.0	32.3	30.1	32.4	38.5	38.6	38.2	34.3	37.4
	CWT(ICCV'21) [158]	30.3	36.6	30.5	32.2	32.4	38.5	46.7	39.4	43.2	42.0
	NTRENet(CVPR'22) [155]	38.3	40.4	39.5	38.1	39.1	42.3	44.4	44.2	41.7	43.2
	DCAMA(ECCV'22) [104]	41.5	46.2	45.2	41.3	43.5	48.0	58.0	54.3	47.1	51.9
	SCCAN(ICCV'23) [157]	42.6	51.4	**50.0**	48.8	48.2	49.4	**61.7**	**61.9**	55.0	**57.0**
	HMMNet(Ours)	**43.3**	**53.4**	49.5	**49.6**	**49.0**	**50.3**	61.2	58.1	**56.0**	56.4

similar to the base class generation process introduced in Section 4.2. To make full use of the K support images, support features $\{F_l^S\}_1^k \in \mathbb{R}^{H_l W_l \times C}$ from the backbone are fused into $\{F_l'^S\}_1^k \in \mathbb{R}^{H_l W_l \times KC}$ through concatenation operation along channel dimension.

All the experiments are conducted on Pytorch platform, using a server equit with Intel Xeon Gold 6226R CPU and Nvidia Quadro RTX 6000 GPU. We use Stochastic Gradient Descent (SGD) as the optimizer, which we apply the "ploy" learningl rate scheduler with the momentum and weight decay of 0.9 and 10^{-5}, respectively. The model was trained for 300 epochs with a base learning rate of 0.0025 and batch size 16 on PASCAL 5^i. For COCO-20^i, models were trained for 150 epochs with a base learning rate of 0.005 and batch size 8. FSS-1000 is trained for 100 epochs using initial learning rate of 0.01 and batch size 32.

We utilize Stochastic Gradient Descent (SGD) as the optimizer with a "ploy" learning rate scheduler. The momentum and weight decay are set to 0.9 and 10^{-5}. Our model trained on PASCAL-5^i for 300 epochs with a learning rate 0.0025 and batch size 16, while these parameters on COCO-20^i experiment are 50, 8 and 0.05 respectively. We train the model on FSS-1000 for 150 epochs with base learing rate of 0.001 and batch size 32. In the training stage, we follow [36] to randomly crop or zero pad 473×473 patches from the processed images as training samples for PASCAL-5^i and COCO-20^i, 224×224 for FSS-1000. The experiments are conducted Pytorch

Table 4.3: FB-IoU results on FSS-1000

Methods	Backbone	1-Shot	5-Shot
OSLSM [33]	VGG-16	70.3	73.0
GNet [135]		71.9	74.3
FSS1000 [48]		73.5	80.1
PFENet [36]		81.5	82.7
HSNet [126]		82.3	85.8
MCE [153]		83.8	86.2
HMMNet(Ours)		**86.5**	**88.3**
PFENet [36]	ResNet-50	84.6	86.1
HSNet [126]		85.5	87.8
MCE [153]		86.6	88.2
HMMNet(Ours)		**89.2**	**89.7**

platform, using a server with Intel Xeon Gold 6226R CPU and Nvidia Quadro RTX 6000 GPU.

4.3.4 Comparison Experiments

We compare the proposed method with other state-of-the-art methods using different backbones including VGG16 [159] and ResNet50 [160]on PASCAL-5^i and COCO-20^i in both 1-shot and 5-shot settings.

PASCAL-5^i: In terms of the mIoU performance in both 1-shot and 5-shot settings using ResNet-50 as the backbone, our method outperforms previous approaches on the PASCAL-5i dataset, as shown in Table 4.1. Specifically, compared to the state-of-the-art performance achieved by BAM and FECANet, our method achieves a performance gain of 0.3% and 0.7% in the 1-shot setting on ResNet-50, respectively. This stems from the adoption of a direct yet efficient approach, which generates multiple base class prototypes to suppress base object regions. Consequently, we are able to explicitly identify the category of each spatial location within the query feature. In fold-2 split, encompassing novel categories like “dog,” “horse,” “motorbike,” and “person,” our method exhibits superior performance compared to other approaches, achieving a minimum improvement of 1.3% with the VGG16 backbone in the 1-shot scenario. Notably, these categories frequently occur in the background of the training set (base classes), emphasizing our model’s capability to enhance the discriminative power of novel classes against base classes, particularly those regarded as background elements. Moreover, in

the 5-shot scenario, our method demonstrates substantial growth and fares favorably against alternative approaches

COCO-20^i: The dataset presents greater challenges compared to the PASCAL-5^i dataset, primarily because of its larger number of categories and the complexity of scenes it encompasses. As shown in Table 4.2, our approach surpasses all the advanced methods by a large margin. Notably, it exhibits a significant performance gain over two cross-attention-based models, CyCTR and FECANet, by 6.7% and 5.9% respectively, under the 1-shot setting, as measured by mean class mIoU metric when using ResNet-50 as the backbone. In the 5-shot setting, our method also outperforms all methods, demonstrating the superior class discrimination ability when encountering a wider range of classes.

FSS-1000: FSS-1000 comprises a larger number of classes and provides only foreground labels for each image, aligning with the tailored setting for FSS, wherein the objective is to predict a binary mask for the novel class. The comparison results are presented in Table 4.3. HMMNet achieves the new state-of-the-art performance in terms of FB-IoU with both VGG-16 and ResNet-50 backbones, across both 1-shot and 5-shot experiments. It surpasses the MCE method introduced in Chapter 3 by substantial margins of 2.6% and 1.5% when utilizing one and five support images, respectively. These findings demonstrate the consistent effectiveness of the proposed hierarchical multi-prototype matching scheme.

Qualitative Evaluation: We visualize the segmentation results of the baseline, CyCTR and our HMMNet in Figure 4.6. CyCTR is the most closely related work to our study which also introduces the cross-attention mechanism to explore image relations. However, our model differs from CyCTR in two main aspects. Firstly, our model utilizes Masked Cross-image Encoding (MCE) to mix support and query features containing the same class as a uniform feature source for generalizing novel class semantic information, which implements bidirectional cross-attention with different image features. In contrast, CyCTR treats support and query features as separate entities and uses standard self-attention to encode query features for unidirectional cross-alignment. Therefore, the model becomes more proficient in accurately discovering the target regions like "plane" without including background regions of the sky. Secondly, our model employs multi-prototype matching that helps to filter irrelevant regions of base classes. As illustrated in the second column of Figure 4.6, our approach performs better in segmenting query images containing multiple objects (e.g., person and bike), where the leg part of the

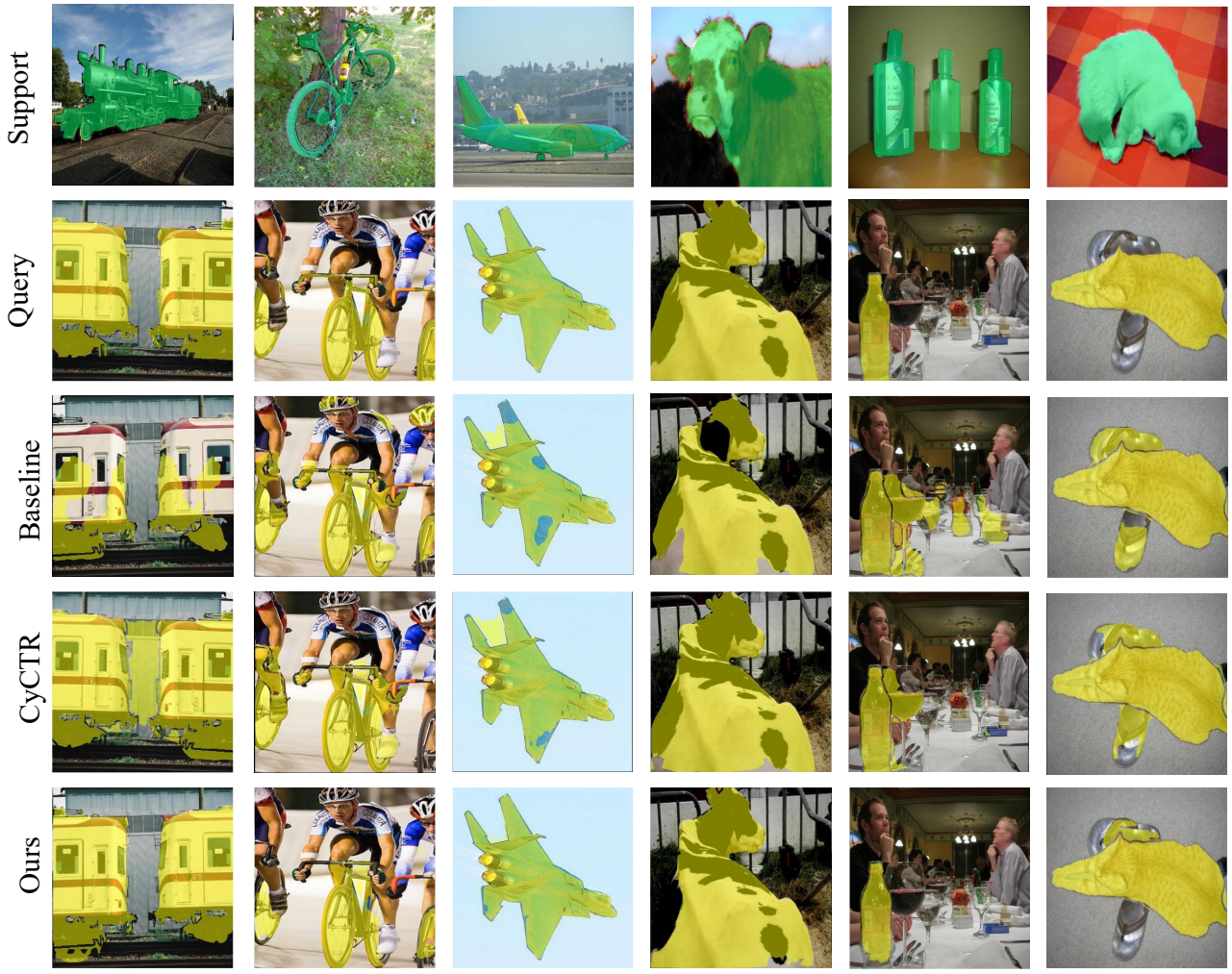


Figure 4.6: Qualitative comparison results of baseline, CyCTR[102] and our method. The yellow shades indicate the predicted area of the target novel class, and the corresponding ground truth is in the case of Query.

person is excluded compared to baseline and CycTR.

Effectiveness on cross-domain few-shot segmentation. To assess the effectiveness and robustness of our method in cross-domain few-shot segmentation, we conduct experiments where the model is trained on COCO-20^i training folds and tested on the non-overlapping testing set of PASCAL, as utilized in prior work [134]. For the COCO to FSS-1000 cross-domain experiment, we randomly sample 240 classes that do not overlap with the COCO training classes to form the testing set. We report the class mIoU and FB-IoU for the COCO $\rightarrow$ PASCAL and COCO $\rightarrow$ FSS-1000 experiments, respectively.

Our approach achieves superior performance in cross-domain semantic segmentation for both 1-shot and 5-shot tasks across the domain shift scenarios. Specifically, we surpass the tradi-

Method	COCO → PASCAL		COCO → FSS-1000	
	1-shot	5-shot	1-shot	5-shot
ASGNet [124]	57.4	66.6	72.3	75.6
PFENet [36]	60.8	61.9	70.6	74.2
RePRI [134]	63.2	67.7	68.4	69.1
SLC [161]	49.1	60.3	76.0	78.8
HSNet [161]	61.5	65.4	75.5	80.1
ABCNet [156]	62.9	68.2	78.9	81.0
HMMNet(Ours)	**65.3**	**69.5**	**82.2**	**86.7**

Table 4.4: Cross-domain few-shot semantic segmentation results on COCO to PASCAL and COCO to FSS-1000. COCO → PASCAL reports the class mIoU and COCO → FSS-1000 presents FB-IoU. ResNet-50 is employed as backbone for all models.

tional single-prototype state-of-the-art few-shot semantic segmentation method ABCNet [156] by 2.4% and 3.3% under the 1-shot setting in the two challenges. Furthermore, our method outperforms the correlation-oriented method HSNet by a significant margin, with over 4.1% mIoU improvement for 5-shot in the COCO → PASCAL experiment and 6.6% FB-IoU improvement for 5-shot in the COCO → FSS-1000 experiment. These noteworthy results suggest that the proposed multi-prototype matching strategy can effectively handle cross-domain scenarios better than other optimization schemes. This improvement can be attributed to the fact that base prototypes learned from one domain may not be present in images from other domains, leading to the suppression of these base classes during matching with the base prototype and thereby resulting in better segmentation performance.

4.4 Ablations

In this section, we conducted all ablation studies on PASCAL-5^i benchmark under the 1-shot setting using ResNet-50 as the backbone. First, we investigate the effectiveness of the proposed modules by observing the overall mIoU after removing one of them. Then we study possible variants of cross-attention of MCE to find out the most effective scheme. Finally, we discuss the influence of employing TCAM on segmenting specific classes.

MCE	Multi-Prototype	TCAM	Mean%	Params	Speed (FPS)
	✓	✓	66.73	8.3 M	24.5
✓		✓	65.34	18.4 M	19.2
✓	✓		67.12	18.4 M	20.3
✓	✓	✓	68.04	19.6 M	18.6

Table 4.5: The ablation results of module performance on PASCAL-5^i under 1-shot setting. "Params" refers to learnable parameters.

4.4.1 Effectiveness of Components

Table 4.5 presents the ablation study results, underscoring the influence of different components in our suggested methods. The bottom row represents the fully integrated model comprising all modules, while the rows above indicate the performance impact when each module is individually excluded. Notably, the multi-prototype matching significantly mitigates the impact of irrelevant false-positive regions from the base class, leading to a performance enhancement of 2.7% in segmentation. Impressively, this multi-prototype module does not introduce additional parameters and only necessitates minimal memory to store the base prototype vectors. Excluding the cross-image encoding module results in a decrease in mIoU by 1.31%, emphasizing the critical role of exchanging support-query features for implicit feature guidance. However, MCE introduces an extra 11.3 million learnable parameters due to numerous attention operations. Moreover, integrating TCAM, designed to activate salient regions in the query image, furthers performance by nearly 1%. It's noteworthy that the lightweight prediction network for TCAM requires only 1.2 million parameters for training. In terms of inference speed, while multi-prototype matching minimally affects the speed, it delivers the most substantial improvement in performance. Discarding the MCE will result in an increase of the inference speed by approximately 6 frames per second (FPS) on average.

Qualitative segmentation examples of 1-shot segmentation on the PASCAL-5^i dataset are provided in Figure 4.7. When examining the results of the baseline model, it becomes apparent that the segmentation masks for the "person" and "sofa" classes are noticeably incomplete. However, upon integrating the MCE module into the network, the model demonstrates an improved ability to identify additional regions corresponding to "person" and "sofa" that were previously overlooked by the baseline methods. Furthermore, it is evident that the incorpora-

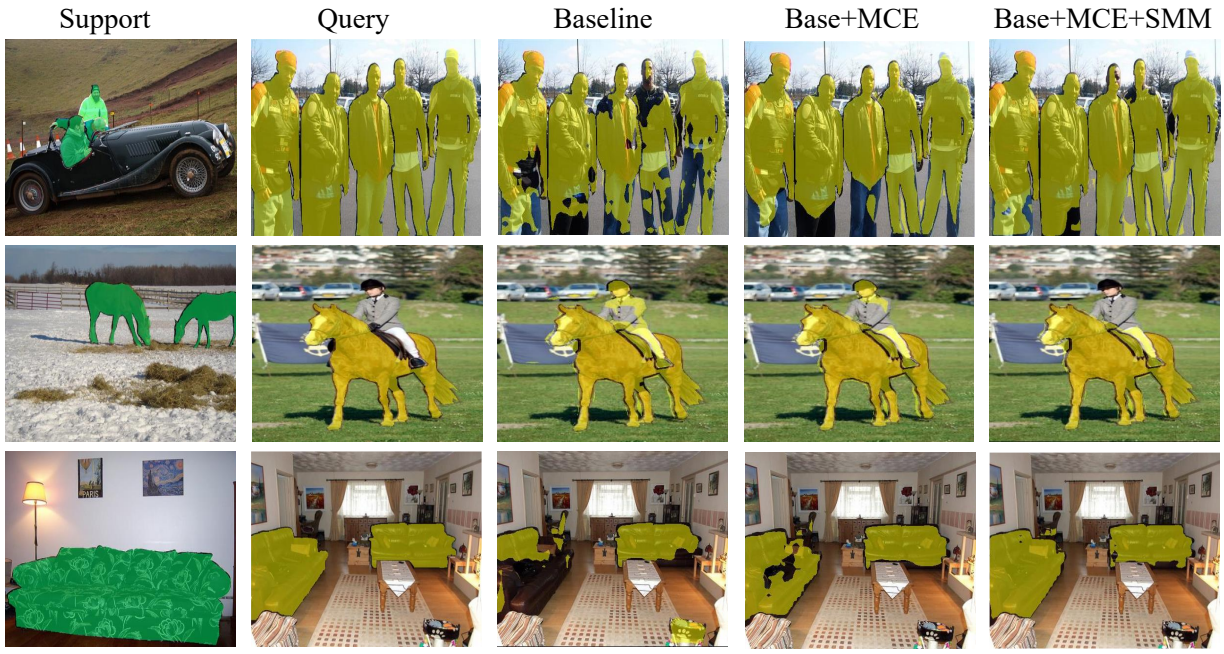


Figure 4.7: Qualitative results for component analysis. From the left to the right are masked support image, masked query image(Ground Truth), baseline, baseline+MCE and baseline+MCE+SMM

tion of the Semantic Multi-prototype Matching (SMM) module has a dual effect. Not only does it effectively filter out false-positive regions associated with the ”person” class in the second-row example, but it also serves to activate more parts of the target novel class (i.e. “person” and “sofa”) in the first and third rows. This increase in activation is attributable to the multi-prototype matching mechanism, which raises the similarity scores for those pixels associated with the target novel class.

4.4.2 TCAM Effectiveness

In the pixel-wise prototype matching paradigm of FSS, each spatial location of the query feature is evaluated independently. This approach inherently disregards the spatial correlation among adjacent pixels. Consequently, it leads to the phenomenon of scattered activation points with lower confidence in the feature map, as demonstrated in the ”w/o TCAM” column in Figure 4.8. The presence of these low-confidence points can have a potentially negative impact on the final segmentation results. It is clear that the proposed TCAM excites the main part of the target object with high confidence. By fusing the TCAM, our model is able to produce relatively pure feature activation maps as the activation score of target pixels is significantly increased with

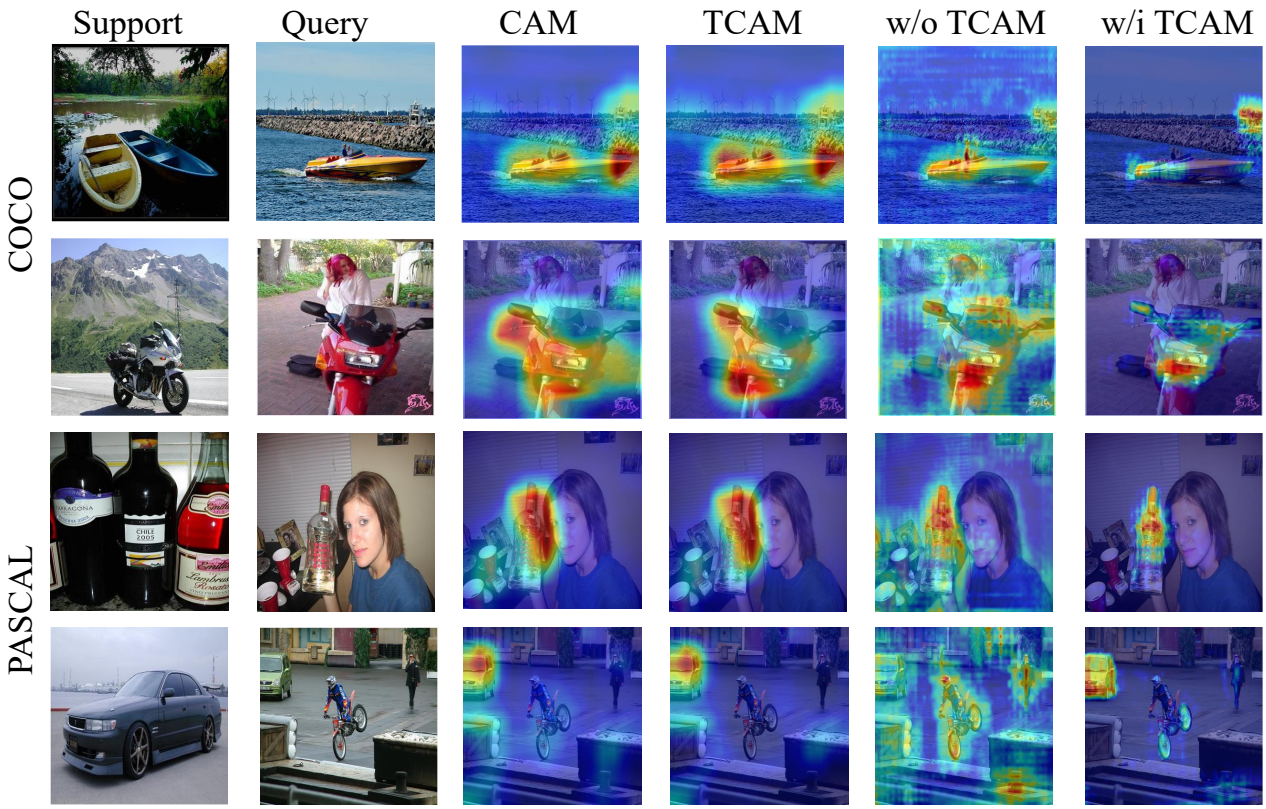


Figure 4.8: Qualitative visualization map. From the left to the right, are support image, query image, our proposed target-aware CAM, the guided query feature without fusing TCAM, and the final fused query feature on both PASCAL-5^i and COCO-20^i

TCAM. The introduced TCAM is evidently successful in activating the primary regions of the target object with high confidence. When incorporating TCAM, as depicted in the rightmost column of Figure 4.8, our model is able to generate notably pure feature activation maps, as the activation scores of target pixels are significantly enhanced with the assistance of TCAM.

4.4.3 Visual Content Impact

The presence of different object categories in an image can significantly impact the performance of semantic segmentation. We randomly select four sets of images from the PASCAL dataset, categorized based on the number of classes present within each image. As shown in Figure 4.9, we evaluate the performance of both the proposed multi-prototype matching and the conventional single-prototype matching methods across various scenarios characterized by different numbers of semantic content. The line chart illustrates that with an increase in the number of classes, there is a noticeable decline in mIoU. However, it's noteworthy that the

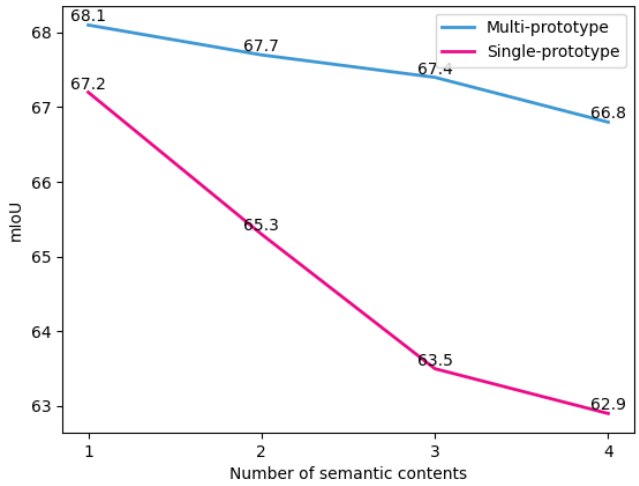


Figure 4.9: The influence of the number of visual content on segmentation performance

multi-prototype matching method exhibits a slower decrease compared to its single-prototype counterpart. This phenomenon can be attributed to the effectiveness of our matching strategy in managing complex scenarios featuring a wide range of classes.

4.5 Limitations

Limitations. Other than addressing issues of foreground feature alignment within the FSS framework, the proposed HMMNet introduces a novel multi-prototype matching strategy. This approach aims to filter background regions by employing base class prototypes. While it effectively addresses the influences of base objects in most query images, its effectiveness in enhancing segmentation performance might be reduced in scenarios where the background lacks objects from the base classes.

4.6 Chapter Summary

This chapter presents a comprehensive FSS approach aimed at addressing the inherent undermatching and mismatching issues within the support-query prototype matching framework. The proposed method introduces two effective modules, MCE and SMM, which serve to bridge the inter-class semantic gap between support prototypes and query features, and facilitate a clear distinction between similar novel and base classes. Furthermore, we have devised an adaptive class activation map to highlight salient regions that might otherwise be overlooked

during strict feature patch-wise alignment. Extensive experiments conducted on PASCAL-5^i and COCO-20^i datasets demonstrate that these proposed modules work synergistically and yield superior few-shot segmentation performance.

Chapter 5

Task Consistent Prototype Learning for Few-shot Incremental Semantic Segmentation

5.1 Introduction

The preceding two chapters have explored the conventional Few-Shot Semantic segmentation (FSS) task from two distinct perspectives of the FSS framework: feature representation and segmentation guidance. While the impressive results in segmenting novel classes suggest that engineers might reduce their time spent on dense labeling tasks, there is a notable limitation in the FSS technology. Specifically, it is constrained to segmenting just one single novel class at a time, neglecting other classes even if they have been identified in previous training, this limitation hinders their real-word application. Imagine a robot tasked with obstacle avoidance during floor cleaning. A FSS model could be trained to identify a new obstacle class, such as "ceramic vase" using just a few labeled images. While successful in navigating around ceramic vase, the robot might still collide with established obstacle classes, like "table", "sofa" even if previously trained on them. This hinders the robot's ability to perform comprehensive obstacle detection and navigation within a single environment.

Moving forward, researchers are actively addressing this limitation by developing multi-class FSS techniques. It is inherently anticipated that a model should exhibit the capability to segment both base and novel classes within an image, a concept referred to as Generalized

Few-shot Semantic Segmentation (GFSS) [162], [163]. However, this semantic segmentation settings typically operate within a fixed output space, where the number of target classes is predetermined and remains constant. This approach may not adequately meet the demands of real-world scenarios, where the total number of categories is uncertain, and new class objects may emerge over time, limiting the applicability and scalability of the model.

Taking a step further, this chapter delves into a more complex and applicable scenario where the model continuously encounters a stream of new image data containing instances of previously unseen classes.The task known as Incremental Few-Shot Semantic Segmentation (iFSS) in the existing literature [164]–[166], is inspired by few-shot class incremental learning (FSCIL) [167], [168]. Unlike conventional FSS, iFSS emphasizes a series of ongoing adaptation tasks. It aims to learn how to segment new classes with only a handful of annotated examples, while crucially retaining the knowledge of previously learned classes. In this way, the extendibility and flexibility of the model can be improved, which is critical for many real-world applications, such as autonomous driving and human-machine interaction.

The objective of iFSS is to update a model to effectively segment new classes using a few annotated samples while retaining its segmentation capability on existing seen classes. While FSCIL is a broad concept applicable to any task where new classes are incrementally introduced with few examples, iFSS specifically focuses on learning pixel-wise classification tasks for new classes. iFSS has more practical applications than FSCIL because it meets the demands of tasks that require fine-grained visual understanding and interaction. Its ability to handle dynamic environments, enhance human-computer interaction, improve decision-making, support robotic object manipulation, and facilitate detailed content creation makes it more broadly applicable in real-world scenarios.

iFSS shares two common challenges with FSCIL, namely catastrophic forgetting of learned knowledge and overfitting to a limited number of novel class examples. This arises due to the absence of access to previous session data during the incremental learning stage. When updating parameters with imbalanced novel class data (where the number of novel classes is considerably smaller compared to base classes), the model tends to exhibit a strong bias towards novel classes in pursuit of rapid adaptation. Consequently, there is a risk of aggressively overwriting crucial knowledge related to old classes in an attempt to accommodate the latest instances, resulting in a loss of generalization ability.

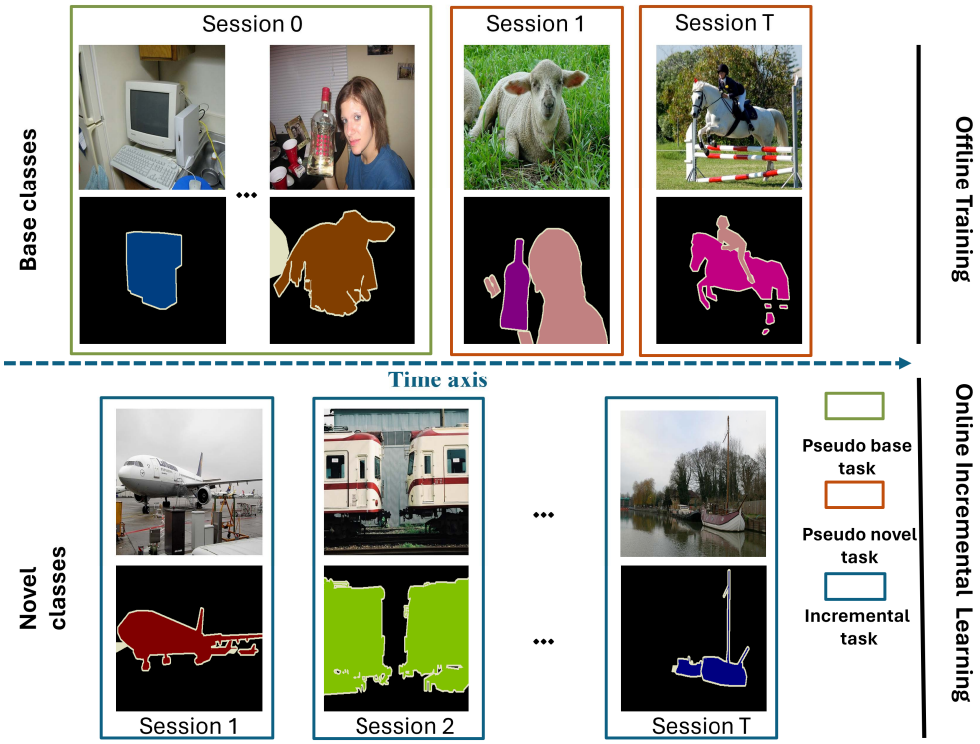


Figure 5.1: Illustration of the evaluation protocol and our meta-training process. During the online incremental learning stage, the model undergoes training solely on new classes within each incremental session, while evaluation is conducted on all classes encountered thus far. Our strategy aims to replicate this evaluation protocol during the offline base class training stage. This is accomplished by randomly sampling a large portion of base class images to constitute the pseudo base dataset, with the remaining classes forming the pseudo novel classes. Initially, the model trains on the pseudo base dataset and subsequently adapts to the pseudo novel classes. This approach enables the model to learn how to swiftly identify new classes while retaining the ability to segment previously encountered ones

Specifically, in the incremental Few-Shot Semantic Segmentation (iFSS) task, an initial base set containing a relatively larger number of training samples is provided to initialize the learnable parameters of a semantic segmentation model. Subsequently, a few pixel-level annotated training samples of novel categories are introduced, aiding in the incremental expansion of the model's segmentation capability to accommodate the encountered novel classes.

The challenges mentioned above inherently stem from the task misalignment inherent in mainstream few-shot class incremental learning (FSCIL) methods. Like FSCIL, the iFSS consists of an offline training stage and an online incremental learning stage. In the offline training stage, the model has access to a large-scale dataset for some base classes. FSCIL learns a model

on these base classes. During the online incremental learning (i.e. evaluation) stage, it will encounter novel classes in a sequential manner, where a few novel classes are presented at each time step (called incremental session). For each novel class, only a few training examples are provided. In addition, it can only access training examples corresponding to the novel classes at the current time step. In other words, we cannot store training examples from previous time steps during evaluation. The evaluation protocol is defined such that at each incremental session, after learning the novel classes, the model is evaluated on all encountered classes (including base classes).

Many incremental learning methods begin by initializing model parameters using fully supervised learning, utilizing ample samples during the base session to achieve optimal segmentation performance for base classes. Subsequently, they employ few-shot prototype learning strategies to rapidly adapt to novel classes during subsequent incremental sessions (evaluation steps). However, the inconsistent training objectives between the base class training stage and the expected evaluation criteria in the incremental stage inevitably compromise the model's ability to swiftly adaptation and resist forgetting. To alleviate forgetting and adaptation problem, we propose a meta-learning (e.g. [76]) based prototype learning approach that directly learns to incrementally adapt to novel classes conditioned on a few examples. This is achieved by simulating the incremental few-shot scenario during base session training. The base dataset is split into a pseudo base set and a pseudo incremental set. As shown in Figure 5.1, we create a sequence of pseudo incremental tasks by sampling a small subset of base classes. The remaining base classes are treated as the pseudo base set. For each pseudo task, consisting of a few labeled images (e.g., 1-5), the model undergoes fast adaptation and updates its parameters. We then evaluate the model's performance on test images from both the old and new classes. This process of sampling pseudo tasks is repeated until the model reaches convergence. In such a meta-learning pattern, we find a good starting point for the model so that it can learn new classes in a sequence without forgetting the old ones.

In addition to the training paradigm, there have been a few attempts to address the catastrophic forgetting and overfitting challenges by employing two types of methods: replay-based and regularization-based. In replay-based methods, samples of previous tasks are either stored or generated at first and then replayed when learning the new task. Zhu et al. [168] propose to store the same number of old samples as each new class to form a joint set during its incremental learning process. Regularization-based methods protect old knowledge from being

covered by imposing constraints on new tasks. For example, an intuitive solution proposed in FSCIL [167], [169] suggests fine-tuning the network on new session data with distillation loss to mitigate forgetting of old classes. However, the few-shot data in novel sessions can easily lead to overfitting, making it challenging to distill useful knowledge from the model of the previous step. Alternatively, some studies [170], [171] suggest to train a backbone network on the base session to serve as a feature extractor. In novel sessions, the backbone network remains fixed to preserve base class knowledge, while a set of novel-class prototypes (classifier vectors) are incrementally learned using the shared backbone features. However, the proximity of newly added prototypes to old-class prototypes may hinder the ability to discriminate between old-class and novel-class samples during evaluation.

To optimize the prototype generation process, we propose a Prototype Space Redistribution Learning (PSRL) to incrementally learn novel class prototypes and adaptively allocate base and novel prototypes into a latent prototype space, maintaining optimal prototype boundaries. Specifically, we fix the pre-trained feature backbone to preserve a unified feature extractor and introduce a prototype projector mapping intermediate class vectors to a subspace for dynamic prototype distribution. The redistribution process aims to enhance discrimination between new class prototypes and existing old class prototypes, thereby improving novel class segmentation performance. Furthermore, it regulates the updated base prototypes placed near their previous position to prevent prototype misalignment, effectively mitigating knowledge forgetting. The contributions of this chapter are summarized as:

- We propose a meta-learning optimization approach that aligns the base step learning objective closely with the evaluation protocol. This method directly optimizes the model to facilitate the discovery of novel objects while preserving its segmentation capability for previously encountered classes

- We present Prototype Space Redistribution Learning (PSRL), a method that projects class prototypes into a subspace where they are redistributed, taking into account inter-prototype discrimination while ensuring consistency among base prototypes. This approach alleviates catastrophic forgetting of base classes and facilitates rapid adaptation to novel classes.

- Extensive experiments on dedicated iFSS benchmark from PASCAL VOC and COCO datasets demonstrate the proposed method outperforms several counterparts.

5.2 Method

In this section, we introduce our proposed method for addressing iFSS, a problem that remains under-explored. We begin by providing a formal problem definition that elucidates the task setting. Next, we describe how a basic prototype-based model is trained for the iFSS problem. Building upon this base framework, we introduce the proposed Prototype Space Re-distribution Learning method and the associated meta-learning process, which aim to mitigate issues of catastrophic forgetting and class overfitting.

5.2.1 Problem setting

iFSS addresses the challenge of updating a pre-trained segmentation model to accommodate newly introduced classes over time, utilizing limited annotated examples for each novel class. Specifically, let $\mathcal{D}^t_{train/test} = \{\mathcal{I}^t_n, \mathcal{M}^t_n\}$, $n \in \{1, 2, \ldots, K\}, t \in \{1, 2, ..., T\}$, denote a sequence of the training and testing sets of image $\mathcal{I}^t_{train/test}$ and their corresponding semantic label masks $\mathcal{M}^t_{train/test}$. The label classes $\mathcal{C}^t$ of each set are disjoint, such that $\mathcal{C}^i \cap \mathcal{C}^j = \emptyset, \forall i \neq j$. iFSS comprises a base session with abundant labeled training images from D^0_{train} and a sequence of incremental sessions with only a few training images for each novel class from $\{D^1_{train}, D^2_{train}, ..., D^T_{train}\}$. We undertake offline training in the base session to initialize a model using base classes $\mathcal{C}^0$. After the base session, the model is expected to adapt to new classes $\mathcal{C}^t (t > 0)$ with a few examples in the subsequent incremental sessions. Note that at the t^{th} session, the model has access only to D^t_{train} for training and then is evaluated on test images containing all the encountered classes so far, i.e. $\{D^0_{test} \cup D^1_{test} ... \cup D^t_{test}\}$.

5.2.2 Prototype-based model for iFSS

Prototype-based models are extensively utilized in various segmentation tasks [36], [92], [162], [163]. These models learn a single weight or multiple representative vectors for each class to perform dense prediction. The segmentation process can be conceptualized as nearest prototype retrieval, wherein pixel-wise classification is accomplished by matching each pixel in the image to the closest prototypes. Unlike traditional deep learning models, which have a fixed output space, prototype-based models benefit from a flexible classification mechanism. The prototype classifier can dynamically expand without requiring adjustments to the model architecture. This property makes prototype-based models highly suitable for the iFSS setting, where the model encounters novel classes over time. As depict in Figure 5.2, a typical prototype-based

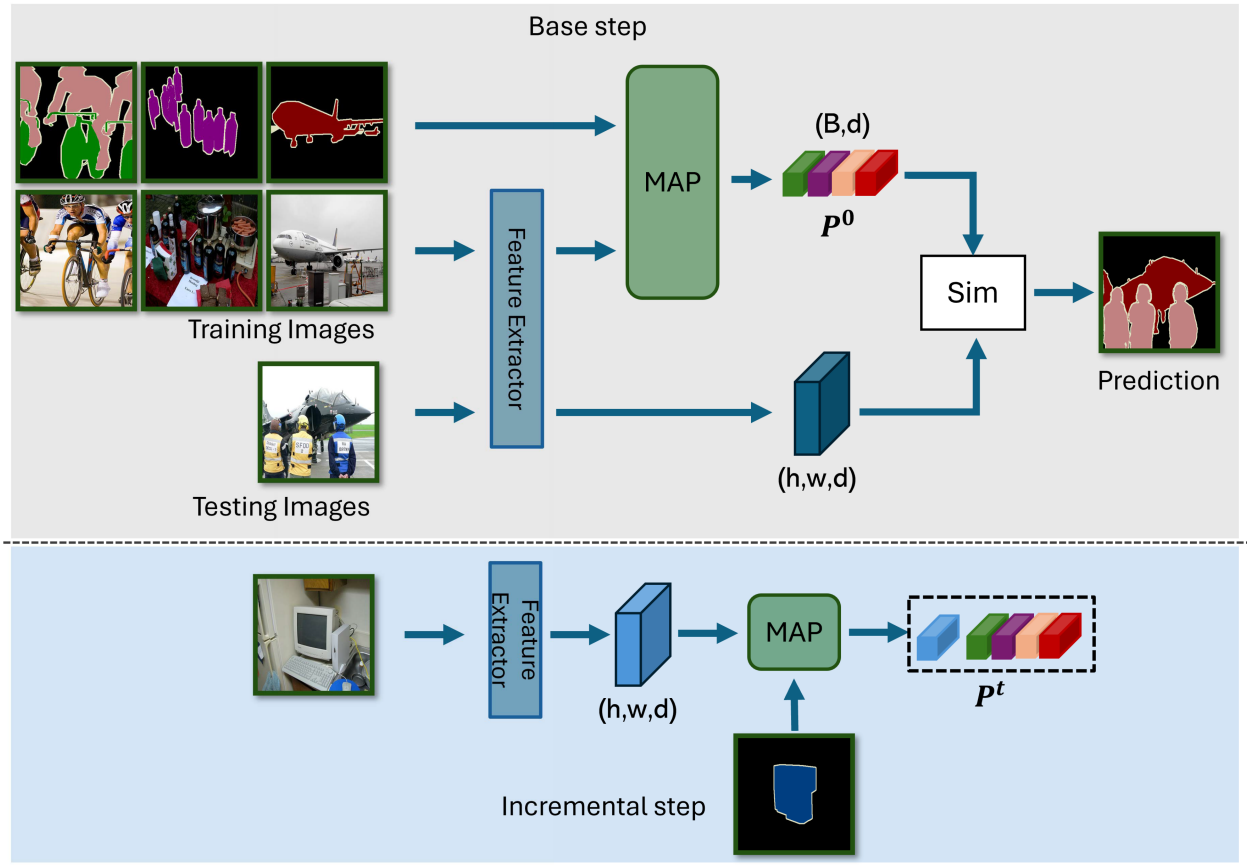


Figure 5.2: Prototype-based model for iFSS: In the base step, the model undergoes training with base class data to acquire the initial prototype classifier $\mathcal{P}^0$, encompassing prototypes for all base classes. The "Sim" function serves as a similarity metric, computing the distance between positional features and prototypes to enable pixel-wise classification. During the incremental step, a prototype of the novel class is derived via Masked Average Pooling (MAP) and integrated into $\mathcal{P}^0$ to establish a new classifier capable of segmenting both novel and base classes

framework for iFSS comprises a feature extractor and a prototype classifier. In the base step, the feature extractor transforms the input image $\mathcal{I} \in \mathbb{R}^{h \times w \times 3}$ into a feature embedding $\mathcal{F} \in \mathbb{R}^{w \times h \times d}$ in a latent space. Subsequently, a prototype classifier $\mathcal{P}^0 \in \mathbb{R}^{B \times d}$ is trained on base classes to perform pixel-wise predictions for B classes on $\mathcal{F}$. For the incremental step, our objective is to progressively expand the base prototype classifier $\mathcal{P}^0$ with prototypes of novel classes, facilitating the continuous segmentation of newly encountered classes without forgetting prior knowledge. Formally, in an N-class K-shot incremental session (N novel classes and each novel class has K training samples), all training samples $\mathcal{I}^t_{c,n}$ are first processed by a feature extractor f and mask average pooling. Subsequently, these samples are averaged over K shots to create

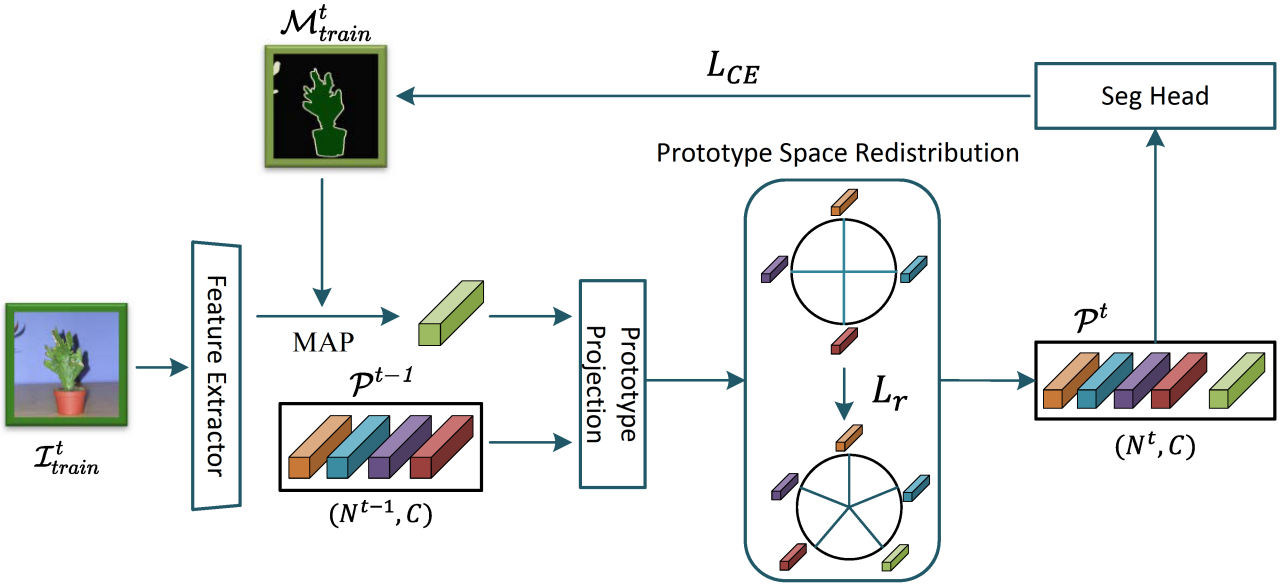


Figure 5.3: The proposed prototype-based approach utilizes masked average pooling (MAP) to derive the novel class prototype. Subsequently, all prototypes are projected into a latent prototype space for redistribution. The resulting prototypes form a new classifier $\mathcal{P}^t$ capable of identifying both base and novel classes. This process is considered as a sequential task of the meta-learning optimization. In the online incremental sessions, the feature extractor remains frozen, and only the prototype projector and segmentation head are updated.

N prototypes, denoted as $p^t_c (c \in \{1, 2, \ldots, N\})$.

$$p^t_c = \frac{1}{K} \sum_{n=1}^{K} \frac{\sum_{h,w} \left[\mathcal{M}^t_{c,n} \circ f\left(\mathcal{I}^t_{c,n}\right)\right]_{h,w}}{\sum_{h,w} \left[\mathcal{M}^t_{c,n}\right]_{h,w}}, \tag{5.1}$$

where $\mathcal{I}^t_{c,n}$ denotes the n-th training image of class $\mathbf{c}$. $\mathcal{M}^t_{c,n} \in \mathbb{R}^{h,w,1}$ is the class mask for class $\mathbf{c}$ on feature $f\left(\mathcal{I}^t_{c,n}\right) \in \mathbb{R}^{h,w,d}$. After obtaining N prototypes, the prediction of pixel i of $\mathcal{F}$ is assigned according to the normalized cosine similarity score $S_{i,c}(\mathcal{F})$ between features and the class prototype p^t_c as:

$$S_{i,c}(\mathcal{F}) = \frac{\exp\left(Sim(\mathcal{F}_i, \mathbf{p^t_c})/\tau\right)}{\sum_{j=1}^{N^t} \exp\left(Sim(\mathcal{F}_i, \mathbf{p^t_j})/\tau\right)}, \tag{5.2}$$

where $\mathcal{F}_i \in \mathbb{R}^d$ are the positional features extracted from input image $\mathcal{I}$, $N^t = N^{t-1} + N$ represents the cumulative category of prototype vectors up to session t, $N^{t-1} = B$ when it comes to the first incremental session. The τ is a temperature parameter that controls the concentration level of the distribution [172]. $Sim(,) = \frac{\mathcal{F}_i^\top \mathbf{p^t}}{\|\mathcal{F}_i\| \|\mathbf{p^t}\|}$ is the cosine similarity metric that measures the pixel classification score.

5.2.3 Prototype Space Redistribution Learning

The previously mentioned basic framework inevitably leads to the problem of catastrophic forgetting. When the model trains in incremental session to derive prototypes for novel classes, the parameters of the feature extractor are updated. This means that the knowledge of the base classes will be gradually overlid by the novel knowledge as the incremental steps progress.

Cermelli et al. [164] proposed a prototype-based distillation loss to recover base class knowledge from the previous status of the model. However, the feature extractor of those methods is frozen which means the feature space distributed for the base class is reused to accommodate extra class. When adapting to new classes, there may be visual similarities or overlapping features between old and new classes. This can lead to interference, where the model incorrectly associates features from previously learned classes with the new classes, causing both catastrophic forgetting and overfitting.

We argue that the misalignment dilemma between features and classifiers is the root cause of the catastrophic forgetting problem for old classes. If a backbone network is updated in incremental sessions, the features of old classes in test images may easily deviate from their classifier prototypes. Conversely, if a backbone network is fixed and a set of new prototypes for novel classes are allocated into the constant feature space, where the new prototypes are lied close to old-class prototypes, it may also induce misalignments with the fixed features. Hence, the objectives of adjusting the classifier prototypes and the network are twofold: (i) maintaining a sufficient distance between the old-class and the novel-class prototypes; (ii) preventing the adjusted old-class prototypes from deviating significantly from their original positions.

Considering that the prototype classifier encompasses both base and newly encountered classes, and base examples are inaccessible during incremental learning, modulating the feature extractor might lead to new classes being mapped into a different feature space from that of base classes. Therefore, to ensure consistent feature mapping, the backbone is prefered to remain consistently fixed. For concern that the newly added prototypes may be close to the base-class prototypes because the prototype is derived from a fixed feature space tailored for base classes. We introduce the prototype projector $\mathbf{g}$ to map the current prototypes into a latent prototype space. This space allows for the adaptive distribution of base and novel prototypes, achieving two objectives: i) ensuring clear inter-prototype discrimination among base and novel prototypes for fast adaptation to new classes, and ii) minimizing the displacement of base prototypes

away from their original positions to prevent catastrophic forgetting and maintain alignment between features and prototypes. Accordingly, we propose a novel prototype redistribution loss that places the new class prototype P_i^t at a position far from base prototypes P_j^{t-1} and relocates base classes to a near-optimal position as:

$$\mathcal{L}_r = \frac{\sum_{i=1}^{N^{t-1}} \sum_{j=1}^{N} Sim\left(P_i^{t-1}, P_j^t\right)}{\sum_{i=1}^{N^{t-1}} Sim(P_i^{t-1}, \hat{P}_i^{t-1})}, \tag{5.3}$$

where N^{t-1}, N are the class prototype number of previous sessions $[0, 1, ..., t-1]$ and current session t. $\hat{P}_i^{t-1}$ represents the redistributed prototype vector derived from the base prototype P_i^{t-1}. We utilize cosine distance as the metric for the similarity matrix. The loss function $\mathcal{L}_r$ is designed to minimize the similarity between new class prototypes and base prototypes while simultaneously maximizing the similarity between the original base prototypes and their respective redistributions.

Figure 5.3 illustrates the process of adapting novel classes. One or a few novel training image pairs $\mathcal{I}_n^t, \mathcal{M}_n^t$ are provided to the model. After applying masked average pooling to the training image, the novel prototype is obtained. Along with the current old class prototypes in P^{t-1}, these prototypes are projected into a latent prototype space. In this space, supervised by the loss function 5.3, the prototypes are redistributed to ensure they are far apart from each other while remaining relatively close to their original positions. The redistributed prototypes then form a new classifier capable of segmenting both novel and base objects. It is noteworthy that the feature extractor continues to update during the base training stage and remains fixed during the incremental training stage. The prototype projection and segmentation head remain consistently trainable to facilitate rapid adaptation. The objective $\mathcal{L}_{task}$ is a conbined loss function designed to achieve optimal performance through the redistribution of class prototypes as:

$$\mathcal{L}_{task} = \mathcal{L}_{CE}(\mathcal{I}_{test}, \mathcal{M}_{test}) + \lambda \mathcal{L}_r. \tag{5.4}$$

5.2.4 Learning to Incrementally Learn

The core idea underlying our approach is meta-learning inspired by MAML [76] for few-shot tasks. During the meta-training phase, the model is trained with a set of novel class adaptation tasks that are formulated as few-shot learning problems, aiming to simulate the scenario encountered during meta-testing. In iFSS, the online incremental stage closely resembles the

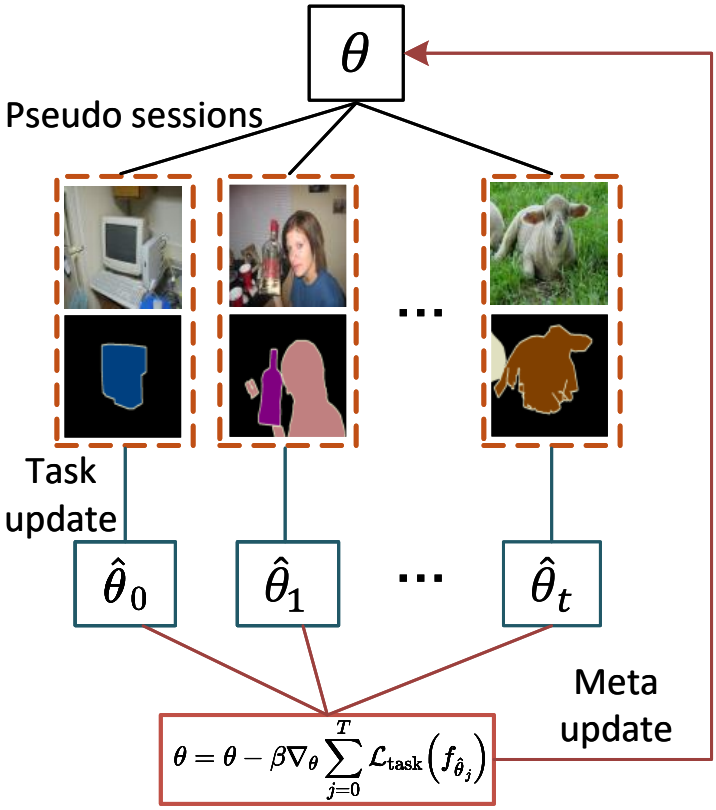


Figure 5.4: The meta-learning optimization strategy samples pseudo-sequential learning tasks on the base set to perform task training. The meta update process encourages the model to learn in a manner that preserves performance on old classes while effectively adapting to novel classes with minimum likelihood of overfitting.

"meta-testing" stage. This stage entails adapting the model to a sequence of incremental sessions, where each session introduces several novel classes with few-shot examples. Inspired by this, the model is meta-trained on base classes with the goal of mimicking the incremental learning scenario anticipated during the subsequent online incremental learning (i.e., evaluation). This ensures that the model is learned in a manner enabling effective adaptation to new classes with less forgetting.

Sequential task sampling. We replicate the evaluation process by utilizing the base classes. More precisely, we segregate the training images of base classes into distinct training and testing sets with no overlap. In each epoch, we initiate the training process by sampling a sequence of T tasks, $\mathcal{D}^s_{train/test} = \left\{\left(\mathcal{I}^j_{train/test}, \mathcal{M}^j_{train/test}\right)\right\}_{j=0}^{T}$, where T is greater than the actual incremental session number, and each session include training and testing image-mask pairs. We define D^0 as the pseudo base set, comprising more classes and training examples than

subsequent tasks (e.g., $j > 0$) in the few-shot setting. To reduce the risk of the model overfitting to a specific sequence, we randomly repeat the sampling process with different classes until the model converges.

Algorithm 1 Meta training process

Require: : θ^g, θ^{seg}: pre-trained weights

Require: : $\mathcal{D}^0$: training set of base classes

Initialize the models with pre-trained weights

while not converged **do**

  $\mathcal{D}^s_{train/test} = \left\{ \left(\mathcal{I}^j_{train/test}, \mathcal{M}^j_{train/test} \right) \right\}_{j=0}^{T}$

  $\mathcal{D}_{meta} = \emptyset$

  for $j = 0, 1, 2, ..., T$ **do**

    $\mathcal{P} = Concat(\mathcal{P}_{old}, \mathcal{P}_{new})$

    $\hat{\theta}^{g,seg} = \theta^{g,seg} - \alpha \nabla_{\theta^{g,seg}} \mathcal{L}_{task} \left(\mathcal{I}^j_{train}, \mathcal{M}^j_{train}; \theta^{g,seg} \right)$

    $\mathcal{D}_{meta} = \mathcal{D}_{meta} \cup \mathcal{D}^j_{test}$

    $\theta^{g,seg} = \theta^{g,seg} - \beta \nabla_{\theta^{g,seg}} \sum_{(\mathcal{I},\mathcal{M}) \in \mathcal{D}_{meta}} \mathcal{L}_{task} \left(\mathcal{I}, \mathcal{M}; \hat{\theta}^{g,seg} \right)$

  end for

end while

Meta-training. During the meta-training phase, for every sampled sequence $\mathcal{D}^s_{train/test}$, we introduce a prototype redistribution-oriented optimization approach grounded in meta-learning. We denote $\theta = \{\theta^f, \theta^g, \theta^{seg}\}$ as the parameter for the whole network, where $\theta^f, \theta^g, \theta^{seg}$ denote the parameters for backbone, prototype projection layer and segmentation head, respectively. We first conduct supervised training of θ on the pseudo base classes using segmentation loss ($\mathcal{L}_{task}$). The meta-training procedure is illustrated in Algorithm 1 and Figure 5.4. At the beginning of training on each sequence, we define an empty cumulative meta test set $\mathcal{D}_{meta}$ to store the test images from previous tasks. At the j^{th} task, we first generate the new class prototypes $\mathcal{P}_{new}$ and then concatenate it into the current prototype classifier $\mathcal{P}_{old}$. Subsequently, we start to perform fast adaptation to new classes and update θ^g and θ^{seg} via a few L gradient steps:

$$\hat{\theta}^{g,seg}_j = \theta^{g,seg} - \alpha \nabla_{\theta^{g,seg}} \mathcal{L}_{task} \left(\mathcal{I}^j_{train}, \mathcal{M}^j_{train}; \theta^{g,seg} \right), \tag{5.5}$$

where $\mathcal{I}^j_{train}, \mathcal{M}^j_{train}$ are the images and labels for training j^{th} pseudo task. The loss $\mathcal{L}_{task}(,:)$ is computed on the output of the current model and the target label $\mathcal{M}^j_{train}$.

The adaptation process mimics the model's learning pattern for new classes during incremental sessions. Ideally, we aim for the adapted parameters to perform well in both the classes from the previous and current tasks. The meta-testing set accumulated from previous tasks is used for evaluating how well the updated model resists catastrophic forgetting on old classes and adaptation on new classes. We append $\mathcal{D}_{test}^{j}$ to $\mathcal{D}_{meta}$, and accordingly, the meta-objective is defined as:

$$\theta^{g,seg} = \theta^{g,seg} - \beta \nabla_{\theta^{g,seg}} \sum_{(\mathcal{I},\mathcal{M}) \in \mathcal{D}_{meta}} \mathcal{L}_{meta} \left(\mathcal{I}, \mathcal{M}; \hat{\theta}^{g,seg} \right) \tag{5.6}$$

In the online incremental learning stage, we execute Lines 5-7 of Algorithm 1 to acquire knowledge about novel classes during evaluation. The steps outlined in Algorithm. 1 align with the evaluation protocol: after being trained on the current session, the model undergoes evaluation on all encountered classes so far. This meta-objective encourages our model to quickly adapt to novel classes without sacrificing remembering old ones.

5.3 Experiments

5.3.1 Dataset

We evaluate the proposed method on two widely used semantic segmentation datasets: PASCAL VOC 2012 [44] and COCO [45]. Following established practices [164], we evenly partition the classes in PASCAL VOC and COCO into four folds as in the previous chapters, with each fold containing 5 and 20 categories, respectively. According to the few-shot incremental task setting, the base set should encompass sufficient labeled samples, while each incremental session consists of only a few samples from previously unseen classes. For both PASCAL and COCO datasets, three folds are designated to form the base set, while the categories from the remaining fold are used for testing purposes.

5.3.2 Implementation Details and Evaluation Metrics

In all experiments, we employ ResNet-101 [160] pre-trained on ImageNet as the feature extractor. Our configuration involves ASPP [20] with a 1x1 convolutional layer as the segmentation head. All the models are trained using SGD and a batch size of 16 on NVIDIA GPU RTX6000.

The learning rate is set to 0.02 and 0.01 for the pre-training and fine-tuning stages respectively. The number of iterations for the pre-training stage is 110000 with two weight decay steps with the rate of 10 at 80000 and 100000 iterations. The number of iterations for fine-tuning stage depends on the number of examples ranging from 500 iterations (with K = 1) to 6000 iterations (with K = 30).

According to the GFSS protocol outlined in [162], we evaluate the performance of a method utilizing three mean intersection-over-union (mIoU) metrics: mIoU on base classes (mIoU-B), mIoU on new classes (mIoU-N), and the harmonic mean of the two (HM). Consistent with [164], all reported results are presented upon the completion of training in the final incremental session. Particularly, the single step means all the *New* classes are given in one session, while multi-step has multiple sessions: 5 sessions of 1 class on VOC and 4 sessions of 5 classes on COCO.

As there are few iFSS works in the literatures, we adapt some incremental and prototype-based models as baselines of the comparison experiments. A very basic baseline is finetune, which directly fine-tune the base model with new classes on each session.

For Weight Imprinting (WI), we extended the methodology from [173] originally designed for image classification. Specifically, we substituted the image-level feature extractor used in with masked average pooling (MAP). This approach eliminates the need for additional hyperparameters, and we initialize the prototypes for new classes while retaining the prototypes of old ones unaltered.

Similarly, in the case of Dynamic Weight Imprinting (DWI) [174], we employed the classifier utilizing the identical attention mechanism and weight generator as described in [174]. However, we substituted the class-specific image-level features with those extracted through MAP. DWI incorporates a secondary meta-learning training stage on the base classes to enhance the performance of the weight generator.

We implemented the knowledge distillation-based method "Modeling the Background (MiB)" following the approach outlined in [175]. MiB utilizes revised cross-entropy and distillation losses, along with initialization of classifier weights for new classes.

For the Semantic Projection Network (SPN) [176], we utilized the implementation provided by the authors, which incorporates a combination of word2vec [177] and fastText [178] as class

embeddings, directly employing them as classifier weights. This method does not entail specific hyperparameters, and we tailored it for iFSS by excluding the retention of old datasets in the learning steps.

5.3.3 Main Results

Quantitive analysis.The results of our method on the PASCAL VOC 2012 and COCO datasets are reported in Table 5.1 and Table 5.2, respectively. Our approach demonstrates superior performance in novel class adaptation across all settings for both PASCAL and COCO datasets. Additionally, it achieves state-of-the-art performance in terms of Harmonic Mean (HM) scores across all settings, indicating that our approach effectively balances the retention of information about old classes while facilitating adaptation to new ones. Particularly noteworthy is our method's performance on the PASCAL dataset, where it achieves significantly higher novel class segmentation mIoU scores compared to all other methods, reaching 35.8% and 29.1% in single-step and multi-step settings, respectively. This surpasses the state-of-the-art method (PIFS) by 2.4% and 1.8%, respectively. Our meta-learning-based approach exhibits superior fast adaptation capability to novel classes without compromising base class segmentation accuracy, achieving competitive base class segmentation performance on both PASCAL and COCO datasets.

On the COCO dataset, our approach showcases significantly greater improvements in HM scores compared to the state-of-the-art method PIFS [164]. For instance, in the task of 5-shot segmentation, our method's HM scores surpass those of PIFS by 5.3% and 4.1%, whereas the margins are only 2.3% and 1.8% on the PASCAL dataset. This highlights the effectiveness of our approach in tackling the more intricate challenges associated with a larger number of classes, which is particularly beneficial in real-world applications.

Qualitative analysis. In Figure 5.6 and Figure 5.7, we showcase visualized segmentation results obtained from training under the multi-step incremental setup, using one training example for each novel class. Figure 5.6 shows prediction results on PASCAL dataset, from left to right: training image, testing image, weight-printing (WI) segmentation results (the baseline method for reference), our proposed model's segmentation predictions, and the ground truth segmentation mask. Segmentation masks are overlaid on the original images for clarity. In comparison to vanilla weight-printing (WI), which simply appends new class prototypes to the

Table 5.1: The experimental results, measured in terms of mIoU, are presented for the PASCAL dataset. "FT" signifies direct fine-tuning of the model solely on novel classes following traditional supervised learning methods. "HM" denotes the harmonic mean of the mIoU scores calculated separately for base and novel classes.

Method	Single step						Multi-step					
	1-shot			5-shot			1-shot			5-shot		
	Base	Novel	HM	Base	Novel	HM	Base	Novel	HM	Base	Novel	HM
FT	58.3	9.7	16.7	55.8	29.6	38.7	47.2	3.9	7.2	58.7	7.7	13.6
WI [173]	62.7	15.5	24.9	64.9	21.7	32.5	66.6	16.1	25.9	66.6	21.9	33.0
DWI [174]	**64.3**	15.4	24.8	64.9	23.5	34.5	**67.2**	16.3	26.2	**67.6**	25.4	36.9
MiB [179]	61.0	5.2	9.7	**65.0**	28.1	39.3	43.9	2.6	4.9	60.9	5.8	10.5
SPN [176]	59.8	16.3	25.6	58.4	33.4	42.5	49.8	8.1	13.9	61.6	16.3	25.8
PIFS [164]	60.9	18.6	28.5	60.5	33.4	43.0	64.1	16.9	26.7	64.5	27.5	38.6
Ours	63.4	**19.7**	**30.1**	61.6	**35.8**	**45.3**	65.5	**20.4**	**31.1**	65.9	**29.1**	**40.4**

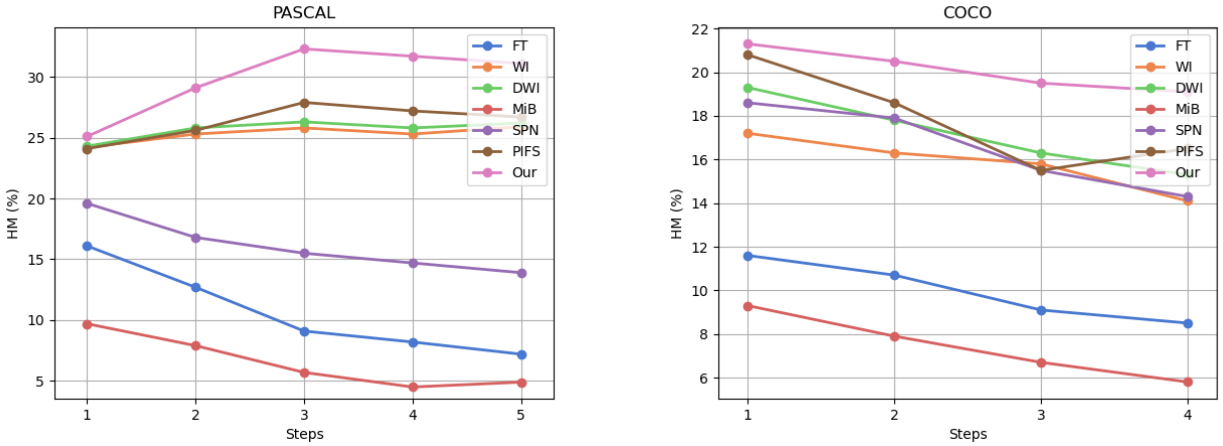
prototype classifier, our approach notably distinguishes novel classes like "bus" from the base class "person" and "sheep" from the background. Additionally, as observed in the third row, WI exhibits overfitting to the "sofa" and completely forgets the knowledge of the "chair". Our method, employing task-consistent meta-learning with prototype distribution loss, preserves the ability to segment learned classes while accurately adapting to new classes.

The iFSS task becomes more challenging when dealing with images from COCO containing a larger number of classes compared to PASCAL. Our qualitative results in Figure 5.7 on COCO demonstrate that our method excels in distinguishing between objects from base and novel classes. For instance, in the results pertaining to the "sandwich" class, our method correctly identifies the majority part of "orange", whereas WI misclassifies it as another class, labeling it with a blue color. This capability stems from our prototype redistribution loss, which pushes similar prototypes away from each other, enabling clear differentiation between similar objects. Similar observations are made in discriminating between "giraffe" and "zebra."

Step by step analysis. To investigatethe process of novel class adaptation and base class forgetting, we systematically assess the model's performance at each incremental step under the multi-step setting. The results for each incremental step on PASCAL and COCO are presented in Figure 5.5. For each incremental step, harmonic mean (HM) scores between the mIoU for

Table 5.2: The experimental results (mIoU) on COCO dataset.

Method	Single step						Multi-step					
	1-shot			5-shot			1-shot			5-shot		
	Base	Novel	HM	Base	Novel	HM	Base	Novel	HM	Base	Novel	HM
FT	41.2	4.1	7.5	41.6	12.3	19.0	38.5	4.8	8.5	39.5	11.5	17.8
WI [173]	43.8	6.9	11.9	43.6	8.7	14.5	**46.3**	8.3	14.1	46.3	10.3	16.9
DWI [174]	**44.5**	7.5	12.8	**44.9**	12.1	19.1	46.2	9.2	15.3	**46.6**	14.5	22.1
MiB [179]	43.8	3.5	6.5	44.7	11.9	18.8	40.4	3.1	5.8	43.8	11.5	18.2
SPN [176]	43.5	6.7	11.7	43.7	15.6	22.9	40.3	8.7	14.3	41.4	18.2	25.3
PIFS [164]	40.8	8.2	13.7	42.8	15.7	23.0	40.4	10.4	16.5	41.1	18.3	25.3
Ours	43.8	**10.4**	**16.7**	44.4	**20.8**	**28.3**	43.1	**12.3**	**19.1**	43.5	**22.2**	**29.4**



(a) PASCAL Multi-step

(b) COCO Multi-step

Figure 5.5: Step by step "Multi-step" HM results on PASCAL and COCO dataset.

base and new classes are shown on the line chart.

From the results, it is evident that fine-tuning (blue) and incremental learning methods such as MiB (dark orange) exhibit the poorest performance across all settings. This can be attributed to their failure in leveraging prototype learning, which hampers their ability to effectively initialize and represent the new classes. Prototypes act as a compact and efficient way to store information about previously learned classes. Unlike storing all the training data, prototypes capture the essential features of each class in a single vector. By comparing new data to these prototypes, the model can maintain knowledge of past classes even as it learns new ones. This reduces memory requirements and computational overhead compared to keeping all the training

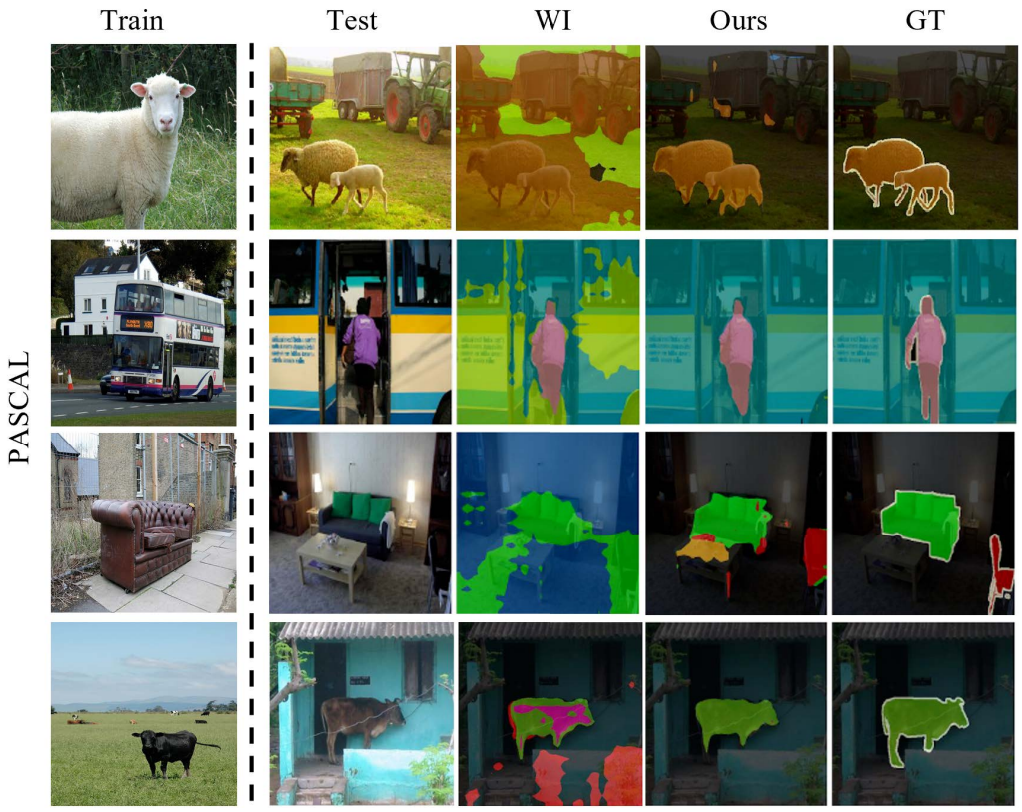


Figure 5.6: Visualization of multi-step results under one shot setting on the PASCAL

data, making it suitable for scenarios where resources are limited.

On the contrary, methods that incorporate prototype learning, such as DWI (green), WI (orange), and PIFS (brown), demonstrate a more favorable balance between learning and forgetting. Particularly noteworthy is PIES, which utilizes distillation loss on a prototype model and achieves commendable performance on PASCAL, closely trailing our approach (pink) in the 1-shot setting. Our method stands out among the counterparts, exhibiting a significant performance advantage, primarily attributable to its remarkable ability to identify new classes while maintaining stability on seen classes.

In the COCO results, all methods exhibit a downward trend as incremental steps progress. Particularly noteworthy is the performance of PIFS, which achieves a high score in the initial step but then experiences a dramatic drop to around 15% by the third step before experiencing a slight increase in the final step. In contrast, our method consistently outperforms the others across all steps. This demonstrates that our prototype generation module enhances class-wise feature representation, while our meta-learning optimization strategy enables the model to learn with reduced forgetting.

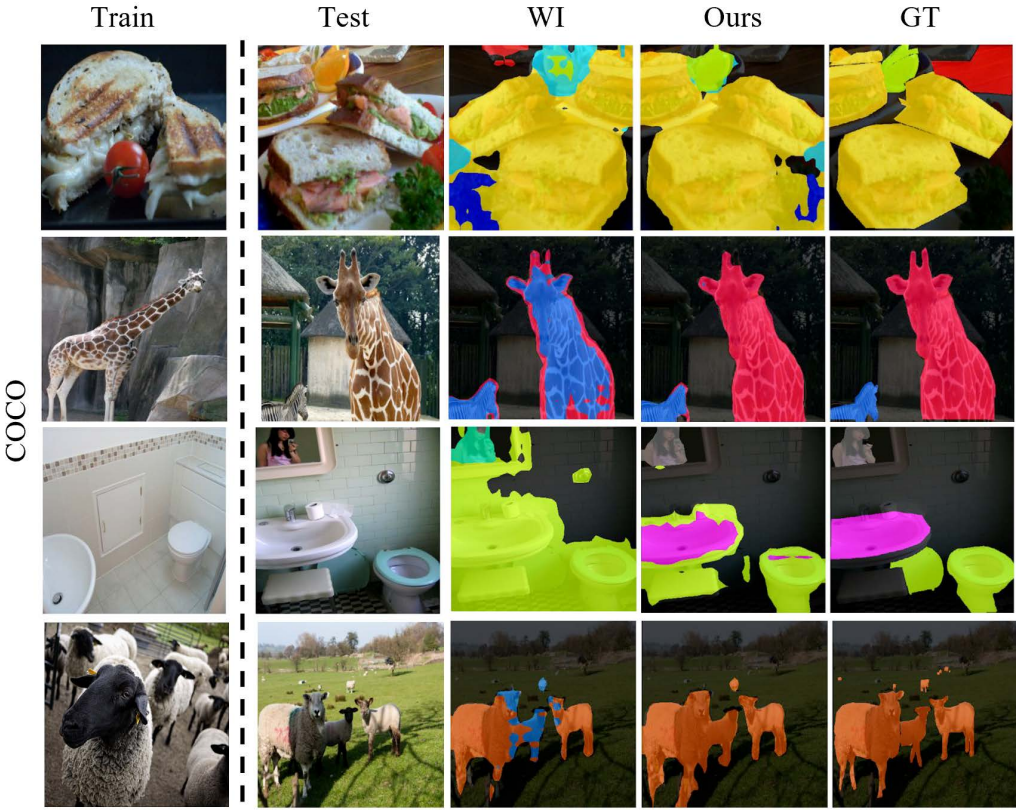


Figure 5.7: Visualization of multi-step results under one shot setting on the COCO

5.3.4 Ablation Study

In this section, all the ablation studies are conducted on COCO dataset adopting 1-shot and multi-step incremental setting.

Componet ablations We investigate the effectiveness of three key components: i) the proposed prototype redistribution loss $\mathcal{L}_r$, which not only maximizes the distance between inter-class prototypes but also prevents base prototypes from shifting away; ii) the meta-learning training strategy, which encourages the model to rapidly learn novel classes while remaining sensitive to old ones; and iii) the inter-prototype discrimination loss $\mathcal{L}inter = \sum_{i=1}^{N^b} \sum_{j=1}^{N^t} Sim\left(P_i^{t-1}, P_j^t\right)$, which focuses on minimizing the similarity between novel and base classes. We use vanilla prototype weight imprinting as the basic model for reference.

As illustrated in the second row of Table 5.3, introducing the meta-learning strategy, which trains the model in a manner aligned with the expected evaluation in the incremental sessions, significantly improves the novel class adaptation performance by 3.4% and mitigates catastrophic forgetting. The application of $\mathcal{L}_{inter}$ upon meta-learning results in a 0.9% increase in novel class accuracy but induces a 1.3% performance reduction in the base class. It suggests

Table 5.3: Ablation study of the meta-learning scheme and prototype redistribution loss on COCO in terms of mIoU (%), under the multi-step one-shot setting. $\mathcal{L}_{inter} = \sum_{i=1}^{N^b} \sum_{j=1}^{N^t} Sim\left(P_i^{t-1}, P_j^t\right)$ merely aims to minimize similarity between novel and base classes. "Base" denote the vanilla prototype weight imprinting.

Base	Meta-learning	$\mathcal{L}_{\text{inter}}$	$\mathcal{L}_{\text{r}}$	Base	Novel	HM
✓				**44.1**	7.2	12.4
✓	✓			42.5	10.6	17.0
✓	✓	✓		41.2	11.5	18.0
✓	✓		✓	43.1	**12.3**	**19.1**

Table 5.4: Ablations on backbones and prototype redistribution. "fix" denotes that the backbone remains fixed during incremental steps, while "update" means that the backbone continues to update. "PR" indicates the addition of the prototype projection layer and the adoption of the prototype redistribution loss $\mathcal{L}_r$.

Methods	Novel	Base	HM
Baseline (fix)	7.2	**44.1**	12.4
Baseline (update)	7.8	36.0	12.8
Baseline (fix) + PR	**10.6**	40.4	**16.8**
Baseline (Update) + PR	10.2	36.5	15.9

that merely focusing on minimizing the similarity between the new class and the old class prototypes while neglecting the drift of the base class can lead to prototype inconsistency before and after adaptation, resulting in knowledge forgetting.

Backbone and prototype redistribution. To investigate the performance difference between frozen and updated backbones, we conduct comparison experiments using two baseline models. In these experiments, the pre-trained backbone is either kept fixed or updated during the incremental steps. The model with the fixed backbone is denoted as Baseline (fix), while the model with the updated backbone is referred to as Baseline (update). As shown in Table 5.4, Baseline (update) outperforms Baseline (fix) in terms of HM score, primarily due to its superior performance on novel classes. However, there is a significant drop in mIoU for base classes, indicating that updating the backbone without any constrain may lead to overfitting

on new classes and result in catastrophic forgetting.

Then, we augment the model by appending a prototype projection layer after the backbone and applying prototype redistribution supervision to obtain the classifier. From the results of the last two rows of Table 5.4, the fixed version outperforms the updated counterpart by a significant margin in both novel and base class segmentation. This superiority is attributed to the fixed backbone's ability to retain information about the base classes, while "PR" ensures that the prototypes in the subspace remain well-separated. These factors mitigate catastrophic forgetting and facilitate rapid adaptation.

5.4 Chapter Summary

This chapter addresses a practical scenario of semantic segmentation that incrementally learns novel classes with a few examples. A meta-learning-based approach is proposed, which directly optimizing the network to acquire the ability to incrementally learn within the few-shot incremental setting. To alleviate catastrophic forgetting and overfitting problems, we introduce a prototype space re-distribution mechanism to dynamically update class prototypes during each incremental session. Extensive experiments on PASCAL and COCO benchmarks demonstrate that the proposed method facilitates a model learning paradigm for quick classes learning without forgetting.

Chapter 6

Conclusions and Future Work

6.1 Summary of Outcomes

This dissertation rigorously investigates a range of methodologies and strategies aimed at enhancing the generalizability and robustness of image semantic segmentation models under conditions of limited data supervision. It addresses this intricate yet promising area through an exploration of two sequential tasks: few-shot semantic segmentation (FSS) and incremental few-shot semantic segmentation (iFSS). FSS involves segmenting target objects in query images with the aid of a small set of pixel-wise annotated support images, while iFSS extends this challenge by necessitating the retention of knowledge across all encountered classes.

Acknowledging the limitation in most FSS approaches, which independently learn class-wise descriptors from support images while neglecting the intricate contextual interplay and mutual dependencies between support and query features, Chapter 3 introduces a novel joint learning methodology named Masked Cross-Image Encoding (MCE). This method aims to elucidate common visual characteristics that define object particulars and to foster bidirectional inter-image dependencies, thereby augmenting feature interaction. MCE transcends a mere visual representation enhancement module by incorporating cross-image mutual dependencies and implicit guidance.

Chapter 4 offers a fresh perspective on the feature-matching mechanism within the FSS framework. Predominant FSS techniques employ a support-query matching approach that activates target regions in the query image based on their resemblance to a singular support class proto-

type. Nonetheless, this prototype vector is susceptible to overfitting support images, potentially leading to under-matching in latent query object regions and erroneous mismatches with base class features in the query image. To confront these challenges, this chapter redefines the conventional single foreground prototype matching to a multi-prototype matching approach. Within this paradigm, query features demonstrating high confidence with non-target prototypes are classified as background. Specifically, target query features are aligned more closely to the novel class prototype using the Masked Cross-Image Encoding (MCE) module introduced in Chapter 3, and a Semantic Multi-prototype Matching (SMM) module is employed to collaboratively filter unintended base class regions on multi-scale features. Moreover, it introduces an adaptive class activation map, termed target-aware class activation map (TCAM), to conserve semantically coherent regions that may be inadvertently suppressed under pixel-wise matching guidance.

Addressing the application of FSS in real-world scenarios, Chapter 5 delves into incremental FSS, where the model is continually exposed to new streams of image data comprising instances of previously unseen classes. Common approaches often involve pre-training models on base classes in a fully supervised manner, followed by the application of few-shot prototype learning during incremental sessions. In the absence of base class data, such paradigms are vulnerable to overfitting novel classes and forgetting prior ones due to misalignments between offline base learning objectives and online incremental learning assessment protocols. This study introduces a meta-learning-based approach for iFSS, emulating the incremental evaluation protocol during base training sessions. Each task in the simulated sequence is trained using a meta-objective to facilitate swift adaptation without forgetting. To enhance discrimination among class prototypes, the dissertation proposes prototype space re-distribution learning, which dynamically updates class prototypes in each incremental session, thereby establishing optimal inter-prototype distances within the prototype space.

6.2 Future Work

Further work in the field of few-shot segmentation can explore various dimensions to address existing challenges and open new avenues for research. Some potential directions include:

- Integration with Unsupervised and Semi-supervised Learning: Exploring how unsupervised or semi-supervised learning can complement few-shot learning to make the most of

unlabeled or partially labeled datasets, reducing the dependency on annotated data.

- Human-in-the-loop Learning: Integrating human feedback into the learning loop to refine model predictions and annotations, ensuring higher accuracy and reliability in scenarios where expert knowledge is crucial.
- Novel class discovery segmentation(GCDSS). It requires the model to learn from both labeled and unlabeled data, and to discover and segment the novel classes without any supervision.
- Generalized class discovery segmentation. Different from novel class discovery segmentation (NCDSS), which assumes that each unlabeled image has at least one novel class and focuses only on foreground objects. GCDSS is more realistic and challenging, as it does not require such prior knowledge and covers the entire image.

Bibliography

[1] Q. Hu, Y. Chen, J. Xiao, *et al.*, "Label-free liver tumor segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7422–7432.

[2] Y. Zhang, X. Li, H. Chen, A. L. Yuille, Y. Liu, and Z. Zhou, "Continual learning for abdominal multi-organ and tumor segmentation," in *International conference on medical image computing and computer-assisted intervention*, Springer, 2023, pp. 35–45.

[3] D. Zhang, H. Wang, J. Deng, T. Wang, C. Shen, and J. Feng, "Cams-net: An attention-guided feature selection network for rib segmentation in chest x-rays," *Computers in Biology and Medicine*, vol. 156, p. 106 702, 2023.

[4] C. Chen, C. Wang, B. Liu, C. He, L. Cong, and S. Wan, "Edge intelligence empowered vehicle detection and image segmentation for autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[5] R. Marcuzzi, L. Nunes, L. Wiesmann, J. Behley, and C. Stachniss, "Mask-based panoptic lidar segmentation for autonomous driving," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 1141–1148, 2023.

[6] J. Yu, J. Zhang, A. Shu, *et al.*, "Study of convolutional neural network-based semantic segmentation methods on edge intelligence devices for field agricultural robot navigation line extraction," *Computers and Electronics in Agriculture*, vol. 209, p. 107 811, 2023.

[7] I. Asante, L. B. Theng, M. T. K. Tsun, H. S. Jo, and C. McCarthy, "Segmentation-based angular position estimation algorithm for dynamic path planning by a person-following robot," *IEEE Access*, 2023.

[8] L. Fang, Y. Jiang, Y. Yan, J. Yue, and Y. Deng, "Hyperspectral image instance segmentation using spectral–spatial feature pyramid network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–13, 2023.

[9] F. Kulwa, C. Li, M. Grzegorzek, M. M. Rahaman, K. Shirahama, and S. Kosov, "Segmentation of weakly visible environmental microorganism images using pair-wise deep learning features," *Biomedical Signal Processing and Control*, vol. 79, p. 104 168, 2023.

[10] P. Bosilj, E. Aptoula, T. Duckett, and G. Cielniak, "Transfer learning between crop types for semantic segmentation of crops versus weeds in precision agriculture," *Journal of Field Robotics*, vol. 37, no. 1, pp. 7–19, 2020.

[11] T. Anand, S. Sinha, M. Mandal, V. Chamola, and F. R. Yu, "Agrisegnet: Deep aerial semantic segmentation framework for iot-assisted precision agriculture," *IEEE Sensors Journal*, vol. 21, no. 16, pp. 17 581–17 590, 2021.

[12] N. Sharma, M. Mishra, and M. Shrivastava, "Colour image segmentation techniques and issues: An approach," *International Journal of Scientific & Technology Research*, vol. 1, no. 4, pp. 9–12, 2012.

[13] J. Ning, L. Zhang, D. Zhang, and C. Wu, "Interactive image segmentation by maximal similarity based region merging," *Pattern Recognition*, vol. 43, no. 2, pp. 445–456, 2010.

[14] O. Mustafa, "Image segmentation using color and texture features," *IEEE*,

[15] D. Comaniciu and P. Meer, "Robust analysis of feature spaces: Color image segmentation," in *Proceedings of IEEE computer society conference on computer vision and pattern recognition*, IEEE, 1997, pp. 750–755.

[16] Z. Kato. 2012. DOI: 10.1561/2000000035.

[17] 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*, 2015, pp. 3431–3440.

[18] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[19] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, Springer, 2015, pp. 234–241.

[20] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," *Springer, Cham*, 2018.

[21] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *ICLR*.

[22] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 077–12 090, 2021.

[23] R. Strudel, R. Garcia, I. Laptev, and C. Schmid, "Segmenter: Transformer for semantic segmentation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 7262–7272.

[24] J.-J. Hwang, S. X. Yu, J. Shi, *et al.*, "Segsort: Segmentation by discriminative sorting of segments," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7334–7344.

[25] B. Zhang, Z. Tian, Q. Tang, X. Chu, X. Wei, C. Shen, *et al.*, "Segvit: Semantic segmentation with plain vision transformers," *Advances in Neural Information Processing Systems*, vol. 35, pp. 4971–4982, 2022.

[26] W. Li, Z. Wang, X. Yang, *et al.*, "Libfewshot: A comprehensive library for few-shot learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[27] S. Liu, A. Davison, and E. Johns, "Self-supervised generalisation with meta auxiliary learning," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[28] Y. Chen, Z. Liu, H. Xu, T. Darrell, and X. Wang, "Meta-baseline: Exploring simple meta-learning for few-shot learning," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 9062–9071.

[29] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, "Meta-learning in neural networks: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 9, pp. 5149–5169, 2021.

[30] Y. Tian, Y. Wang, D. Krishnan, J. B. Tenenbaum, and P. Isola, "Rethinking few-shot image classification: A good embedding is all you need?" In *ECCV*, 2020, pp. 266–282.

[31] A. A. Rusu, D. Rao, J. Sygnowski, *et al.*, "Meta-learning with latent embedding optimization," 2018.

[32] H. Liu, L. Gu, Z. Chi, *et al.*, "Few-shot class-incremental learning via entropy-regularized data-free replay," in *European Conference on Computer Vision*, Springer, 2022, pp. 146–162.

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

[34] 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*, 2019, pp. 5217–5226.

[35] K. Wang, J. H. Liew, Y. Zou, D. Zhou, and J. Feng, "Panet: Few-shot image semantic segmentation with prototype alignment," in *ICCV*, 2019, pp. 9197–9206.

[36] 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*, vol. 44, no. 2, pp. 1050–1065, 2020.

[37] L. Yang, W. Zhuo, L. Qi, Y. Shi, and Y. Gao, "Mining latent classes for few-shot segmentation," in *CVPR*, 2021, pp. 8721–8730.

[38] H. Liu, P. Peng, T. Chen, Q. Wang, Y. Yao, and X.-S. Hua, "Fecanet: Boosting few-shot semantic segmentation with feature-enhanced context-aware network," *IEEE Transactions on Multimedia*, 2023.

[39] C. Lang, G. Cheng, B. Tu, and J. Han, "Learning what not to segment: A new perspective on few-shot segmentation," in *CVPR*, 2022, pp. 8057–8067.

[40] Y. Wu, Y. Chen, L. Wang, *et al.*, "Large scale incremental learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 374–382.

[41] A. Cheraghian, S. Rahman, S. Ramasinghe, *et al.*, "Synthesized feature based few-shot class-incremental learning on a mixture of subspaces," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 8661–8670.

[42] X. Tao, X. Hong, X. Chang, S. Dong, X. Wei, and Y. Gong, "Few-shot class-incremental learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 183–12 192.

[43] B. Yang, M. Lin, B. Liu, *et al.*, "Learnable expansion-and-compression network for few-shot class-incremental learning," *arXiv preprint arXiv:2104.02281*, 2021.

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

[45] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, *et al.*, "Microsoft coco: Common objects in context," in *ECCV*, 2014, pp. 740–755.

[46] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 633–641.

[47] M. Cordts, M. Omran, S. Ramos, *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[48] X. Li, T. Wei, Y. P. Chen, Y.-W. Tai, and C.-K. Tang, "Fss-1000: A 1000-class dataset for few-shot segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2869–2878.

[49] O. J. Tobias and R. Seara, "Image segmentation by histogram thresholding using fuzzy sets," *IEEE transactions on Image Processing*, vol. 11, no. 12, pp. 1457–1465, 2002.

[50] S. S. Al-Amri, N. V. Kalyankar, *et al.*, "Image segmentation by using threshold techniques," *arXiv preprint arXiv:1005.4020*, 2010.

[51] Z. Kato, J. Zerubia, *et al.*, "Markov random fields in image segmentation," *Foundations and Trends® in Signal Processing*, vol. 5, no. 1–2, pp. 1–155, 2012.

[52] K. Held, E. Kops, B. Krause, W. Wells, R. Kikinis, and H.-W. Muller-Gartner, "Markov random field segmentation of brain mr images," *IEEE Transactions on Medical Imaging*, vol. 16, no. 6, pp. 878–886, 1997. DOI: 10.1109/42.650883.

[53] Y. Ganin and V. Lempitsky, "Fields: Neural network nearest neighbor fields for image transforms," in *Asian conference on computer vision*, Springer, 2014, pp. 536–551.

[54] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1520–1528.

[55] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1520–1528.

[56] 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, 2018.

[57] A. Kirillov, R. Girshick, K. He, and P. Dollár, "Panoptic feature pyramid networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 6399–6408.

[58] Z. Zhong, Z. Q. Lin, R. Bidart, *et al.*, "Squeeze-and-attention networks for semantic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 13 065–13 074.

[59] J. Fu, J. Liu, H. Tian, *et al.*, "Dual attention network for scene segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3146–3154.

[60] L. Li, W. Wang, and Y. Yang, "Logicseg: Parsing visual semantics with neural logic learning and reasoning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4122–4133.

[61] C. Liang, W. Wang, J. Miao, and Y. Yang, "Gmmseg: Gaussian mixture based generative semantic segmentation models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 31 360–31 375, 2022.

[62] W. Wang, T. Zhou, F. Yu, J. Dai, E. Konukoglu, and L. Van Gool, "Exploring cross-image pixel contrast for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7303–7313.

[63] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[64] H. Li, P. Xiong, J. An, and L. Wang, "Pyramid attention network for semantic segmentation," *arXiv preprint arXiv:1805.10180*, 2018.

[65] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," 2017. [Online]. Available: `https://arxiv.org/pdf/1706.03762.pdf`.

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

[67] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *International conference on machine learning*, PMLR, 2021, pp. 10 347–10 357.

[68] Z. Liu, Y. Lin, Y. Cao, *et al.*, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[69] Y. Jiang, S. Chang, and Z. Wang, "Transgan: Two pure transformers can make one strong gan, and that can scale up," *Advances in Neural Information Processing Systems*, vol. 34, pp. 14 745–14 758, 2021.

[70] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*, Springer, 2020, pp. 213–229.

[71] S. Zheng, J. Lu, H. Zhao, *et al.*, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 6881–6890.

[72] J. Chen, Y. Lu, Q. Yu, *et al.*, "Transunet: Transformers make strong encoders for medical image segmentation," *arXiv preprint arXiv:2102.04306*, 2021.

[73] Y. Sha, Y. Zhang, X. Ji, and L. Hu, "Transformer-unet: Raw image processing with unet," *arXiv preprint arXiv:2109.08417*, 2021.

[74] H. Wang, Y. Zhu, H. Adam, A. Yuille, and L.-C. Chen, "Max-deeplab: End-to-end panoptic segmentation with mask transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 5463–5474.

[75] B. Cheng, A. Schwing, and A. Kirillov, "Per-pixel classification is not all you need for semantic segmentation," *Advances in Neural Information Processing Systems*, vol. 34, pp. 17 864–17 875, 2021.

[76] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *ICML*, PMLR, 2017, pp. 1126–1135.

[77] Y. Hu, S. Zhang, X. Chen, and N. He, "Biased stochastic first-order methods for conditional stochastic optimization and applications in meta learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 2759–2770, 2020.

[78] S. Furrer, Y. E. Erginbas, and M. Kayaalp, "Meta-learner lstm for few shot learning,"

[79] K. Lee, S. Maji, A. Ravichandran, and S. Soatto, "Meta-learning with differentiable convex optimization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 657–10 665.

[80] G. Koch, R. Zemel, R. Salakhutdinov, *et al.*, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, Lille, vol. 2, 2015.

[81] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1199–1208.

[82] 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.

[83] X. Yang, X. Nan, and B. Song, "D2n4: A discriminative deep nearest neighbor neural network for few-shot space target recognition," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 5, pp. 3667–3676, 2020.

[84] H. Li, W. Dong, X. Mei, C. Ma, F. Huang, and B.-G. Hu, "Lgm-net: Learning to generate matching networks for few-shot learning," in *International conference on machine learning*, PMLR, 2019, pp. 3825–3834.

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

[86] F. Hao, F. He, J. Cheng, L. Wang, J. Cao, and D. Tao, "Collect and select: Semantic alignment metric learning for few-shot learning," in *Proceedings of the IEEE/CVF international Conference on Computer Vision*, 2019, pp. 8460–8469.

[87] C. Doersch, A. Gupta, and A. Zisserman, "Crosstransformers: Spatially-aware few-shot transfer," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 981–21 993, 2020.

[88] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*, PMLR, 2020, pp. 1597–1607.

[89] R. Zhang, T. Che, Z. Ghahramani, Y. Bengio, and Y. Song, "Metagan: An adversarial approach to few-shot learning," *Advances in neural information processing systems*, vol. 31, 2018.

[90] Z. Chen, Y. Fu, Y.-X. Wang, L. Ma, W. Liu, and M. Hebert, "Image deformation meta-networks for one-shot learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 8680–8689.

[91] T. Chen, G.-S. Xie, Y. Yao, *et al.*, "Semantically meaningful class prototype learning for one-shot image segmentation," *IEEE Transactions on Multimedia*, vol. 24, pp. 968–980, 2021.

[92] T. Zhou, W. Wang, E. Konukoglu, and L. Van Gool, "Rethinking semantic segmentation: A prototype view," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2582–2593.

[93] W. Wang, T. Zhou, F. Yu, J. Dai, E. Konukoglu, and L. Van Gool, "Exploring cross-image pixel contrast for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7303–7313.

[94] Y. Liu, X. Zhang, S. Zhang, and X. He, "Part-aware prototype network for few-shot semantic segmentation," in *ECCV*, 2020, pp. 142–158.

[95] 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*, Springer, 2020, pp. 730–746.

[96] B. Liu, J. Jiao, and Q. Ye, "Harmonic feature activation for few-shot semantic segmentation," *IEEE Transactions on Image Processing*, vol. 30, pp. 3142–3153, 2021.

[97] T. Hu, P. Yang, C. Zhang, G. Yu, Y. Mu, and C. G. Snoek, "Attention-based multi-context guiding for few-shot semantic segmentation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, 2019, pp. 8441–8448.

[98] Y. Yang, F. Meng, H. Li, Q. Wu, X. Xu, and S. Chen, "A new local transformation module for few-shot segmentation," in *MultiMedia Modeling: 26th International Conference, MMM 2020, Daejeon, South Korea, January 5–8, 2020, Proceedings, Part II 26*, Springer, 2020, pp. 76–87.

[99] X. Zhang, Y. Wei, Y. Yang, and T. S. Huang, "Sg-one: Similarity guidance network for one-shot semantic segmentation," *IEEE Transactions on cybernetics*, vol. 50, no. 9, pp. 3855–3865, 2020, ISSN: 2168-2267. DOI: 10.1109/TCYB.2020.2992433. [Online]. Available: https://ieeexplore.ieee.org/document/9108530/.

[100] Y. Liu, B. Jiang, and J. Xu, "Axial assembled correspondence network for few-shot semantic segmentation," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 3, pp. 711–721, 2022.

[101] 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*, 2019, pp. 9587–9595.

[102] G. Zhang, G. Kang, Y. Yang, and Y. Wei, "Few-shot segmentation via cycle-consistent transformer," *NeurIPS*, vol. 34, pp. 21 984–21 996, 2021.

[103] H. Gao, J. Xiao, Y. Yin, T. Liu, and J. Shi, "A mutually supervised graph attention network for few-shot segmentation: The perspective of fully utilizing limited samples," *IEEE Transactions on neural networks and learning systems*, 2022.

[104] X. Shi, D. Wei, Y. Zhang, *et al.*, "Dense cross-query-and-support attention weighted mask aggregation for few-shot segmentation," in *European Conference on Computer Vision*, Springer, 2022, pp. 151–168.

[105] Y. Zhuge and C. Shen, "Deep reasoning network for few-shot semantic segmentation," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 5344–5352.

[106] L. Liu, J. Cao, M. Liu, Y. Guo, Q. Chen, and M. Tan, "Dynamic extension nets for few-shot semantic segmentation," in *Proceedings of the 28th ACM international conference on multimedia*, 2020, pp. 1441–1449.

[107] X. Yang, B. Wang, K. Chen, *et al.*, "Brinet: Towards bridging the intra-class and inter-class gaps in one-shot segmentation," *arXiv preprint arXiv:2008.06226*, 2020.

[108] P. Tian, Z. Wu, L. Qi, L. Wang, Y. Shi, and Y. Gao, "Differentiable meta-learning model for few-shot semantic segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 12 087–12 094.

[109] 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*, 2021, pp. 13 979–13 988.

[110] G. Zheng and A. Zhang, "Few-shot class-incremental learning with meta-learned class structures," in *2021 International Conference on Data Mining Workshops (ICDMW)*, IEEE, 2021, pp. 421–430.

[111] Z. Chi, L. Gu, H. Liu, Y. Wang, Y. Yu, and J. Tang, "Metafscil: A meta-learning approach for few-shot class incremental learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 14 166–14 175.

[112] Y. Zou, S. Zhang, Y. Li, and R. Li, "Margin-based few-shot class-incremental learning with class-level overfitting mitigation," *Advances in neural information processing systems*, vol. 35, pp. 27 267–27 279, 2022.

[113] A. F. Akyürek, E. Akyürek, D. T. Wijaya, and J. Andreas, "Subspace regularizers for few-shot class incremental learning," *arXiv preprint arXiv:2110.07059*, 2021.

[114] D.-Y. Kim, D. J. Han, J. Seo, and J. Moon, "Warping the space: Weight space rotation for class-incremental few-shot learningdl," in *The International Conference on Learning Representations, ICLR 2023*, ICLR, 2023.

[115] G. Shi, J. Chen, W. Zhang, L.-M. Zhan, and X.-M. Wu, "Overcoming catastrophic forgetting in incremental few-shot learning by finding flat minima," *Advances in neural information processing systems*, vol. 34, pp. 6747–6761, 2021.

[116] P. Mazumder, P. Singh, and P. Rai, "Few-shot lifelong learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 2021, pp. 2337–2345.

[117] A. Kukleva, H. Kuehne, and B. Schiele, "Generalized and incremental few-shot learning by explicit learning and calibration without forgetting," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 9020–9029.

[118] A. Agarwal, B. Banerjee, F. Cuzzolin, and S. Chaudhuri, "Semantics-driven generative replay for few-shot class incremental learning," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 5246–5254.

[119] B. Yang, M. Lin, Y. Zhang, *et al.*, "Dynamic support network for few-shot class incremental learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 2945–2951, 2022.

[120] J. Yoon, S. Madjid, S. J. Hwang, C.-D. Yoo, *et al.*, "On the soft-subnetwork for few-shot class incremental learning," in *International Conference on Learning Representations (ICLR) 2023*, International Conference on Learning Representations, 2023.

[121] G. Koch, R. Zemel, R. Salakhutdinov, *et al.*, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, Lille, vol. 2, 2015.

[122] 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 *CVPR*, 2019, pp. 5217–5226.

[123] B. Yang, C. Liu, B. Li, J. Jiao, and Q. Ye, "Prototype mixture models for few-shot semantic segmentation," in *ECCV*, 2020, pp. 763–778.

[124] G. Li, V. Jampani, L. Sevilla-Lara, D. Sun, J. Kim, and J. Kim, "Adaptive prototype learning and allocation for few-shot segmentation," in *CVPR*, 2021, pp. 8334–8343.

[125] 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 *CVPR*, 2021, pp. 8741–8750.

[126] J. Min, D. Kang, and M. Cho, "Hypercorrelation squeeze for few-shot segmentation," in *CVPR*, 2021, pp. 6941–6952.

[127] B. Zhang, J. Yuan, B. Li, T. Chen, J. Fan, and B. Shi, "Learning cross-image object semantic relation in transformer for few-shot fine-grained image classification," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 2135–2144.

[128] W. Liu, C. Zhang, G. Lin, and F. Liu, "Crnet: Cross-reference networks for few-shot segmentation," in *CVPR*, 2020, pp. 4165–4173.

[129] A. Shaban, S. Bansal, Z. Liu, *et al.*, "One-shot learning for semantic segmentation," *arXiv preprint arXiv:1709.03410*, 2017.

[130] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik, "Semantic contours from inverse detectors," in *ICCV*, 2011, pp. 991–998.

[131] K. Nguyen and S. Todorovic, "Feature weighting and boosting for few-shot segmentation," in *CVPR*, 2019, pp. 622–631.

[132] Z. Zheng, G. Huang, X. Yuan, C.-M. Pun, H. Liu, and W.-K. Ling, "Quaternion-valued correlation learning for few-shot semantic segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, 2022.

[133] J. Liu and Y. Qin, "Prototype refinement network for few-shot segmentation," *arXiv preprint arXiv:2002.03579*, 2020.

[134] 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 *CVPR*, 2021, pp. 13 979–13 988.

[135] K. Rakelly, E. Shelhamer, T. Darrell, A. A. Efros, and S. Levine, "Few-shot segmentation propagation with guided networks," *arXiv preprint arXiv:1806.07373*, 2018.

[136] H. Huang, J. Zhang, J. Zhang, J. Xu, and Q. Wu, "Low-rank pairwise alignment bilinear network for few-shot fine-grained image classification," *IEEE Transactions on Multimedia*, vol. 23, pp. 1666–1680, 2021. DOI: 10.1109/TMM.2020.3001510.

[137] A. Phaphuangwittayakul, Y. Guo, and F. Ying, "Fast adaptive meta-learning for few-shot image generation," *IEEE Transactions on Multimedia*, vol. 24, pp. 2205–2217, 2021.

[138] X. Zhong, C. Gu, M. Ye, W. Huang, and C.-W. Lin, "Graph complemented latent representation for few-shot image classification," *IEEE Transactions on Multimedia*, vol. 25, pp. 1979–1990, 2023. DOI: 10.1109/TMM.2022.3141886.

[139] K. Guo, C. Shen, B. Hu, M. Hu, and X. Kui, "Rsnet: Relation separation network for few-shot similar class recognition," *IEEE Transactions on Multimedia*, vol. 25, pp. 3894–3904, 2023. DOI: 10.1109/TMM.2022.3168146.

[140] Y. Zhu, Z. Zhang, C. Wu, *et al.*, "Improving semantic segmentation via efficient self-training," *IEEE transactions on pattern analysis and machine intelligence*, 2021.

[141] S. Liu, S. Zhi, E. Johns, and A. J. Davison, "Bootstrapping semantic segmentation with regional contrast," *arXiv preprint arXiv:2104.04465*, 2021.

[142] D. Li, J. Yang, K. Kreis, A. Torralba, and S. Fidler, "Semantic segmentation with generative models: Semi-supervised learning and strong out-of-domain generalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8300–8311.

[143] X. Zhang, Y. Wei, Y. Yang, *et al.*, "Sg-one: Similarity guidance network for one-shot semantic segmentation," *IEEE Trans. Cybern*, vol. 50, no. 9, pp. 3855–3865, 2020, ISSN: 2168-2267. DOI: 10.1109/TCYB.2020.2992433. [Online]. Available: https://ieeexplore.ieee.org/document/9108530/.

[144] B. Yang, C. Liu, B. Li, J. Jiao, and Q. Ye, "Prototype mixture models for few-shot semantic segmentation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*, Springer, 2020, pp. 763–778.

[145] Y. Liu, N. Liu, X. Yao, and J. Han, "Intermediate prototype mining transformer for few-shot semantic segmentation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 38 020–38 031, 2022.

[146] B. Peng, Z. Tian, X. Wu, *et al.*, "Hierarchical dense correlation distillation for few-shot segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 23 641–23 651.

[147] X. Shi, D. Wei, Y. Zhang, *et al.*, "Dense cross-query-and-support attention weighted mask aggregation for few-shot segmentation," in *European Conference on Computer Vision*, Springer, 2022, pp. 151–168.

[148] T. Zhou, W. Wang, E. Konukoglu, and L. Van Gool, "Rethinking semantic segmentation: A prototype view," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2582–2593.

[149] 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*, 2021, pp. 9747–9756.

[150] X. Lu, W. Wang, J. Shen, D. J. Crandall, and L. Van Gool, "Segmenting objects from relational visual data," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 11, pp. 7885–7897, 2021.

[151] J. Chen, B.-B. Gao, Z. Lu, J.-H. Xue, C. Wang, and Q. Liao, "Apanet: Adaptive prototypes alignment network for few-shot semantic segmentation," *IEEE Transactions on Multimedia*, 2022.

[152] 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*, 2021, pp. 9747–9756.

[153] W. Xu, H. Huang, M. Cheng, L. Yu, Q. Wu, and J. Zhang, "Masked cross-image encoding for few-shot segmentation," in *2023 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE, 2023, pp. 744–749.

[154] 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*, 2021, pp. 517–526.

[155] 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*, 2022, pp. 11 573–11 582.

[156] Y. Wang, R. Sun, and T. Zhang, "Rethinking the correlation in few-shot segmentation: A buoys view," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7183–7192.

[157] Q. Xu, W. Zhao, G. Lin, and C. Long, "Self-calibrated cross attention network for few-shot segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 655–665.

[158] 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*, 2021, pp. 8741–8750.

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

[160] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[161] 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*, 2021, pp. 8312–8321.

[162] Z. Tian, X. Lai, L. Jiang, *et al.*, "Generalized few-shot semantic segmentation," in *CVPR*, 2022, pp. 11 563–11 572.

[163] S. Hajimiri, M. Boudiaf, I. Ben A., *et al.*, "A strong baseline for generalized few-shot semantic segmentation," in *CVPR*, 2023, pp. 11 269–11 278.

[164] F. Cermelli, M. Mancini, Y. Xian, *et al.*, "Prototype-based incremental few-shot semantic segmentation," *arXiv preprint arXiv:2012.01415*, 2020.

[165] R. Qiu, P. Chen, W. Sun, *et al.*, "Gaps: Few-shot incremental semantic segmentation via guided copy-paste synthesis," 2022.

[166] Y. Zhou, X. Chen, Y. Guo, *et al.*, "Advancing incremental few-shot semantic segmentation via semantic-guided relation alignment and adaptation," *arXiv preprint arXiv:2305.10868*, 2023.

[167] X. Tao, X. Hong, X. Chang, *et al.*, "Few-shot class-incremental learning," in *CVPR*, 2020, pp. 12 183–12 192.

[168] J. Zhu, G. Yao, W. Zhou, *et al.*, "Feature distribution distillation-based few shot class incremental learning," in *PRAI*, IEEE, 2022, pp. 108–113.

[169] S. Dong, X. Hong, X. Tao, X. Chang, X. Wei, and Y. Gong, "Few-shot class-incremental learning via relation knowledge distillation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 2021, pp. 1255–1263.

[170] M. Hersche, G. Karunaratne, G. Cherubini, L. Benini, A. Sebastian, and A. Rahimi, "Constrained few-shot class-incremental learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9057–9067.

[171] A. F. Akyürek, E. Akyürek, D. T. Wijaya, and J. Andreas, "Subspace regularizers for few-shot class incremental learning," *arXiv preprint arXiv:2110.07059*, 2021.

[172] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3733–3742.

[173] H. Qi, M. Brown, and D. G. Lowe, "Low-shot learning with imprinted weights," in *CVPR*, 2018, pp. 5822–5830.

[174] S. Gidaris and N. Komodakis, "Dynamic few-shot visual learning without forgetting," in *CVPR*, 2018, pp. 4367–4375.

[175] A. Kukleva, H. Kuehne, and B. Schiele, "Generalized and incremental few-shot learning by explicit learning and calibration without forgetting," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 9000–9009. DOI: 10.1109/ICCV48922.2021.00889.

[176] Y. Xian, S. Choudhury, Y. He, *et al.*, "Semantic projection network for zero-and few-label semantic segmentation," in *CVPR*, 2019, pp. 8256–8265.

[177] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, 2013.

[178] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, "Fasttext. zip: Compressing text classification models," *arXiv preprint arXiv:1612.03651*, 2016.

[179] F. Cermelli, M. Mancini, S. R. Bulo, *et al.*, "Modeling the background for incremental learning in semantic segmentation," in *CVPR*, 2020, pp. 9233–9242.