Generative Deep Learning Methods for Improving Few-Shot Segmentation of Infrared Images

A THESIS

SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL

OF THE UNIVERSITY OF MINNESOTA

BY

Junno Yun

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

Master of Science

Prof. Mehmet Akçakaya, advisor

July, 2023

Acknowledgements

I would like to extend my sincere appreciation to Prof. Mehmet Akçakaya for his exceptional support, invaluable advice, and unwavering patience during my academic journey at the University of Minnesota. It has been an immense honor to be his student, and his mentorship has been instrumental in shaping my research and personal growth. I was incredibly fortunate to have the opportunity to take his image processing course, which ignited my passion and inspired me to explore the fascinating fields of computer vision and image semantic segmentation. I am immensely grateful to him for providing me with the chance to explore the captivating realm of a few-shot segmentation on Infrared images.

I am also deeply grateful to my committee members, Prof. Patrick Bolan and Prof. Mingyi Hong, for their consistent support and thoughtful feedback throughout my thesis preparation and final examination. Their expertise and dedication have significantly contributed to the quality of my work, and I am truly indebted for their mentorship.

Furthermore, I would like to thank the Minnesota Supercomputing Institute (MSI) for their supports. The powerful computational capabilities provided by MSI have been crucial in implementing and running the complex algorithms and codes for my research. Without the support and resources of MSI, I would not have been able to accomplish my thesis objectives.

Lastly, I would like to express my profound gratitude to my beloved family and friends for their unwavering love, encouragement, and support throughout this entire journey. I owe the successful completion of my thesis to every one of you.

Dedication

This thesis is dedicated to my loving wife Suji Kim. Thank you for always being there for me.

Abstract

Image semantic segmentation is an essential topic in computer vision. However, most current deep learning-based networks primarily focus on large-scale visible spectrum datasets, specifically RGB images. Consequently, these models often struggle to perform well on segmentation in adverse environmental conditions such as total darkness, poor illumination, and smog, etc. On the other hand, a thermal infrared (IR) imaging system overcomes the limitations of visible spectrum imaging. Leveraging the advantages of thermal IR imaging, semantic segmentation using infrared images has extensive real-world applications, including defense applications, autonomous driving, and medical imaging.

However, there remain challenges in harnessing the benefits of IR images. Firstly, accessing large-scale annotated IR images is challenging due to security considerations. Secondly, we can face data with unseen classes or rare categories in various IR imaging applications. Lastly, thermal IR images typically exhibit low-resolution, low-contrast, and obscure object boundaries due to the nature of thermal cameras. These images are composed of a single grayscale channel, which provides limited information compared to RGB images. Thus, applying only IR images to existing semantic segmentation models leads to inaccurate performance.

In this thesis, our aim is to improve existing few-shot segmentation models to enable robust few-shot segmentation of IR images. To this end, we propose the use of generative methods to enhance our few-shot segmentation model by generating two types of synthesized data: one for augmenting the training data and the other for providing conditioning information. Results show that these strategies substantially improve few-shot segmentation of IR images.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Objective and Challenges

The interest in the segmentation of infrared (IR) images has increased rapidly following the advent of powerful deep learning methods for other segmentation tasks, including dense predictions for semantic segmentation. IR images have significant advantages in adverse environmental conditions such as low illumination, smog, and fog. The utilization of IR spectrum for semantic segmentation presents a wide range of real-world use cases, including defense applications, autonomous driving and medical imaging. Consequently, designing accurate semantic segmentation models for the IR spectrum has become an important challenge.

Designing IR image segmentation encounters several challenges and considerations. Firstly, despite the popularity and potential of IR image segmentation, obtaining large-scale annotated IR datasets remains difficult for several reasons. For instance, IR imaging systems are often expensive technologies, and most IR images are used in military or medical applications, making them inaccessible due to security and intellectual property rights concerns. As a result, the publicly available IR datasets are limited compared to the vast number of labeled RGB images [1]. This scarcity of annotated IR image datasets limits the training of the deep learning networks, particularly in supervised learning scenarios. Additionally, data with unseen or rare classes are common in various IR image applications, which poses a challenge for supervised convolutional neural networks (CNNs) to generalize to images containing new classes. Lastly, thermal IR images typically contain low resolution, low contrast, and unclear object boundaries captured by thermal cameras [2]. IR images consist of a grayscale 1-channel, which provides limited information compared to RBG's 3-channel images. Therefore, relying on only IR images when applying existing semantic segmentation models causes the models to perform inaccurately.

In this thesis, our objective is to overcome the aforementioned challenges and improve the segmentation of thermal IR images. To accomplish the goal, we consider the scenario in which labeled IR datasets are scarce and new classes not included during training are present.

1.2 Our Approach

In our work, we address the challenges of the limited availability of IR datasets and the need for generalization to unseen datasets by considering a *Few-Shot Segmentation* task. Additionally, we utilize *Generative Deep Learning Methods* to address the issues of the insufficiency and limited information of IR images. **Figure 1** illustrates our proposed methods.

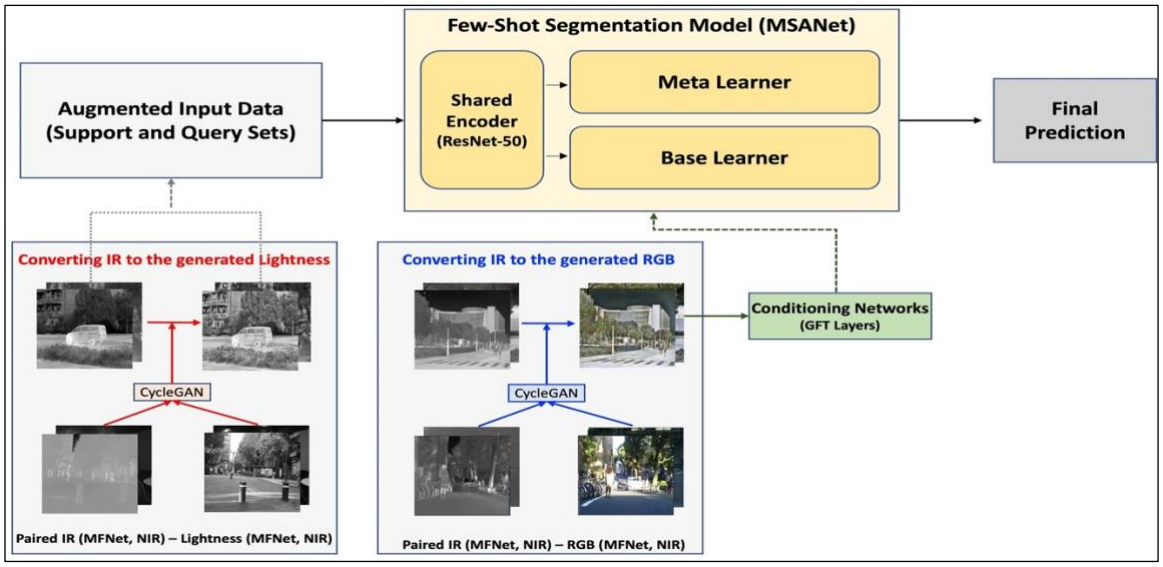

Figure 1: The overall architecture of our proposed methods applied to a few-shot segmentation model.

1.2.1. Few-Shot Segmentation

We demonstrate how the few-shot segmentation model works on IR images. To incorporate IR images into the few-shot segmentation model, we introduce the SODA-5^i dataset by partitioning the SODA dataset [2] into four folds. A detailed explanation of this dataset can be found in Section 3.2.1. For our few-shot segmentation model, we adopt MSANet [3], which combines a 'Base learner' and a 'Meta learner' trained using supervised learning and few-shot learning, respectively. This model is used to show how our proposed methods enhance the results in both few-shot and supervised learning scenarios simultaneously.

1.2.2. Generative Deep Learning Methods (CycleGAN Applications)

As a representative method in the field of ‘Generative Deep Learning Method’, Cycle-Consistent Generative Adversarial Networks (CycleGAN) [4] enables image-to-image translations between two different domains in various fields using unpaired or paired dataset. In our study, we leverage CycleGAN to generate two kinds of images: ‘Generated Lightness Images’ for data augmentation and ‘Generated RGB Images’ for conditioning prior information.

Generated Lightness Images for Data Augmentation: Data Augmentation is a common and effective method to improve the model performance. Traditionally, simple techniques such as cropping, rotation, blurring, have been used for data augmentation. However, these techniques have limitations, particularly when the raw data is scarce. To address the scarcity of an IR dataset, we utilize an IR-lightness paired dataset, where the lightness domain images are obtained by converting RGB images (please refer to Section 3.2.2 for detailed information). The purpose of using CycleGAN in this context is to train a model that converts the IR domain to the lightness domain. By applying the trained CycleGAN model to SODA-5^i training dataset, we obtain a generated lightness dataset that has higher contrast without losing the inherent properties of the original images. These generated images are used to increase the training data and enhance the performance of the few-shot segmentation model through augmentation-based learning method [5, 6].

Generated RGB Images as Conditioning Prior Information: The performance of a segmentation model can be enhanced when an image contains additional channel information, such as a 3-channel image rather than 1-channel image. To deal with the limited information in IR images, we generate RGB images using different RGB datasets. Leveraging CycleGAN [4], we utilize a paired IR-RGB dataset to convert our IR dataset to the RGB domain. This generated RGB dataset enriches the IR image as prior information by employing 'Conditioned Networks,' which will be introduced in the next Section 1.2.3.

1.2.3. Conditioned Networks (Gated Feature-wise Transform)

Using the generated RGB images from Section 1.2.2 directly such as replacing existing data or concatenation at the network input may degrade performance. Li et al. [2] proposed a gated feature-wise transform (GFT) to incorporate conditioning data as a prior information for image

segmentation. In our study, we employ GFT layers to leverage the information from the generated RGB images.

Chapter 2

Background

2.1 Few-Shot Image Segmentation

Deep learning has made significant advancements in semantic segmentation with various CNN models [7, 8, 9]. However, learning these models requires a large-scale dataset with pixel-level annotations, which can be expensive and labor-intensive to obtain. Moreover, recent models also have struggled to generalize to unseen classes. To deal with these current challenges, there has been notable progress in few-shot segmentation tasks [10, 11, 12, 13, 14], aiming to learn effective segmentation from only a small number of labeled examples. In this thesis, we employ the MSANet [3] as our few-shot segmentation model.

2.1.1. Few-Shot Learning (FSL)

Few-shot learning aims to train the model for generalizing knowledge obtained from classes seen previously to new classes with only a few examples. Although FSL approaches can be specified and defined in many ways, they can be categorized into four main types: transfer learning-based [15], optimization-based [16, 17, 18], augmentation-based [5, 6], and metric-based [19, 20, 21, 22]. In this thesis, we focus on augmentation-based and metric-based approaches.

Augmentation-based FSL approaches have a simple concept that attempts to augment the data using various techniques. One approach in augmentation-based FSL is to utilize generative networks, which aim to synthesize and translate entirely new data from the existing data domain [5, 6]. Even though data augmentation does not provide entirely new information, it is still valuable in handling the limited data available for FSL training.

Metric-based approaches are widely employed in the few-shot segmentation (FSS) task with the goal of learning to compare. Siamese networks [19], matching networks [22], prototypical networks [21], and relation networks [20] are representative examples of metric learning. The primary

concept behind these approaches is to calculate the similarity between features extracted from a support set (containing labeled examples) and a query set (containing unlabeled examples) in the embedding space. This similarity is evaluated using various metrics such as distance score, mean vectors (prototypes), or relations score. The networks are then optimized based on the similarity measures to improve the performance of few-shot segmentation.

In this thesis, we adopt MSANet [3] model which is designed based on the closely related prototypical networks of metric-based learning. MSANet model leverages the concept of multi-similarity between support set and query set in the embedding space. Additionally, we incorporate an augmentation-based learning method by utilizing CycleGANs. A CycleGAN is employed to enhance the performance of our few-shot segmentation model by generating synthetic data for data augmentation. By training the CycleGAN on paired IR-lightness datasets, we generate realistic synthetic data in the lightness domain. These generated lightness data augment the training set and help overcome the lack of IR data, leading to improved performance in few-shot segmentation.

2.1.2. Few-Shot Segmentation (FSS)

There is also abundant amount of research focusing on few-shot segmentation as a part of FSL in computer vision. Shaban et al. [10] proposed an FSS model that utilizes a two-branch architecture. The first branch, called the conditioning branch, generates classifier parameters by taking support images as input. These parameters capture relevant information from the support images. The second branch, known as the segmentation branch, receives a query image as input and generates a segmentation mask as output, effectively applying the learned parameters from the first branch to perform segmentation on the query image.

There have been outstanding achievements in FSS by exploiting the prototype learning method, one of the metric-based methods discussed in Section 2.1.1. One notable advancement is the introduction of masked average pooling, which enhances the extractions of class representative prototype vectors from the support set [13]. PFENet [23] proposed non-parametric prior mask generation method that calculates cosine similarity on high-level features to achieve high generalization. HSNet [24] designed the hyper-correlation squeeze networks that leverage multi-level feature correlations between a pair of input images and utilize lightweight 4D convolutions to predict fine-grained segmentation mask in a query image. Another approach, an attention-based multi-context guiding network [25], has been proposed to emphasize context information from small-to-large scale for guiding query branches globally. Additionally, several notable approaches

have been introduced to utilize more guidance from class representative prototype vectors [14, 13, 26].

To alleviate the issues of bias towards the seen classes in previous frameworks that depend on meta-learning, Base and Meta (BAM) architecture [27] introduced a novel perspective on FSS consisting of three components: a base learner, a meta learner, and an ensemble module. The base learner is trained in a supervised manner on base classes that are already known. Its role is to explicitly predict the regions belonging to the base classes in the query images and suppress falsely activated regions of base categories in the meta learner output. The meta learner aims to recognize novel classes that have not seen before. Lastly, the ensemble module adaptively integrates the coarse predictions from both learners to generate accurate segmentation results.

Building upon the architecture of BAM, MSANet [3] further enhances the FSS model by introducing two guiding modules: the multi-similarity module and the lightweight CNN attention block in the meta-learner component. The multi-similarity module calculates visual correspondences between the features extracted from the intermediate and high-level layers of support and query image. This module helps establish meaningful relationships between the two sets of features. The lightweight attention module leads the meta learner to focus on the ideal target of the query image.

During training, base learner and meta learner of MSANet [3] are trained in a supervised and few-shot learning way, respectively. This approach is meaningful as it allows for the simultaneous evaluation of our proposed methods in both training scenarios.

2.1.3. Problem Definition

FSS tasks aim to segment the region belonging to unseen classes with a few labeled support samples. Succinctly, FSS dataset is divided into two sets: $\mathcal{D}_{train}$ and $\mathcal{D}_{test}$. These do not overlap in terms of object categories. $\mathcal{D}_{train}$ is composed of seen class (*Base*), while $\mathcal{D}_{test}$ consists of unseen class (*Novel*). The models are optimized on $\mathcal{D}_{train}$ to learn transferable knowledge with a few labeled examples. The objective is for the models to demonstrate good generalization on $\mathcal{D}_{test}$ with insufficient annotated samples.

The episodic training paradigm [22] has been the technique employed by most FSS models. Episode refers to a simulated learning scenario in which new objects with limited annotated examples are encountered. As shown in **Figure 2,** each episode consists of a small support set $S =$

$\{(\mathbf{x}_i^s, \mathbf{m}_i^s)\}_{i=1}^K$ and a query set $Q = \{(\mathbf{x}^q, \mathbf{m}^q)\}$, where $\mathbf{x}$ represents an input image and $\mathbf{m}$ denotes its corresponding binary mask for a specific category. During the training stage, each episode is randomly selected from the *Base* ($\mathcal{D}_{train}$) dataset. However, during testing, each episode is randomly chosen from the *Novel* ($\mathcal{D}_{test}$) dataset. Note that query masks $\mathbf{m}^q$ are used for validation during training, and for testing during the actual testing phase. *K* indicates the number of support images in each episode. In recent FSS methods, *K* is commonly set to one or five, referred to as 1-shot and 5-shot, respectively.

During each episode of $\mathcal{D}_{train}$, FSS models are trained to make predictions for the query image $\mathbf{x}^q$ based on the support set S. For instance, if *K* is set as one, the models predict the regions corresponding to the designated class in the query image $\mathbf{x}^q$ with the assistance of the one support set $\{(\mathbf{x}_i^s, \mathbf{m}_i^s)\}_{i=1}$. Once the training is complete, the performance evaluation of FSS models is conducted on $\mathcal{D}_{test}$ across all the test episodes.

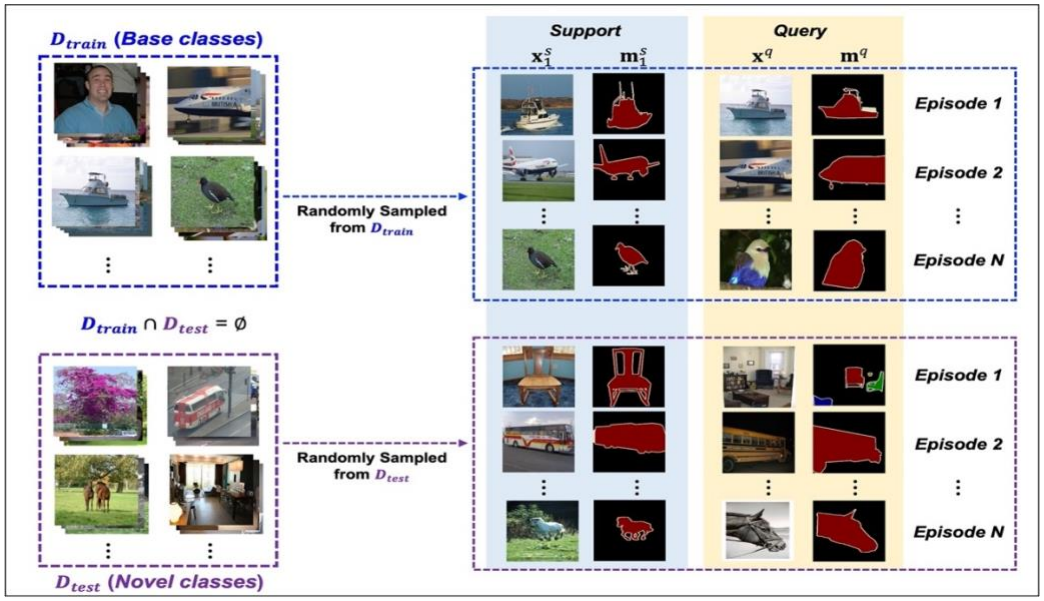

Figure 2: Illustration of 1-shot episodic training in the FSS task (figure adapted from **[28]**).

As mentioned earlier in Section 2.1.2., MSANet [3] comprises a base learner and a meta learner. The meta learner follows the episodic training paradigm using the defined data structure as described above. On the other hand, the base learner complies with the traditional supervised method, utilizing only the $\mathcal{D}_{train}$ $(Base)$ set. The composition of FSS datasets is further elaborated in Section 3.2.1.

2.2 Generative Adversarial Networks (CycleGAN)

Generative modeling involves creating new examples that plausibly come from an existing distribution of samples, such as generating new images that are similar but specifically different from a dataset of existing images. Generative Adversarial Networks (GANs) [29] are generative models that use two neural networks, called Discriminator and Generator. These two networks compete against each other. The generator learns to generate new plausible looking fake images, while the discriminator learns to differentiate whether the data is real or fake. During training, both networks are optimized in a competitive manner. The generator is optimized to create fake data that can deceive the discriminator into classifying it as real, while the discriminator is trained to better classify the generated images.

In certain real-world scenarios, it may be difficult to find paired sets of images. For example, obtaining paired images between the thermal infrared spectrum and the visible spectrum can be challenging. To address this issue, CycleGAN [4] was proposed to be able to translate image-to-image using unpaired images. CycleGAN extends the GAN architecture and involves training two generator networks and two discriminator networks simultaneously. One generator converts input from the first domain to the second domain, while the other generator converts input from the second domain to the first domain. The discriminators take the generated images from each generator and find the likelihood of the generated image belonging to their respective domains.

CycleGAN introduces an additional concept called cycle consistency, which guarantees that the reconstructed image, obtained by passing an image through the first generator and then the second generator, should be similar to the original input image. The cycle consistency loss is computed as the ℓ_1 norm between the input image and its reconstruction, serving as a regularization term for the generator models. By incorporating cycle consistency losses, CycleGAN encourages the generators to learn meaningful mappings between unpaired domains while preserving the inherent characteristics of the input images. Although CycleGAN is advantageous for image translation using unpaired data, it also exhibits strong capabilities in translating images using paired data as well.

CycleGAN has rapidly gained immense popularity due to its versatility and has been widely used in various applications such as style transfer, object transfiguration, photograph enhancement, and

more. Its flexibility makes it applicable to a wide range of deep learning tasks. In this thesis, CycleGAN is employed to facilitate the translation of images between two different domains, using existing paired data to expand the training dataset size and enrich the channel information.

2.3 Gated Feature-wise Transform

In many real-world cases, integrating different sources of information is necessary in various tasks, such as classification, and segmentation. For example, infrared and visible spectrum information have distinct insights as they are derived from different light spectra. By combining these sources, it becomes possible to compensate for each other's deficiencies [30], making them valuable auxiliary inputs for one another. In semantic segmentation task, several fusion networks have been proposed [31] to effectively merge the main input information with the auxiliary source information.

Finding an effective approach to condition on multiple sources of information is still a promising research field [32]. Perez et al. [33] proposed feature-wise affine transformations, which fuse and leverage the relationship between multiple information sources as a conditioning information. Wang et al. [34] proposed the spatial feature-wise transform (SFT) to incorporate spatial priors for image super-resolution. The formulation of SFT is as follows:

$$\mathrm{SFT}(\mathbf{x}) = \gamma(\mathbf{z}) \odot \mathbf{x} \oplus \beta(\mathbf{z}),$$

where $\mathbf{x}$ is an input of SFT layer, $\mathbf{z}$ is a conditioning input, and γ and β are $\mathbf{z}$-dependent functions that represent the scaling and shifting vectors, respectively. The symbols $\odot$ and $\oplus$ denote element-wise product and sum operations, respectively.

Inspired by SFT, Li et al. [2] proposed a gated feature-wise transform (GFT) to address the noise problems associated with conditioning on edge images. The main difference between GFT and SFT lies in the incorporation of a gate scheme as an information control, which adaptively incorporates prior information using a sigmoid function before applying the feature-wise transform. The structure of GFT is defined as follows:

$$(\gamma^*(\hat{\mathbf{z}}),\ \beta^*(\hat{\mathbf{z}})) = \sigma(Conv(\gamma(\hat{\mathbf{z}})), Conv\big(\beta(\hat{\mathbf{z}})\big))$$

$$\hat{\gamma}(\hat{\mathbf{z}}) = \gamma(\hat{\mathbf{z}}) \odot \gamma^*(\hat{\mathbf{z}}),$$

$$\hat{\beta}(\hat{\mathbf{z}}) = \beta(\hat{\mathbf{z}}) \odot \beta^*(\hat{\mathbf{z}})$$

$$\mathrm{GFT}(\mathbf{x}) = \hat{\gamma}(\hat{\boldsymbol{z}}) \odot \mathbf{x} \oplus \hat{\beta}(\hat{\boldsymbol{z}}),$$

where σ is the sigmoid function, and $\hat{\boldsymbol{z}}$ is the output of the feature modulation module, which ensures that the conditioning input **z** has the same size as the input features **x.** The *Con*v operation denotes a 1×1 the convolutional operation. GFT layers are employed in our FSS model instead of SFT layers due to the presence of noise and incomplete channel information in the generated RGB images.

Li et al. [2] explained that incorporating the prior information features from edge images into the low-level features of the input data using GFT block improves fusion in IR image semantic segmentation that is conducted by supervised learning task. However, by obtaining well-extracted intermediate or high-level features from the informative prior information, the GFT block can integrate of prior information features with the corresponding features of the input data. This late fusion approach allows the model to leverage both the available labeled data and the prior information, leading to improved segmentation performance in few-shot scenarios.

Therefore, in addition to its benefits in supervised learning tasks, we propose that the GFT block is valuable for late fusion in FSS, enabling the integration of conditioning information features into the intermediate or high-level features from the input data.

Chapter 3

Few-Shot Segmentation of IR images with Generative Deep Learning Methods

3.1 Introduction

Considering the significance of using IR images in various fields, it is worthwhile to explore the application of FSS on IR images. In this chapter, we present the details of how we incorporate IR images into the FSS model and how our proposed methods contribute to enhancing the performance of FSS on IR images.

3.2 Datasets

3.2.1. Few-Shot Segmentation Dataset

For the FSS problem described Section 2.1.3., PASCAL-5^i [10] and COCO-20^i [35] datasets are commonly used. For instance, PASCAL-5^i dataset is derived from PASCAL VOC 2012 [28] and contains 20 object categories. The dataset is specifically designed to fit the purpose of a few-shot task by uniformly dividing the object classes into four folds, denoted as $\{5^i: i \in \{0,1,2,3\}\}$, where each fold contains disjoint 5 classes. FSS models are trained on 3-folds and tested on the remaining fold in a cross-validation manner. The validation fold comprises 1,000 randomly paired samples, consisting of support images and a query.

Following the methodology of creating the PASCAL-5^i dataset, we partitioned our dataset for use in the FSS task. Our dataset is the SODA (Segmenting Objects in Day And night) dataset proposed in [2]. The SODA dataset includes 2,168 (Train 1,168, Test 1,000) IR images captured with a FLIR thermal camera. The dataset consists of labeled images of 20 semantic regions collected from real-

world scenes, as presented in **Table 1**. While most IR data is biased towards either outdoor or indoor environments, SODA [2] includes both outdoor and indoor scenes containing a wide range of environments with different lighting conditions, imaging blur, and varied resolutions. To facilitate the FSS task, we divided the SODA dataset into four folds as shown in **Table 1.** The resulting dataset is named SODA-5^i to distinguish it from the original SODA dataset.

Group	***i* = 0**					***i* = 1**				
Class No	1	2	3	4	5	6	7	8	9	10
Category	Person	Building	Tree	Road	Pole	Grass	Door	Table	Chair	Car
Group	***i* = 2**					***i* = 3**				
Class No	11	12	13	14	15	16	17	18	19	20
Category	Bicycle	Lamp	Monitor	Traffic Lane	Trash Can	Animal	Fence	Sky	River	Side Walk

Table 1: Semantic regions of SODA [2] dataset divided into 4 folds.

After grouping, we define four folds as shown in **Table 2**:

Few-Shot Data	**Train (*Base*)**				**Test (*Novel*)**			
	***i* = 0**	***i* = 1**	***i* = 2**	***i* = 3**	***i* = 0**	***i* = 1**	***i* = 2**	***i* = 3**
Fold 0		✓	✓	✓	✓			
Fold 1	✓		✓	✓		✓		
Fold 2	✓	✓		✓			✓	
Fold 3	✓	✓	✓					✓

Number	**Train**	**Test**
Total	**1,168**	**1,000**
Fold 0	103	927
Fold 1	435	647
Fold 2	765	281
Fold 3	356	676

Table 2: Each fold has three folds for training and one fold for testing [Left]. The number of training data and testing data for each fold [Right].

Each split consists of 3 folds for training and 1 fold for testing. For instance, when we use fold 0, it is assumed that we have seen classes 6 to 20 (i = 1, 2, 3) and have not seen classes 1 to 5 (i = 0). The models are then trained on the 3 folds (i = 1, 2, 3), and evaluated on 1 folds (i = 0). The reduction in the number of data for each split compared to the total train and test dataset is due to the exclusion of images containing unseen categories (*Novel*) in the train set and seen categories (*Base*) in the test set, respectively. For example, the train data in fold 0 shows a significant reduction, indicating that images containing unseen data (i = 0), such as person, building, tree, road, and pole, which account for a high proportion of the entire dataset, are not selected for training.

In the meta learner training stage, an episodic training paradigm is implemented using datasets as above. On the other hand, in the base learner training stage, the base learner is trained using the supervised protocol with only the *Base* categories. This means that every fold has 1,168 train data

and 1,000 validation data during training base learner, where the labels for the *Novel* classes data are set to the background.

3.2.2. CycleGAN Dataset

Some publicly available IR-RGB paired datasets are utilized in our study. We employ the MFNet dataset [36] and the RGB-NIR Scene dataset [37] to train CycleGAN which aims to generate the lightness and RGB images for data-augmentation and conditioning information, respectively.

The MFNet dataset contains a total of 1,569 city view images, consisting of 820 taken at daytime and 749 taken at nighttime. For better contrast, we only use 820 daytime paired images from this dataset. The RGB-NIR Scene dataset comprises 477 images in 9 categories captured in both RGB and Near-Infrared. However, we exclude 159 images from categories such as forest, mountain, and field, as they are not related to the SODA-5^i. We use the remaining 318 images in 6 categories to train the CycleGAN. During training, we combine two paired datasets as the target domain for the CycleGAN models.

Data Augmentation (Generated Lightness Images): The data augmentation process involves generating lightness images. LAB color space consists of three channels (L, A, B), where the first channel, 'L', represents lightness of each pixel in grayscale. The 'A' and 'B' channels indicate the amount of green-red and yellow-blue, respectively, present in each pixel. The LAB color space is commonly used for colorization which aims to predict the output channels 'A' and 'B' based on the input 'L' channel. The lightness of an image contains clear boundaries and grayscale information. Initially, we convert RGB images to LAB images. Then, we train the CycleGAN model to convert the IR domain into the Lightness domain using paired IR and Lightness dataset.

Conditioning Information (Generated RGB Images): For conditioning information, our aim is to generate realistic RGB images. To achieve this, we utilize two types of paired datasets: IR-RGB datasets and lightness-RGB datasets. The model trained with the first dataset generates RGB images to be paired with the original IR images, while the model trained with the second dataset generates RGB images to be paired with augmented lightness images. MFNet and RGB-NIR paired datasets have numerous indoor and outdoor images captured from various perspectives. By taking advantage of color information in these datasets, CycleGAN model is trained to convert IR domain into RGB domain using paired RGB images and IR images.

Before applying trained weight to generate the lightness and RGB images using the trained CycleGAN models, we use gamma correction and histogram equalization [38] to SODA dataset to produce nonlinear and adjust contrast.

3.3 Model Details

3.3.1. Few-Shot Segmentation (MSANet)

In this thesis, we utilize MSANet as a baseline, which is the same as the one used in [3]. The MSANet networks consist of three main parts including the base learner, the meta learner, and an ensemble module. First two learners share the encoder, ResNet-50 [39]. Using features extracted from shared backbone networks, base learner and meta learner are utilized to recognize the base and novel classes, respectively. The ensemble module takes coarse predictions from two learners and an adjustment factor ψ to suppress the falsely predicted regions of base classes, further performing accurate segmentation. The overall architecture of MSANet is illustrated **Figure 3**.

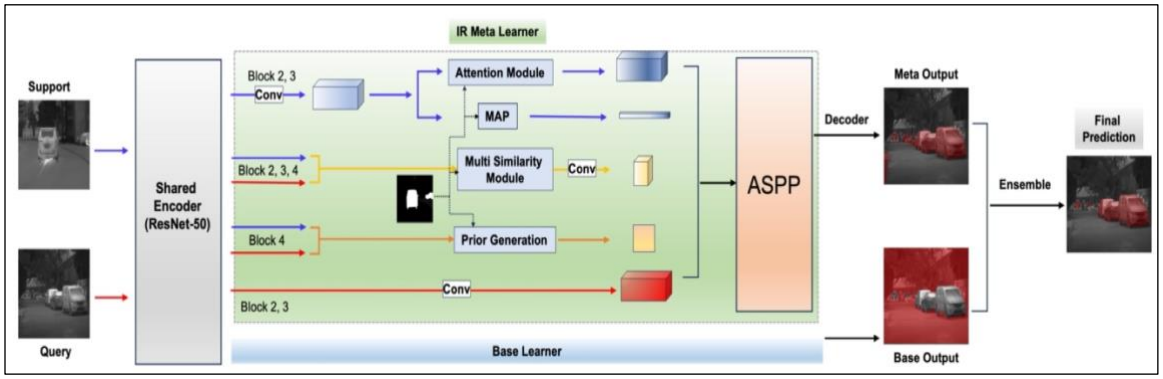

Figure 3: The overall architecture of MSANet with samples of SODA-5^i (figure adapted from [3])

Backbone Networks (Shared Encoder): We use pre-trained ResNet-50 [39] on ImageNet dataset [40] as the backbone network to extract features from both query set and support set. The shared encoder is optimized during the training of the base learner and remains fixed during the training of the meta learner.

Base Learner: PSPNet [9] is an efficient pyramid scene parsing networks for pixel-level prediction by using global pyramid pooling features, as shown in **Figure 4**. This model is served as the base learner. ResNet-50 is used to extract the features of image. A pyramid pooling module takes these feature map and makes the global context information using different 4-level pyramid pooling

operations. The resulting global priors are concatenated with the original features followed by a simple classifier convolution layer and a softmax operation to produce the final prediction map. Final prediction map can be described as:

$$P_b = softmax\left(D_b\left(f_b^q\right)\right) \in \mathbb{R}^{(1+N_b)\times H\times W},$$

where, f_b^q represents the features from the sequential convolutions contains encoder, PPM module, and classifier. D_b is the decoder network that enlarges the spatial scale of f_b^q using interpolation operation. N_b refers to the number of base categories, $N_b = 15$ in our dataset. The cross-entropy loss measures the difference between the prediction $\mathbf{P}_{\mathrm{b}}$ and the ground-truth $\mathbf{m}_b^q$, and loss for base learner can be denoted as:

$$L_b = \frac{1}{N_{batch}}\sum_{i=1}^{N_{batch}} CE\left(\mathbf{P}_{\mathrm{b}(i)}, \mathbf{m}_{b(i)}^q\right),$$

where, N_{batch} is the total number of training samples in each batch.

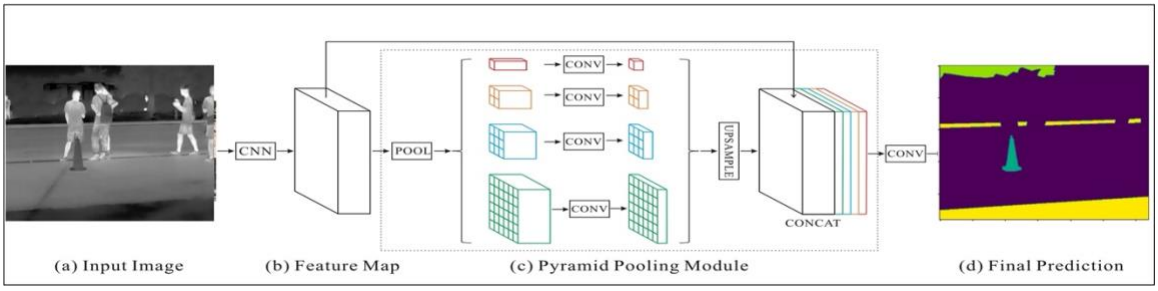

Figure 4: The PSPNet [9] is served as base learner (figure adapted from [9]).

Meta Learner: For a meta learner, as shown in **Figure 3**, training starts following the base learner training. The features of the query and support images are extracted from a pre-trained encoder, whose parameters are fixed during meta learner training. The multi-similarity module generates visual correspondence between the support and query images. In this module, cosine similarity distances are calculated between all intermediate and high-level features of both the query image and the support image, which are extracted from blocks 2, 3, and 4. The proposed attention module utilizes the features extracted from blocks 2 and 3 of the support image, along with their corresponding binary mask, to focus on the targeted class of the query image. The attention module consists of a pooling operation, two convolutional networks and an activation function with the masked support feature. Additionally, the features of the query image and support image generated from block 4 are used to create the prior mask, as proposed in PFENet [23]. All generated features

in the meta learner part are concatenated and proceeded through the atrous spatial pyramid pooling (ASPP) module [41], which includes dilated convolution layers for feature enhancement. The classifier convolution layers, followed by a softmax operation, are used to generate the final binary meta prediction as follows:

$$P_m = softmax\left(D_m\left(f_m^{all}\right)\right) \in \mathbb{R}^{(1+1)\times H\times W},$$

Here, f_m^{all} means the concatenation of all features generated by the proposed modules, including the multi-layer similarity module, attention features, prior mask, prototype vector, and the concatenated query features extracted from blocks 2 and 3 of backbone network. D_m indicates to the decoder networks consisting of the ASPP, convolution block and classifier.

The binary cross entropy (BCE) loss between the prediction mask P_m of the query image and its corresponding ground truth mask $\mathbf{m}^q$ is calculated as follows:

$$L_{meta} = \frac{1}{N_{episode}} \sum_{i=1}^{N_{episode}} BCE\left(\mathbf{P}_{\mathrm{m}(i)}, \mathbf{m}_i^q\right),$$

where, $N_{episode}$ denotes the total number of training episodes in each batch.

Ensemble module: MSANet [3] utilizes an ensemble module, inspired by BAM [27], to estimate the scene differences between query-support image pairs, as shown in **Figure 5**. During the training of the meta learner, the ensemble module calculates the Gram matrices of the support and query images using the low-level features extracted from the shared encoder. The Frobenius norm is then evaluated on the difference between these Gram matrices to derive the adjustment factor ψ, which guides the adjustment process. The equations are follows:

$$\mathbf{A_s} = \mathcal{F}_{\mathrm{reshape}}(\mathrm{f}_{\mathrm{low}}^{s}) \in \mathbb{R}^{C_1\times N},$$

$$\mathbf{A_q} = \mathcal{F}_{\mathrm{reshape}}(\mathrm{f}_{\mathrm{low}}^{q}) \in \mathbb{R}^{C_1\times N}$$

$$\mathbf{G^s} = \mathbf{A_s}\,\mathbf{A_s^T} \in \mathbb{R}^{C_1\times C_1},$$

$$\mathbf{G^q} = \mathbf{A_q}\,\mathbf{A_q^T} \in \mathbb{R}^{C_1\times C_1},$$

$$\psi = \|\mathbf{G^s} - \mathbf{G^q}\|_F,$$

where $\mathrm{f}_{\mathrm{low}}^{s}$, $\mathrm{f}_{\mathrm{low}}^{q} \in \mathbb{R}^{C_1 \times H_1 \times W_1}$ denotes the low-level features extracted from support and query image. N means $H_1 \times W_1$ and $\mathcal{F}_{\mathrm{reshape}}$ reshapes the size of the input to $C_1 \times N$. $\|\cdot\|_F$ refers to the Frobenius norm.

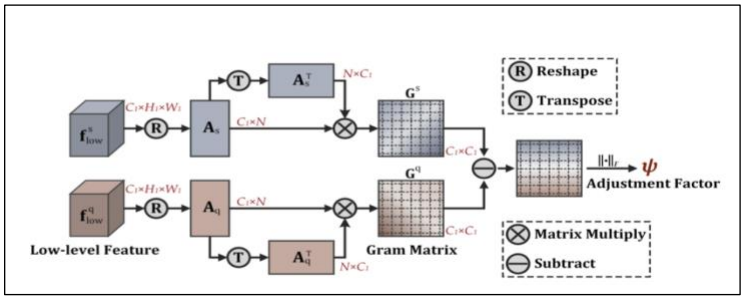

Figure 5: The calculation process of the adjustment factor $\boldsymbol{\psi}$ (figure adapted from **[27]**).

As illustrated in **Figure 6**, the coarse predictions of the two learners are integrated under the guidance of adjustment factor ψ. This adjustment factor ψ is applied to both the background probability map and the foreground probability map of the meta learner's prediction through a 1×1 convolution operation. Subsequently, the background probability map, fused with the adjustment factor ψ, is combined with the prediction of the base learner using another 1×1 convolution operation. This process aims to connect the two learners.

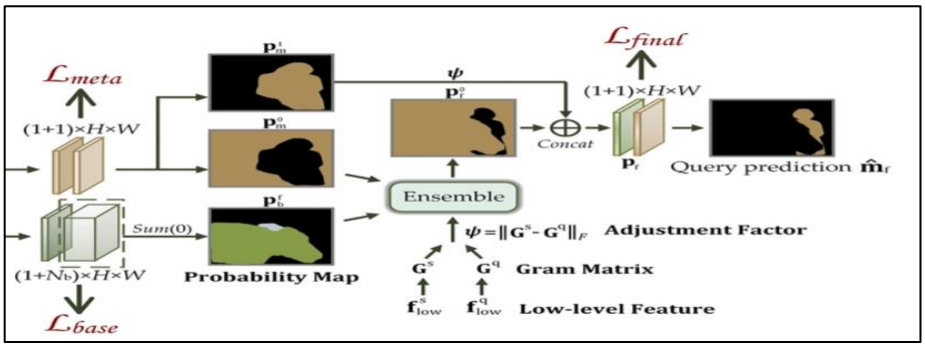

Figure 6: The process of the ensemble module to obtain the final prediction using both the meta and base predictions (figure adapted from **[27]**).

The final prediction map is obtained by concatenating the foreground probability map fused with adjustment factor ψ, and ensembled background probability map. The overall loss is defined as follows:

$$L_{final} = \frac{1}{N_{episode}} \sum_{i=1}^{N_{episode}} BCE\left(\mathbf{p}_i^q, \mathbf{m}_i^q\right),$$

$$L_{full} = L_{final} + L_{meta} + L_{base}$$

3.3.2. CycleGAN

We implemented the CycleGAN models as described in [4]. As mentioned in Section 2.2, CycleGAN consists of two generator models and two discriminator models.

Two generators are based on the approach described for style transfer [42]. Each generator has three components: an encoder, a transformer, and a decoder as shown in **Figure 7**. In the encoder part, convolution layers are used to the extraction of progressively higher-level features from input data. The transformation part, which contains residual networks, transforms the input features from one domain to the target domain. This includes using 9 residual blocks instead of 6 for training on higher-resolution images. In the decoder part, the low-level features are reconstructed by applying transpose convolution layers followed by a 7×7 convolution layer.

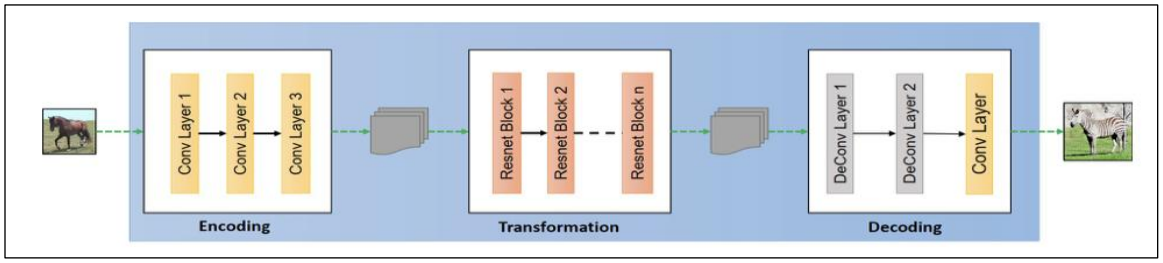

Figure 7: The generator consists of an encoder, a transformer, and a decoder (figure adapted from [45]).

In CycleGAN, the two discriminators are implemented as PatchGAN discriminators, following the approach in [43]. Instead of classifying the entire image as real or fake, the PatchGAN discriminator focuses on classifying each N×N patch in the image. In our work, the patch size is set to 70×70, matching that of [43]. The discriminator network is depicted in **Figure 8**. Both discriminators are composed of four convolutional layers with a kernel size of 4×4. Following these convolutional layers, a final convolutional layer is applied to produce a 1-dimensional output, indicating the classification of each patch as real or fake.

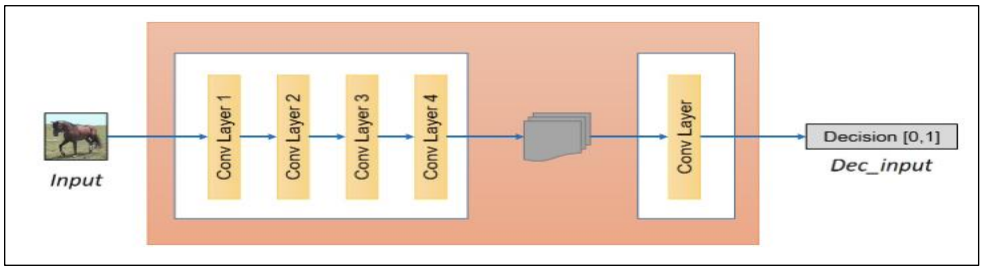

Figure 8: The discriminator is composed of convolution layers based on PatchGAN (figure adapted from [45]).

Loss functions: CycleGAN has two adversarial losses and a cycle consistency loss. Firstly, two adversarial losses are defined as follows:

$$\mathcal{L}_{GAN}(G_{A\to B}, D_B, A, B) = \mathbb{E}_{b\sim P_{data}(b)}[logD_B(b)] + \mathbb{E}_{a\sim P_{data}(a)}[\log\,(1 - D_B(G_{A\to B}(A))],$$

$$\mathcal{L}_{GAN}(G_{B\to A}, D_A, B, A) = \mathbb{E}_{a\sim P_{data}(a)}[logD_A(a)] + \mathbb{E}_{b\sim P_{data}(b)}[\log\,(1 - D_A(G_{B\to A}(B))]$$

where G represents the mapping function, D represents its discriminator, and A and B indicate real samples from different domains. $G_{A\to B}$ and $G_{B\to A}$ aim to generate images that look like images from domain B and A, respectively. On the other hand, D_B and D_A strive to distinguish between generated samples and real samples. Secondly, the cycle consistency loss is defined as follows:

$$\mathcal{L}_{Cycle}(G_{A\to B}, G_{B\to A}) = \mathbb{E}_{a\sim P_{data}(a)}\left[\left\|G_{B\to A}\left(G_{A\to B}(a)\right) - a\right\|_1\right] + \mathbb{E}_{a\sim P_{data}(b)}\left[\left\|G_{A\to B}\left(G_{B\to A}(b)\right) - b\right\|_1\right]$$

This loss ensures that an input image from domain A or B can be translated back to its original image using the image translation cycle. For example, for input a from domain A, the image translation cycle should be able to bring a back to the original image, i.e., $a \to G_{A\to B}(a) \to G_{B\to A}\left(G_{A\to B}(a)\right) \approx a$. This is achieved by comparing the original image to the generated cycle image using the ℓ_1 norm. By combining all these losses, a full objective can be described as:

$$\boldsymbol{\mathcal{L}}(G_{A\to B}, G_{B\to A}, D_A, D_B) = \mathcal{L}_{GAN}(G_{A\to B}, D_B, A, B) + \mathcal{L}_{GAN}(G_{B\to A}, D_A, B, A) + \mathcal{L}_{Cycle}(G_{A\to B}, G_{B\to A})$$

3.3.3. Gated Feature-wise Transform

Li et al. [2] proposed the gated feature-wise transform (GFT) layer, which consists of three components of feature modulation, information control, and feature-wise transform, as shown in **Figure 9**. The GFT layer takes two feature maps, one from the input image and the other from the conditioning image. Input features are extracted from the middle of the encoder by passing the input image through the encoder networks. On the other hand, the conditioning image features are created by applying convolutional layers to the conditioning image. The feature modulation involves two 1×1 convolution layers to ensure that the conditioning input features have adaptively the same size as the input features. To handle the noise in prior information, an information control module was introduced. Li et al. [2] suggested a gate scheme by adding a sigmoid function before applying the features from feature modulation to the feature-wise transform module. In the feature-wise transform module, spatial feature-wise transform (SFT) is employed to embed the conditioning feature and guide infrared image semantic segmentation. Input features are scaled and

shifted using element-wise product and sum operations, respectively, with the conditioning features generated by information control modules. GFT blocks, called the conditioning-CNN block, consist of three GFT layers and convolutional layers (two 3×3 convolution layers and one 1×1 convolution layer) with a residual layer following specific input features extracted from the encoder networks, as shown in **Figure 9**.

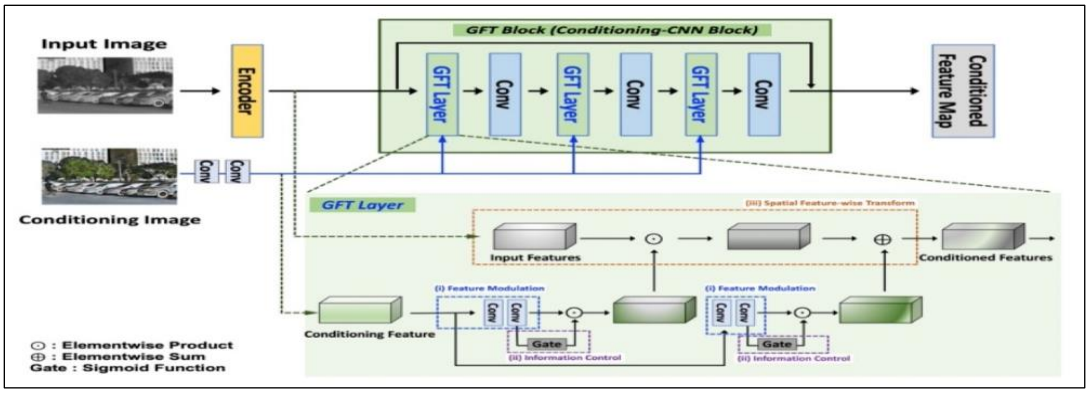

Figure 9: GFT layer consist of Feature Modulation, Information Control, and Spatial Feature-wise Transform. GFT Block consists of three GFT layers and convolution layers following each GFT layer (figure adapted from [2]).

As mentioned in Section 2.3, incorporating conditioning features into the low-level input features has shown to improve the overall embedding. The conditioned features, influenced by prior information, have an impact on the subsequent layers of the encoder and decoder. To evaluate the effect of a GFT block following a low-level layer in a supervised learning scenario, we conducted tests by adding a GFT block following 'Layer 0' of the ResNet-50 encoder, as shown in **Figure 10**.

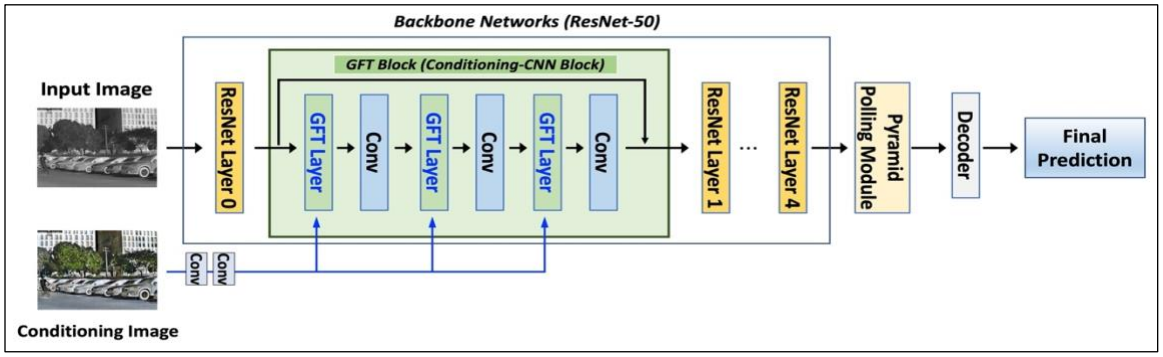

Figure 10: Applying GFT block into Base Learner (PSPNet)

However, unlike the supervised-learning segmentation, it is important to note that adding a GFT block following a low-level layer of the shared encoder may have a negative impact on the meta learner in the BAM architecture. This is because it introduces changes to the intermediate and high-

level layers of the encoder, which are crucial for the meta learner's training. These layers play a significant role in enabling the meta learner to find the relationship between the support features and query features.

To efficiently utilize the GFT block, we propose a modification to the meta learner part, as illustrated in **Figure 11**. Our approach involves using two encoders and two meta learners. The first encoder, called Encoder1, is trained for 100 epochs using both the generated RGB images and augmented IR images. This training ensures that adequate features are obtained from both types of images simultaneously. It is worth noting that if we were to extract conditioning data features using simple convolutional layers, as shown in **Figure 9**, these features would not provide sufficient information compared to the intermediate or high-level features extracted from IR images. In addition, if we use a shared encoder trained with IR images, it would not capture the features of the generated RGB images properly. Therefore, we utilize a separate encoder specifically for the meta learners. The IR features and the generated RGB features extracted from Encoder1 are then fed into their respective meta learners. Finally, the feature maps from each meta learner are fused using GFT blocks. The second encoder, Encoder2, is trained using IR images, as same as the baseline approach.

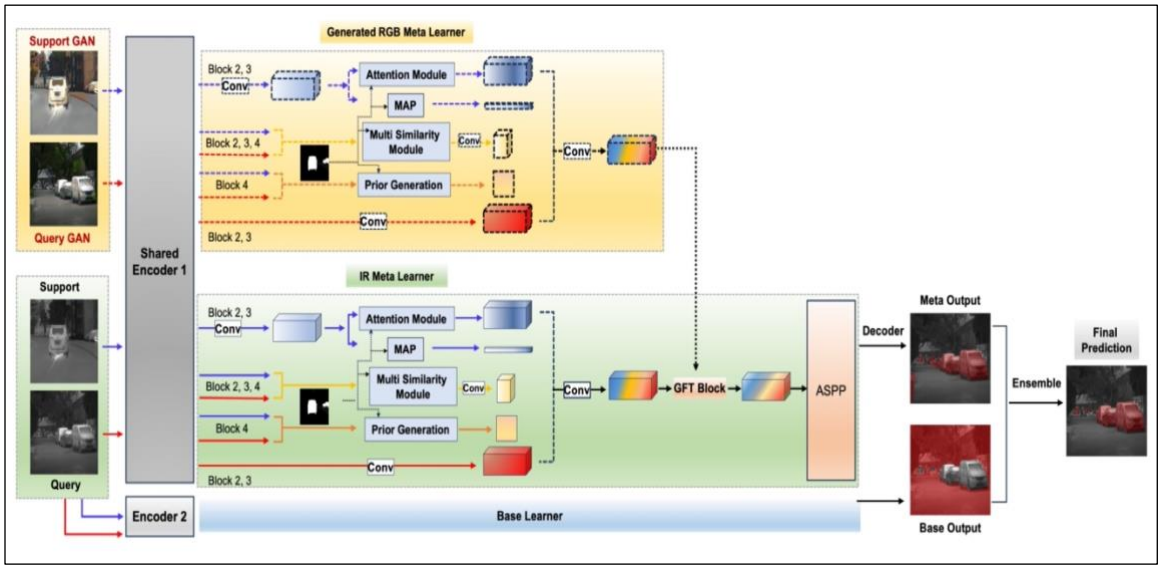

Figure 11: Proposed GFT method in MSANet

3.4 Experiments

3.4.1. Implementation Details

In our implementation, we followed the basic training settings for both MSANet and CycleGAN models, with a few modifications. The implementation codes for MSANet and CycleGAN can be found publicly in references [44, 45, 46].

CycleGAN: We trained the models from scratch. The input images were paired and had a size of 640×480. We scaled the images to 286×286 and then cropped them to 256×256 to generate lightness and RGB images. We set a learning rate of 0.0002 and kept this rate for the first 100 epochs. Afterwards we linearly decayed the learning rate to zero over the next 100 epochs. The Adam optimizer was used, and we set the batch size to 32.

MSANet: The input images were paired and had a size of 640×480. We cropped them to 473×473. The training of MSANet was divided into two stages. In the first stage, the base learner was optimized on each fold of the SODA-5^i dataset for 300 epochs using supervised training. We utilized a pre-trained PSPNet [9] on the Image-net dataset [40] as the backbone network. The batch size for this stage was set to 12. In the second stage, the meta learner was trained using the episodic training paradigm with the features extracted from the trained backbone networks. The batch size for this stage was set to 8. For evaluation, we averaged the results from 5 testing runs with different random seeds. For both the base learner and meta learner, we employed the SGD optimizer with a learning rate of 0.005 and 0.0075, respectively. We stopped training if the validation set results did not improve for 75 epochs. Various augmentation techniques such as random scaling, rotating, Gaussian blur, cropping, and normalization from -1 to 1 were utilized during training.

3.4.2. Metrics

In our evaluation, following the previous FSS works, we adopt mean intersection-over-union (mIoU) and foreground-background IoU (FB-IoU) as the evaluation metrics. mIoU and FB-IoU are defined as:

$$mIoU = \frac{1}{N_c} \sum_{c=1}^{N_c} IoU_c \ ,$$

$$FB\ IoU = \frac{1}{2}(IoU_f + IoU_b)$$

where N_c represent the number of classes in the targeted fold (e.g, N_c = 15 for Base Learner, 5 for Meta Learner) and IoU_c means the intersection-over-union of class c. IoU_f and IoU_b refer to foreground and background intersection over union values in each fold, respectively.

3.5 Results

First, we demonstrate the qualitative results of the generated images, including the lightness images and RGB images, on the SODA-5^i dataset. These images demonstrate the effectiveness of our approach in generating realistic and visually appealing results. Next, we present the results of testing using three different methods: the baseline method, the baseline method combined with augmentation-based learning, and the baseline method combined with augmentation-based learning and conditioned networks. For each method, we report the mIoU and FB-IoU results. We conduct a detailed analysis of the results, examining the performance of the base learner, meta learner, and final prediction separately. This allows us to gain insights into the individual contributions of each component and understand the overall performance of our approach.

3.5.1. Qualitative results of the generated images by CycleGAN

Generated Lightness Image: The bottom row of **Figure 12** presents the examples of the generated lightness domain images. Note that the generated lightness images exhibit enhanced contrast compared to the original SODA-5^i. These retain the essential properties and characteristics of the original dataset. The generated lightness images serve as valuable data augmentation resources, providing additional diversity and variations to the training data.

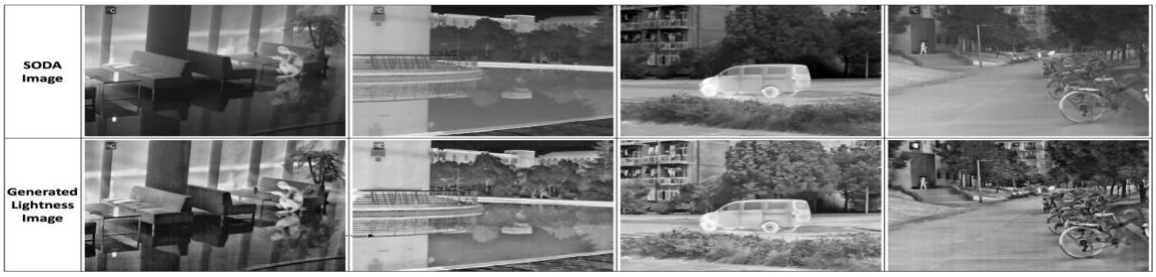

Figure 12: The first row shows samples of the original SODA-$\mathbf{5}^{\boldsymbol{i}}$. The second row shows their corresponding generated lightness images translated by CycleGAN.

Generated RGB Image: The samples of generated RGB image are shown in **Figure 13**. It can be observed that the gray channel is converted to RGB channels. The generated RGB images are enriched in the aspect of channel information. Some categories, such as tree and car, are translated clearly as shown in the 1st and 2nd columns. However, in some images, such as the 4th column, the color distinction is not clear. Despite the presence of ambiguous colors in the generated RGB images, they still exhibit distinct object contours and useful channel information. The conditioning networks (GFT blocks) are expected to handle and alleviate these problems.

Figure 13: The first row shows samples of the original SODA-5^i. The second row shows their corresponding generated RGB images translated by CycleGAN.

3.5.2. Quantitative Evaluation on FSS with proposed methods

Base Learner: To evaluate the performance of the base learner trained in a supervised learning manner, we compare the results of each method using mIoU. **Table 3** presents the mIoU results of different methods on SODA-5^i. The results show that adding an augmentation-based learning method yields better performance compared to the base learner of the baseline. The MSANet with the proposed method, incorporating both the augmentation-based method and conditioning networks, demonstrates improvements of 1.28% to 3.35% over the results of the baseline. It is important to note that during the training of the meta learner, we utilize the base learners from the 1st and 2nd columns in **Table 3** as encoders, instead of the base learner from the 3rd column, as discussed in Section 3.3.3.

Fold	**Base Learner**		
	Baseline	**Baseline + Augmentation**	**Baseline + Augmentation + GFT**
Fold-0	61.80	64.85	**65.15**
Fold-1	69.13	**70.75**	70.41
Fold-2	63.81	64.80	**66.26**
Fold-3	62.31	63.81	**64.99**

Table 3: Performance of base learner on SODA-5^i in terms of mIoU

Meta Learner and Final results: Table 4 illustrates the performances of the different approaches in terms of mIoU over the SODA-5^i dataset. The results show that the addition of the augmentation method with the generated lightness images substantially improves the mIoU by 3.6% and 3.5% compared to the baseline under 1-shot and 5-shot settings, respectively. Furthermore, incorporating conditioning networks (GFT layers) into the meta learner yields the highest improvements in aspect of the mean over 4 folds.

mIoU		**mIoU Final Results**					
		1-shot			**5-shot**		
Fold		**Baseline**	**Baseline + Augmentation**	**Baseline + Augmentation + GFT (Meta)**	**Baseline**	**Baseline + Augmentation**	**Baseline + Augmentation + GFT (Meta)**
Fold 0	**Final**	41.04	41.34	**42.85**	44.27	49.85	**50.69**
	Meta	33.69	35.07	36.34	34.56	43.96	44.33
	Base	55.27	60.13	60.13	55.27	60.13	60.13
Fold 1	**Final**	35.17	39.17	38.43	39.26	41.60	41.21
	Meta	24.67	28.01	28.19	32.93	33.75	34.96
	Base	59.48	61.34	61.34	59.48	61.34	61.34
Fold 2	**Final**	48.32	51.61	**51.79**	59.63	61.25	**62.17**
	Meta	42.71	42.95	42.18	57.58	57.13	59.52
	Base	52.42	52.68	52.68	52.42	52.68	52.68
Fold 3	**Final**	51.86	58.99	**61.03**	58.80	63.59	**65.38**
	Meta	35.58	38.18	40.24	43.50	48.89	46.37
	Base	49.56	51.77	51.77	49.56	51.77	51.77
Mean	**Final**	44.10	47.78	**48.53**	50.49	54.07	**54.86**

Table 4: Performance of meta learner and final ensembled model on SODA-$\mathbf{5^i}$ in terms of mIoU.

Table 5 depicts the FB-IoU for different methods over over the SODA-5^i dataset. Consistent with the previous findings, the use of the augmentation shows the best results under the 1-shot setting, while the GFT method achieves the highest score under the 5-shot setting, as in shown **Table 5**.

FB-IoU		**FB-IoU Final Results**					
		1-shot			**5-shot**		
Fold		**Baseline**	**Baseline + Augmentation**	**Baseline + Augmentation + GFT**	**Baseline**	**Baseline + Augmentation**	**Baseline + Augmentation + GFT**
Fold 0	**Final**	65.45	65.87	**66.41**	68.98	73.93	**73.99**
	Meta	56.31	59.06	58.78	53.96	68.45	67.65
Fold 1	**Final**	65.85	69.48	68.14	69.46	70.62	**70.77**
	Meta	51.74	54.50	54.08	60.27	58.65	62.56
Fold 2	**Final**	73.35	75.16	74.54	76.60	77.07	**78.51**
	Meta	66.41	66.99	66.10	73.64	73.34	75.64
Fold 3	**Final**	71.06	74.09	**74.56**	76.41	76.76	**77.77**
	Meta	46.14	50.92	49.56	55.86	58.64	55.60
Mean	**Final**	68.93	**71.15**	70.91	72.86	74.60	**75.26**

Table 5: Performance of meta learner and final ensembled model on SODA-$\mathbf{5^i}$ in terms of FB-IoU.

3.5.3. Qualitative Evaluation of the proposed methods for FSS

As shown in **Figure 14** to **Figure 16**, our model predicts three parts prediction: meta, base, and final predictions, conditioned on support image with a blue-labeled mask. The meta learner aims to predict the target category (Novel class) of the query image, which is represented by a green-labeled mask. On the other hand, the base learner focuses on predicting the base categories. By combining both predictions through the ensemble module, final segmentation results are obtained. **Figure 14** to **Figure 16** illustrate the predictions of the baseline method (yellow box) and the proposed method (blue box). The proposed method achieves more visibly precise predictions for the target class compared to the baseline method. In cases where the meta learner falsely predicts the target, the base learner effectively suppresses these erroneous predictions, leading to improved segmentation results.

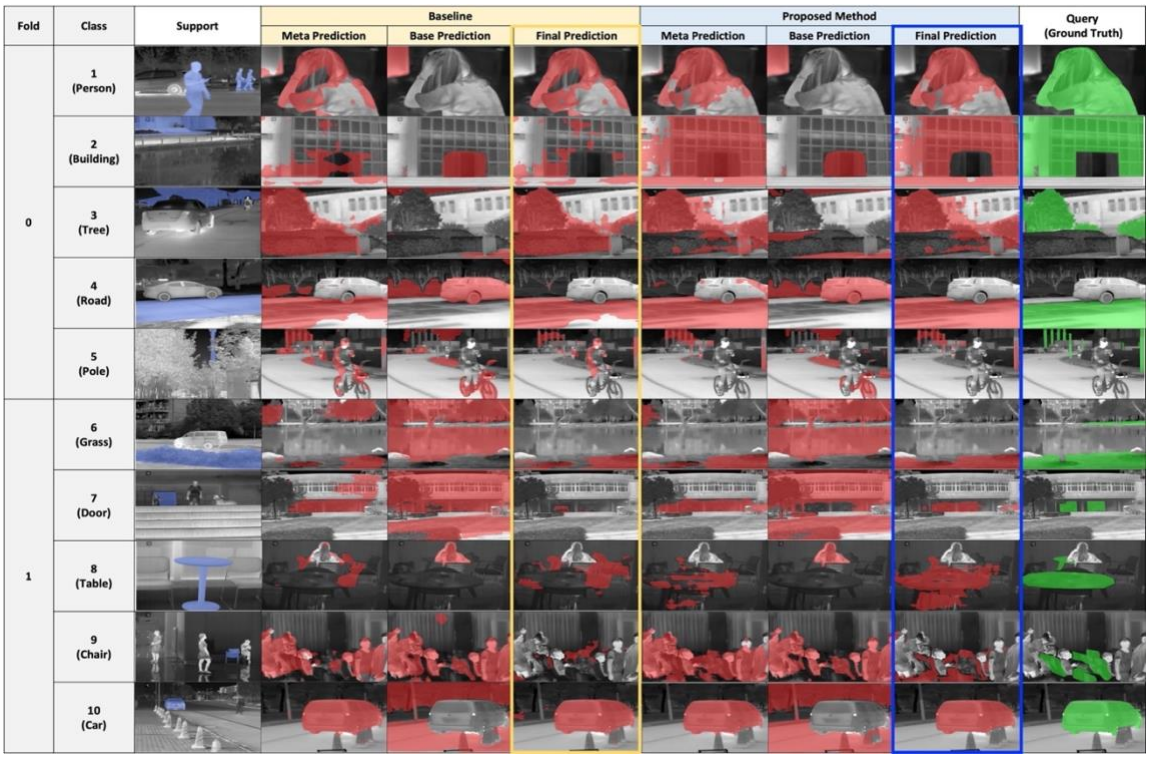

Figure 14: Implementation results (Fold 0, 1) of baseline and proposed methods under 1-shot setting.

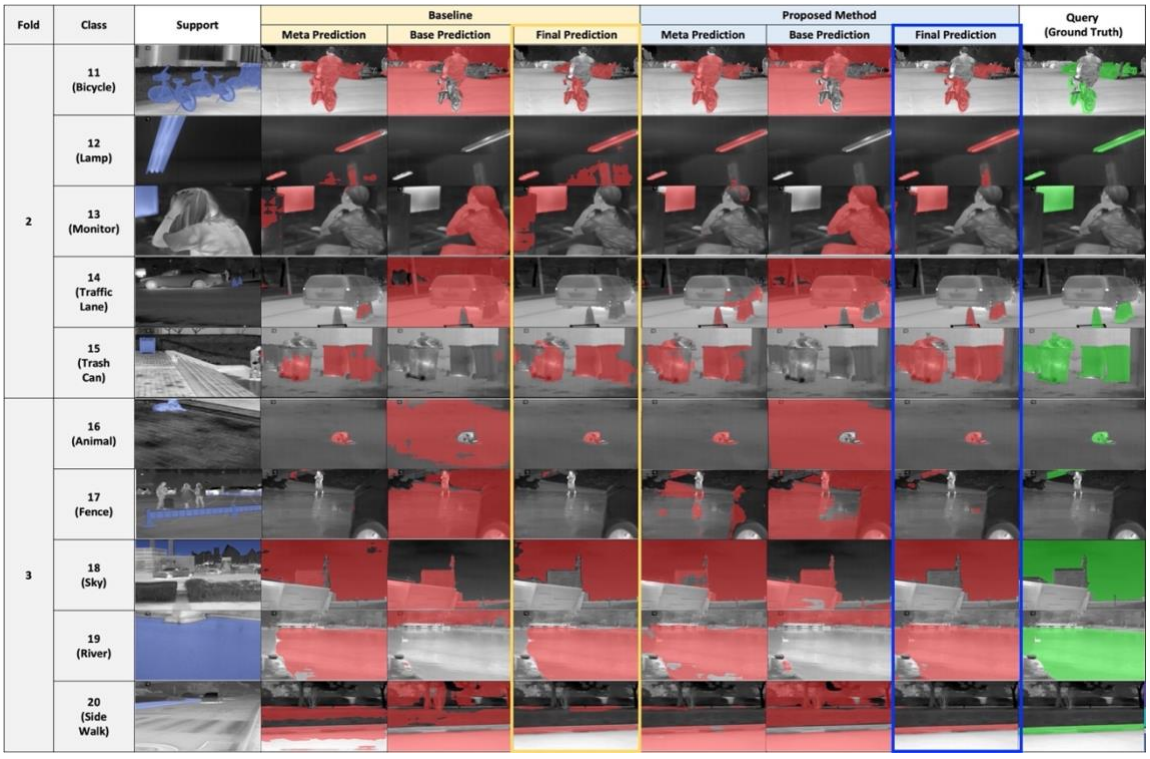

Figure 15: Implementation results (Fold 2, 3) of baseline and proposed methods under 1-shot setting.

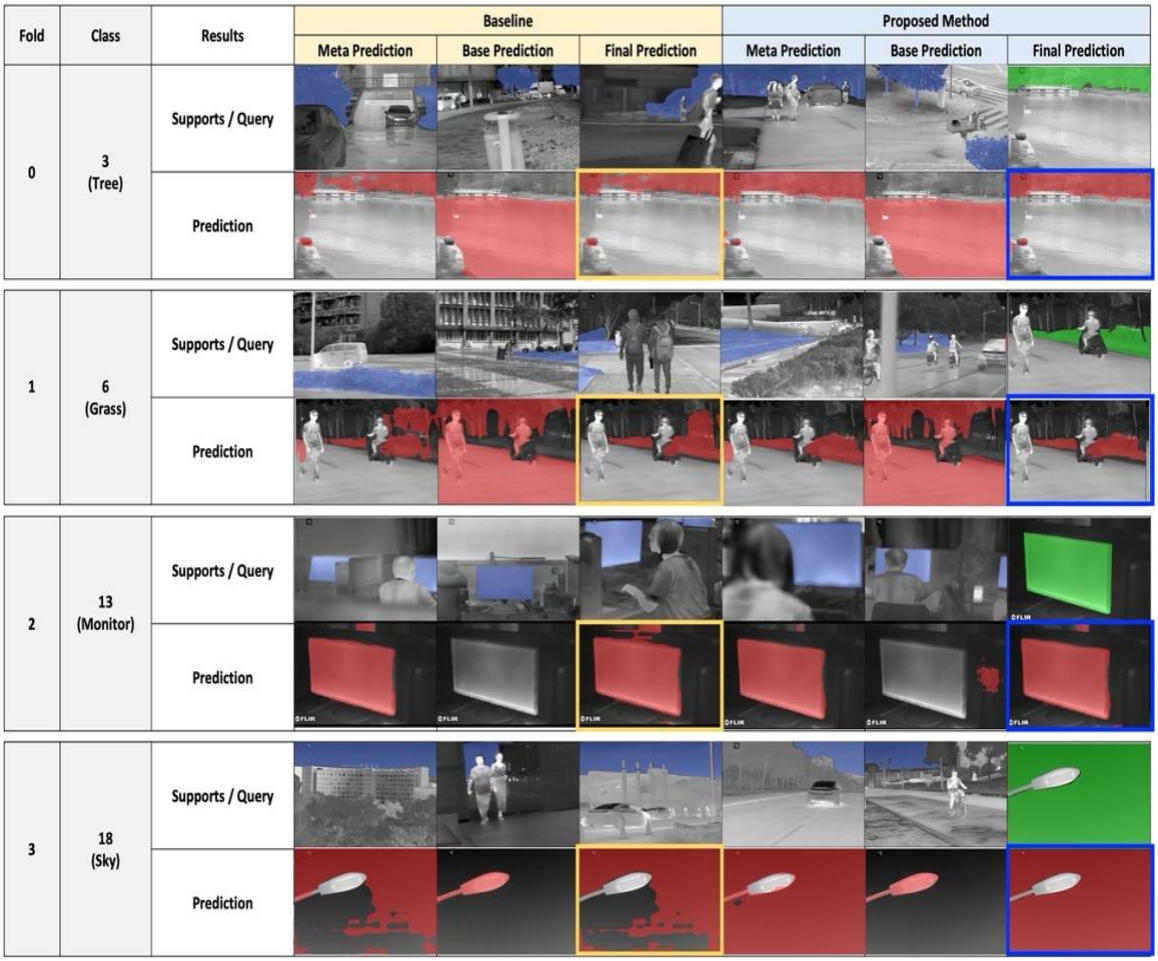

Figure 16: Implementation results, ones of each fold, of baseline and proposed methods under 5-shot setting

Chapter 4

Discussion

4.1 Summary

In this thesis, we consider the few shot segmentation of thermal IR images, as thermal IR imaging has extensive applications in various fields. The FSS task addresses the challenges associated with the lack of large databases of IR data for supervised training, as well as the emergence of new classification classes in various applications. Based on our proposed improvements and implementation results, the MSANet FSS model can be effectively applied to segment IR images in a FSS scenario.

To enhance the MSANet model, we leverage generative deep learning methods utilizing CycleGAN in two ways to generate the available data: augmenting the raw training data and enriching the channel information. Additionally, we exploit gated feature-wise transform blocks which help the model segment IR images more effectively by using conditioning information. The proposed methods show improvement in segmenting IR images. By utilizing these methods, we are able to overcome the challenges posed by limited data and leverage the benefits of generative networks to enhance the FSS task for IR images.

4.2 Limitations

IR image data for a few-shot segmentation: The use of the SODA dataset as the source of IR data introduces certain limitations. When splitting the data into four folds for the few-shot segmentation task, each fold contains both base and novel categories. This leads to a drastic and irregular reduction in the number of training and testing samples, as illustrated in **Table 2**. The presence of multiple categories within a single image, such as people, trees, poles, skies, buildings, cars, bicycles, and sidewalks in a city scene, often results in the exclusion of such images in many

fold cases. Like PASCAL-5^i [10] and COCO-20^i [35], we need to have enough IR data for FSS after dividing the data.

Conversion IR domain to RGB domain: We can make SODA IR data colorized by using CycleGAN on other publicly available paired IR-RGB databases. While the generated RGB images contain richer channel information compared to IR images, they still exhibit noise and sometimes overly emphasize certain colors. Exploring better colorization methods for IR images may yield improved results.

Training additional generative networks: Although our proposed generative deep learning methods improve segmentation results, they require the training of additional models. This inevitably entails a longer training time and the need to find suitable paired or unpaired data for training the generative networks, adding extra complexity and effort to the overall process. Finally, combination of FSS and generative networks in an end-to-end training approach may further alleviate the time-consuming aspects and potentially lead to further improvements in performance.

4.3 Future Work

We have found that augmenting train data using CycleGAN leads to noticeable enhancements, while the utilization of the conditioning block with the generated RGB images further improves performance. By leveraging the features extracted from the fixed encoders, which is a characteristic of the BAM architecture, we have demonstrated that incorporating the conditioning block in the meta learner stage, in terms of late fusion, substantially improves the accuracy of segmentation. However, it is worth exploring the incorporation of the conditioning block (GFT block) following the low-level layer of the backbone networks and parameterizing the backbone networks during meta learner training, similar to other end-to-end few-shot learning models. This modification may potentially enable early fusion and yield even better results. Hence, it would be valuable to investigate an alternative few-shot segmentation model incorporating conditioning networks for the early fusion.

References

[1] Kevser Irem Danaci, Erdem Akagunduz, "A Survey on Infrared Image and Video Sets," *arXiv preprint arXiv:2203.08581,* 2022..

[2] Li, Chenglong and Xia, Wei and Yan, Yan and Luo, Bin and Tang, Jin, "Segmenting Objects in Day and Night:Edge-Conditioned CNN for Thermal Image Semantic Segmentation," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 32, no. IEEE, pp. 3069--3082, 2019.

[3] Iqbal, Ehtesham and Safarov, Sirojbek and Bang, Seongdeok Iqbal, Sirojbek Safarov, Seongdeok Bang, "MSANet: Multi-Similarity and Attention Guidance for Boosting Few-Shot Segmentation," *arXiv preprint arXiv:2206.09667,* 2022.

[4] Zhu, Jun-Yan and Park, Taesung and Isola, Phillip and Efros, Alexei A, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2223–2232, 2017.

[5] Antoniou, Antreas and Storkey, Amos and Edwards, Harrison, "Data Augmentation Generative Adversarial Networks," *arXiv preprint arXiv:1711.04340,* 2017.

[6] Wang, Yu-Xiong and Girshick, Ross and Hebert, Martial and Hariharan, Bharath, "Low-Shot Learning from Imaginary Data," 2018, pp. 7278--7286.

[7] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla, "SegNet. A deep convolutional encoder-decoder architecture for image segmentation," *arXiv preprint arXiv:1511.00561,* 2015.

[8] Jonathan Long, Evan Shelhamer, Trevor Darrell, "Fully convolutional networks for semantic segmentation," *arXiv preprint arXiv:1411.4038*, 2015.

[9] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia., "Pyramid scene parsing network," *arXiv preprint arXiv:1612.01105*, 2017.

[10] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots, "One-shot learning for semantic segmentation," *arXiv preprint arXiv:1709.03410*, 2017.

[11] Kate Rakelly, Evan Shelhamer, Trevor Darrell, Alexei A Efros, and Sergey Levine, "Few-shot segmentation propagation with guided networks," *arXiv preprint arXiv:1806.07373*, 2018.

[12] Dong, Nanqing, and Eric P. Xing, "Few-shot semantic segmentation with prototype learning," *BMVC*, vol. Vol. 3. No. 4., 2018.

[13] Zhang, Xiaolin and Wei, Yunchao and Yang, Yi and Huang, Thomas S, "Sg-one: Similarity guidance network for one-shot semantic segmentation," *IEEE transactions on cybernetics*, vol. 50, pp. 3855--3865, 2020.

[14] Wang, Kaixin and Liew, Jun Hao and Zou, Yingtian and Zhou, Daquan and Feng, Jiashi, "Panet: Few-shot image semantic segmentation with prototype alignment," *proceedings of the IEEE/CVF international conference on computer vision*, pp. 9197--9206, 2019.

[15] Chen, Wei-Yu and Liu, Yen-Cheng and Kira, Zsolt and Wang, Yu-Chiang Frank and Huang, Jia-Bin, "A closer look at few-shot classification," *arXiv preprint arXiv:1904.04232*, 2019.

[16] Finn, Chelsea and Abbeel, Pieter and Levine, Sergey, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International conference on machine learning*, PMLR, 2017, pp. 1126--1135.

[17] Antoniou, Antreas and Edwards, Harrison and Storkey, Amos, "How to train your MAML," *arXiv preprint arXiv:1810.09502*, 2018.

[18] Li, Zhenguo and Zhou, Fengwei and Chen, Fei and Li, Hang, "Meta-sgd: Learning to learn quickly for few-shot learning," *arXiv preprint arXiv:1707.09835*, 2017.

[19] Koch, Gregory and Zemel, Richard and Salakhutdinov, Ruslan and others, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, Lille, 2015.

[20] Sung, Flood and Yang, Yongxin and Zhang, Li and Xiang, Tao and Torr, Philip HS and Hospedales, Timothy M, "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.

[21] Snell, Jake and Swersky, Kevin and Zemel, Richard, "Prototypical networks for few-shot learning," *Advances in neural information processing systems*, vol. 30, 2017.

[22] Vinyals, Oriol and Blundell, Charles and Lillicrap, Timothy and Wierstra, Daan and others, "Matching networks for one shot learning," *Advances in neural information processing systems*, vol. 29, p. 2016.

[23] Tian, Zhuotao and Zhao, Hengshuang and Shu, Michelle and Yang, Zhicheng and Li, Ruiyu and Jia, Jiaya, "Prior guided feature enrichment network for few-shot segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 4, no. IEEE, pp. 1050--1065, 2020.

[24] Min, Juhong and Kang, Dahyun and Cho, Minsu, "Hypercorrelation squeeze for few-shot segmentation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 6941--6952.

[25] Hu, Tao and Yang, Pengwan and Zhang, Chiliang and Yu, Gang and Mu, Yadong and Snoek, Cees GM, "Attention-based multi-context guiding for few-shot semantic segmentation," in *Proceedings of the AAAI conference on artificial intelligence*, 2019, p. 84418448.

[26] Li, Gen and Jampani, Varun and Sevilla-Lara, Laura and Sun, Deqing and Kim, Jonghyun and Kim, Joongkyu, "Adaptive prototype learning and allocation for few-shot segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 8334--8343.

[27] Lang, Chunbo and Cheng, Gong and Tu, Binfei and Han, Junwei, "Learning what not to segment: A new perspective on few-shot segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 8057--8067.

[28] Everingham, Mark and Van Gool, Luc and Williams, Christopher KI and Winn, John and Zisserman, Andrew, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. Springer, pp. 303--338, 2010.

[29] Goodfellow, Ian and Pouget-Abadie, Jean and Mirza, Mehdi and Xu, Bing and Warde-Farley, David and Ozair, Sherjil and Courville, Aaron and Bengio, Yoshua, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. ACM New York, NY, USA, pp. 139-144, 2020.

[30] Kütük, Zülfiye, and Görkem, Algan., "Semantic Segmentation for Thermal Images: A Comparative Survey," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 286-295.

[31] Sun, Yuxiang and Zuo, Weixun and Liu, Ming, "Rtfnet: Rgb-thermal fusion network for semantic segmentation of urban scenes," *IEEE Robotics and Automation Letters*, vol. 4, no. IEEE, pp. 2576-2583, 2019.

[32] Dumoulin, Vincent and Perez, Ethan and Schucher, Nathan and Strub, Florian and Vries, Harm de and Courville, Aaron and Bengio, Yoshua, "Feature-wise transformations," Distill, 2018. [Online]. Available: https://distill.pub/2018/feature-wise-transformations.

[33] Perez, Ethan and Strub, Florian and De Vries, Harm and Dumoulin, Vincent and Courville, Aaron, "Film: Visual reasoning with a general conditioning layer," in *Proceedings of the AAAI Conference on Artificial Intelligence*, p. 2018.

[34] Wang, Xintao and Yu, Ke and Dong, Chao and Loy, Chen Change, "Recovering realistic texture in image super-resolution by deep spatial feature transform," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 606--615.

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

[36] Ha, Qishen and Watanabe, Kohei and Karasawa, Takumi and Ushiku, Yoshitaka and Harada, Tatsuya, "MFNet: Towards real-time semantic segmentation for autonomous vehicles with multi-spectral scenes," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2017, pp. 5108-5115.

[37] Brown, Matthew and Süsstrunk, Sabine, "Multi-spectral SIFT for scene category recognition," in *CVPR 2011*, IEEE, 2011, pp. 177--184.

[38] K. Zuiderveld, "Contrast limited adaptive histogram equalization," *Graphics gems*, no. Academic Press, pp. 474-485, 1994.

[39] He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.

[40] Deng, Jia and Dong, Wei and Socher, Richard and Li, Li-Jia and Li, Kai and Fei-Fei, Li, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, IEEE, 2009, pp. 248-255.

[41] Chen, Liang-Chieh and Papandreou, George and Kokkinos, Iasonas and Murphy, Kevin and Yuille, Alan L, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *arXiv preprint arXiv:1412.7062*, 2014.

[42] Johnson, Justin and Alahi, Alexandre and Fei-Fei, Li, "Perceptual losses for real-time style transfer and super-resolution," in *Computer Vision--ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, Springer, 2016, pp. 694-711.

[43] Isola, Phillip and Zhu, Jun-Yan and Zhou, Tinghui and Efros, Alexei A, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125-1134.

[44] Lang, Chunbo, "Official PyTorch Implementation of Learning What Not to Segment: A New Perspective on Few-Shot Segmentation," 2022. [Online]. Available: https://github.com/chunbolang/BAM.

[45] Zhu, Jun-Yan and Park, Taesung, "Image-to-Image Translation in PyTorch," [Online]. Available: https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix.

[46] Iqbal, Ehtesham and Safarov, Sirojbek, "Official Pytorch implementation of Multi-Similarity and Attention Guidence for Boosting Few-Shot Segmentation.," AIVResearch, [Online]. Available: https://github.com/AIVResearch/MSANet.