NTU CE6190 Group Project: Experimental Study of Few Shot Segmentation using PANet

Seow Wei Liang
WEILIANG003@e.ntu.edu.sg

Han Shuangpeng
shuangpe001@e.ntu.edu.sg

Zhang Hanwen
hanwen001@e.ntu.edu.sg

Abstract

In our paper, we have experimented with a Few-Shot Segmentation using the PANet architecture. For the main results of our experiments, we provided comparisons between 1-way 1-shot and 1-way 5-shot segmentation. To complement the 1-way 1-shot and 1-way 5-shot results, we have included the 2-way 1-shot results for our implementation. We also included a study on weak annotations (Bounding Box and Scribble) and compared it with dense annotations. For our experiments, we provided a comprehensive study of different loss functions (Cross entropy loss, Focal loss, Tversky loss, Focal Tversky loss, Dice loss, and Unified Focal loss) to tackle the class imbalance issue in image segmentation. We explored different hyperparameters for PANet such as learning rate and batch size. We also experimented with changing the VGG16 feature extractor to improve results such as VGG19, ResNet50, and ResNet101. Using the VGG16 feature extractor, we also examined freezing all layers of the neural network except 1 layer where weights are updated. We explored a different optimizer Adam and compared it with the SGD optimizer which helped to improve the mean-IoU due to the adaptive learning rate. Lastly, we also compared the effect of prototype alignment regularization on the segmentation results.

1. Introduction

Few-shot segmentation is an emerging field in computer vision that aims to address the challenge of performing image segmentation with only a few labeled examples. Unlike traditional segmentation tasks that rely on extensive, densely annotated datasets for training, few-shot segmentation focuses on models that can quickly adapt to new tasks or object categories using a minimal number of annotations. This approach is particularly valuable in scenarios where collecting large amounts of labeled data is impractical or expensive. The fundamental challenge in few-shot segmentation is to develop algorithms that can generalize well from limited data. This requires the model to learn robust and transferable features that can be applied to new, unseen categories based on a small number of examples. The process typically involves two sets of images: a few annotated images known as the 'support set', and the 'query set' of images that need to be segmented. The goal is to use the knowledge gained from the support set to accurately segment the query set.

To achieve this, few-shot segmentation techniques often leverage advanced machine learning approaches like meta-learning, transfer learning, and metric learning. Meta-learning, or learning to learn, trains the model on a variety of tasks so that it can quickly adapt to new tasks with minimal data. Transfer learning allows the model to transfer knowledge from a related task with abundant data to a new task with scarce data. Metric learning focuses on learning a distance function that can effectively measure the similarity between the support and query images. One common approach in few-shot segmentation is to train a model to learn a general representation of images and then fine-tune it on the small support set for each new task. This approach enables the model to quickly adapt to new categories with minimal additional training. The practical applications of few-shot segmentation are extensive, especially in fields where annotated data is scarce or costly to obtain, such as medical imaging, remote sensing, and real-time video analysis. The ability to efficiently segment new objects or scenarios with limited examples opens up new possibilities in automated image analysis and aids in the development of more flexible and adaptable computer vision systems.

Few-shot Segmentation is a technique for image segmentation in the scenario when we have few labeled data. We can use few-shot learning to transfer the learning from known classes to unseen novel classes. Few Shot learning usually presents as an N-way K-shot scenario where N represents the number of classes (way) in an episode and K represents the number of examples (shot) in each class. In a scenario of three sampled classes (N = 3) and five examples (K = 5) per class to compose an episode, the total number of images in the support set will be 15. Image segmentation involves grouping or labeling similar regions or segments in an image on a pixel level. Image Segmentation can be divided into three main types which are semantic segmenta-

tion, instance segmentation, and panoptic segmentation. In our few shot learning experiments, we will focus on Semantic Segmentation. Semantic segmentation aims to classify both stuff and things within an image and predicts a unique class label for each image pixel. However, semantic segmentation cannot differentiate instances of the same class within an image. We will use the PANet model from [11] to experiment with few-shot Segmentation. Specifically, we experiment with different hyperparameters such as the learning rate and batch size. We provide a comparison between 1-way 1-shot and 1-way 5-shot segmentation for the original PANet paper, the Reproduced PANet paper, and our implementation. We also experiment with different feature extractors to improve results such as VGG19, ResNet50, and ResNet101. Using the VGG16 feature extractor, we experiment with freezing all layers of the network except 1 layer and analyze the results. To improve on loss optimization using training, we experiment with different optimizer (Adam) which allows adaptive learning rates. To tackle the class imbalance problem in typical image segmentation scenarios, we try a variety of different loss functions (Cross entropy loss, Focal loss, Tversky loss, Focal Tversky loss, Dice loss, and Unified Focal loss) to mitigate this problem. We also explored different loss function combinations for main loss and align loss. we also tested our improved PANet model on weak annotations such as bounding boxes and scribbles which showed competitive results compared to Dense annotations. Lastly, we compared the effect of prototype alignment regularization on the results.

2. Related Works

2.1. One-Shot Learning for Semantic Segmentation

In an earlier publication from 2017, Amirreza Shaban et. al. [9] introduced a novel two-branched neural network approach for semantic image segmentation with minimal data. Unlike traditional methods requiring extensive training data, their model efficiently learns from a single or a few annotated examples. The first branch of the network processes a labeled image to generate parameters, while the second applies these parameters to a new image for segmentation. This architecture significantly reduces training time and computation, and when tested on the PASCAL VOC 2012 dataset, it outperformed baseline methods by 25% in mean Intersection over Union (IoU). This breakthrough demonstrates a promising direction in low-shot learning and its potential applications in computer vision.

2.2. co-FCN

In 2018, Kate Rakelly et. al. [7] proposed a conditional network, termed co-FCN, which is optimized end-to-end for fast and accurate few-shot segmentation. This network leverages a support set of annotated images to perform inference on an unannotated query image. The co-FCN network operates by conditioning on the support set through feature fusion, allowing inference on a query image without further optimization. This method is suitable for interactive use due to its ability to condition annotations in a single forward pass. The co-FCN demonstrates competitive accuracy even with sparse annotations, like a single positive and negative pixel, thereby reducing the annotation burden for segmenting new concepts. The network is trained by simulating few-shot tasks using a densely labeled semantic segmentation dataset. The paper presents results on the PASCAL VOC dataset, comparing their method with others like fine-tuning and global parameter prediction. The co-FCN outperforms traditional fine-tuning methods in terms of accuracy and efficiency, particularly in handling sparse annotations. Its design eliminates the need for optimization at test time, making it faster and more suitable for interactive applications.

2.3. SG-One network

In the same year, Xiaolin Zhang et. al. [13] demonstrated the SG-One network, a novel approach for one-shot image semantic segmentation. The technique hinges on a unified framework that uses masked average pooling and cosine similarity to generate robust feature representations from a single densely labeled support image. This approach enables the network to guide the segmentation process of a query image more effectively. SG-One distinguishes itself by its ability to predict segmentation masks for unseen classes without parameter adjustment, outperforming baseline methods with a mean Intersection over Union (mIoU) score of 46.3% in tests on the Pascal VOC 2012 dataset. The research demonstrates significant advancements in reducing the need for extensive labeling in object semantic segmentation, showing promise in applications requiring minimal supervision and rapid adaptation to new classes.

2.4. BAM

In 2022, a more recent BAM paper [5] presented a novel approach to address the bias problem in Few-Shot Segmentation (FSS) models. The main focus of the study is on improving the performance of FSS models, which are often biased towards classes they have seen during training, thereby hindering their ability to recognize new concepts. The authors propose a novel scheme, named BAM, which adds a base learner to the conventional FSS model (meta learner). The base learner explicitly predicts the regions of base classes, i.e., the areas that do not require segmentation. This approach helps in identifying regions in the query image that might confuse the meta learner. The results from the base learner and meta learner are then adaptively integrated to produce accurate segmentation predictions. The paper introduces an adjustment factor to estimate

the scene differences between input image pairs, aiding in model ensemble forecasting. This factor is calculated based on the Frobenius norm of the Gram matrices difference of the two input images. An ensemble module integrates the predictions from the base and meta learners. This module plays a crucial role in suppressing falsely activated regions of base classes, thereby enhancing the accuracy of novel object localization. The proposed approach is extended to a more challenging setting, generalized FSS, where pixels from both base and novel classes need to be identified. The results demonstrate that the proposed approach not only mitigates the bias problem but also sets new state-of-the-art performances across various settings, including 1-shot and 5-shot scenarios.

2.5. PANet

The architecture of PANet is shown in Fig. 1 retrieved from original PANet paper [11]. The example shown in Fig. 1 is a 2-way 1-shot example with two classes and 1 example per class in a training episode. PANet training contains two parts, support to query (main loss) and query to support (align loss). Initially, a VGG16 [10] feature extractor is used to extract features from the support and query set. In the support-to-query scenario, Masked Average Pooling is applied to the support features extracted from VGG16 to obtain class prototypes. The pooled support prototypes are then used to compute the cosine similarity with the query features. The similarity results are then upsampled and compared with query ground truth using the cross entropy loss. In the query-to-support scenario, the predicted mask from the support to query is used to perform Masked Average pooling on the query features instead and obtain the query prototypes. The query prototypes are then used to compute the cosine similarity with the support features. The final result is then upsampled and compared to support ground truth to obtain the loss using cross entropy loss. This acts like a regularization to the original loss function using support to query.

2.6. Class Imbalance

The class imbalance problem refers to the imbalance in the number of classes during model training in classification tasks. The image segmentation task is prone to class imbalance problems due to some classes having pixels that are under-represented in the training data, whereas the background of the image is the dominant class and over-represented. To tackle this problem, we explored different loss functions such as Cross entropy loss, Focal loss [6], Tversky loss [8], Focal Tversky loss [1], Dice loss [14] and Unified Focal loss [12] to mitigate this class imbalance problem. We found that Unified Focal loss obtains the best mean-IoU score among all loss functions explored.

2.7. Loss Optimization

In the paper, the Stochastic Gradient Descent (SGD) optimizer with momentum is used to optimize the loss function. SGD optimizer, however, does not provide adaptive learning rates. In our experiments, we leverage the Adam [4] optimizer which has an advantage over the SGD by using the running average of the gradients as well as the running average of the squared gradients.

3. Experiments

3.1. Dataset

In our paper, we utilized the Pascal VOC 2012(PASCAL Visual Object Classes) dataset [2] containing objects images from different visual object types that are not pre-segmented objects and typically used for the supervised learning task. The dataset can be used for different computer vision tasks such as classification, detection, segmentation, and person layout. There are 11,530 images in the train/validation data set, including 27,450 ROI tagged objects and 6,929 segmentations consisting of a total of 20 object classes.

We adhere to the methodology outlined in the original PANet paper. The PASCAL VOC dataset comprises 20 categories, evenly split into 4 groups, each consisting of 5 categories. Models undergo training in 3 of these groups and are assessed on the remaining one using cross-validation. You can locate the categories in each split in reference [9]. For testing, we average results from 5 separate runs, each comprising 1,000 episodes and utilizing different random seeds.

3.2. Implementation Details

We performed our experiments and compared them with the baseline PANet model with initial hyperparameters set similar to PANet using SGD optimizer with a learning rate of 1e-3, momentum of 0.9, weight decay of 0.0005, and batch size set to 1. The learning rate decayed by 0.1 every 10,000 iterations using MultiStepLR Scheduler. In the baseline model, we use the VGG16 feature extractor similar to the paper and cross entropy loss. Following the original implementation in PANet paper, Input images are resized to (417, 417) and augmented using random horizontal flipping. The total number of training iterations is 30000. We mostly compare our experiments using 1-way 1-shot settings due to time constraints but we also provide a comparison with 1-way 5-shot settings.

Our experiments include different hyperparameter settings for learning rate, batch size, studying the effect of different loss functions, feature extractors, and optimizers. We also examine the effect of freezing layers in the feature extractor and prototype alignment regularization (PAR).

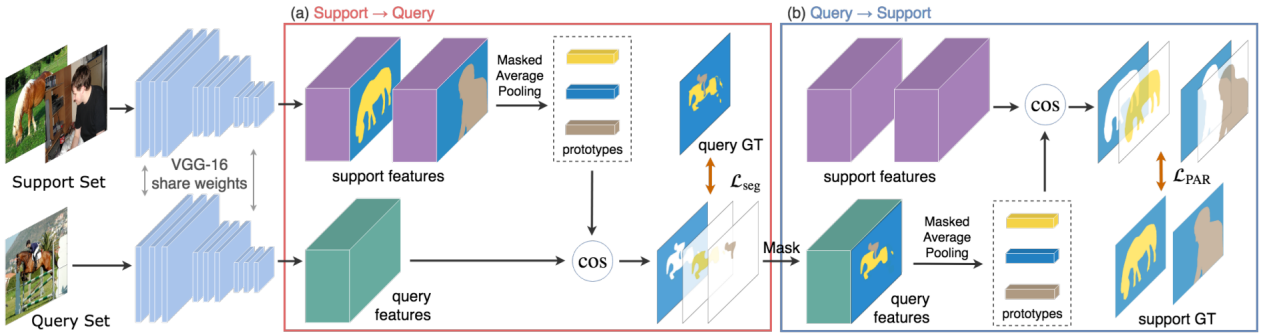

Figure 1. A schematic diagram of PANet [11] architecture.

3.3. Evaluation Metrics

We utilize two metrics to evaluate the PANet model similar to the original paper. One of them is mean-IoU which measures the IoU of each foreground class and averages over all classes. The other is the binary-IoU which takes all object categories as foreground and averages the IOU of foreground and background. Mean-IoU is the preferred metric among the two as it considers each foreground category which is more reflective of the actual model performance.

3.4. Improved Performance of PANet

The performance of improved PANet is shown in Tab. 1 and Tab. 2. The first rows in tables are the reported result in [11]. Based on the same experiment configurations, the model performances are reproduced shown in the second rows. In order to improve the model performance, we fine-tuned hyper-parameters, modified model architectures, and improved optimization methods. The performances of our improved PANet are shown in the last rows. It can be seen that the mean-IoU of the 1-way-1-shot case can be improved from 47.6% to 48.2%, and the mean-IoU of the 1-way-5-shot configuration is increased to 56.2%.

3.5. 2-way 1-shot Comparison

Besides 1-way 1-shot and 1-way 5-shot experiments shown in Section 3.4, we also attempt to evaluate our improved PANet model under 2-way 1-shot and display our results in Tab. 3. From Tab. 3, results for 2-way 1-shot with mean-IoU of 42.3% and binary IoU of 65.1% is slightly lower compared to 1-way 1-shot with mean-IoU of 43.9% and binary IoU of 67.0%. However, the 2-way 1-shot results are still quite close to the 1-way 1-shot. This shows that our model can be performed comparatively well in both 1-way 1-shot and 2-way 1-shot settings. It is worth noting that, due to time constraints, we just show the result for the 1-shot split-1 scenario.

3.6. Comparison of Dense Annotations with Weak Annotations

In accordance with the guidelines outlined in the original PANet paper [11], we carried out experiments to assess the impact of incorporating weak annotations such as scribbles and bounding boxes. In the testing phase, the detailed pixel-level annotations in the support set were substituted with either scribbles or bounding boxes. These annotations were automatically derived from dense segmentation masks. Each bounding box was generated based on a randomly selected instance mask from each support image. Tab. 4 presents a comparison of the performance of an image segmentation of PANet using three different types of annotations: Dense, Scribble, and Bounding box.

Dense annotations yield the highest mean-IoU of 41.4% indicating that detailed, pixel-level annotations provide the best performance for the model. This is followed by scribble annotations with a mean-IoU of 36.0%, and bounding box annotations with a mean-IoU of 38.1%. For the binary mean-IoU, dense annotations still perform best with a value of 66.0%, but bounding box annotations achieve comparable performance with scribble annotations. This suggests that while detailed annotations lead to better overall performance, the simpler bounding box annotations are almost as effective as scribbles when only the presence of an object (not the specific class) is considered. In other words, the difference between using scribbles and bounding boxes is minimal for binary segmentation tasks.

We also compare dense annotations and weak annotations such as scribble and bounding box using our own implementation of the VGG16 feature extractor, Adam optimizer, and unified focal loss. Our implementation showed an improvement across different types of annotations for dense, scribble, and bounding boxes in Tab. 4. For dense annotations, we present some few-shot segmentation examples using 1-way 1-shot in Fig. 3 (Plane, Bird, and Bicycle), and some failure examples in Fig. 4 (Goose and Ship). To compare against the results from dense annotations, we

Method	1-shot					5-shot				
	split-1	split-2	split-3	split-4	Mean	split-1	split-2	split-3	split-4	Mean
PANet [11]	42.3	58.0	51.1	41.2	48.1	51.8	64.6	59.8	46.5	55.7
Reproduced PANet [11]	41.6	57.7	49.6	41.4	47.6	52.4	64.6	59.2	47.7	56.0
Improved PANet (ours)	42.5	58.1	51.6	40.4	**48.2**	51.6	64.6	60.6	47.8	**56.2**

Table 1. Results of 1-way 1-shot and 1-way 5-shot segmentation on Pascal VOC 2012 dataset in terms of mean-IoU metric.

Method	1-shot	5-shot	Δ
PANet [11]	66.5	70.7	4.2
Reproduced PANet [11]	66.0	71.2	5.2
Improved PANet (ours)	65.9	71.2	5.3

Table 2. Results of 1-way 1-shot and 1-way 5-shot segmentation on Pascal VOC 2012 dataset in terms of Binary-IoU metric.

	Mean-IoU	Binary-IoU
1-way 1-shot	43.9	67.0
2-way 1-shot	42.3	65.1

Table 3. Results for comparing 1-way 1-shot and 2-way 1-shot settings for the split-1 scenario.

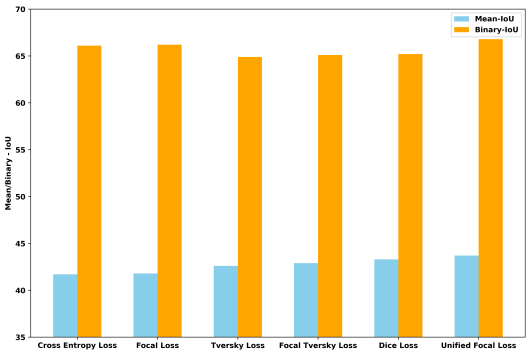

Figure 2. Effect of different loss functions on mean-IoU score and binary IoU score for the 1-way 1-shot split-1 scenario.

provide some example cases (Boat and Bird) for weak annotations (Bounding box, Scribbles) in support images and showed that weak annotations also performed competitively well against dense annotations in Fig. 5.

Annotations	Adam & Unified focal loss	Mean-IoU	Binary-IoU
Dense	✗	41.4	66.0
Dense	✓	43.9	67.0
Scribble	✗	36.0	62.8
Scribble	✓	38.7	64.1
Bounding box	✗	38.1	62.8
Bounding box	✓	39.3	63.0

Table 4. Effect of weak annotations, including scribbles and bounding boxes, is examined. In the original PANet model, dense annotations are employed as a fundamental baseline. The results are obtained from the 1-way 1-shot split-1 configuration.

3.7. Ablation Study

3.7.1 Effect of Different Loss functions

In this section, we experimented with a variety of loss functions designed to tackle class imbalance and compared it to cross entropy loss used in the Baseline model for the single split scenario. We presented the results for 1-way 1-shot in Fig. 2 and Fig. 6. For distribution-based losses, Focal loss (gamma=2) seems to perform better than cross entropy loss with 41.8% and 66.2% for the mean-IoU and binary-IoU respectively which indicates focusing on the hard examples improves performance. For region-based losses, Dice loss achieves the best performance with 43.3% and 65.2% for the mean-IoU and binary-IoU respectively. Tversky loss (alpha=0.7, beta=0.3) and Focal Tversky loss (alpha=0.7, beta=0.3, gamma=0.7) performed worse than Dice loss which indicates that different weighting for false negative and false positive worsened the results. Focal Tverksy loss performs better than Tversky loss which shows again focusing on the hard examples works. We also found that unifying the region-based losses and distribution-based loss achieves the best results. Specifically, we combined focal tverksy loss and focal loss with equal weightings for each loss achieving the best results with 43.7% and 66.8% for the mean-IoU and binary mean-IoU respectively.

3.7.2 Effect of Learning Rate

We have tested the baseline PANet on different learning rates, 1e-2, 1e-3, 1e-4, and present the results in Tab. 5. Learning rate of 1e-4 gives the best result which obtains the highest mean-IoU of 42.2% and binary mean-IoU of 66.3%. Having a lower learning rate of 1e-5 lowers the mean-IoU and binary mean-IoU as the model takes longer to converge. Having a higher learning rate of 1e-2 and 1e-3 may cause the model to diverge or overshoot near the optima.

3.7.3 Effect of Batch Size

We have tested the baseline PANet on different batch size of 1 and 2 and present the results in Tab. 6. We did not try

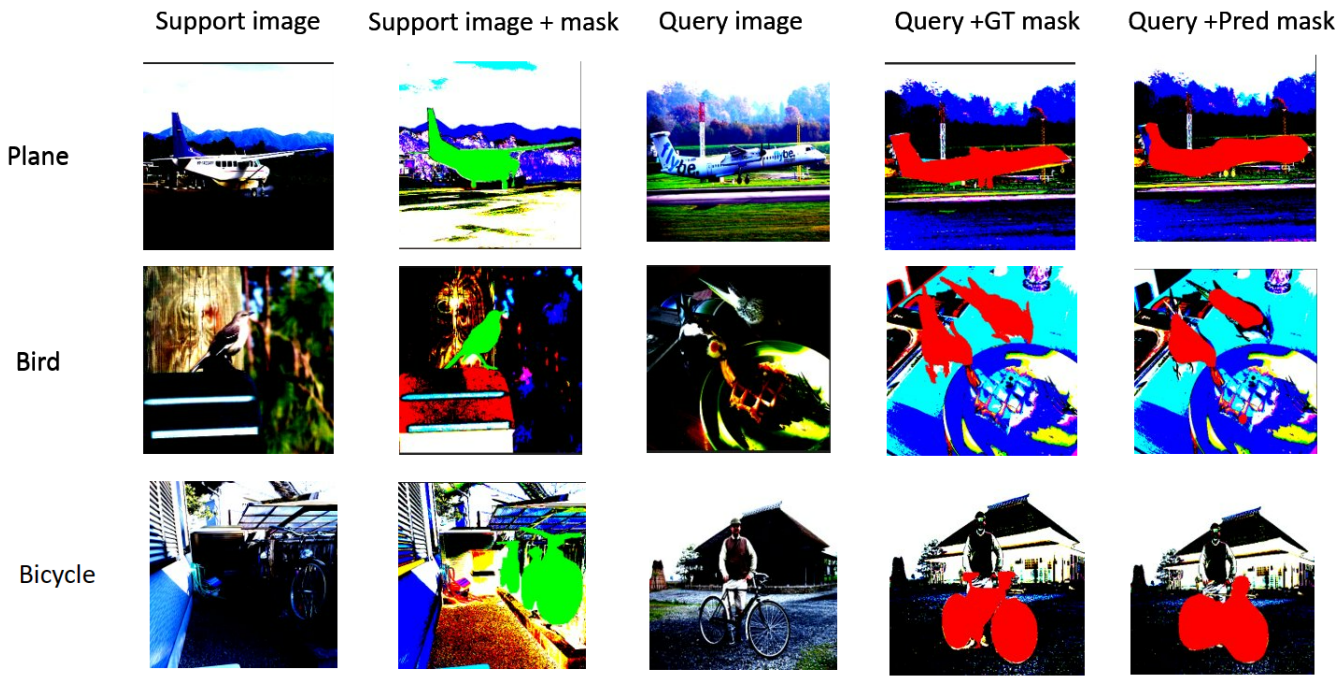

Figure 3. Few Shot Segmentation examples for our model (1 Way 1 Shot) on Parscal VOC 2012

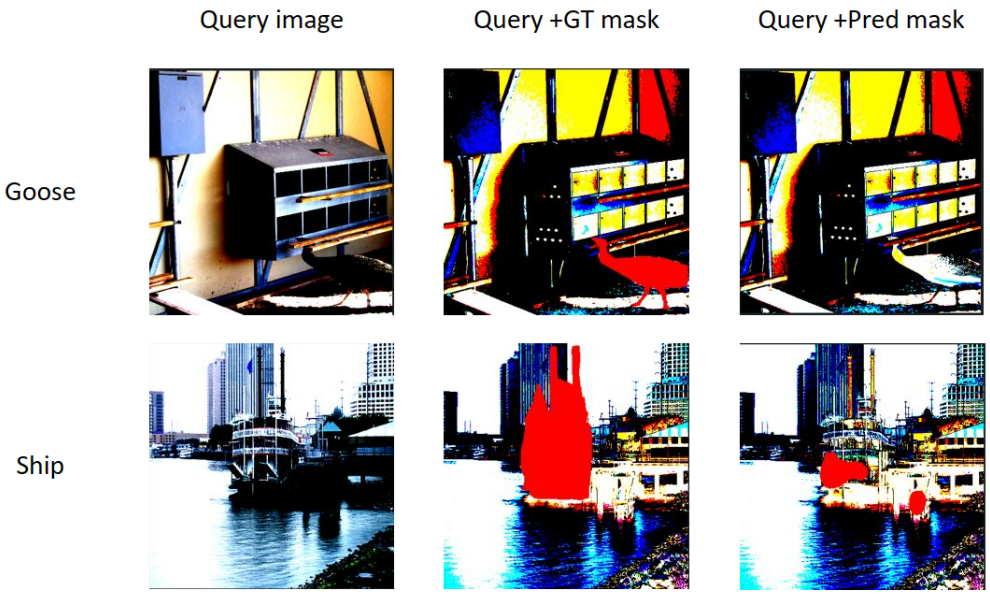

Figure 4. Failure examples for our model (1 Way 1 Shot) on Parscal VOC 2012

batch size 3 due to time constraints and the initial loss was much higher than batch size 2. A batch size of 2 gives the best result as shown in the Tab. 6 which obtains the highest mean-IoU of 42.2% and 66.3% for the mean-IoU and binary mean-IoU respectively.

3.7.4 Effect of Different Feature Extractor

In this section, we experimented with different pre-trained feature extractors such as VGG19 [10], ResNet50, ResNet101 [3] and compared it with the baseline VGG16 feature extractor and present the results in Fig. 7 and Fig. 8.

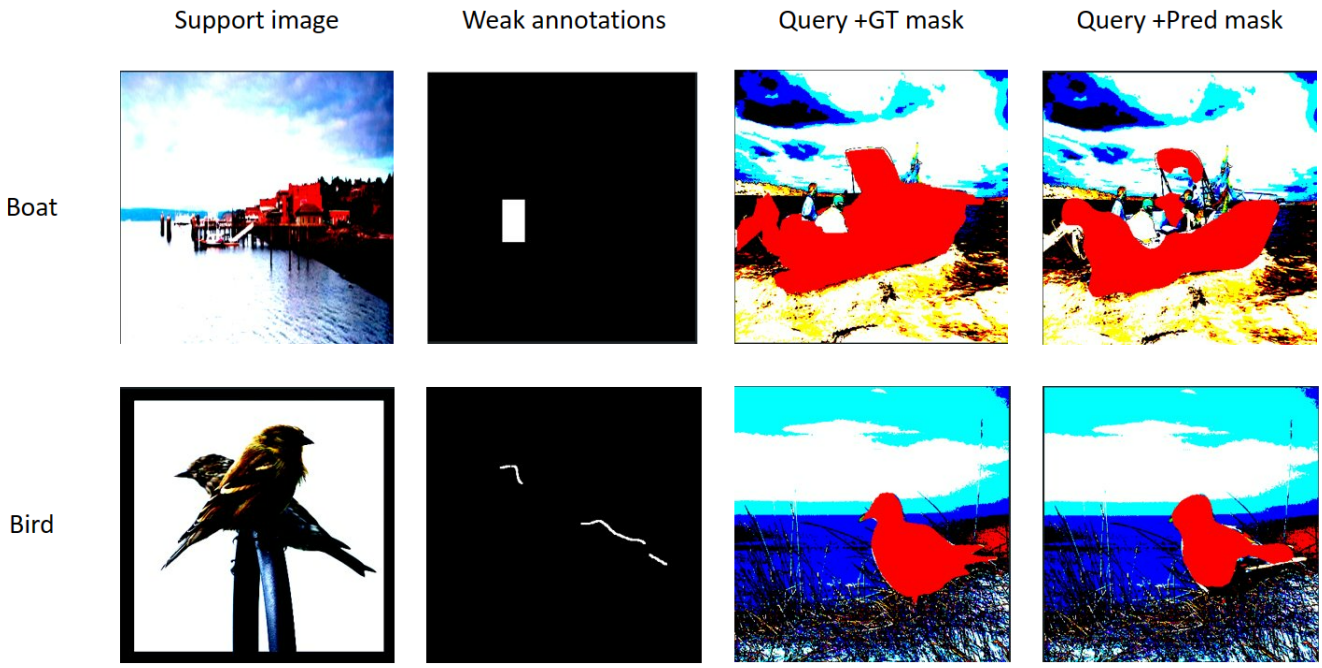

Figure 5. Few Shot Segmentation using Weak annotations (Bounding box and Scribbles) for our model (1 Way 1 Shot) on Parscal VOC 2012

Learning Rate	Mean-IoU	Binary-IoU
1e-2	diverge	diverge
1e-3	41.7	66.1
1e-4	42.2	66.3
1e-5	40.1	65.0

Table 5. Effect of learning rate on mean-IoU score and binary IoU score for the 1-way 1-shot split-1 scenario.

Batch Size	Mean-IoU	Binary-IoU
1	41.7	66.1
2	42.2	66.3

Table 6. Effect of batch size on mean-IoU score and binary IoU score for the 1-way 1-shot split-1 scenario.

Comparing ResNet and VGG architectures, VGG16 and VGG19 architectures seem to perform better than ResNet50 and ResNet101. Within the same model architecture, more neural network layers improve results. VGG16 achieves the best performance of 47.6% and 66.0% for the mean-IoU and binary mean-IoU respectively.

3.7.5 Effect of Different Optimizers used

In this section, we experimented with different optimizers of Adam and compared it to SGD used in the baseline model. Adam optimizer was configured with a learning rate of 1e-5, weight decay of 0.0005, betas=(0.9, 0.999). From the results, the Adam optimizer seems to perform better performance of 42.6% and 66.6% for the mean-IoU and binary mean-IoU respectively. This could be attributed to the adaptive learning rate used in Adam compared to SGD which helps to converge the model faster.

Optimizer	Mean-IoU	Binary-IoU
SGD	41.7	66.1
Adam	42.6	66.6

Table 7. Effect of different Optimizer on mean-IoU score and binary IoU score for the 1-way 1-shot split-1 scenario.

3.7.6 Effect of Freezing Layers in Feature Extractor

In this section, we explore the impact of layer freezing within the feature extractor. In the PANet architecture, VGG-16 serves as the encoder responsible for feature extraction. The encoder comprises a total of 9 layers, with 4 of them being non-trainable max-pooling layers, meaning

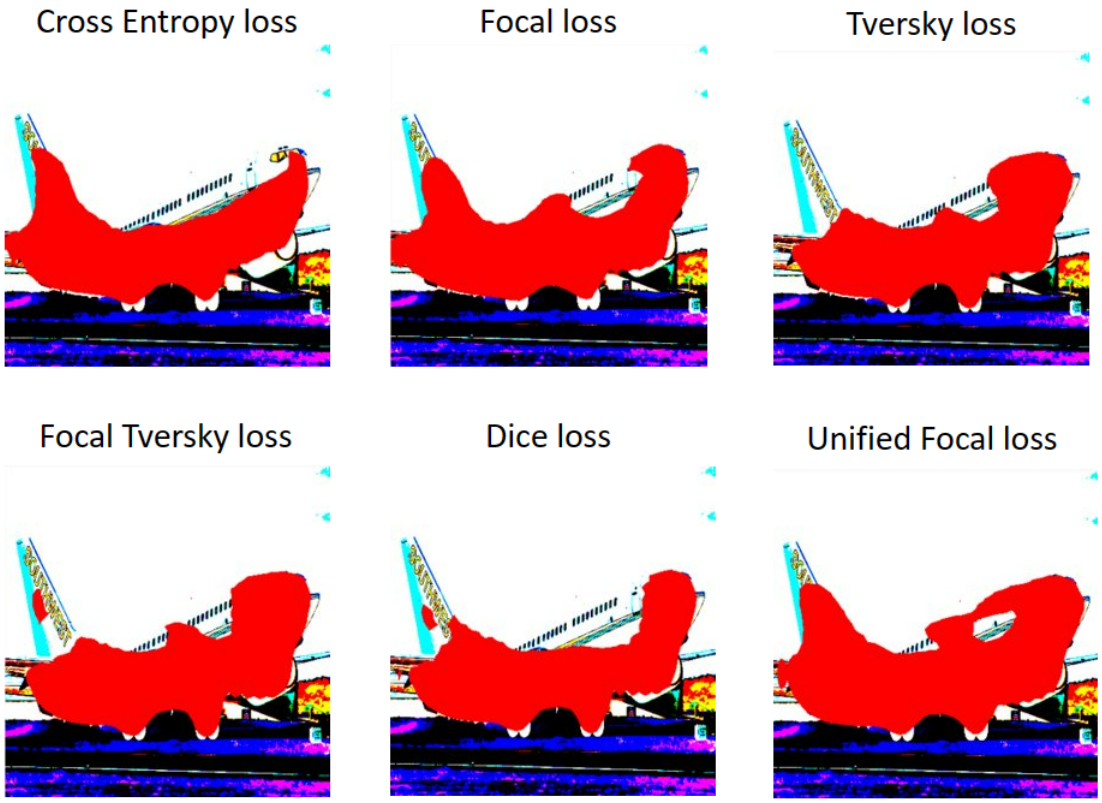

Figure 6. A comparison of different loss functions for Few shot segmentation

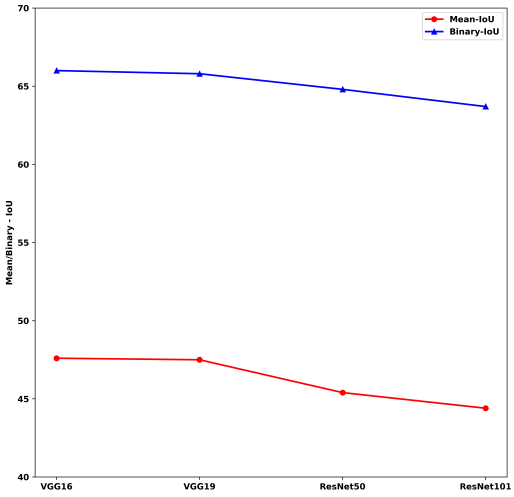

Figure 7. Effect of different encoders on mean-IoU score for 1-way 1-shot settings.

they have no adjustable parameters. The remaining layers, specifically layers 1, 3, 5, 7, and 9, possess trainable parameters that can be frozen or learnable. Consequently, the experiment aims to permit one of these layers to be trainable while maintaining the others in a frozen state. Fig. 9 shows the results obtained from this experiment.

In the conducted experiment analyzing the effects of selective layer training within a VGG-16 based PANet architecture, it was discovered that the layer-specific unfreezing significantly impacts the model's performance. The experiment's design allowed for individual layers to be trainable while keeping others frozen, aiming to discern the contribution of each layer to the overall accuracy. Results indicated that higher layers when made trainable, notably enhanced model accuracy. Layer 9, in particular, demonstrated the most substantial improvement, achieving a mean-IoU of 41.6% and a Binary mean-IoU of 65.9%. This suggests that the later stages of the feature extraction process hold more potential for optimization through training compared to earlier layers. In contrast, the training of Layer 1 resulted in the lowest performance, with a mean-IoU of 32.4% and a Binary mean-IoU of 60.3%. The trend observed from the data suggests that allowing deeper layers to learn can lead to better feature representations, which translates into higher

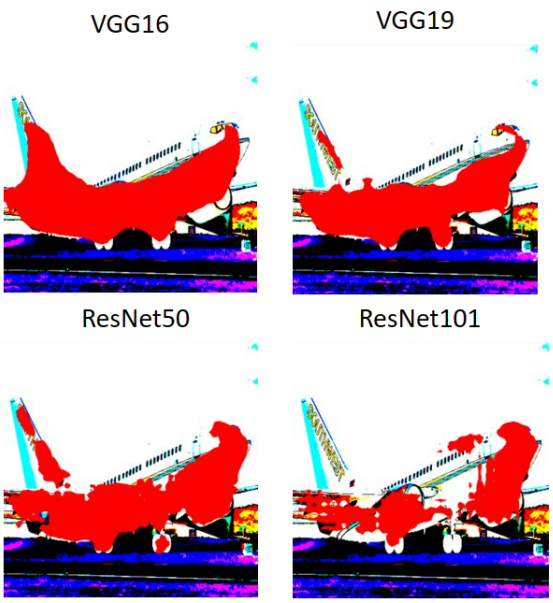

Figure 8. A comparison of different feature extractor for Few shot segmentation

accuracy in segmentation tasks, as evidenced by the incremental increases in both mean-IoU and binary mean-IoU values. This pattern underscores the importance of targeted layer training in fine-tuning deep learning models for complex tasks such as feature extraction.

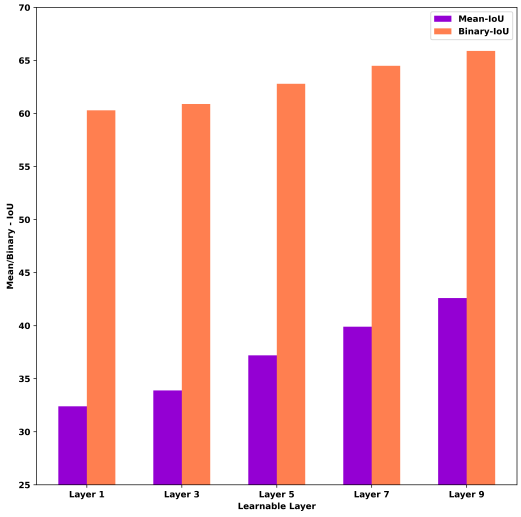

Figure 9. The effects of freezing layers in feature extractor. The learnable layer represents the layer of the VGG-16 pretrained model with learnable parameters, while the remaining layers remain in a frozen state. The results are obtained from the 1-way 1-shot split-1 scenario.

3.7.7 Effect of Prototype Alignment Regularization (PAR)

In this section, we used the model with dice loss, Adam optimizer with a learning rate of 1e-5, weight decay of 0.0005, betas=(0.9, 0.999) to test the effect of PAR and present the results in Tab. 8. We observed that for the single split scenario, without PAR performs better with 43.6% and 65.6% for mean-IoU and binary mean-IoU respectively than with PAR which shows that PAR may not improve results for all cases.

	Mean-IoU	BinaryIoU
w PAR	43.2	65.3
w/o PAR	43.6	65.6

Table 8. Effect of PAR on mean-IoU score and binary IoU score for the 1-way 1-shot split-1 scenario.

4. Conclusion

In conclusion, our paper delved into Few Shot Segmentation using the PANet architecture. We presented significant findings by comparing 1-way 1-shot and 1-way 5-shot segmentation and complemented these results with the 2-way 1-shot outcomes. Additionally, we conducted a thorough analysis of weak annotations (Bounding box and Scribble) versus dense annotations. Our experiments encompassed an extensive examination of various loss functions aimed at addressing class imbalance in image segmentation, including Cross entropy loss, Focal loss, Tversky loss, Focal Tversky loss, Dice loss, and Unified Focal loss. We scrutinized PANet's performance concerning diverse hyperparameters such as learning rate and batch size, and experimented with altering the VGG16 feature extractor, exploring enhancements via VGG19, ResNet50, and ResNet101. A focused study involved freezing all neural network layers except one for weight updates using the VGG16 feature extractor. Our research also entailed the evaluation of different optimizers, comparing Adam with SGD, which notably enhanced mean-IoU due to adaptive learning rates. Finally, we examined the impact of prototype alignment regularization on segmentation results, culminating in a comprehensive assessment of various aspects vital to Few-Shot Segmentation with PANet.

5. Acknowledgement

We would like to express our sincere appreciation to Prof. Lin Guosheng for his invaluable support and guidance throughout the duration of this Group Project. Prof. Lin's dedication to organizing this project and imparting knowledge to us has been instrumental in our learning experience.

We are grateful for his commitment to our academic growth.

6. Additional Notes

In the past, our group consisted of four members. However, towards the end of the semester, we received notice that one of our group members, Jin Zhengyang, had chosen to withdraw from NTU's PhD program. Fortunately, we were able to successfully complete and deliver this project on schedule. Consequently, the work has been carried out by only three students listed on the first page of this report.

7. Project video

Project video is uploaded to the OneDrive SharePoint. Anyone with NTU account can view the following link. To access the project video, please use a NTU account to access the following link: Link to SharePoint

References

[1] Nabila Abraham and Naimul Mefraz Khan. A novel focal tversky loss function with improved attention u-net for lesion segmentation, 2018. 3

[2] Zheng Dong, Ke Xu, Yin Yang, Hujun Bao, Weiwei Xu, and Rynson W. H. Lau. Location-aware single image reflection removal, 2021. 3

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 6

[4] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 3

[5] Chunbo Lang, Gong Cheng, Binfei Tu, and Junwei Han. Learning what not to segment: A new perspective on few-shot segmentation, 2022. 2

[6] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2018. 3

[7] Kate Rakelly, Evan Shelhamer, Trevor Darrell, Alyosha Efros, and Sergey Levine. Conditional networks for few-shot semantic segmentation, 2018. 2

[8] Seyed Sadegh Mohseni Salehi, Deniz Erdogmus, and Ali Gholipour. Tversky loss function for image segmentation using 3d fully convolutional deep networks, 2017. 3

[9] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation, 2017. 2, 3

[10] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015. 3, 6

[11] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmentation with prototype alignment. In *proceedings of the IEEE/CVF international conference on computer vision*, pages 9197–9206, 2019. 2, 3, 4, 5

[12] Michael Yeung, Evis Sala, Carola-Bibiane Schönlieb, and Leonardo Rundo. Unified focal loss: Generalising dice and cross entropy-based losses to handle class imbalanced medical image segmentation, 2021. 3

[13] Xiaolin Zhang, Yunchao Wei, Yi Yang, and Thomas Huang. Sg-one: Similarity guidance network for one-shot semantic segmentation, 2018. 2

[14] Rongjian Zhao, Buyue Qian, Xianli Zhang, Yang Li, Rong Wei, Yang Liu, and Yinggang Pan. Rethinking dice loss for medical image segmentation. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 851–860, 2020. 3