Few-shot Semantic Segmentation Exploration

Bingnan Li
2020533092
libn@shanghaitech.edu.cn

Yifan Qin
2020533005
qinyf1@shanghaitech.edu.cn

Yan Zeng
2020533182
zengyan@shanghaitech.edu.cn

Haoyuan Tian
2020533013
tianhy@shanghaitech.edu.cn

Shuhao Zhang
2020533164
zhangshh2@shanghaitech.edu.cn

Abstract

Semantic image segmentation, a critical task in computer vision, has traditionally relied on large-scale labeled datasets for training deep learning models. However, acquiring and annotating such datasets can be arduous and impractical. To address this challenge, few-shot segmentation methods have emerged, aiming to achieve accurate segmentation with limited annotated samples. In this paper, we propose a framework that combines the power of diffusion models and the cross-convolution network to enable few-shot segmentation. The diffusion model captures complex data distributions and encodes images into semantic feature maps, while the cross-convolution network effectively fuses information from the support set to facilitate the fine-tuning process. We formerly employed the SENet framework to analyze feature importance, and FiLM framework to align features and labels, but we removed them in the new pipeline and apply support embedding instead. The model is trained on a single category, such as the person category, then fine-tune the model to other categories. By leveraging these techniques, our approach aims to achieve few-shot segmentation performance comparable to fully-supervised segmentation. Experimental results demonstrate the effectiveness and potential of our proposed framework in overcoming the limitations of traditional segmentation methods and reducing the reliance on extensive labeled datasets.

1. Introduction

Semantic image segmentation, which involves assigning pixel-level labels to different objects or regions within an image, plays a crucial role in various computer vision tasks, such as autonomous driving [7], medical image analysis [27], and scene understanding [10]. Traditional segmentation methods typically rely on large-scale labeled datasets for training deep learning models to achieve accurate and generalizable results [13]. However, acquiring and annotating such datasets can be labor-intensive, time-consuming, and impractical, especially for rare or novel object categories.

To address the limitations of traditional segmentation approaches, recent research has focused on the development of few-shot segmentation methods. Few-shot segmentation aims to enable accurate segmentation with only a limited number of annotated samples. It allows models to adapt quickly to new, unseen object categories or scenarios, thereby reducing the dependence on large amounts of labeled data.

In this paper, we utilize the diffusion model [9] to capture the complex data distribution and encode the images into highly semantic feature maps. By leveraging the power of diffusion models, which have demonstrated impressive performance in modeling image distributions, we aim to extract rich and informative features that can facilitate accurate segmentation.

To enhance the segmentation capability of our model, we incorporate the cross-convolution network proposed by [1] into our framework. This network allows us to effectively fuse information between the target image and the support set. By leveraging the complementary information from the support set, our model becomes capable of segmenting unseen objects solely based on the limited support set, thus achieving few-shot segmentation.

Furthermore, we investigate the characteristics of the features extracted from the diffusion model by employing the SENet [11] framework to visualize the importance weights associated with each feature level. Through this analysis, we gain insights into the relative importance of different feature levels in the segmentation process. By understanding the contribution of each feature level, we can better comprehend the hierarchical nature of the segmentation task and optimize our model accordingly. The networks mentioned above formed the original pipeline of our model,

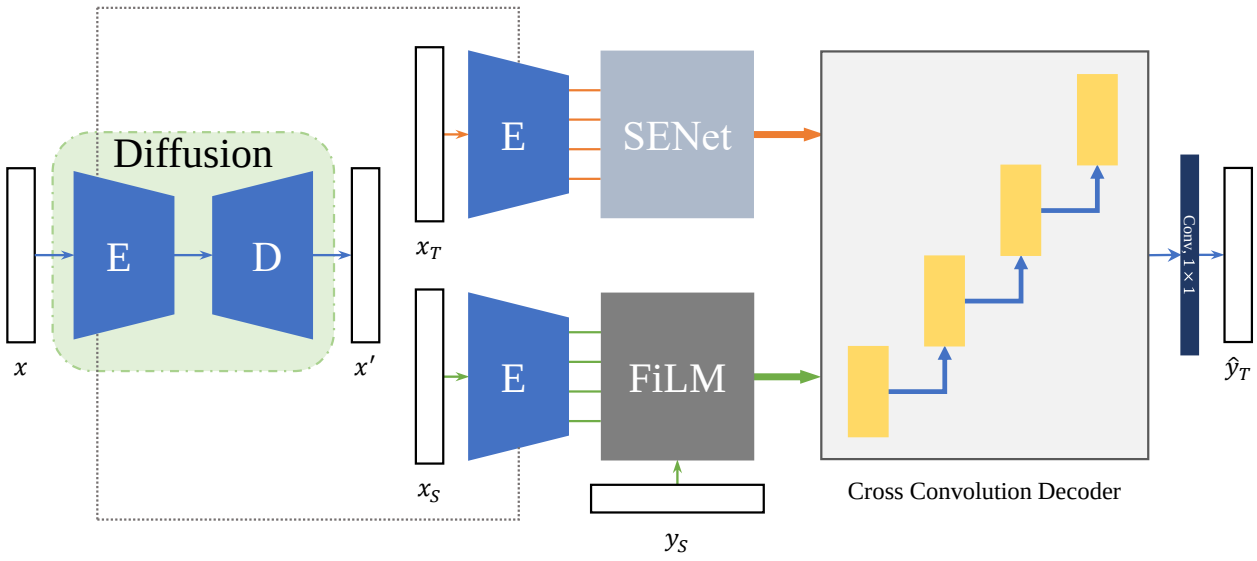

Figure 1. Original Pipeline of Our Model

which is shown in Figure 1.

However, the diversity of the dataset is too high regarding the number of categories. In the experiment, we found that it was hard to converge if the model is trained on the whole dataset with all categories. This is the consequence of the number of categories being too high, and the bad annotation, which make it challenging for a model to converge. We refine the pipeline by replacing the cross convolution decoder with support embedding, and we automatically stack the target image, support images and support labels as concatenation, so we remove the aligning framework from our model. Regarding the training schemes, we trained the model on the dataset limited with a single category, the person category is chosen in practice, which gave an effective performance, then we fine-tune the model to other categories. By leveraging the effective pre-training on specific categories, our model was able to successfully transfer its performance to other single categories through fine-tuning. The refined pipeline of our model is presented in Figure 2.

In summary, our approach combines the strengths of the diffusion model, the cross-convolution network, and fine-tuning. By leveraging the diffusion model to capture complex data distributions, incorporating the cross-convolution network for effective information fusion, and attaching support embedding to the decoder, we trained the model on just one category and fine-tuned it to include other categories. We aim to gain a few-shot segmentation performance as close as fully-supervised segmentation performance.

2. Related Work

In this section, we briefly describe the existing lines of research relevant to our work.

Diffision models DDPMs from [9] and [20] are a class of probabilistic generative models trained to denoise the image blurred with the Gaussian noise, and approximate the distribution of real images by the endpoint of the Markov chain. Here, we provide a detailed review of the formulation of Gaussian diffusion models from [9].

Starting with $x_0 \in \mathbb{R}^D$ or $\{0, \dots, 255\}^D$ being the original data. Let $x_1, \dots, x_T$ denote T latent variables $\mathbb{R}^D$ or $\{0, \dots, 255\}^D$. The forward diffusion process is defined as

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I).$$

It can be derived iteratively that

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I),$$

where $\beta_1, \dots, \beta_T$ is a variance schedule and $\alpha_j = 1 - \beta_j$, and $\bar{\alpha}_t = \Pi_{s=1}^{t}\alpha_s$. The reverse diffusion process starts with Bayes' theorem, deriving that

$$q(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu(x_t, t), \sigma_t^2),$$

where $\sigma_t^2 = \sum_\theta(x_t, t)$ is often fixed to untrained time dependent constants, e.g. $\sigma_t^2 = \beta_t$. The neural network $\mu_\theta(x_t, t)$ is shared among all time steps and is conditioned on t, which must predict the forward process posterior mean given x_t, that is

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)).$$

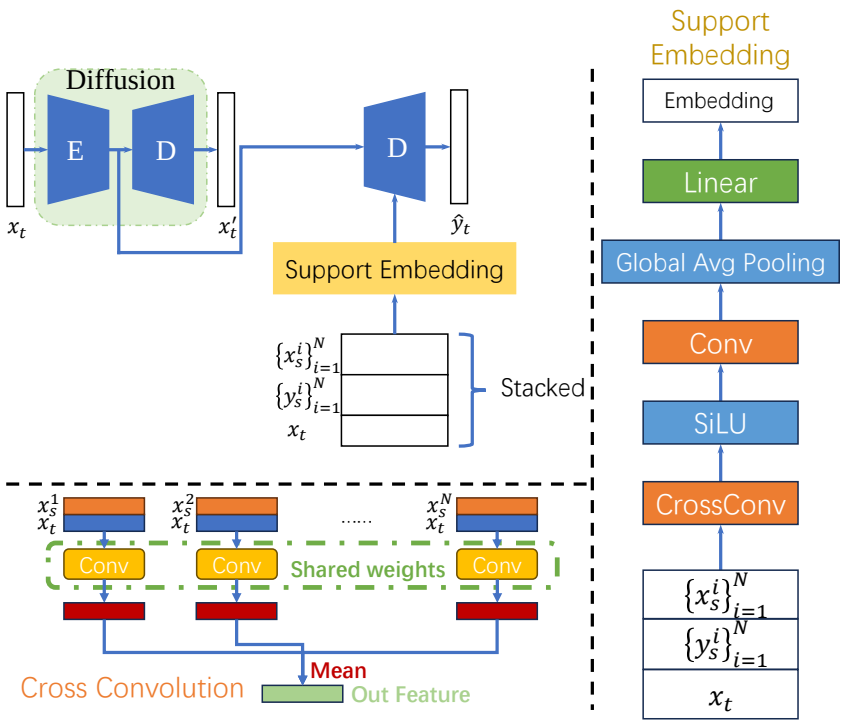

Figure 2. Refined Pipeline of Our Model

The dynamics of x is that

$$x_{t-1} = \mu_\theta(x_t, t) + \sigma_t z,$$

where $z \sim \mathcal{N}$. The noise scheme $\epsilon_\theta(x_t, t)$ that will be subtracted from x_t during sampling according is learned by the U-Net models. We will excavate the semantic information from $\epsilon_\theta(x_t, t)$ and the U-Net models and present a novel semantic segmentation method based on diffusion models.

Few-shot semantic segmentation Few-Shot Semantic Segmentation (FSS) is a technique that addresses the issue of limited availability of annotated data for new object classes in computer vision applications, it can generate dense masks for newly introduced classes with only a limited number of annotations. Previous approaches, which follow metric learning [6, 21], can be classified into prototype-based and matching-based methods. In prototype-based few-shot semantic segmentation [2, 6, 23], prototypes are learned during the training phase of the FSS model using a limited number of annotated samples for each new class. These prototypes are subsequently employed to classify the pixels within the query image. Due to its intuitive and computationally efficient nature, the prototype-based approach is widely used in few-shot semantic segmentation tasks. However, recent research [12, 25] has highlighted the limitation of a single prototype in covering all regions of an object, particularly for tasks involving pixel-wise dense segmentation. To overcome this issue, methods have been proposed that utilize EM and cluster algorithms [12, 25] to generate multiple prototypes for different parts of the objects, enabling more accurate segmentation. Unlike prototype-based methods, matching-based methods [16, 18, 22, 26] do not rely on prototypes for specific-class representation. Instead, they utilize pixel-level features and supplement more detailed support context, which enables them to capture more complex relationships between the query image and the support annotations. The efficacy of FSS has been validated in a wide range of computer vision tasks, such as medical image analysis [8, 19], autonomous driving [3, 7, 17], and robotics [14], among others. By reducing the cost and time required for annotating new object classes, FSS is now a valuable technique for real-world applications.

Transfer learning Fine-tuning is an approach to transfer learning in which the weights of a pre-trained model are trained on new data. It is common to keep the earlier layers frozen because they capture lower-level features, while later layers often discern high-level features that can be more related to the task that the model is trained on [24]. [15] adapted classification networks into fully convolutional networks and transfer their learned representations by fine-tuning to the segmentation task. Fine-tuning is also widely used in natural language processing, as [4] taking advantage of contextual token representations pre-trained from

unlabeled text and fine-tuned for a supervised downstream task. [5] proposed a pre-trained model that can be fine-tuned with one additional output layer to create models for a wide range of tasks. In our work, we take the advantage of transfer learn by training the model on one category from the dataset with a high diversity, then fine-tuning it to other categories with few efforts.

3. Method

Given the task of semantic segmentation, which involves a set of image-label pairs (x_i, y_i), a commonly-used approach is to learn a parametric function $y = f_\theta(x)$ for this task. Typically, this parametric function is modeled using a convolutional neural network (CNN). However, when the number of annotated images is limited, this approach often struggles, leading to challenges in few-shot semantic segmentation.

To address the problem of few-shot semantic segmentation, in this project, we first proposed a fusion of the Diffusion model and a segmentation model specifically designed for few-shot segmentation. The original proposed approach is illustrated in Figure 1.

In our dataset, we have two partitions: the support set and the target set. The support set, denoted as S, consists of images that are fully annotated. Each image in the support set is represented as $(x_s^i, y_s^i) \in S$, where x_s^i represents the image and y_s^i represents its corresponding annotated labels. On the other hand, the input set, denoted as T, contains images with categories that are typically different from those in the support set. In other words, the images in the input set are annotated as $(x_t^i, y_t^i) \in T$, where x_t^i represents an image from the target set and y_t^i represents its annotated labels. However, it is important to note that the annotated labels y_t^i of the images in the input set will never be provided to the model as input during the few-shot semantic segmentation task.

3.1. Diffusion Model

In our proposed approach, we integrate the Diffusion Model into our model, serving as both a reconstruction component for the original images and a feature extraction component for the segmentation part. This integration allows us to leverage the benefits of the Diffusion Model in the context of few-shot semantic segmentation.

It's important to highlight that the Diffusion Model not only takes the images from the support set, represented as x_s^i, as input but also utilizes the images from the input set, represented as x_t^i. By incorporating both sets of images, we can leverage the valuable information present in the input set, which may have different categories compared to the support set, enhancing the model's ability to generalize to new and unseen categories.

Furthermore, the reconstruction of all the images, including those from both the support set and the input set, can be utilized to supervise the Diffusion Model.

3.2. Segmentation Model

The segmentation model first requires the Encoder in the Diffusion Model to perform feature extraction from both the image from the Input set x_t^i and the image from the Support set x_s^i. After feature extraction, the channel number of the features map will increase dramatically and different strategies are applied to the feature map extracted from the input image and the feature map extracted from the support images.

For the feature map extracted from the input images x_t^i, we aim to apply a channel-wise emphasis, which helps us understand the information that the network is leveraging. The emphasis is created with the help of SENet [11] structure. This emphasis allows us to identify important features within the feature map.

In contrast, for the feature map extracted from the support images x_s^i, we want to integrate the annotated label information with them. However, the channel number of the annotated label is much less than the channel number of the feature map. If the annotated label is directly concatenated with the feature map, the feature map from the Diffusion Model Encoder will still dominate the information. What's worse, the annotated label is based on the original image, but the feature map is extracted from the Encoder, so the misalignment between the annotated label and the feature map in spatial dimension will occur. In order to tackle this problem, we decide to adopt the module called Feature-wise Linear Modulation (FiLM). With the help of the FiLM, the information from the annotated label can be integrated into the feature map of the support images, aligning them properly.

After the modification of the feature maps from the input image and the support images, the feature maps are passed to the Cross Convolution Decoder, where the feature map from the input image and the feature map from the support images are fully fused with each other. Additionally, the Cross Convolution Decoder gradually restores the spatial resolution of the feature map to match that of the original image.

In detail, our model concatenates the query image as well as the N images from the support set with their labels along the feature dimension to apply cross convolution. We design the decoder block with a cross convolution block and a convolution block, where the cross convolution process is defined as

$$\text{CrossConv}(x_t, S; \theta_z) = \{\text{Conv}(x_t || s^i; \theta_z)\}_{i=1}^n, \quad (1)$$

where x_t, S respectively refer to the target and the support set $S = \{x_s^i || y_s^i\}_{i=1}^n = \{s^i\}_{i=1}^n$, and $||$ is the concatenation

operation along the feature dimension and $\mathrm{Conv}(x;\theta_z)$ is a convolutional layer with learnable parameters θ_z. The result then goes through a non-linear activation function, where we apply LeakyReLU in practical. Then we update the target feature map $\tilde{x_t}$ by taking average of the results of cross convolution with the non-linear activation. Denote $\mathrm{A}(x)$ as the non-linear activation, so the output of the cross convolution block is $\tilde{x_t} = \frac{1}{N}\sum_{i=1}^{n}\mathrm{A}(\mathrm{CrossConv}(x_t, S; \theta_z))$, and $\tilde{S} = \mathrm{A}(\mathrm{CrossConv}(x_t, S; \theta_z))$.

As is considered in the Cross Convolution Decoder, the results of the cross convolution block are then forwarded into a convolution block, where $\tilde{x_t}$ and $\tilde{S}$ are respectively updated with $\mathrm{A}(\mathrm{Conv}(\tilde{x_t};\theta_u))$ and $\mathrm{A}(\mathrm{Conv}(\tilde{S};\theta_v))$, with two learnable parameters θ_u and θ_v. The Cross Convolution Decoder enables the representations of each support set entry and query to interact with the others through their average representation, and facilitates variably sized support sets.

At the end of the Cross Convolution Decoder, we employ a convolution layer to map the resulting feature map to the predicted semantic segmentation of the input image, denoted as $\hat{y}_t^i$. The difference between the predicted semantic segmentation and the annotated label y_t^i can be used to supervise the network. By comparing the predicted segmentation with the annotated label, we can guide the model to improve its segmentation performance.

3.3. Modification of model

According to some experiments that we have conducted on the original model, we found that the results generated by the model is not good. After the discussion, we thought that the reason why the results generated by the model is frustrated may be that the model is too complex for it to learn segmentation. In the meantime, the dataset is also poorly annotated, which further hinder the learning of our model. The poor annotation of the dataset will be fully depicted and illustrated in Section 4.2.

Due to the stated problems above, we have modified the structure and the pipeline of our model, the refined model is shown in Figure 2. For the new pipeline, the input for the Diffusion Model only contains the images x_t^i in the target set, instead of containing not only the images in the target set but also the images in the support set. The feature map of the images in the target set is passed to the decoder for segmentation.

Comparing to our original model, we majorly retained the structure of it except several modifications on the decoder. We still utilize the encoder of the diffusion model, but we reconstruct the structure of the decoder in the segmentation part, which is formerly constructed with several Cross Convolution Decoder blocks. Instead, we followed the U-Net structure of the diffusion model, and attach support embedding to every time step, which pass the feature

Figure 3. Some bad annotations in COCO dataset. In the left picture, the base of the clock is separated from the clock and wrongly shared same annotation with the object aside; In the middle picture, the top part of the sign is missing; In the right picture, in addition to human labeling, other labeling is fragmented.

of support sets along time steps. This allows the model to capture the dynamics of the support set through diffusion process, and to make predictions about the future spread of information based on the past history of the process. In the process of support, our model still take advantage of cross convolution to combine information from the input image and the set of support images and labels.

The forward process of support embedding resembles that of the original Cross Convolution Decoder block. We concatenate the target image x_t with both the image $\{x_s^i\}_{i=1}^N$ and the label of the support set $\{y_s^i\}_{i=1}^N$ along the channel dimension. We practically apply the Cross Convolution block constructed in the original model here, just as the cross convolution declared with equation 1. However, we only keep the averaged cross convolution results as an output, i.e. $\tilde{x_t}$ defined in Section 3.2, and this is followed by a SiLU as a non-linear activation. Then the output is forwarded into a convolution as it was in the original model. The result is then passed through an adaptive average pooling layer, and a fully connected layer to produce the final support embedding.

We remove the SENet block as well as the FiLM block in our original model as you can tell from the refined pipeline of our model. Considering the refined pipeline of our model, the support images are concatenated to their labels naively, since there is naturally no channel-wise misalignment nor spacial misalignment between the image and its label. So we remove the FiLM block which is applied to address the problem of misalignment given the label and the support feature from the encoder. Regarding the SENet, which is a correspondence to FiLM that gives channel-wise weighting of target feature map after the target image passes encoder. In the original model, we have to apply the SENet and FiLM blocks to align those features and labels, so as to perform concatenation in the process of cross convolution. While in the modified model, we do not have to apply these blocks since the images and the label can be stacked to concatenation in nature.

Figure 4. Some pictures of objects which are labeled as road sign.

4. Experiment

4.1. Dataset

We train and evaluate our method on the **COCO Dataset** [13]. It is a widely-used benchmark dataset for object detection, segmentation and captioning tasks in the computer vision community. The dataset consists of over 120k images with 181 object categories in the training set. These object categories include commonly recognized everyday objects such as vehicles, animals, and household items, as well as more specific categories such as sports equipment and musical instruments.

4.2. Our first approach

Our initial approach did not yield the expected results and the model failed to converge. Upon closer examination of both the method and the dataset, we identified two main reasons for this.

Inadequate annotation We found that the COCO annotations were not as accurate as we had originally thought. As illustrated in Figure 3, there were numerous issues such as incorrect and missing annotations, which greatly impacted the training process.

Vast diversity within category We encountered the challenge of significant diversity within the same object category. Figure 4 highlights this issue, wherein objects labeled as street signs exhibited variations in their positions and shapes. This posed a substantial challenge for the training model.

4.3. Fine-tuning on dataset

When it comes to improving the performance of our model, there are several crucial steps that we took. Firstly, we focused on enhancing our model architecture, which we have mentioned in Section 3.3. Equally important was the cleaning of our data set, which we did through two major steps.

To limit the scope of our data set, we narrowed down our categories to only include animals and humans. While this may seem like a significant reduction, it has actually streamlined our data set and helped to remove any noise that may have existed previously. As a result, we now have a more targeted and specific set of categories that we can train our model on.

The second step was to delete any data with less than 10% label area from our data set. This was a strategic move to ensure that we only include high-quality data that is suitable for the training of our model. We found that the annotation of our data set was not always accurate, and by using this threshold, we were able to exclude any images that may have had insufficient information to be useful for learning.

Although we understand that these steps have reduced the diversity of our data set to a certain extent, we believe that they have actually improved the stability of our model in the long run. Additionally, it's worth noting that having a limited set of categories can actually encourage our model to make more accurate decisions, especially when dealing with similar categories. Overall, this process has been key in helping us to optimize our model for maximum performance.

During the fine-tuning process, we conducted experiments to determine the optimal number of images required for the pretrained model to effectively segment a novel category. Surprisingly, we found that a modest dataset of just 200 images was sufficient for this purpose. The training time for each iteration was remarkably fast, taking only approximately 1 minute when utilizing a single NVIDIA RTX 2060 GPU. In the subsequent sections, we will delve into the comprehensive results obtained from these experiments, shedding light on the efficacy of our approach.

4.4. Ablation experiment and result

To evaluate the effectiveness of our method, we trained our model on 4×Nvidia Tesla A40 using the streamlined dataset. We have also visualized the results, which are depicted in Figure 5. We also conducted an ablation experiment to gain further insight into the presence and location of the support embedding, and how it affects the model. For this purpose, we designed three different settings.

With Support (Ours): The support set ($\{x_s^i\}_1^N$ where $N = 16$) is passed to the Support Embedding part to get the embedding for the decoder.

Without Support: The support set is removed, the embedding vector passed to the decoder is set as constant.

From Scratch: Both the DDPM and the decoder are trained from scratch, no pre-trained weights is loaded.

We then calculated the mean intersection over union (MIOU) under each of these conditions, and the results are summarized in Table 1. We found that the model performs best in the With Support condition, indicating the importance of the support embedding in obtaining good results. It also demonstrates that the incorporation of the Support Set and the Support Embedding block significantly enhances the learning capabilities of the model. This improvement is not only observed in the pre-trained category but also extends to the categories that undergo fine-tuning. Such findings highlight the remarkable effectiveness and excep-

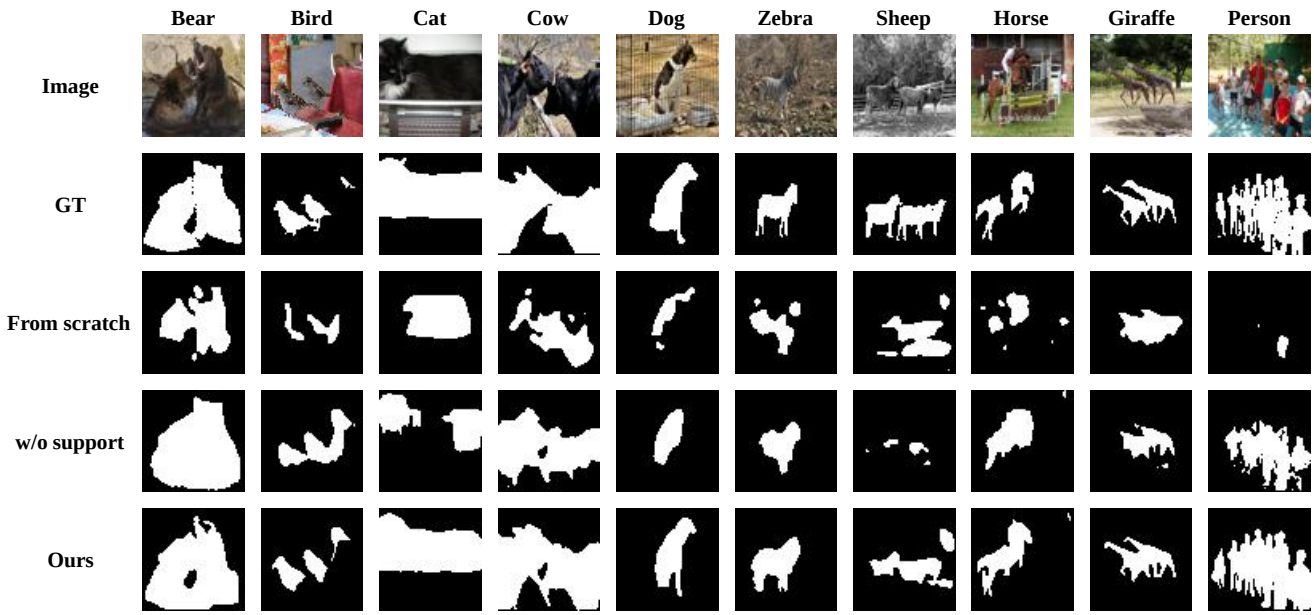

Figure 5. Some visualizations of the result. It is clear that our method can correctly identify and segment objects.

tional capability of our Support Embedding block in facilitating the learning process. However, we also found that the DDPM still requires pre-trained weights to perform well.

Categories	With Support	Without Support	From Scratch
Person	0.77	0.59	0.23
Bird	0.70	0.65	0.30
Cat	0.75	0.68	0.37
Dog	0.71	0.58	0.18
Horse	0.70	0.61	0.31
Sheep	0.70	0.58	0.37
Cow	0.72	0.63	0.44
Elephant	0.74	0.70	0.52
Bear	0.78	0.76	0.47
Zebra	0.77	0.73	0.49
Giraffe	0.71	0.60	0.49

Table 1. Mean intersection over union for different conditions

5. Conclusion

In this project, we thoroughly investigated the potential of the model with the Diffusion Model as the backbone in the challenging task of Few-shot Semantic Segmentation. Throughout our exploration, we developed and presented a novel model along with its enhanced version. The initial iteration of our model combined the Diffusion Model with a complex decoder for segmentation, but unfortunately, it yielded unsatisfactory results primarily due to the model's high complexity and the limited quality of the annotated dataset available.

To address these issues, we further refined our model and introduced its improved version. This enhanced model incorporated the Diffusion Model with a simpler yet meticulously designed decoder specifically tailored for few-shot semantic segmentation. The results obtained with the improved model demonstrated its enhanced capability in tackling the few-shot semantic segmentation task. This success was achieved despite the considerable challenges we encountered in modifying the model and the underlying pipeline.

Undoubtedly, this project and exploration provided us with invaluable insights and experience in the realm of few-shot semantic segmentation. We gained a deeper understanding of the intricacies involved in this task, including the importance of model design and the influence of dataset quality. Our endeavor allowed us to refine our taste and expertise in the field, paving the way for further advancements in few-shot semantic segmentation.

6. Contribution List

- **Bingnan Li** Multilevel dataset construction and data sieving. Distributed Data Parallel Training. Integrate Diffusion Model and Segmentor block.
- **Yifan Qin** Cross Convolution Block construction. SENet and FiLM Block construction in original model. Integrate Diffusion Model and Segmentor block.
- **Yan Zeng** Naive U-Net segmentation, where we conduct the early experimental feasibility verification. Re-

sults collection and verification. Related work research.

- **Haoyuan Tian** Cross Convolution Block construction. Convolution Block in original model. Related Work research about diffusion, fine-tuning, and cross convolution.

- **Shuhao Zhang** Related Work research. Results Collection and Classification. Workflow visualization.

References

[1] Victor Ion Butoi*, Jose Javier Gonzalez Ortiz*, Tianyu Ma, Mert R. Sabuncu, John Guttag, and Adrian V. Dalca. Universeg: Universal medical image segmentation. *arXiv:2304.06131*, 2023. 1

[2] Fabio Cermelli, Massimiliano Mancini, Yongqin Xian, Zeynep Akata, and Barbara Caputo. Prototype-based incremental few-shot semantic segmentation. *arXiv preprint arXiv:2012.01415*, 2020. 3

[3] Yuan Cheng, Yuchao Yang, Hai-Bao Chen, Ngai Wong, and Hao Yu. S3-net: A fast scene understanding network by single-shot segmentation for autonomous driving. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 12:1 – 19, 2021. 3

[4] Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. *Advances in neural information processing systems*, 28, 2015. 3

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. 4

[6] Nanqing Dong and Eric P. Xing. Few-shot semantic segmentation with prototype learning. In *British Machine Vision Conference*, 2018. 3

[7] Di Feng, Christian Haase-Schütz, Lars Rosenbaum, Heinz Hertlein, Claudius Glaeser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(3):1341–1360, 2020. 1, 3

[8] Yong Feng, Yonghuai Wang, Honghe Li, Ming Qu, and Jinzhu Yang. Learning what and where to segment: A new perspective on medical image few-shot segmentation. *Medical image analysis*, 87:102834, 2023. 3

[9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. 1, 2

[10] Markus Hofmarcher, Thomas Unterthiner, José Arjona-Medina, Günter Klambauer, Sepp Hochreiter, and Bernhard Nessler. Visual scene understanding for autonomous driving using semantic segmentation. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 285–296, 2019. 1

[11] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 1, 4

[12] Gen Li, V. Jampani, Laura Sevilla-Lara, Deqing Sun, Jonghyun Kim, and Joongkyu Kim. Adaptive prototype learning and allocation for few-shot segmentation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8330–8339, 2021. 3

[13] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 1, 6

[14] Xiaozhen Liu, Yunzhou Zhang, and Dexing Shan. Unseen object few-shot semantic segmentation for robotic grasping. *IEEE Robotics and Automation Letters*, 8:320–327, 2023. 3

[15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation, 2015. 3

[16] Zhihe Lu, Sen He, Xiatian Zhu, Li Zhang, Yi-Zhe Song, and Tao Xiang. Simpler is better: Few-shot semantic segmentation with classifier weight transformer. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8721–8730, 2021. 3

[17] Jilin Mei, Junbao Zhou, and Yu Hu. Few-shot 3d lidar semantic segmentation for autonomous driving. *ArXiv*, abs/2302.08785, 2023. 3

[18] Juhong Min, Dahyun Kang, and Minsu Cho. Hypercorrelation squeeze for few-shot segmenation. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6921–6932, 2021. 3

[19] Cheng Ouyang, Carlo Biffi, Chen Chen, Turkay Kart, Huaqi Qiu, and Daniel Rueckert. Self-supervised learning for few-shot medical image segmentation. *IEEE Transactions on Medical Imaging*, 41:1837–1848, 2022. 3

[20] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015. 2

[21] Zhuotao Tian, Hengshuang Zhao, Michelle Shu, Zhicheng Yang, Ruiyu Li, and Jiaya Jia. Prior guided feature enrichment network for few-shot segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44:1050–1065, 2020. 3

[22] Oriol Vinyals, Charles Blundell, Timothy P. Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. *ArXiv*, abs/1606.04080, 2016. 3

[23] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmentation with prototype alignment. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9196–9205, 2019. 3

[24] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks, 2013. 3

[25] Chi Zhang, Guosheng Lin, Fayao Liu, Jiushuang Guo, Qingyao Wu, and Rui Yao. Pyramid graph networks with connection attentions for region-based one-shot semantic segmentation. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9586–9594, 2019. 3

[26] Gengwei Zhang, Guoliang Kang, Yunchao Wei, and Yi Yang. Few-shot segmentation via cycle-consistent trans-

former. In *Neural Information Processing Systems*, 2021. 3

[27] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4*, pages 3–11. Springer, 2018. 1