# Interactive Few-Shot Segmentation (iFSS): Bridging Interactive Segmentation and Few-Shot Segmentation for Efficient Object Masking

Gejapati Tanmay
*Computer Science and Engineering*
*Indian Institute of Technology, Bombay*
Mumbay, India
tanmaygejapati@cse.iitb.ac.in

*Abstract*—**This work addresses the challenge of interactive few-shot segmentation (iFSS), a novel framework that combines the benefits of interactive segmentation (IS) and few-shot segmentation (FSS) while mitigating their limitations. IS typically requires significant user input to generate accurate object masks, while FSS allows segmentation of unseen object classes with minimal labeled examples. The proposed iFSS approach leverages interactive segmentation to efficiently generate support sets, which are then used in an FSS pipeline to segment new object classes. Experiments were conducted on the PascalVOC-5i dataset, comparing the baseline iFSS approach and an improved version incorporating iterative refinement from the RITM model. Results indicate that iFSS, particularly with iterative refinement, significantly improves segmentation accuracy for both support and query images. Insights from these experiments suggest that while mIoU does not always increase with clicks for query images, the query network benefits from improved support. This work highlights potential areas for future improvements, such as better information transfer techniques and incorporating more advanced segmentation models.**

*Index Terms*—**Interactive segmentation, few-shot segmentation, PascalVOC-5i, iterative refinement.**

## I. Introduction

This work aims to bridge two important areas of image segmentation: **interactive segmentation (IS)** and **few-shot segmentation (FSS)**.

**Interactive segmentation (IS)** enables users to iteratively refine segmentation masks of objects in images by providing feedback, typically through clicks or strokes. State-of-the-art methods in this domain, such as BRS [7], f-BRS [8], and RITM [1], inspire our approach, as they demonstrate how user input can significantly improve segmentation accuracy. However, while IS is fast, it often requires many user interactions to achieve dense segmentations, especially for complex datasets.

**Few-shot segmentation (FSS)**, on the other hand, relies on a small set of labeled examples (support images) to segment new instances of the same class in a query image. This method is grounded in prototypical meta-learning, where the support set generates prototype embeddings for a specific object class, allowing the model to generalize to unseen samples. FSS methods, such as PFENet, show that segmentation can be performed with just a few labeled examples, but the challenge remains in acquiring the required dense masks for support images, which is time-consuming.

Both IS and FSS reduce the time and effort needed to adapt segmentation models to new, out-of-distribution images. For instance, IS can segment new classes with minimal user input (as few as 20 clicks), reducing the need for dense data annotation. Similarly, FSS can segment objects with minimal labeled examples, thus alleviating the need for large datasets. However, both methods still require significant manual effort in generating support sets or refining segmentations for entire datasets.

This work introduces **Interactive Few-Shot Segmentation (iFSS)**, a novel approach that combines the strengths of both IS and FSS while minimizing their limitations. Specifically, we propose using interactive segmentation to quickly generate a small set of support images for FSS, thereby reducing the need for dense, fully-annotated support masks. This approach not only decreases the time required to generate support sets but also improves the segmentation accuracy compared to traditional IS methods. By leveraging minimal user input, iFSS significantly reduces the overall data annotation effort, making it particularly useful for time-sensitive applications in fields such as medical imaging and remote sensing.

In medical imaging, for example, IS can rapidly segment new medical images with minimal user input, while FSS can be used to apply this segmentation across a larger set of similar images. This combination allows for faster and more efficient analysis, especially for rare diseases where obtaining labeled data is costly and time-consuming. Additionally, iFSS can be extended to semi-supervised learning scenarios, where only a small portion of a class is interactively segmented, and the remaining images are automatically processed and refined as needed.

By bridging these two approaches, our method paves the way for more efficient segmentation pipelines that require less data, fewer annotations, and less time to train on new classes or datasets. This work has the potential to significantly improve applications in medical imaging, remote sensing, and other domains where segmentation tasks are both critical and data-scarce.

## II. Related Work

The following three approaches are the major inspiration for this work. Their brief account is given below.

### A. RITM (Refinement through Iterative Temporal Masking)

The **RITM** (Refinement through Iterative Temporal Masking) method [1] is a notable approach within the interactive segmentation (IS) domain. It enhances the quality of segmentation by incorporating iterative refinement, which refines segmentation masks progressively based on user feedback. RITM improves upon traditional IS models by leveraging an iterative backpropagation refinement scheme (f-BRS), which allows the model to incrementally update its predictions through multiple feedback cycles. This approach uses a stronger backbone architecture, such as HRNet, which excels at capturing fine-grained details, particularly the boundaries of objects. By iteratively refining the segmentation, RITM can handle more complex segmentation tasks and generate more accurate masks with fewer user interactions compared to earlier methods.

### B. PFENet (Prototypical Few-Shot Segmentation Network)

**PFENet** [2] addresses the challenge of few-shot segmentation (FSS), where the goal is to segment objects from novel classes using only a few labeled examples. PFENet improves FSS performance by effectively learning class prototypes from the support set, which are used to guide the segmentation of the query images. The key idea behind PFENet is to form a prototype for each object class by averaging feature representations of the support set images. This prototype is then used as a reference to segment the query image, allowing the model to generalize to unseen instances of the same class with minimal labeled data.

### C. IFSENet (Interactive Few-Shot Segmentation Network)

The **IFSENet** framework combines the strengths of both interactive segmentation (IS) and few-shot segmentation (FSS) [3]. In IFSENet, user interactions are used to quickly generate a small set of labeled support images, which are then used to segment novel objects in the query images. This method minimizes the data annotation effort while maintaining high segmentation accuracy for new object classes. The architecture of IFSENet uses a refined IS model like RITM to generate the support set and employs the prototypical learning approach from PFENet to generalize to unseen classes. Additionally, IFSENet incorporates a more efficient information transfer mechanism between the support set and query set, allowing the query path to benefit from the prototypes derived from the support set. This hybrid approach significantly reduces the time and effort required to adapt segmentation models to new classes, making it particularly useful in data-scarce domains such as medical imaging and remote sensing. The IFSENet approach thus represents an important step toward making segmentation tasks faster, more efficient, and less reliant on large datasets.

The above work is carried forward but this work and we aim to build a stronger and more robust iFSS model.

## III. Proposed Approach

### A. Fundamentals of IS and iFSS

*Interactive Segmentation*

An IS model works by taking in user-defined positive and negative clicks, where the objective is to segment out the object indicated by the positive clicks, and the negative clicks help the model refine the segmentation mask. The input to an IS model includes:

- **Click masks**, which are circles of radius that decrease with each iteration of clicks, helping the model obtain detailed boundaries.
- **Segmentation output of the previous iteration**, where zeros are used for the first iteration.
- **The image itself**, with augmentations to help generalize better.

These inputs are processed by the encoder, and the decoder generates a binary segmentation mask. Some methods concatenate these inputs to form 6 channels, then increase the number of channels and squeeze them down to 3, which is typically the input dimension of standard segmentation models. Other methods change the initial layer of the encoder to relearn the mapping of such inputs. These methods have varied results, but the latter is typically more effective according to [1].

*Few-Shot Segmentation*

A few-shot semantic segmentation system involves two sets: the *query set* $Q$ and the *support set* $S$. Given $K$ samples from the support set $S$, the goal is to segment the area of an unseen class $C_{\text{test}}$ from each query image $I_Q$ in the query set.

Models are trained on classes $C_{\text{train}}$ (base) and tested on previously unseen classes $C_{\text{test}}$ (novel) in episodes where $C_{\text{train}} \cap C_{\text{test}} = \emptyset$. The episode paradigm was proposed in [4] and first applied to few-shot segmentation in [5]. Each episode is formed by a support set $S$ and a query set $Q$ of the same class $c$. The support set $S$ consists of $K$ samples $S = \{S_1, S_2, \ldots, S_K\}$ of class $c$, which is known as a 'K-shot scenario'. The $i$-th support sample $S_i$ is a pair $\{I_{S_i}, M_{S_i}\}$, where $I_{S_i}$ is the support image and $M_{S_i}$ is the support mask of class $c$, respectively.

For the query set, $Q = \{I_Q, M_Q\}$, where $I_Q$ is the input query image and $M_Q$ is the ground truth mask of class $c$. The query-support pair $\{I_Q, S\} = \{I_Q, I_{S_1}, M_{S_1}, I_{S_2}, M_{S_2}, \ldots, I_{S_K}, M_{S_K}\}$ forms the input data batch to the model. The ground truth $M_Q$ of the query image is invisible to the model and is only used to evaluate the prediction on the query image in each episode.

### B. Interactive Few-Shot Segmentation

The principles of training an iFSS model are grounded in the prototypical networks used in FSS models. The task now boils down to efficiently obtaining information from interactive segmentation for the support set. For this, we first implement a baseline approach and then improve upon it by enhancing the IS over the support and better prototypes for the query set.

*The Baseline iFSS Approach:* In this approach, the support set is trained by an IS network, and simultaneously, the query network is being trained using prototypes from the IS network. Some simple design choices used for this approach are as follows:

- DeepLabV3 segmentation model is used.
- Different encoders are employed, as only the support set has access to click supervision.
- Inspired by RITM [1], the input channels are expanded from 6 to 8 and then reduced back to 3 by using 'conv1x1' layers and passed to the DeepLab encoder.
- Inspired by PFENet [2], the outputs from the middle layer of the encoder are used from the support set to form the prototype by global average pooling, which gives a $1 \times 1 \times C_{\text{mid}}$, where $C_{\text{mid}}$ is the number of channels from the middle layer of the encoder. This is then expanded by duplication and concatenated with the query features, before being passed through a decoder.
- The PascalVOC-5i dataset is used with its standard train-val split, and CCE loss is applied for training.

*iFSS-RITM:* The above approach was improved by using a stronger interactive segmentation model, RITM [1]. The key features of this model are as follows:

- Uses a stronger backbone: HRNet [10], which is better at capturing object boundaries.
- Iterative refinement at test time using the *Backpropagation Refinement Scheme (BRS)* [7] [8].

f-BRS refines segmentation by iterating over multiple steps, where the model's output at each iteration is used as guidance to update and improve the mask in the next step. It leverages feedback from previous iterations to iteratively correct errors and refine the segmentation, especially focusing on fine details like object boundaries, through backpropagated gradients.

On top of RITM, an FSS pipeline was built, which involved re-writing the data loader, writing a separate query path, and implementing changes typical for an FSS pipeline, similar to the baseline approach.

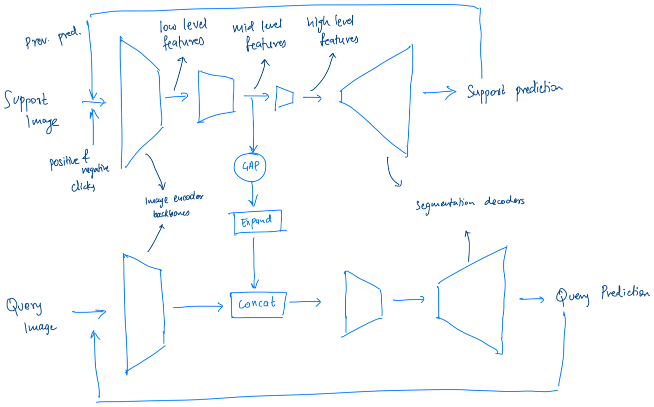

Fig. 1: Model Architecture

## IV. Results and Insights

### A. Baseline iFSS Results

| Clicks | IoU (Q) | IoU (S) |
|---|---|---|
| 1 | 27.47% | 48.27% |
| 2 | 28.15% | 59.82% |
| 3 | 28.27% | 65.95% |
| 5 | 27.67% | 71.55% |
| 10 | 27.61% | 75.62% |
| 20 | 27.71% | 79.42% |

TABLE I: Baseline iFSS Results for Query (Q) and Support (S) IoU at various clicks.

### B. iFSS-RITM Results

The training is conducted in two steps: IS-pretraining and iFSS training. Initially, the support path is trained to segment well (half-way through the full training to prevent overfitting in the latter stage), followed by joint training of both support and query paths.

| Clicks | IoU (S) |
|---|---|
| 1 | 55.33% |
| 2 | 68.09% |
| 3 | 73.84% |
| 5 | 78.95% |
| 10 | 84.60% |
| 20 | 88.36% |

TABLE II: Pretraining Results for IoU on the Support Path.

| Clicks | IoU (Q) | IoU (S) |
|---|---|---|
| 1 | 42.62% | 58.16% |
| 2 | 42.98% | 69.98% |
| 3 | 43.06% | 76.25% |
| 5 | 42.86% | 81.87% |
| 10 | 42.87% | 86.28% |
| 20 | 42.88% | 89.05% |

TABLE III: iFSS Trained Results for Query and Support IoU at various clicks.

| . | NoC@85% | NoC@90% | $\geq$ 20@90% | SPC |
|---|---|---|---|---|
| Q | 17.01 | 18.64 | 413 | 0.081 |
| S | 7.38 | 9.82 | 153 | 0.081 |
| f-BRS [8] | 5.51 | 8.58 | | |

TABLE IV: Additional metrics: NoC stands for Number of Clicks, SPC stands for Seconds per Click.

### C. Insights

- The mIoU for the query path (Q) does not significantly increase with the number of clicks, which suggests that iterations in the query path are not being used effectively.
- The NoC of our training and the f-BRS [8] approach are quite close, but ours is slightly worse, likely due to conflicting gradients from the query path.
- The query NoC@90% is 18, which is comparable to results from the pre-deep learning era in interactive segmentation (e.g., graph cuts, geodesic matting, and random walkers) and has plenty of room for improvement.

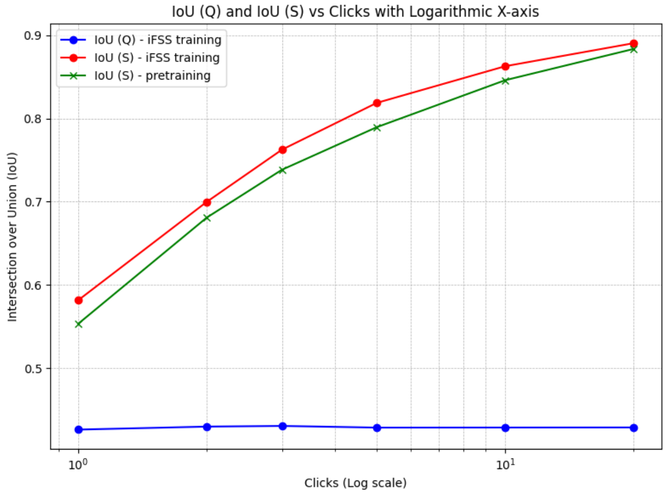

Fig. 2: IoU across clicks seems to increase for support set but not for query set.

## V. Conclusions

The results clearly show a lack of effective learning over the iterations, and more work needs to be done over the following months to get the query results to approach *near-FSS* results in order to transform this work into a full-fledged research paper. Nonetheless, the objective of this research and development (RnD) was to conduct a literature survey and develop an initial understanding of Interactive Few-Shot Segmentation (iFSS) systems. We have successfully achieved this, as well as identified what doesn't work and the areas where we can focus future efforts.

### *Future Improvements*

There are many improvements that can be made to the current work, but it sets a strong foundation for an interactive segmentation model that refines at test time as well. This can be improved by enabling better information transfer between the support set and the query set by utilizing more advanced Few-Shot Segmentation (FSS) techniques, such as [9], [10], [11], and others. Additionally, powerful segmentation backbones, such as SAM [12], can also be explored. Some other ideas we aimed to try include:

- Generalizing to multiple support and query elements.
- Improving the performance of the model using active learning with a corpus of unlabeled images from a similar domain.

## References

[1] K. Sofiiuk, I. A. Petrov, and A. Konushin, ”Reviving Iterative Training with Mask Guidance for Interactive Segmentation,” *Proceedings of CVPR*, 2020.

[2] Z. Tian, H. Zhao, M. Shu, Z. Yang, R. Li, and J. Jia, ”PFENet: Prior Guided Feature Enrichment Network for Few-Shot Segmentation,” *Proceedings of CVPR*, 2020.

[3] S. Chandgothia, A. Sekhar, and A. Sethi, ”IFSENet: Harnessing Sparse Iterations for Interactive Few-shot Segmentation Excellence,” *Proceedings of CVPR*, 2022.

[4] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, ”Matching Networks for One Shot Learning,” *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[5] A. Shaban, S. Bansal, Z. Liu, I. Essa, and B. Boots, ”One-shot Learning for Semantic Segmentation,” *Proceedings of BMVC*, 2017.

[6] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao, ”Deep High-Resolution Representation Learning for Visual Recognition,” *Proceedings of CVPR*, 2020.

[7] W.-D. Jang and C.-S. Kim, ”Interactive Image Segmentation via Backpropagating Refinement Scheme,” *Harvard University*, 2024.

[8] K. Sofiiuk, I. Petrov, O. Barinova, and A. Konushin, ”Rethinking Backpropagating Refinement for Interactive Segmentation,” *Proceedings of ICCV*, 2021.

[9] C. Lang, G. Cheng, B. Tu, and J. Han, ”Learning What Not to Segment: A New Perspective on Few-Shot Segmentation,” *Proceedings of ICCV*, 2021.

[10] J. Min, D. Kang, and M. Cho, ”Hypercorrelation Squeeze for Few-Shot Segmentation,” *Proceedings of ICCV*, 2021.

[11] S. Hong, S. Cho, J. Nam, and S. Kim, ”Cost Aggregation is All You Need for Few-Shot Segmentation,” *Proceedings of CVPR*, 2020.

[12] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W. Lo, P. Dollár, and R. Girshick, ”Segment Anything,” *Proceedings of CVPR*, 2023.

# APPENDIX
## VISUALIZATIONS AND ERROR ANALYSIS

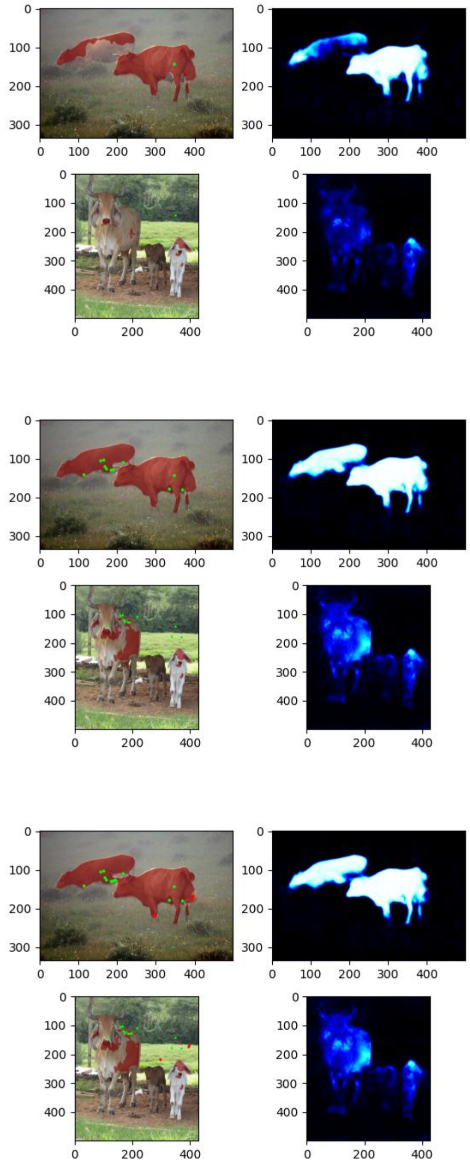

Fig. 3: Support and Query images (top and bottom) for 1st, 10th, and 20th clicks respectively.

Note: Please ignore the duplication of clicks on the support images, as they were not used to derive query results and were just an artifact.

The query path seems to improve gradually as the support improves in Fig. 3 and Fig. 4. However, the performance of the query path still heavily depends on the initial predictions from the query network.

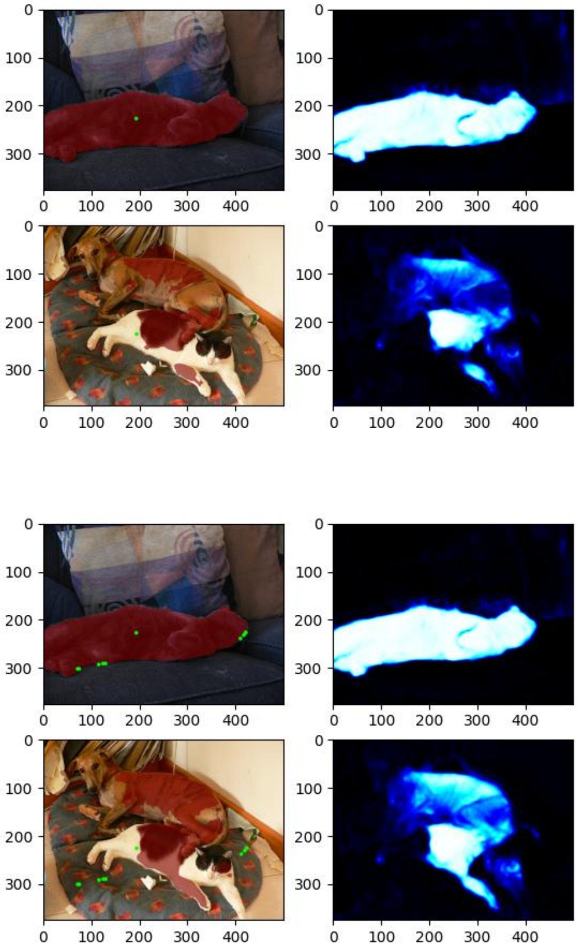

Fig. 4: Support and Query images for 1st and 10th clicks respectively.

### *Failure Cases*

Fig. 5 and Fig. 6 are failure cases where support network failed to converge or where the query network was unable to generalize.

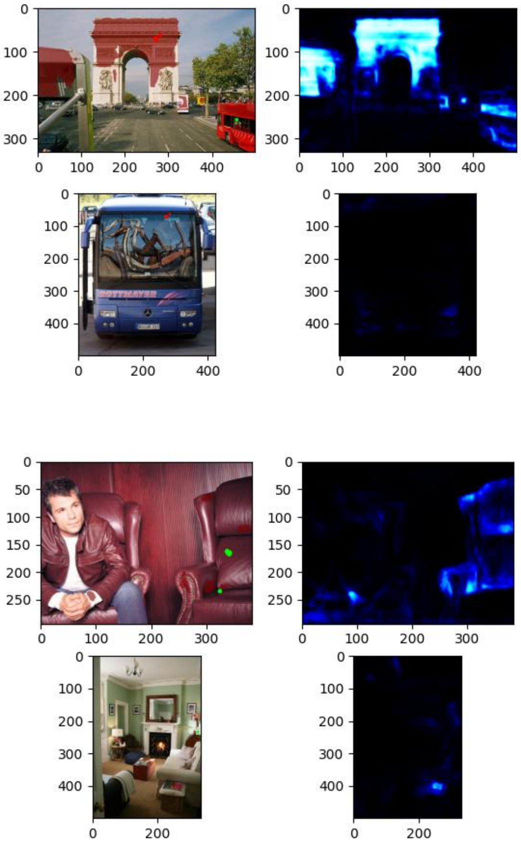

Fig. 5: Failure cases where the support network never converged.

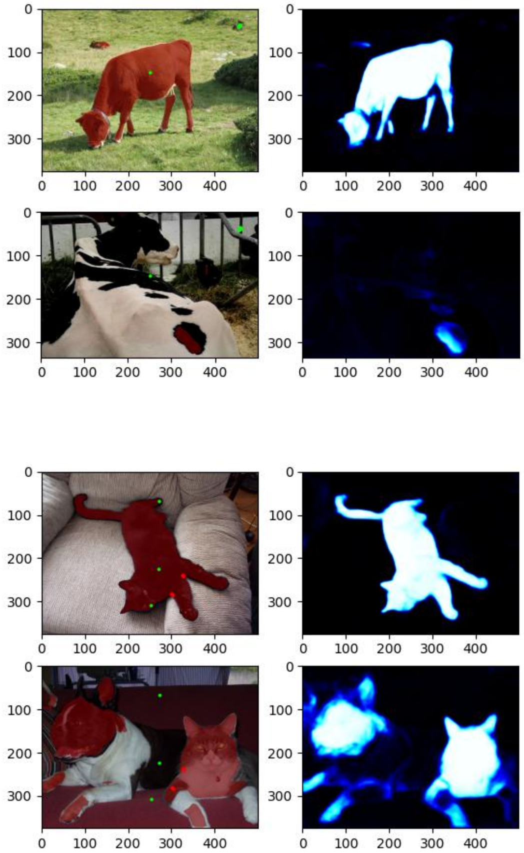

Fig. 6: Failure cases where the support network succeeded but the query network failed. The first case likely failed due to an inability to generalize the object to the cow class and overfitting to the color of the support cow. The same issue occurred in the latter case, suggesting that the model may be overfitting or relying too heavily on lower-level features.