



DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Guided Research

# Unsupervised Segmentation of Light Microscopy Images

**Baris Zöngür**





DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Guided Research

# Unsupervised Segmentation of Light Microscopy Images

Author: Baris Zöngür  
Examiner: Dr. Felix Dietrich  
Submission Date: 23/04/2023





I confirm that this guided research is my own work and I have documented all sources and material used.

Munich, 23/04/2023

Baris Zöngür

# Abstract

Pixel-wise semantic segmentation on different domains is a well-researched area that shows satisfactory results when supervision is provided. However, supervision for such methods is only available for a small subset of real-world scenarios. Lack of supervision for real-world scenarios requires new methods that can work directly on input images without supervision. Unsupervised segmentation methods provide a solution for such cases.

We experimented with a domain where supervision is not available. Our domain is the black-and-white microscopy images of Arbuscular mycorrhizal, a symbiotic relationship between soil fungi and vascular land plants. We segment the nutrition-transferring instances in the symbiotic structure inside the plants. We perform this segmentation with clustering on different embedding spaces that are extracted from the input images only. We use Visual Transformer-based self-supervised feature extractors to obtain a meaningful representation and cluster the pixels on the obtained embedding space. We show that our method outperforms the baseline method of color-based clustering. Finally, we trained a UNet model to experiment with different levels of supervision and compare how the fully unsupervised method performs compared to different levels of supervision.

# Contents

<b>Abstract</b>	<b>iii</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Related Work</b>	<b>3</b>
2.1 Previous Works on Segmentation . . . . .	3
2.2 Visual Transformer . . . . .	4
<b>3 Unsupervised Segmentation on Microscopy Images</b>	<b>5</b>
3.1 Data . . . . .	5
3.2 Clustering on Intensity Values . . . . .	7
3.3 Clustering on DINO Features . . . . .	10
3.4 Clustering on Fine Tuned Features . . . . .	13
3.5 Extension on Semi-Supervision . . . . .	17
3.6 Experiments and Results . . . . .	19
<b>4 Conclusion</b>	<b>28</b>
<b>List of Figures</b>	<b>29</b>
<b>List of Tables</b>	<b>30</b>
<b>Bibliography</b>	<b>31</b>

# 1 Introduction

In the last years, semantic segmentation has become a well-researched area due to its increase in performance and suitable application areas. As a result, semantic segmentation found use in various application areas such as agriculture[1], medicine[2], and robotics[3]. Datasets like ImageNet[4], CocoStuff[5], and CityScapes[6] provide an annotated dataset that can be used to train a segmentation model to be then inferred in various real-world applications. Many works use mentioned annotated datasets that can acquire a meaningful feature representation and multi-channel semantic segmentation[7] [8]. One of the most popular approaches in this area is to use Convolutional Neural Networks(CNN)[9] [10] [11] . However, most of these applications depend on a humanly annotated ground truth segmentation in a subset of the domain-specific databases, and inference on those domains is not feasible with an already existing annotated dataset. Due to the lack of annotated datasets in such domains, self-supervised learning(SSL) approaches became increasingly popular. SSL methods provide an application that can extract meaningful semantic segmentation without needing ground truth labels[12] [13] [14]. SSL methods generally depend on an SSL feature extractor that can extract a meaningful embedding space without supervision. These feature use already established models such as CNNs[15] or transformers[16]. Our domain is microscopy images containing Arbuscular mycorrhizal, a symbiotic relationship between soil fungi and vascular land plants[17]. Each plant contains a few microscopy images, which are taken with different configurations. For our dataset, we have no annotation provided. So, to obtain robust and meaningful segmentation with no ground truth, we turn to SSL feature extractor-based unsupervised segmentation methods.

SSL methods that use transformers are called Visual Transformers(ViT)[18]. Previous works showed that ViT methods could achieve and outperform CNN-based models in various experiments[14] [19]. The main advantage of ViT feature extractors is that they contain a piece of global information in the corresponding feature vector of each patch. This global information comes from the transformer-based structure of ViT-based models. Each patch transforms information with each other patch, unlike in CNN methods, which contain only neighborhood information in each lower-resolution feature vector. This information transfer results in the mentioned feature embeddings that contain global information. The SSL feature extractor that we use is the DINO model[19]. The DINO model is used as a backbone that works on various SSL embedding tasks for unsupervised segmentation. We also use the DINO model as our SSL feature extractor to extract a high-dimensional, meaningful representation for each image patch. This basis transformation gives us a more informative representation than the vanilla information from the image itself. We then can experiment with clustering with these extracted features to obtain a meaningful pixel-wise instance segmentation mask. Since we can extract feature representation for each image patch, we can acquire a more

detailed segmentation than bounding boxes.

We use two separate datasets to experiment with different methods for unsupervised segmentation. One dataset is authentic microscopy images of AMF. These black-and-white images are obtained by staining method with ink and vinegar[20], and corresponding microscopy images are recorded under the microscope. Another dataset that we use is a synthetic dataset that generates mimics the general structure of the original AMF images[21]. These synthetic images are RGB images. These images are rendered using Blender software[22]. We explain how we preprocess and use these data in section 3.1.

We first experiment with color-based clustering to set up a baseline for other methods. In these experiments, we only cluster with the averaged color intensity information of each image patch and obtain a single channel instance mask for the image. We explain the method in detail in section 3.2. Then we move on to clustering based on SSL features that we obtain using the DINO feature extractor in section 3.3. We also use PCA on obtained feature embeddings to reach a more refined embedding space. Then we cluster new vectors on the transformed basis with concatenated color information. In section 3.4, we show how we fine-tune the DINO model in order to obtain more meaningful feature vectors for our specific domain. We build a reconstruction decoder on top of the DINO model and fine-tune the model with a reconstruction loss. With these fine-tuned features, we can further increase the performance of our method. To compare our unsupervised method, we then experiment with different levels of supervision on synthetic data in section 3.5. We built a UNet to train with different levels of supervision and show the resulting extracted masks. To set up a meaningful baseline, we use a small subset of synthetic data for such experiments to match the number of images that are available in a real-world scenario. Finally, in section 3.6, we do quantitative and qualitative analysis and comparison of the mentioned methods and compare unsupervised segmentation with different levels of supervision.

## 2 Related Work

### 2.1 Previous Works on Segmentation

Image Segmentation can improve various applications in medicine[2], computer vision[23], and robotics[3]. In that regard, Image Segmentation has been researched extensively in the last decade. Various methods showed that a Deep Neural Network for the task achieves significant results[24] [8] [7]. Using mentioned methods in the Microscopy Image domain, "Light Microscopy Image Analysis using Neural Networks" proposes a U-Net model that uses synthetic images and respected ground truth annotations[21]. However, these supervised methods require an annotated dataset such as COCO[5] or Cityscapes[6] to minimize the error between generated segmentation mask and the ground truth mask. Therefore, mentioned constraints limit the usage of supervised methods in many cases.

On the other hand, self-supervised approaches show promising results while overcoming annotation constraints. Many self-supervised Image Segmentation methods showed that these methods could achieve comparable results[14] [19] [25] [26]. One of the common approaches in self-supervised segmentation is using SSL features due to their success and the emergence of segmentation capabilities[19]. Even though CNN-based methods have been state-of-the-art in computer vision tasks for many years, they cannot model long-range interactions. To overcome this issue, Visual Transformer(ViT)[18] models use a Transformer-based model, which has already proved successful in modeling the long-range interactions in NLP. Because of the properties of ViT models, they can be used as SSL feature extractors. DINO[19] is a ViT-based model that can generate semantically meaningful object segmentation. It generates features for each image patch containing local and global information.

Various approaches use SLL features for unsupervised segmentation tasks.[27] [14] One approach is to use feature similarity between different patches. They formulate the segmentation task as a patch or pixel wise partitioning task.[25] They are using a clustering approach on extracted features to partition each patch. SelfMask[14] evaluates the performance of different feature extractors by proposing a method that combines different features. STEGO[25] tunes the cluster of the entire database by using a student-teacher network and finetuning on DINO[19].

Another line of research focuses on Generative Models to achieve an instance or segmentation mask that can be used to generate samples during training[27] [28] [29] [30]. In these works, the mask is generally a byproduct of the model and is extracted from the generator during inference time. In the MOVE[27] model, an instance mask is upsampled from DINO[19] features and is used to create a novel image with various shifts. Generative models are also used in unsupervised domain adaption, aiming for a generalizable model trained with synthetic data to be used with real-world data. "Unsupervised Bidirectional

Cross-Modality Adaptation via Deeply Synergistic Image and Feature Alignment for Medical Image Segmentation"[31] propose a generative model that performs domain adaptation with style transfer from synthetic data to real-world data.

## 2.2 Visual Transformer

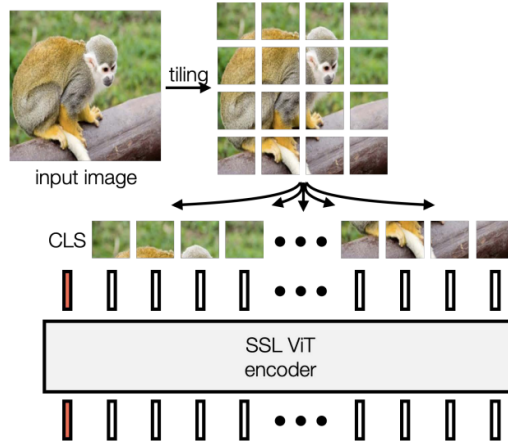


Figure 2.1: Visualization of ViT model based on the MOVE paper Figure 3.[27]

We are using a ViT-based feature extractor for our methods. Specifically, we are using the small version of the DINO[19] feature extractor, which contains 11 attention layers. In Visual transformer models, each image is processed as a separate sequence. Each image is divided into smaller patches, and an initial feature is extracted for each image patch. Each extracted feature represents a node in the sequence in the transformer blocks. A class token is also appended before the first transformer block. An example visualization of this method is in Figure 2.1. For an image size of  $(32 \times 32)$ , if the patch size is  $(8 \times 8)$ , 16 separate feature vectors are extracted during this process. This can also be seen as a 16-word sequence in NLP-based transformer models. For our example, with an embedding dimension of  $N$ , we have an input tensor of  $(16 \times N)$ . As mentioned before, a randomly initialized class token is also appended, giving us a tensor of  $(17 \times N)$ . This sequence is then processed with a series of transformer blocks, which transforms the initial basis of the color information of each patch into a more informative embedding space. Throughout this process, the locality of each patch is preserved, meaning that, for our example, 16 separate features are generated, and the first feature corresponds to the first patch of the image. Each generated feature contains both local information about the patch and, because of the transformer layers, it also contains global information about the entire image. This property makes these models suitable as a backbone for various operations. For example, a segmentation head can be built on top of the extracted features for performing a segmentation task. Also, a reconstruction head can be added, performing various tasks such as denoising.

## 3 Unsupervised Segmentation on Microscopy Images

We use various approaches that mainly focus on the feature similarity of pixels or patches. These methods require zero supervision and utilize the repeating feature patterns across the images. We aim to cluster the features in the embedding space and achieve a multi-channel segmentation mask where each channel represents a different cluster. We extract informative features from images to accomplish this goal so that each cluster center represents a distinctive set of features. Presented methods mainly vary in feature extraction to examine how informative each embedding space is. In the end, we compare the clustering performance in each embedding space, comparing different basis transformations and the informativeness of resulting manifolds.

### 3.1 Data

We are using the images obtained under a microscope of vascular land plants infected by fungi. Vascular land plants have lignified tissues for conducting water and minerals throughout the plant. Sections of the plants infected by fungi represent a symbiotic relationship between plants and fungi. This association is called Arbuscular Mycorrhiza Fungi (AMF)[17]. Fungi spores invade the host plant through branch-like structures called fungal hyphae. Fungi also form node-like structures in the plant, penetrating its walls, called arbuscules and vesicles. These node-like structures increase the nutrient transfer between the plant and fungi and store products. Plants that form this symbiotic relationship have grown in nutrition, fertility, and structural integrity. The combined nutrition-transferring structure of the fungi consisted of hyphae, arbuscules, and vesicles alongside the root cortex, if present, representing our instance mask for unsupervised segmentation tasks.



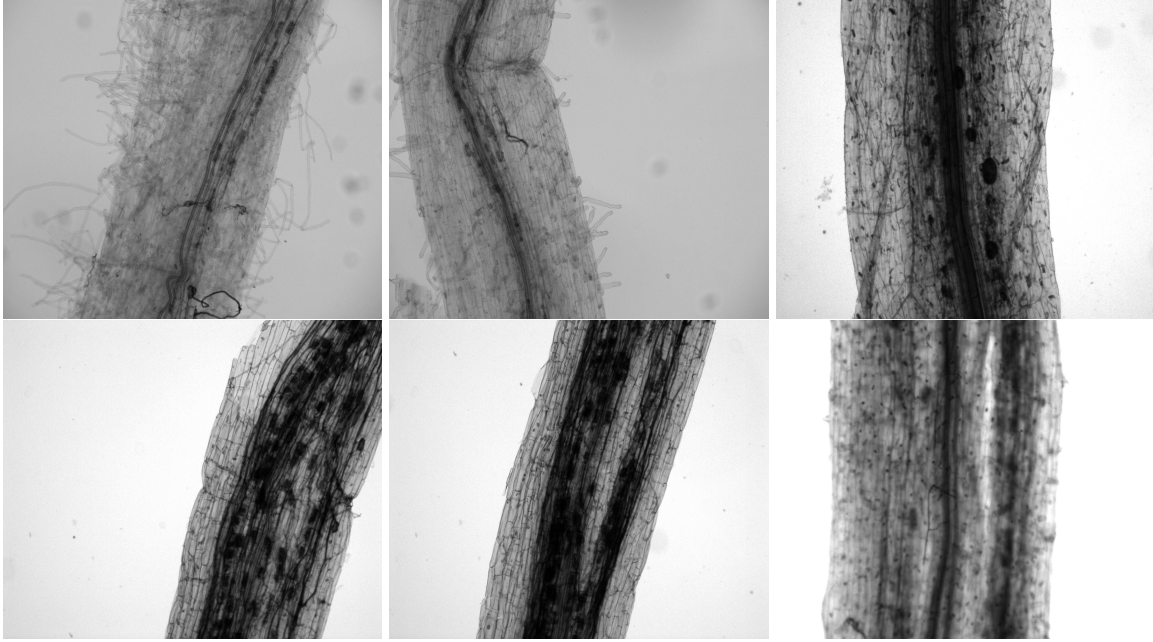


Figure 3.1: Example images of Microscopy Data

We have 21 black-and-white microscopy images, which Dr. Catarina Cardoso provides from the Gutjahr Lab. Examples from the dataset are provided in Figure 3.1. The original sizes of the images are  $1536 \times 2048$ . A channel size of 1 gives us a tensor of  $[1, 1536, 2048]$ . We resize the input image using bilinear interpolation to match the model dimensions. After the interpolation, the image size is  $[1600 \times 1920]$ , which results in a tensor of  $[1 \times 1600 \times 1920]$ . After that, we convert black-and-white data to RGB data by expanding on the color channel. 2nd and third channels are the replication of the initial channel. This operation gives each image a tensor of  $[3 \times 1600 \times 1920]$ . Then we normalize the image with the mean and standard deviation of the dataset. When color values are directly used to cluster, we use the mean and standard deviation of the microscopy data. However, when the pre-trained DINO[19] model is used, we normalize it by the mean and standard deviation of ImageNet[4]. As a last operation, we extract multiple smaller image patches from each image for training and inference. We extract images of size  $320 \times 320$ , which results in 30 sub-images for each image. This operation results in a tensor of  $[30 \times 3 \times 320 \times 320]$  for each image, each 30 sub-image representing a specific part of the image. Even though our proposed methods can be used with the mentioned data, it is unsuitable for a precise evaluation. The reason is that ground truth labels are not provided. To get an accurate quantitative evaluation, we use synthetic data from "Light Microscopy Image Analysis using Neural Networks." [21]

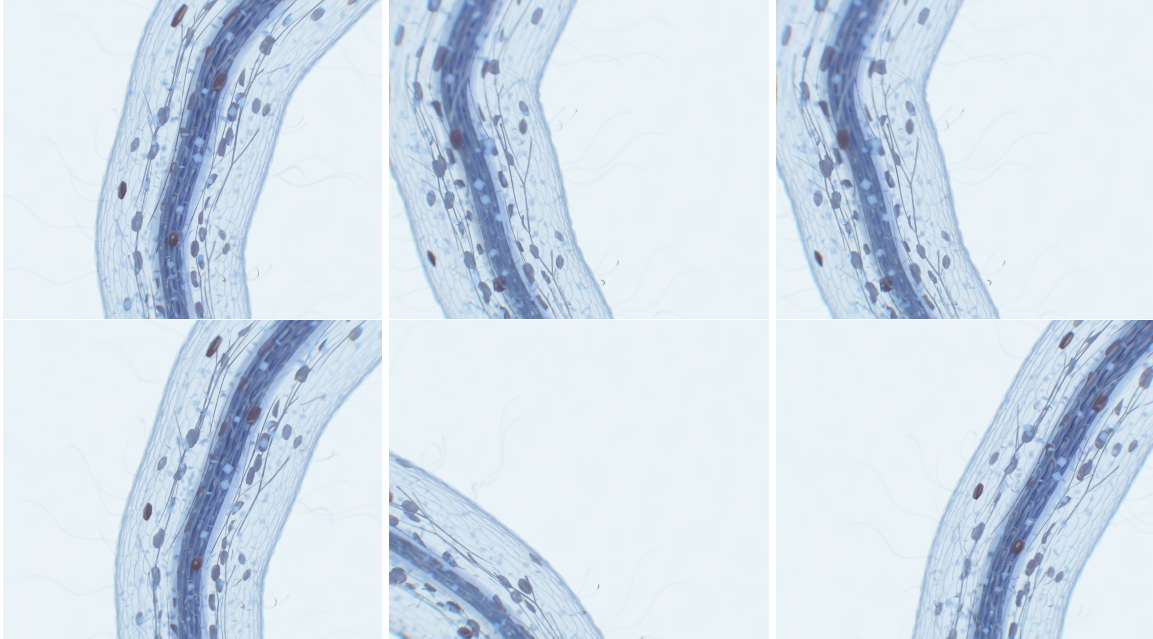


Figure 3.2: Example images of Synthetic Data

The synthetic data we are using is synthesized using Blender software. The 3D Model is an editable model which imitates the general structure of the Microscopy images. The synthetic images successfully represent the main instances of hyphae, arbuscules, and vesicles and the general structure of the plant. In addition, the viewing angle, the orientation of the 3D Model, and the degree of infection are all editable in the 3D Model. We randomize these properties to obtain 50 different images. This 3D Model also provides the ground truth labels for the image, which we use to evaluate our proposed models. A set of examples of Synthesized images are in Figure 3.2. A preprocessing of this data is also necessary. Different from the original images, this data has three color channels. And we directly render the images of size  $[3 \times 1600 \times 1920]$ . We then apply the same normalization operations, as normalizing with a mean and standard deviation of either synthetic dataset or ImageNet[4]. As the last operation, we apply the same sub-division, giving each image a tensor of  $[30 \times 3 \times 320 \times 320]$ . With image generation and preprocessing, we get similar data to the original microscopy images. Another aspect of this data generation is that it also augments our input data. We have 21 microscopy images, but we can synthesize a larger dataset with this method.

### 3.2 Clustering on Intensity Values

In many cases, directly using the L2-norm of the input vectors as a feature does not yield a reasonable representation. Mainly in real-world scenarios, one does not expect a repeating pattern in different instances of the same class based on their intensity values. These intensity values depend on external factors such as lighting, shading, and projection angle. Normalizing the input means shifting all vectors under the same distribution, but this does not counter

the effect of different random distributions of external impacts on each image. For example, separate instances of the same class captured during varying times of the day, clustered with intensity values, would fall under the distinct categories. However, norms and dot products of the extracted features are widely used while performing unsupervised segmentation. The transformed basis yields a space where the norm and dot product are informative operations. So, the informativeness of different functions is dependent on the manifold itself. In our case, images are low-channel tensors obtained after several physical transformations. When we examine the input space, we found that intensity values give distinctive information on certain parts of the image. This property is mainly due to normalizing the distribution of external effects during the physical transformations performed while obtaining the final images.

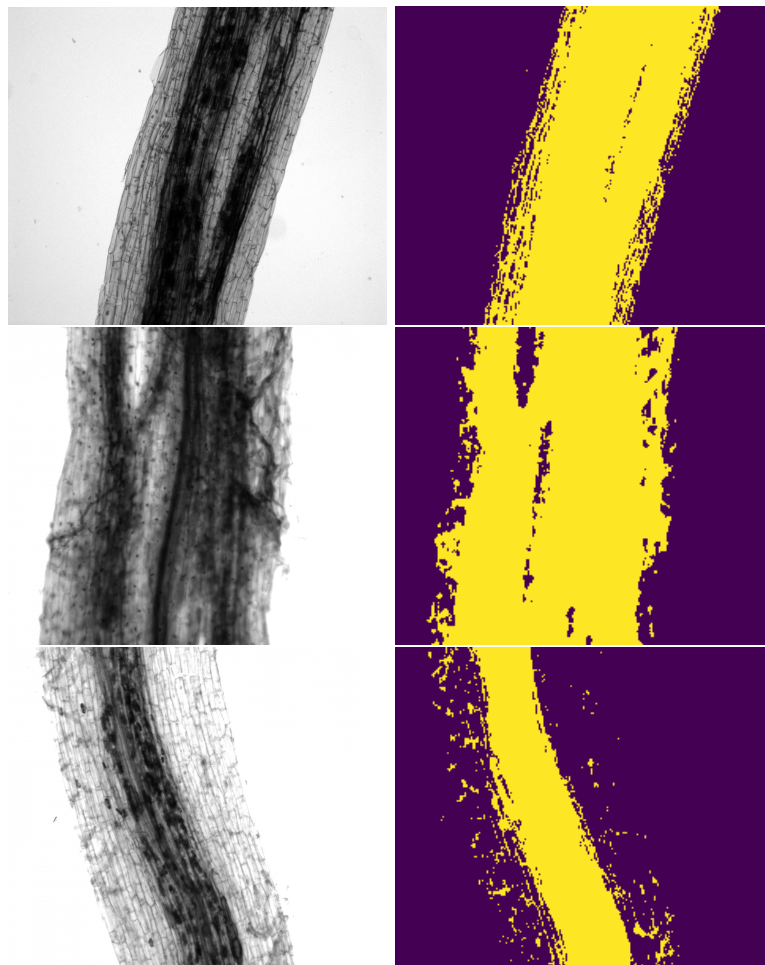


Figure 3.3: Visualization of Extracted Masks with Clustering on Color on Microscopy Images. (Original image on the left column, extracted mask on the right)

By using this already-normalized quantity of Microscopy Images, we can set up a baseline for our model. In this basic approach, we directly cluster each pixel or patch of images

by their color value. By clustering smaller patches in each (8 x 8) image, we obtain a less detailed but more robust segmentation. Also, we can directly compare with further patch-wise methods. With this method, we use k-nn[32] for each color value. We have a color vector for  $[H/m*W/m]$  number of patches.  $m=1$  for each pixel segmentation and  $m>1$  for patch-wise segmentation. We get  $n$  number of clusters after applying the k-nn[32] algorithm to each patch. The assigned class for each patch is the id of the cluster to which they belong. When we use  $k=2$ , we acquire two masks. One represents the concatenated instances, and another mask is the background. In our case, the background is not the pixels outside the plant but pixels that are not part of any instance of fungi or root cortex when present. The acquired mask of the Microscopy images is in Figure 3.3. Also, acquired mask and ground truth segmentation of Synthetic image is in Figure 3.4. We can see that, on Microscopy and Synthetic images, this method gives comparable results. The resulting segmentation masks could be more accurate but give a good baseline for further methods. They show the general structure of the nutrition-transferring instances in the image.

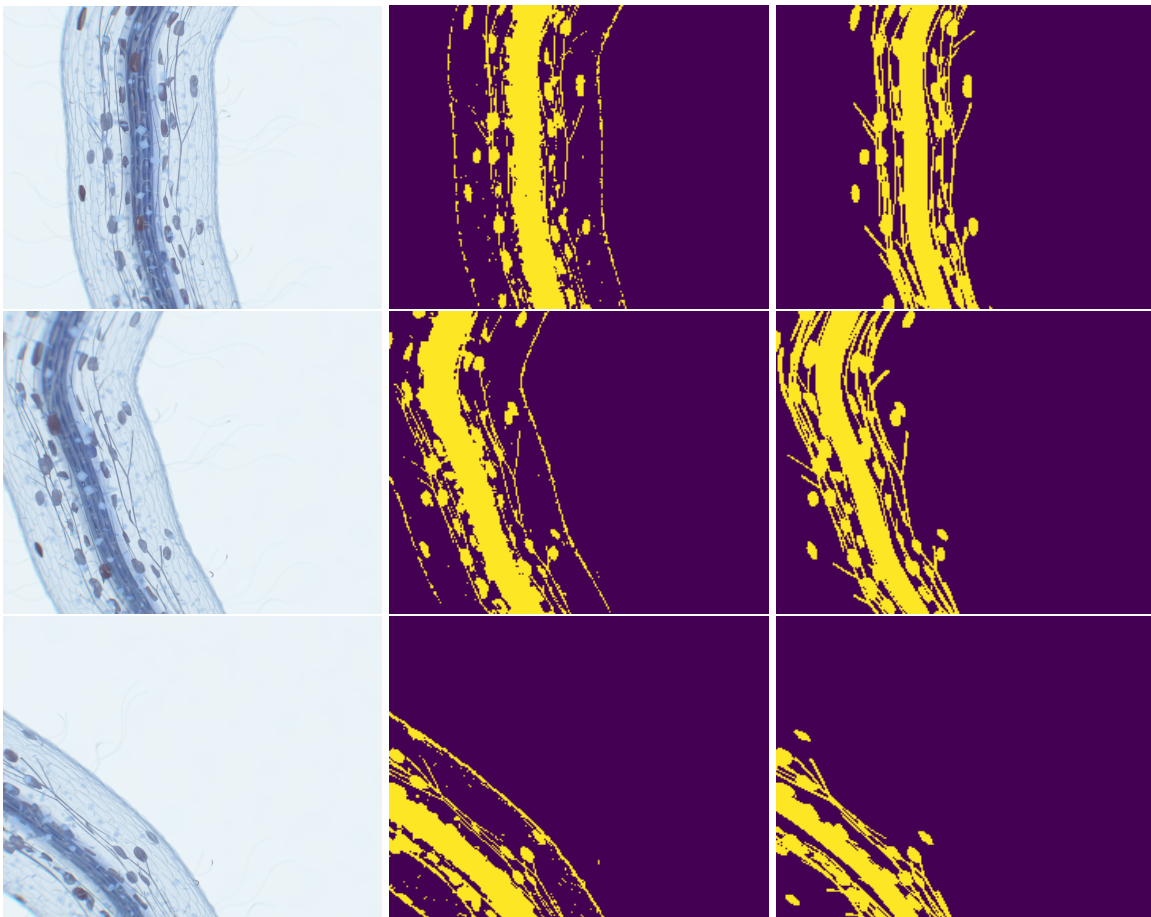


Figure 3.4: Visualization of Extracted Masks with Clustering on Color on Synthetic Data. (Original image on the left column, extracted mask on the middle, ground truth mask on the right)

### 3.3 Clustering on DINO Features

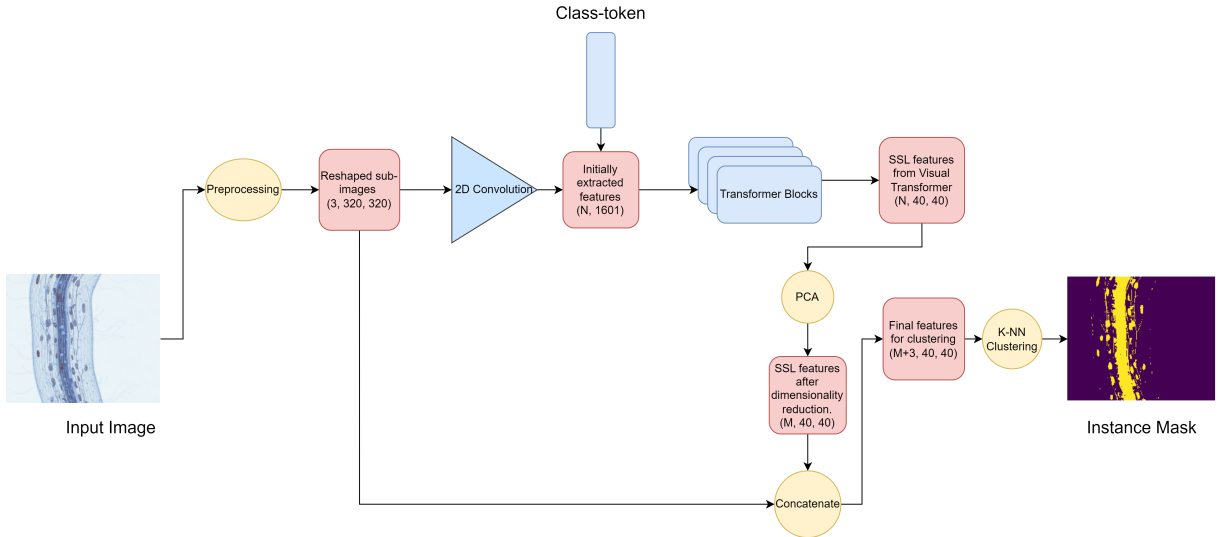


Figure 3.5: Model for DINO Feature Clustering

Clustering based on the color values gives a meaningful baseline for our experiments but needs to be more informative. Color values, interpreted as feature vectors, lack information about the neighborhood and general structure of the image. When we provide this information to each patch, clustering on the embedding space is more distinctive in intensively similar image patches. Clustering using the mean of patches using the color values contains more information about the neighborhood. However, any pooling operation to the same dimensional embedding space loses information about each pixel. Using a convolutional neural network extracts high-dimensional feature embedding for each patch but also may not contain global information about the image. So, to extract more informative feature vectors for each image patch, we use a Visual Transformer model DINO[19]. The General Structure of the DINO[19] model we use is in Figure 3.5. The first layer of the DINO[19] model extracts high dimensional embedding for each image patch in the image. This operation extracts a feature embedding for each  $(8 \times 8)$  sub-image by a convolutional layer. After this operation, we have a  $[N, 40, 40]$  tensor for a  $[3, 320, 320]$  input image. After resizing the tensor, we have a sequence of 1600, with  $N$ -dimensional feature embedding for the transformer model. Each member of the sequence represents a different patch on the image. Also, a class token is concatenated to the sequence, making it a total length of 1601. This class token is important for even unsupervised SSL extraction to get the attention map for the images. Then we send this sequence to 11 transformer layers and get a feature embedding for each patch on the image. Also, attention values from the class token to each image patch encode global information about the image. We use an attention head of 6. Visualization of these attention heads is in Figure 3.6.



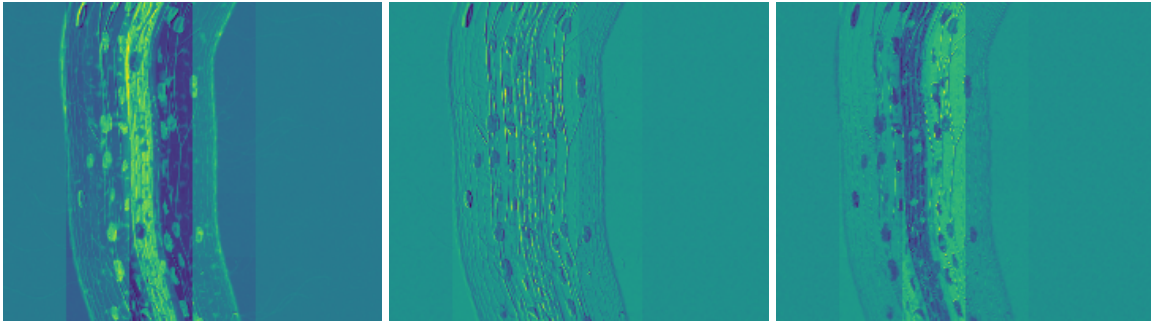


Figure 3.6: Visualization of Different Attention Heads

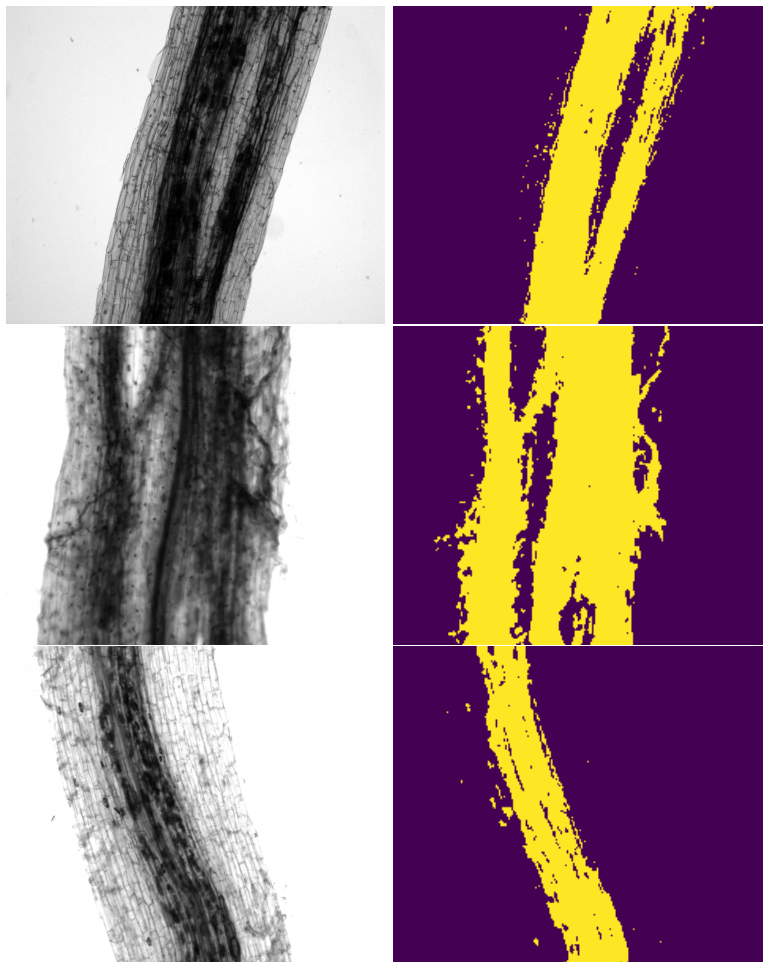


Figure 3.7: Visualization of Extracted Masks with Clustering on Extracted Features on Microscopy Images. (Original image on the left column, extracted mask on the right)

After obtaining a more informative feature encoding space for the images, we cluster the patches based on their feature vectors. After the feature extraction, also we can apply PCA[33]

on the feature encodings, decreasing the dimension of embedding space and resulting in a more robust representation. Also, we found that, concatenating the initial color values to the extracted features gives a better clustering result. We use k-nn[32] on obtained feature encodings; each cluster center gives a different class. Similar to color-based clustering, when  $k=2$ , it gives us an instance segmentation based on the nutrition transferring parts. The results from the experiment on Microscopy images are in Figure 3.7. Also, we can see the obtained mask and ground truth mask for Synthetic images in Figure 3.8. Qualitatively, we can already see the improvement compared with the color-based approach. We can see that our obtained mask is less noisy, and it is also able to distinguish parts where the color-based approach failed to. One example is the edges of the plant, which in the color-based method, are included in the mask of Synthetic images, but now they are not included.

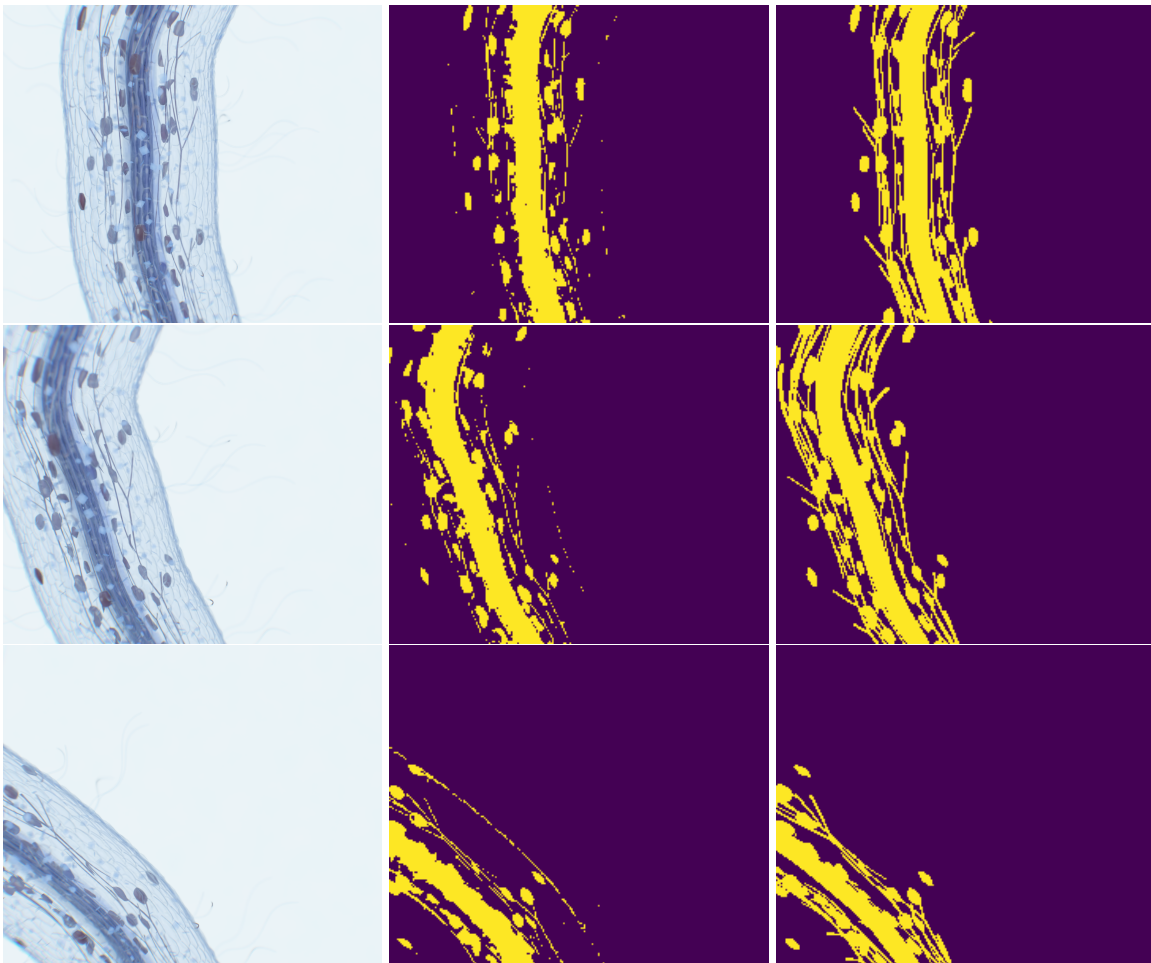


Figure 3.8: Visualization of Extracted Masks with Clustering on Features on Synthetic Data. (Original image on the left column, extracted mask on the middle, ground truth mask on the right)

### 3.4 Clustering on Fine Tuned Features

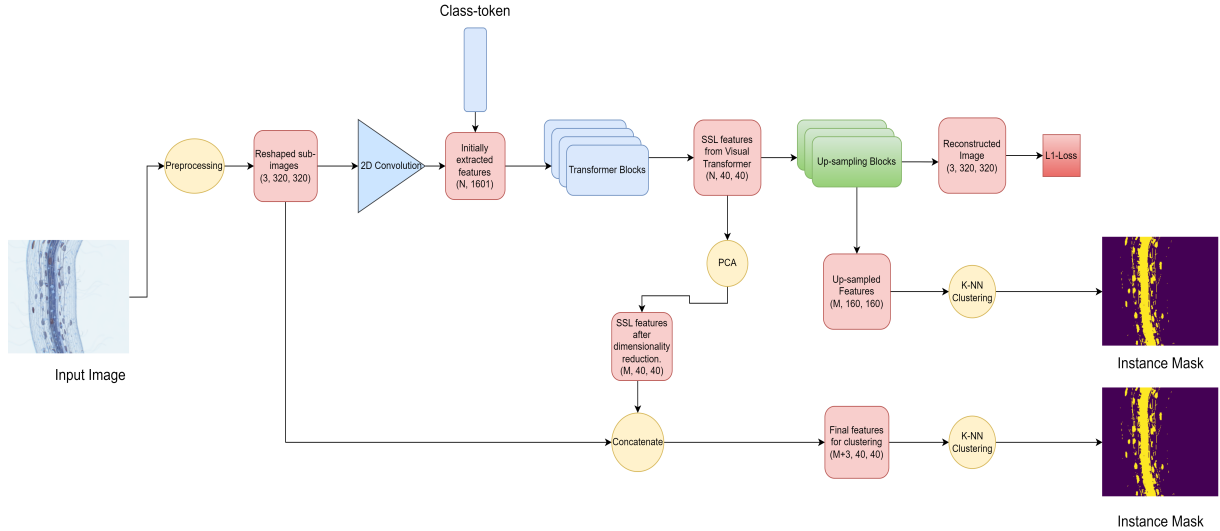


Figure 3.9: Model for DINO Feature Clustering with Added Fine Tuning and Up-sampling Training

DINO[19] model is trained on the ImageNet[4] dataset, which includes object-centric real-world scenes. Therefore, when we directly use the DINO[19] model trained on the ImageNet[4], we do not consider the domain adaptation. In other words, the distribution of ImageNet[4] images and Microscopy images vastly differ, which results in non-informative embedding dimensions during the feature extraction. Also, we want to adapt the information extraction to be more informative on the Microscopy Images domain. To overcome the presented issues, we train a separate model using DINO[19] as a backbone while finetuning the DINO[19] model. The general structure of the model is in Figure 3.9. We train for the last two layers of the DINO[19] alongside three upsampling layers. With this model, the output is the exact size of the input image, and we train this model with reconstruction loss. Reconstruction loss is L1-Loss[34] on the three output channels.



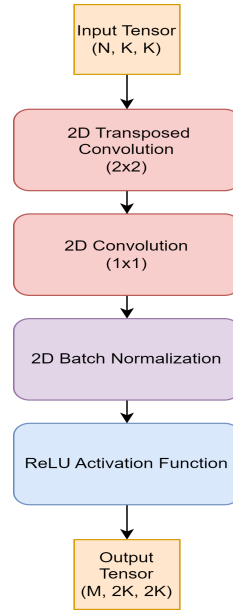


Figure 3.10: Blocks that are used in Up-sampling model.

We extract the feature embeddings from the DINO[19] model and send them through 3 upsampling layers. Each upsampling layer increases the spatial resolution by 2. As we can see from the Figure, starting with a spatial size of  $[H/8, W/8]$ , after the added blocks, we have a spatial size of the original input image. Detailed visualization of each upsampling block is in Figure 3.10. We use a Transposed Convolutional layer with a stride of 2, which increases the spatial size. After that, we use a convolutional layer with a kernel size of 1. This operation is equivalent to a linear operation in each feature vector, which performs a direct basis transformation. We follow by batch-norm and activation layer for each upsampling block. We use three blocks to achieve the aimed spatial resolution. With the finetuning, now, features extracted from the DINO[19] model are more informative in the Microscopy image domain. An example result of the instance mask gathered by the k-nn[32] method while  $k=2$  for the Microscopy image is in Figure 3.11. Also, an example result from the Synthetic image alongside the ground truth mask is in Figure 3.12.

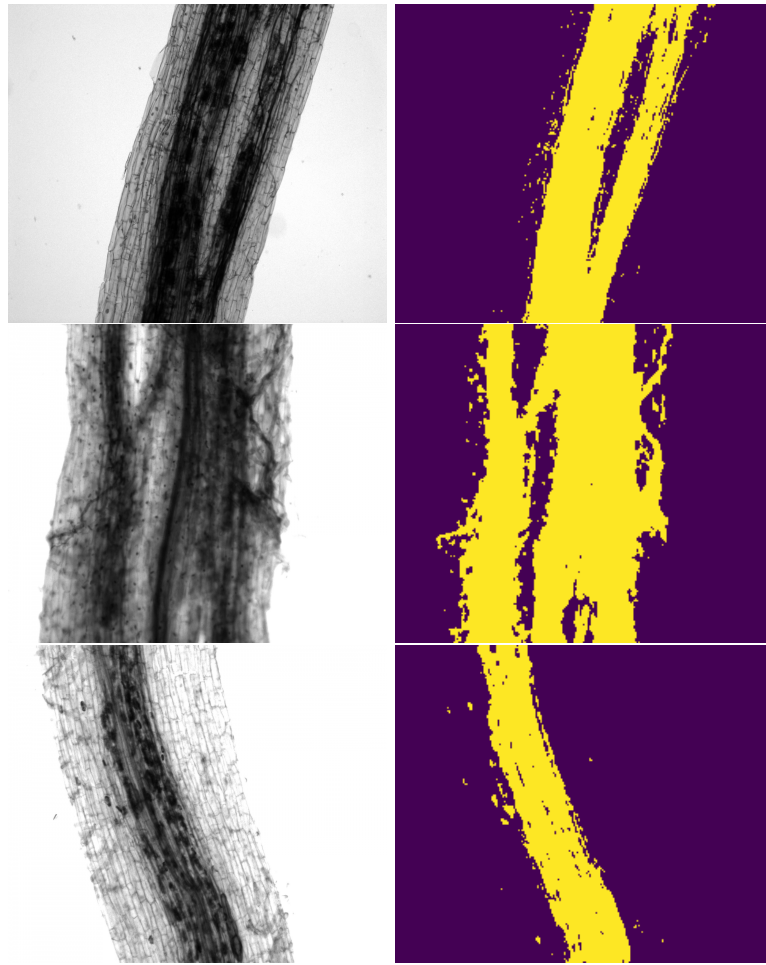


Figure 3.11: Visualization of Extracted Masks with Clustering on Fine-tuned Features on Microscopy Images. (Original image on the left column, extracted mask on the right)

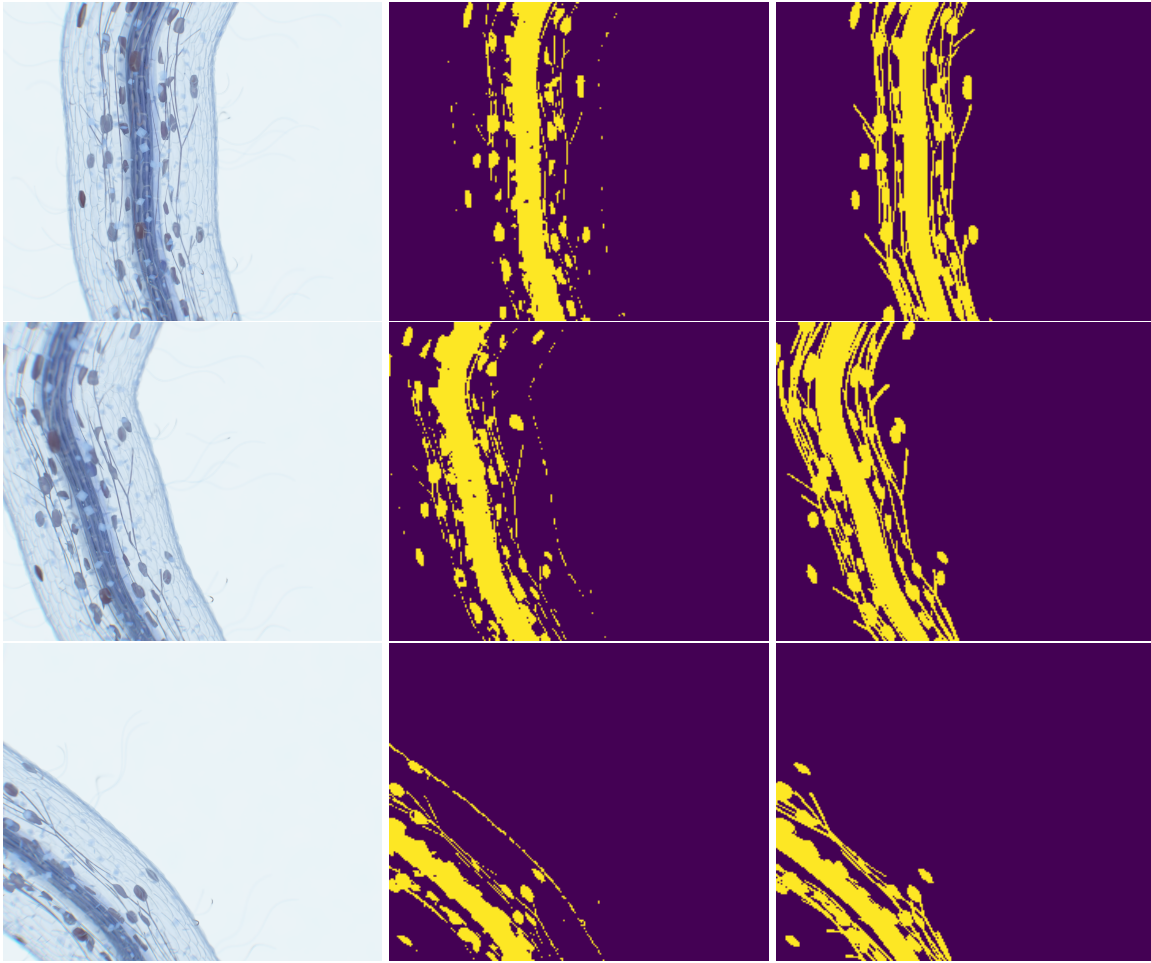


Figure 3.12: Visualization of Extracted Masks with Clustering on Fine-tuned Features on Synthetic Data. (Original image on the left column, extracted mask on the middle, ground truth mask on the right)

### 3.5 Extension on Semi-Supervision

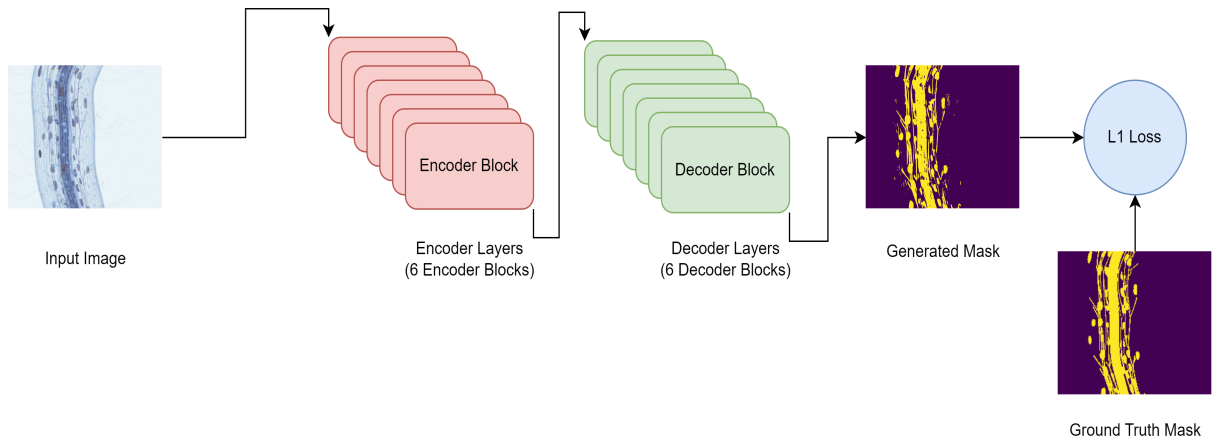


Figure 3.13: UNet model for supervised and semi-supervised training.

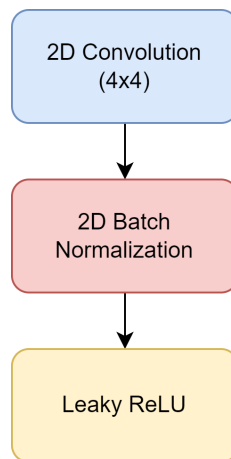


Figure 3.14: UNet Encoder Block.

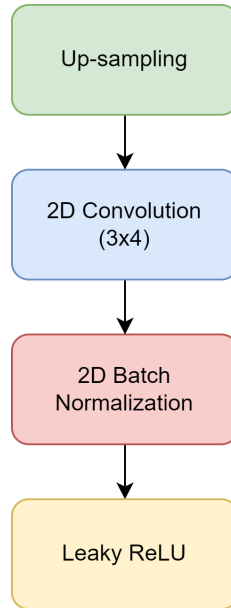


Figure 3.15: UNet Decoder Block.

We extend our experiments with a supervised set up to establish a baseline for unsupervised cases and further evaluate methods from the previous sections. Since only the synthetic data has the ground truth mask values, we can only evaluate the synthetic data with this method. To mimic the actual case of the microscopy image dataset, we use a tiny portion of the synthetic data for the experiments. We use ten synthetic images for the fully supervised training and two different subsets of the 10-image dataset for the supervised and semi-supervised training cases. We experiment with 20%, 50%, and 100% supervision. When trained with 20% supervision, the model is only trained with two images; with 50% supervision, the model is trained with five images. We experiment if we can generalize with overfitting to a notably smaller subset when our database is already tiny. To make use of the supervision, we use a UNet[10] model. The overall structure of the model is in Figure 3.13. We use six encoding blocks in Figure 3.14 and six decoding blocks in Figure 3.15. The output of the model is a 1-channel mask, which is then trained with a pixel-wise cross-entropy loss.[10] We trained each model for 1000 epochs.

The resulting masks from the UNet model are in Tables 3.5, 3.6, and 3.7. We can see that, even with low supervision, the model can generate the general structure of the aimed mask. The main downside of the low supervision is that it can fail to distinguish between similarly structured sections on the original image, whereas those locally similar structures have different classes. For example, black lines that occur throughout the image, which is a repeating part of the root but not part of the instance, are also selected because of their similarity to the actual nutrition-transferring hyphae. This problem is eliminated with more supervision, and we can see the potential ideal baseline with 100% supervision, which is significantly similar to the ground truth mask.

### 3.6 Experiments and Results

To successfully evaluate different methods, we compare each method qualitatively and quantitatively. We discuss each approach’s performance and mention each method’s required investment. Unfortunately, we can only qualitatively evaluate the performance of original microscopy images because we lack the ground truth mask for the original dataset. However, we can extend our discussion with quantitative assistance from synthetic data. We achieve valuable insights about the methods from synthetic data; however, since it doesn’t perfectly represent the nature of the original dataset, we can not assess the performance of the models on original data purely based on synthetic data. One example is, in original black and white data, color is a one-dimensional feature that is not distinctive enough to get a good enough baseline with clustering merely on color intensity values. But in synthetic data, 3-channel color information is distinct enough to set up a significantly high baseline to other methods. This is also due to how we generate the synthetic data. We render the image with each distinct nutrition transferring part as a different object with different distribution of color value. As a result, the nutrition transferring parts have a significantly different color structure. With this informative dataset, we can finally set a high baseline with color clustering with synthetic data. But when we examine the case for the original images, clustering based on color does not perform comparably to the synthetic case.

In Tables 3.1, 3.2, and 3.3, we compare the performance of different methods on original data. Tables 3.1 and 3.2 show that color-based clustering works as a foreground-background segmentation for the entire plant root. However, it can not distinguish between the darker nutrition-transferring part and the rest of the plant. This is because single-channel intensity information is not distinctive enough to discriminate between the parts inside the plant. In Table 3.3, we can see a case performs comparably well compared to Tables 3.1 and 3.2. This performance increase is due to image intensity distribution. In Table 3.3, the original image has a much closer color intensity distribution to the background in the plant region when there are no instances. However, obtained mask still has a lot of noise in many empty plant areas. This is also a robustness problem. Clustering with only color values highly depends on the intensity distribution across the image. We can overcome this problem by preprocessing the image and changing the distribution of the image. But this requires a special, non-generalizable preprocessing for each separate image, which requires comparable investment to segmenting each image by hand because each image would require a different shift in its intensity distribution.

To achieve a more robust, generalizable, and well-performing performance, we use Self Supervised Learning features to cluster. This method extracts high-dimensional features for each image patch with the DINO[19] feature extractor. Furthermore, these features are normalized for different images, making the method more robust than clustering on color. Also, extracted SSL features contain neighboring and global information alongside the information about the image patch. We can see how these features enhance the segmentation performance in Tables 3.1, 3.2, and 3.3. Now we can generalize the method better for different images, with the only hyperparameter being the number of dimensions for PCA[33]. However, a single parameter(20) works well for almost all images. With this method, we can

discriminate between nutrition-transferring darker segments of the image with the rest of the plant. In Tables 3.1 and 3.2, we can now discriminate inside the plant. And in Table 3.3, we can get a better, more refined segmentation than color-based segmentation. We further finetune the segmentation result by training the DINO[19] segmenter as described in section 3.4. It does not result in a significant improvement since DINO[19] is a generalizable network already. However, we can see small improvements in segmentation results. The primary improvement is that we can now better segment small nodules outside the main segment, which is lighter regions inside the plant. We can directly see this effect in Table 3.1. With this experiment, we can see that it is possible to achieve robust and convincing segmentation results with no supervision and a small dataset in the context of microscopy images. So this method can be used as a guiding preprocess for segmenting and processing microscopy images.

For the second part of the experiments, we evaluate synthetic images. Segmentation results for the synthetic images are in Tables 3.5, 3.6, and 3.7. The main advantage of the synthetic dataset compared to the original dataset is that it contains ground truth masks. By using this mask, we can extend our experiments to different levels of supervision and do quantitative analysis. Even though some parts of the results may not perfectly represent the performance of the original case, it gives us valuable insight into the potential performance of the mentioned methods. The main difference in terms of performance between original and synthetic images is the color clustering case. As mentioned before, this is due to the nature of the synthetic dataset. As we can see from the examples, color-based clustering now gives segmentation results even comparable to the semi-supervised cases with low supervision. This simple method can now set up a high baseline for other models and generate a mask that contains the overall structure of all nutrition-transferring sections of the plant. Also, compared to the original data, with synthetic data, clustering based on color is robust. Meaning that segmentation results do not change between different images due to the nature of rendering synthetic data and the resulting non-changing color distribution between different generated images.

Similar to the original microscopy image case, we extend our experiments with SSL features from the DINO[19] feature extractor. Color-based clustering is already performing close to the possible performance capacity, being close to a low semi-supervision case, so now the extracted features concatenated to color values perform as a finetuning on the segmentation results. The main improvement over the color-clustering approach is that we don't see false negatives over the edges of the plant structure. Even though this improvement does not affect the quantitative results significantly, it resolves an occurring problem in a critical section of the image. We also further finetuned the method with training, as mentioned in section 3.4. This method also increases the performance slightly. One main difference can be seen in Table 3.6. Now it can segment the image's hyphae section better than the previous SSL method.

We further extend our experiments with a small supervision case to set up a better baseline for the quantitative results. With ten synthetic images, we experiment with 20%, 50%, and fully supervised cases. Please note that 20% supervised, in our context, means that we only use two synthetic images for the full training. Even with the fully supervised case, we only

train with ten images. We chose to set a significantly small dataset to match the real-world case. And the semi-supervised cases are even smaller subsets of an already small dataset. These experiments could be performed with thousands of generated images, but they can be misleading since, in real-world scenarios, one does not have thousands of microscopy images of the same scene. Since unsupervised methods clusters on each image and can work even with a single image, we set up a baseline with the mentioned tiny dataset. We can see that with 20% supervision, we can outperform the unsupervised cases. However, segmentation results have artifacts. These artifacts are in the sections of the plant, where a non-instance segment has a similar structure to the nutrition-transferring segments. These artifacts occur in the part of the image that looks like hyphae but is not. With that small supervision, the model can not distinguish between mentioned parts. These artifacts are not present with higher supervision, and we can see a significant performance increase with increasing supervision. With the fully supervised case, we can see near-perfect segmentation results for many cases.

To compare the methods more clearly, we use pixel-based IoU scores to evaluate the performance of different models. IoU(Intersection over Union) can be formulated as:

$$IoU(X, Y) = \frac{|X \cap Y|}{|X \cup Y|} \quad (1)$$

where  $X$  is the set of instance patches in the generated sample and  $Y$  is the set of ground truth instance patches in the corresponded ground truth mask. IoU score ranges from 0 to 1, 0 meaning no patches matched with the prediction and 1 meaning a perfect prediction. A direct comparison is in Table 3.4. The best model for the unsupervised case is SSL with finetuning, and the best model for the supervised case is with full supervision. IoU scores between color-based clustering and SSL clustering are quite insignificant. This is due to the number of pixels that are present on the edge of the image and the hyphae being low, which is the main improvement of SSL features over color-based clustering in synthetic images. We can also say that, as expected, supervision greatly increases performance on segmentation. Each level of supervision significantly increases segmentation performance.

In cases where supervision is available, any additional level of supervision proves quite useful. However unsupervised SSL feature method provides a solution for most real-world cases where a ground truth segmentation is not available. Also, it can act as a preliminary guide to segment critical parts of the image. Also, since the clustering on SSL features works at each image, using the model with even one image does not make a difference in the result. So when ground truth segments are available, it is ideal to use a supervised method, but when it is not available, the unsupervised method can achieve comparable results with even one image.



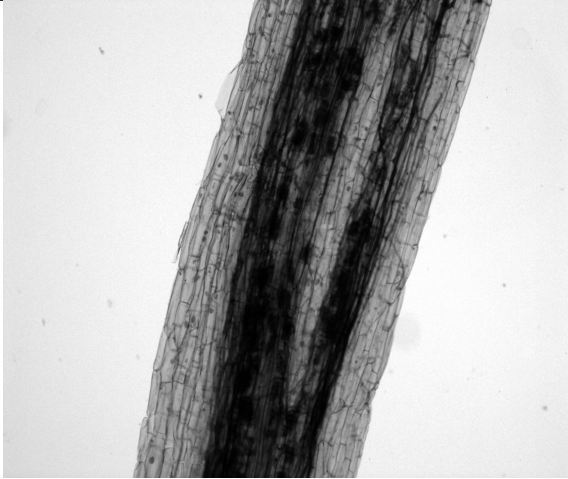



Original Image	Color Clustering
	
Clustering with SSL Features	Clustering with Fine-tuned SSL Features
	

Table 3.1: Comparison of different unsupervised clustering methods on Microscopy image Example 1.





Original Image	Color Clustering
	
Clustering with SSL Features	Clustering with Fine-tuned SSL Features
	

Table 3.2: Comparison of different unsupervised clustering methods on Microscopy image Example 2.

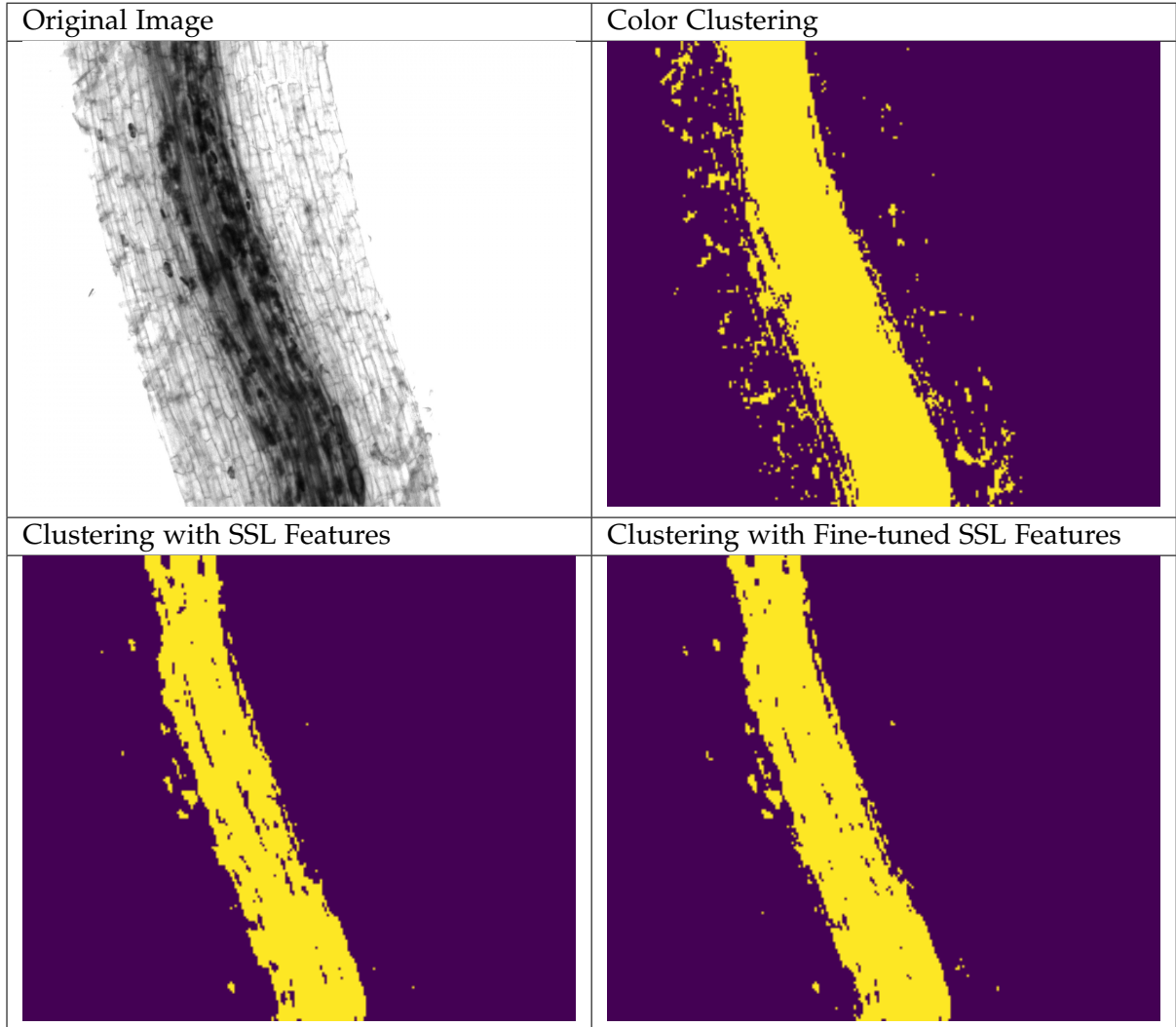


Table 3.3: Comparison of different unsupervised clustering methods on Microscopy image Example 3.

Supervision	Features and Clustering	Model	IoU Scores
0%	Color intensity values	No Model(Only K-means)	0.637
0%	SSL features	DINO	0.644
<b>0%</b>	<b>Finetuned SSL features</b>	<b>DINO</b>	<b>0.647</b>
20%	L1 pixel loss	UNet	0.735
50%	L1 pixel loss	UNet	0.800
<b>100%</b>	<b>L1 pixel loss</b>	<b>UNet</b>	<b>0.896</b>

Table 3.4: IoU scores of all unsupervised and supervised experiments on Synthetic data.

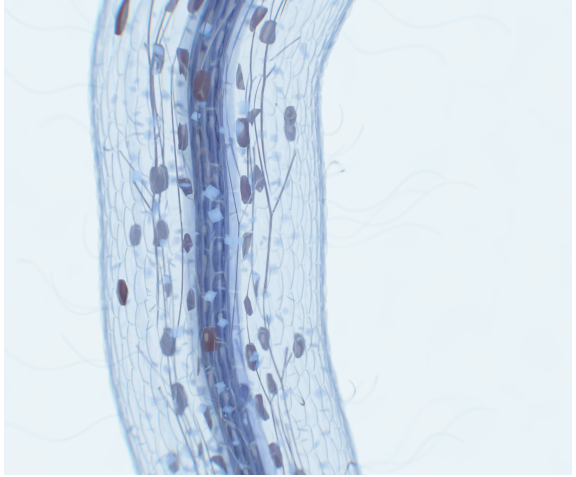

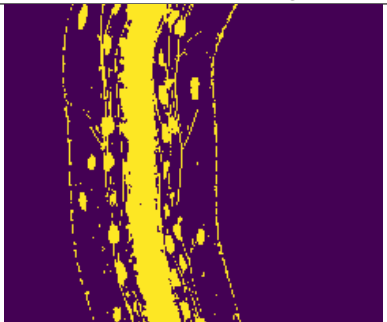
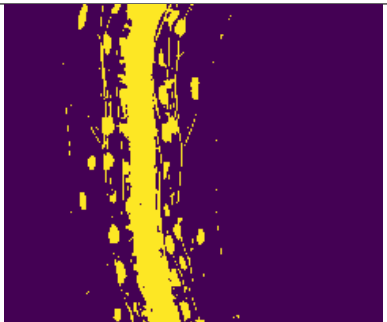
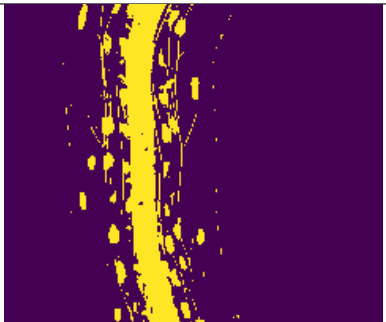



Original Image	Color Clustering	
		
Color Clustering	SSL Features	Fine-tuned SSL Features
		
20% Supervised	50% Supervised	100% Supervised
		

Table 3.5: Comparison of different unsupervised and supervised clustering methods on Synthetic image Example 1.

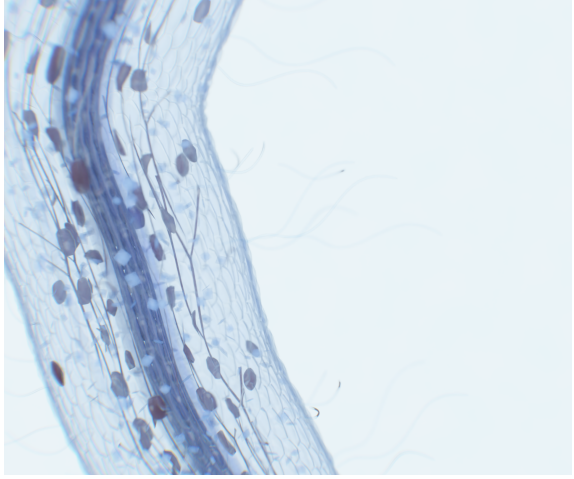
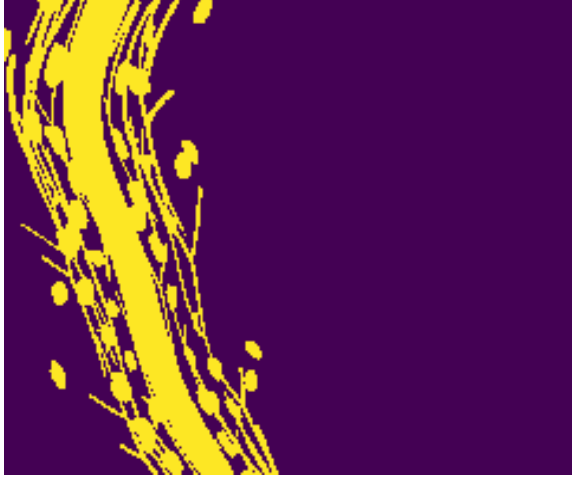
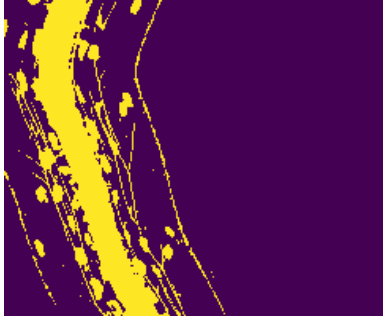
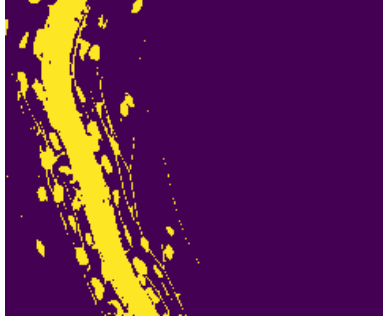
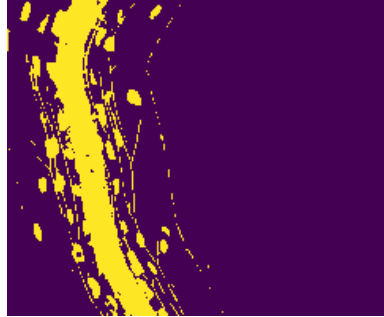


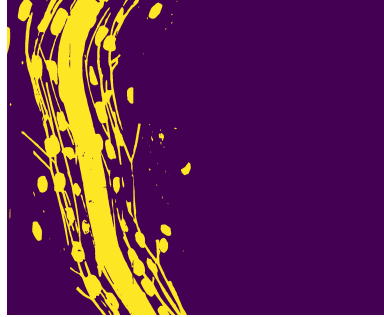
Original Image		Color Clustering	
			
Color Clustering	SSL Features	Fine-tuned SSL Features	
			
20% Supervised	50% Supervised	100% Supervised	
			

Table 3.6: Comparison of different unsupervised and supervised clustering methods on Synthetic image Example 2.

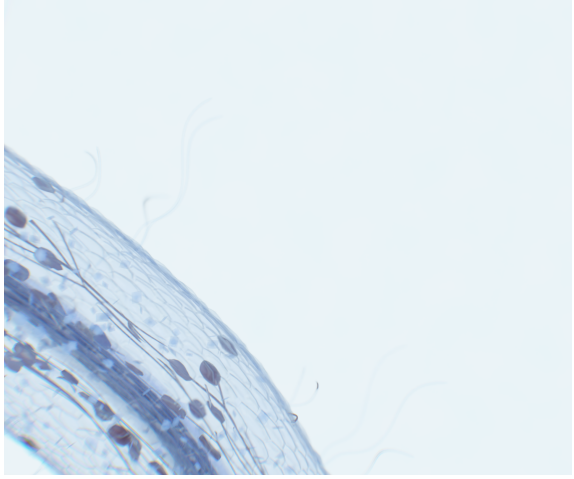

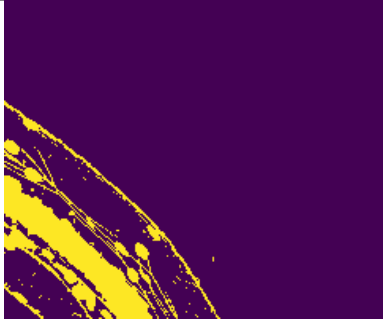
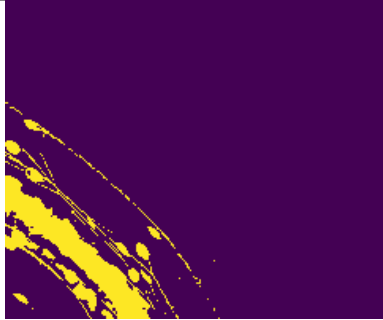
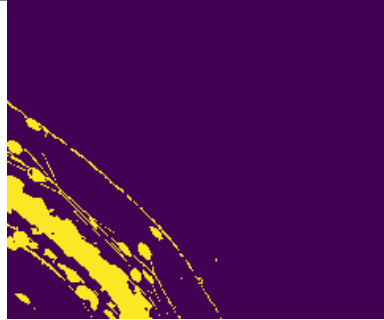



Original Image		Color Clustering	
			
Color Clustering	SSL Features	Fine-tuned SSL Features	
			
20% Supervised	50% Supervised	100% Supervised	
			

Table 3.7: Comparison of different unsupervised and supervised clustering methods on Synthetic image Example 3.

## 4 Conclusion

This research project examined different feature clustering methods to obtain patch-wise semantic segmentation masks. We did a series of experiments with two separate datasets. Another dataset that we use is a synthetic dataset that mimics microscopy images. The first dataset that we used is the original AMF microscopy images. First, we experimented with clustering based on averaged color intensity on each patch and clustered each patch by its intensity value. With this approach, when we use the k-nn method with  $k=2$ , we can obtain a mask that shows nutrition-transferring instances in AMF images. With this implementation, we showed that color-based clustering is a simple and fast method that gives good results in a short time with little investment. However, it is not a robust method that can be generalized with all different intensity distributions in various settings. Also, in a real-world scenario, it performs as a foreground-background segmentation rather than an instance segmentation. Color-based segmentation also set a baseline for other segmentation methods we experimented with. The second method we tried was clustering on SSL features we extracted using a pre-trained DINO model. In this model, we clustered DINO's feature vectors in the high-dimensional embedding space. With this clustering approach, we improved the results in both quantitative and qualitative analysis. We also showed that this method is more robust and does not require preprocessing like the color-based method. In order to further increase the performance, we fine-tuned the DINO model with reconstruction-based training by adding an upsampling head on top of the DINO features. With this training, we were able to extract more fine-tuned features and get the overall best results for the unsupervised methods. To compare our methods with different levels of supervision, we trained a UNet model. We used 20%, 50%, and 100% subsets of an already small dataset to match the real-world scenarios. We see that unsupervised methods give a comparable result with 20% supervision; however, highly supervised methods overperform the unsupervised methods. This shows us that supervised methods for this domain are better-performing models. However, supervised methods require already annotated datasets. Even though supervised methods perform better than unsupervised methods, supervision is generally unavailable for specific domains. We showed that unsupervised methods could be used to extract meaningful semantic segmentation even in cases where the annotation is not available. For future work, unsupervised methods with SSL features can be further improved by refining embedding space[25] or using an additional generative model to extract the mask as a byproduct.[27]

# List of Figures

2.1	Visualization of ViT model based on the MOVE paper Figure 3.[27]	4
3.1	Example images of Microscopy Data	6
3.2	Example images of Synthetic Data	7
3.3	Visualization of Extracted Masks with Clustering on Color on Microscopy Images. (Original image on the left column, extracted mask on the right)	8
3.4	Visualization of Extracted Masks with Clustering on Color on Synthetic Data. (Original image on the left column, extracted mask on the middle, ground truth mask on the right)	9
3.5	Model for DINO Feature Clustering	10
3.6	Visualization of Different Attention Heads	11
3.7	Visualization of Extracted Masks with Clustering on Extracted Features on Microscopy Images. (Original image on the left column, extracted mask on the right)	11
3.8	Visualization of Extracted Masks with Clustering on Features on Synthetic Data. (Original image on the left column, extracted mask on the middle, ground truth mask on the right)	12
3.9	Model for DINO Feature Clustering with Added Fine Tuning and Up-sampling Training	13
3.10	Blocks that are used in Up-sampling model.	14
3.11	Visualization of Extracted Masks with Clustering on Fine-tuned Features on Microscopy Images. (Original image on the left column, extracted mask on the right)	15
3.12	Visualization of Extracted Masks with Clustering on Fine-tuned Features on Synthetic Data. (Original image on the left column, extracted mask on the middle, ground truth mask on the right)	16
3.13	UNet model for supervised and semi-supervised training.	17
3.14	UNet Encoder Block.	17
3.15	UNet Decoder Block.	18



## List of Tables

3.1	Comparison of different unsupervised clustering methods on Microscopy image Example 1. . . . .	22
3.2	Comparison of different unsupervised clustering methods on Microscopy image Example 2. . . . .	23
3.3	Comparison of different unsupervised clustering methods on Microscopy image Example 3. . . . .	24
3.4	IoU scores of all unsupervised and supervised experiments on Synthetic data.	24
3.5	Comparison of different unsupervised and supervised clustering methods on Synthetic image Example 1. . . . .	25
3.6	Comparison of different unsupervised and supervised clustering methods on Synthetic image Example 2. . . . .	26
3.7	Comparison of different unsupervised and supervised clustering methods on Synthetic image Example 3. . . . .	27

# Bibliography

- [1] M. T. Chiu, X. Xu, Y. Wei, Z. Huang, A. Schwing, R. Brunner, H. Khachatrian, H. Karapetyan, I. Dozier, G. Rose, D. Wilson, A. Tudor, N. Hovakimyan, T. S. Huang, and H. Shi. *Agriculture-Vision: A Large Aerial Image Database for Agricultural Pattern Analysis*. 2020. arXiv: 2001.01306 [cs.CV].
- [2] E. Smistad, T. L. Falch, M. Bozorgi, A. C. Elster, and F. Lindseth. “Medical image segmentation on GPUs – A comprehensive review”. In: *Medical Image Analysis* 20.1 (2015), pp. 1–18. ISSN: 1361-8415. DOI: <https://doi.org/10.1016/j.media.2014.10.012>.
- [3] C. Godard, O. M. Aodha, M. Firman, and G. Brostow. *Digging Into Self-Supervised Monocular Depth Estimation*. 2019. arXiv: 1806.01260 [cs.CV].
- [4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.
- [5] H. Caesar, J. Uijlings, and V. Ferrari. *COCO-Stuff: Thing and Stuff Classes in Context*. 2018. arXiv: 1612.03716 [cs.CV].
- [6] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. “The Cityscapes Dataset for Semantic Urban Scene Understanding”. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.
- [7] K. He, G. Gkioxari, P. Dollár, and R. Girshick. *Mask R-CNN*. 2018. arXiv: 1703.06870 [cs.CV].
- [8] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. *End-to-End Object Detection with Transformers*. 2020. arXiv: 2005.12872 [cs.CV].
- [9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira, C. Burges, L. Bottou, and K. Weinberger. Vol. 25. Curran Associates, Inc., 2012. URL: [https://proceedings.neurips.cc/paper\\_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf).
- [10] O. Ronneberger, P. Fischer, and T. Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. arXiv: 1505.04597 [cs.CV].
- [11] R. Girshick, J. Donahue, T. Darrell, and J. Malik. *Rich feature hierarchies for accurate object detection and semantic segmentation*. 2014. arXiv: 1311.2524 [cs.CV].

- [12] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. *Unsupervised Learning of Visual Features by Contrasting Cluster Assignments*. 2021. arXiv: 2006.09882 [cs.CV].
- [13] X. Chen, H. Fan, R. Girshick, and K. He. *Improved Baselines with Momentum Contrastive Learning*. 2020. arXiv: 2003.04297 [cs.CV].
- [14] G. Shin, S. Albanie, and W. Xie. "Unsupervised Salient Object Detection With Spectral Cluster Voting". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2022, pp. 3971–3980.
- [15] K. O’Shea and R. Nash. *An Introduction to Convolutional Neural Networks*. 2015. arXiv: 1511.08458 [cs.NE].
- [16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. *Attention Is All You Need*. 2017. arXiv: 1706.03762 [cs.CL].
- [17] S. Smith and D. Read. *Mycorrhizal Symbiosis*. Elsevier Science, 2010. ISBN: 9780080559346.
- [18] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. arXiv: 2010.11929 [cs.CV].
- [19] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. *Emerging Properties in Self-Supervised Vision Transformers*. 2021. arXiv: 2104.14294 [cs.CV].
- [20] H. Vierheilig, A. P. Coughlan, U. Wyss, and Y. Piché. "Ink and Vinegar, a Simple Staining Technique for Arbuscular-Mycorrhizal Fungi". In: *Applied and Environmental Microbiology* 64.12 (1998), pp. 5004–5007. DOI: 10.1128/AEM.64.12.5004-5007.1998.
- [21] J. Watter. "Light Microscopy Image Analysis using Neural Networks". en. MA thesis. Technical University of Munich, Apr. 2021.
- [22] B. O. Community. *Blender - a 3D modelling and rendering package*. Blender Foundation. Stichting Blender Foundation, Amsterdam, 2018.
- [23] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. *MobileNetV2: Inverted Residuals and Linear Bottlenecks*. 2019. arXiv: 1801.04381 [cs.CV].
- [24] R. Hu, P. Dollár, K. He, T. Darrell, and R. Girshick. *Learning to Segment Every Thing*. 2018. arXiv: 1711.10370 [cs.CV].
- [25] M. Hamilton, Z. Zhang, B. Hariharan, N. Snavely, and W. T. Freeman. *Unsupervised Semantic Segmentation by Distilling Feature Correspondences*. 2022. arXiv: 2203.08414 [cs.CV].
- [26] X. Wang, Z. Yu, S. D. Mello, J. Kautz, A. Anandkumar, C. Shen, and J. M. Alvarez. *FreeSOLO: Learning to Segment Objects without Annotations*. 2022. arXiv: 2202.12181 [cs.CV].
- [27] A. Bielski and P. Favaro. *MOVE: Unsupervised Movable Object Segmentation and Detection*. 2022. arXiv: 2210.07920 [cs.CV].

- [28] X. He, B. Wandt, and H. Rhodin. *GANSeg: Learning to Segment by Unsupervised Hierarchical Image Generation*. 2022. arXiv: 2112.01036 [cs.CV].
- [29] A. Bielski and P. Favaro. *Emergence of Object Segmentation in Perturbed Generative Models*. 2019. arXiv: 1905.12663 [cs.CV].
- [30] Y. Yang, A. Loquercio, D. Scaramuzza, and S. Soatto. “Unsupervised Moving Object Detection via Contextual Information Separation”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 879–888. DOI: 10.1109/CVPR.2019.00097.
- [31] C. Chen, Q. Dou, H. Chen, J. Qin, and P. A. Heng. *Unsupervised Bidirectional Cross-Modality Adaptation via Deeply Synergistic Image and Feature Alignment for Medical Image Segmentation*. 2020. arXiv: 2002.02255 [eess.IV].
- [32] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer. “KNN model-based approach in classification”. In: *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings*. Springer, 2003, pp. 986–996.
- [33] I. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics. Springer, 2002. ISBN: 9780387954424.
- [34] H. Zhao, O. Gallo, I. Frosio, and J. Kautz. “Loss Functions for Image Restoration With Neural Networks”. In: *IEEE Transactions on Computational Imaging* 3.1 (2017), pp. 47–57. DOI: 10.1109/TCI.2016.2644865.