Image Segmentation of Adenovirus Particles in Food Vacuoles of Eukaryotic Organisms

Kang Fu, Runke Zhou, Shanci Li
mentored by Margot Olive, Environmental Chemistry Laboratory
EPFL Switzerland

***Abstract*—The aim of this project is to develop a reproducible virus segmentation program which can help to detect adenovirus particles in food vacuoles of eukaryotic organisms. In this report, we show that with the help of neural networks, U-net in particular, we can accomplish the task efficiently. Based on the model we develop, we can achieve the IoU of 0.3762 on the test set, which can achieve a satisfactory prediction.**

I. Introduction

In this project, we dealt with a real-world challenge to accomplish the image segmentation of adenovirus particles in food vacuoles of eukaryotic organisms offered by the environmental chemistry lab (LCE) of EPFL with machine learning methods.This task is originally done with human eyes, but it's very time consuming and hard to detect the position of the virus with human eyes. Thus, developing a method to detect the virus with the help of machine learning method is essential.

Image segmentation is the preliminary step for computer vision and object recognition[1]. The goal of image segmentation is to segment the area that users are interested in with the help of machine learning methods which are able to accomplish the task more accurate and faster. Specifically, solving the image segmentation task for adenovirus in organisms is very time-consuming and hard to detect with human eyes. Thus, designing a machine learning method for image segmentation in this project is essential.

Traditional image segmentation method attempts to detect the region of interests through edge detection or region extraction[2] by analyzing the mathematical interpretation of images. However, the traditional image segmentation method is sensitive to noise, which makes it not suitable for our project, as the image of the virus and the background is hard to differentiate. Also, the traditional methods are time-consuming, especially when the input are large images.

Neural networks based segmentation is applied in our project, considering about characteristics of robustness and noise insensitivity for this kind of method[1]. In addition, the availability of GPU has shortened the training time. Considering about the limited amount of data, we applied data augmentation techniques and transfer learning to solve the problem of limitation of training samples.

The rest of this report is organized as follows. In section II, the observation of the dataset is given and we introduced how we accomplish the dataset preprocessing to make the original data fit for later operation. Characteristics of the virus are introduced in section III. In section IV, transfer learning is introduced. Our models and methods are introduced in section V. Then our training parameters and results of experiments are specified in section VI. In the end, the discussion and conclusion are givens in section VII.

II. Dataset

A. Observation of Dataset

The dataset of adenovirus particles in food vacuoles of eukaryotic organisms are collected and provided by the environmental chemistry lab (LCE) of EPFL. It contains 30 images with virus indicated on images and 58 images without indications of virus. These images have different scales, different sizes and different exposure time.

In order to extract the feature of virus with neural networks, the dataset needs to be preprocessed to be fed to the training of neural networks.

B. Dataset Preprocessing

The preprocessing of the dataset contains two parts, including cutting images and labeling images.

1) Image Cut: Considering that original images are of different size and the U-net requires the same input image size, the original dataset needs to be cut before training in the neural network. Thus, in our project, we cut the original image of different sizes (2048*2048, 4096*4096) into smaller images of the size 512*512 to standardize the size of the dataset. Besides, some of original images have a large area of irrelevant background of the organisms which will affect the learning of neural network. After cutting the image, we chose a part of the background for training and discarded others, as too many images for the background would slow down the speed of training and can't improve the performance of our network.
2) Image Labling: U-net is trained with input images and their corresponding labelled masks[3]. Thus, we need to firstly label the dataset that has been cut into the same size. To solve this problem, we implemented the Matplotlib widget and the Lassoselecter to obtain a code that can generate the mask of different images by manually select the area of virus on the original picture. In Figure 1, the process for labeling image is shown.

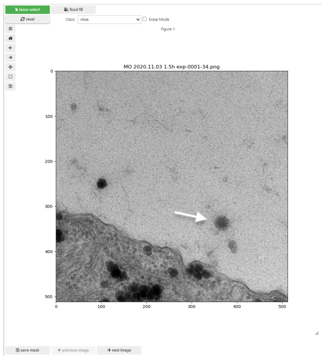

Fig. 1: The process of labeling original images

C. Composition of the dataset

After preprocessing the dataset, the final dataset that is used to train and validate the neural network consists of 340 images for the training set and 90 images for the validation set. All these images have a corresponding labelled mask and have the same size of 512*512. The test set consists of 40 unlabelled images to test the performance of the model.

III. Characteristics of the Virus

The virus we are interested in has several characteristics. Firstly, the virus has a shape of circle with two-layer structure. The first layer in the center of the virus is darker, while the second layer around the center of the virus is lighter. Secondly, the diameter of the virus is around 60nm to 90nm. To be more specific, the example of the virus is shown in Figure 2.

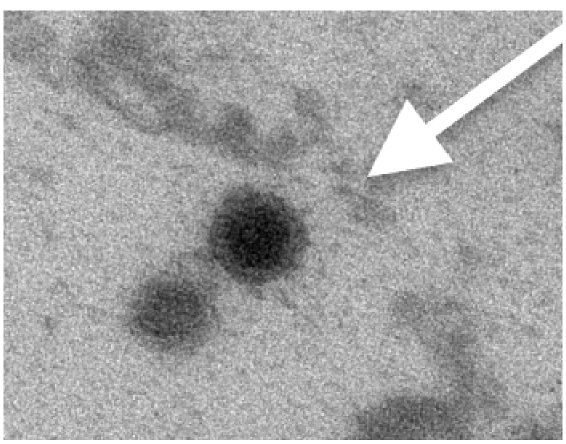

Fig. 2: Example of the virus in the organism

IV. Transfer Learning

Although after the cut of image, we have 470 images in the dataset, they are still not enough to train the neural network. Thus, we considers about introducing transfer learning in our project.

Transfer in machine learning is introduced in many literature[4][5]. It focus on the problem of applying general features learned in a base network on a base dataset to another target task[5]. Transfer learning is essential as it is beneficial when the target task has smaller dataset compared with the size of neural networks. With the help of transfer learning, this can not only reduce the training time but also get an effective result even with a small dataset.

This characteristic is very essential in our project, as we have only 470 images compared with a large neural network, U-net. Thus in this project, we apply the segmentation_models[6] library which enables us to use the pre-trained weight of multiple neural networks, for example, VGG, resnet or mobilenet. These weight are trained on the 2012 ILSVRC ImageNet dataset which includes more than 14 million annotated images.

V. Models and Methods

A. Data Augmentation

Even though we apply transfer learning considering about the small size of the dataset, we still apply data augmentation to get more training examples from our existing dataset.

We apply the Albumentations[7] library to accomplish the task of data augmentation. The original images may be transformed by different actions, for example, horizontal flip, random contrast and brightness, elastic transform and grid distortion. In our implementation, we have kept 10% of training images unchanged. The remaining images will have a 50% likelihood to be modified as one of the above transformation. Here is a data augmentation example shown in Figure 3.

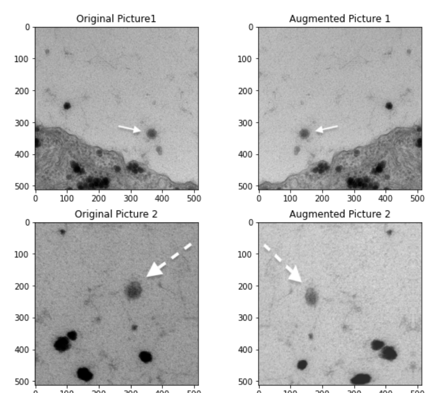

Fig. 3: Example of data augmentation

B. Models

1) Introduction: During the past few years, convolutional neural networks have shown outstanding performance on the task of image classification and segmentation. Traditional CNN adds several fully connected layers after convolutional layers, which converts the feature map to a feature vector which gives a prediction of classification of the image. But this method is actually not very suitable for our task for segmenting images because it consumes too much time and cannot predict precisely. Thus, fully convolutional networks[8] are implemented to solve the problem. Fully convolutional networks change fully connected layer into convolutional layers which enable pixelwise prediction of images.

2) U-net: Among all neural networks built on the basis of Fully Convolutional Networks(FCNs), we decided to accomplish the task with U-net. The first reason that we decide to use this network is designed for biomedical image segmentation which is very similar for our task to segment virus from organisms. The second reason is that U-net has been proved to have good performance on small dataset with data augmentation techniques. The architecture of U-net is shown in Figure 4.

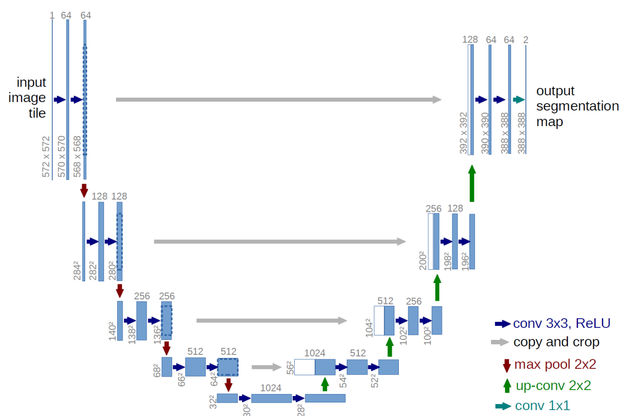

Fig. 4: The architecture of U-Net[3]

The architecture of U-net consists of a contracting part and an expansive part[3] as shown in Figure 3. The contracting part adapts the architecture of a convolutional network. It repeatedly applies 2 3*3 convolution with a ReLU. After that, a 2*2 max pooling with stride 2 is applied for downsampling. After each step of this process, the number of feature channel doubles. The expansive part consists of a 2*2 up-convolution, the concatenated with the corresponding feature map extracted in the contracting part, and then applies 2 3*3 convolution with a ReLU. In the end, a 1*1 convolutional layer is applied to map the feature vector into the classes we desired.

VI. Experiments

A. Training

The model trained in our project is a U-net trained with a dataset of training, validation and test sets. The dataset consists of 340 training images, 90 validation images and 40 testing images. Training and validation set also consists corresponding labeled masks in the file. All images and masks are of the same image size of 512*512 and same depth of 24. The U-net is initialized with pre-trained weights of backbone, mobilenet, trained with the imagenet. Optimization is accomplished with Adam optimizer. The activation function applied in the training is softmax. The loss function applied is cross entropy.

B. Model Optimization

The key problem in deep learning is how to fine-tune a large number of hyper-parameters in the neural network. Here hyper-parameters we used to optimize the model are introduced.

1) Batch size: Batch size determines the number of samples the neural networks trains every time. The number of batch size determines the optimization degree and speed of the network.
2) Steps per epoch: Steps per epoch refers to the number of steps in each epoch. Normally, steps per epoch is chosen to be the value of the number of samples divided by the batch size. But as we enables the data augmentation technique, steps per epoch can be chosen to a value a little bit larger than the normal value.
3) Epoch: The number of epoch indicates the number of time that the entire datasest needs to be passed forward and backward through the whole network. Choosing an appropriate epoch number can help us to find the best result of our model.

C. Model Evaluation

Optimization of our model also includes another important task to decide the evaluation metrics of our model. For an image segmentation task in our project, we decided to introduce the evaluation metric, intersection over union(IoU) to evaluate the performance of our model. As the name suggests, IoU reveals the division of the overlap area and the area of union of the predicted mask and the ground truth of the input image, which directly reveals whether our model makes a good prediction on the dataset. The image which gives an introduction about the definition of IoU is shown in Figure 5.

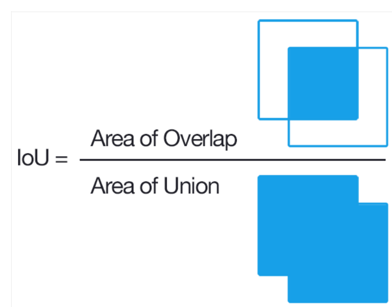

Fig. 5: The interpretation of IoU

D. Results

To start with, we need to decide the pre-trained weights of which backbone model we applied in our model. From the library[6],we can import different backbone models trained with ImageNet dataset. VGG, ResNet and Mobilenet are two popular methods applied in image segmentation. Thus, in this project, we implemented with mobilenet and ResNet to decide which has a better performance. The comparison between different backbone models under 30 epochs are shown in Table 1. As we can conclude from the result, the backbone model, MobileNet, provides the best result. Thus we choose to apply the pre-trained weights of MobileNet in our project.

Backbone Models	Validation IoU
MobileNet	0.3326
ResNet18	0.1681
VGG19	0.08168

TABLE I: Comparison between different backbone models

To choose optimized hyper-parameters, we decided to use a fixed validation set to find the final hyper-parameters. We didn't use cross-validation for choosing hyper-parameters as it's not very suitable for our project. Firstly, to apply cross-validation in deep learning is very time-consuming and expensive. What's more, considering about the size limitation of our dataset, cross-validation will have a large deviation.

The segmentation accuracy of our model is compared with the baseline, U-net trained without data augmentation technique. The comparison is shown in Table 2.

Methods	Batch Size	Steps	Epoch	Validation IoU
U-Net without data augmentation	32	11	60	0.173
Our model	32	20	30	0.3326

TABLE II: Comparison of our model and U-Net trained without data augmentation

After iterating through all hyper-parameters in our model, the optimized result is achieved with the batch size of 32, steps per epoch of 20 and 30 epochs. The optimized result obtain the validation IoU of 0.3326 and the training IoU of 0.2949. The IoU score of the training and validation set is shown in Figure 6. In Figure 7, we show the prediction of our model on the test set. The IoU evaluated on our test set is 0.3762. Based on the prediction mask and the IoU score on the test, we can conclude that our model has a satisfying result on the test set.

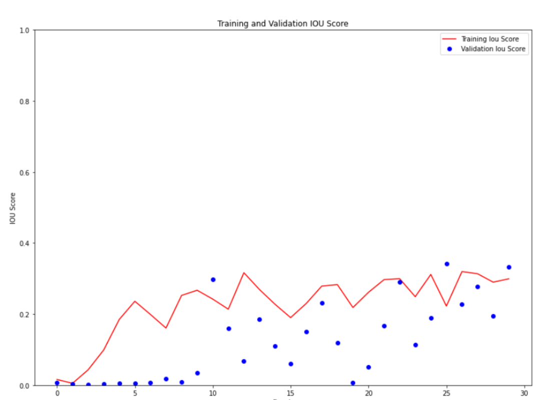

Fig. 6: Training and Validation IoU score during training of neural net

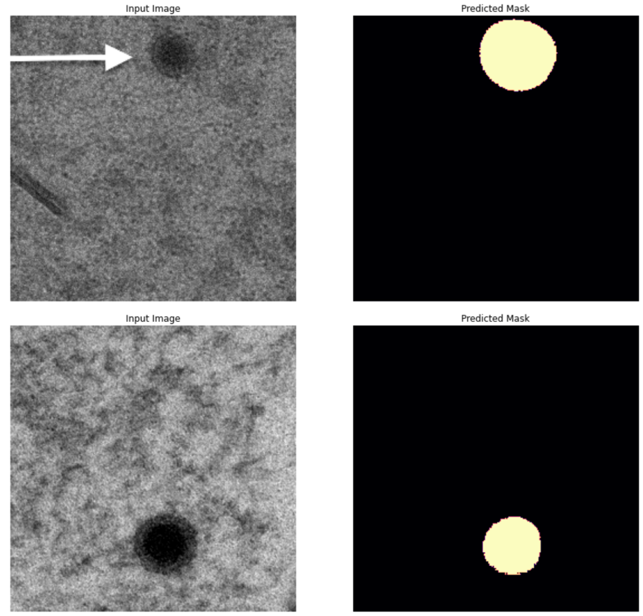

Fig. 7: Prediction of the position of the virus

VII. Discussion and Conclusion

A. Discussion

There are some aspects we think can be done in the future to improve the model.

1) The training of neural networks relies on the amount of the dataset. But in our project, the size of the dataset is very limited. Even though we apply cutting image and data augmentation techniques, the size of the dataset is still not enough. Thus, the performance of the model may be improved if we can get more data in the future.
2) What's more, the existing dataset also has the problem that the scale of each image can be different, which means that the size of the virus is different in each image depending on the scale. Thus, standardizing the scale of the dataset can also possibly improve the model.
3) Furthermore, applying other data pre-processing techniques may also be helpful. In our virus dataset, the key problem is that some characteristics of the virus are very similar to the backgrounds. Thus, introducing some image pre-processing techniques to remove the background noise can also be helpful for the model.

B. Conclusion

This report shows with appropriate data pre-processing techniques, the U-net can segment the adenovirus particles in food vacuoles of eukaryotic Organisms. With a small dataset, the U-net is trained by transfer learning using pre-trained weights on mobilenet. In conclusion, we can obtain a final model trained based on the U-net with a IoU score of 0.3326 on the validation set and 0.3762 on the test set, which is satisfying compared with the IoU score of the baseline.

References

[1] A. Ghosh, N. R. Pal, and S. K. Pal, "Image segmentation using a neural network," *Biological Cybernetics*, vol. 66, no. 2, pp. 151–158, 1991.

[2] K.-S. Fu and J. Mui, "A survey on image segmentation," *Pattern recognition*, vol. 13, no. 1, pp. 3–16, 1981.

[3] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[4] Y. Bengio, "Deep learning of representations for unsupervised and transfer learning," in *Proceedings of ICML workshop on unsupervised and transfer learning*, 2012, pp. 17–36.

[5] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in neural information processing systems*, 2014, pp. 3320–3328.

[6] P. Yakubovskiy, "Segmentation models," https://github.com/qubvel/segmentation_models, 2019.

[7] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and flexible image augmentations," *Information*, vol. 11, no. 2, 2020. [Online]. Available: https://www.mdpi.com/2078-2489/11/2/125

[8] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.