



## Microscopic Image Filtering with Semantic Segmentation using U-Net Architecture

<sup>1,\*</sup>Muhammad, K. S. and <sup>2</sup>Usman, A.

<sup>1</sup>Department of Information and Communication Technology, Federal University Birnin Kebbi, P.M.B 1157, Kebbi State, Nigeria.

<sup>2</sup> Department of Physics with Electronics, Federal University Birnin Kebbi, P.M.B 1157, Kebbi State, Nigeria.

### ABSTRACT

Deep learning has been extremely ubiquitous in the field of computer vision. Deep learning has surpassed other approaches for image segmentation. Semantic image segmentation is the task of classifying each pixel in an image from a predefined set of classes. This paper introduces a simple model U-Net architecture that would filter the microscopic images. The proposed model includes copy connect (concatenation) layers, which is the backbone of ReLUNet architecture. The model possesses an encoding part (contraction part) and decoding part (expansive part) that capture the context with sampling layers and merging layers for a pixel-wise segmentation. The architecture is trained through a dataset of microscopic images and tested through their corresponding masks. Segmentation of a 256x256 image takes some micro seconds on a recent GPU. The U-net achieves 89.9% IoU (Intersection of Union) scores and 10% for testing the model in python IDE. It indicates that the proposed method has excellent performance in measurement accuracy and efficiency.

**Keywords:** CNN, U-Net, Semantic segmentation, Microscopic image, ReLUNet.

### INTRODUCTION

Image segmentation is a task in computer vision; it aims to identify groups of pixels and image regions that are similar and belong together. Different similarity measures can be used for grouping pixels; this includes texture and color features. The process of linking each pixel in an image to a class label is referred to as semantic segmentation (Ciresan *et al.*, 2012). The label could be, for example, cat, flower, lion etc. Semantic segmentation can be thought of as image classification at pixel level (Simonyan *et al.*, 2015). Therefore, in semantic segmentation, every pixel of the image must be associated with a certain class label. One of the approaches used in image segmentation is the encoder - decoder method. In (Zeiler *et al.*, 2014) the encoder is made up of a set of layers that extract features from an image using filters. In many cases, the encoder is pre-trained in a task such as image classification where it learns the correlations from multiple images. This knowledge can then be transferred during the process of image segmentation. The final output, which is a segmentation mask, is generated by the decoder. The decoder is made of layers whose responsibility is to generate the final output. The decoder also ensures that the generated mask bears resemblance to the

pixel resolution of the input image (Ren *et al.*, 2018). The sample of segmentation model is presented in Figure 1.

### Semantic Segmentation using U-Net Architecture

U-NET is an architecture developed by Olaf Ronneberger *et al.* (2015) for Biomedical Image Segmentation in 2015 at the University of Freiburg, Germany. It is one of the most popularly used approaches in any semantic segmentation task today. It is a fully convolutional neural network that is designed to learn from fewer training samples. It consists of an encoder (left side) and decoder (right side). The encoder follows the typical architecture of a convolutional network. It consists of the repeated application of two 3x3 convolutions (unpadded convolutions), each followed by a Rectified Linear Unit (ReLU) and a 2x2 max pooling operation with stride 2 for down-sampling. At each down-sampling step we double the number of feature channels. Every step in the decoder consists of an up-sampling of the feature map followed by a 2x2 convolution (“up-convolution”) that halves the number of feature channels, a concatenation with the correspondingly

\*Corresponding author. E-mail: [ks.muhammad@fubk.edu.ng](mailto:ks.muhammad@fubk.edu.ng)

cropped feature map from the decoder, and two 3x3 convolutions, each followed by a ReLU (Ronneberger *et al.*, 2015). The U-Net also applied in a Fringe Phase Extraction Method based on Neural Network, from which a phase map can be generated, and phase information can be converted into the desired parameters (Hu *et al.*, 2021). According to (Crespo *et al.*, 2020) and (Feng *et al.*, 2021) the

architecture is applied for Fringe Pattern Denoising using U-Net based Neural Network for Fringe visibility and noise removal and generalized framework for non-sinusoidal fringe analysis. In this paper, a model is developed to filter 256 by 256 greyscale microscopic image, the coding procedure and flow chat demonstrate it. The network architecture is illustrated in Figure 2

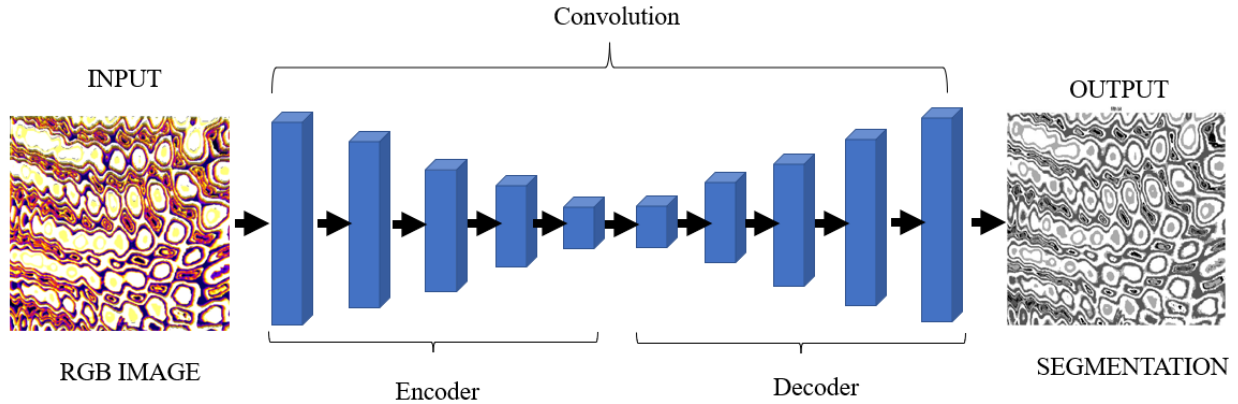


Figure 1: Image Segmentation model

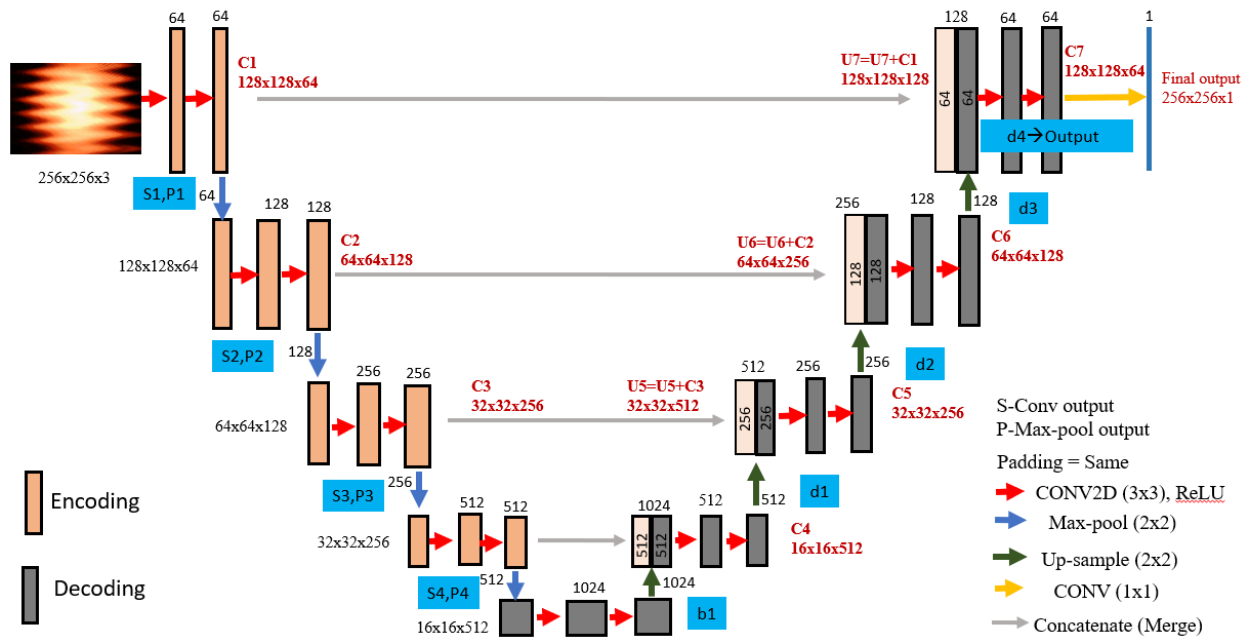


Figure. 2: 256x256 U-Net architecture. We improve (Ronneberger *et al.*, 2015).

## MATERIALS AND METHODS

According to Kevin *et al.* (2010) and the flowchart illustrated in Fig. 3, the original image is divided into individual patches of 256 by 256 to their corresponding masks and the mask are divided the same way so that your mask matches your images. The function in a python file is created called a simple U-net model and defined its parameters together with three input which are image height, width, and channels. Another file ID is created in the same python IDE called the main code file. The modelling is started by importing the function (Dzung *et al.*, 2000) called the simple U-Net model. All the relevant libraries were imported including the OS library, which is basically for navigating throughout directories, CV2 and pillow libraries were used to open and resize the images respectively. The image and mask directories were named and then all the images were added as an empty list and defined the sizes to 256, in case of non-uniform size. Starting the empty lists image dataset in massive data sets so that all the images will be captured into the list and then read and convert to the pillow then to a NumPy arrays. Thus, the lists of all the images and their masks were returned respectively. The resultant image is converted to a NumPy arrays so that it could be applicable for the deep learning. The images are normalized to 8-bit images, meaning the values go from zero to 255 of all the scale. Now the image dataset is an area of low 64 after normalization, hence each of the images that is 256 by 256 at the input would be expanded to the dimensions by 1 or 3, for gray scale or color images respectively (Jonathan *et al.*, 2015). The same procedure is applied for the mask since the format both are the same.

## RESULTS AND DISCUSSIONS

According to the images depicted in Figure 2, the original image was splits into training and testing datasets by holding 10% for testing throughout, especially during semantic segmentation. A sanity

check is always recommended to avoid crappy results which lead to mismatching between the image and their corresponding mask before the model summary is printed. In the convolutional layer, 256 by 256 generates 16 filters during convolution. In that convolution, this is dropping out from 256 to 128 to 64 to another 16 down to 16 by 16 by 256 which is a very small image size at the base of the contraction path. It is then reconstructed in the expansive path back to 256 by 256 by 1 which is the original image size at the output with aid of concatenation during each decoding stage. The expansion and contraction processes for the encoding and decoding parts are presented in Tables 1 and 2 respectively. According to the Table 1, the output image dimension at each stage of both encoding and decoding sections is considered as the input of the proceeding stage. At the last stage in the decoding section, the 2D output image is achieved. Tables 1 and 2 can be explained respectively as follows by considering the Figure 2.

Table 1 describes the contraction path (encoding section) of the U-net model with 256x256x3 as the input. Initially 64 filters would be produced in the convolution stage making the image to be 128x128x64 during the max-pooling. 128 filters were produced in the second convolution, consequently the image become 64x64x128 during max-pooling. In the third stage, 256 filters were produced during the convolution, making the image to be 32x32x256 for max-pooling. At the base, the image is contracted to 16x16x512 because of 512 filters produced in the convolution after max-pooling. Table 2 describes the expansive path (decoding section) of the U-net model, where the encoded image tensor (i.e. 16x16x512 and 512x512x1024) is reconstructed back to its original image size and print the model at the end as 256x256x1 with help of concatenation. The 32x32x256 image is decoded to 256x256x1 as final output through up-sampling of 64x64x128 and 128x128x64 respectively.

**Table.1:** Encoding section of a U-Net Model

S/N	Input(3x3)	CONV2D	Max-pool output(2x2)	Output
1	256x256x3	64x64	64	128x128x64
2	128x128x64	128x128	128	64x64x128
3	64x64x128	256x256	256	32x32x256
4	32x32x256	512x512	512	16x16x512

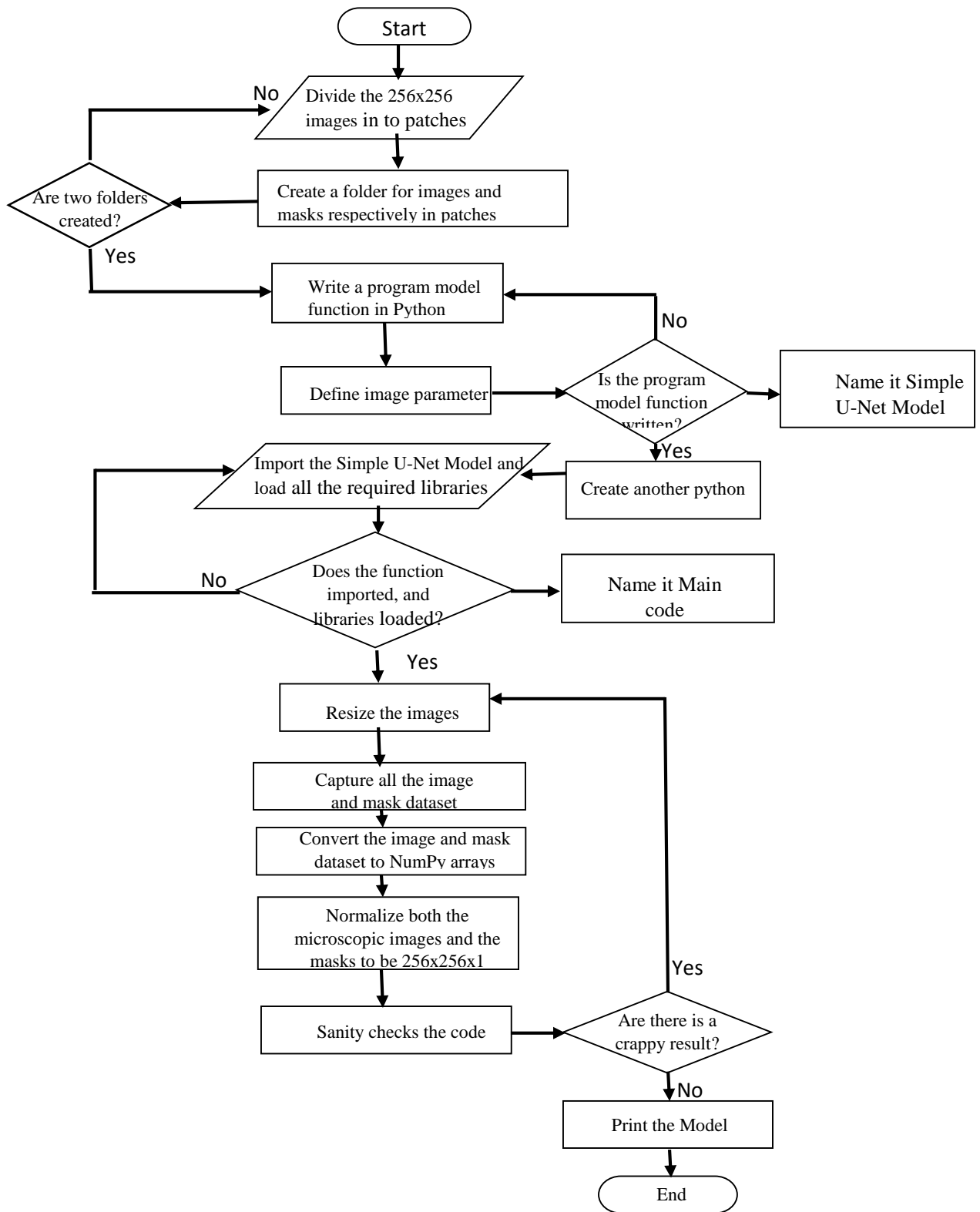
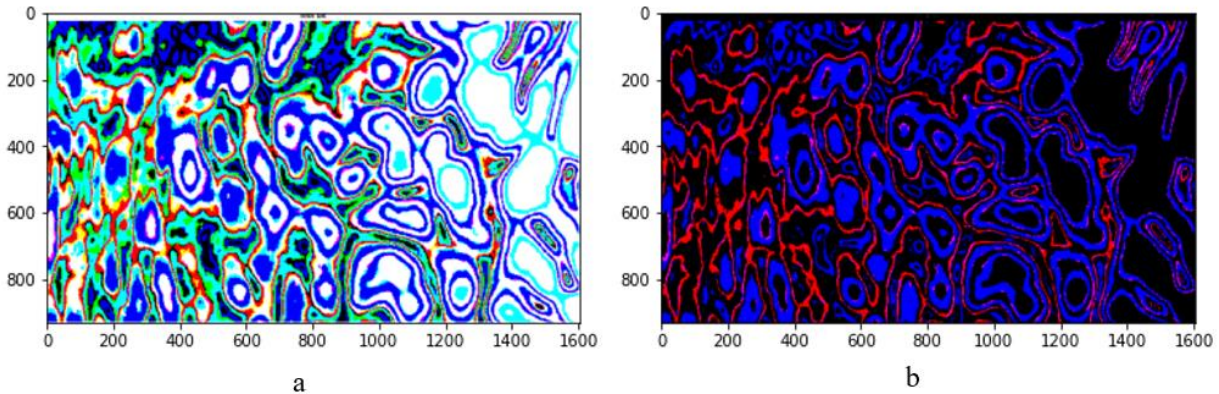


Figure 3: A Flowchart Showing the Model Method of Sematic Segmentation for Microscopic Image Filtering.



**Figure 4:** The 2D image of the spirulina cell of (a) Original and (b) Trained image

**Table 2:** Decoding section of a U-Net Model

S/N	Input	CONV2D	Up-Sampling output(2x2)	Output
1	16x16x512	1024x1024	1024	512X512
2	512x512x1024	512x512	512	16x16x512
3	32x32x512	256x256	256	32x32x256
4	64x64x256	128x128	128	64x64x128
5	128x128x128	64x64	64	128x128x64
6	128x128x64	1x1(Conv:1x1)	1	256x256x1

## CONCLUSION

The U-net architecture achieves very good performance on very different microscopic image segmentation applications. The method was programmed in Python based on the Tensor Flow framework, it only needs very few annotated images with a very reasonable training time on a computer equipped with Nvidia and GeForce graphic card with GPU of at least 2GB. Alternatively, the cloud training with the same code could be done with help of framework developed by Google with dedicated GPU. It is sure that the u-net architecture can be applied easily in the services robot domain for object segmentation.

## REFERENCES

- Ciresan, D., Giusti, A., Gambardella, L. and Schmidhuber, J. (2012). Deep neural networks segment neuronal membranes in electron microscopy images. *Advances in Neural Information Processing Systems*, 25:2852-2860.
- Crespo, J. M., Moreno, V., Rabuñal, J. R., Pazos, A. and Carbia, M. C. (2020). Fringe Pattern Denoising using U-Net based neural network. In *EPJ Web of Conferences* (238: 06009). EDP Sciences.
- Feng, S., Zuo, C., Zhang, L., Yin, W. and Chen, Q. (2021). Generalized framework for non-sinusoidal fringe analysis using deep learning. *Photonics Research*, 9(6): 1084-1098.
- Hu, W., Miao, H., Yan, K. and Fu, Y. (2021). A Fringe Phase Extraction Method Based on Neural Network. *Sensors*, 21(5):1664.
- Long, J., Shelhamer, E. and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431-3440).
- McGuinness, K. and O'connor, N. E. (2010). A comparative evaluation of interactive segmentation algorithms. *Pattern Recognition*, 43(2): 434-444.

- Pham, D. L., Xu, C. and Prince, J. L. (2000). Current methods in medical image segmentation. *Annual Review of Biomedical Engineering*, 2(1): 315-337.
- Ren, Y., Peng, L., Bai, W. and Yu, J. (2018). A Practical Study Of Channel Influence On Radio Frequency Fingerprint Features. 2018 IEEE International Conference on Electronics and Communication Engineering (ICECE). doi:10.1109/icecome.2018.8644931
- Ronneberger, O., Fischer, P. and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (pp. 234-241). Springer, Cham
- Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision* (pp. 818-833). Springer, Cham.