Introducing “W-Nets”: A modification in convolutional network (U-Nets) for bio-medical image segmentation

Qaiser Mehmood

PhD Student

Istanbul Medipol University, Electrical/Electronic Engineering Department. Istanbul

qaiser.mehmood@std.medipol.edu.tr

ABSTRACT

U-Nets are designed for semantic segmentation of images, especially in cases where number of trainable images is less comparatively. The method works fine for semantic segmentations with very good results as are available in different literature discussed. In this document, a modification was done in the U-Nets architecture to see the accuracy of the model in comparison to the original model. The modification is a concatenation of U-Nets to make a W-Nets instead of traditional modification in terms of depth in the number of layers in the deep net. The experiment was performed on the same dataset with same number of images on both U-Nets and W-Nets. It was found that, for the data of 114 trainable images, W-Nets performed slightly better in terms of accuracy in comparison to the U-Net. Although the result is a positive in terms of accuracy, still a further experimentation in terms of different datasets, devices (good GPUs), number of epochs and other parameters can be done to further investigate the proposed W-Nets. However, the initial results in terms of accuracy on a CPU machine implies that W-Nets might be given a chance for experimentation.

Introduction

Medical images are very important source of medical diagnosis procedure and digital image analysis is a great help in this perspective [1]. There has been significant research over the digital image analysis algorithms [1]. Most importantly, automatic image analysis to extract the important information has been significantly improved over the last few years [1]. One of the basic steps of such automatic analysis is image segmentation [1]. Image segmentation is a subdivision of an image based on visual regions that are distinct and have some semantic meaning of the given problem [1].

There are many different types of techniques for image segmentation that are discussed in literature and they can be grouped into three categories [1]. 1) Manual segmentation, 2) semi-automatic segmentation and 3) fully automatic segmentation [1][2]. Deep learning approach is mostly used

for the fully automatic segmentation [1] and for the purpose of this research, we shall be focusing on automatic image segmentation.

There are different deep learning architectures and there are different approaches for implementing deep learning architectures [1]. One such approach is called as U-Nets [1]. U-Nets is an extension or modification of 'fully convolutional network' in such a way that it can work with small number of images and produces more precise segmentation [3].

U-Net architecture was design to improve two aspects of previous research [3]. One of them was to improve the localization or labeling supposed to be assigned to each pixel, and secondly, medical images has always been a challenge in terms of availability of number of images in a particular scenario. So, U-Net architecture is aimed to work with fewer number of images and get desired segmentation results [3].

Literature Review

In literature, we can find different modifications in U-Nets like coupled U-Nets [4], graph U-nets [5] and stacked U-Nets [6] and some of others as well that describe different perspective of U-Nets for different data sets. But in this research project, we aim to study U-Nets architecture for semantic image segmentation and analyze the architecture for any possible improvements for more accurate results. There are very few discussions in literatures about U-Net architecture and its improvements in perspective of semantic segmentation. It might be because the researchers might be satisfied with the results they get from U-Net's architecture already. However, since their introduction in 2016. Most of them use U-Net or discuss the model in relation to other convolutional models like [1]. But for analysis of the model, as to best of my knowledge, there is no particular discussion that have had analyzed the model as we intend to.

Data

This data set is taken from Kaggle.com data since bowl 2018 [7]. The images in this dataset are segmented nuclei images. Cell types may vary as the acquiring of images was done under different conditions.

After downloading data set it was found that there are two main folders that contain the relevant images. The names of the folders are "***stage1_train"*** and "***stage1_test"*** respectively. It is important to mention these details because the images present in the dataset were not organized in a user-friendly manner. For example, if we open-up either of these folders, we find some randomly meaningless names of the folders as shows in Figure1.

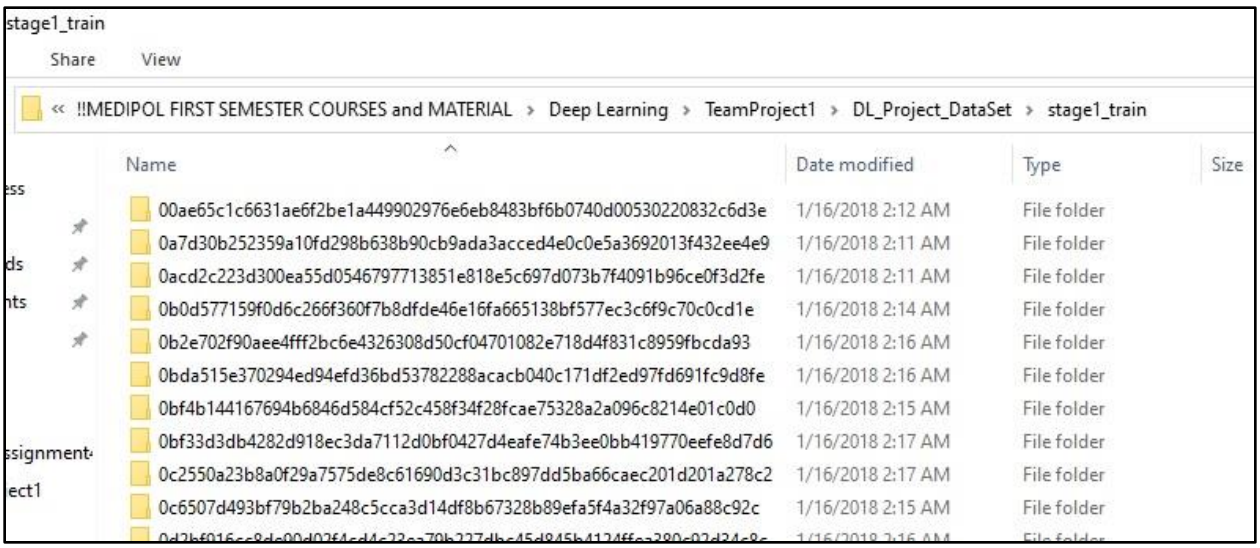

Figure 1 Meaningless Folder Names inside dataset

As a result, an extra bot of code was written to properly fetch and manage the data into desired form. Every weirdly named folder in trainable folder contained two sub folders; ‘image’ and ‘mask’. The folder named as ‘image’ contains only one item, an image, but the folder named as ‘masks’ contains about 70 items. This was a difficult situation because we needed to gather up this 70 images data into one image that should contain all the masks. With the help of YouTube tutorials of different tutors, a solution was managed regarding this situation [8].

Also, there were about 670 trainable images but we used 114 because of not being able to upload data on google drive. It took so long to upload. Also, training time for 670 images was much to high on the device in use. Therefore 114 images were used for both the original model and the test model to fairly compare the results.

Method

We intend to experiment with the U-Net architecture to see and analyze the changes in results. To achieve this, first we shall build a case model as built by [3] and visualize the results. After that we shall experiment with the architecture to achieve more accuracy, if we could. At the end, we shall analyze the experimentation results.

Figure-2 shows the detailed U-Net Architecture [3]. The size and number of layers at each step shows the depth of the architecture and the overall architecture makes a U shape. The layers are successive and up-sampling operators are used instead of pooling operators [3].

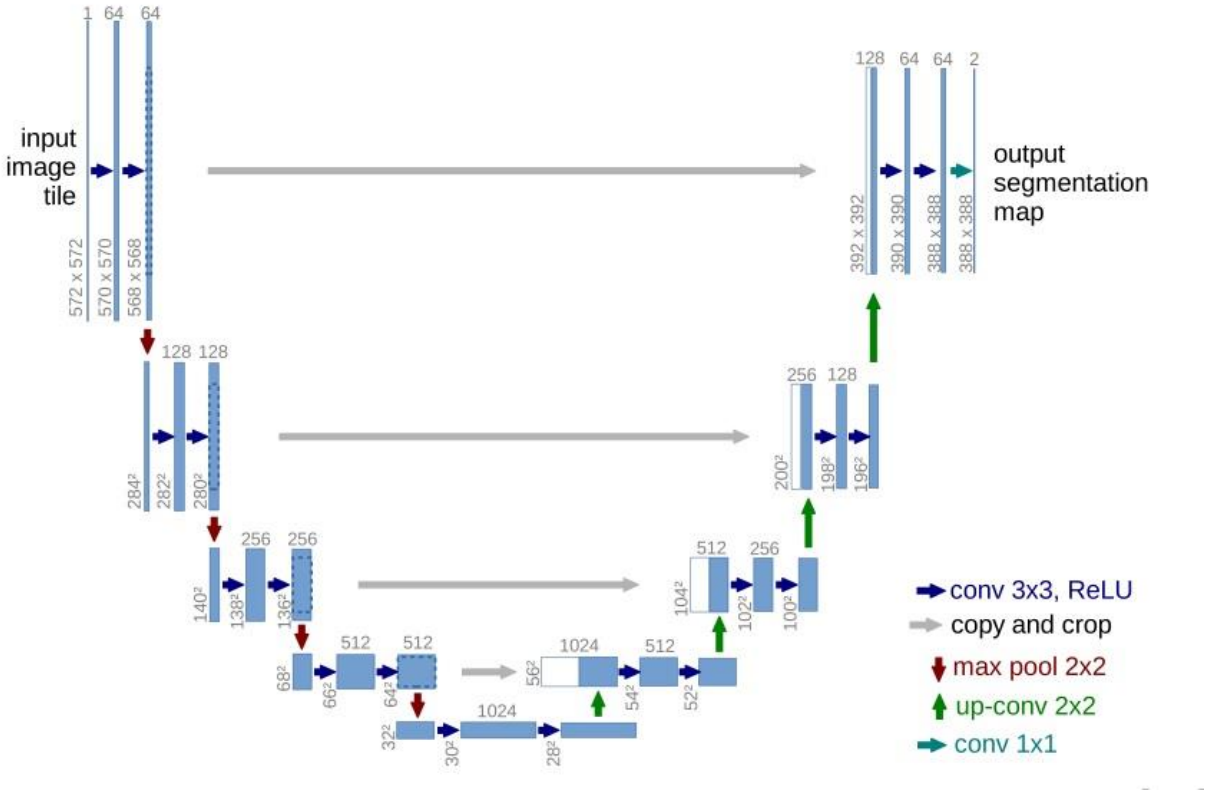

Figure 2 U-Net Architecture

Figure 2: U-Net Architecture

The segmentation of the images will be carried out using same methodology as done by [3]. However, our experimentation intends to manipulate the architecture using implementation methodology to see some different results. Therefore, the research methodology that we use during this project is a conjunction of experimentation and implementation both, similar to the one used in [3].

Experiments

First, we shall implement the entire model, using Python and Keras as deep learning implementation tools, most probably using same dataset as was used in [3]. We intend to get the same results initially as of [3] because we need those results for validation of results with our modifications. Secondly, we shall experiment some changes in the architecture and see the results. We intend to improve the results with this experimentation so that the model can be improved accordingly. However, we intend to analyze the results and the architecture on the basis of whatever the result might be. Also, we shall study and discuss the reasons if we could not improve the results.

1. Implementing Original Model (U-Nets)

So, first we implemented the original model with a slide bit of modification in the number of layers because of the normal computer system we are using for this implementation (I do not have access

to a physical GPU, I use my simple normal laptop computer and borrow GPU of google Colab). Figure 3 shows the modification made in the original model (Figure 2) to accommodate our device in use. So, the model implemented is what is visible in Figure 3.

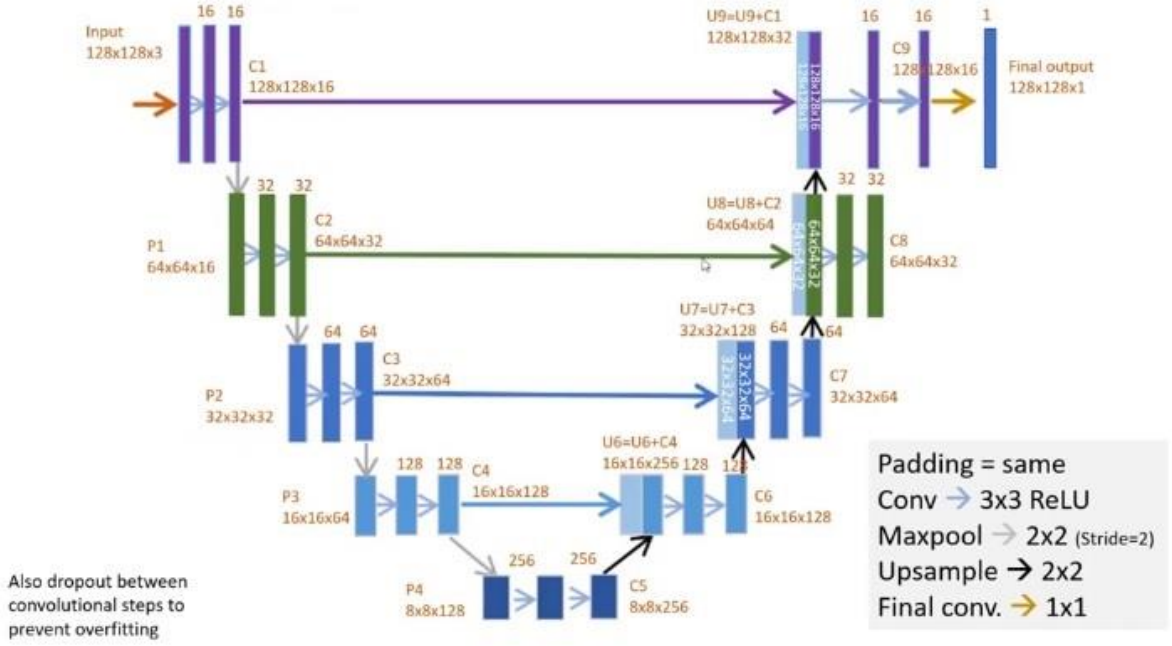

Figure 3 Easy version of U-Net implemented

The code of this model is attached. Detailed model summary is also attached with another file. Here, just the total parameters and the trainable parameters are shown, as below.

a. Parameters

```
Total params: 1,941,105
Trainable params: 1,941,105
Non-trainable params: 0
_________________________________________________________________
_________________________________
```

This model gave us a total of about 2 million parameters. Also, these are the trainable parameters as well.

b. Outcome

The model was executed for 30 epochs. This number was set because of the limitations of our computer specifications and also, we set 'patience' to 5 so that all 30 epochs run unless there is tight accuracy at each epoch. Here we show only the 30 epochs, rest of 29 epochs are in the attached document. Also, we find the time taken to perform each step by the machine in use.

```
Epoch 30/30
```

```
7/7 [==============================] - 0s 38ms/step - loss: 0.1116 -
accuracy: 0.9578 - val_loss: 0.1116 - val_accuracy: 0.9580
4/4 [==============================] - 1s 11ms/step
1/1 [==============================] - 0s 15ms/step
3/3 [==============================] - 0s 141ms/step
```

Two randomly selected images are shown below. Each of the images first shows the original, then it shows the semantic segmentation performed in grayscale and then it shows the semantic segmentation performed in color.

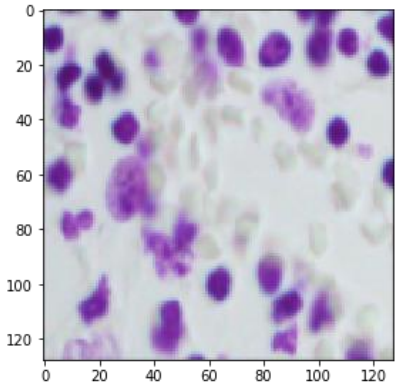

Figure 4 Randomly selected image 1

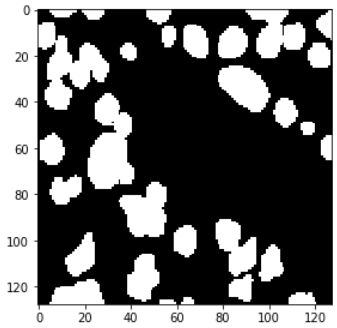

Figure 5Randomly selected image 1 (gray scale semantic segmentation)

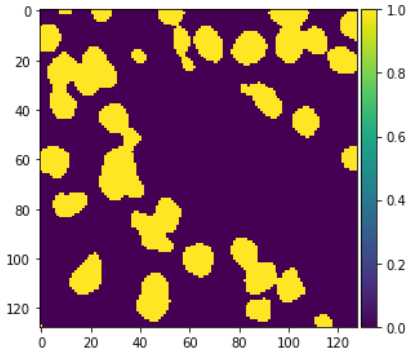

Figure 6 Randomly selected image 1 (color semantic segmentation)

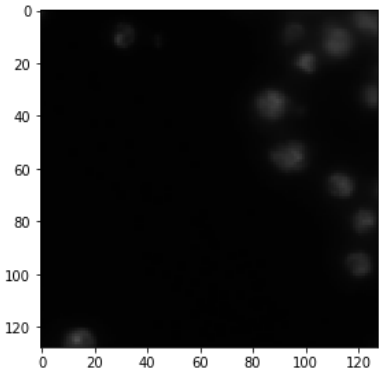

Figure 7 Randomly selected image 2

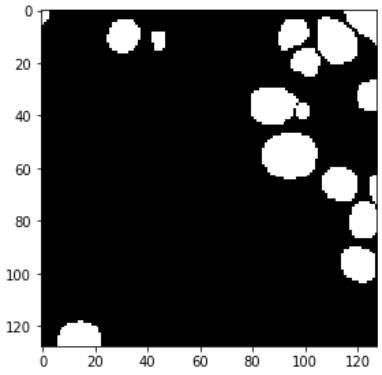

Figure 8 Randomly selected image 1 (gray-scale semantic segmentation)

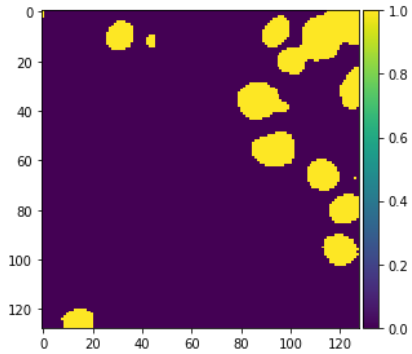

Figure 9Randomly selected image 1 (color semantic segmentation)

2. Implementing Modified Model (W-Nets)

The modified model was proposed to see a concatenation of the entire architecture instead of the traditional concatenation of deep layers in the deep net. Since U-Nets are by design architected to deal with the smaller number of images and a greater number of trainable parameters, so the modified model intends to repeat the process twice to see the increased number of parameters and more importantly to see the validation accuracy. Just because the model looks like the letter 'W', so it is named as "W-Nets", as shown in Figure 10.

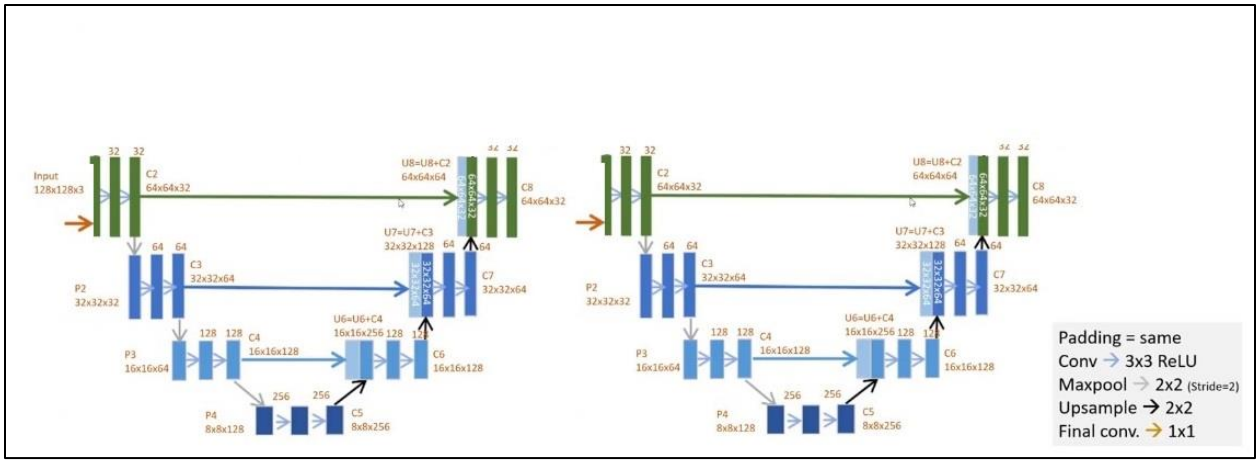

Figure 10 Proposed W-Nets

Again, details of the code and model summary are provided in the attached files, but here we just show the total number of parameters generated and the accuracy at 30 epochs along with a couple of randomly selected outcome images with their gray scale and color semantic segmentation.

Parameters

As, shown below, the total number of trainable parameters about now about the double of the previous model.

```
Total params: 3,859,521
Trainable params: 3,859,521
Non-trainable params: 0
_________________________________________________________________
_________________________
```

This implies that the number of parameters will be doubled every time we increase a ‘u’ at the end of the architecture.

Discussion

Although it was just experimental and with increased number of trainable parameters on this machine, it was not expected that the code will run all along, yet the code ran fine and it took a small bit of more time but more importantly, the validation accuracy increased by about 1 percent.

```
Epoch 30/30
7/7 [==============================] - 1s 127ms/step - loss: 0.0866 -
accuracy: 0.9652 - val_loss: 0.0994 - val_accuracy: 0.9603
4/4 [==============================] - 2s 42ms/step
1/1 [==============================] - 0s 15ms/step
3/3 [==============================] - 1s 251ms/step
```

Might be at this moment of time it cannot be considered as a major success, however the working of proposed “W-Net” slightly better in terms of accuracy was a positive achievement. To further

test the working of “W-Nets”, there is a need of further experimentation with different datasets and different number of images and different number of epochs, depending upon the devices available for this purpose. But the initial results suggest that the if accuracy is given the higher priority, then this proposed ‘W-Nets’ model might be given a chance in terms of a comprehensive testing in comparison to U-Nets. Figure 11 to Figure 13 show a randomly selected image from same datasets of 114 images and show original, grayscale semantic segmentation and color semantic segmentation respectively. Similarly, Figure 14 to Figure 16 show another randomly image.

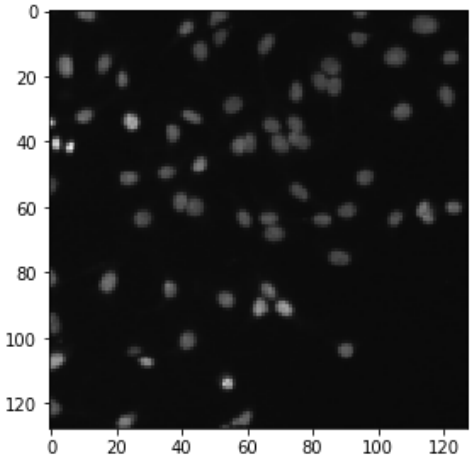

Figure 11 Randomly selected image 1 (original)

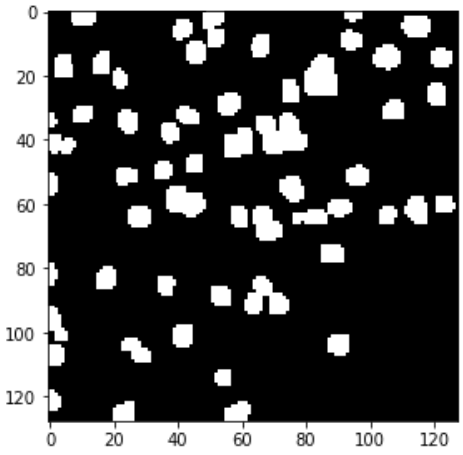

Figure 12 Randomly selected image 1 (color semantic segmentation)

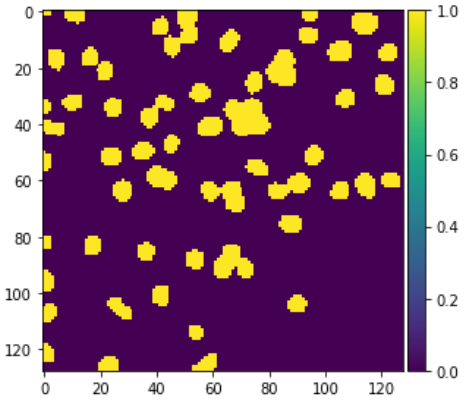

Figure 13 Randomly selected image 1 (color semantic segmentation)

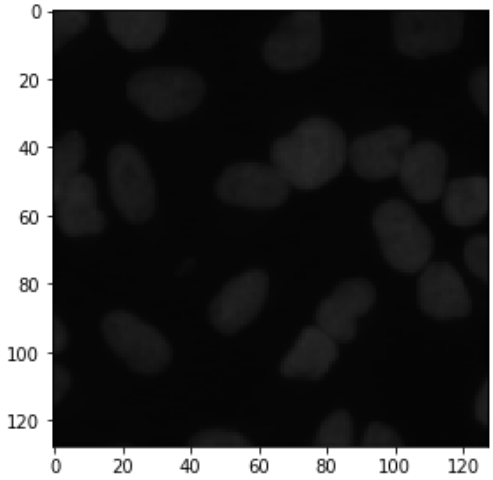

Figure 14 Randomly selected image 2 (original)

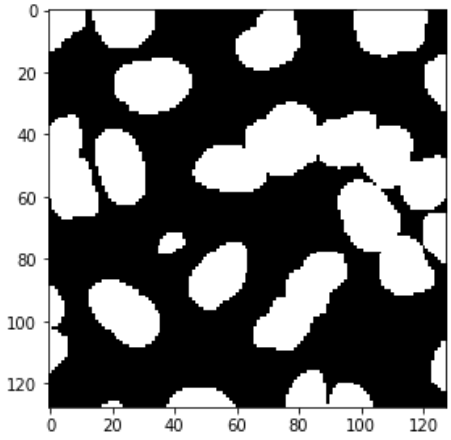

Figure 15 Randomly selected image 2 (gray-scale semantic segmentation)

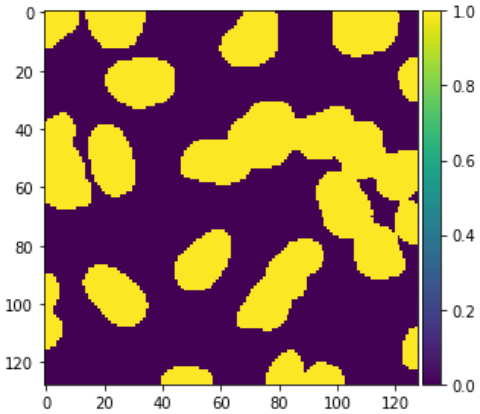

Figure 16 Randomly selected image 2 (color semantic segmentation)

Analyzing images of both models

It we analyze the semantic segmentation done by both the models to these images, we might conclude that visually, there might not be a very major difference visible, however, the accuracy measure tells us that, the semantic segmentation done by the W-Nets, is slightly better. Images of both the architectures look similar though in terms of grayscale and color semantic segmentations done.

Conclusion

U-Nets are designed for semantic segmentation of images, especially in cases where number of trainable images is less comparatively. The method works fine for semantic segmentations with very good results as are available in different literature discussed. In this document, a modification was done in the u nets to see the accuracy of the model in comparison to the original model. The modification is a concatenation of U-Nets to make a W-Nets instead of traditional modification in terms of depth in the number of layers in the deep net. The experiment was performed on the same dataset with same number of images on both U-Nets and W-Nets. It was found that, for the data of 114 trainable images, W-Nets performed slightly better in terms of accuracy in comparison to the U-Net. Although the result is a positive in terms of accuracy, still a further experimentation in terms of different datasets, devices (good GPUs), number of epochs and other parameters can be done to further investigate the proposed W-Nets. However, the initial results in terms of accuracy on a CPU machine implies that W-Nets might be given a chance for experimentation.

Acknowledgement

YouTube is a great source of knowledge and a lot of help is taken from this platform in terms of specific topic as well as general knowledge of the field. We specially acknowledge a YouTube channel named as "DigitalSreeni" and the tutor "Sreenivas Bhattiprolu" [7][8]. The material provided on this channel by the tutor was very helpful.

References

[1] Haque, I. R. I., & Neubert, J. (2020). Deep learning approaches to biomedical image segmentation. *Informatics in Medicine Unlocked*, *18*, 100297.

[2] Işın, A., Direkoğlu, C., & Şah, M. (2016). Review of MRI-based brain tumor image segmentation using deep learning methods. *Procedia Computer Science*, *102*, 317-324.

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

[4] Tang, Z., Peng, X., Geng, S., Zhu, Y., & Metaxas, D. N. (2018). CU-net: coupled U-nets. *arXiv preprint arXiv:1808.06521*.

[5] Gao, H., & Ji, S. (2019, May). Graph u-nets. In *international conference on machine learning* (pp. 2083-2092). PMLR.

[6] Shah, S., Ghosh, P., Davis, L. S., & Goldstein, T. (2018). Stacked U-Nets: a no-frills approach to natural image segmentation. *arXiv preprint arXiv:1804.10343*.

[7] https://www.kaggle.com/c/data-science-bowl-2018/data

[8] https://github.com/bnsreenu/python_for_microscopists