



UNIVERSITY OF  
EASTERN FINLAND

## **Industrial Project – Group 6**

Matti Auvinen 276711

Muktar Bello 320521

Pingxin Gao 300279

Awais Hassan

Md Salah Uddin Sajib 308852

25.05.2021

# Contents

1. Introduction.....	3
1.1. Image segmentation .....	3
1.1.1. Threshold-based segmentation .....	3
1.1.2. Region growth segmentation.....	3
1.1.3. Edge detection based segmentation.....	3
1.1.4. Clustering-based segmentation.....	3
1.1.5. Multilayer perceptron .....	4
1.1.6. U-Net .....	5
2. Methods .....	6
2.1. GUI model selection .....	6
2.2. U-Net model structure.....	6
3. Experiments .....	7
4. Results.....	8
4.1. U-Net for image segmentation .....	8
4.2. Graphical user interface .....	9
5. Conclusion .....	10
References.....	11
Appendix. U-Net model structure.....	11

# 1. Introduction

This project aims to perform image segmentation on spectral images.

## 1.1. Image segmentation

The segmentation of images is the method for separating an image into several areas with special characteristics and representing the objective of interest. This is the important step from image treatment to image analysis.

In general, there are four main categories of algorithms used for image segmentation.

### 1.1.1. *Threshold-based segmentation*

The first one is the threshold segmentation method. In essence, the optimal threshold is automatically determined based on certain criteria and these pixels are used to achieve clustering based on gray levels. The basic theory for this method is the calculation of one or more thresholds of grayscale thresholds based on grayscale characteristics of each pixel and the comparison of the threshold value with the grayscale in the image<sup>1</sup>.

### 1.1.2. *Region growth segmentation*

Next is region growth segmentation. The core idea of such an algorithm is to combine pixels with similar attributes to form regions, i.e., first divide each region to find seed pixels as growth points, and then merge surrounding neighborhoods with similar attributes to merge pixels in their regions<sup>2</sup>.

### 1.1.3. *Edge detection based segmentation*

The third one is the edge detection segmentation method. In the frequency domain converted by Fourier transform, the grayscale value of pixels on the boundary of different regions usually varies dramatically, high frequency component is linked to the edges<sup>3</sup>. Another algorithm is the differential operator method, which uses the nature of discontinuity of pixel values in adjacent regions and uses first- or second-order derivatives to detect edge points<sup>4</sup>. The characteristics of edge detection include accurate and fast edge localization, but it cannot guarantee the continuity and closure of edges. Therefore, edge detection can only produce edge points, rather than an image segmentation process. This means that after the edge point information is extracted, further processing is needed to complete the task.

### 1.1.4. *Clustering-based segmentation*

The fourth type is clustering-based segmentation. The clustering-based algorithm is based on the similarity between pixels as a criterion for classification, i.e., dividing the sample set into subclasses based on their internal structure in order to make classes of the same type as similar as possible and

classes of different types as dissimilar as possible<sup>5</sup>. The general steps include first initializing a random clustering, and then using iteration to cluster pixel points with similar features such as color, luminance, texture, etc., and iterating until convergence to obtain the final image segmentation result.

In this project, we use two machine learning algorithms to perform image segmentation, namely, multi-layer perceptron (MLP), support vector machine (SVM), and U-Net<sup>6</sup>.

### 1.1.5. Multilayer perceptron

The MLP is a forward-structured ANN that maps a set of input vectors to a set of output vectors. It consists of multiple node layers, each fully connected to the next layer. In addition to the input nodes, each node is a neuron with a nonlinear activation function. A supervised learning approach using the BP algorithm is used to train the MLP, which is a generalization of the perceptron and overcomes the weakness that the perceptron cannot recognize linearly indistinguishable data. Compared to single-layer perceptrons, MLP has changed from one to multiple outputs; instead of just one layer between the input and output, there are now two layers: the output layer and the hidden layer.

In an MLP, it is possible to use any kind of activation function, for example, a logistic sigmoid function or a step function. However, it is better to use a function that is easily differentiable due to the use of the BP algorithm<sup>7</sup>. This is the reason why many sigmoid functions are suitable for this task, such as logistic sigmoid function or hyperbolic tangent (tanh) function (shown in Figure 1) due to good differentiability. The key role of the activation function is to introduce nonlinearity to the network.

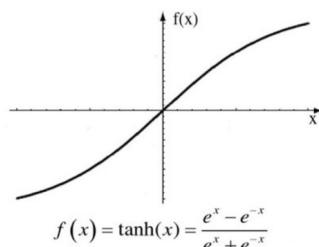
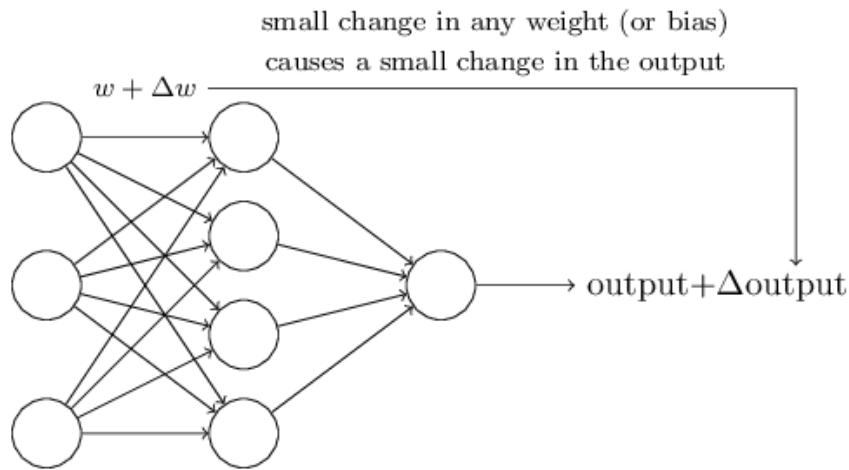


Figure 1. Hyperbolic tangent function.

The structure of a feedforward neural network includes many neurons that are interconnected. And for each connection, weights are assigned and later updated.

- 1) Input layers are responsible for passing the data in the training dataset into the neural network.
- 2) Hidden layers will perform computation and pass information from the input node to the output node. In a feedforward neural network, it is possible to have no or few hidden layers.
- 3) Output layers are responsible for computing and passing information out of the network.

In a feed-forward network, information moves only one way, and there are no loops or circuits in the network.



*Figure 2. Mechanism of MLP.*

The MLP is trained using the BP algorithm. When a mistake happens, the supervisor will correct the network based on the error and after each cycle of training, the weights of nodes are updated as shown in Figure 2. The learning process of a network means finding the correct weights to each of the connections between nodes. The general steps of training MLP are as follows:

- 1) Random assignment of weights to all connections between neurons.
- 2) Forward propagation: The artificial neural network is triggered for all inputs in the training dataset, and then forward propagated to extract the output values, using the input characteristics of all samples in the training data as the input layer.
- 3) Backpropagation: the weights are updated based on the errors which are calculated from the output value.
- 4) Repeat steps 2) and 3), and stop the repetition after the error is low enough.

#### 1.1.6. *U-Net*

In the field of image segmentation, especially medical image segmentation, U-Net is undoubtedly one of the most successful methods, which was presented at the MICCAI 2015 conference<sup>6</sup>. The structure of U-Net is shown in Figure 3, where the left side can be considered as an encoder and the right side as a decoder. The encoder has four submodules, each containing two convolutional layers, and each submodule is followed by a downsampling layer implemented by the max pool. The resolution of the input image is 572\*572, and the resolutions of the 1st-5th modules are 572\*572, 284\*284, 140\*140, 68\*68, and 32\*32 respectively. The decoder contains four sub-modules, and the resolution rises sequentially through the up-sampling operation until it matches the resolution of the input image.

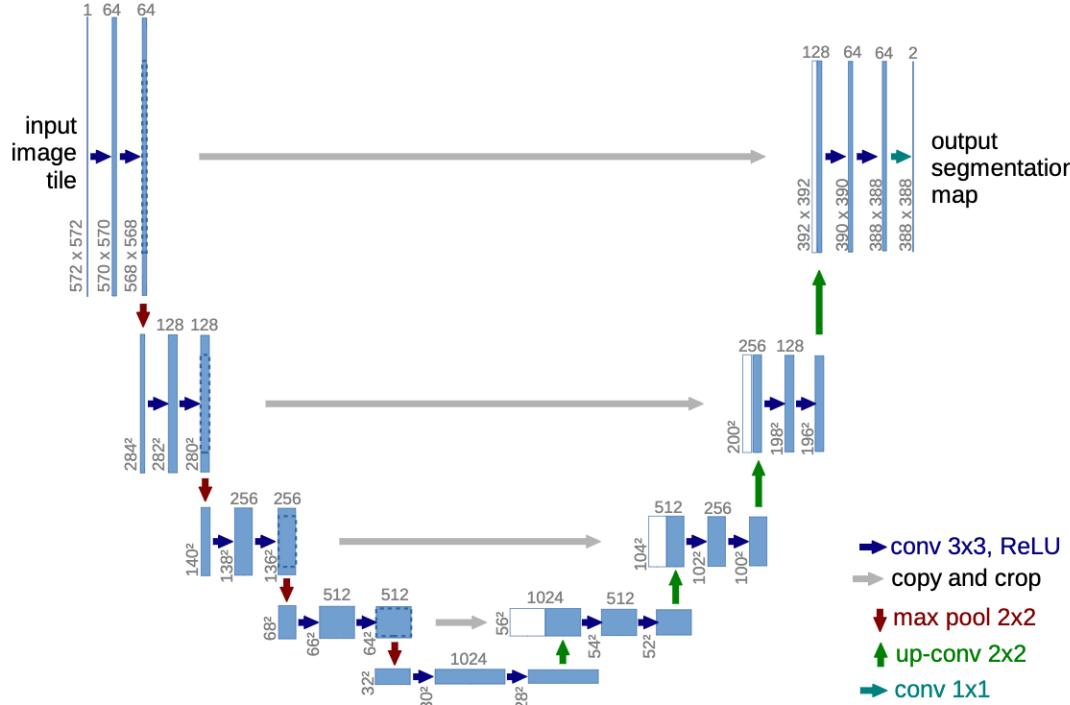


Figure 3. Structure of U-Net.<sup>6</sup>

## 2. Methods

### 2.1. GUI model selection

The GUI was initially being developed in Python using Tkinter. Due to Tkinter's relative simplicity and dated-looking results, the decision was made to explore other options. We ultimately settled on using PyQt and the Qt suite of software to make use of features such as more advanced widget options and the Qt Designer.

Due to time constraints, the GUI functionality was kept to the bare necessities. After some prototyping, the layout seen in the final product was confirmed and implemented using PyQt5.

### 2.2. U-Net model structure

We used U-Net for training it is a convolutional network architecture for fast and precise segmentation of images. The structure of U-Net consists of an encoder and a decoder. The encoder has four submodules, each containing two convolutional layers, and one dropout layer in the middle. Each submodule is followed by a downsampling layer implemented by the max pooling. The resolution of the input image is 128\*128\*38, and the shape of the 1st-5th modules is 128\*128\*16, 64\*64\*32, 32\*32\*64, 16\*16\*128, and 8\*8\*256 respectively. The decoder contains four sub-modules, and the resolution rises sequentially through the up-sampling operation until it matches the resolution of the input image. A more detailed model structure can be found in the Appendix.

### 3. Experiments

We train the model using Adam function as optimizer, and **categorical\_crossentropy** as the loss, while we used loss score as the evaluation metric during the training. We divided the dataset into training (80%) and testing (20%) datasets. The training data was used in training the model, while the test data are unseen datasets to the model which was used to evaluate the model after the training. In order to evaluate the performance of the image segmentation model in training and on the test data, we plot the training loss (Figure 4), Intersection over Union (IoU) (Figure 5), and Confusion Metrics (Figure 6).

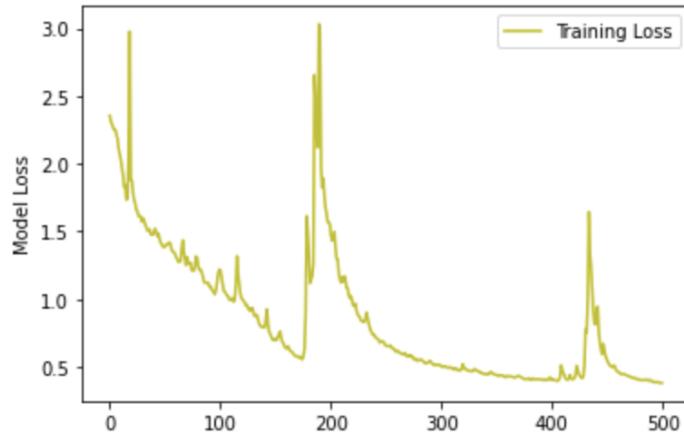


Figure 4. Plot showing the training loss of the model.

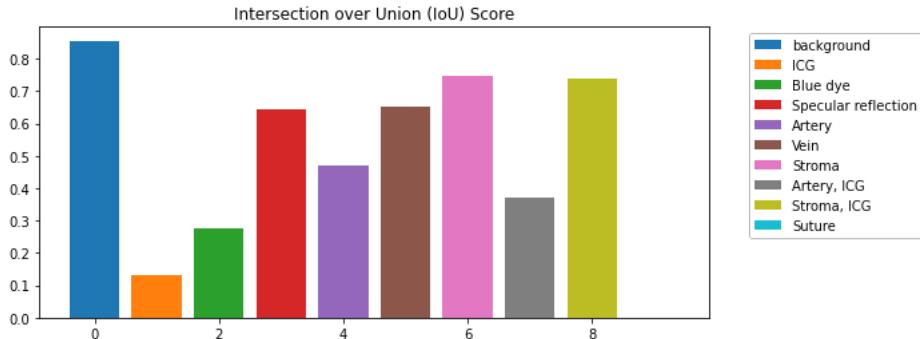
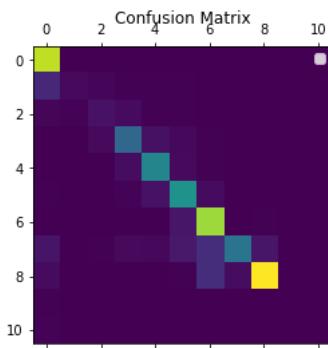


Figure 5. Plot showing the Intersection over Union Score of the model on the test dataset.

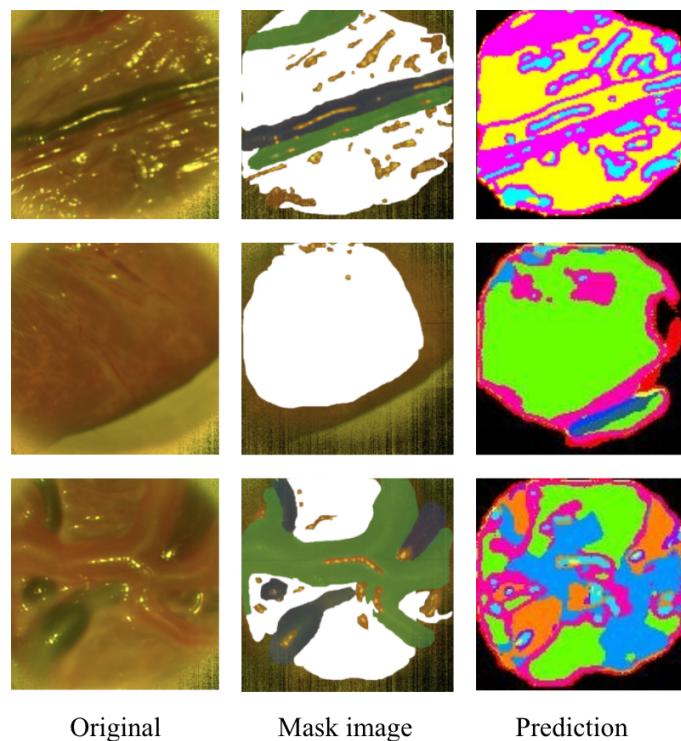


*Figure 6. Plot showing the Confusion matrix of the model on the test dataset.*

## 4. Results

### 4.1. U-Net for image segmentation

After training and multiple runs, we were able to get a success rate of 79 - 83%.



*Figure 7. Side-by-side comparison between original image, mask image, and predicted results.*

The images shown in Figure 7 are the final results of the trained model.

## 4.2. Graphical user interface

As shown in Figure 8, our GUI is simple to use and consists of three parts, file path selection section, original spectral image preview section, and the preview section of the segmented image.

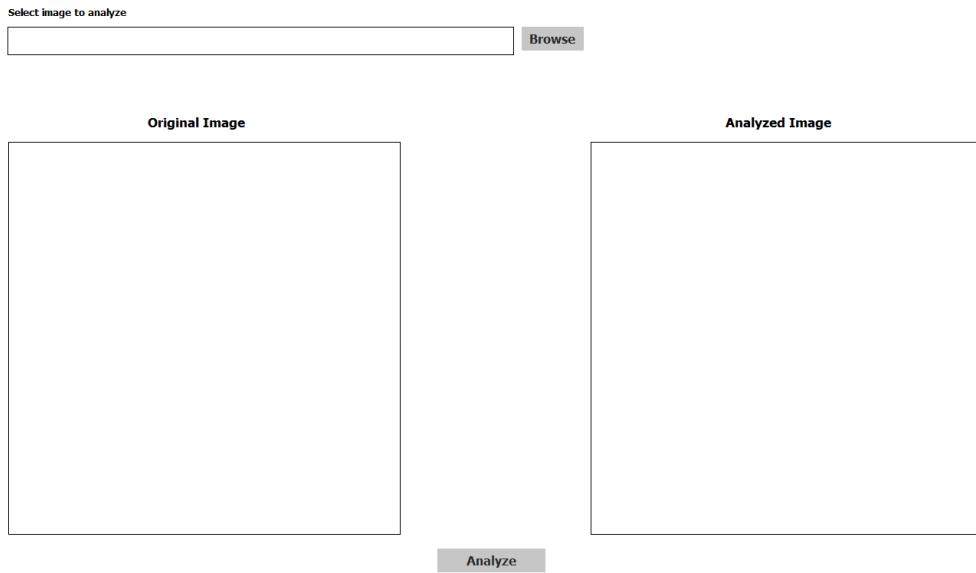


Figure 8. GUI was developed for our U-Net segmentation model.

The simplistic graphical user interface is displayed in Figure 8. The file is selected by clicking on the 'Browse' button at the top-middle of the interface. This opens a dialog window that the user can use to navigate to the wanted file as shown in Figure 9.

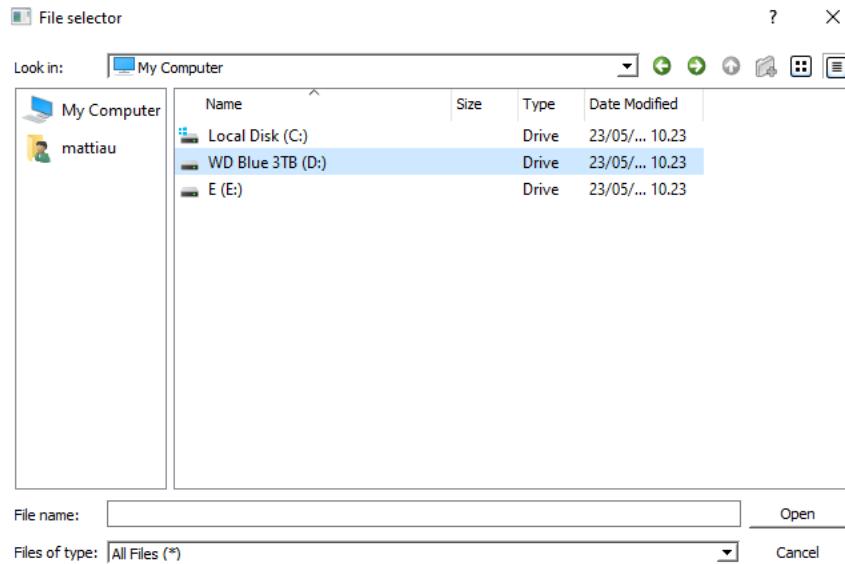
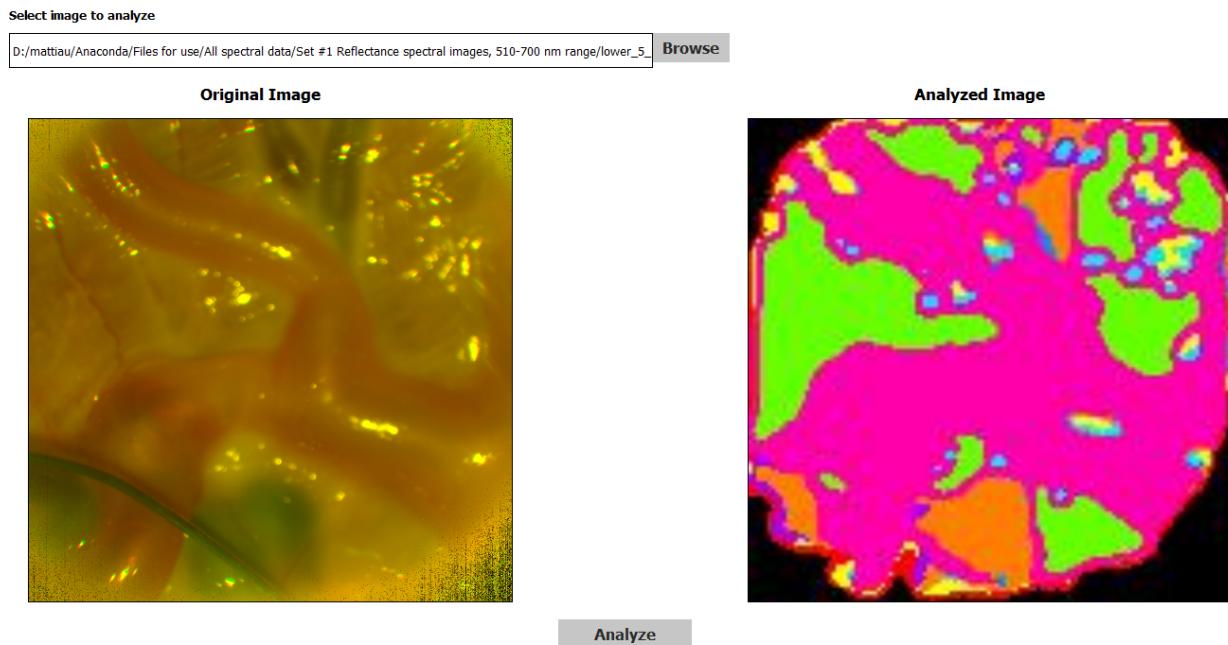


Figure 9. GUI dialog window for navigation.

After a file is chosen, it will appear on the left in the frame labeled ‘Original Image’. Next, the user presses the ‘Analyze’ button and the program starts running the segmentation model. Due to the compressed images used for the segmentation, this should only take less than five seconds after which the segmented image appears on the right. The interface is shown in Figure 10.

The user can continue segmenting new images by repeating the loop. Upon choosing a new image, the frame for the analyzed image on the right will change to be blank until a new segmentation is done.



*Figure 10. GUI after the spectral image file is chosen.*

## 5. Conclusion

We implemented a segmentation method for the spectral image using the Unet architecture. The main idea is for the model to learn the feature mapping of the spectral images and exploit it to make more nuanced feature mapping. We also used sample weights to ensure the metrics derived from the data are representative of all the classes. The experiment results indicate the trained model can achieve 79-83% of segmentation accuracy.

## References

- (1) Pal, N. R.; Pal, S. K. A Review on Image Segmentation Techniques. *Pattern Recognit.* **1993**, *26* (9), 1277–1294. [https://doi.org/10.1016/0031-3203\(93\)90135-J](https://doi.org/10.1016/0031-3203(93)90135-J).
- (2) Seo, H.; Khuzani, M. B.; Vasudevan, V.; Huang, C.; Ren, H.; Xiao, R.; Jia, X.; Xing, L. Machine Learning Techniques for Biomedical Image Segmentation: An Overview of Technical Aspects and Introduction to State-of-Art Applications. **2019**.
- (3) Salem, M.; Al-Amri, S.; Kalyankar, N. V; Khamitkar, S. D. IMAGE SEGMENTATION BY USING EDGE DETECTION. *IJCSE Int. J. Comput. Sci. Eng.* **2010**, *02* (03), 804–807.
- (4) Song, Y.; Yan, H. Image Segmentation Techniques Overview. *AMS 2017 - Asia Model. Symp. 2017 11th Int. Conf. Math. Model. Comput. Simul.* **2018**, 103–107. <https://doi.org/10.1109/AMS.2017.24>.
- (5) Coleman, G. B.; Andrews, H. C. Image Segmentation by Clustering. *Proc. IEEE* **1979**, *67* (5), 773–785. <https://doi.org/10.1109/PROC.1979.11327>.
- (6) Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. **2015**.
- (7) Shenouda, E. A. M. A. A Quantitative Comparison of Different MLP Activation Functions in Classification. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)* **2006**, *3971 LNCS*, 849–857. [https://doi.org/10.1007/11759966\\_125](https://doi.org/10.1007/11759966_125).

## Appendix. U-Net model structure

