Introduction

In this project, UNet deep learning model is implemented for detecting cancer regions in medical pictures. This model serves as a powerful tool for picture segmentation.The importance of the project is that it is to some extent a cumbersome duty to detect the regions by hand in a picture of large size. Therefore, The UNet structure has been developed to perform the job.

About Dataset

The dataset in this project is medical picture of human tissues in .tif format in which some regions contain cancer cells. The dataset are two parts: train and test data. Each of the datasets contain pictures and masks.

Train data consists of 56 medical color pictures and 56 corresponding grayscale masks. A mask somehow plays the role of the label compared to machine learning processes. A mask picture is constructed only by black and white pixels. White pixels stand for the cancer regions of their corresponding color picture.

In order to evaluate the performance of the model. Test data is used which contain only two pictures and their masks. Test data are not used in the training step of the model. You can see a sample of training picture and its mask in the figure below.

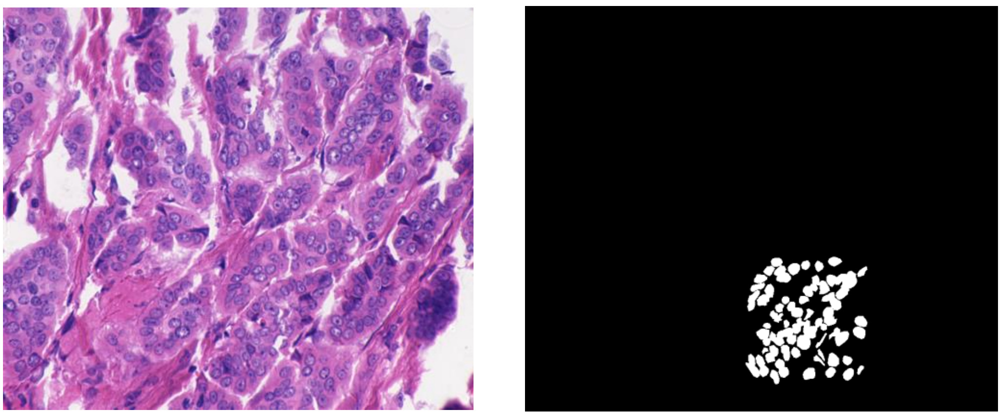

Fig. 1: Medical picture and its mask showing cancer regions.

Algorithm Implementation

In this project, UNet model is utilized for detecting cancer regions of medical pictures. To do so, tensorflow and keras packages are used. In this project, the model consists of 5 layers of contraction (convolution and pooling) and 5 layers of expansion (transpose convolution and unpooling). Eventually, the model output is a grayscale picture in which segmented regions of the picture is illustrated by a pixel of specific grayscale sample.

In our work, UNet model with the following blocks are used.

Convolution blocks:

Input size of pictures: to increase the speed of training, only the cancer region of the original pictures is cropped and resized to a specific size (for instance, 256*256*3) and used as the input picture.

Initial block 1: in this block, 2D convolution consisting 16 filters of 3*3 with activation of ‘elu’ and kernel_initializer of 'he_normal' and no padding is used. Also, 10 percent of the nodes are dropped out in training in each epoch. In the end, 2D maxpooling of size 2*2 is used.

Block No. 2: this block is the same as previous one with just the number of filters changed to 32.

Block No. 3: the same as before but 64 filters are used here.

Block No. 4: 128 filters are used in this block.

Block No. 5: this is the last block of encoding part which has 256 filters. This block doesn’t have maxpooling.

Up convolution blocks:

Block 1: up convolution with 128 filters of size 2*2 with stride of 2*2. Also, 2D convolution of 3*3 and activation of ‘elu’ is implemented. Dropout of 0.2 is used too.

Block 7 to 9: they are the same only with number of filers being 64, 32 and 16, respectively

Finally:

A block consisting of 2D convolution with 1 filter of size 1*1and activation of ‘sigmoid’ is used.

Additionally, number of epochs is 50 and number of patience for early stopping is set to 5 epochs. In each epoch, 10 percent of train data are used as validation data.

Preprocessing

In this section, in order to increase the speed of training, the cancer regions of the pictures are cropped and used as the training data. The following figure shows the results of preprocessing and the corresponding original data.

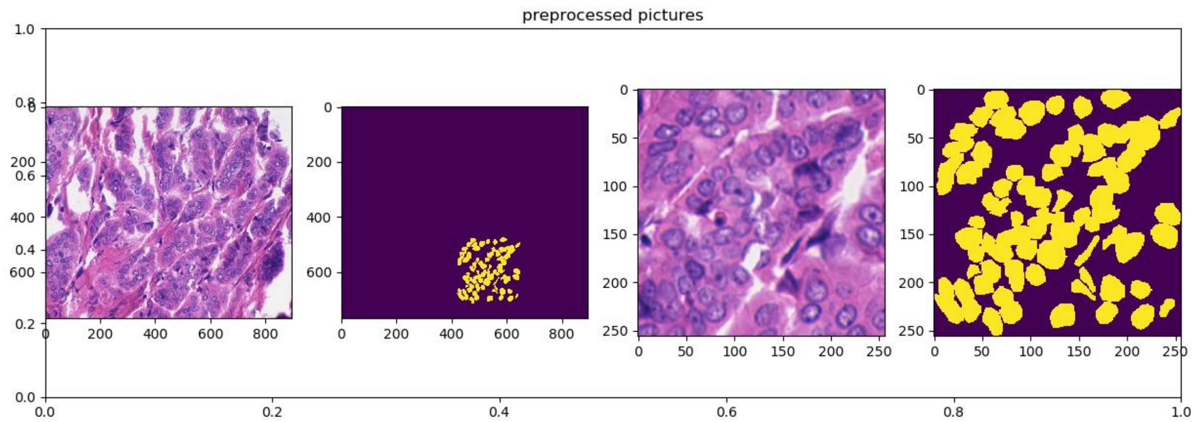

Fig. 1: From left to right: original train picture, corresponding original mask picture, cropped train picture, cropped mask picture.

Results and Conclusion

The best way to analyze the capability of the model in training is calculating IOU and Loss function. The IOU and Loss curves are plotted here.

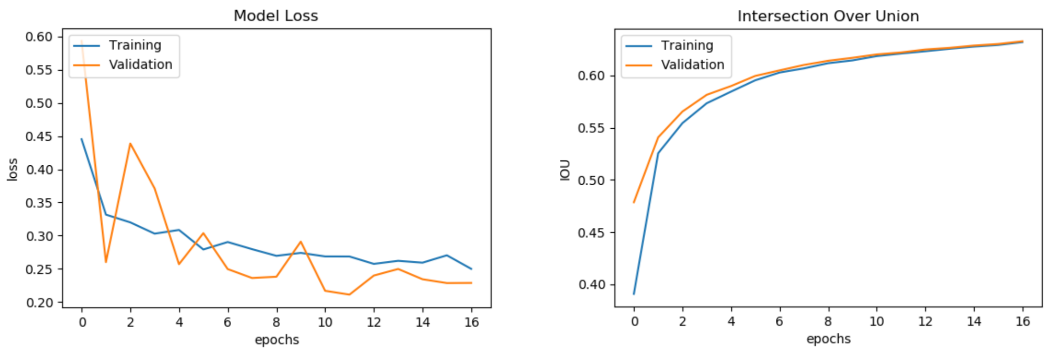

Fig. 1: Loss and IOU of train and validation data.

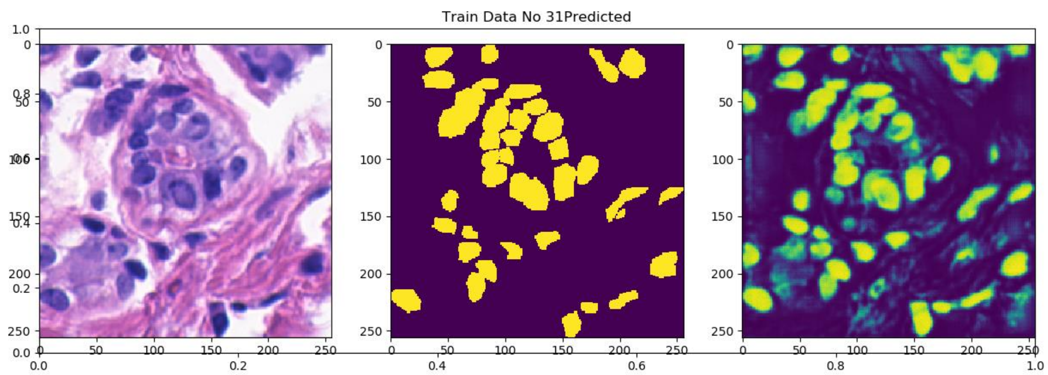

Fig. 1: From left to right: train picture, corresponding mask picture, predicted mask picture by UNet model.

Finally, test data is predicted by the model and the result of prediction is compared with train mask. As it is obvious in the following figure, UNet model has done a fabulous job in segmentation after being trained by a few number of data.

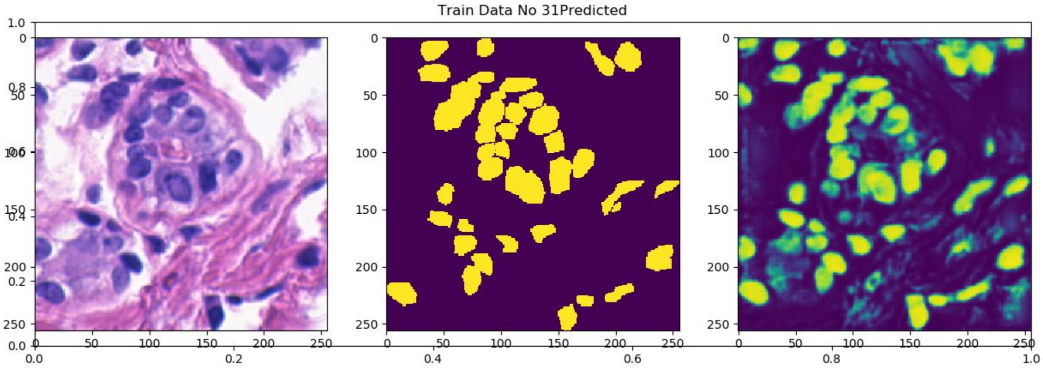

Fig. 1: From left to right: test picture, corresponding mask picture, predicted mask picture by UNet model.

Appendix A: Modules used in coding

The following Modules are used in python codes:

- **Class:** is created and used for better programming quality.
- **Logging:** an enhanced logger file is created to monitor the progression of codes and log any messages emitted from process. The results of run are logged in the file too in a pretty manner.
- **Argparse:** to read the data input by user.
- **Plots** are saved in the current directory for ease of access and analysis.