Medical image segmentation

Ho Ngoc Hoang Minh
Da Nang University of Science and Technology - Faculty of Information Technology

Abstract*: The aim of the project is to build a medical image segmentation model using deep learning. This model helps the user to identify the cell's nucleus.*

Keywords: U-Net, Deep Learning

1. Introduction

We've all seen people suffer from diseases like cancer, heart disease, chronic obstructive pulmonary disease, Alzheimer's, and diabetes. Many have seen their loved ones pass away. Think how many lives would be transformed if cures came faster.

Identifying the cells' nuclei is the starting point for most analyses because most of the human body's 30 trillion cells contain a nucleus full of DNA, the genetic code that programs each cell. Identifying nuclei allows researchers to identify each individual cell in a sample, and by measuring how cells react to various treatments, the researcher can understand the underlying biological processes at work.

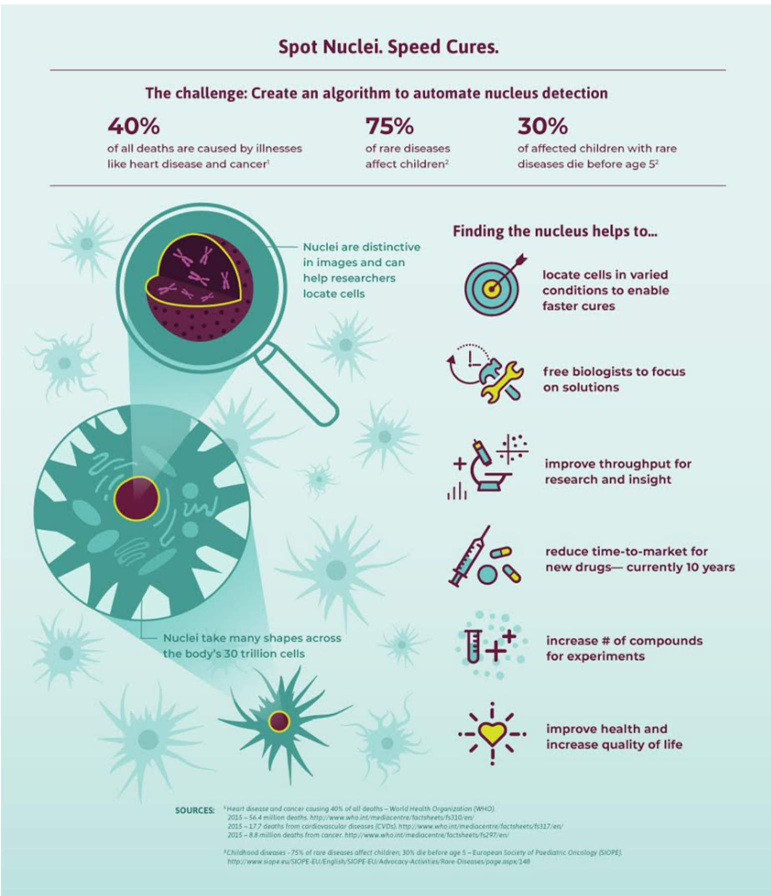

Figure 1.1. Finding the nucleus helps to

2. System Analysis and Design

2.1 Introduction

The model that I use here is the U-Net model.
To model landmark recognition using Deep Learning, we take the following stages:

- Data collection
- Data processing
- Feature extraction
- Training model
- Visualize some test image

2.2 The model of image segmentation

2.2.1 Data collection stage

To train the model I have to collect data initially. The data I'm using here is from the 2018 Data Science Bowl that was created with the goal of find the nuclei in divergent images to advance medical discovery.

This dataset contains a large number of segmented nuclei images. The images were acquired under a variety of conditions and vary in the cell type, magnification, and imaging modality (brightfield vs. fluorescence). The dataset is designed to challenge an algorithm's ability to generalize across these variations.

2.2.2 Data preprocessing stage

The training data is divided into two sets - train set : validate set = 9:1.

The data processing process consists of 4 main parts: reading data, data augmentation and data preparation for training. U-Net type architecture requires input image size to be divisible by 2^N where N is the number of max pooling layers. In the vanilla UNet, N=5. Train and validate images are resized to 128 x 128 for faster training and fit the requirements of the U-Net model's input size.

Data Augmentation: Using random horizontal flip for images to increase the richness of data.

At the end, the data normalized with the mean and standard deviation calculated before.

2.2.3 Feature extraction stage

The model that I use here is the U-Net model.

The *U-Net* was developed by Olaf Ronneberger et al. for Bio Medical Image Segmentation. The architecture contains two paths. First path is the contraction path (also called as the encoder) which is used to capture the context in the image. The encoder is just a traditional stack of convolutional and max pooling layers. The second path is the symmetric expanding path (also called as the decoder) which is used to enable precise localization using transposed convolutions. After each up-conv, we also have a concatenation of feature maps (gray arrows) that are with the same level. This helps to give the localization information from contraction path to expansion path.

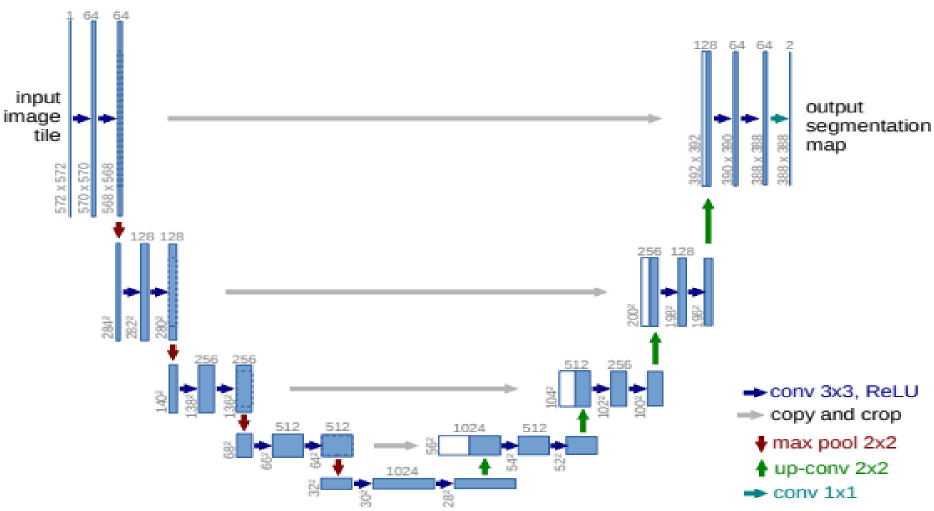

Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

Figure 2.1. U-Net model architecture.

Overlap-tile strategy: U-Net does not use any fully connected layers and uses the valid part of each convolution. This strategy allows the seamless segmentation of arbitrarily large images by an overlap-tile strategy. At the image boundary, the image is extrapolated by mirroring.

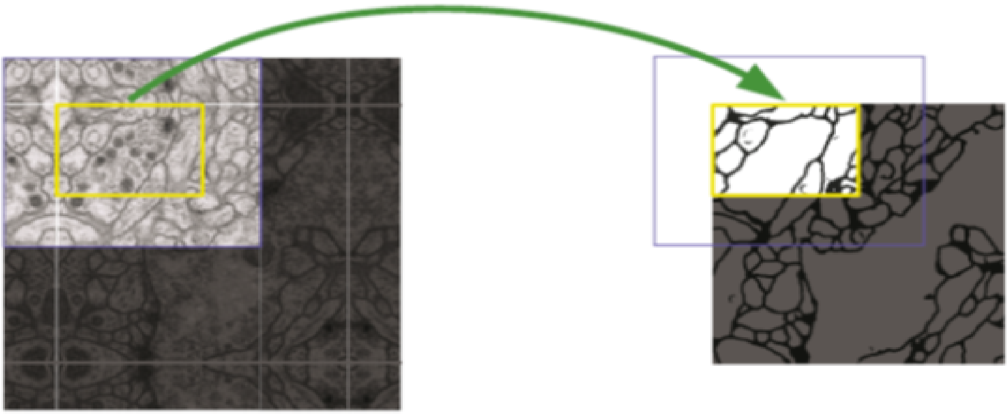

Figure 2.2. Overlap-tile strategy for seamless segmentation of arbitrary large images.

2.2.4 Training stage

1. Batch size: 10.
2. Learning rate: 3e-4.
3. Epochs: 25.
4. Save the weights only if there is improvement in validation loss.
5. Loss function: DiceBCELoss.

   This loss combines Dice loss with the standard binary cross-entropy (BCE) loss that is generally the default for segmentation models. Combining the two methods allows for some diversity in the loss, while benefiting from the stability of BCE.

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|}$$

Figure 2.3. Dice Loss formula

$$J(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N} H(p_n, q_n) = -\frac{1}{N}\sum_{n=1}^{N}\left[y_n \log \hat{y}_n + (1 - y_n)\log(1 - \hat{y}_n)\right],$$

Figure 2.4. Binary Cross-entropy Loss formula

6. Optimizer: Adam (Adaptive Moment Estimation).

   Here are some pictures from the training of the model:

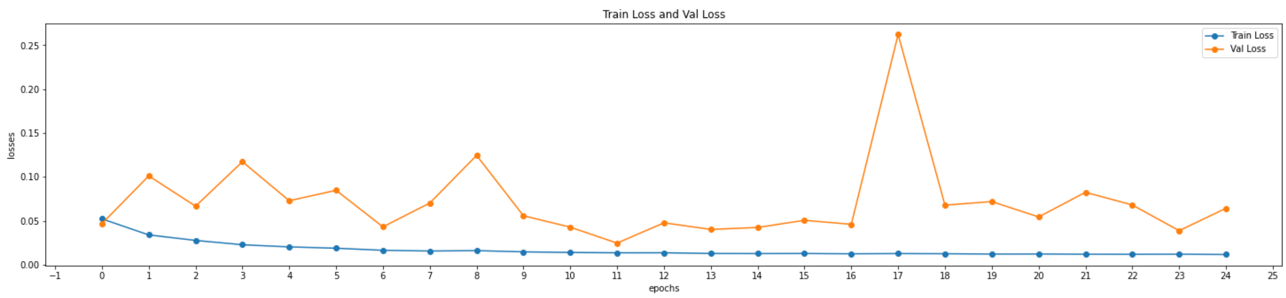

Figure 2.5. Graph of loss during training.

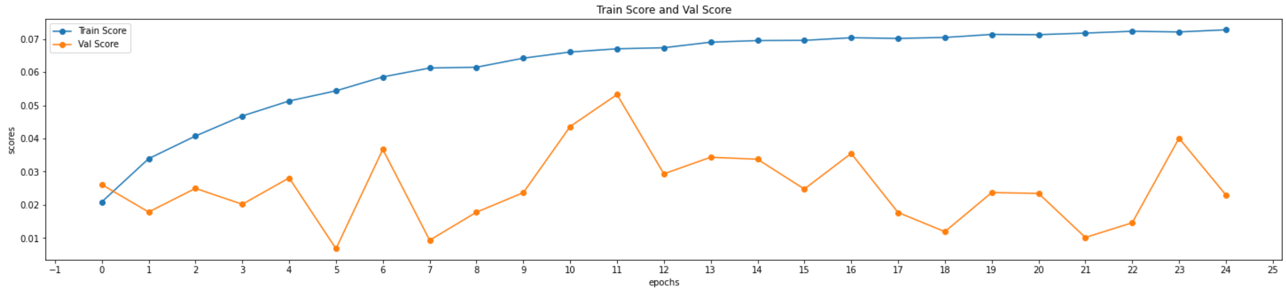

Figure 2.6. Graph of IOU score during training.

3. Experiment

3.1 Implementation environment

- The main processing language: Python.
- The used editor: Kaggle.

3.2 Demo result

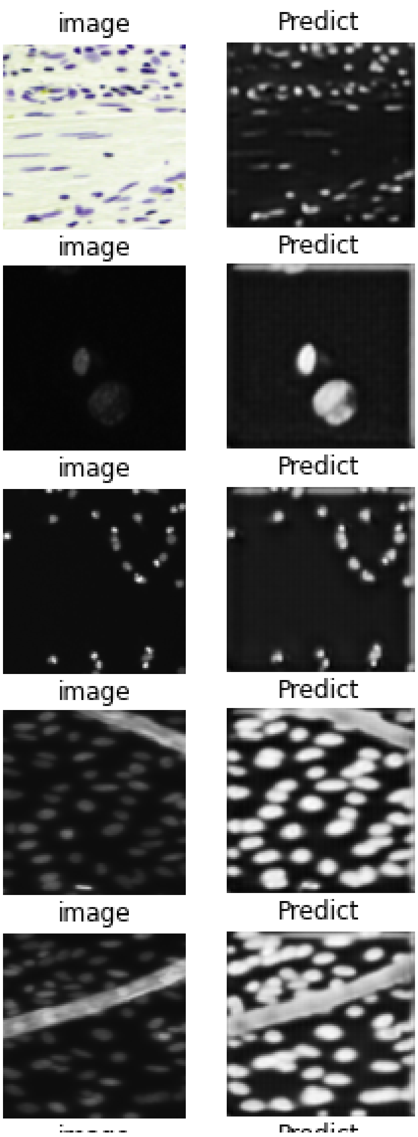

Figure 3.2. View predict result on test set.

4. Conclusion

4.1 Development directions

The model can improve the quality and change it to better fit the actual problem in the following directions:

- Collect more data, retrain the model to increase accuracy.
- Use more augmentation on the training set.
- Try another pretrained encoder like ResNet34 for reducing training time.

※Reference:

[1] [1505.04597] U-Net: Convolutional Networks for Biomedical Image Segmentation
[2] Loss Function Library - Keras & PyTorch | Kaggle
[3] Using Albumentations for a semantic segmentation task
[4] Up-sampling with Transposed Convolution | by Naoki
[5] Review: U-Net (Biomedical Image Segmentation) | by Sik-Ho Tsang | Towards Data Science.