U-Net: Convolutional Networks for Biomedical Image Segmentation

Indraneel Sunil Mane
Khoury College of Computer Sciences
Northeastern University
Boston, MA 02215
mane.i@northeastern.edu

Nitin Kumar Mittal
Khoury College of Computer Sciences
Northeastern University
Boston, MA 02215
mittal.nit@northeastern.edu

Abstract

In this paper, we implemented the neural network architecture U-Net proposed in the paper U-Net: Convolutional Networks for Biomedical Image Segmentation to perform semantic image segmentation. The architecture consists of a contracting path and a symmetric expansive path along with bridge connections. The authors promote use of data augmentation to facilitate U-Net to train from few annotated training samples originally. The original implementation of U-Net was done in Caffe. We implemented our version of U-Net in Python using Pytorch and trained it to segment membranes of neuronal structures.

1 Introduction[1]

Semantic image segmentation [1] is the process of labeling each pixel of an image with its corresponding class. Image segmentation in the medical field [2] helps clinicians to focus on a particular area of the disease and extract detailed information for a more accurate diagnosis. Image segmentation if done manually is a tedious job. Therefore, we require tools that can automatically label each pixel with its corresponding class. Also, another key challenge in the field of medical image segmentation is the unavailability of a large number of annotated samples (pixels with their correct classes). Therefore, such tools should be able to learn to annotate pixels from fewer original[2] samples.

Convolutional Neural Networks (CNNs) have shown state-of-the-art performance for automated medical image segmentation [3]. In this paper, we made use of U-Net architecture to perform end-to-end pixel-wise prediction.

2 Implementation

2.1 Network Architecture

Our version of U-Net architecture is illustrated in Figure 1. The illustrated architecture is inspired by original U-Net architecture.

A general U-Net architecture consists of a contracting path/ analysis path (left side) and an expansive path/ synthesis path (right side). In the analysis path, deep features are learned, and in the synthesis path, segmentation is performed based on the learned features. Additionally, U-Net uses connections between contracting and expansive paths to propagate dense feature maps from the analysis path to the corresponding

[1] scientific context
[2] available data prior to data augmentation

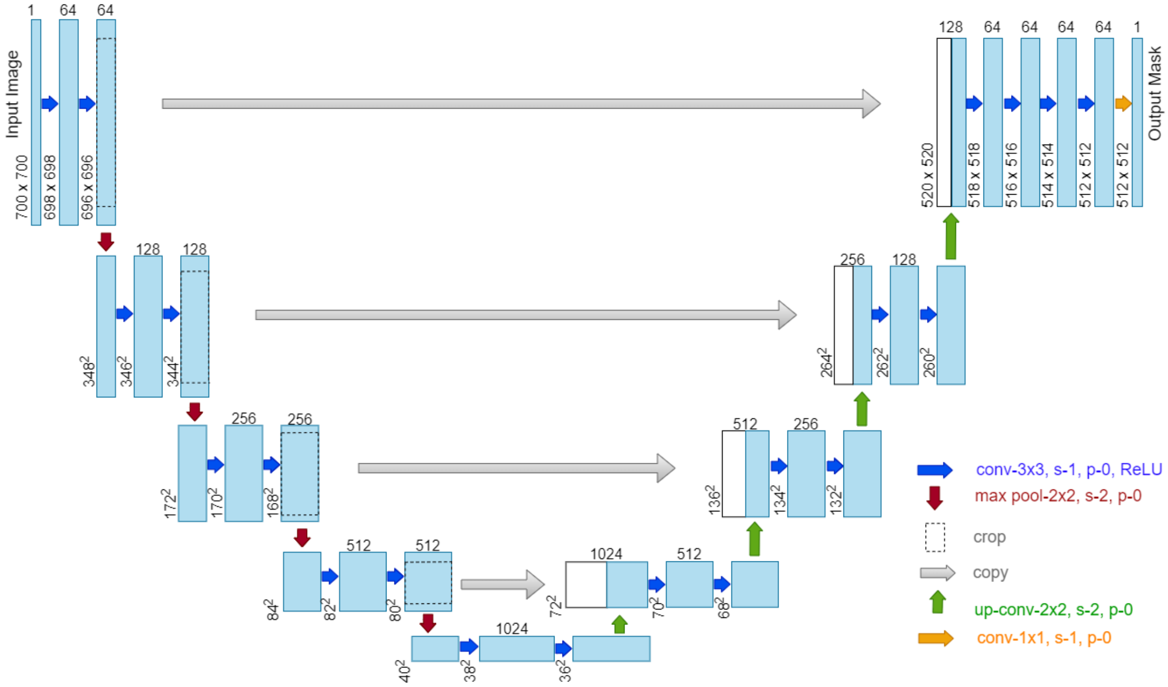

Figure 1: U-Net Architecture

After each down-sampling (max-pooling) step, the number of feature channels are doubled. After each up-sampling (up-convolution) step, the number of feature channels are halved. The cropping is necessary due to the loss of border pixels in every convolution. Across all convolution operations, zero padding is used.

layers in the synthesis part. In this way, the spatial information is applied to the deeper layer, which significantly produces a more accurate output segmentation map. Thus, adding more layers to the U-Net will allow the network to learn more representative features leading to better output segmentation [4].

In our version of U-Net architecture, we added two 3*3 convolutions before the last layer as compared the original version to the output segmentation map of shape 512x512. The final layer consist of 1x1 convolution operations to output a segmentation map with channels equal to the desired number of classes (can output a single channel segmentation map for binary classification of pixels).

2.2 Data

The data for biomedical image segmentation is obtained from ISBI Challenge: Segmentation of neuronal structures in electron microscopic stacks. Originally, the data is available in 2 parts - train and test. The train data consist of 30 images of neuronal structures[3] with corresponding masks containing binary labels - white[4] for the pixels of segmented objects and black[5] for the rest of pixels (which correspond mostly to membranes). The test data consist of images without corresponding masks. The shape of each image and mask is 512x512 respectively.

We used 20 image-mask pairs for training and 5 image-mask pairs each for validation and testing respectively. All image-mask pairs were selected from original train data. In the original paper, the authors made use of all 30 image-mask pairs for training and validation. For testing, they submitted probability maps of masks of corresponding test images.

[3] from a serial section Transmission Electron Microscopy (ssTEM)
[4] pixel-value 255
[5] pixel-value 0

2.2.1 Data Augmentation

Since we originally had 20 samples to train our model, we made use of data augmentation techniques (similar to as in the original paper) to teach the model the desired invariance and robustness properties.

For each image-mask pair, we obtained 8 rotated versions by rotating it at quarter angles and then flipping it. Another major data augmentation technique used was elastic deformation. To perform elastic deformations on image-mask pair, we performed Affine Transformation by randomly displacing 3 points within the image. Elastic deformations were then made by performing cubic interpolation on pixels sampled from Gaussian distribution with a mean of 0 and a standard deviation of 10 pixels.

Overall we generated 3360 augmented image-mask pairs from 20 image-mask pairs originally to train our model. Data augmentation was not applied for image-mask pairs in the validation and test set created by us.

2.3 Training

We performed normalization on pixel values of images to convert them from [0, 255] to [0., 1.]. Pixel values of masks where also normalized from {0,255} to {0,1} respectively. We used weighted binary cross-entropy as a loss function for our model. The weights for pixel values 0 and 1 are obtained by taking inverse frequencies within masks from the train set (augmented).

We took a batch-size of 3 and trained our model for 24 epochs. We set an initial learning rate of $3\mathrm{e}{-4}$ and a learning rate scheduler to decrease the learning rate by a factor of .99 after every 500 model parameter updates. We took Adam optimizer with β_1= 0.99 and β_2= 0.999. We trained our model for ~6 hours on P100 GPU with 12GB of memory. Loss across epochs have been plotted in Figure 2. Our model starts over-fitting after epoch 12. The best model was selected on the basis of minimum validation loss.

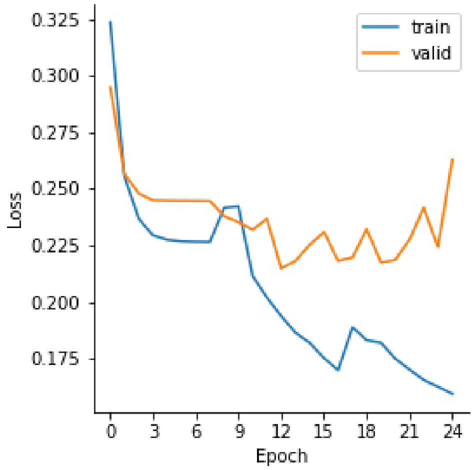

Figure 2: Train and Validation loss

3 Results

Segmented map from the original paper and our implementation can be seen in Figure 3. and Figure 4. respectively. Pixel Error and Rand Error from original and our implementation of U-Net for segmentation of neuronal structures can be seen in Table 1[6]. Pixel Error and Rand Error from original and our implementation are not obtained for same segmented mask/s, therefore the exact comparison is not possible. As mentioned above, the authors obtained pixel and rand errors by submitting probability maps of masks of corresponding test images to the ISBI Challenge. Overall, they did 78 submissions. To generate a segmented mask, we

[6] values rounded to 4 decimal places

used a threshold of .5 to classify each pixel as 0 or 1. We computed mean pixel and rand errors respectively for 5 segmented masks from the held-out-set/ test-set. From Figure 4. we can see that our model did work to segment cell membranes of neuronal structures but on comparing pixel and rand errors from the original implementation, the metrics are better for the latter. We suspect this is because the authors used morphological operations to compute separation borders between touching cells.

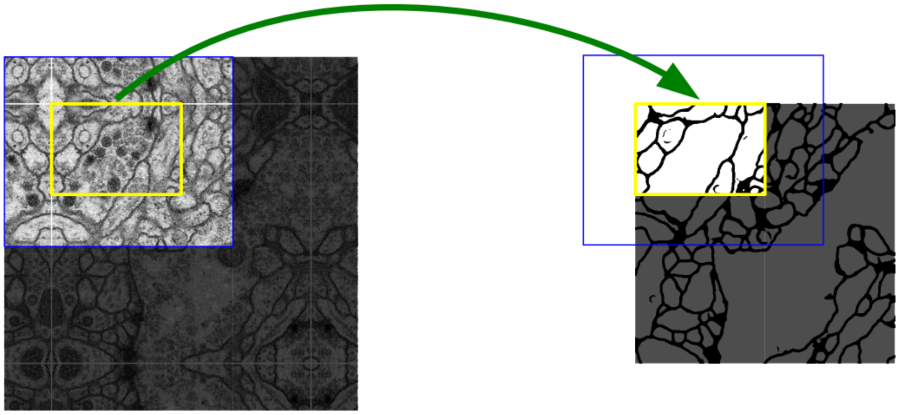

Figure 3: Image and Predicted Mask from Original Implementation

Overlap-tile strategy for seamless segmentation of arbitrary large images (here segmentation of neuronal structures in EM stacks). Prediction of the segmentation in the yellow area, requires image data within the blue area as input. Missing input data is extrapolated by mirroring.

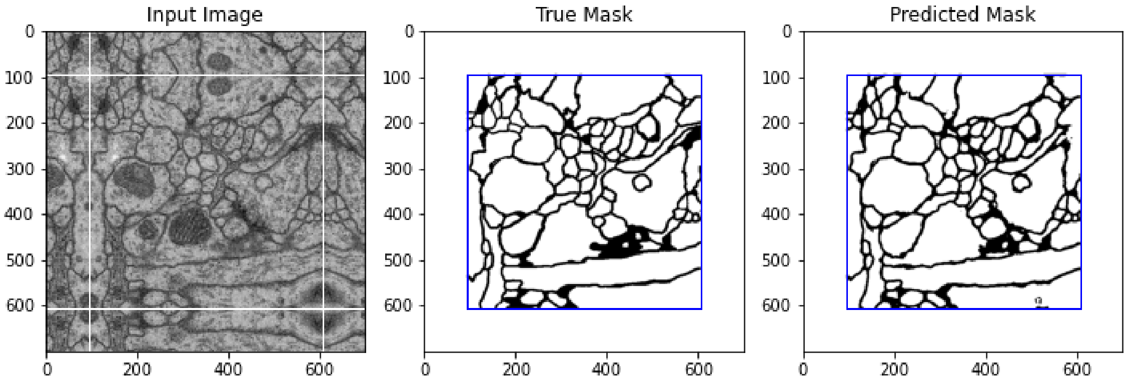

Figure 4: Test Input Image, True and Predicted Masks from our Implementation

Prediction of the segmentation map of size 512x512 in blue box requires input image of size 700x700. To convert an input image of size 512x512 to 700x700, extrapolation by mirroring operation is used around edges.

Table 1: Results

Group Name	Pixel Error	Rand Error
original u-net	0.0611	0.0382
our u-net	0.0828	0.1517

References

[1] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully Convolutional Networks for Semantic Segmentation" CVPR (2015)

[2] F. Zhao and X. Xie. "An overview of interactive medical image segmentation”. *Annals of the BMVA* Vol. 2013, No. 7, pp 1–22 (2013)

[3] Samir S. Yadav and Shivajirao M. Jadhav. "Deep convolutional neural network based medical image classification for disease diagnosis".

[4] Debesh Jha, Michael A. Riegler, Dag Johansen, Pal Halvorsen, Havard D. Johansen "DoubleU-Net: A Deep Convolutional Neural Network for Medical Image Segmentation".