B.Tech 7th Sem Project Report

Name: Lenka Ayyappa

Roll No: 1801098

Electronics and Communication Engineering Department

Indian Institute of Information Technology, Guwahati

Bachelor of Technology

28th November 2021

Certificate of Approval

This is to certify that the 7th Sem B.Tech. project report entitled “Analysis of Laser Speckle Images” is a record of Bonafide work carried out by Lenka Ayyappa under my supervision and guidance.

The report has partially fulfilled the requirements towards the degree of Bachelor of Technology in Electronics and Communication Engineering at Indian Institute of Information Technology, Guwahati.

Signed:
Dr. Rusha Patra
Assistant Professor
Department of Electronics and Communication
Indian Institute of Information Technology, Guwahati
Guwahati 781001, Assam, India.

Contents

1. Introduction

In the last two years, deep convolutional networks have outperformed the state of the art in many visual recognition tasks. While convolutional networks have already existed for a long time, their success was limited due to the size of the available training sets and the size of the considered networks. For resolving such issues we present a network and training strategy that relies on the strong use of data augmentation to use the available annotated samples more efficiently. Using the same network trained on transmitted light microscopy images (phase contrast and DIC) and Moreover, the network is fast. Segmentation of a 512x512 image takes less than a second on a recent GPU.

1.1. Literature Review:

The breakthrough by Krizhevsky et al was due to supervised training of a large network with 8 layers and millions of parameters on the ImageNet dataset with 1 million training images. Since then, even larger and deeper networks have been trained. The typical use of convolutional networks is on classification tasks, where the output to an image is a single class label. However, in many visual tasks, especially in biomedical image processing, the desired output should include localization, i.e., a class label is supposed to be assigned to each pixel. Moreover, thousands of training images are usually beyond reach in biomedical tasks.

Hence, Ciresan et al trained a network in a sliding-window setup to predict the class label of each pixel by providing a local region (patch) around that pixel as input. First, this network can localize. Secondly, the training data in terms of patches is much larger than the number of training images. The resulting network won the EM segmentation challenge at ISBI 2012 by a large margin.

Obviously, the strategy in Ciresan et al has two drawbacks. First, it is quite slow because the network must be run separately for each patch, and there is a lot of redundancy due to overlapping patches. Secondly, there is a trade-off between localization accuracy and the use of context. Larger patches require more max pooling layers that reduce the localization accuracy, while small patches allow the network to see only little context. More recent approaches proposed a classifier output that takes into account the features from multiple layers. Good localization and the use of context are possible at the same time.

2. Software required:

Implemented using Python 3.7 in Google Colaboratory by cloning git-hub repository in Windows 10.

Libraries required:

- Keras: Keras is an open-source deep learning framework for python. This is based on minimal structure that provides a clean and easy way to create deep learning models based on TensorFlow.
- NumPy (Numerical Python): NumPy is a python library consisting of multidimensional array objects and a collection of routines for processing the arrays. Using NumPy, mathematical and logical operations on arrays can be performed.
- OpenCV: OpenCV is a python library of functions. This mainly focuses on image processing, video capture and analysis including features like face detection and object detection. This is used to develop real-time computer vision applications.
- Matplotlib: This is a python library for plotting. It is used for data visualization and especially for making 2D plots from data in arrays.

Architecture:

The network architecture consists of a contracting path (left side) and an expansive path (right side). The contracting path follows the typical architecture of a convolutional network. It consists of the repeated application of two 3x3 convolutions (unpadded convolutions), each followed by a rectified linear unit (ReLU) and a 2x2 max pooling operation with stride 2 for down sampling.

At each down sampling step we double the number of feature channels. Every step in the expansive path consists of an up sampling of the feature map followed by a 2x2 convolution (“up-convolution”) that halves the number of feature channels, a concatenation with the correspondingly cropped feature map from the contracting path, and two 3x3 convolutions, each followed by a ReLU.

The cropping is necessary due to the loss of border pixels in every convolution. At the final layer a 1x1 convolution is used to map each 64-component feature vector to the desired number of classes. In total the network has 23 convolutional layers. To allow a seamless tiling of the output segmentation map is important to select the input tile size such that all 2x2 max-pooling operations are applied to a layer with an even x and y-size.

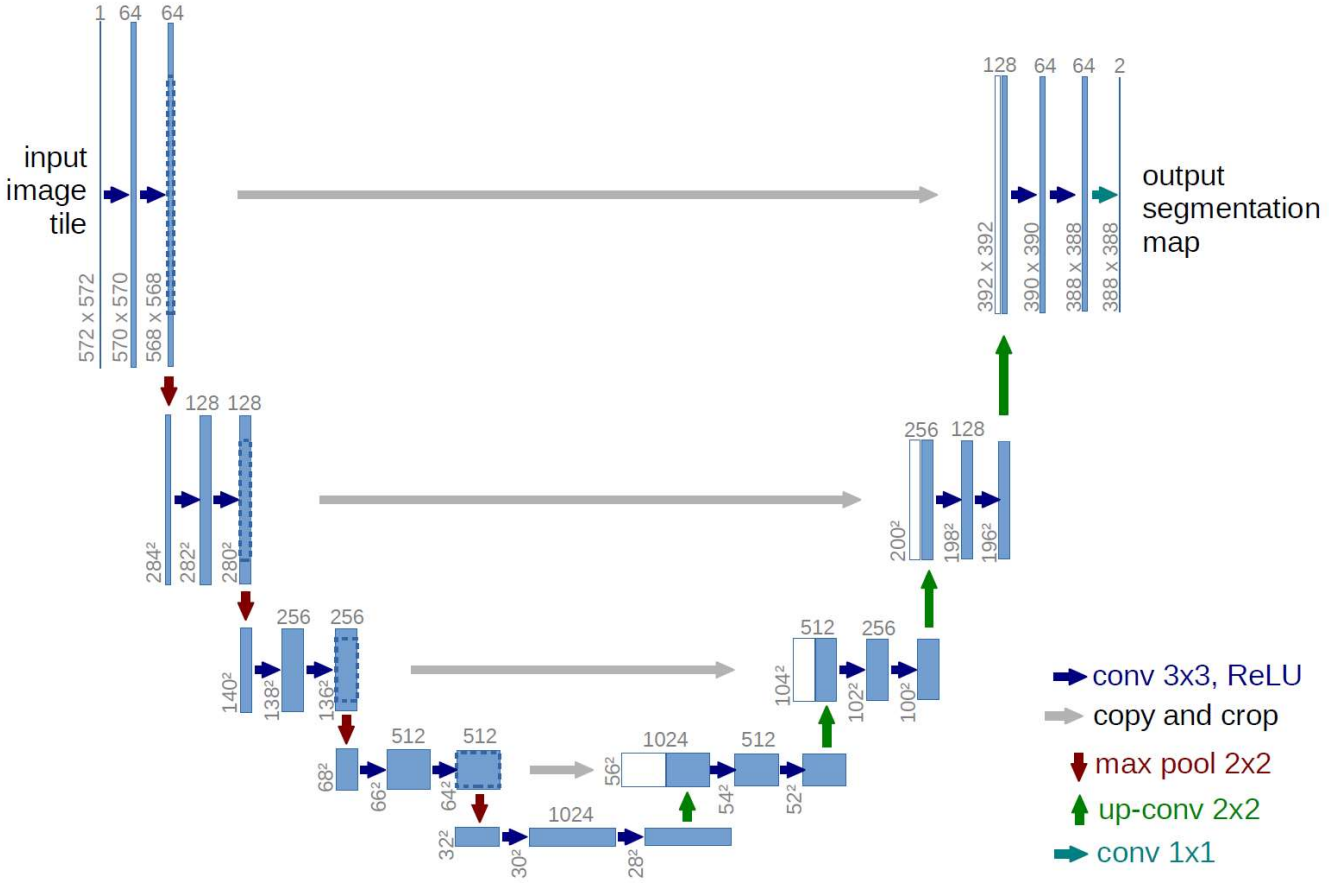


Diagram-a :- Unet Architecture

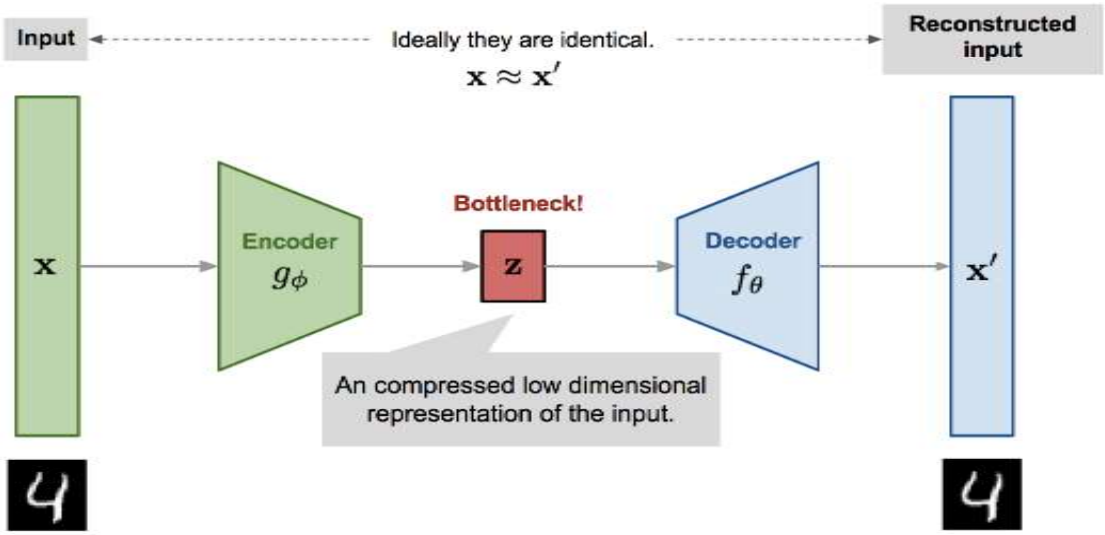


Diagram
Encoder Decoder

Implementation:

Firstly, we converted the dataset raw images into tiff format. Later we divided the dataset into two parts mainly training (80%) and testing (20%). We won't be needing the image class labels or annotations of the dataset. We have done cloning Git-hub repository. And included the training folder in image folder. And within the data folder we created the sample folder. And in U-Net folder we created main.py file and model.py file and we modified the following classes by giving the correct paths. After running the main.py we get the segmented results in stage1.

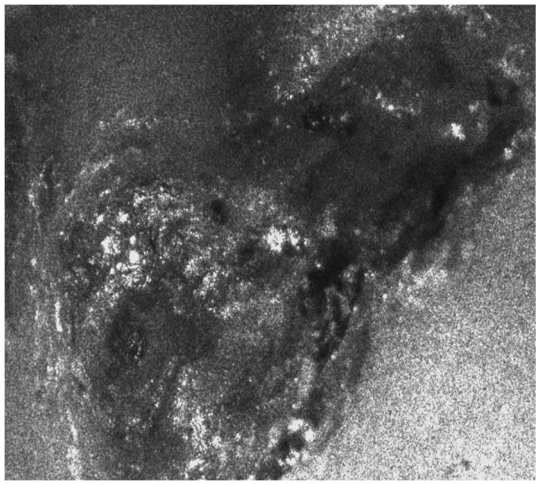


Figure1.Test Image Input

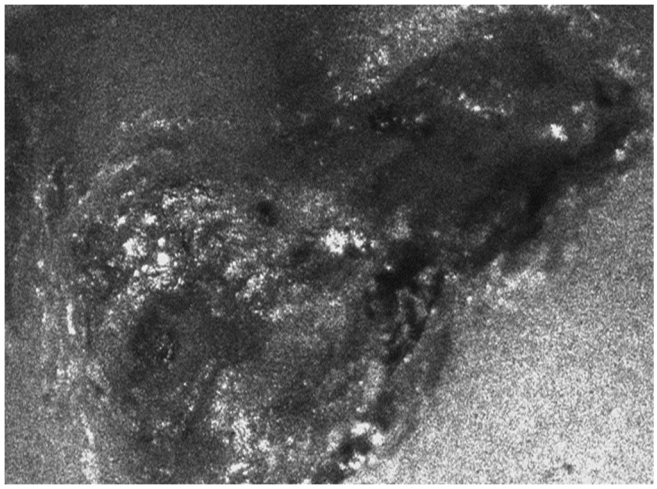
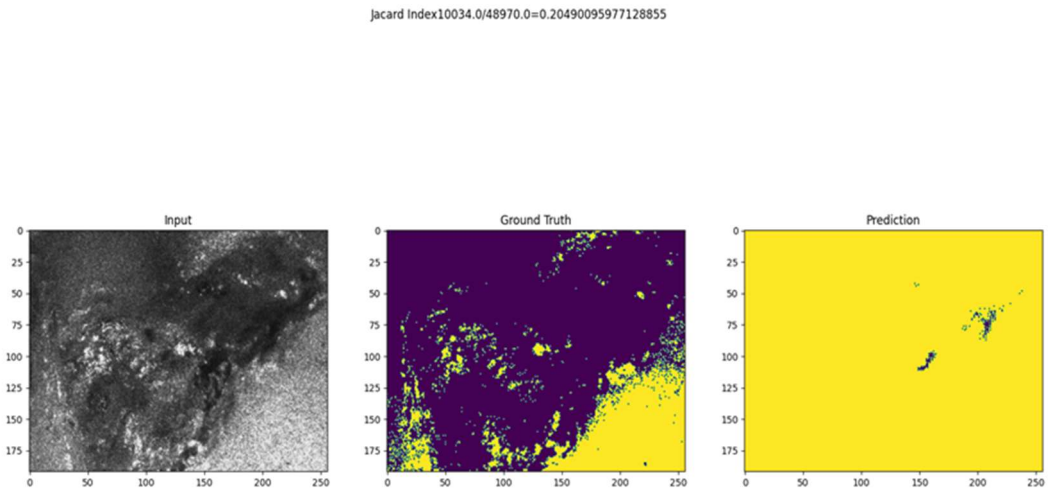
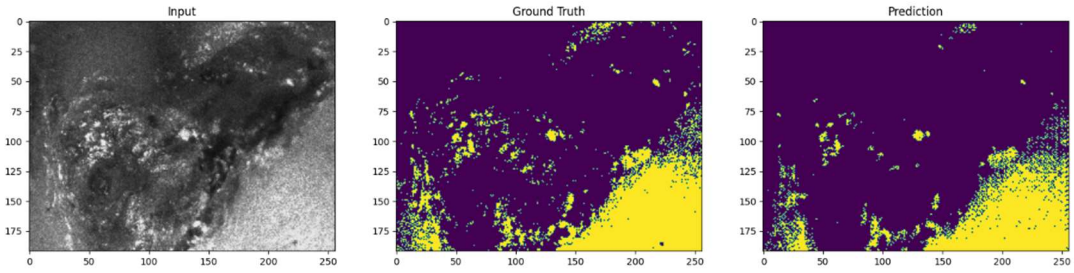


Figure2. Train Image Input



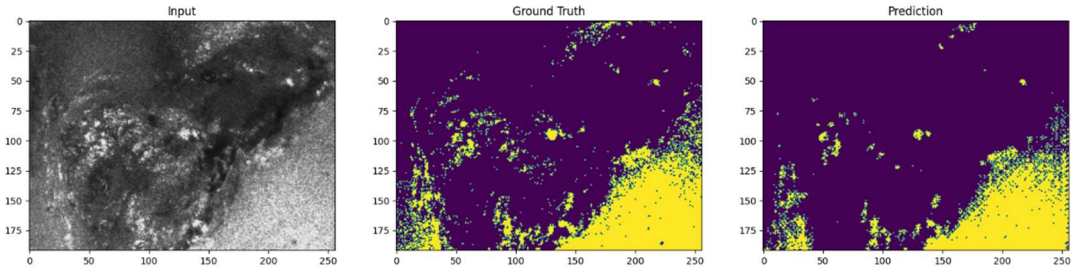
Output 1 for test and train dataset

Jacard Index6996.0/10827.0=0.6461623718481574



Output 2 for test and train dataset

Jacard Index6930.0/10936.0=0.6336869056327725



Output 3 for test and train dataset

In stage2 we take input as wound segmented images of previous step and output will be the image which is given as ground truth. Here ground truth is used as target output. Now in that image 4 classes are present and we remove the healthy tissue class and background part then we are left with progressive and non-progressive part.

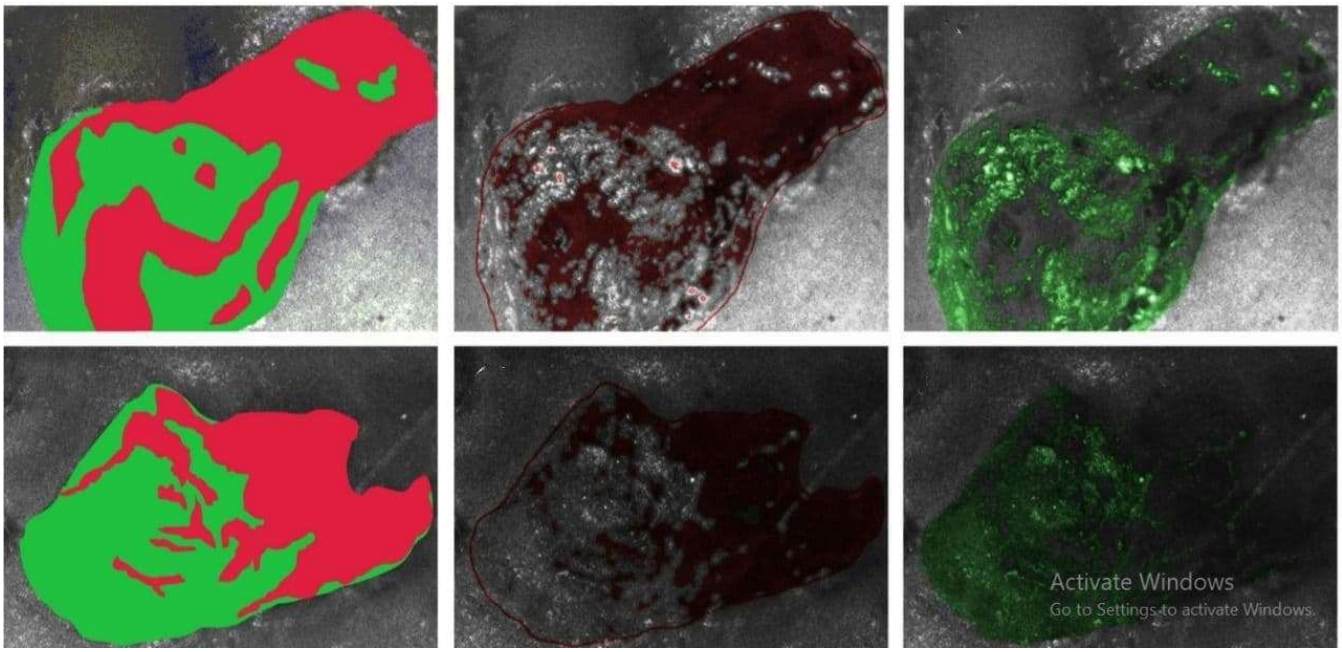


Fig 3. Annotated tissue labels, progressive and non-progressive wound regions for the dataset image respectively. The progressive region is marked in red and relatively non progressive region is marked in green

Optimization:

For optimizing the parameters, we used Adaptive Moment estimation ADAM, we minimized the cost function using Adam, which is used to optimize the network at convergence rate.

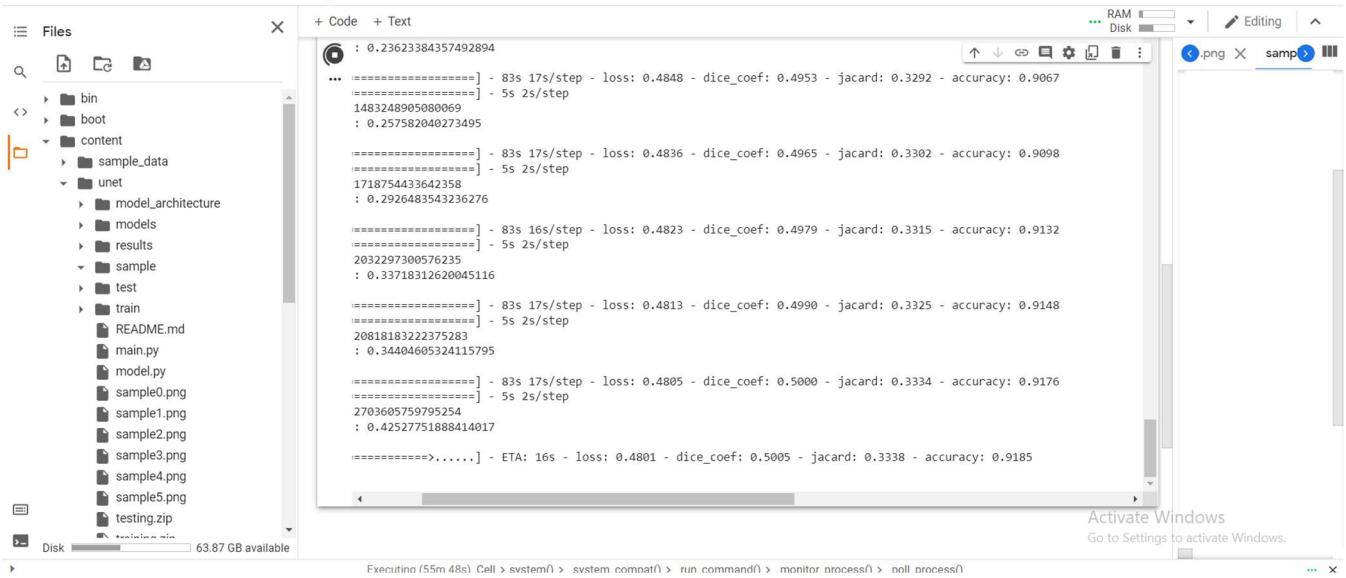


Fig 4. Accuracy, loss, dice-coefficient, Jacard values respectively

Conclusion:

U-Net architecture is used for image localization which helps in predicting the image pixel by pixel. Good performance on very different biomedical segmentation applications was achieved by U-Net. This network is strong enough to make good predictions based on even fewer data sets by using excessive data augmentation techniques.

We got an overall accuracy of 94.71% on 84 training images and 32 validation images of size 256X256. We have used binary cross-entropy as the loss function. It took me approximately 3 hours to train on an 8 GB Windows with a 1.8 GHz Intel Core i5 processor (CPU). We separated the wound labels into progressive and non-progressive parts using the ground truth.

References:

[1] Ciresan, D.C., Gambardella, L.M., Giusti, A., Schmidhuber, J.: Deep neural networks segment neuronal membranes in electron microscopy images. In: NIPS, pp. 2852–2860

[2] Dosovitskiy, A., Springenberg, J.T., Riedmiller, M., Brox, T.: Discriminative unsupervised feature learning with convolutional neural networks. In: NIPS (2014)

[3] U-NET for Biomedical Image Segmentation by Muthuraman S.

[4] Learn How to Train U-Net On Your Dataset by Sukriti Paul.

[5] U-Net Convolution Networks for Biomedical Image Segmentation by Olaf Renneberger et.al