



National University
of computer and emerging sciences

2D UNET Segmentation of Mitochondria from Electron Microscopy

Presented To:

Sir. Kashif Zafar

Presented By:

Hassan Rehman L217302

Faiz Lewis L217272

Annas Butt L202379

Introduction:

Semantic segmentation of mitochondria from electron microscopy images, it is a technique to automatically detect and segments the mitochondria from the images using deep learning technique. Semantic segmentation is also called pixel level classification. It is a task to classify parts of the image together which belongs to the same object. It has many application and now a days it is very common technique for the classification of bio medical images. Biomedical images are different from traditional images as mos of them when combined together represent a voxel. A voxel are pixels in 3 dimension.

Image segmentation can be treated as the pixel level prediction because unlikely the normal classification where object is classified on the basis of kernel slides over the image and returns the bounding box, it classify each pixel into category.

Segmentation is also of two types:

- Semantic Segmentation
- Instance Segmentation

Currently we are going to use semantic segmentation. Instance segmentation not only segment the object but also gives unique identity to each segmentation which makes it more efficient to differentiate between two segmentation.

Semantic segmentation has multiple application:

- Land classification
- brain and tumor segmentation
- lumbar segmentation

And there many more segmentation problems that can be solved by this technique.

Problem Statement:

The current problem that we are going to solve is the segmentation of mitochondria from electron microscopy images. Mitochondria are present inside the cells of living organism and it provides the energy. They are of different shapes and length. They are mostly rounded or elliptical in shape. In this project we are focusing on segmenting the mitochondria.

We have to segment in such a way that we can represent that segmentation into 3D object. And for solving this problem the architecture that we are going to use in 2D UNET architecture. Our line of research is motivated by the primary need to correctly segment the mitochondria

Dataset:

The dataset is available <https://www.epfl.ch/labs/cvlab/data/data-em/> and author of datasets are following:

- Aurelien Lucchi
- Yunpeng Li

- Carlos Becker
- Pascal Fua

The dataset is taken from electron microscopy and represents 5x5x5 micrometer section taken from CA1 hippocampus region of the brain corresponding to volume. The data initially was provided in multipage TIF data which is then converted into png.

Dataset consist of training and testing data. Each data has original image and an annotated image.

Architecture:

Architecture that is selected for this problem solving is 2D UNET architecture. It is currently the best and state of the art architecture for solving segmentation problem specially bio medical images. The architecture look as follows:

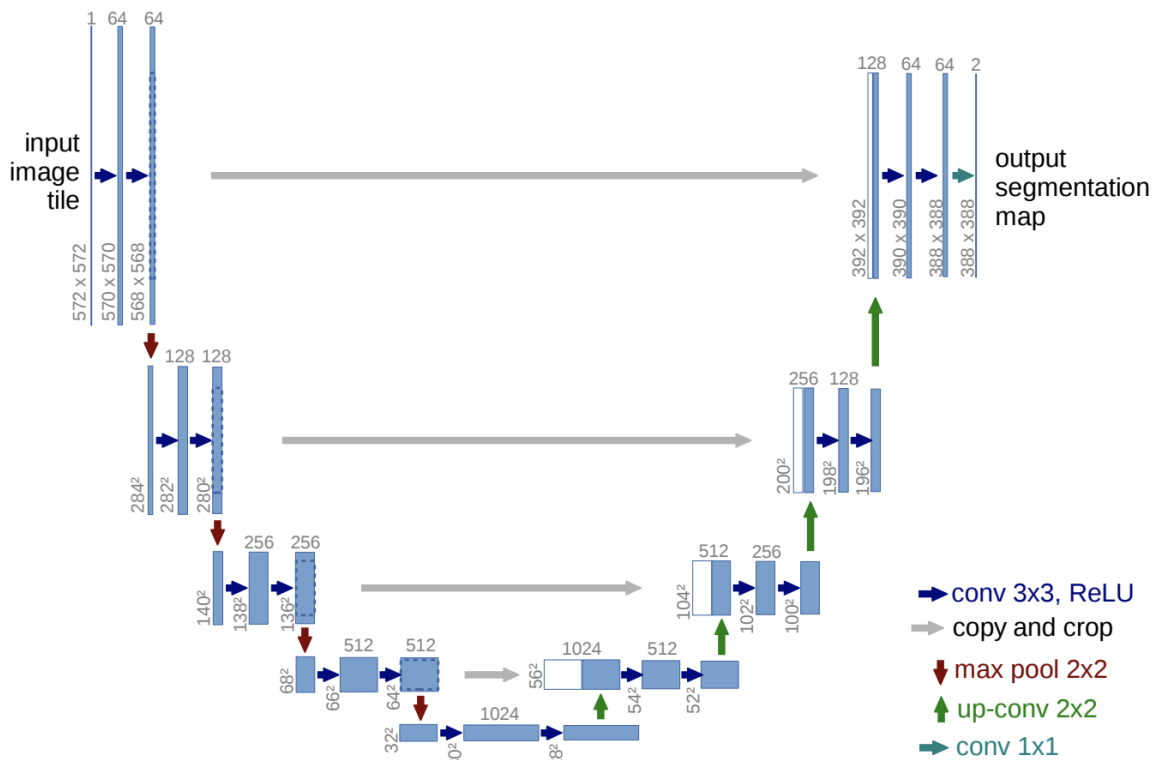


Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows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

U-net is an architecture for segmentation. It consists of contracting and expansive path. The contracting path contains the traditional convolutional network. The contracting path is also called downsampling where we double the features. The expansive path is up convolution where we merge all those feature to make the image again.

Data Preparation:

The data set is the benchmark data so we don't have to do much preprocessing. But the data count was very low. Data consists of 165 images of training data set and 165 images of testing data set. So now the target was to generate the new data set. The problem was to depict the same augmentation on annotated data and actual images and for that we use python library named as albumentation.

Following are the steps we took to prepare the data:

- Convert the series TIF file to PNG and store separately in training and testing data
- After that we apply augmentation method to increase the dataset. Following are the methods we apply:
 - Random Crop
 - Horizontal Flip
 - Center Crop
 - Random brightness contrast

The conversion was done on masked images and actual images simultaneously.

- Resizing of the training data to 384, the size that our network accepts.
- Converting the 255 values of masked image to 1 to convert our labels. 0 describe the background and 1 describes the mitochondria.
- Finally we compress the data into .npz combining both the mask and actual image, this is done to make ease for making keras data generator to load the data.

The DATA folder hierarchy can be seen in the **hierarchy.txt** file provided in the code. After augmentation a sample image look like as follows:

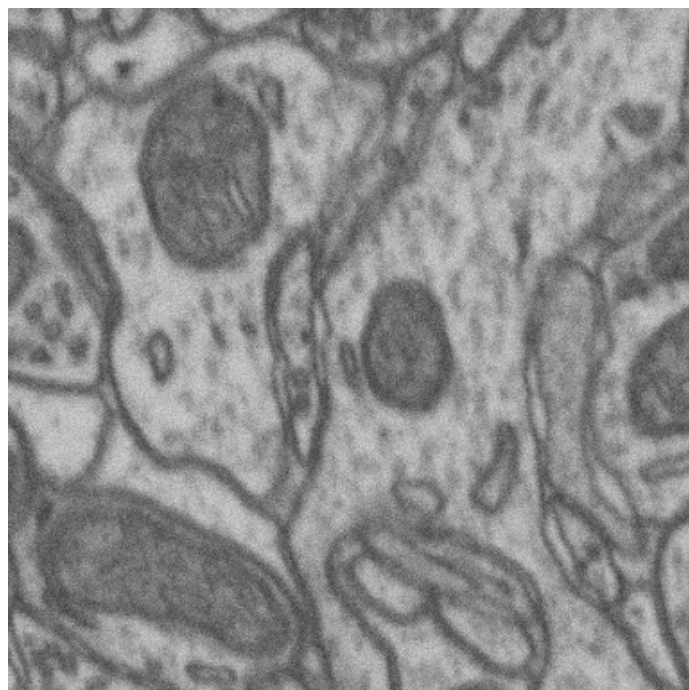
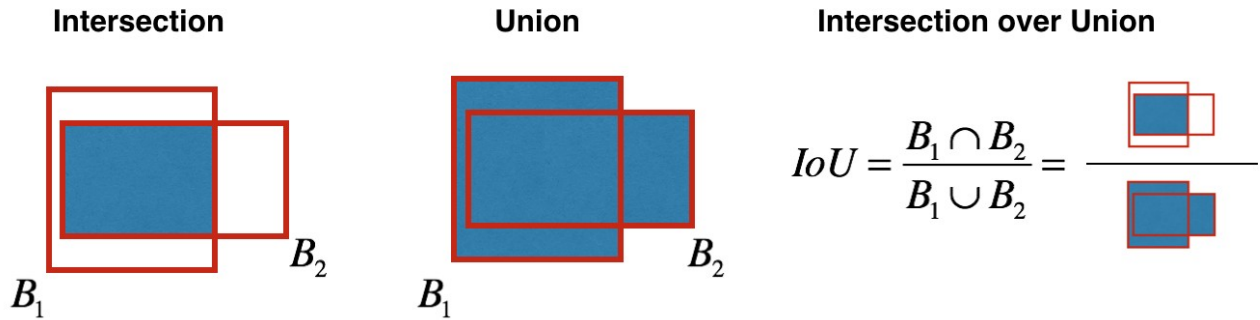


Figure 1: Data after random crop

Evaluation Metrics:

Unlike other evaluation metrics in classification problem we cannot use those traditional metrics in segmentation. As we know that segmentation is pixel level classification so here we want to see how much area is perfectly predicted by the network. For that purpose the evaluation metric that we use here is IOU score.

IOU score stands for intersection over union score. It can be understood by the following image more appropriately:



The IOU score we achieve is about 95%.

Loss:

Loss used in this problem is dice loss. It is originated from Sorensen Dice coefficient. It is used to gauge the similarity between two samples. It is introduced in computer vision community in 2016 and is suitable for 3D medical images. The formula is given as:

$$D = \frac{2 \sum_i^N p_i g_i}{\sum_i^N p_i^2 + \sum_i^N g_i^2}$$

in boundary detection scenario the value of p and g are either 0 or 1. the denominator is the sum of total boundary pixels correctly predicted.

Training hardware, libraries and Configuration:

Following is the hardware used in training:

- 32 GB RAM
- NVIDIA GTX 1080 Ti 8GB VRAM

and following python libraries have been used in the procedure:

- tensorflow
- ternsorboard

- opencv
- numpy
- skimage
- segmentation_models

Configuration parameter for training the data is as follows:

Learning rate: Learning rate was kept 0.001. It was optimum for this solution. It was set after trials.

Batch Size: The value of batch size is 4. System can handle upto 12 or 16 but keep it low so that we dont face any memory error while training.

Channel: As we used gray scale images so the channel is 1

Input image size: it was set to be 384 because our network accept size of 384x384 image.

Activation: As we have binary classes so sigmoid was chosen.

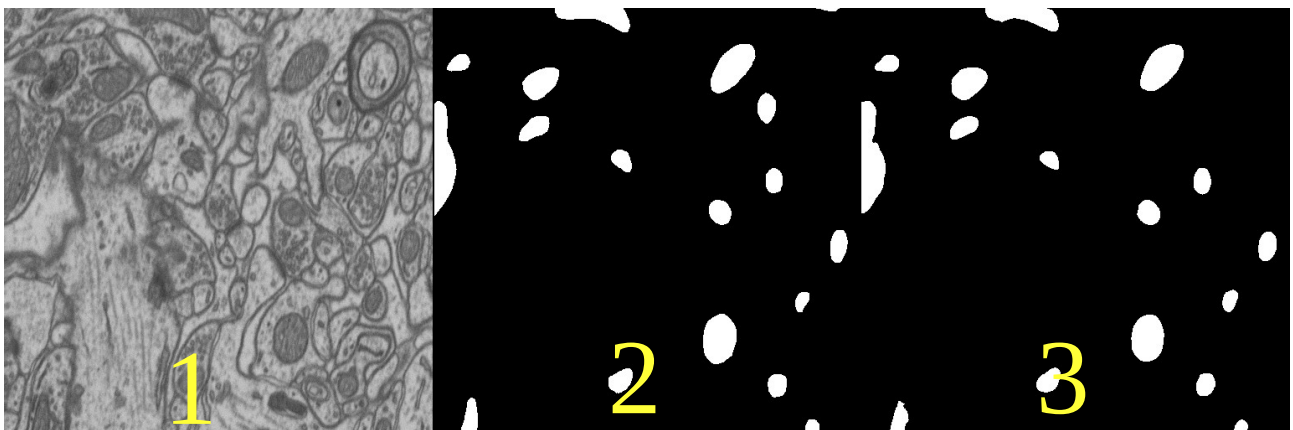
Results:

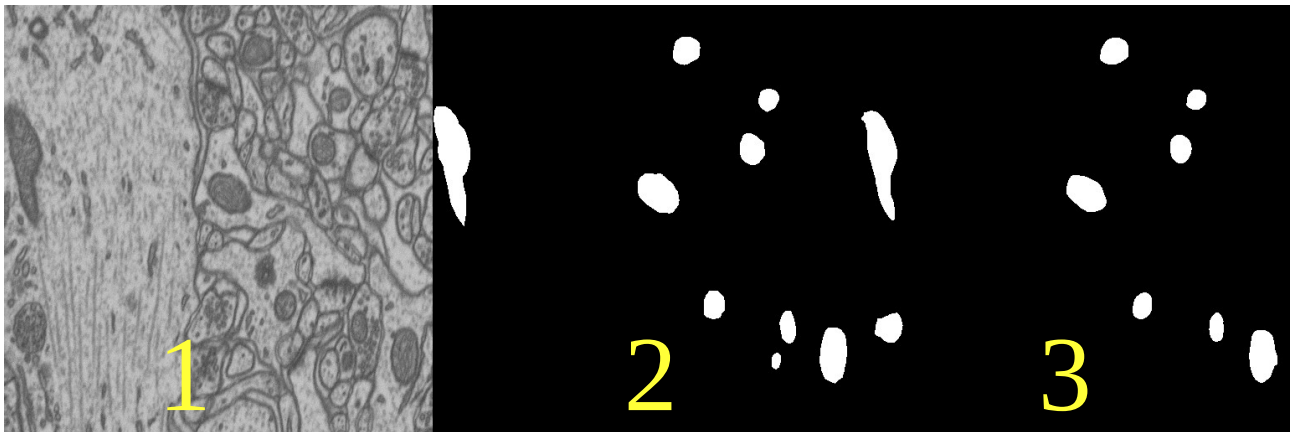
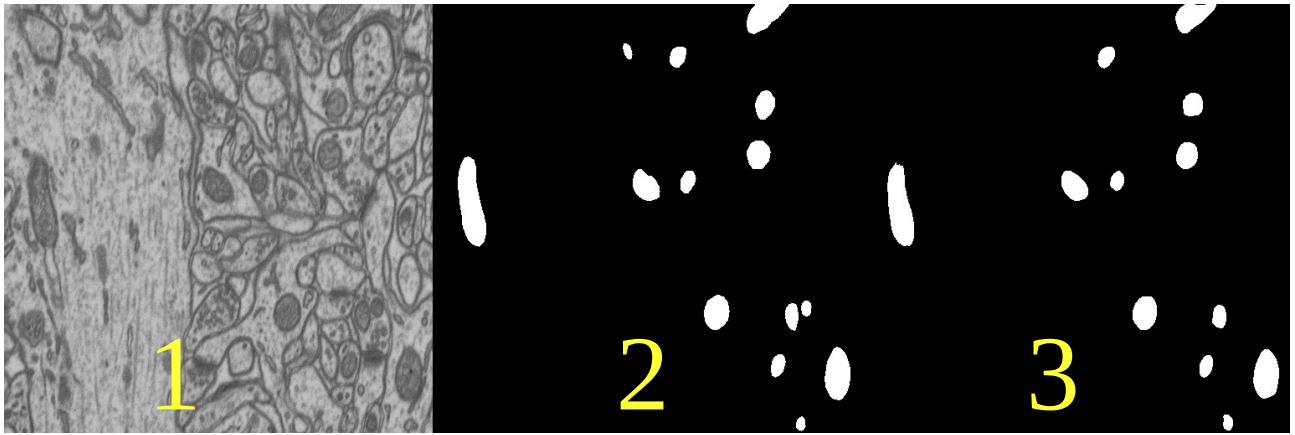
Following are the results taken from the tensor board after completion of training the data:

Dice Loss	0.022
IOU Score	96%
f1-Score	98%
IOU Score on Testing data	87%

Testing Results:

Following are the results on testing Data set:

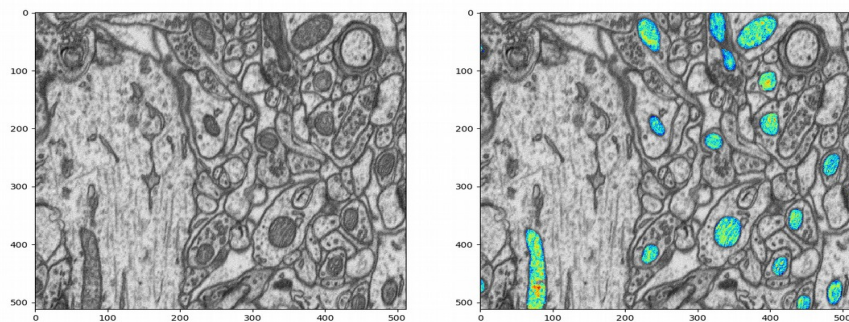


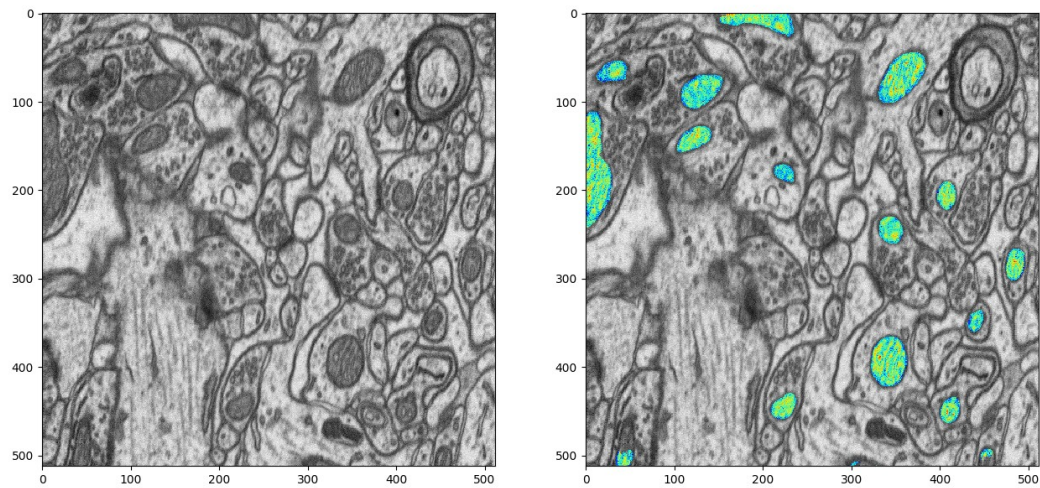


- 1 is representing original image
- 2 is representing original mask
- 3 is representing predicted image.

Overlaid Inference:

Overlaid testing is as follows:





Conclusion:

The problem that we want to solve is predicting pretty good. The results on testing data are good enough by taking in consideration of the amount of data we have. For achieving a good accuracy we must have a good variety of data to increase the performance. In this problem we did not provide the validation data set as we dont have much data. By adding validation data we can increase the performance on testing data significantly.