Microscopic Neuron image Segmentation

Submitted by EE19S042

October 2020

1 Abstract

It is becoming increasingly clear that mitochondria play an important role in neural function. Recent studies show mitochondrial morphology to be crucial to cellular physiology and synaptic function and a link between mitochondrial defects and neuro-degenerative diseases is strongly suspected. Electron microscopy (EM), with its very high resolution in all three directions, is one of the key tools to look more closely into these issues but the huge amounts of data it produces make automated analysis necessary. State-of-the-art computer vision algorithms designed to operate on natural 2-D images tend to perform poorly when applied to EM data for a number of reasons. First, the sheer size of a typical EM volume renders most modern segmentation schemes intractable. Furthermore, most approaches ignore important shape cues, relying only on local statistics that easily become confused when confronted with noise and textures inherent in the data. Finally, the conventional assumption that strong image gradients always correspond to object boundaries is violated by the clutter of distracting membranes. Here I have demonstrated different methods for segmenting out the membranes of the neuron cells using different traditional methods along with deep learning based Unet.

2 Introduction

The throughput of electron microscopes has increased significantly in recent years, enabling detailed analysis of cell morphology and ultra structure in fairly large tissue volumes. Analysis of neural circuits at single-synapse resolution remains the flagship target of this technique, but applications to cell and developmental biology are also starting to emerge at scale. On the light microscopy side, continuous development of light-sheet microscopes has led to a rapid increase in imaged volume dimensions, making Terabyte-scale acquisitions routine in the field.
In this project I have used distance watershed based method for initial segmentation based on region growing which is further used in multicut based segmentation method. The data set is taken from [1].The data set consists of 30 images for training and 30 for testing.Training set consists of original raw em image along with 17 number of precomputed affinities for each image. It also consists of ground truth and membrane segmented ground truth image.The contents of ISBI data set for a single image is as shown in figure 10

First image along with its 17 affinities is as shown in figure2

[1]ISBI EM Data set https://hcicloud.iwr.uni-heidelberg.de/index.php/s/6LuE7nxBN3EFRtLf.

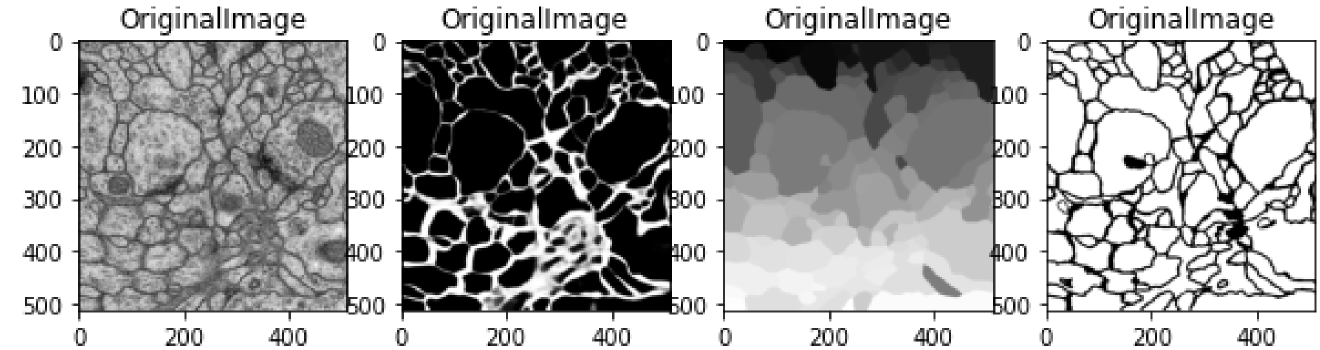

Figure 1: Em Dtaset (a) Original Image (b)first Affinity map (c)Ground truth (d) Membrane segmentation ground truth

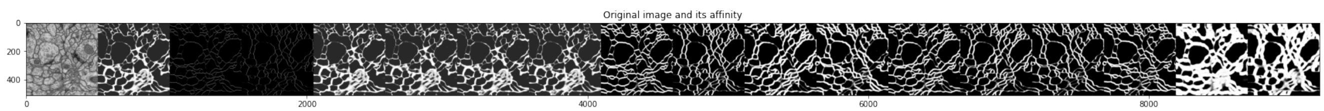

Figure 2: Distance transform based watershed$_{s}egmentationresults$

3 Methods :

3.1 Preprocessing :

First the image is enhanced using Hisogram based contrast enhancement method and gamma correction method. The results are as shown in figure 10

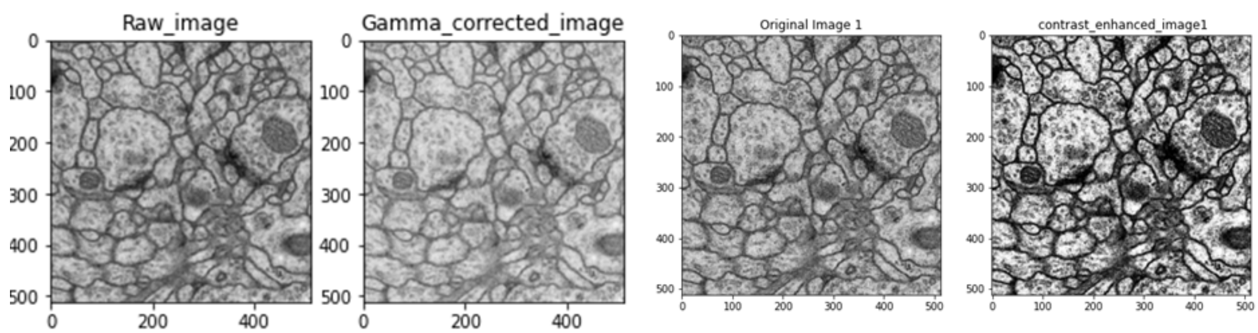

Figure 3: Image enhancement result (a) First 2 original image and gamma correction enhancement result (b) Last two are original image and histogram based enhancement method

From above two as histogram based method enhance the noise along with edges so, I have considered Gamma correction image for further processing. For removing the existing noise present in EM images I have applied 3 types of filters, Gaussian, Bilateral and Non local mean as described below.

3.1.1 Gaussian Filter

It is a linear filter that is used to remove noise from the image along with the blurring of image similar to average filter. It differs from average filter in the aspect that it uses different kernel from mean filter which is in the shape of bell curve (Gaussian PDF). It gives more weightage to center pixel unlike mean filter which gives equal wightage to all the pixels in the kernel. Convolution of gaussian kernel with the noised image result in filtered image. Here I have used 3 types of kernel size of 3,5,7 for filtering.

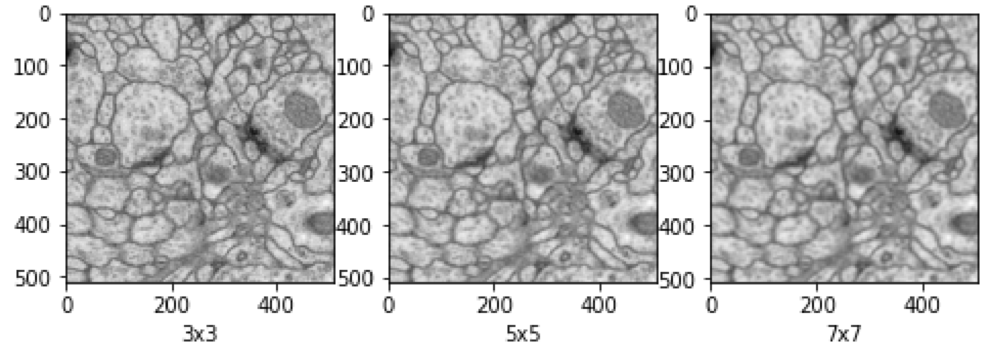

Figure 4: Result of Gaussian filtering with 3x3,5x5,7x7 filter kernel

In 2-Dimensional, Gaussian has the equation:

$$G(x,y) = \frac{1}{2 * \pi * \sigma^2} \exp -(\frac{x^2 + y^2}{2\sigma^2}) \tag{1}$$

where mean is (0,0) and σ^2 is the variance between 0 to 1.

3.1.2 Bilateral Filter

In em image processing, the suppression noise, while preserving edges and image details, plays a crucial role for the diagnosis. Bilateral filter, filter the image based on distance value as well as intensity value and hence helps in preserving the edges. Convolution of gaussian kernel with spatial kernel result in kernel for bilateral filtering which again convolved with the noised image result in filtered image.

$$BF[I_p] = \frac{1}{W_p} * \sum_{q \epsilon S} (G_{\sigma_S}(||\,(p-q)\,|| * (G_{\sigma_S}(||\,(I_p - I_q)\,||) * I_q \tag{2}$$

The result of Bilateral filter for filtering out Gaussian and speckle noise is as shown in figure**??**,

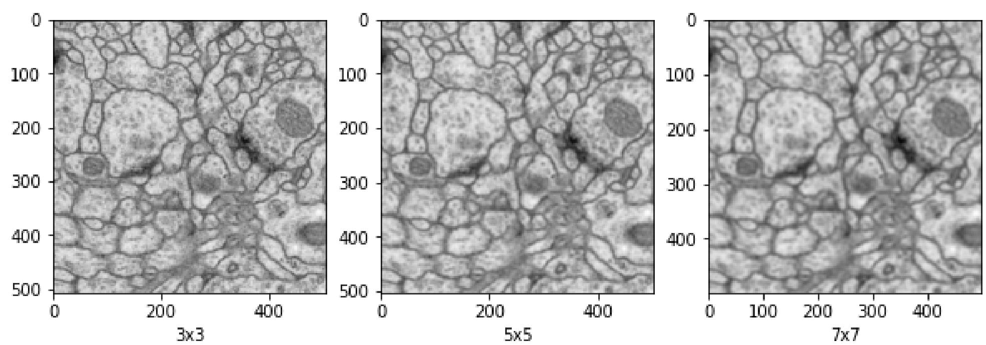

Figure 5: Result of Bilateral filtering with 3x3,5x5,7x7 filter kernel

3.2 Non local mean filter:

The non-local means algorithm replaces the value of a pixel by an average of a selection of other pixels values: small patches centered on the other pixels are compared to the patch centered on the pixel of interest, and the average is performed only for pixels that have patches close to the current patch. As a result, this algorithm can restore well textures, that would be blurred by other denoising algorithm.

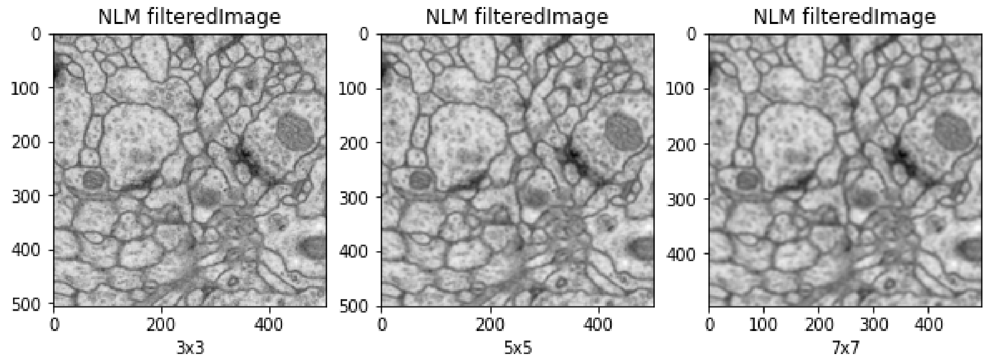

Figure 6: Result of NLM filtering with 3x3,5x5,7x7 filter kernel

3.3 Analysis of preprocessing results:

PSNR As the name explains, it is the ratio of the maximum/peak value of the signal to the noisy signal value. PSNR formally describes the quality of the reconstructed image after the application of any technique on it. Higher the PSNR, better the quality of reconstructed image. PSNR is expressed as:

$$PSNR = 10\log_{10}\frac{(peakvalue)^2}{MSE} \tag{3}$$

where Peakvalue is the maximum difference in the input image value and MSE is Mean Square Error and is computed as

$$MSE = \frac{1}{m \times n}\sum_{i=1}^{m\times n}(\hat{y}(i,j) - y(i,j))^2 \tag{4}$$

where $m \times n$ specifies the size of the image, $\hat{y}(i,j)$ is the recovered image and y(i,j) is the Original image. Table7 below shows the PSNR values for different filtering techniques with respect to different noises. σ_n^2 in db.
The peak to noise ratio of filtered image obtained from above filtering method is as shown in 7

Filter name	3x3	5x5	7x7
Gaussian Filter	30.120107851239624	25.2149124972 2159	23.2494028964750
Bilateral Filter	23.180543563071907	23.065058624251257	22.1469898572065
NLM Filter	23.18594507124242	20.21556313269697	20.589659099314

Figure 7: PSNR value for different Filters being implemented here

From above table, it is clear that for EM image both Gaussian and Bilateral filter performs well for image segmentation.

3.4 Image Segmentation Methods :

3.4.1 Threshold Based segmentation :

1. The ISBI 2012 data set I have considered here has both raw image as well as precomputed affinities. For threshoding based image segmentation the maximum of first two affinities is considered and threhoded with a constant thresholding value. Here, I have considered 0.9 as the default threshold value.

2. The 4 or 8 connected components of the thresholded image is found out, excluding the background value.

The result of above thresholding based method is as shown below,

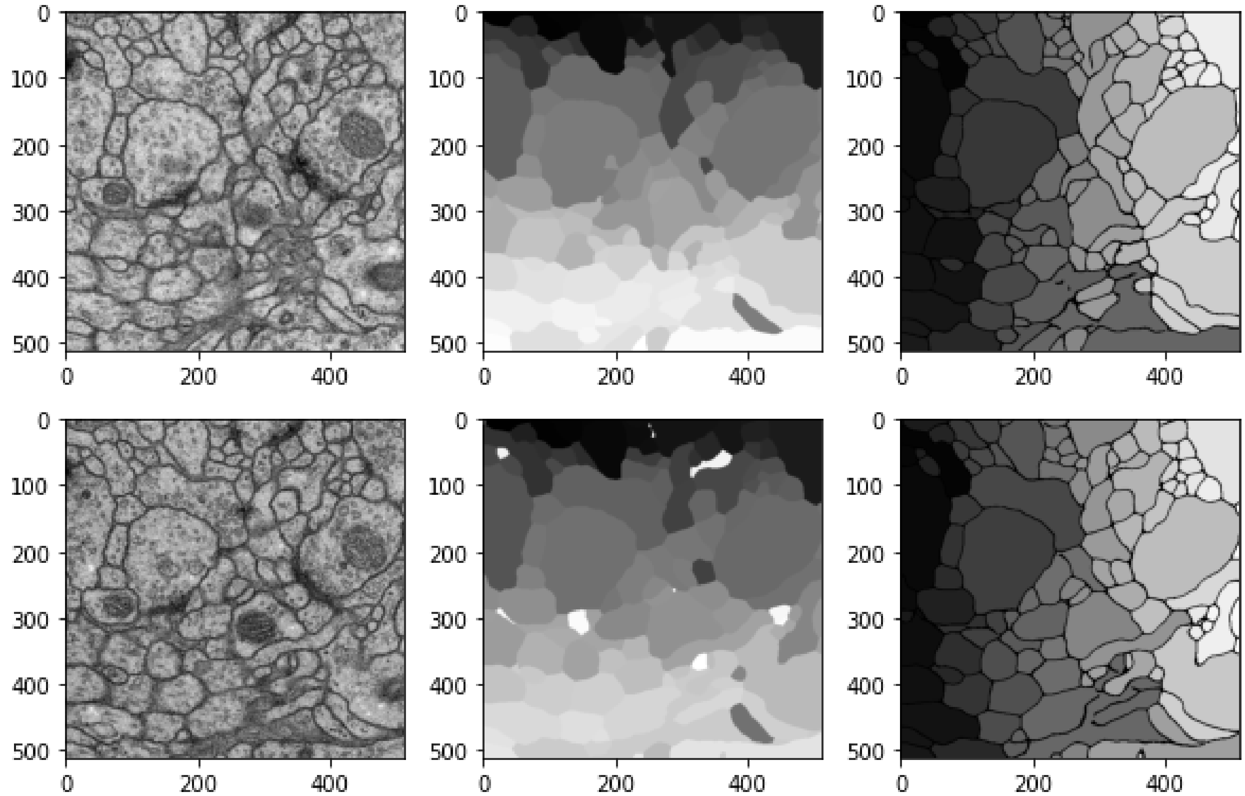

Figure 8: Thresholded based Image segmentation

3.4.2 Distance Transform based Watershed Image segmentation :

This is a region based segmentation method.In this case the intensity information available in image or statistical feature are used to segment the image.Watershed algorithm is based on technique of flood filling. As the paddles grow, the boundary merges. As the water increases the topology of the topography changes and the basins keep merging. The flowchart of Distance transform based watreshed segmentation is as shown below:
Algorithm:

1. First the boundary input is considered by finding out the mean of the the affinities given in the data set.For given ISBI 2012 data set the number of 17 affinities are given.The mean boundary is thresholded. Here I have considered threhold value as 0.25.

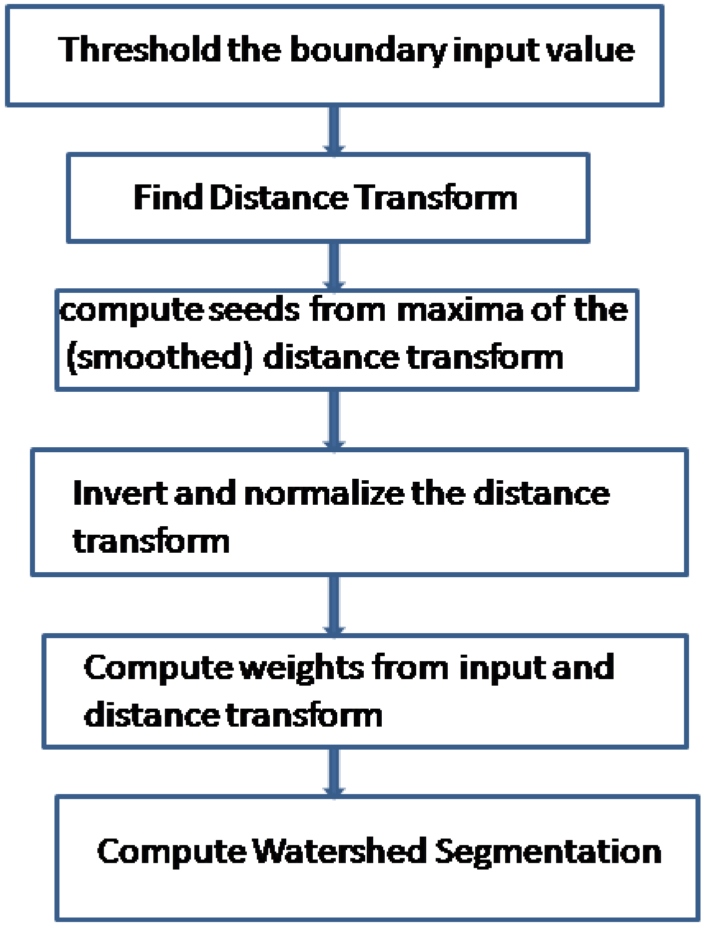

Figure 9: Distance transform based watershed$_{s}egmentationresults$

2. Distance transform is computed by calculating the eucledian distance transform. The distance can be computed in two ways. Either computing distance from all back ground pixels to the nearest foreground pixel or vice versa.If pixel pitch is given, it must contain the pixel distance along three axes. In this case distance is computed anisotrpically. If no pixel pitch is given then the data is treated isotropically with unit distance between pixels.

3. If image array has multiple channels, each channel is smoothed independently.
   - In this case first the image is smoothed using gaussian filter.
   - Maxima is computed using the smoothed image.Local maxima of image is marked with given marker.Seeds are computed on distance transformed image using the local maximum value calculated above.

4. The connected component from extracted multidimensional array is computed excluding the background.

5. The distance transform is normalised and inverted.

6. The weight (hmap) is computed from the input and its distance transform. The formula for calculating the weight is given as,

$$heat_{m}ap = alpha * (Gaussian_{s}moothed_{i}nputimage) + (1 - alpha) * distance_{t}ransform \quad (5)$$

where alpha is the probability map of the image.

7. Watershed is computed using the seeds and the weights.Algorithm for watershed is given as,
   - The seeds are used to form the basis for initial watershed.
   - For each group of pixel intensity level k,
     - If it is adjacent to exactly one existing region, these pixels are added to that region.
     - Else if it is adjacent to more than one existing region, then it is marked as boundary. start a new region.

The result of above Distance Transform based watershed segmentation method is as shown below,

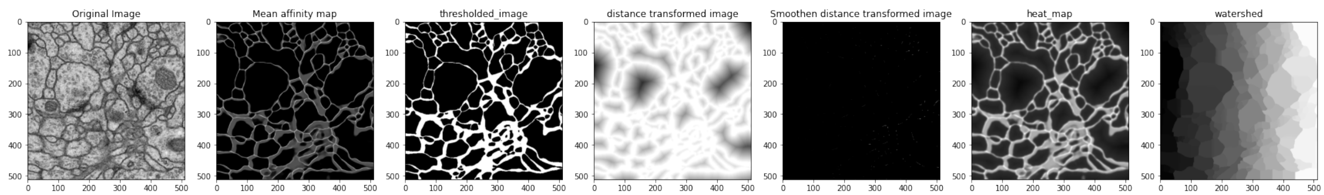

Figure 10: Distance transform based watershed$_{s}egmentationresults$

3.4.3 Multicut Image Segmentation :

Given a boundary probability map, multicut breaks the image up into superpixels and then merges them to recover segments limited by closed surfaces (no dangling edges). The main algorithm, known as multicut or correlation clustering.
compute superpixels by the watershed algorithm, running it on the distance transform of the boundary probability map. The seeds are computed from the local maxima of the distance transform (each maximum then gives rise to a separate superpixel). The motivation for this approach is as follows: Commonly used superpixel algorithms, for example SLIC, group pixels based on their similarity in brightness. This is, however, not desired here since it would result in superpixels which lie on the boundaries rather then be separated by them. Instead, for our use case, superpixels should group pixels based on which object they belong to. To achieve this, the high-contrast boundaries can be used. Here, the technique of choice is a watershed.
calculating the watershed directly on the boundary prediction, only works if the boundary prediction perfectly separates each object from its neighbors. The smallest hole in the prediction can lead to merging different objects into the same superpixel. Obtaining a perfect edge prediction is hard in itself and is often further complicated by boundary gaps due to errors in the sample preparation procedure. Consequently, we would prefer an algorithm which is robust to boundary holes.

Now that we have superpixels, we need to train the algorithm how to decide which of them should be merged and which not. The general approach we use was first described in this publication. Briefly, given the superpixels computed in the previous step, we now compute features on the edges of adjacent superpixels. These features include, for example, the summed intensity of the edge and the minimal and maximal intensity along it, as well as statisitics of the probability map and of the intensity inside the superpixels ("Select Features" button brings up a dialog which lets you choose features). After the features are computed, we predict – for every edge independently – if this edge should be dropped or preserved to achieve a correct segmentation. The "naive" way to proceed would be to then only take the edges which are classified as "to preserve" and use those as the final segmentation. This, however, would lead to an inconsistent segmentation with dangling edges inside the objects. Instead, we formulate a so-called multicut problem, where special constraints ensure no dangling edges are present and all segmented objects are closed surfaces, while following the classifier preferences for which edges to keep. This problem is NP-hard in general, but for highly structured graphs, such as our superpixel region adjacency graphs, it usually converges fairly fast. This can be achieved by performing the watershed on the distance transformation of the boundary prediction.

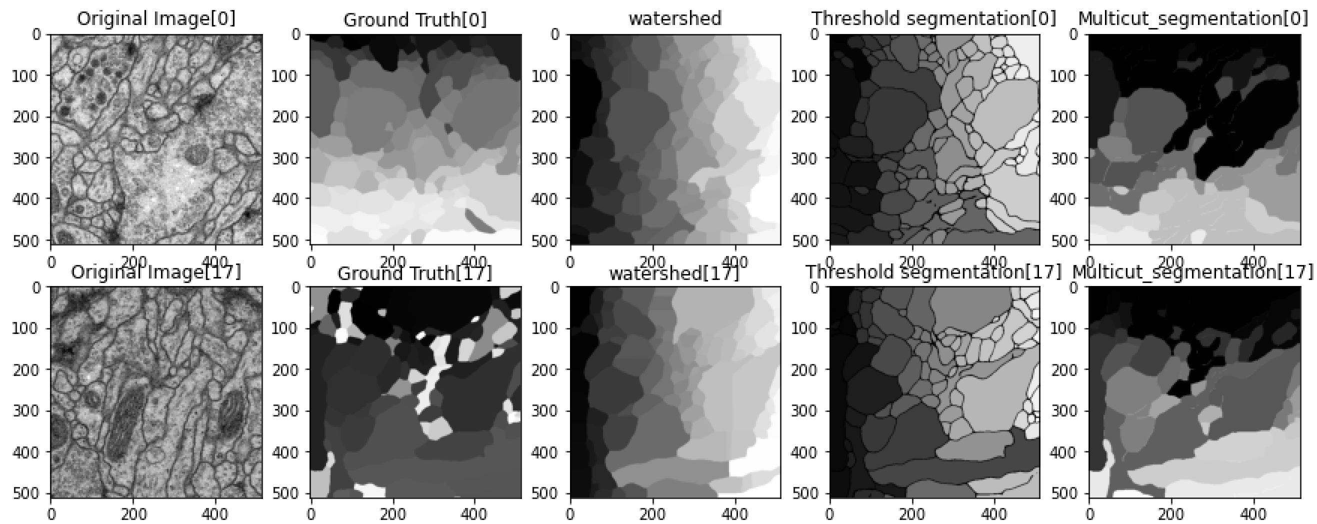

Figure 11: Segmentaion result based on watershed DT, Thresholding technique and Multicut based segmentation

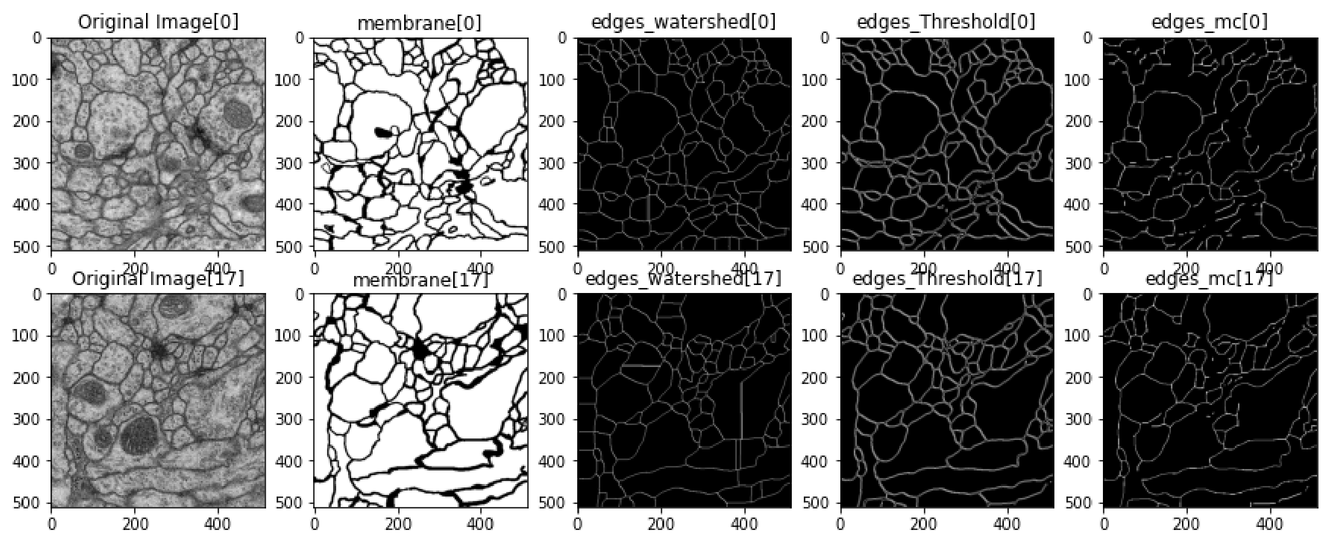

Figure 12: Membrane Segmentaion result based on watershed DT, Thresholding technique and Multicut based segmentation

3.4.4 Deep Network based Image Segmentation :

For comparison with of conventional image segmentation method with deep learning based method, Here I have used UNet for EM image segmentation. In the given data set the number of images for training is 30. To make the model more robust to color, brightness, rotation, height etc. Data augmentation

technique is used to generate more training data set. Here I have used batch size of 2 so in total 60 training images would be there for training.
For UNet ReLU is used as the activation function, He initializer, 10convolution layer and input size of (256,256). The Unet architecture is as shown in figure,

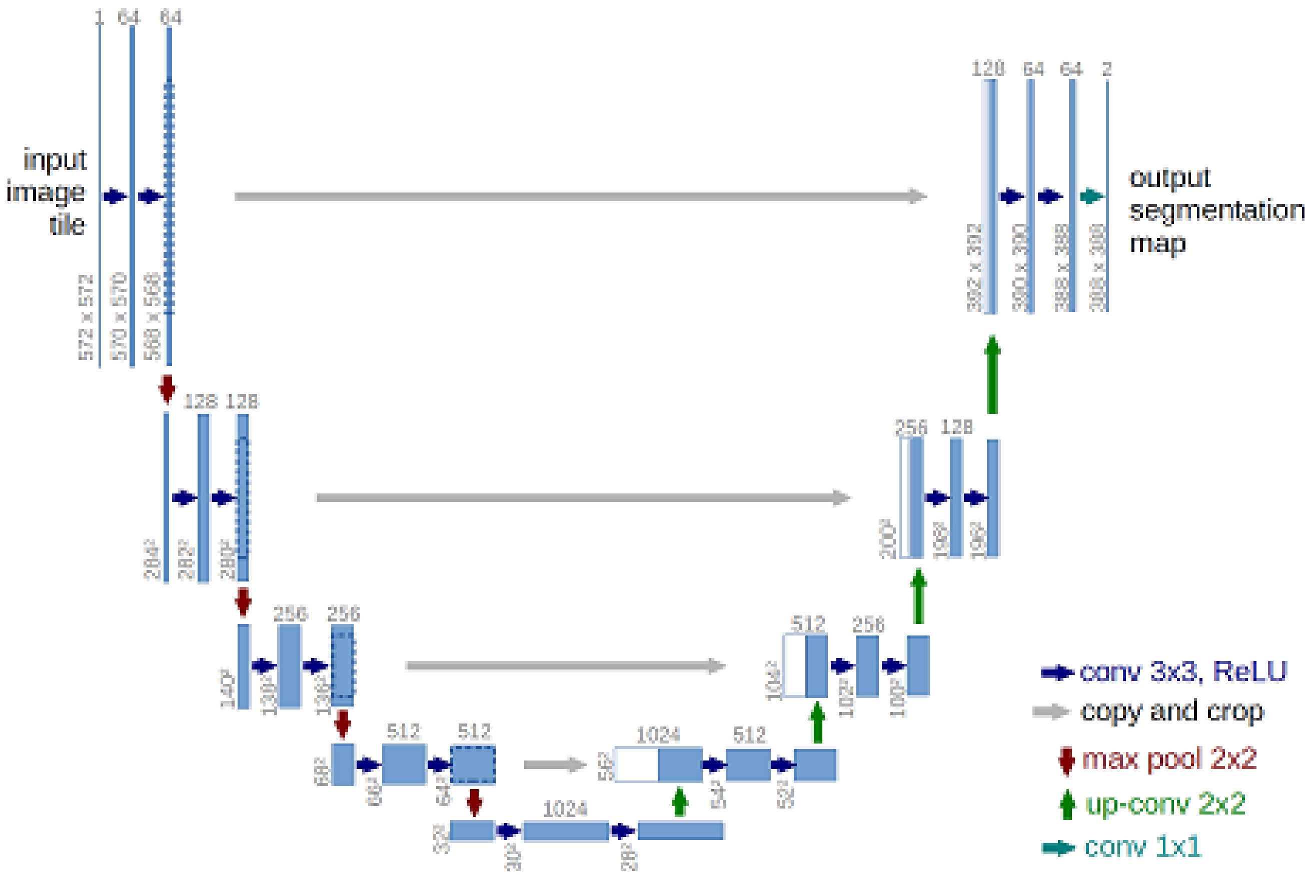

Figure 13: UNet architecture

Unet uses fully convolutional neural network model. Different operations that are typically used in convolutional neural network are,

1. Convolution Operation- There are two inputs to a convolutional operation
   - A 3D volume (input image) of size (9 x 9 x channels)
   - A set of 'k' filters (also called as kernels or feature extractors) each one of size (f x f x channels), where f is typically 3 or 5.

   The output of a convolutional operation is also a 3D volume (also called as output image or feature map) of size (nout x nout x k).

2. Max pooling operation -The function of pooling is to reduce the size of the feature map so that we have fewer parameters in the network.Toth convolution operation and specially the pooling operation reduce the size of the image. This is called as down sampling. Typical convolutional network, the height and width of the image gradually reduces (down sampling, because of pooling) which helps the filters in the deeper layers to focus on a larger receptive field (context).

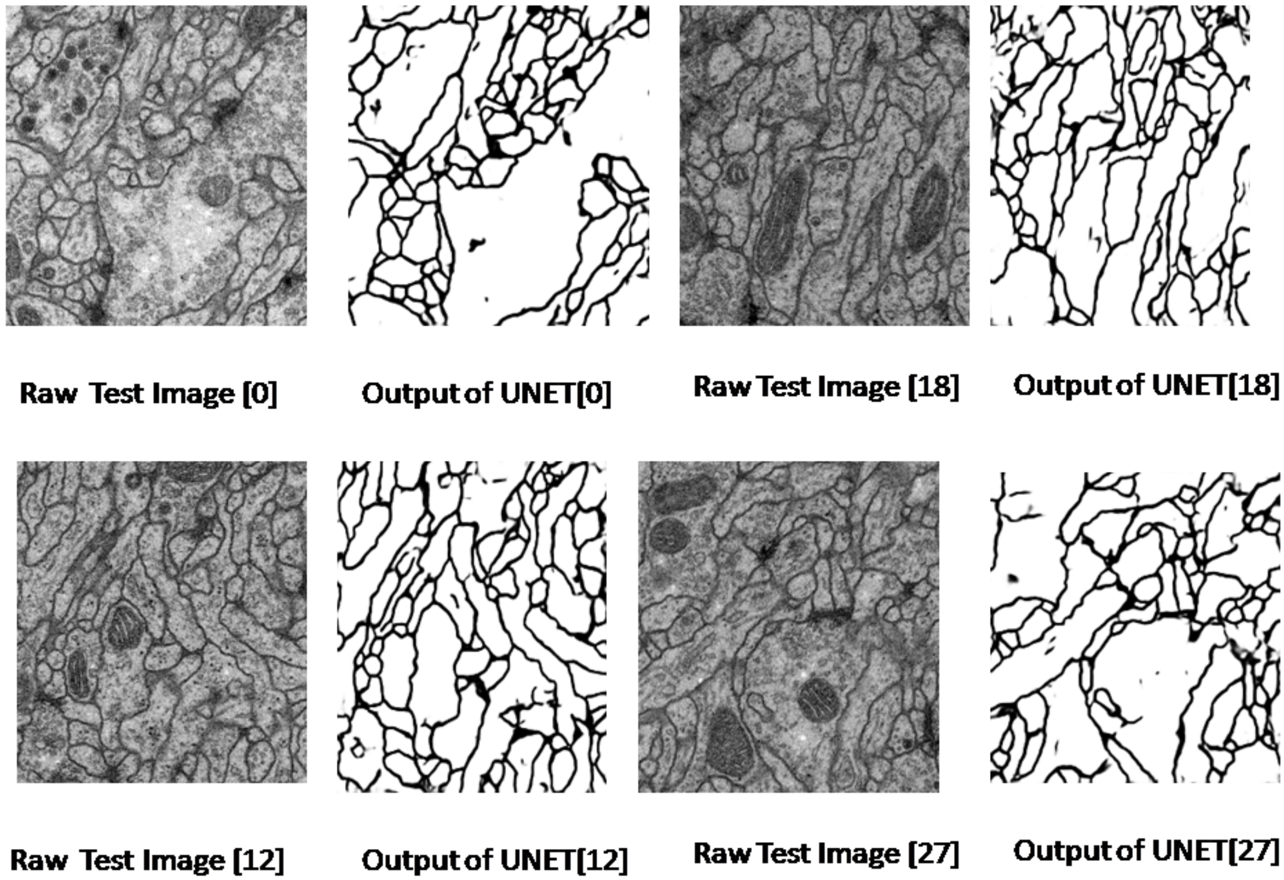

Figure 14: UNet test result

3. Up sampling - Up sampling the image convert a low resolution image to a high resolution image. The techniques for up sampling the image are bi-linear interpolation, cubic interpolation, nearest neighbor interpolation, un pooling, transposed convolution, etc. Here transposed convolution is used for up sampling.

4. Transposed Convolution -Transposed convolution (sometimes also called as deconvolution or fractionally strided convolution) is a technique to perform up sampling of an image with learnable parameters.

5. copy and crop -It helps up sampling path to re-cover fine grained information from down sampling path.

Optimizer-The optimizer used in the training process of UNet model is ADAM optimizer.

- Adam can be looked at as a combination of RMSprop and Stochastic Gradient Descent with momentum. It uses the squared gradients to scale the learning rate like RMSprop and it takes advantage of momentum by using moving average of the gradient instead of gradient itself like SGD with momentum. Let's take a closer look at how it works.

- Adam is an adaptive learning rate method, which means, it computes individual learning rates for different parameters. Its name is derived from adaptive moment estimation, and the reason it's called that is because Adam uses estimations of first and second moments of gradient to adapt the learning rate for each weight of the neural network.

Loss Function - The loss function used for EM semantic segmentation is Binary cross entropy. It is used specifically for binary classification. The equation of the loss function is given as,

$$H_p q = -\frac{1}{N}\sum_{1}^{N} y_i.\log(p(y_i)) + (1-y_i).\log(1-p(y_i)) \tag{6}$$

where $\log(p(y_i))$ is the label and $p(y_i)$ is the predicted probability.

4 Result Analysis

4.1 Metrics for analysing Segmentation :

The metric I have used for performance analysis of above methods is Adapted rand error, which is defined as: 1− the maximal F-score of the Rand index (excluding the zero component of the original labels), rand precision, and adaptive rand recall. The adapted Rand error; equal to $1 - \frac{2pr}{p+r}$, where p is the precision and r is the recall.
The adapted Rand precision: this is the number of pairs of pixels that have the same label in the test label image and in the true image, divided by the number in the test image.
The adapted Rand recall: this is the number of pairs of pixels that have the same label in the test label image and in the true image, divided by the number in the true image.The metric result of above method is as shown in figure 15

From above results 15 it can be interpreted that based in adapted rand error and adapted rand recall Multicut method performs better than other two, where its precision is very low and precision wise Thresholding method has performed better. Overally as adapted rand error takes into consider both precision and recall so it can be concluded that muticut has performed better than any of the above two.

Segmentation Method	Image Number	adapted Rand error	Rand precision	adapted Rand recall
Watershed - DT method	1st image	0.3578339975	0.87724995	0.5064487308
	19th image	0.3073503	0.8423443	0.5881316
	28th image	0.2478016	0.8811552	0.6561682
Thresholding method	1st image	0.17535196704	0.855002820	0.79637469
	19th image	0.0681548	0.99236231472	0.878284720
	28th image	0.0958068344	0.9432701281	0.8682251019
Multicut method	1st image	0.5091741898237 192	0.3393176153	0.886781094
	19th image	0.40968200	0.4330113511	0.9271310651 1
	28th image	0.4806141706 10	0.3595068534	0.9353541575 6

Figure 15: Metric for trainig result

5 Conclusion and Discussion

From Figure 12 It can be seen that the multicut based method has eliminated the false merge and split regions of distance transform based watershed method and has given membrane up to single pixel boundary value. To consider domain region lifted multicut based segmentation method can be implemeted to consider more region than just neighborhood instensity values and the graph is computed considering connecteness of the region around a single region.

6 Reference

[2] [3] [4]

[2] https://openaccess.thecvf.com/content_ECCV_2018/papers/Steffen_Wolf_The_Mutex_Watershed_ECCV_2018_paper.pdf.

[3] https://arxiv.org/pdf/1905.10535.pdf.

[4] ISBI EM Data set https://hcicloud.iwr.uni-heidelberg.de/index.php/s/6LuE7nxBN3EFRtLf.