Electron Microscopic Data Analysis

Tushar Singhal, Prashant Kolkur
Master of Advanced Studies,
Data Science and Engineering
Advisors: Dr. Mark H. Ellisman, Dr. Ilkay Altintas

1. Abstract:

Biomedical Image segmentation has been an important piece of work for researchers and Scientists. Image segmentation of medical images helps Scientists and Researchers in their research work for solving a particular medical science problem. In recent years, Deep Learning has brought a major breakthrough in the field of Medical Image Segmentation. This project aims image segmentation to highlight cell objects such as mitochondria, nuclei and vessels from unique biopsy image samples of cerebral cortex from patients suffering from Alzheimer's disease (AD). The final output from the model has segmented objects highlighted which are present in a given image. A 3D convolutional neural networks (CNN) for deep learning-based image segmentation known as DeepEM3D and MultiResUnet are being explored to extract spatially accurate and quantitatively valid geometries of key structures associated with the cell-level AD neurodegenerative processes. Model evaluation on the test-set resulted with mean F-Score of 0.69 on DeepEM3D and 0.82 on MultiResUnet. Models are also been tested for Robustness and Scalability. Model Robustness is tested by adding Salt-and-Pepper noise of about 5% and 10% of the image. Results with 5% noise showed F-Score of 0.7 while it was 0.56 for the image with 10% noise. Models are scalable to any input image size since the input images are cropped into sub-images of fixed size of 1024x1024 pixels. Prediction on bigger input images showed a linear increase in runtime based on the number of image packages created out of big images.

2. Introduction:

The "Electron Microscopic Data Analysis" project aims to support the processing, analysis and dissemination of large-scale 3D electron microscopic (EM) data derived from a remarkable collection of legacy biopsy brain samples from patients suffering from Alzheimer's disease (AD). Alzheimer's disease is a progressive neurodegenerative disease, leading to dementia accompanied by several structural changes in the brain of patients.

The project intends to facilitate the processing and downstream analysis of complete whole cell reconstructions of neurons from unique biopsy image samples of cerebral cortex taken from AD cases by Robert Terry in the 1960's, focusing on early onset cases, where cells effected to differing extents are neighbored by cells without AD-associated forming paired helical filaments, (PHF) now known to be largely made up of tau proteins. These samples were screened and preliminarily reported on by Ellisman, Masliah and Terry (Ellisman et al., 1987) and manifest near perfect preservation of ultrastructure PHF and amyloid accumulations as well as modifications to subcellular organelles and cytoskeletons of the cell bodies, axonal and dendritic processes.

Image samples are EM reconstructed to produce small 3D data volumes to employ serial block-face scanning EM (over much larger fields in XYZ) to provide a large reference collection of fully reconstructed brain cells. Since the project aims image segmentation for analysis, 3D convolutional neural networks for deep learning-based image segmentation is performed to extract spatially accurate and quantitatively valid geometries of key structures associated with the cell-level AD neurodegenerative processes (i.e., perform large scale, automated image segmentation).

Understanding and investigating the components and network structure of a brain neuron circuit is one of the top priorities in neuroscience research which will help in research on the Alzheimer’s disease. High definition and clean Electron Microscopic brain images that are available needs to be segmented to find various components of brain such as Mitochondria, membrane, Nuclei in the brain image samples. Segmentation of the images will help Scientists and researchers in their research work. Given the large number of image samples, manual image segmentation is not practical.

The large number of image samples are available, and the image samples are manually labelled. These images along with the labelled images can be used to apply a Data Science method to build a model which can effectively and efficiently segment the images with a high precision. That model can also be used to segment any new input image samples which are not labelled.

3. Data Acquisition:

Raw data (or data) is coming from Cell Image Library (CIL: http://cellimagelibrary.org/cdeep3m) as shown in Figure-1. Cell Image library is a public and easily accessible resource database of images, videos, and animations of cells, capturing a wide diversity of organisms, cell types, and cellular processes. The purpose of this database is to advance research on cellular activity, with the ultimate goal of improving human health. This data is accessed manually from the website and copied to the memory disk for processing. The data is a serial block face scanning electron microscopy (SBEM) 3D data volumes. Serial block-face scanning electron microscopy (SBEM) is a way to obtain high resolution 3D images from a sample. This method is particularly good at imaging large fields-of-view in X,Y,Z that is 3dimension at nanometer resolution. The data is brain image samples of patients suffering from Alzheimer’s disease. The Alzheimer's image data images are of very high definition. The size of a typical image in multiple datasets is in the range of 16000x10000x400 pixels images.

Figure-1: Data is collected from Cell Image Library

Data source, as mentioned above, for this project are image samples of size 16000x10000x400 pixel as shown in Figure-2. These raw images are high definition images in PNG format (Portable Network Graphic). The data is collected from the Cell Image Library data server.

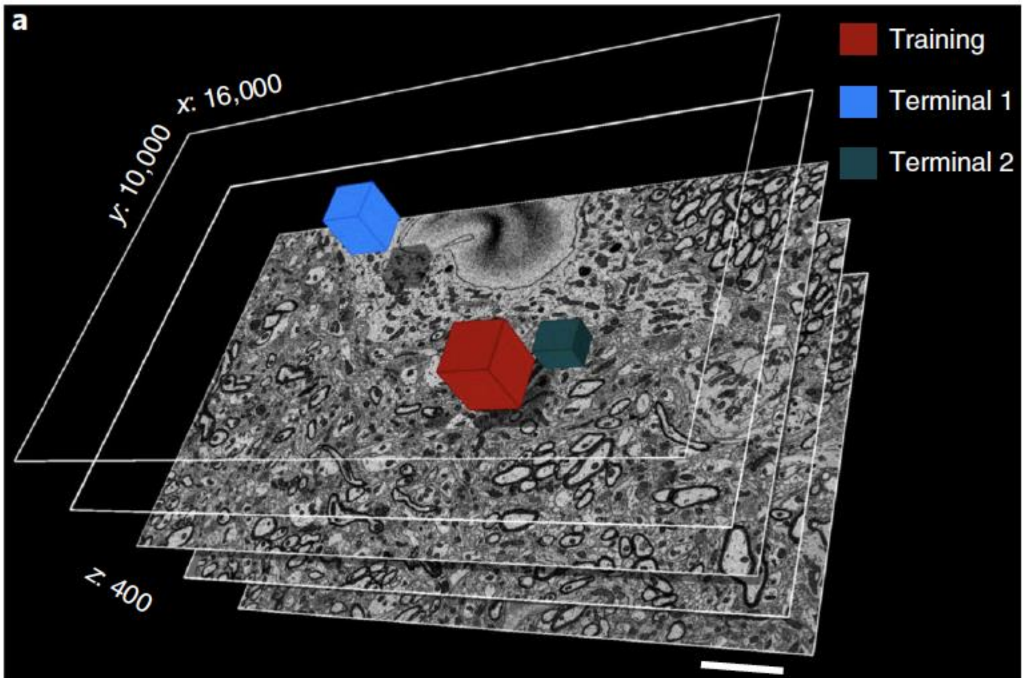


Figure-2: An image of size 16000x10000x400 pixels with Training and Terminals regions

4. Exploratory Data Analysis (EDA):

Data, coming from Cell Image Library (CIL), is rich in information and is of size 16000x10000x400 pixels. Performing EDA on this data is very important to decide on solution architecture. After performing Exploratory Data Analysis on these images, following finding were drawn:

- Original Image data is of size 10000x16000 pixels with 400 images deep. Modeling any solution architecture for such huge data is irrational because of computational limitations. That's why data needs to be down-sampled or sub-divided to meet the architecture requirements.
- Subpart from original images should be taken (images of size 1024x1024 pixels images), as shown in Figure-2 (colored cubes) and should be used to feed to the proposed solution.
- Data images has many objects into it such as mitochondria, nuclei and membranes.
- Data available for this project is non-isotropic images. Some region of some images contain more objects than the other regions.
- Data images don’t contain any noise, images are clean.
- Data needs to be augmented to cover most orientation of the objects present.
- Data is present in the form of many images for a given brain sample and there are many folders corresponding to each brain sample. Data should be preprocessed by loading these

images, one folder at a time and sorted and filtered to create a stack of images. The image stack should then written out in a file so that it can be easily used by solution architecture.

- Data label images are all binary images having values 1 and 0. 1 represents the area of interest. It is seen that the ratio of number of 1’s is very less when compared to number 0’s

After performing EDA, it can be finalized that only a small fraction of different datasets will be used for solution architecture for performance optimization and scaling. A typical image within the dataset will be of the size of 1024x1024 pixels.

5. Hypothesis:

After sub-dividing original images of size 10000x16000 pixels into images of size 1024x1024 pixels, Exploratory Data Analysis on those images’ dataset (From CIL) shows that Data is non-isotropic. Some region of images contain more objects than the other regions. Segmenting such images having varied number of objects using conventional image processing algorithms is difficult. So a deep learning-based Convolution Neural Network (CNN) can be deployed as solution architecture to properly segment the images including non-isotropic images.

As per the EDA on the data labels, since only a small part of the image needs to be segmented, precision and recall are the best ways to use as evaluation metrics because of large True negative values and less True positive values. Hence the measure of success can be calculated by evaluating the precision and recall of the predictions. Data available to the project should have labelled images containing mitochondria, nuclei and Membrane, and hence accuracy of the model can be calculated easily using precision and recall. Precision of about 80% and recall of about 95% should indicate the success of the model.

A typical solution architecture, in this case a CNN, should have a pipeline structure as shown in Figure-4 which will get explained later in this report.

6. Data Engineering

As described above, a CNN model is used towards solution architecture and to realize this network, Image Data should be sub-divided into three categories: Training data, validation data and test data. The description of all 3 datasets is described below:

➢ **Training data:**

Currently 80 images of data and labels are being used for training the model. Each raw data image is of size 1024x1024 pixels gray scale. And each label corresponding to each image is also an image of size 1024x1024 pixels binary scale (containing only binary data 0 or 1). More images along with their labels will get added during the course of the project.

- **Validation data:**

Only 15 images are being used (data and their labels) for validation set. Each raw data image is of size 1024x1024 pixels gray scale. The validation label images are of size 1024x1024 pixels binary scale. More validation images will get added later.

- **Test:**

Currently, only 5 images are being used for testing the model. Each raw data image is of size 1024x1024 pixel gray scale. For accuracy, the testset image will get examined manually and performance of the model will get evaluated based on this result.

Table-1 describes the dataset name, folder and sub-folder name, their location and destination in pipeline and their size.

Dataset name	folders	Sub-folders	Dataset location	destination	Data size
mito_testsample	training	images	https://drive.google.com/open?id=1ddo1SOk9bHzacX6e_7v2Bp0yRSklbcRF	Preprocess block	80 images (75 MB)
		labels			80 images (770 KB)
	validation	images		Preprocess block	15 images (15 MB)
		labels			15 images (100 KB)
	testset	-		Model block	5 images (5 MB)

Table-1: Dataset - training, validation and testset definition along with their location

7. Data Visualization

Since the images are of 3-D nature, not all types of image viewers can be used to visualize the data. It is decided to save the data as a stack of images, saved as h5 format. HDFView and h5pyviewer software are available to view h5 files. Hdfview 3.0 is used in this project to visualize the data as shown in Figure-3.

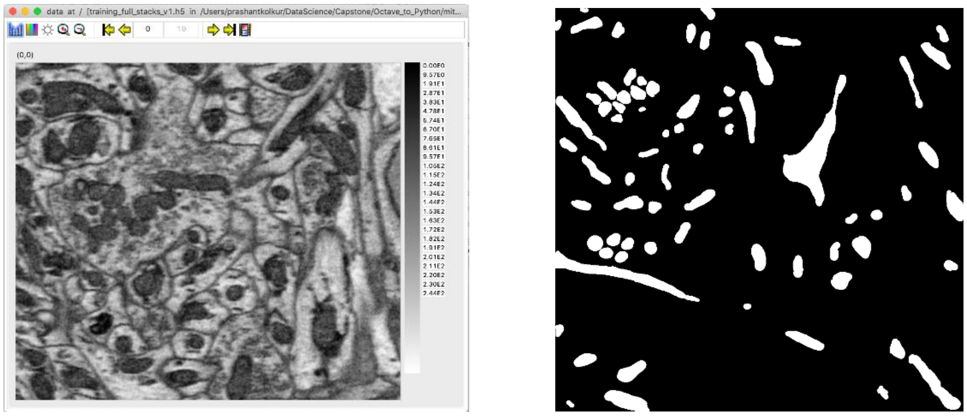


Figure-3: 3-D Image visualization of a typical image (left) and label (right)

8. Data Pipeline:

Data is presented in the form of many images for a given brain sample and there are many folders corresponding to each brain sample. Preprocesses loads these images, one folder at a time and sorts them and filters them and create a stack of images as shown in Figure-4. The image stack is then write out as h5 file so that it can be used as input to the model which predicts the output based on the current learning algorithm.

Post-processing on these outputs would result in segmented images that identifies objects such as mitochondria, membranes, nucleal etc.

Above processing is sequential in the sense that each block works on a given sample and forwards it to the next block, before proceeding to the next sample. All of the above processing is time multiplexed in a way that postprocessing is working on sample-1, while model is busy in predicting sample-2 and preprocessing is working on sample-3 as shown in Figure-4.

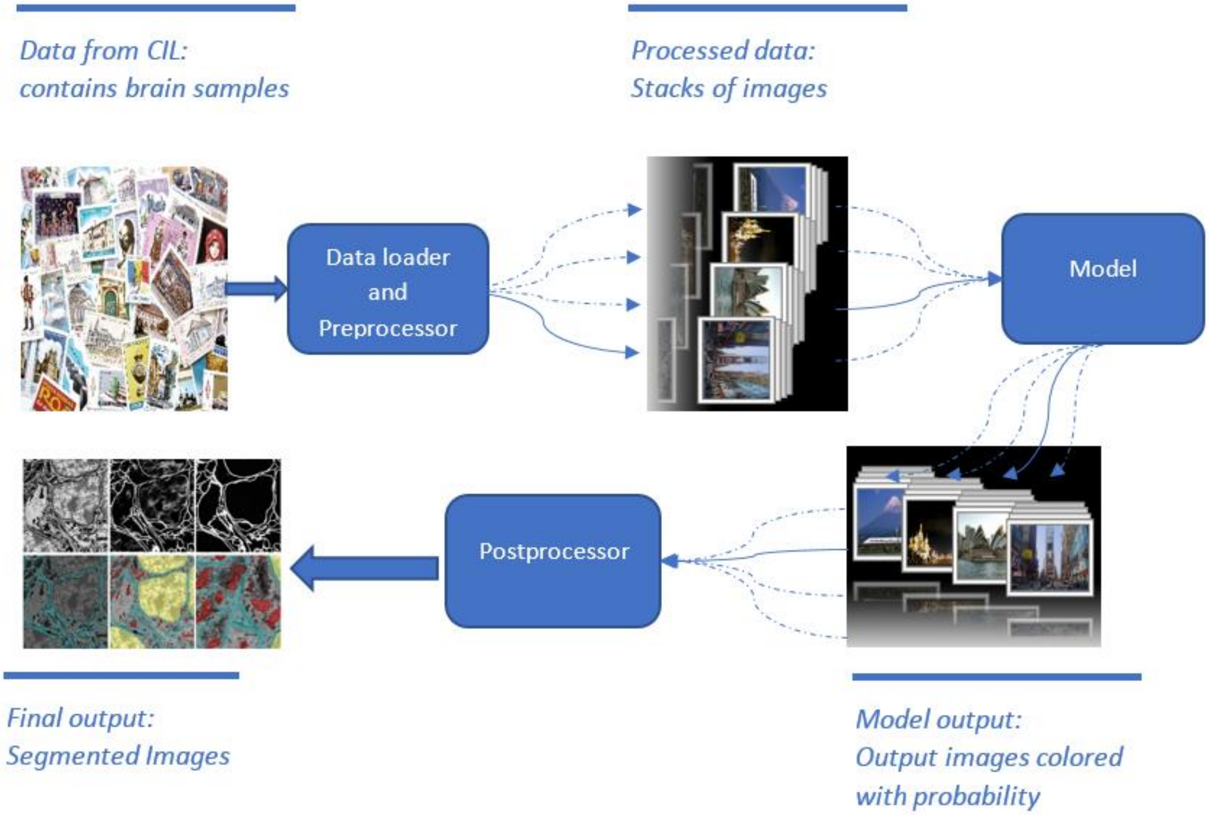


Figure-4: A CNN based solution architecture

9. Data Preprocessing:

Samples in the form of discreate images are stored in folders, one folder for each patient brain. Preprocessing module loads these images and processes it and makes a stack of images for each folder so that it can be used by model for prediction.

Preprocessor can accept a h5 file, a tiff file or a folder containing PNG files. It then creates a stack of these images and passes it to subsequent sub-modules for further processing.

- **Data Cleaning:**

1. Firstly, the size of images are checked and if minimum size requirement is not met, images are padded with zeros to match this requirement.
2. Training, validation and test data is checked for no-binary. These data value should vary from 0 to 255 (8 bits value), it cannot be binary data. Model will throw out error if training/validation/test data is binary data.
3. Training and validation label data is checked for binary. Label data is also an image which should contain pixels with value either 0 or 1. If label data is found to be no-binary, model will throw out error.

- **Data Wrangling (Augmented data):**

1. Data in preprocessor goes to augment sub-block, where augmented data is generated.
2. Various rotations are applied on data (0, 90, 180, and 360 degree) to generate augmented data.
3. Data (original and rotated) is also flipped to generate more data.
4. Finally, this array (stack) of data is written out in h5 format to be available for reading by model.

10. Processed and integrated data storage format:

The PNG stack of image files are stored in a HDF5 format. HDF5 (.h5, .hdf5) is a file format suitable for storing large collections of multidimensional numeric arrays (e.g. image files). HDF stands for Hierarchical Data Format. A HDF5 file can hold groups of datasets, where datasets are multidimensional arrays of a homogeneous type, and a single HDF5 file can thus act like a file system, which is more portable and efficient than having an actual folder that holds many files.
Table-2 describes the processed dataset name, input dataset name, their location and sizes, and scripts location.

Processed dataset name	Input dataset name	Dataset location	Scripts location	Data size
augmentedtraining	Mito_testsample/training	https://drive.google.com/open?id=1CiQa_4Y8PuS34kYkjFfzPtw5_1vh3C3c	https://drive.google.com/open?id=1TEOcJjBPdNqFwGtnhxj3BU_mWUXQD4PU	2.68 GB
augmented validation	Mito_testsample/validation			1.34 GB

Table-2: Processed and augmented Dataset information

Augmentedtarining dataset contains the h5 files which are stack of training images and can directly be used by the model. It contains data and labels both.

Augmentedvalidation dataset also contains the h5 files which are stack of validation images and their labels.

11.Features used by the model:

The input data (images) are in grayscale and all the pixels from an image contribute towards feature set. For an image of width W and height H, total feature set will have WxH features (pixel value).

Table-3 describes the input dataset (to model) name, their location and sizes and total features:

input dataset name	Dataset location	Feature set	Data size
augmentedtraining	https://drive.google.com/open?id=1CiQa_4Y8PuS34kYkjFfzPtw5_1vh3C3c	20x1024x1024	2.68 GB
augmentedvalidation		10x1024x1024	1.34 GB

Table-3: Dataset information to be used by model

12.Data environment setup:

No cloud setup was made for data storage since data is coming manually from CIL. But Amazon Web Service (AWS) Cloud has been set up to run complete model there, to use more than one CPU/GPU.

San Diego Super computer Comet cluster is also used to run this project.

All data (images) are saved as flat files, as such no database is used other than h5 format.

13.Solution Architecture:

The final product will segment objects such as mitochondria, nuclei and membrane segments in the brain image samples. To achieve this using CNN model, several steps can be taken so that success can be measured at each step.

- The first model implementation will be a basic model which can segment only one kind of object in the image: Mitochondria, Nuclei or membrane using only one image at a time in the model. Total of 3 models can be developed, one for each object.
- The second step consist of implementing a model which will be a 3D model and will be using a stack of 3 images to learn and predict. Again, there will be three separate models, each will segment one type of object out of three.
- The third model will also be a 3D model but will be using a stack of 5 images to learn and predict and it can segment only one kind of object in the image. A total of 3 models will be deployed, one for each object.
- Finally, all the three models can be combined to segment all 3 objects in one go, to get better prediction.
  - The above described workflow is shown in Figure-5. Each step can be measured easily for its success using Dice metric, also called as F-Score.

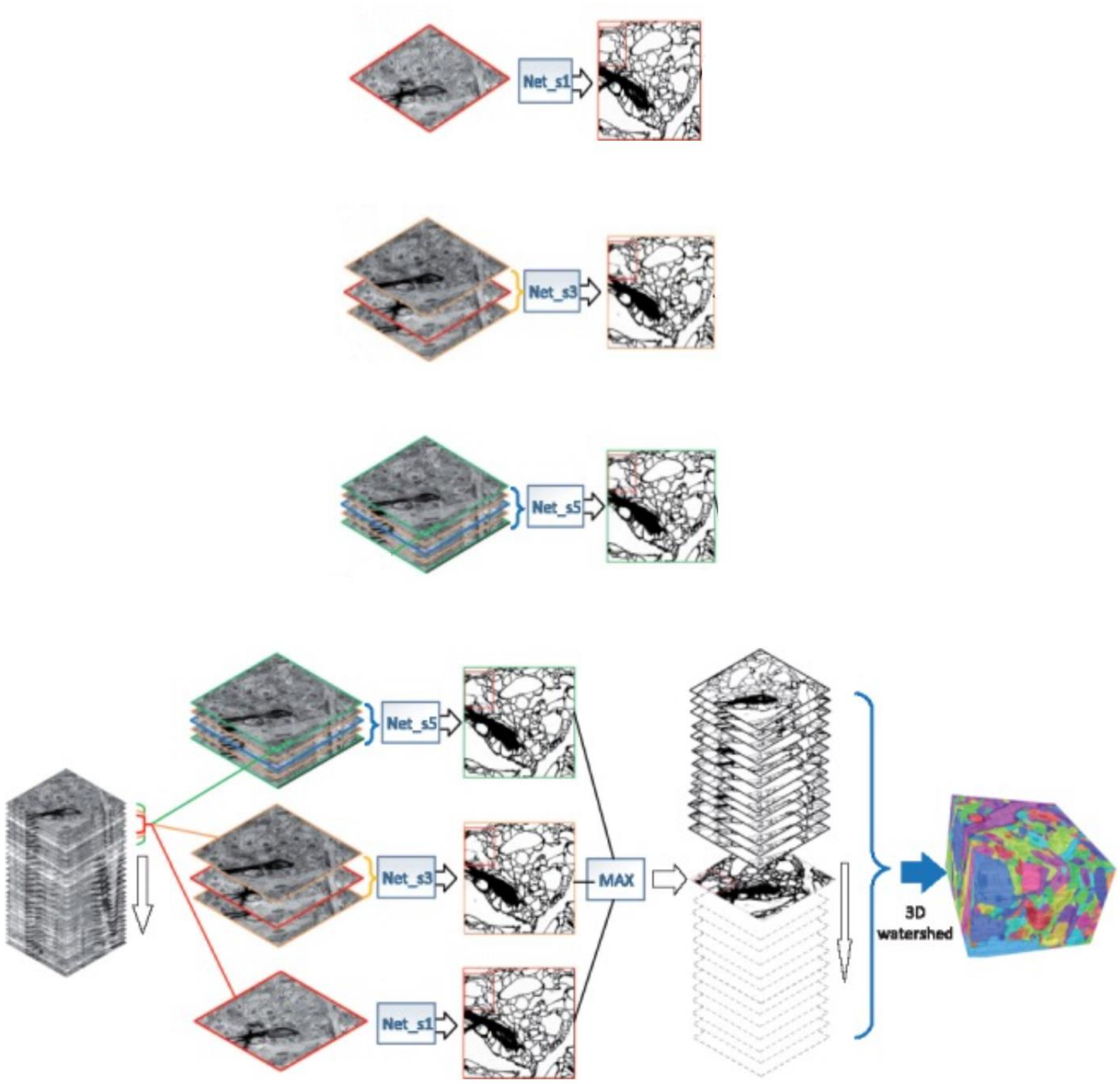


Figure-5: Solution Architecture: 1fm, 3fm, 5fm and combined model in order

14. Modeling:

➢ DeepEM3D-Net Model:

A convolution Neural Networks is used for segmentation of images. This neural network is based on deep learning where it will have several convolution layers and several neural layers. Currently the models being implemented is taken from the following paper: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6248556/

The architecture of the DeepEM3D-Net model, based on the above paper is shown in Figure-6. This model will go through rigorous training based on training dataset. To evaluate the score of the resulted model, validation dataset can be used. Validation dataset has input images and labelled output which can be used to calculate validation accuracy and finally, model success.

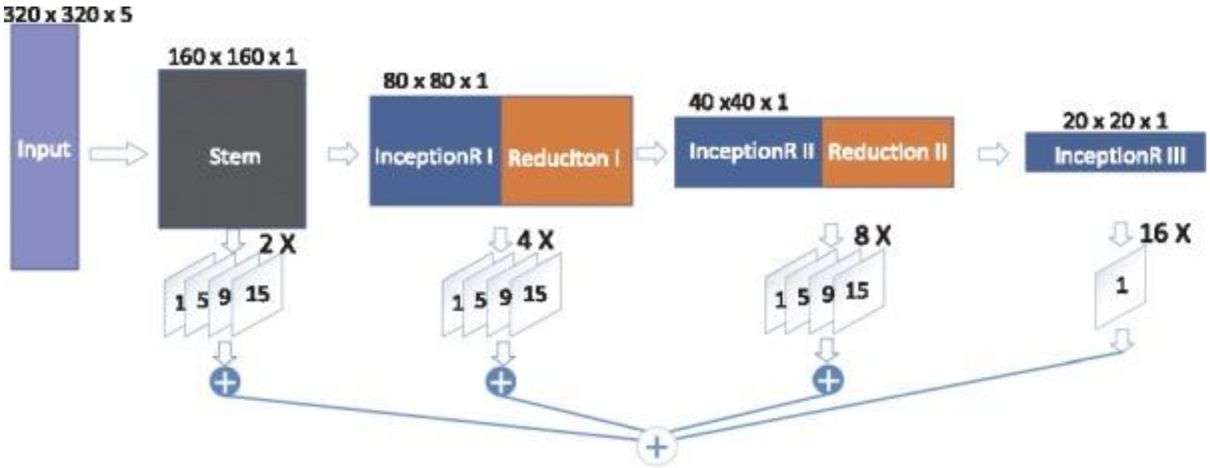


Figure-6: The architecture of DeepEM3D-Net

- **MultiResUNet Model:**

CNN based model "MultiResUNet : Rethinking the U-Net Architecture For Multimodal Biomedical Image Segmentation", taken from https://arxiv.org/abs/1902.04049, is build to check if more enhanced results can be obtained. The architecture of same is presented in Figure-7. Again, to evaluate this model, same procedure can be used by using validation data.

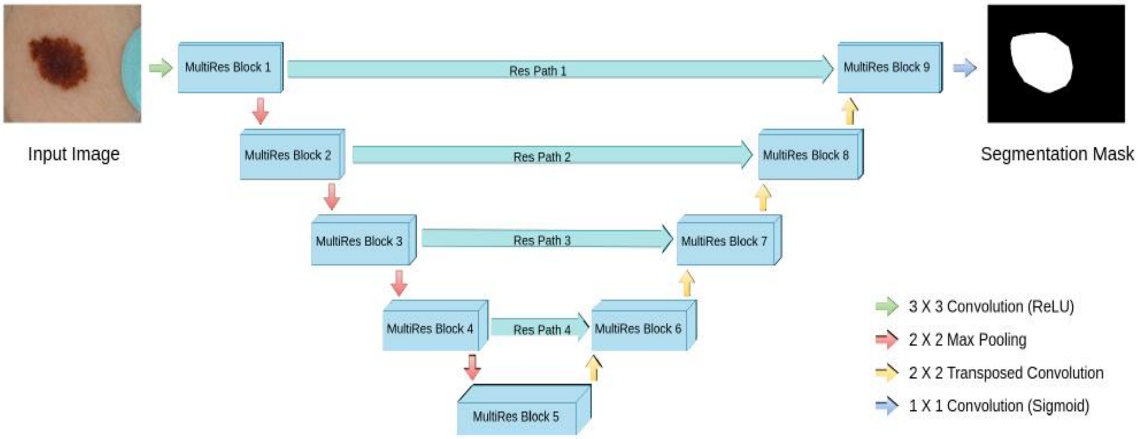


Figure-7: The MultiResUNet architecture

15. Model Training:

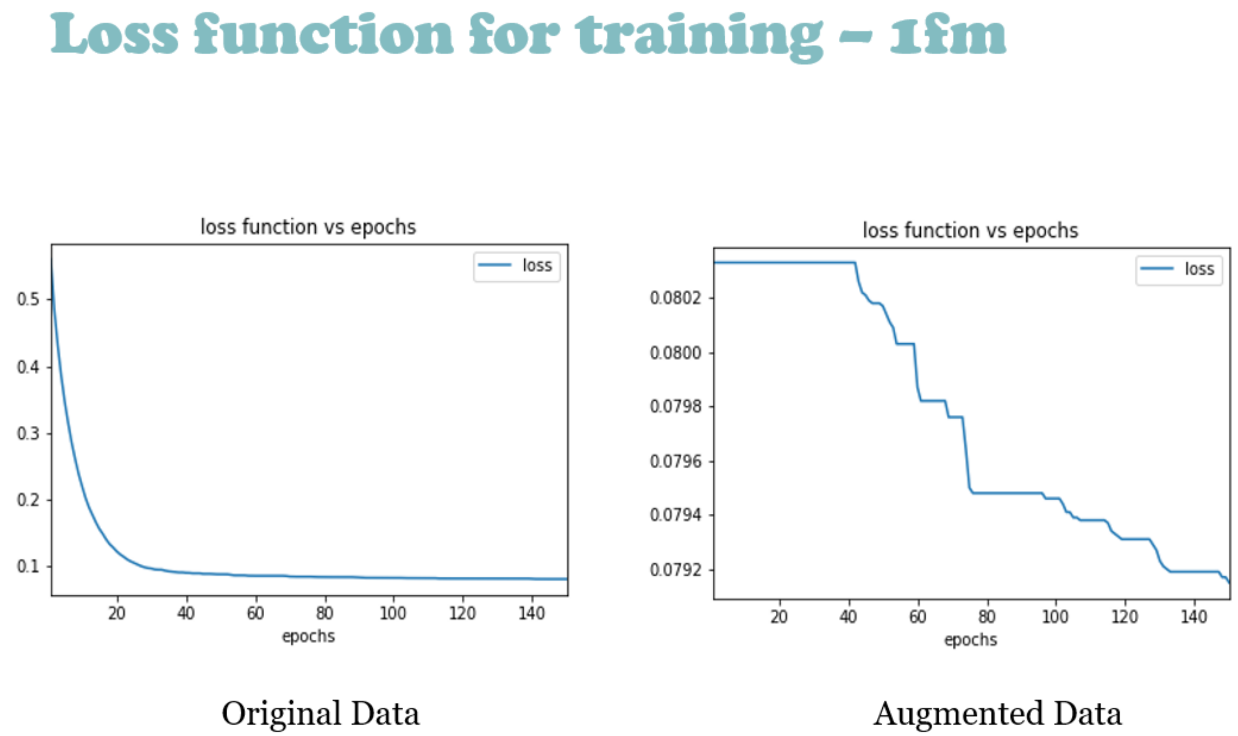
- **1fm - 2D model and 3fm/5fm - 3D model:**

    - Used 4 Tesla P100 GPUs with 12GB per instance
    - DeepEM3D-Net is implemented in Caffe.
    - MultiResUNet is implemented in Keras
    - Training is done using

        - 80 images, each of size 1024x1024 pixels
        - 16 different augmentations
        - 150 Epcohs per augmentation

    - Training time for 1fm model: 24hours-30hours
    - Training time for 3fm/5fm models: 48hours-72hours each

Loss function for training 1fm, 3fm and 5fm on original and augmented data are shown in Figure-8, Figure-9 and Figure-10 respectively. It is seen that loss didn't increase after around 150 Epochs in the augmented data and loss didn't increase after 40 Epochs in original data. Hence the model was trained for 150 Epochs.

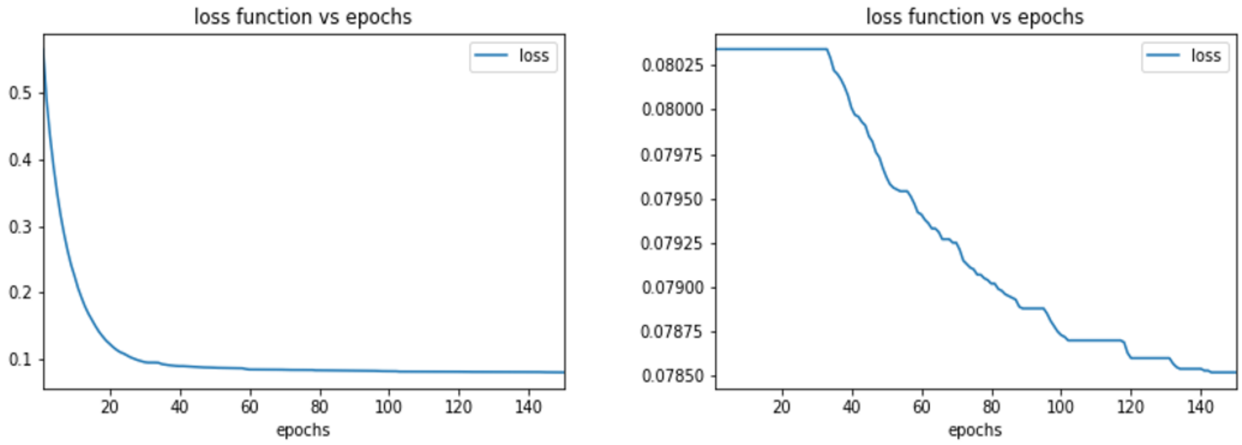

Loss function for training – 1fm

Original Data

Augmented Data

Figure-8: Loss function for training 1-fm

Loss function for training – 3fm

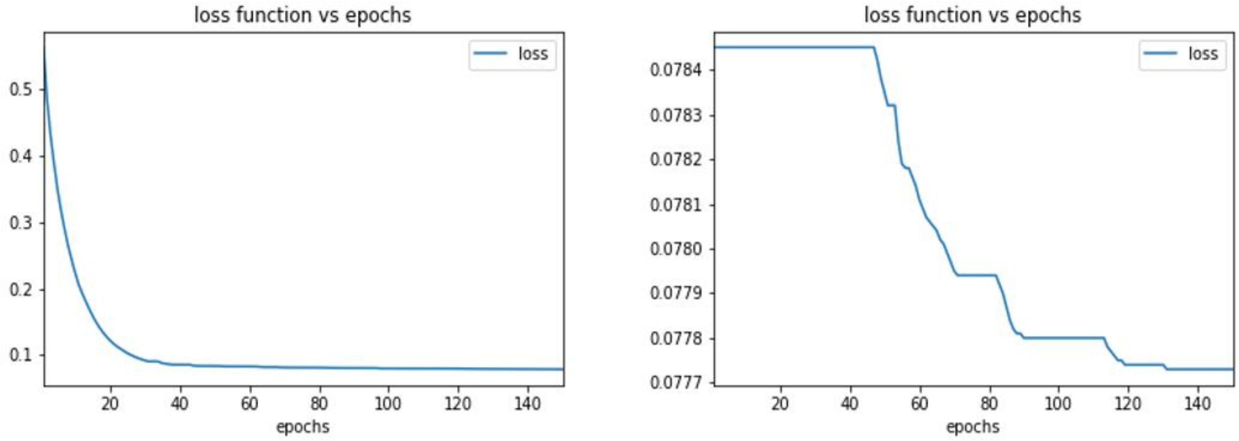


Original Data

Augmented Data

Figure-9: Loss function for training 1-fm

Loss function for training – 5fm



Original Data

Augmented Data

Figure-10: Loss function for training 1-fm

16. Model Prediction:

- Used 4 Tesla P100 GPUs with 12GB per instance
- Prediction is done using
  - 5 images, each of size 1024x1024 pixels
  - 16 different augmentations
- Threshold value used for DeepEM3D
  - 1fm: 40/255
  - 3fm: 128/255
  - 5fm: 50/255
  - Ensembled: 118/255
- Threshold used for MultiResUnet
  - 1fm: 45/255
  - 3fm: 78/255
  - 5fm: 24/255
  - Ensembled: 68/255
- Prediction time
  - 1fm: 4mins-5mins
  - 3fm/5fm: 10mins-13mins each

17. Results and Evaluation:

➢ Results - Model Predictions:

Test images are predicted on DeepEM3D-Net and MultiResUNet model. Below figures shows the results of the predicted outputs along with the ground truth.

- Prediction – 1fm:

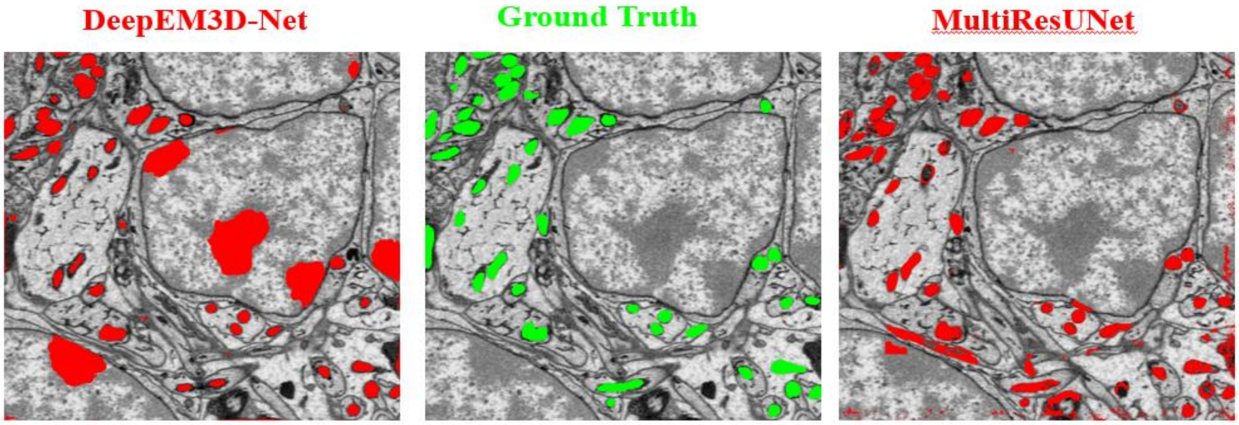


Figure-11: 1fm Prediction

- Prediction – 3fm:

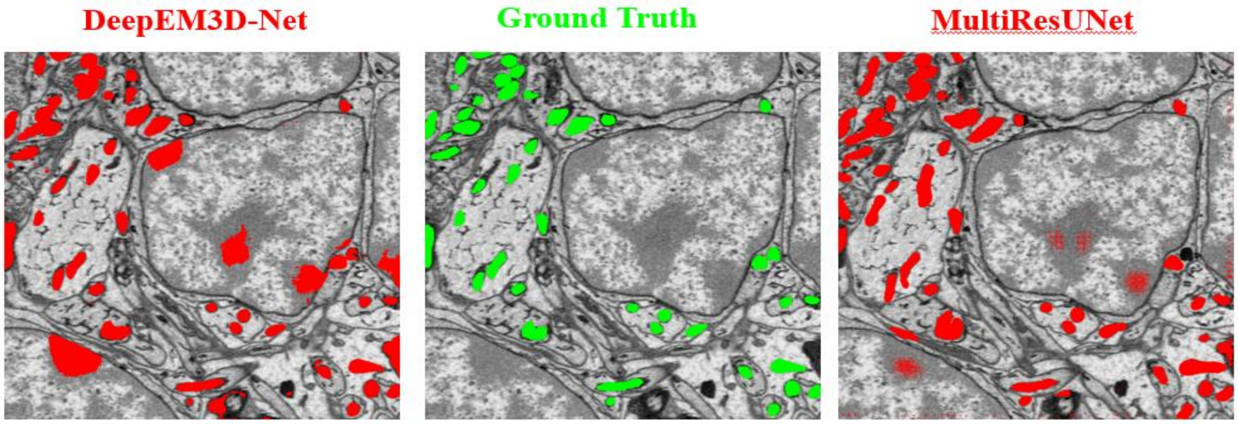


Figure-12: 3fm Prediction

- Prediction – 5fm:

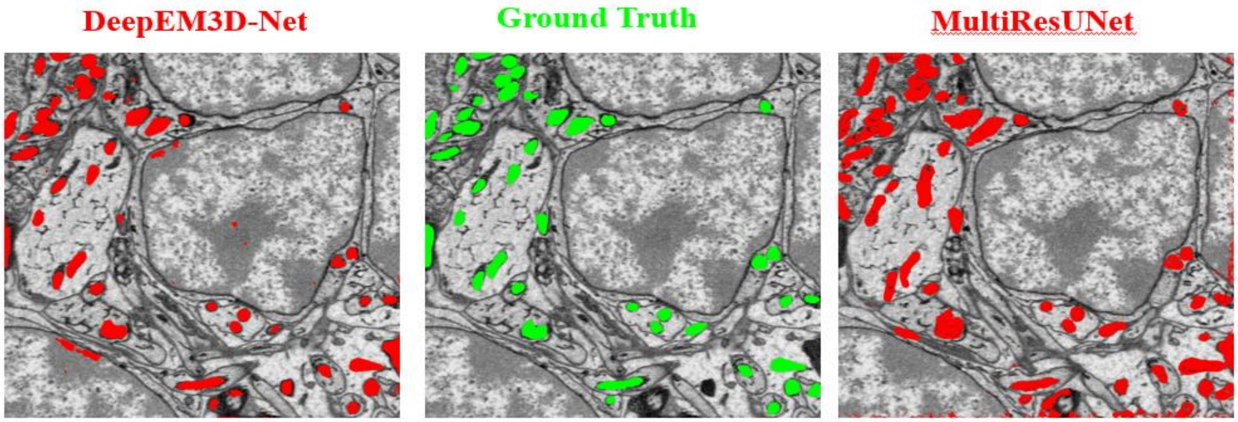


Figure-13: 5fm Prediction

- Prediction – ensembled:

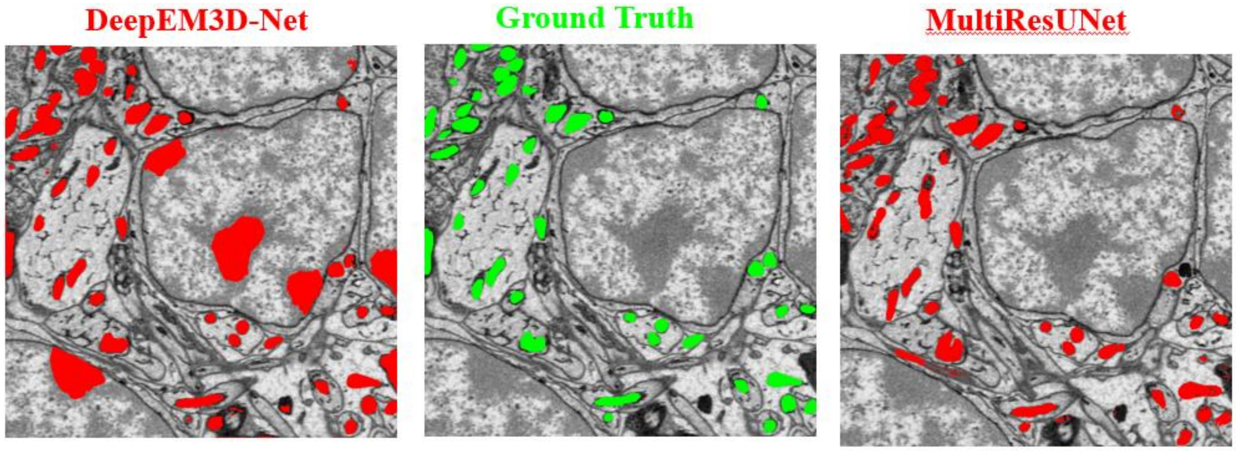


Figure-14: Ensemble of 1fm, 3fm and 5fm Prediction

➢ Evaluation:

For Accuracy, Dice similarity coefficient also known as F1 score is used. Below is the formula used to calculate the F1 Score. Confusion matrix is calculated using the predicted values and the true label values to obtain True Positive, Total positive predicted and Total positive present values.

$$\boldsymbol{F} = \frac{\mathbf{2}}{\frac{\mathbf{1}}{\mathbb{P}} + \frac{\mathbf{1}}{\boldsymbol{r}}}$$

- $\mathrm{P} = \mathrm{precision} = \frac{True\ Positive}{Total\ positive\ predicted}$
- $\mathrm{r} = \mathrm{recall} = \frac{True\ Positive}{Total\ Positive\ Present}$

➢

A line chart can be shown to compare the F-score of models for each image. In the line chart, each line will be representing one kind model (1fm, 3fm, 5fm and Ensemble) and various values at (x, y) will be showing model F1-score of value y for each image x as shown in Figure-15 for DeepEM3D-Net and Figure-16 for MultiResUNet. Figure-17 shows the comparison of ensemble prediction F1-Score between DeepEM3D-Net and MultiResUNet. From Figure-17 it can be seen that MultiResUNet model is performing better than DeepEM3D-Net.

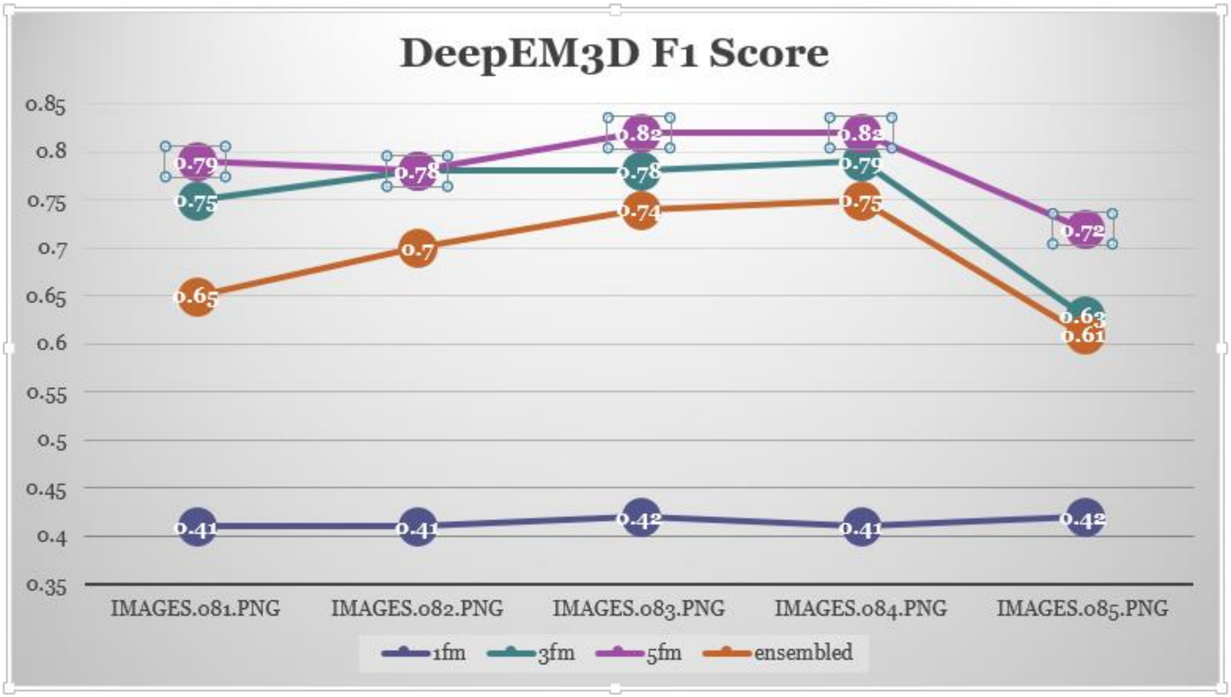


Figure-15: Line chart of F1-Score for 1fm, 3fm,5fm and ensemble prediction for DeepEM3D model

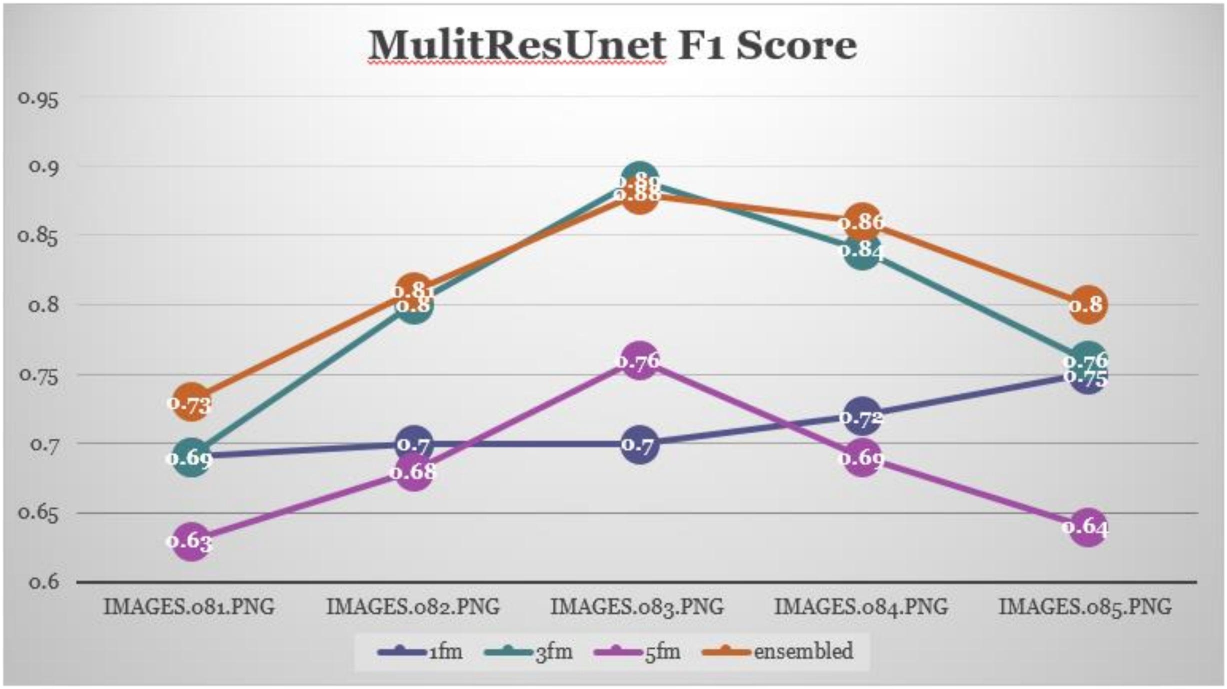


Figure-16: Line chart of F1-Score for 1fm, 3fm,5fm and ensemble prediction for MultiResUNet model

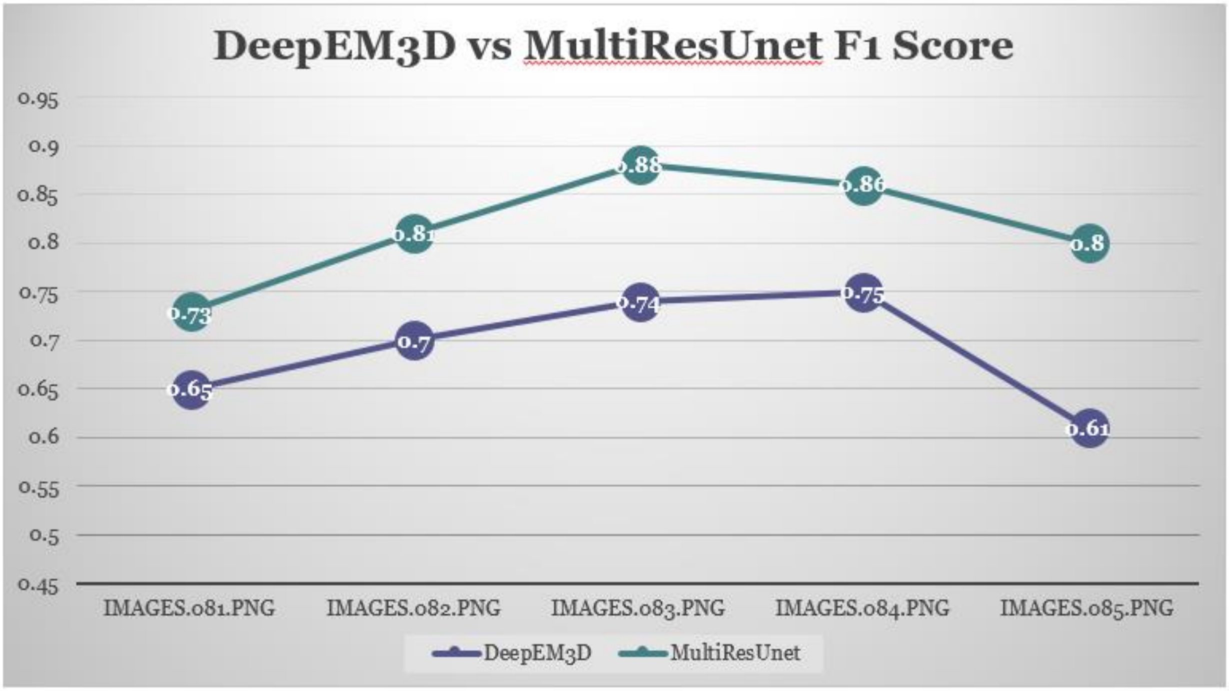


Figure-17: Comparison of Ensemble prediction F1-Score between DeepEM3D model and MultiResUNet model

18.Model Robustness:

Models are also tested for Robustness by adding noise. To test the robustness, 5% and 10% Salt-and-Pepper noise is added. Figure-18 shows the model results for 5% and 10% Salt-and-Pepper noise. Figure-19 shows Prediction output F1 Score comparison of 0%, 5% and 10% Salt-and-Pepper noise. From Figure-19 shows that with F1-Score reduced for subsequent noise in the image.

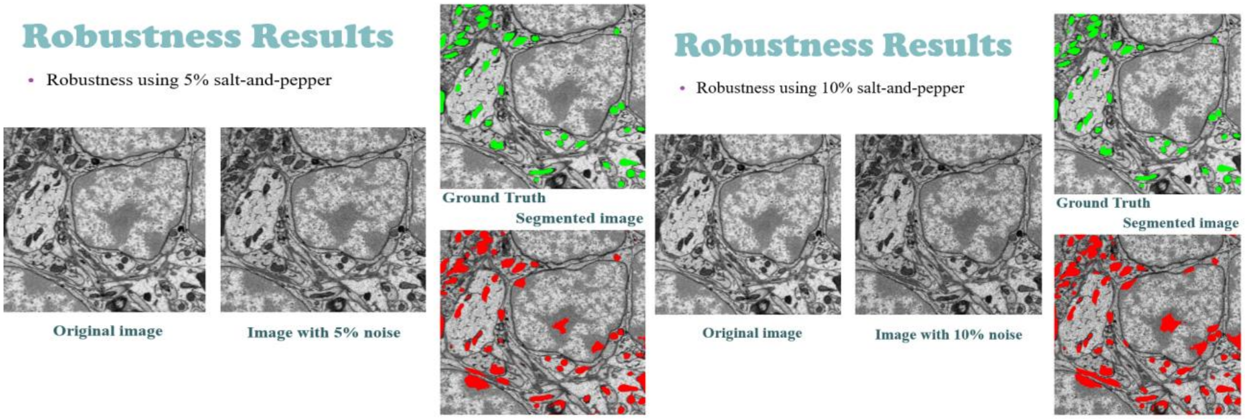


Figure-18: Model results for 5% and 10% Salt-and-Pepper noise

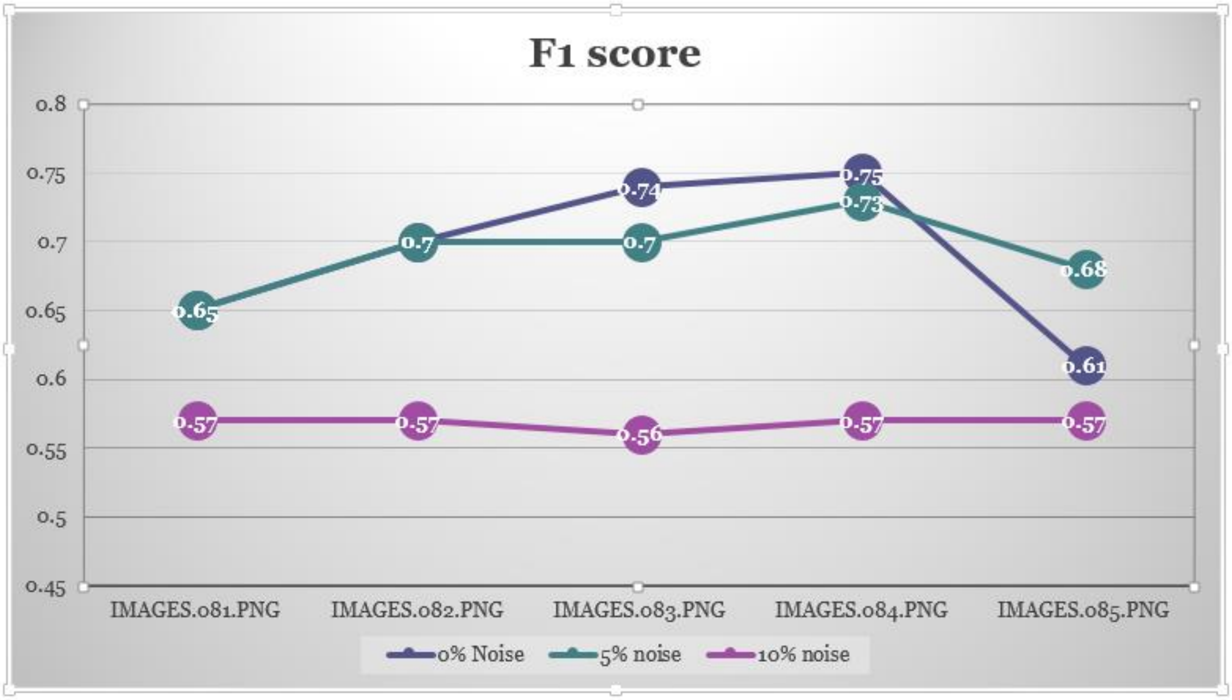


Figure-19: Prediction output F1 Score comparison of 0%, 5% and 10% Salt-and-Pepper noise

19.Mode Scalability:

Scalability of the model is checked by providing bigger images to the model. The system is designed in such a way that it can handle bigger images by cropping the images to 1024x1024 size for the model input and then later stitching back the predicted images to the original size. Experiments on Scalability shows that the model prediction runtime increased linearly with respect to the number of packages created from the original bigger image. Figure-20 shows the scalability results obtained by using images of size 1024x1024, 2048x2048 and 4096x4096.

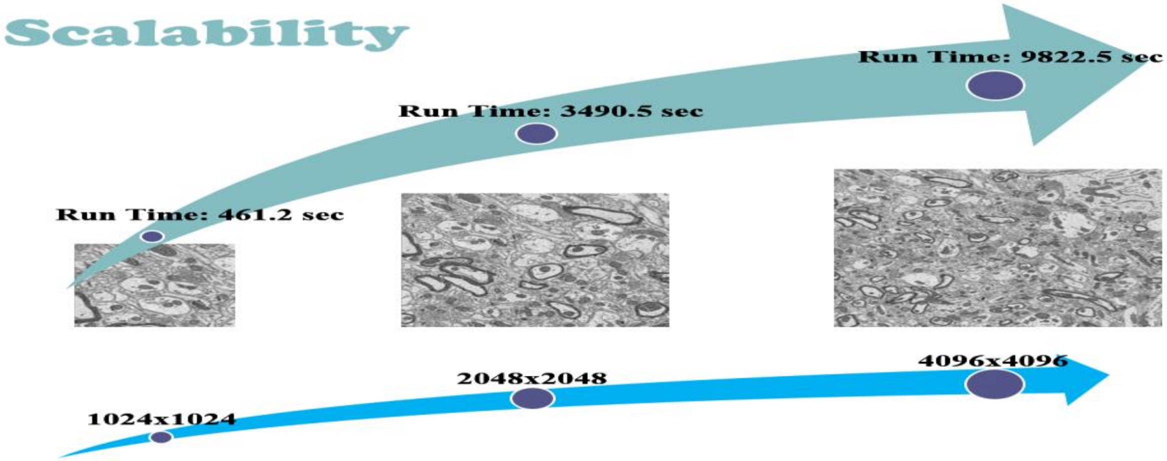


Figure-20: Image to show model scalability

20. Model Interpretation:

The result in terms of images obtained from model is called as predicted output. The predicted data has to be verified manually for its accuracy and significance. The model can be judged for its accuracy of segmenting objects such as mitochondria, nuclei as shown in Figure-21 below.

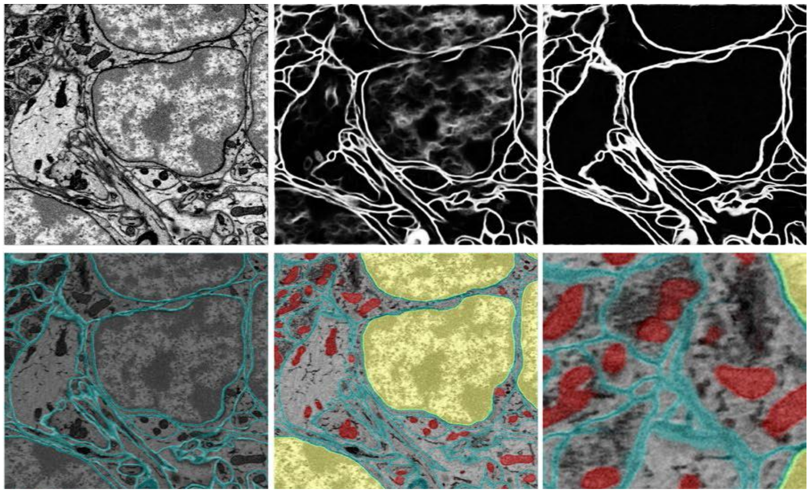


Figure-21: **Membranes Mitochondria Nuclei**

21. Deployment:

To make this solution architecture available for and used by end user, complete solution can be encapsulated in a Software. This deployment software can be made feature rich which can include offline mode, online mode and various formats to store output images.

An example software is presented in Figure-22 to showcase the idea. This software has mainly top level two feature:

i. Online mode
ii. Offline mode

i. **Online mode:** The online mode of this software does following tasks:
   a. Requirements: It requires high speed Internet connection to be able to send images, to be predicted, to servers
   b. User can upload image(s), folder containing images or h5 files
   c. To predict the uploaded images, a 'Predict' button is present in the software which tells software to send the uploaded image(s) to the server where actual prediction will take place
   d. Image(s) with object(s) identified is/are sent back to software after prediction
   e. Original image and image with object(s) highlighted are shown together in software panel

f. Predicted image can be saved in many available options: HDF5, JPEG, PNG, RAW, TIFF and PDF

g. Zoom in and Zoom out options are also provided in the software to do analysis in detail

ii. **Offline mode:** The offline mode of this software does following tasks:

a. Requirements: To run this software in offline mode, it requires a user to meet lot more requirements such as having a computer with at least 1 GPU, Python3 with Keras with tensorflow backend running on it

b. Software is shipped with inbuilt trained model which is going to segment an image in offline mode on user's system

c. User can upload image(s), folder containing images or h5 files

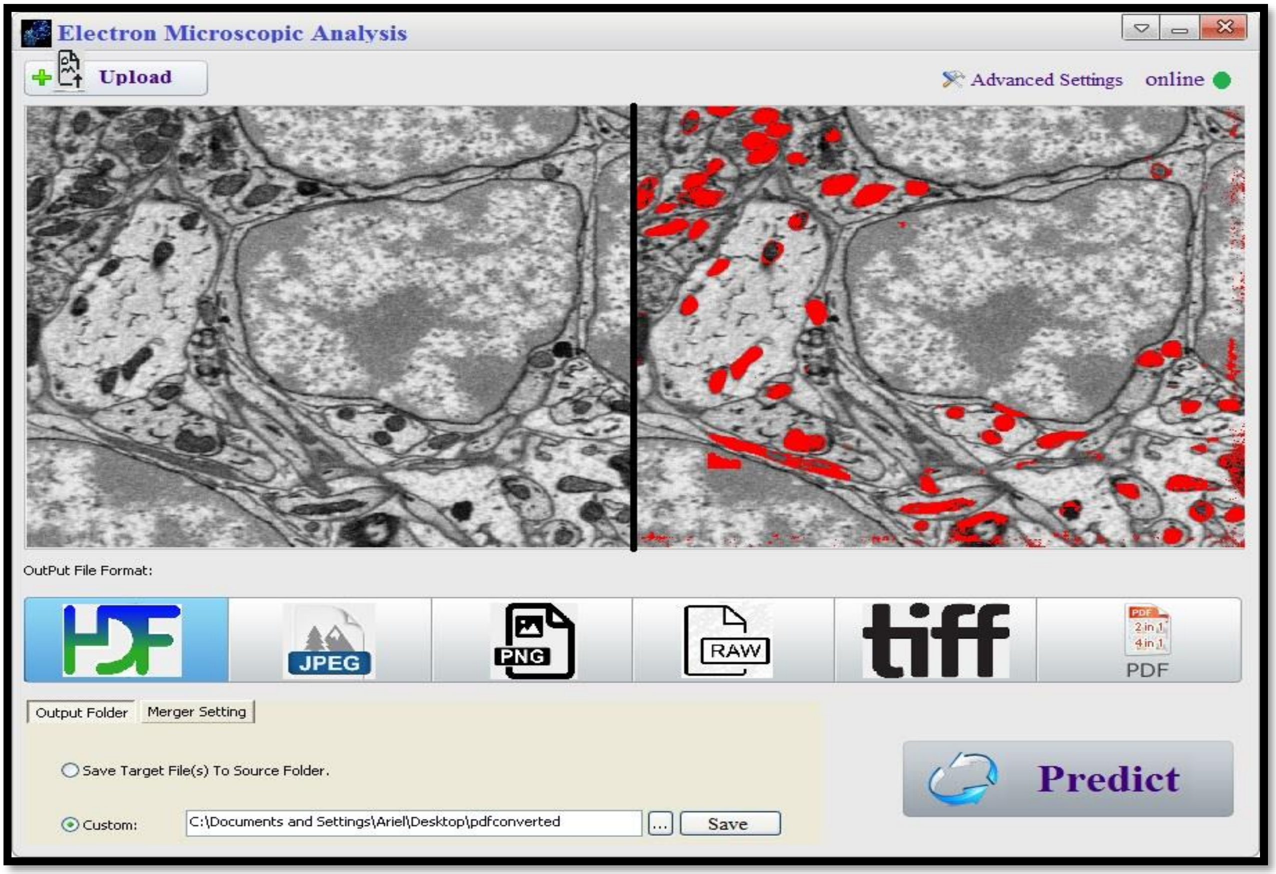
d. User can press 'Predict' button and select single/multiple object(s) to be predicted

e. Model is run in user's system with image as input

f. Predicted image with object(s) identified gets generated

g. Original image and image with object(s) highlighted are shown together in software panel

h. Predicted image can be saved in many available options: HDF5, JPEG, PNG, RAW, TIFF and PDF



○ Save Target File(s) To Source Folder.

⦿ Custom: C:\Documents and Settings\Ariel\Desktop\pdfconverted

Figure-22: Sample of Models Accuracy comparison chart at various Iterations

22.Conclusion:

The main goal of the project is image segmentation to segment membrane, mitochondria and nuclei of the brain image samples. After doing exploratory data analysis on the raw image samples, a hypothesis was made and a deep learning-based Convolution Neural Network (CNN) was chosen as solution architecture to properly segment the non-isotropic images.

Based on the data engineering performed on raw data, a data pipeline is laid out for the CNN model. The performance of the model is tuned using a validation dataset to generate a final model. This final trained model can be used by NCMIR (National Centre for Microscopy and Imaging research) professional, scientists, researcher or anyone for the purpose of segmenting the images. The model can also be trained different kinds of biomedical datasets.

23.Team Roles and Responsibilities:

Project coordinator/manager: Tushar Singhal
Budget manager: Prashant Kolkur
Record keeper: Prashant Kolkur

24.References

- Zeng T, Wu B, Ji S. DeepEM3D: approaching human-level performance on 3D anisotropic EM image segmentation. Bioinformatics. 2017;33(16):2555–2562. doi:10.1093/bioinformatics/btx188
- Ibtehaz, Nabil, and M. Sohel Rahman. "MultiResUNet: Rethinking the U-Net Architecture for Multimodal Biomedical Image Segmentation." arXiv preprint arXiv:1902.04049 (2019)
- Keras Documentation
- Dataflow reference code for 3D models: https://github.com/ellisdg/3DUnetCNN
- MultiResUnet model reference code: https://github.com/nibtehaz/MultiResUNet
- DeepEM3D model reference code: https://github.com/divelab/deepem3d
- Unet model reference code: https://github.com/zhixuhao/unet
- DeepEM3D Matlab model reference code: https://github.com/CRBS/cdeep3m

25.Appendices

Many of the DSE MAS courses knowledge were used in this project. Those projects include, but not limited to Python for Data Analysis, Machine Learning, Big Data, and Data Visualization. DSE MAS knowledge enabled us to handle the large data (big images) and to process this data in time and space efficient way. Machine learning helped us in understanding the model in detail and finding the optimal number of hidden layers to solve the problem.
The code for reproducibility can be found at https://doi.org/10.6075/J03N21QH.