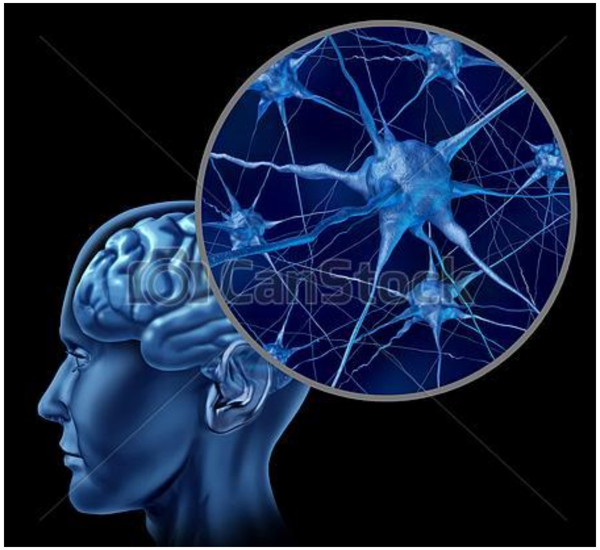
Electron Microscopic Data Analysis

Prashant Kolkur
Tushar Singhal

Advisors:

Prof Mark Ellisman
Prof Ilkay Altintas

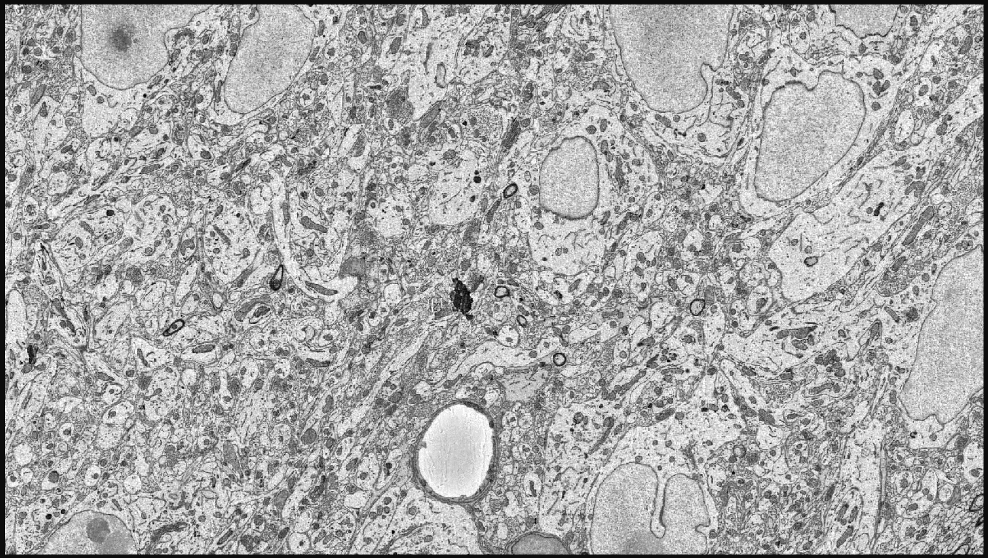


Motivation

- Alzheimer's disease is a type of dementia that causes problems with memory, thinking and behavior
- 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 Data is available that needs to be segmented to find various components of brain such as Mitochondria, membrane, Nuclei in the brain image samples
- Given the large number of image samples and the size of an image, manual image segmentation is not practical
- **So what is the problem in doing segmentation..**

Problem

SBEM Datasets are Large-Scale, High Resolution 3D Image Stacks



- Large datasets (typically ca. 1 – 6 TB)
- Each SBEM microscope acquisition speeds *~300 GB / day (developments ongoing to keep increasing speed and resolution)*
- Can be kept running up to month/s for one dataset

Perez *et al.*, 2014

How do we get from image to segmentation?

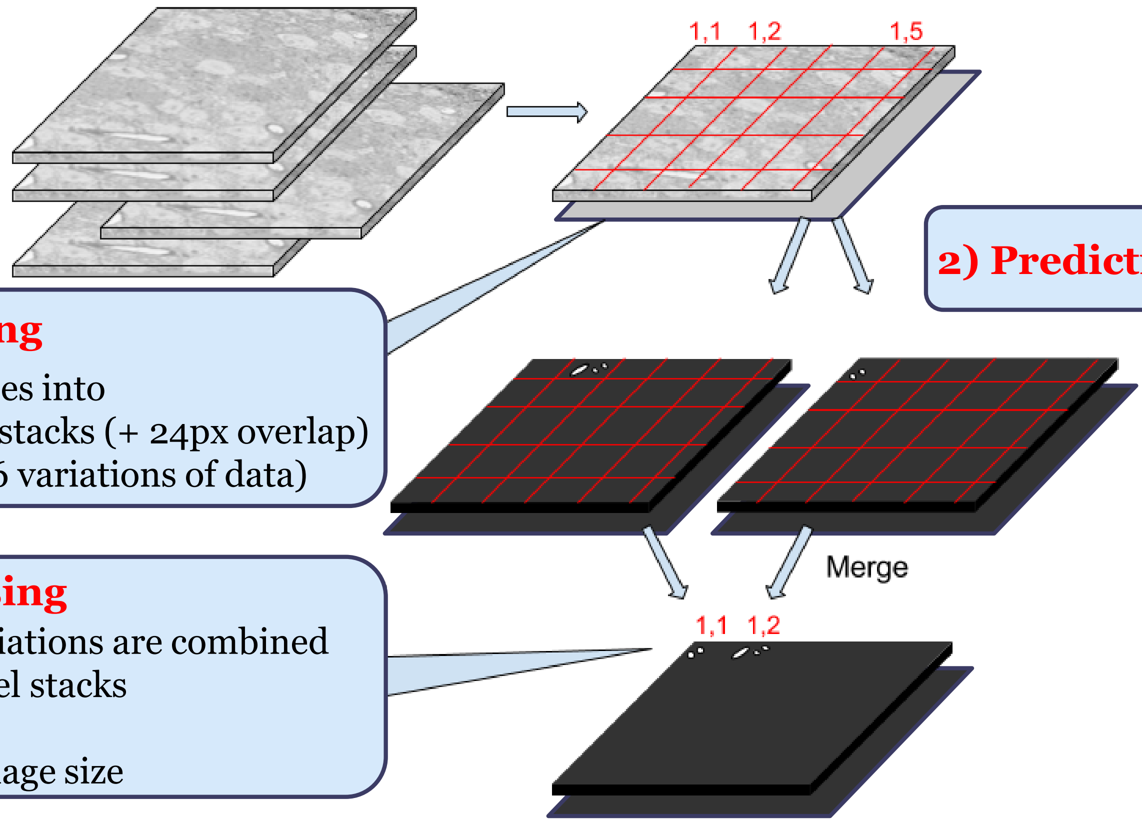
= Computer vision task → Use deep convolutional neural networks to identify cell structures

But: images noisy, fluctuations in intensity levels, no 'typical acquisition parameters' across datasets, different structures look similar (larger context important), training data extremely time consuming to produce, large datasets but training data is small.

Also: misalignments across z occur, image distortions, data highly anisotropic (lower resolution in z than x/y)

Processing large data in CDeep3M

CDeep3M creates image stacks that can be processed on single GPUs (limited by vRAM)



2) Prediction

1) Pre-processing

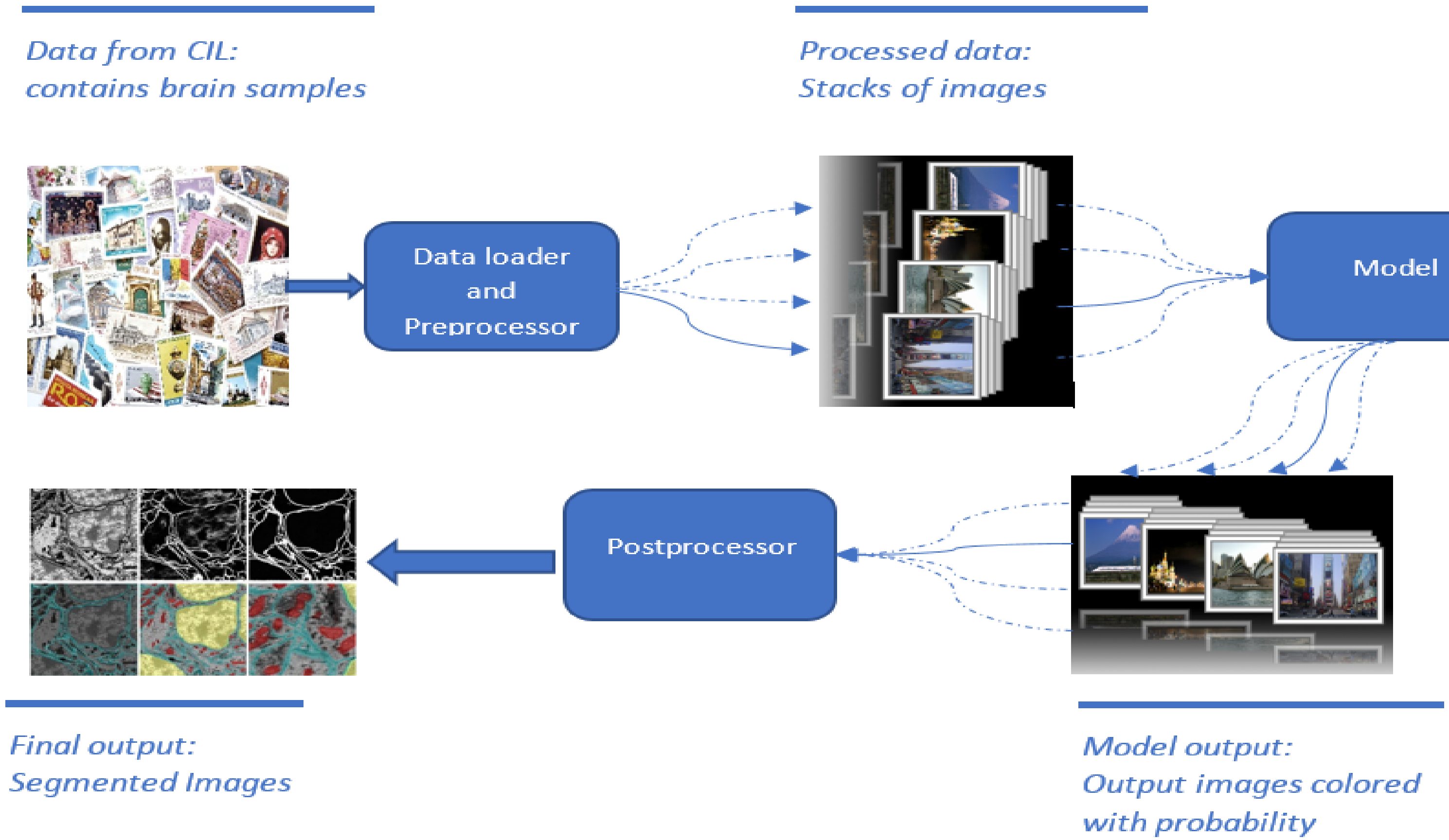
CDeep3M breaks images into
1000*1000*100 voxel stacks (+ 24px overlap)
Data-augmentation (16 variations of data)

3) Post-processing

predictions of 16 variations are combined
1000*1000*100 voxel stacks
are merged
back into original image size

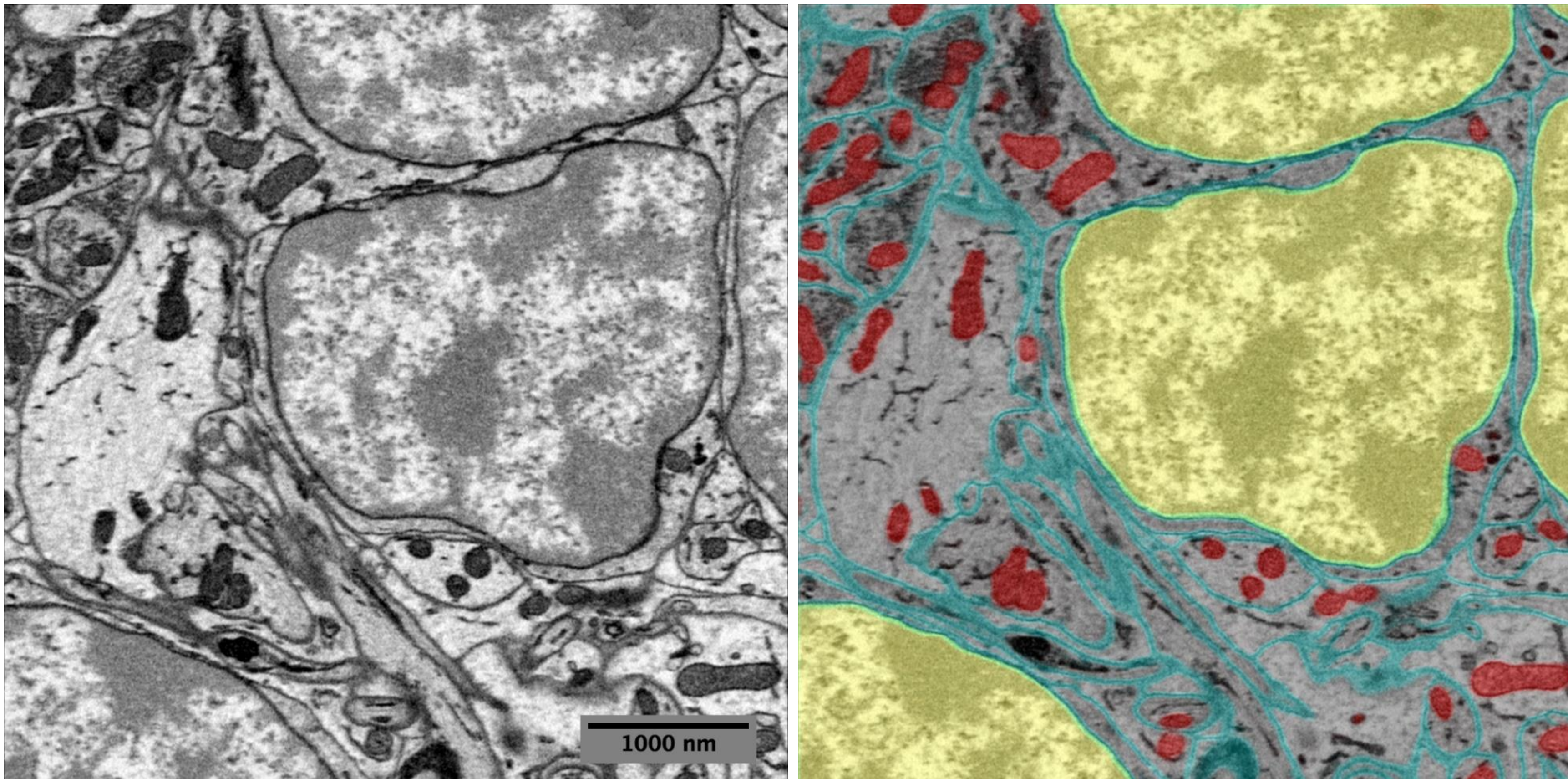
Development

Introduction of CDeep3M



Applying CDeep3M: examples

Serial block-face scanning electron microscopy (SBEM)

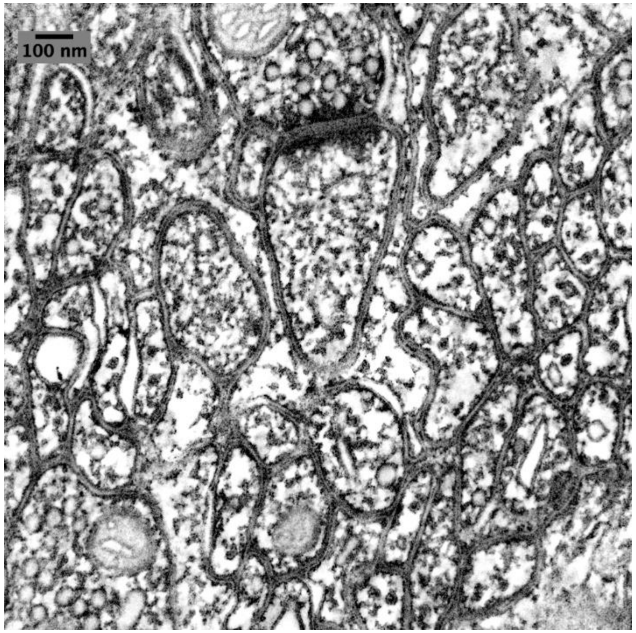


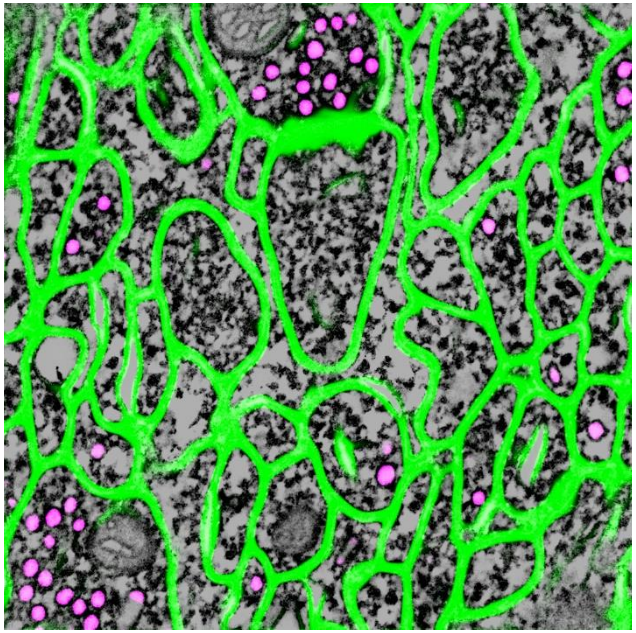
Haberl *et al.*, 2018

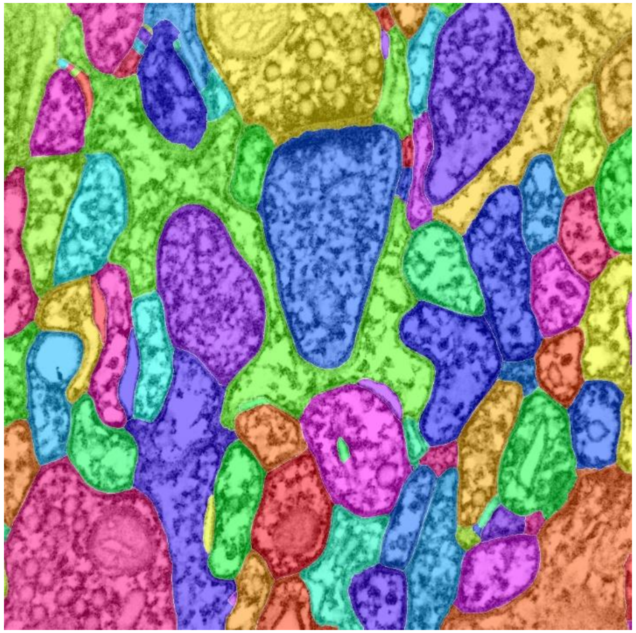
Membranes Mitochondria Nuclei

Applying CDeep3M: examples

Multi tilt electron tomography







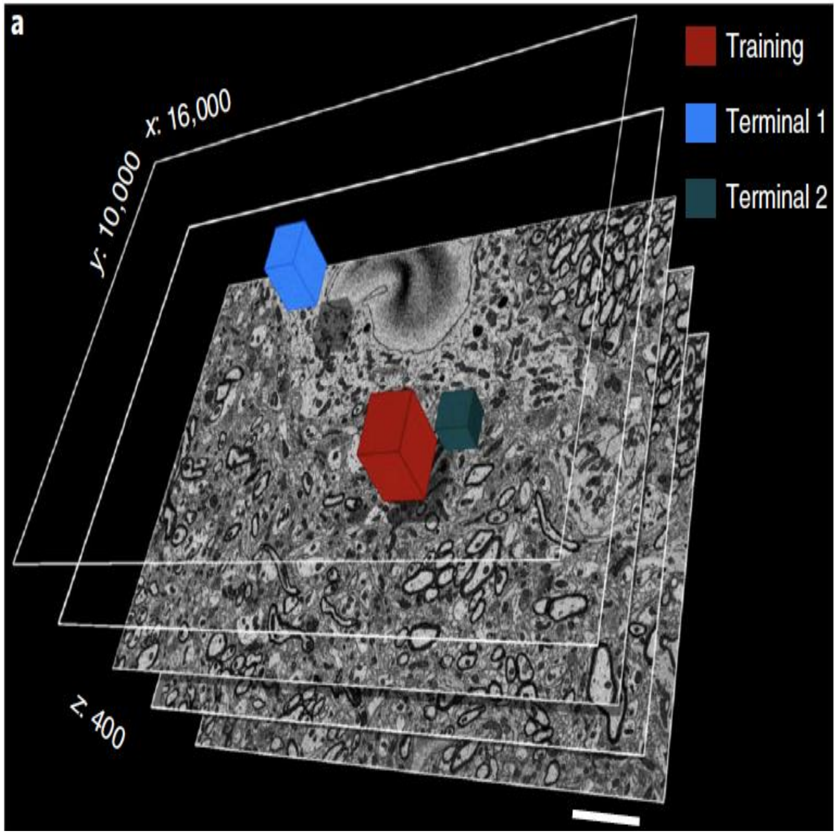
Membranes **Vesicles**

Cells

Haberl *et al.*, 2018

Data

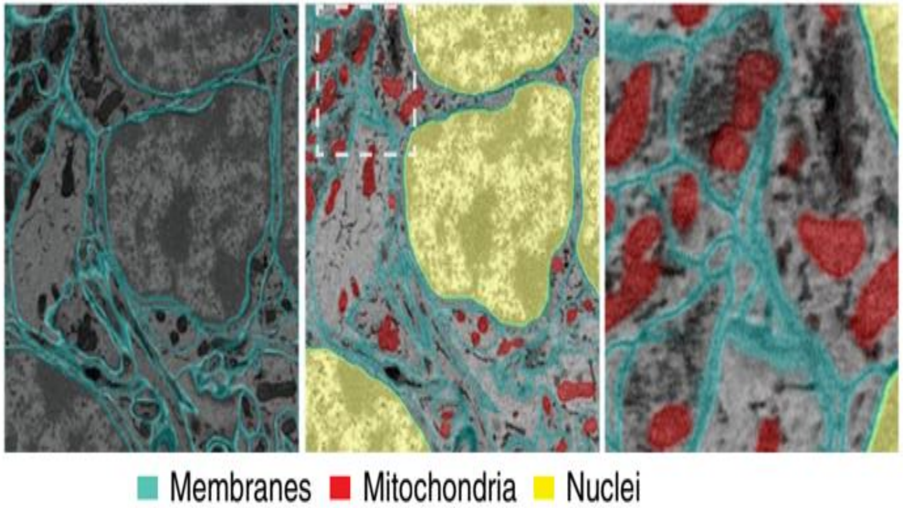
- Data source is a serial block face scanning electron microscopy (SBEM) – an image data of brain samples of patients
- Raw data (SBEM) is collected from Cell Image Library (CIL)
- This image data is the multiple datasets in the range of 16000x10000x400 images in PNG format



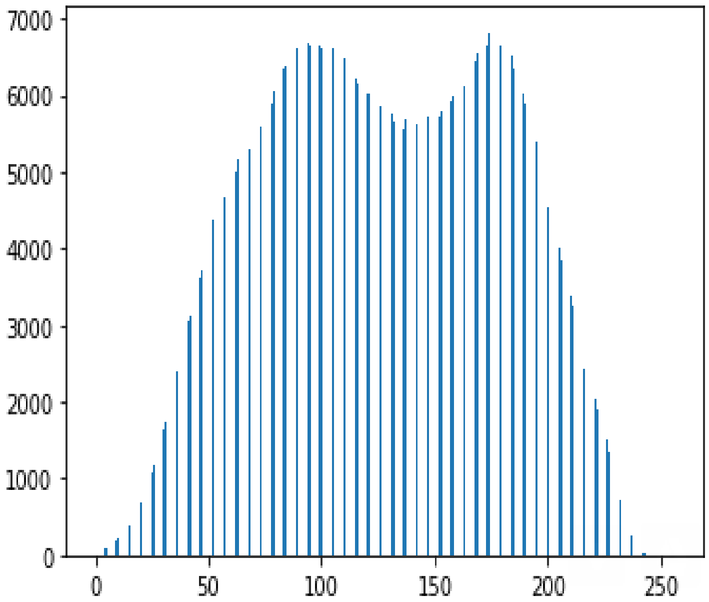
Data Exploration

Objects of interest in the image

- Mitochondria
- Membranes
- Nuclei



Histogram of Image



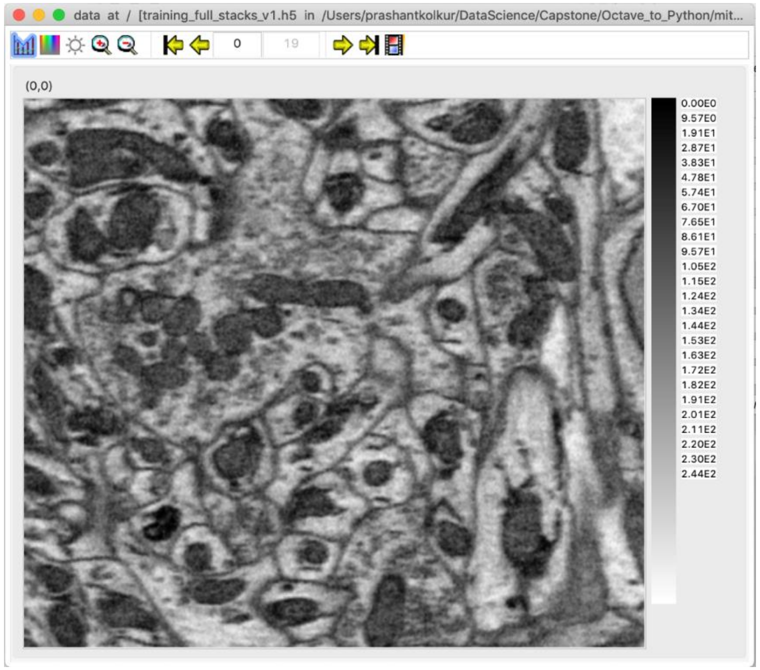
Data Exploration

- Sub-blocks of original images are taken as data
- Images are of size 1024x1024 pixels
- Images are not similar
  - Some images have more objects than others
- Images are non isotropic
  - Some parts of an image have more objects when compared to other parts
- Images are clean

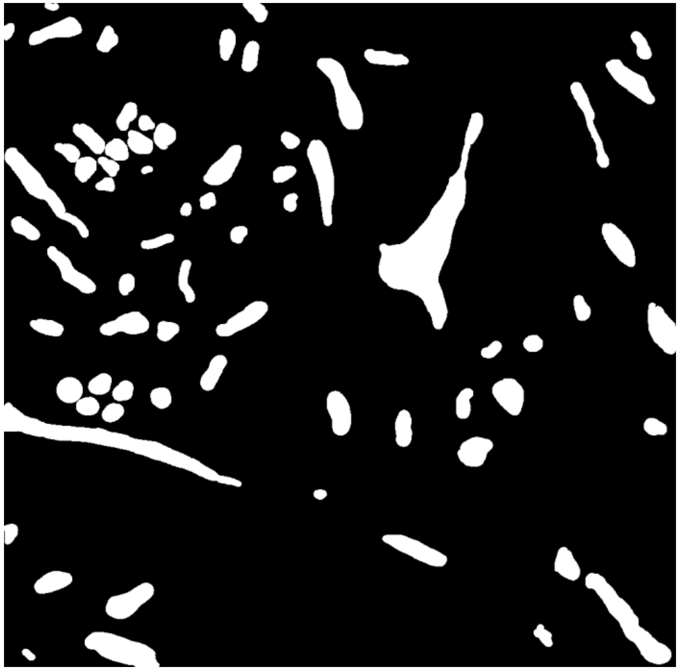
Dataset name	folders	Sub-folders	destination	Data size
mito_testsample	training	images	Preprocess block	80 images (75 MB)
		labels		80 images (770 KB)
	validation	images	Preprocess block	15 images (15 MB)
		labels		15 images (100 KB)
	testset	-	Model block	5 images (5 MB)

Data Visualization

- Data (images) are 3-D – Continuous images in Z-direction
- Software to visualize images: HDFView, h5pyviewer
- An image is shown using HDFview 3.0



Image



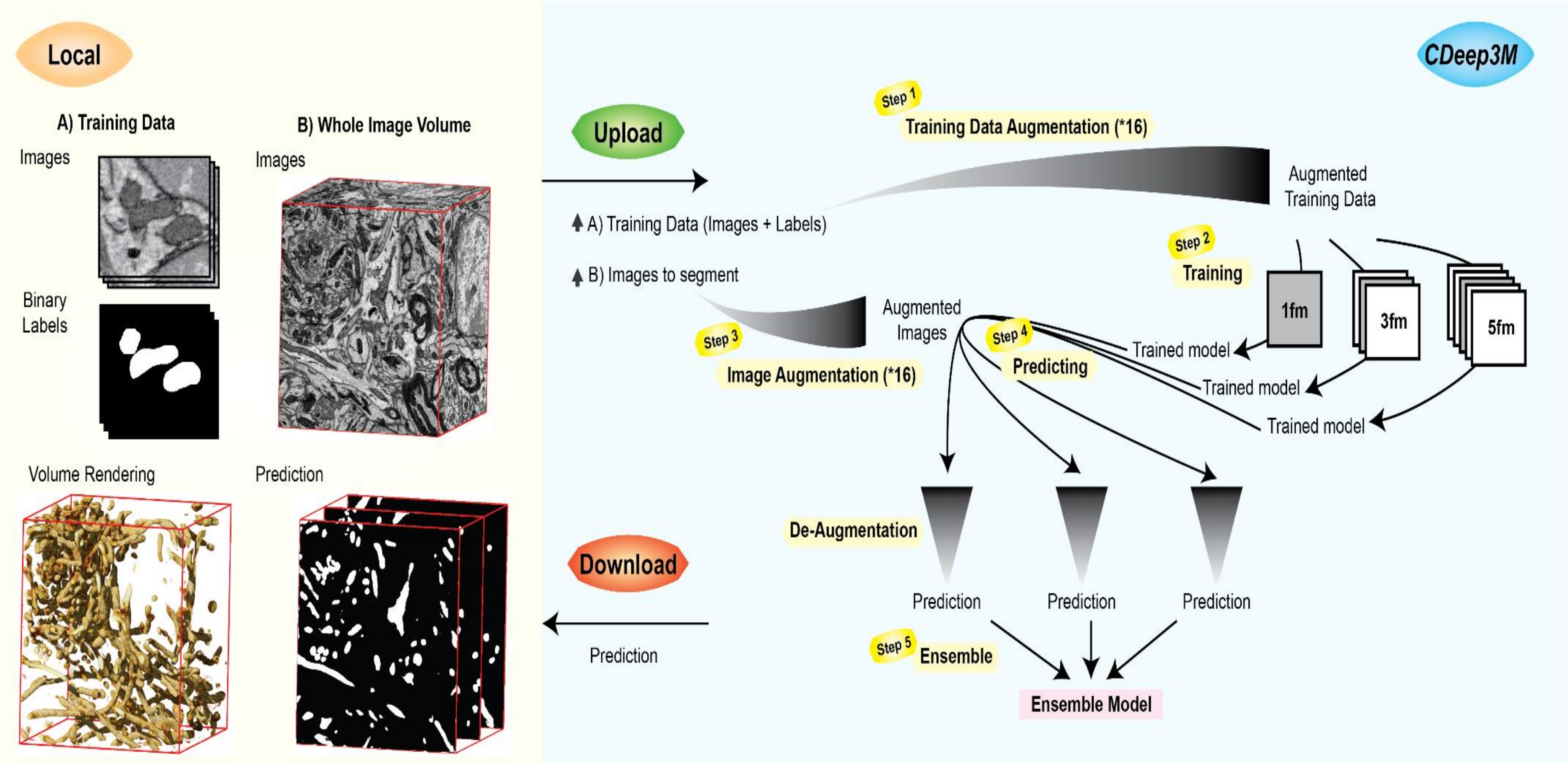
Label

Data Pipeline: Learning & Prediction

nature methods BRIEF COMMUNICATION
https://doi.org/10.1038/s41592-018-0106-z

CDeep3M—Plug-and-Play cloud-based deep learning for image segmentation

Matthias G. Haberl[1,2*], Christopher Churas[2], Lucas Tindall[1], Daniela Boassa[1], Sébastien Phan[1,2], Eric A. Bushong[1], Matthew Madany[1], Raffi Akay[1], Thomas J. Deerinck[1], Steven T. Peltier[1,2] and Mark H. Ellisman[1,2*]



Parallel Processing

- Producer consumer scheme to enhance throughput for large scale image segmentation:
  1) Pre-processing → Step1
     - extract subarea (size 1024*1024*100 with overlap between neighboring areas, and mirroring at end of image)
     - Image augmentation (16 variations)
  2) Prediction (uses number of GPUs available) → Step2
  3) Post-processing → Step3
     - De-augmentation (combine 16 variations)
     - Ensemble (1fm, 3fm, 5fm)

- Using CPUs for pre- and post-processing, GPUs for prediction (in parallel)

Package 1: Step 1 (CPUs) -> Step 2 (GPU/s) -> Step 3 (CPUs)
Package 2: Step 1 (CPUs) -> Step 2 (GPU/s) -> Step 3 (CPUs)
Package 3: Step 1 (CPUs) -> Step 2 (GPU/s) -> Step 3 (CPUs)

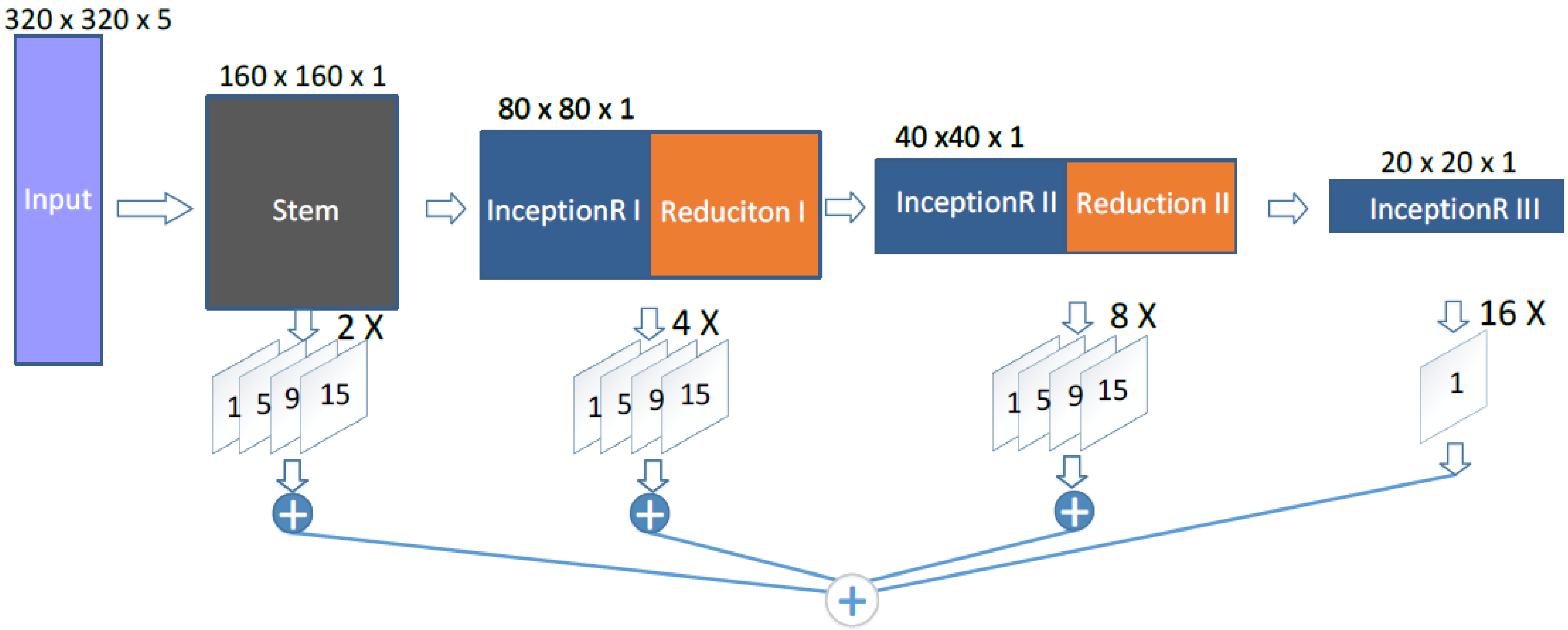
Data Preprocessing

- Samples are stored in a folder
- Preprocessor loads these images and makes a stack of processed image
  - Files are written out in memory in h5 format
- Data cleaning
  - Images are being padded with zero to meet the minimum size requirement
  - Data is checked for no-binary
  - Label data is checked for binary
- Data Wrangling
  - Augmentation by rotation: 0, 90, 180, 360 and flipping

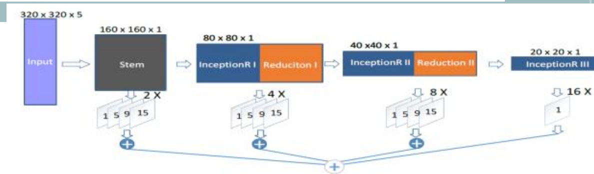
Processed dataset name	Input dataset name	Data size
augmentedtraining	Mito_testsample/training	10.26 GB
validation	Mito_testsample/validation	161 MB

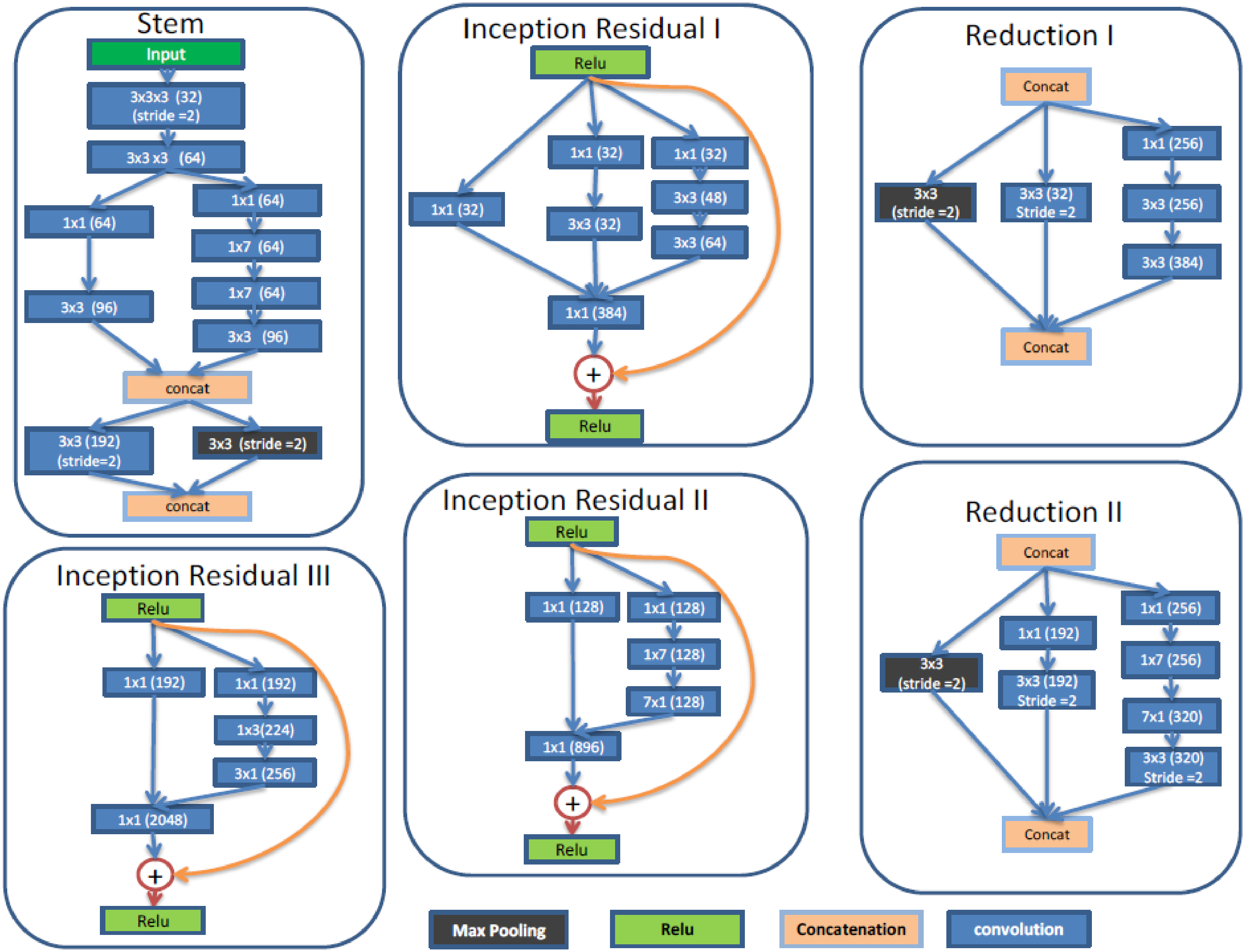
Model(1) : DeepEM3D

- The CNN model is based on the paper DeepEM3D-Net model: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6248556/
- The basic architecture of the model is shown



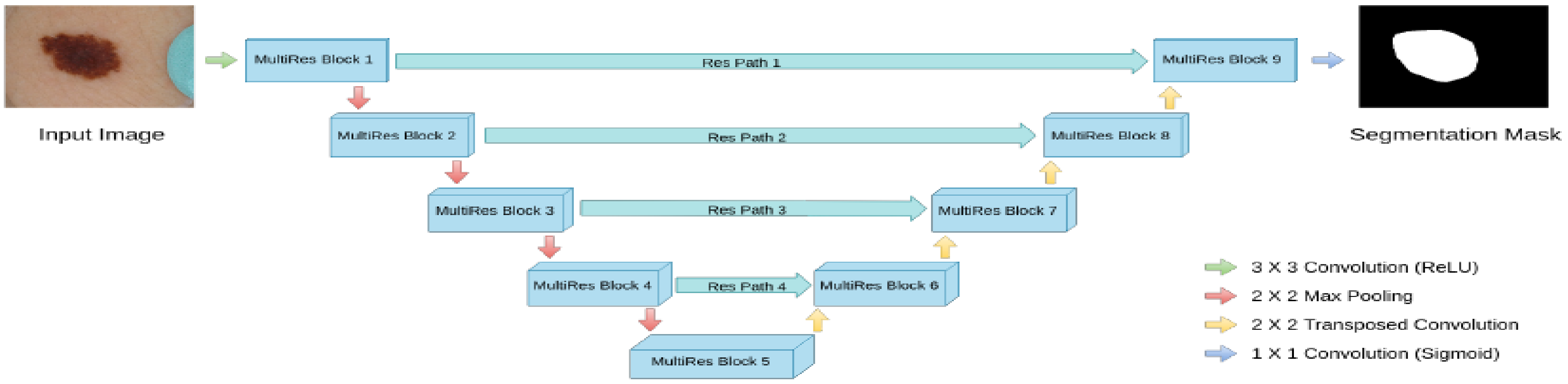
Model(1) : DeepEM3D

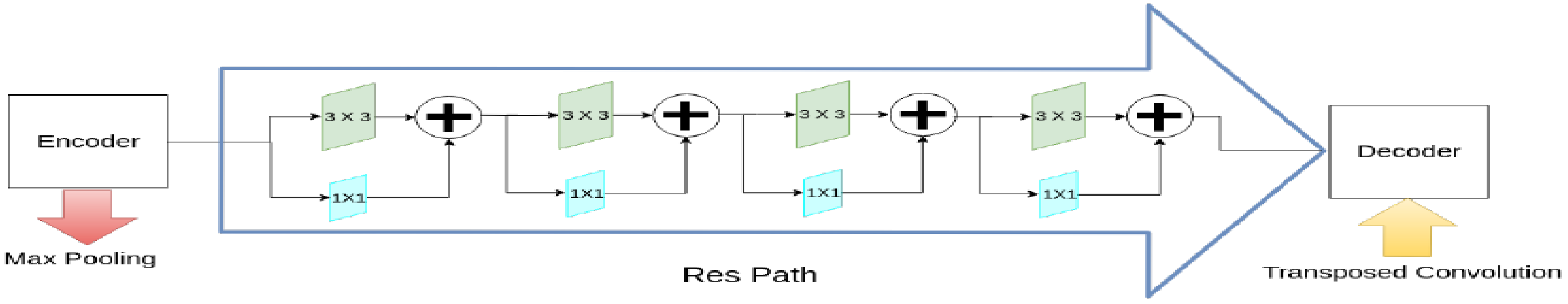




Model(2) : MultiResUnet

- The CNN MultiResUnet model is based on the paper: https://arxiv.org/abs/1902.04049
- The basic architecture of model is shown below





Model Training

- 1fm - 2D model and 3fm/5fm - 3D model
  - Used 4 Tesla P100 GPUs with 12GB per instance
  - DeepEM3D Model
    - written in Caffe
  - MultiResUnet Model
    - written in Python using 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

Model Prediction

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

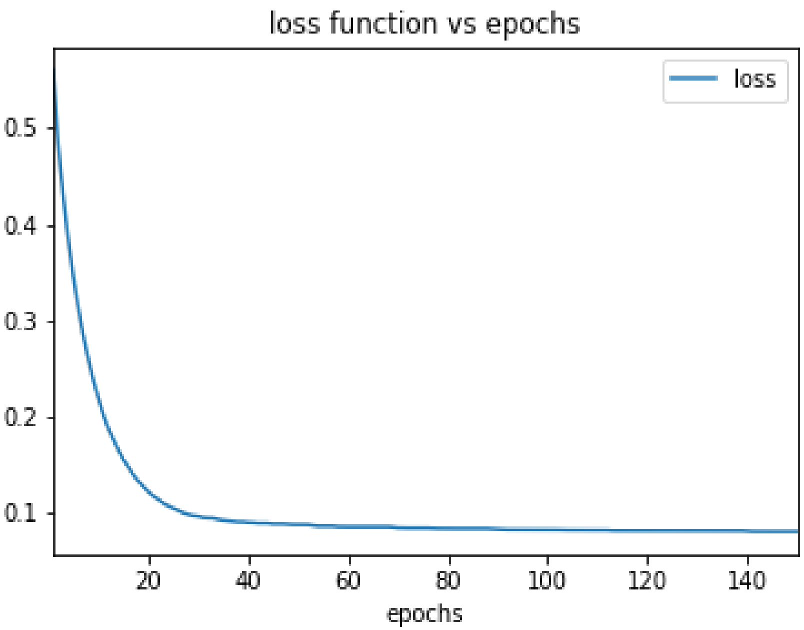
Metric for Evaluation

- For Accuracy, Dice similarity coefficient or F_1 score is used:

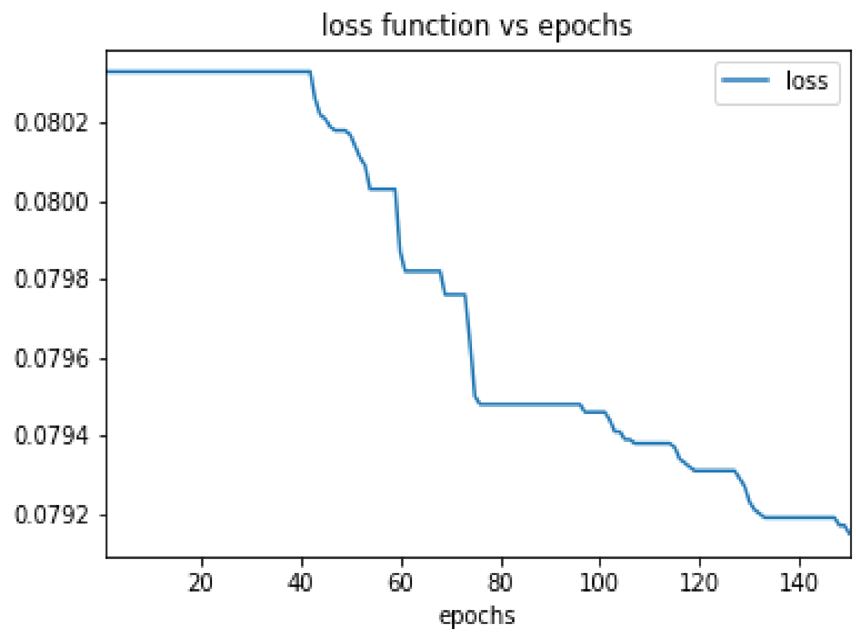
$$\boldsymbol{F} = \frac{\mathbf{2}}{\frac{\mathbf{1}}{\mathbb{P}} + \frac{\mathbf{1}}{\boldsymbol{r}}}$$

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

Loss function for training – 1fm

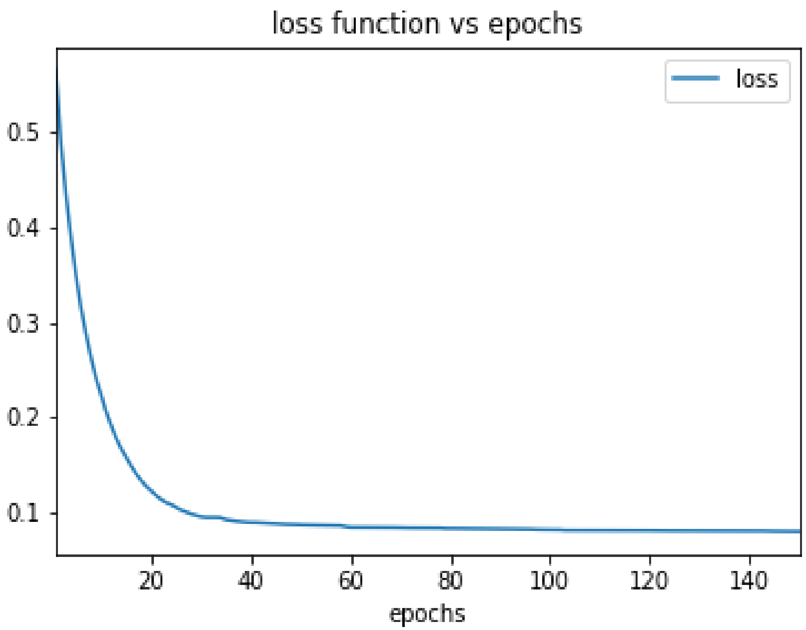


Original Data

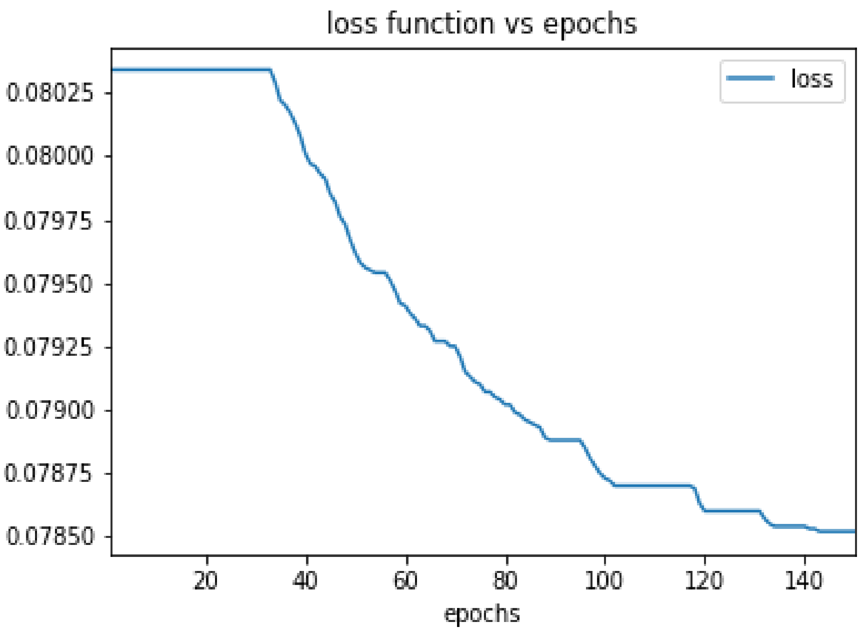


Augmented Data

Loss function for training – 3fm

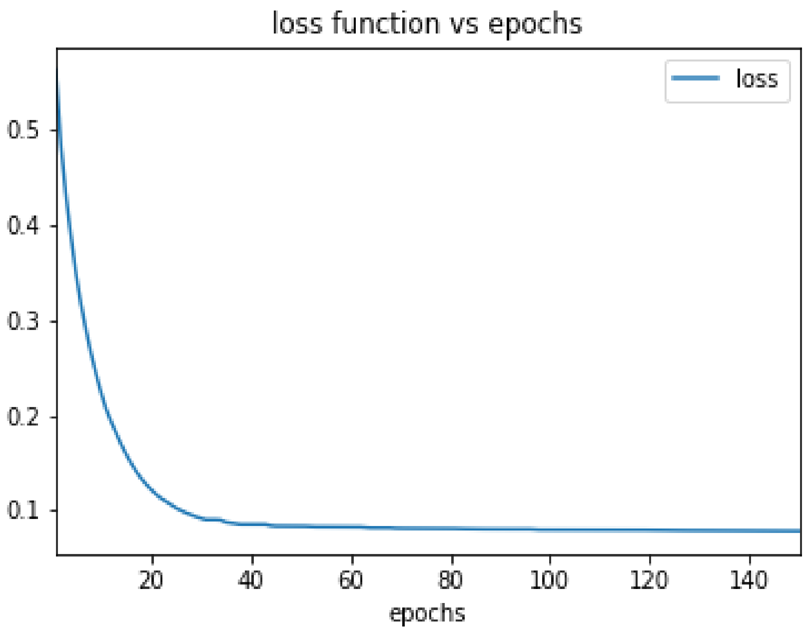


Original Data

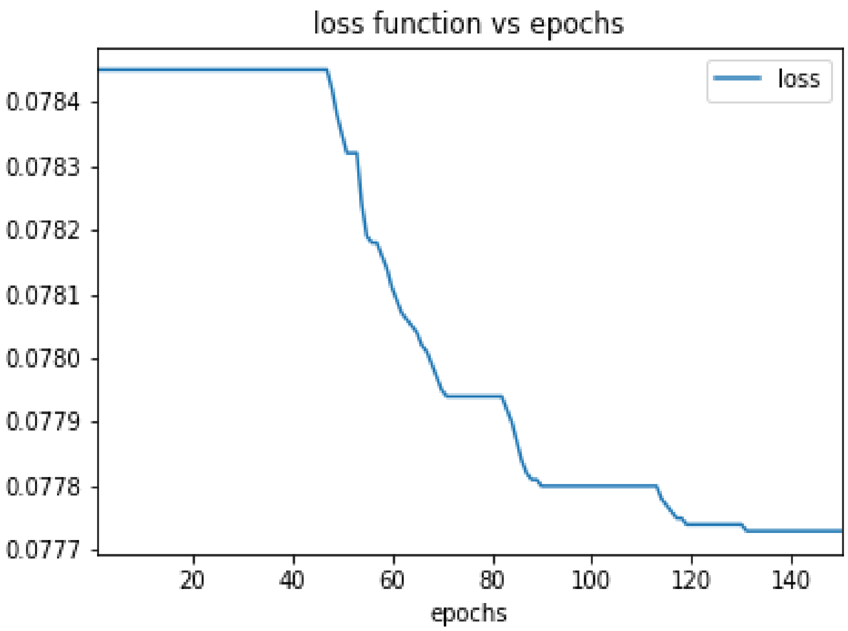


Augmented Data

Loss function for training – 5fm



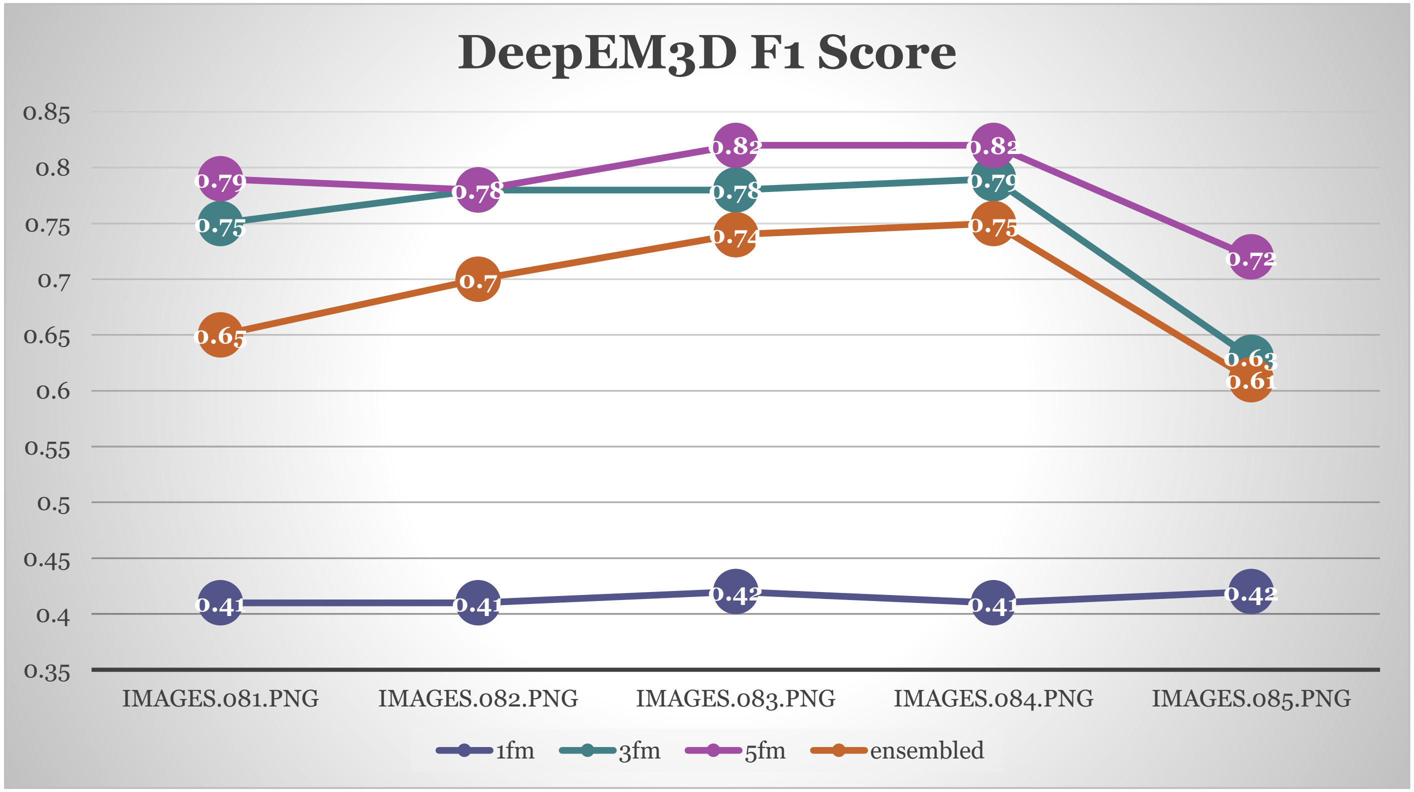
Original Data



Augmented Data

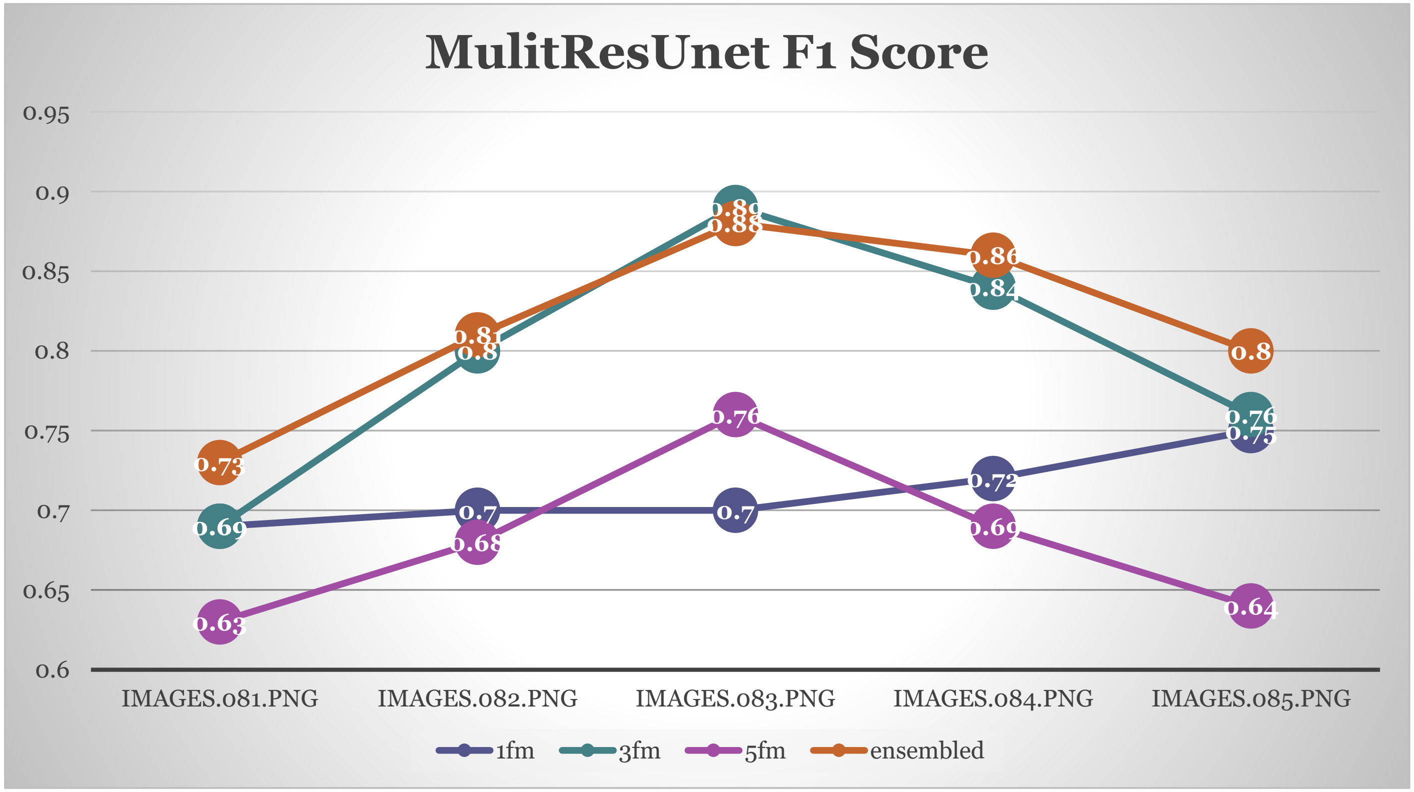
Results: DeepEM3D

- DeepEM3D results using 1fm, 3fm, 5fm and ensembled (merged)



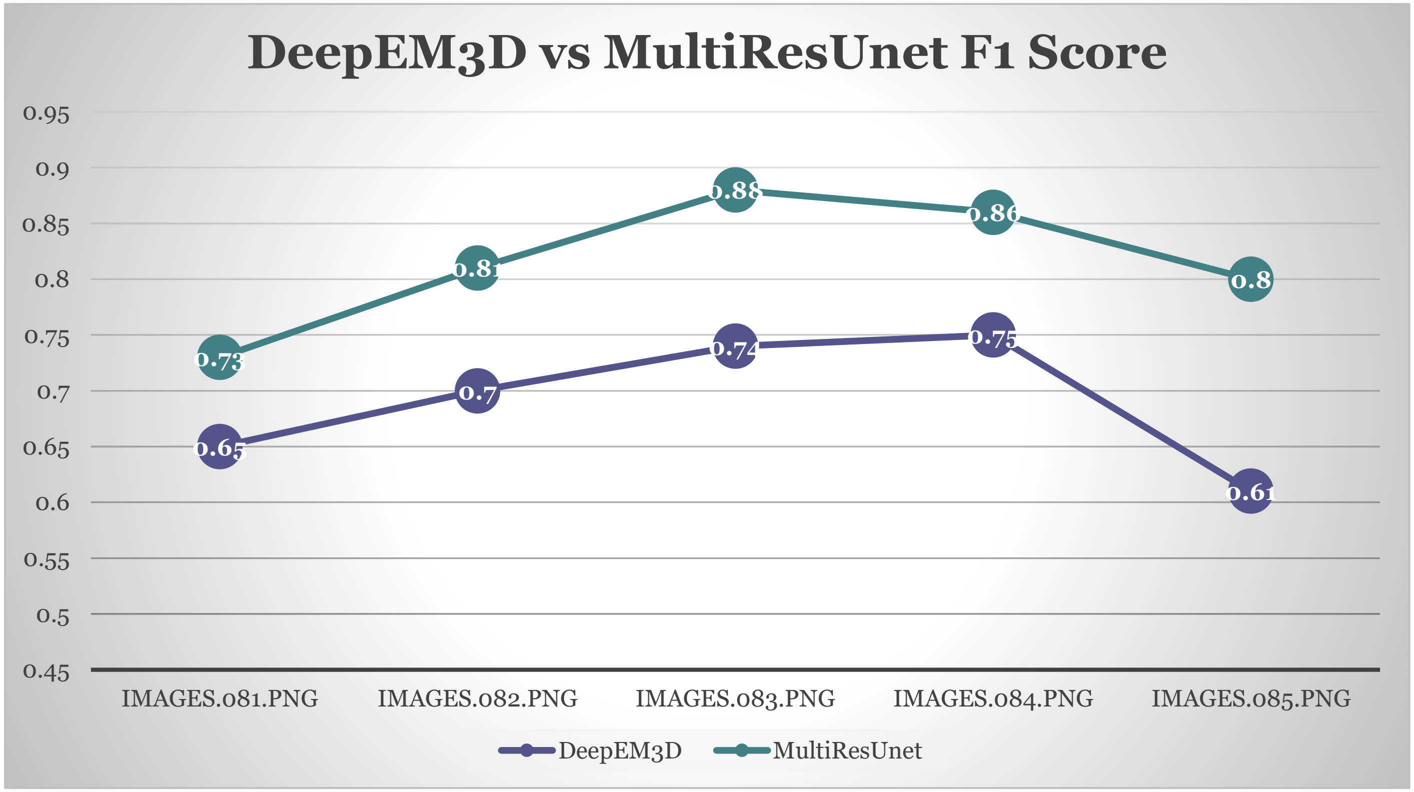
Results: MultiResUnet

- MultiResUnet results using 1fm, 3fm, 5fm and ensembled (merged)

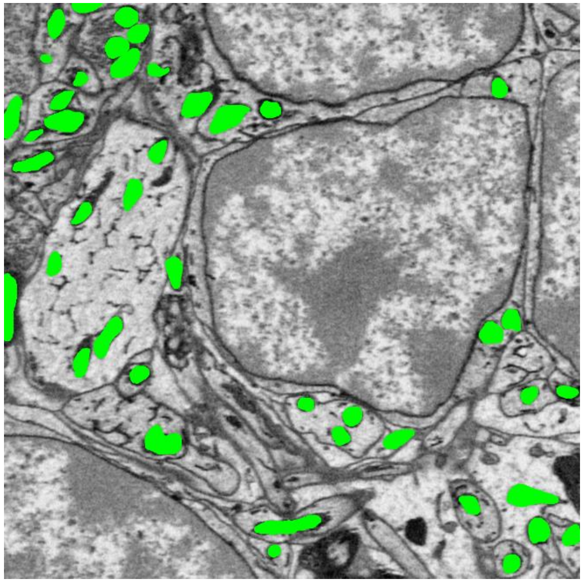


DeepEM3D vs MultiResUnet

- DeepEM3D and MultiResUnet results using ensembled

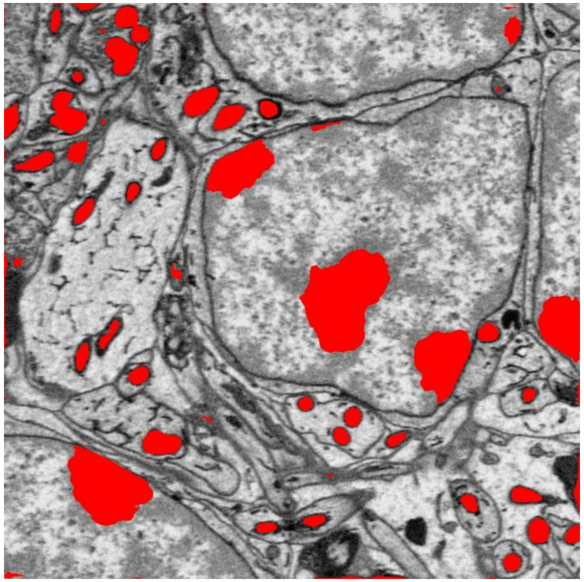


Prediction – 1fm

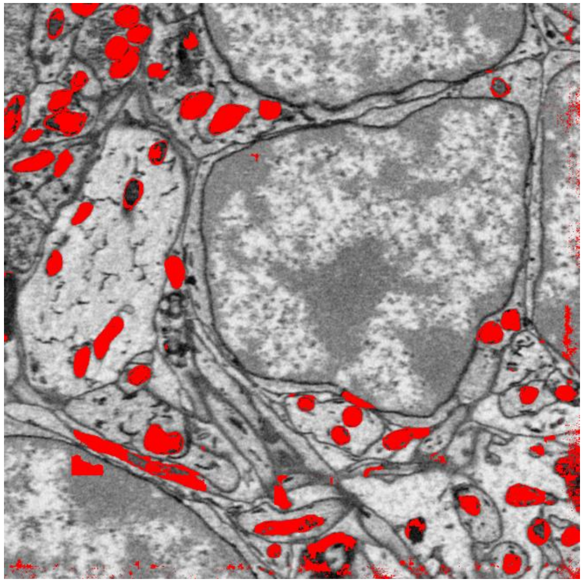


Ground Truth

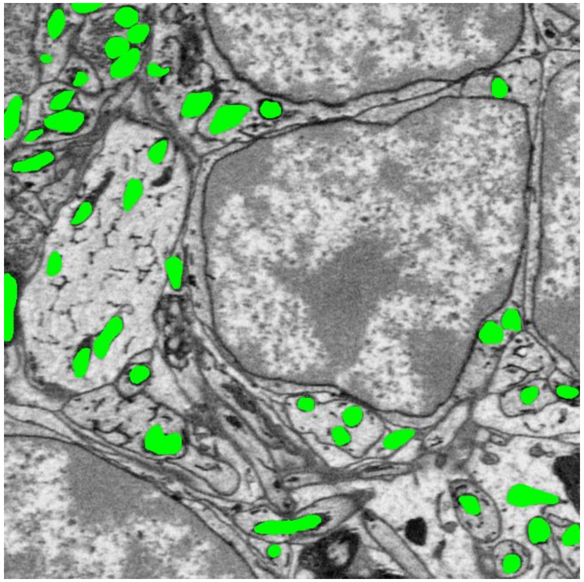
DeepEM3D



MultiResUNet

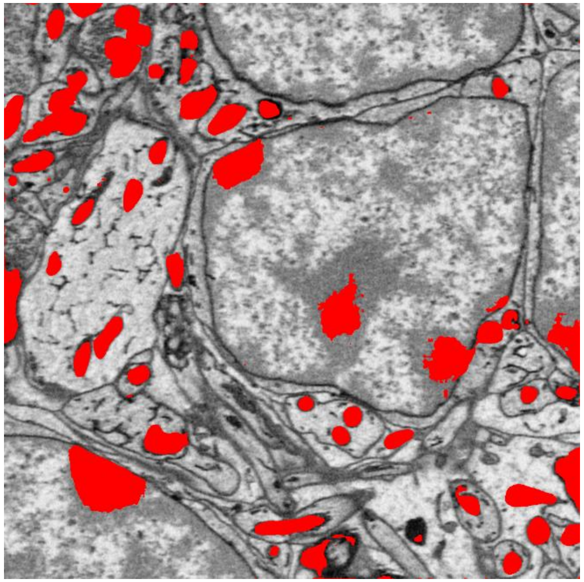


Prediction – 3fm

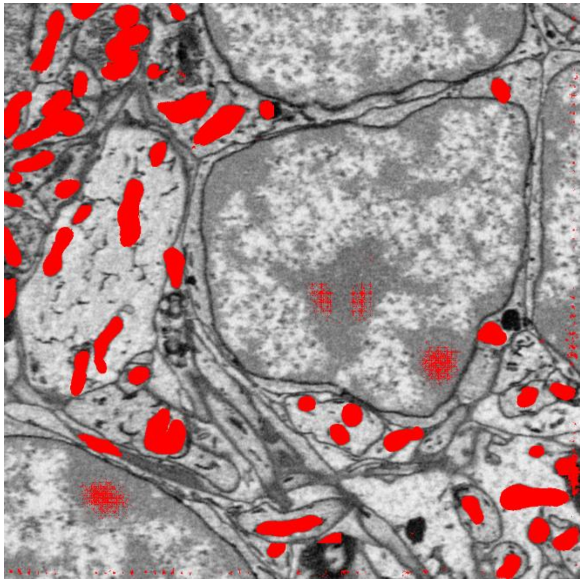


Ground Truth

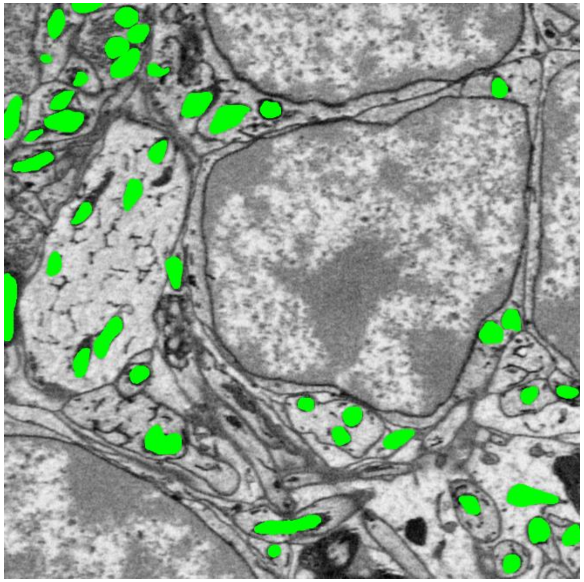
DeepEM3D



MultiResUNet

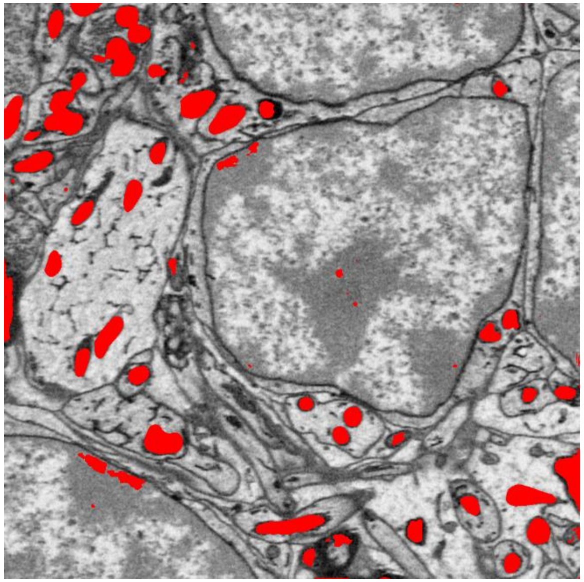


Prediction – 5fm

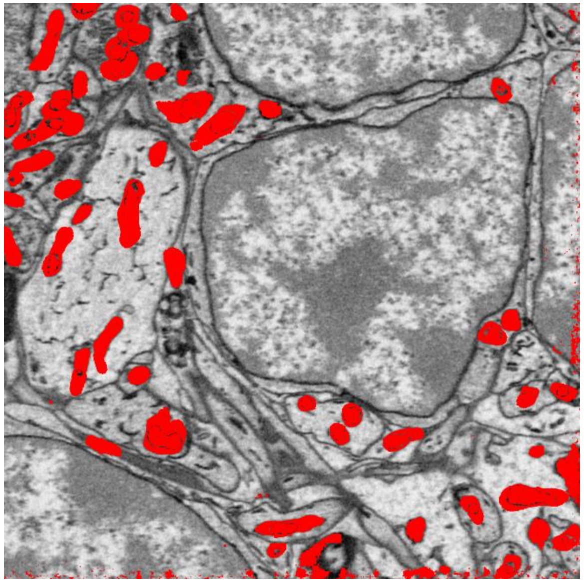


Ground Truth

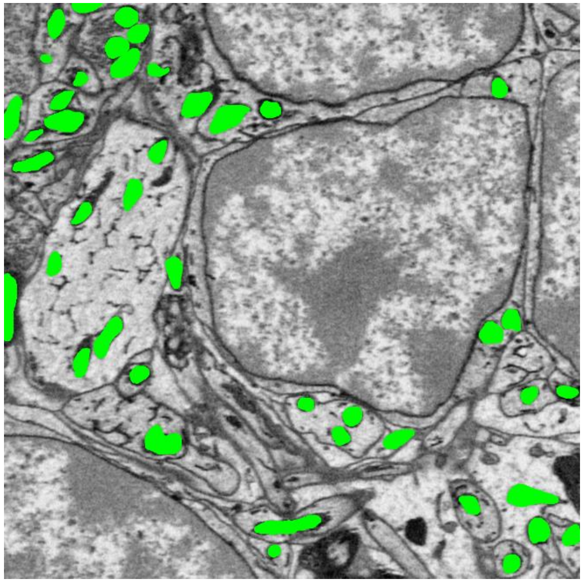
DeepEM3D



MultiResUNet

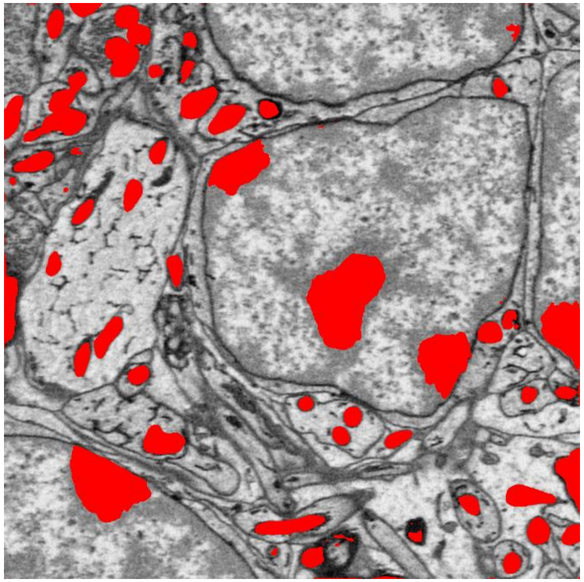


Prediction – ensembled

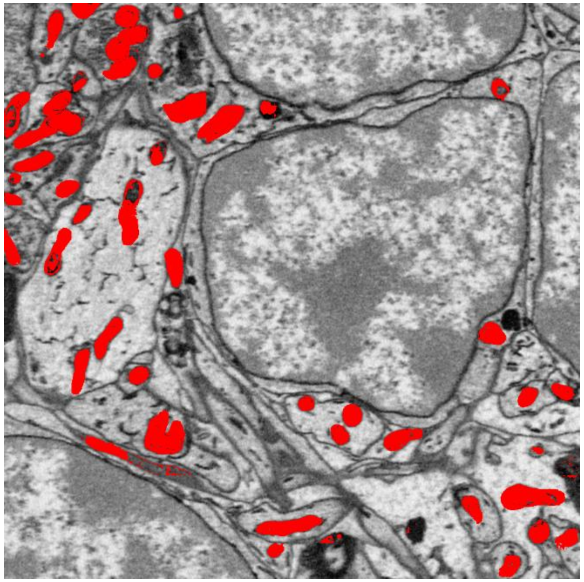


Ground Truth

DeepEM3D

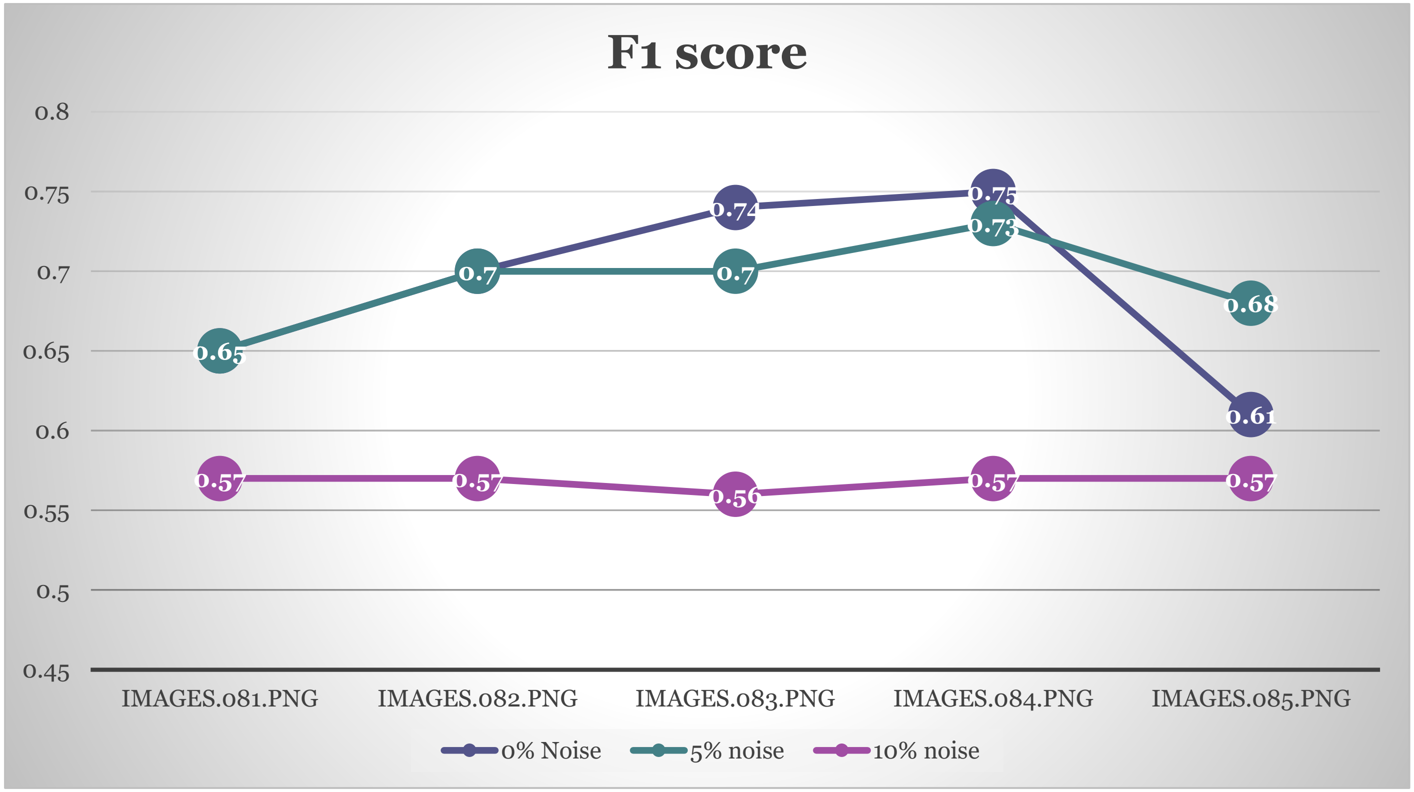


MultiResUNet



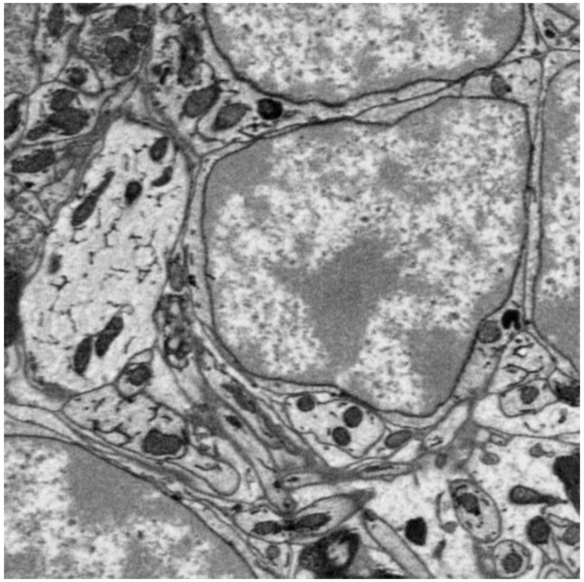
Robustness

- Robustness using 5% and 10% salt-and-pepper noise



Robustness

- Robustness using 5% salt-and-pepper



Original image

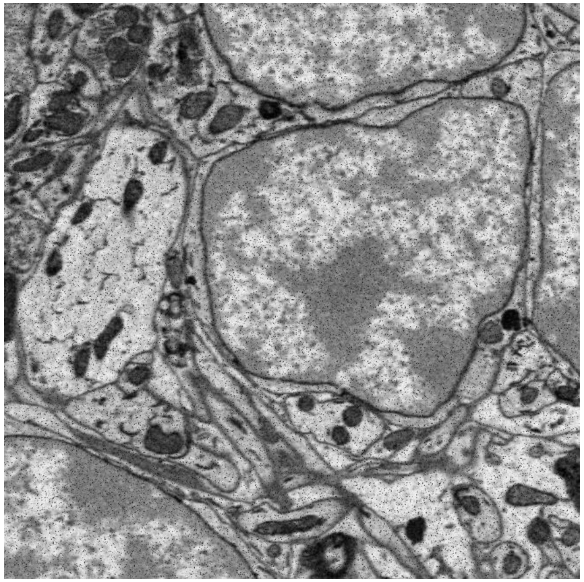
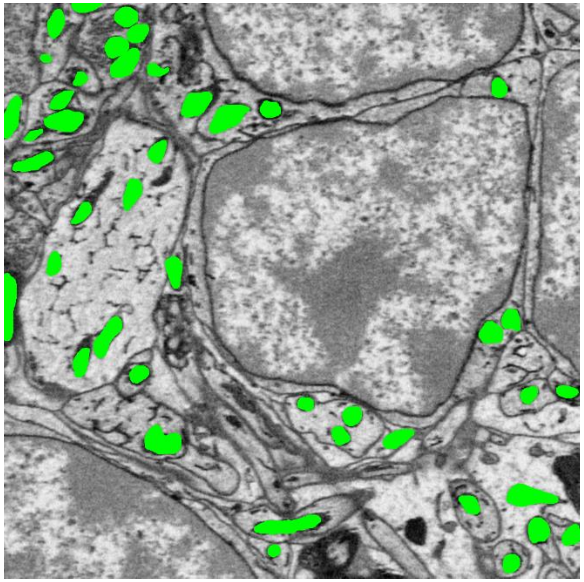
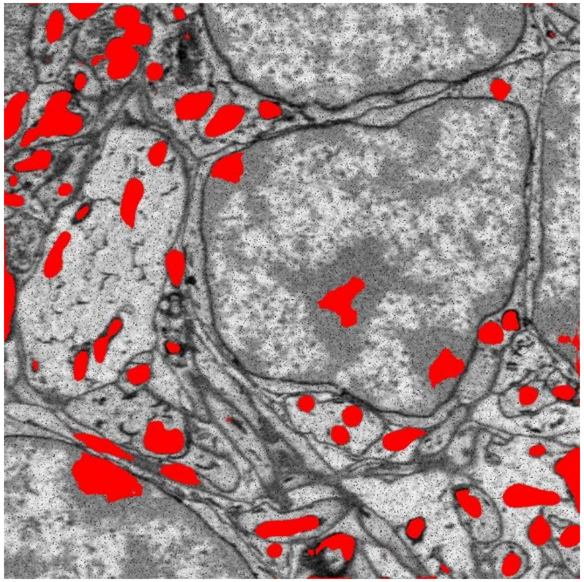


Image with 5% noise



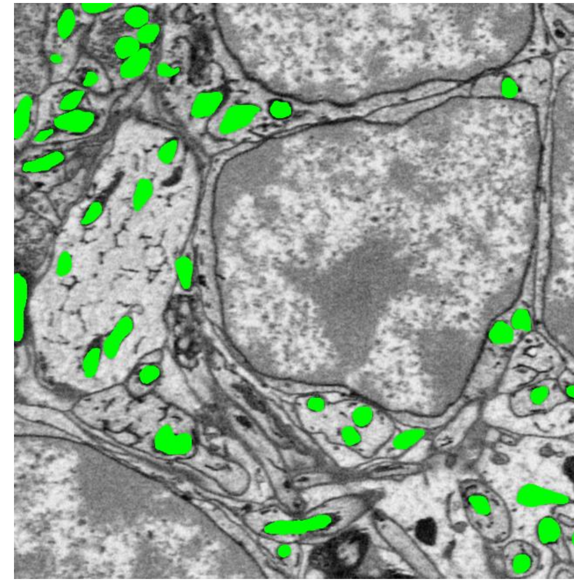
Ground Truth

Segmented image



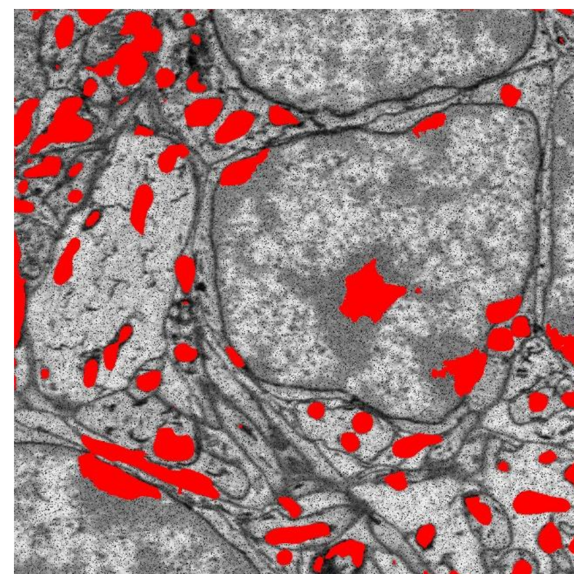
Robustness

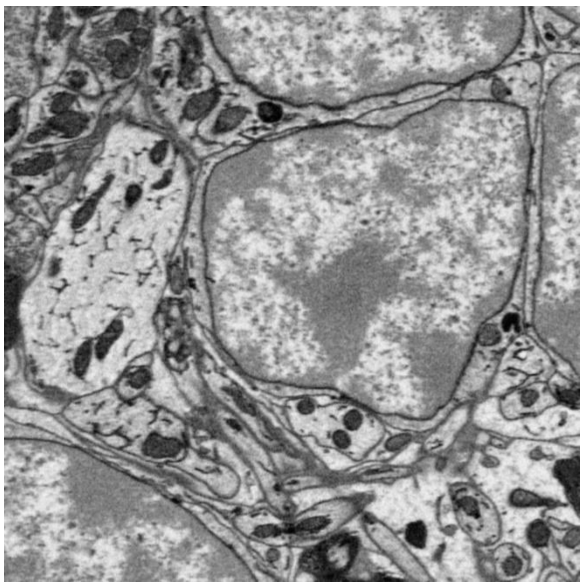
- Robustness using 10% salt-and-pepper



Ground Truth

Segmented image





Original image

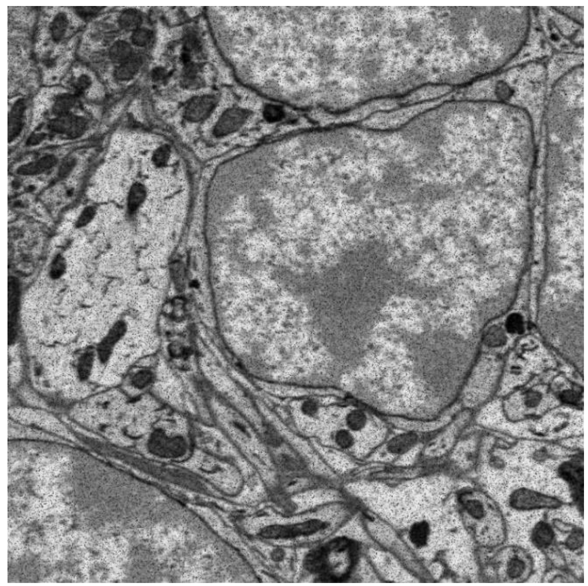
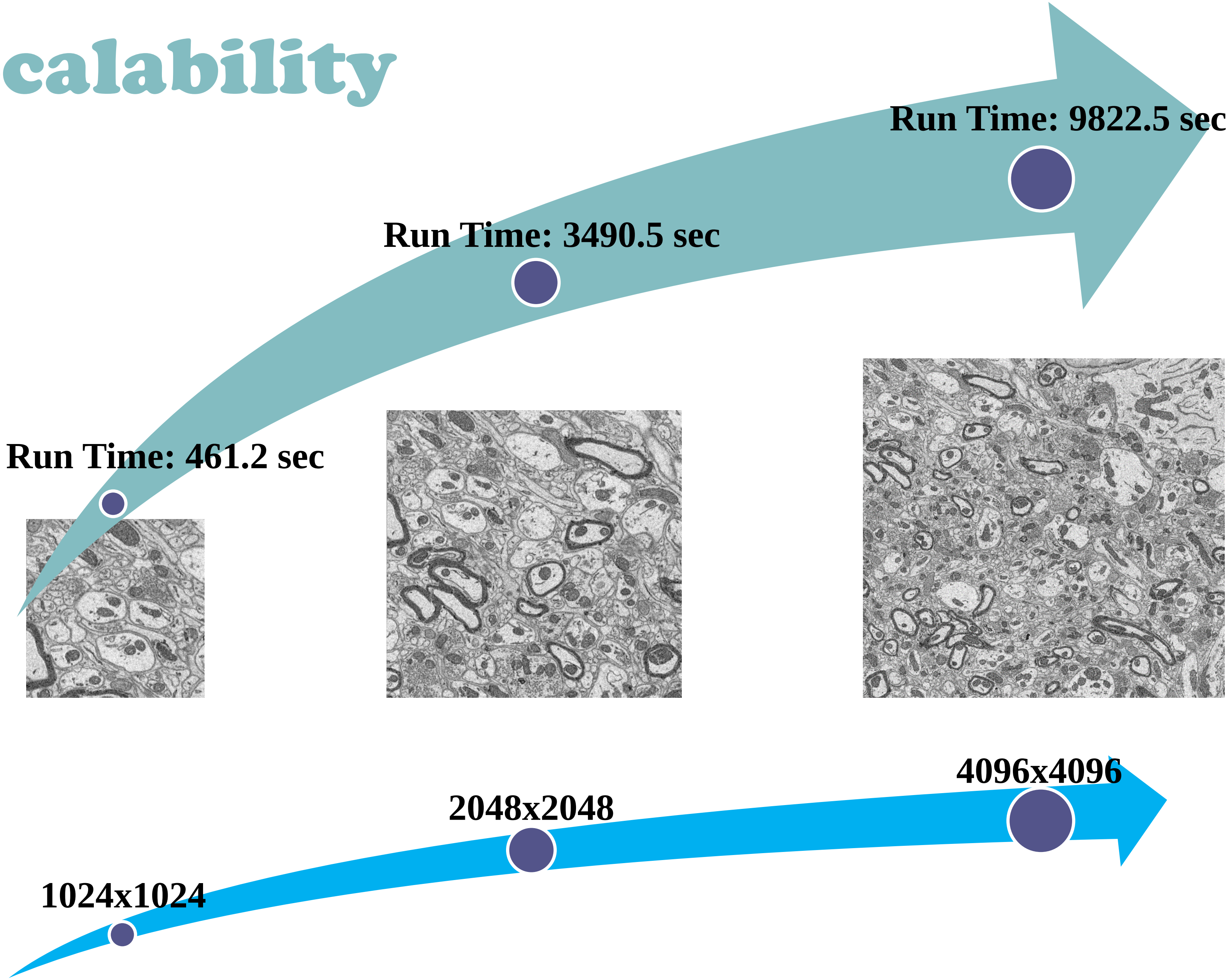


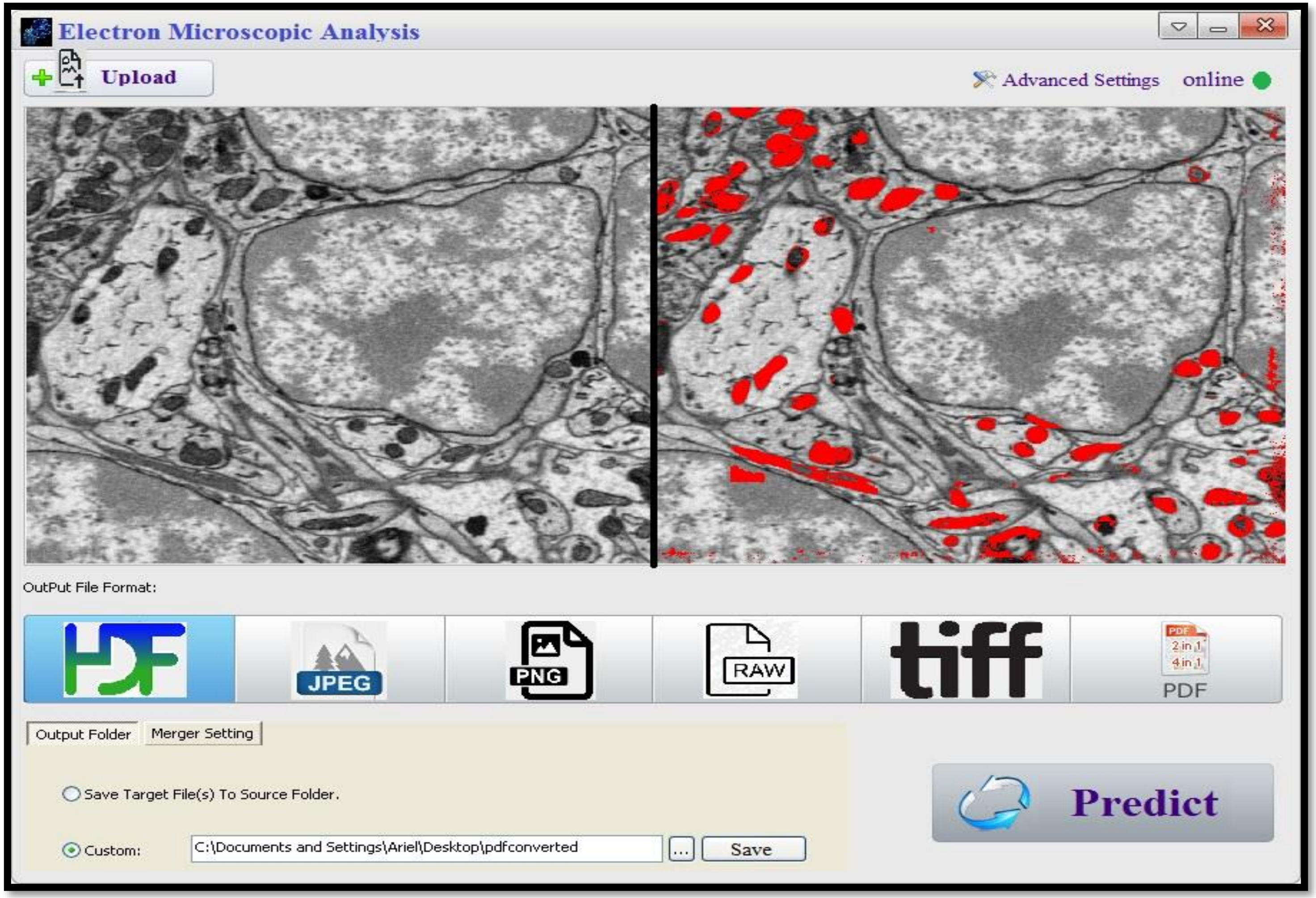
Image with 10% noise

Scalability



Deployment

- Complete solution can be encapsulated in a Software for end user

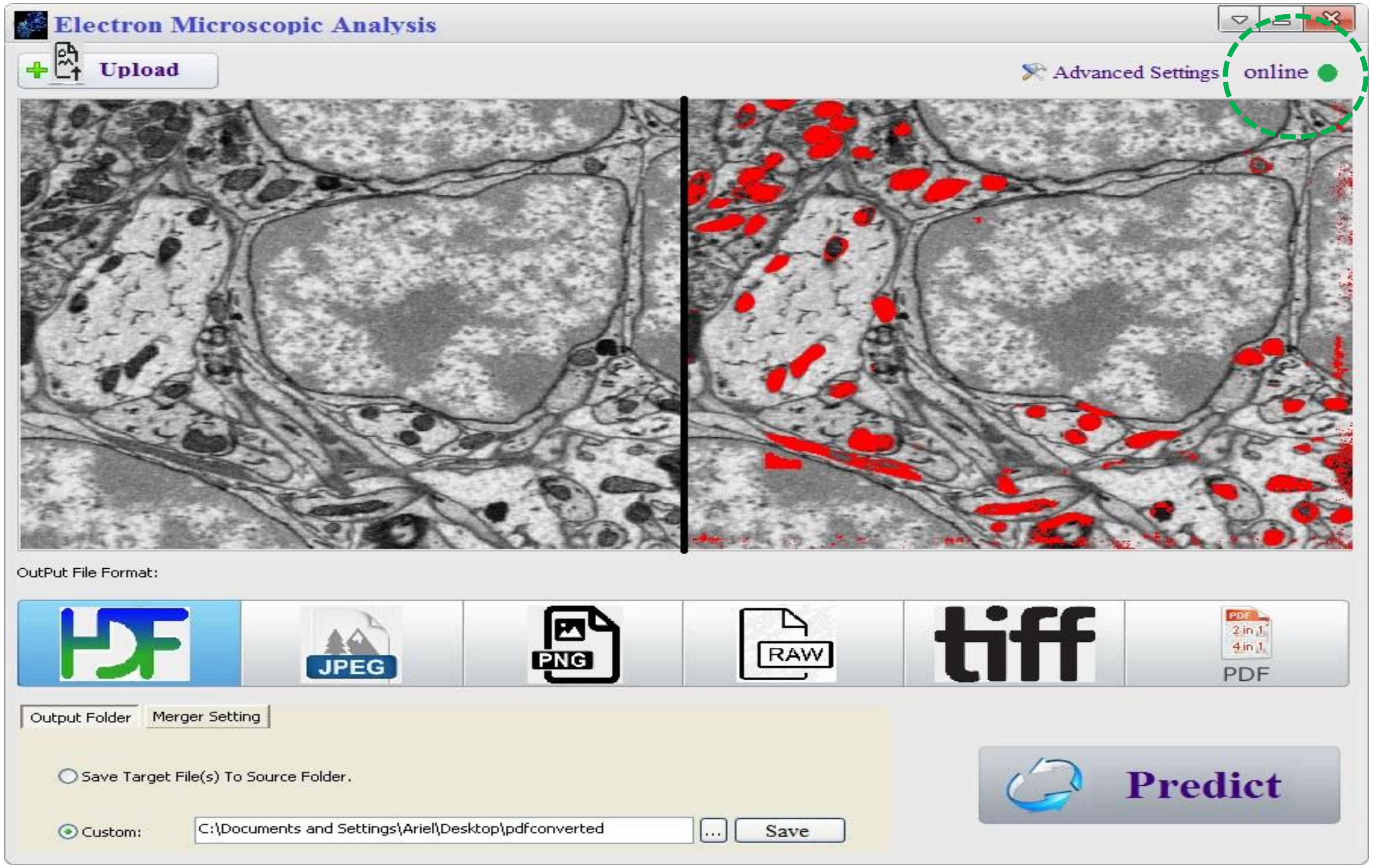


Deployment - Software

- Software Functionality – Online mode
  - Requirements: Internet connection
  - Upload an image
  - Press ‘Predict’ button and select single/multiple object(s) to be predicted
    - Image is sent to server
    - Prediction takes place at server
    - Image with object(s) identified is sent back to software
  - Original image and image with object(s) highlighted are shown together
  - Predicted image can be saved in many available options
  - Zoom in, Zoom out option to do analysis in detail

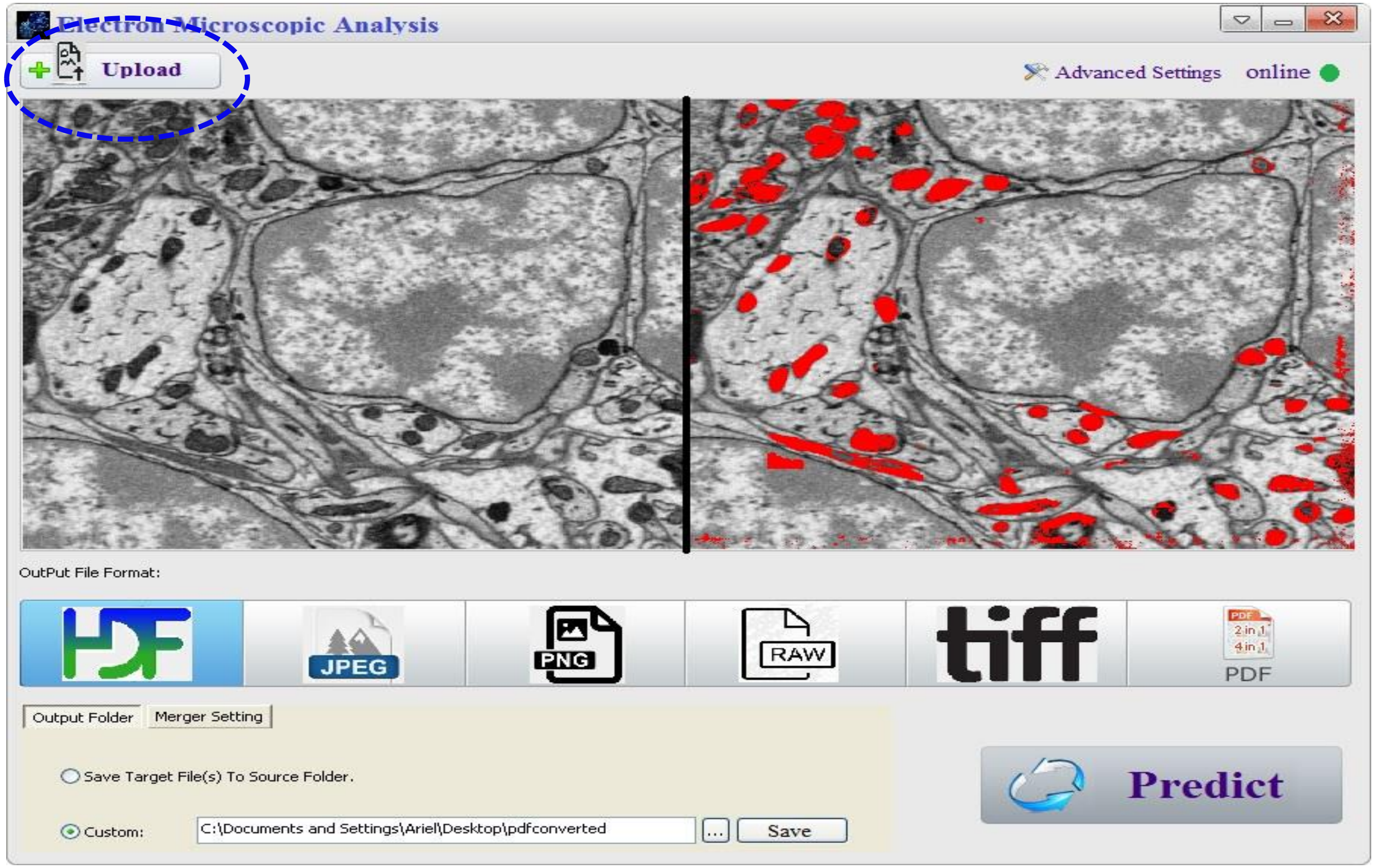
Deployment - Software

- Software Functionality – Online mode
  - Requirements: Internet connection



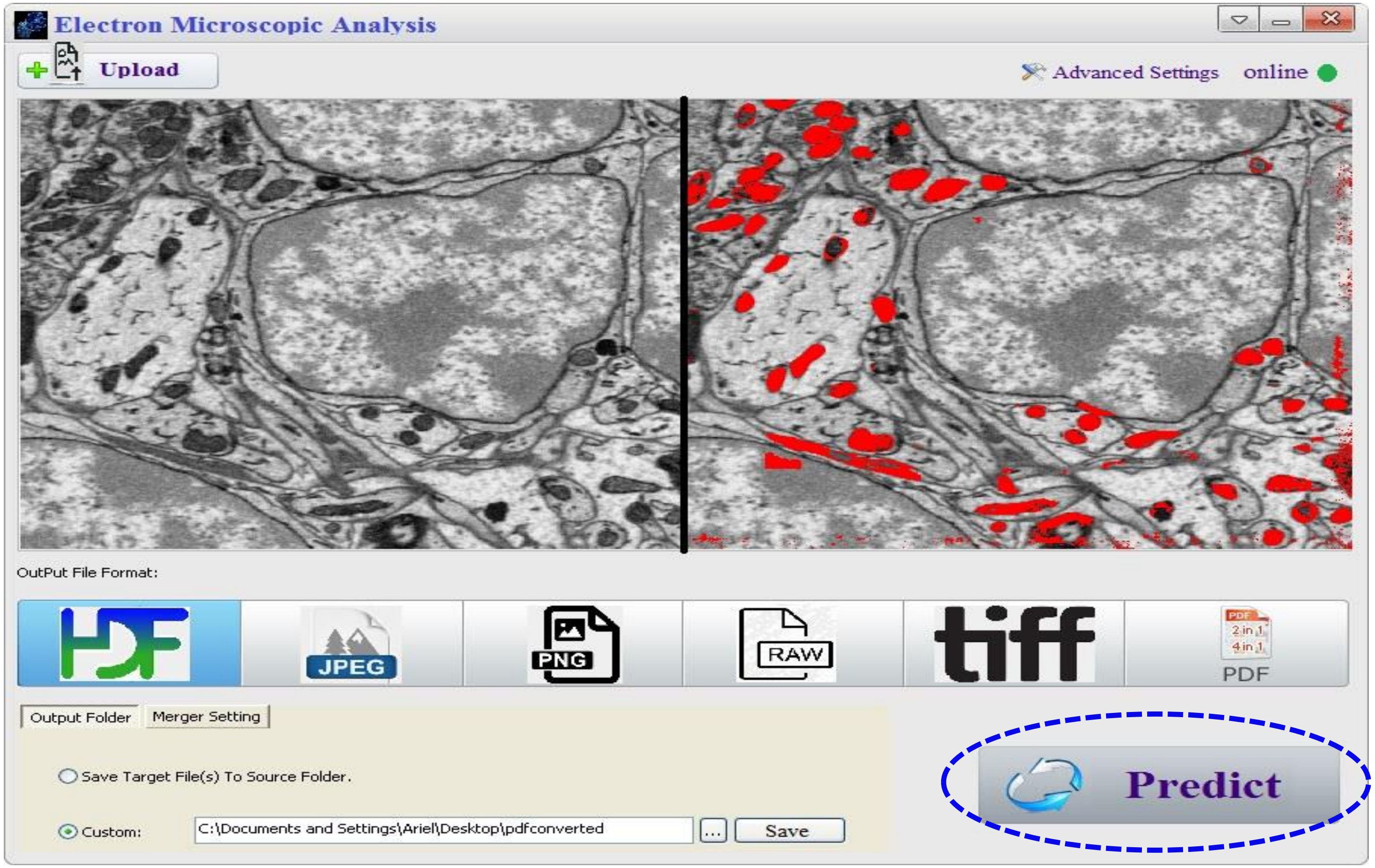
Deployment - Software

- Software Functionality – Online mode
  - Upload an image



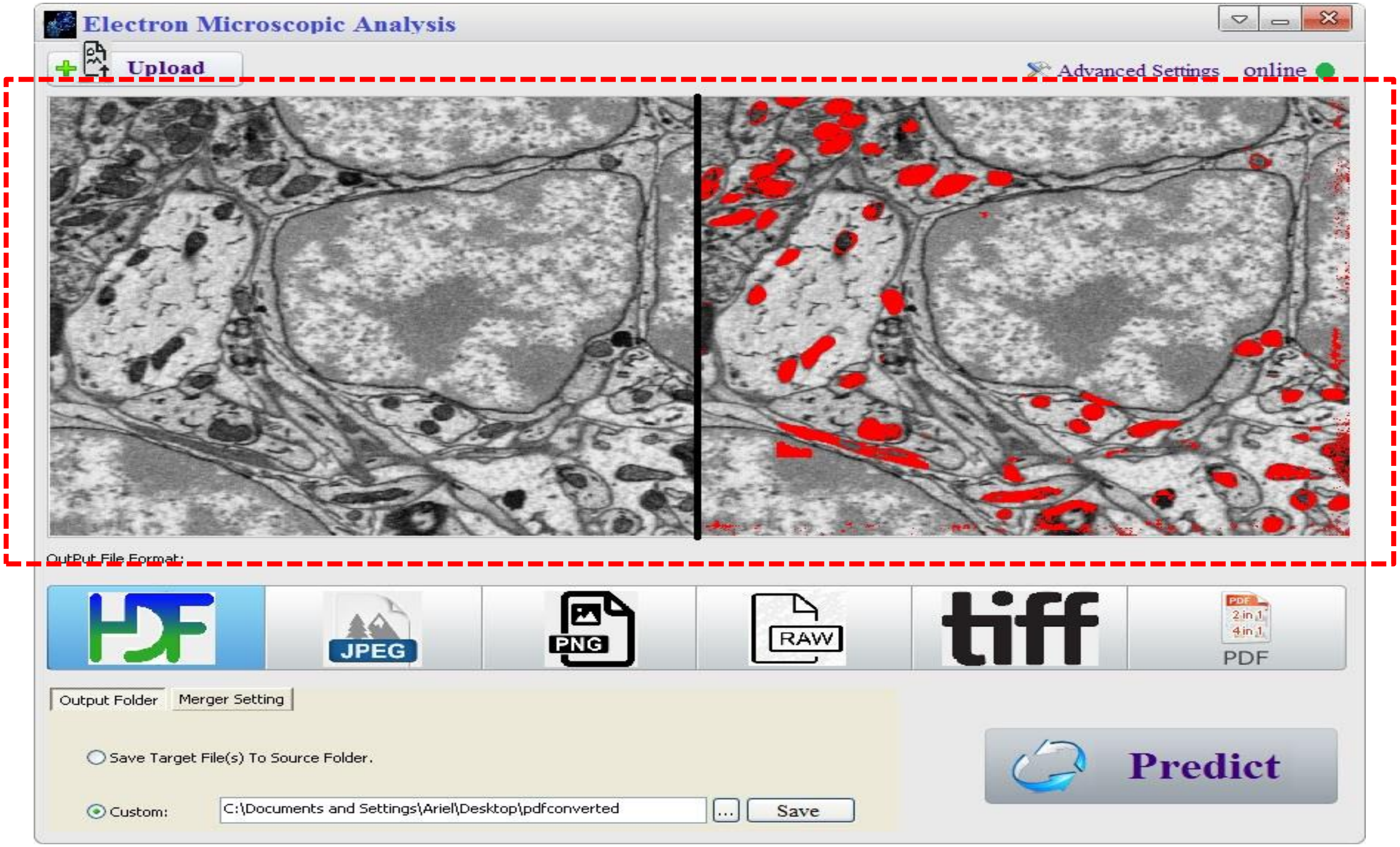
Deployment - Software

- Software Functionality – Online mode
  - Press 'Predict' button and select an/multiple object(s) to be predicted



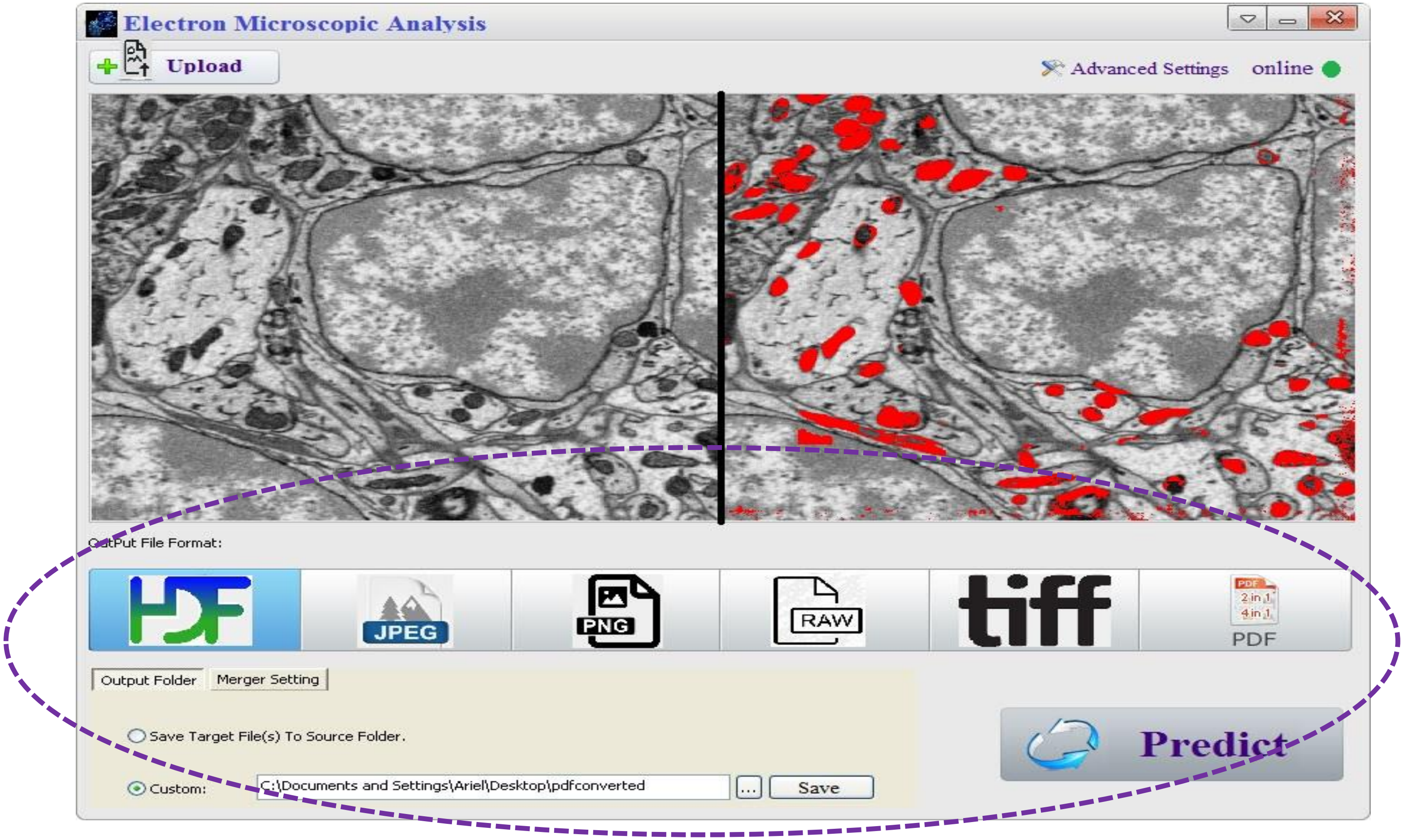
Deployment - Software

- Software Functionality – Online mode
  - Original image and image with object(s) highlighted are shown together



Deployment - Software

- Software Functionality – Online mode
  - Predicted image can be saved in many available options

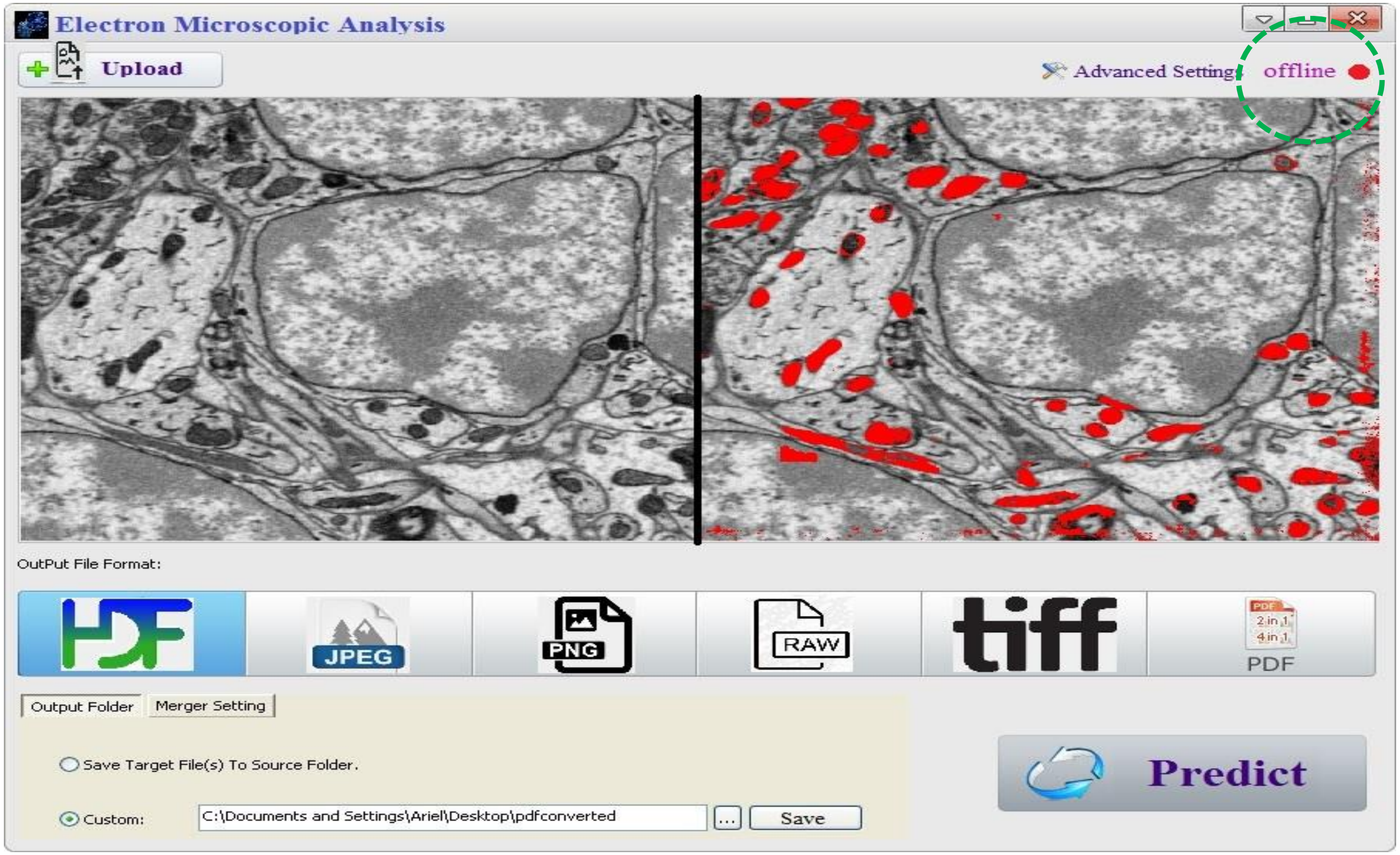


Deployment - Software

- Software Functionality – Offline mode
  - Requirements: Computer with at least 1 GPU, Python3 with Keras with tensorflow backend
  - Software is shipped with inbuilt model
  - Select an image
  - Press ‘Predict’ button and select single/multiple object(s) to be predicted
    - Model is run in user’s system with image as input
    - Predicted image with object(s) identified is generated
  - Original image and image with object(s) highlighted are shown together
  - Predicted image can be saved in many available options

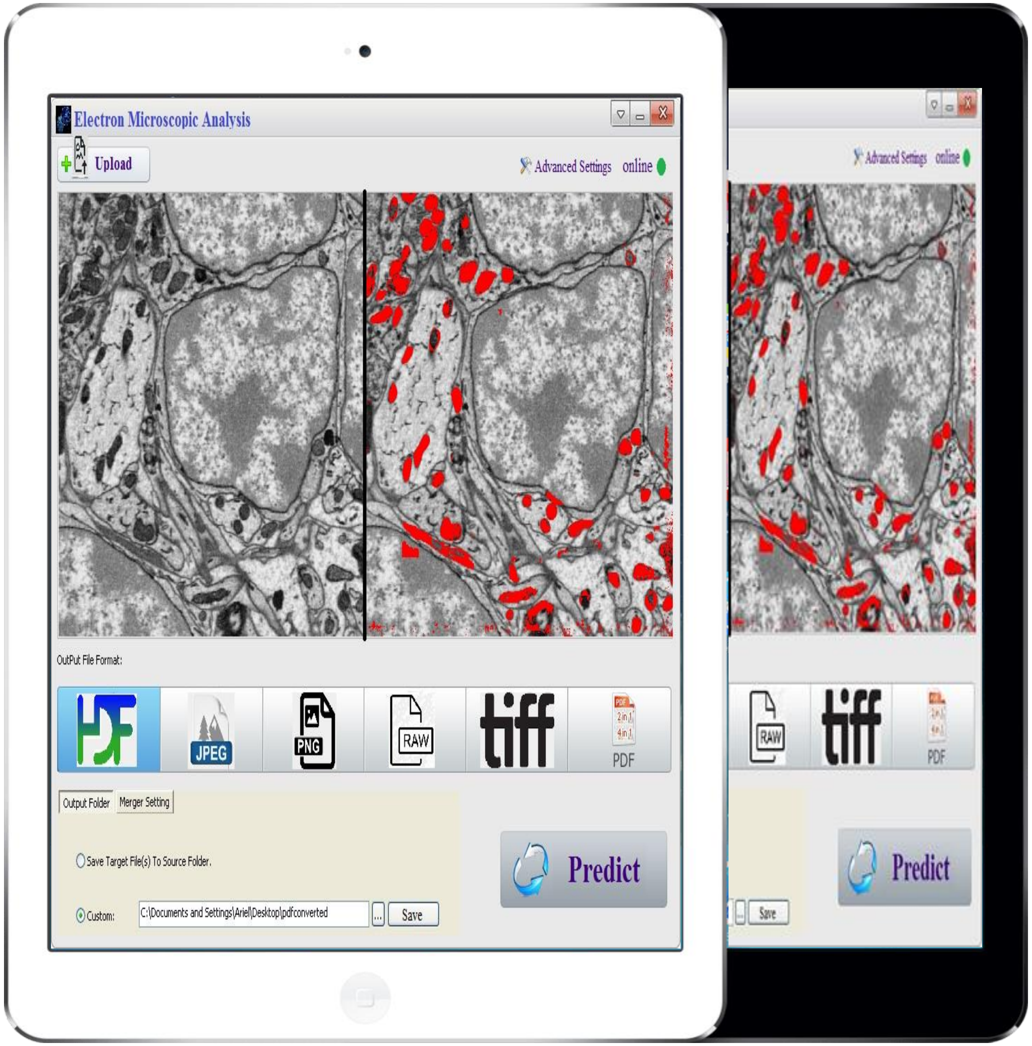
Deployment - Software

- Software Functionality – Offline mode
  - Requirements: Computer with at least 1 GPU, Python3 with Keras with tensorflow backend



Deployment - App

- Software Functionality – only Online mode
  - Requirements: iPad, Notepad, Tablet with internet connection
  - All other procedure is same as Software with online mode
  - Touchscreen zoom in zoom out support



Conclusion

- Alzheimer’s brain image segmentation using the CNN based model can help doctors/researchers in studying the changes in the brain over the time

- The models can also be used for training and segmenting other biomedical images such as skin lesions, polyps in the intestine, tumors

- The model can also be used by pharmaceutical companies for the image segmentation of animal organs

DEMO

Questions

Thank You!