Segmenting Microscopy Images

Capston Project #2

Chris Malec

Springboard Data Science Career Track

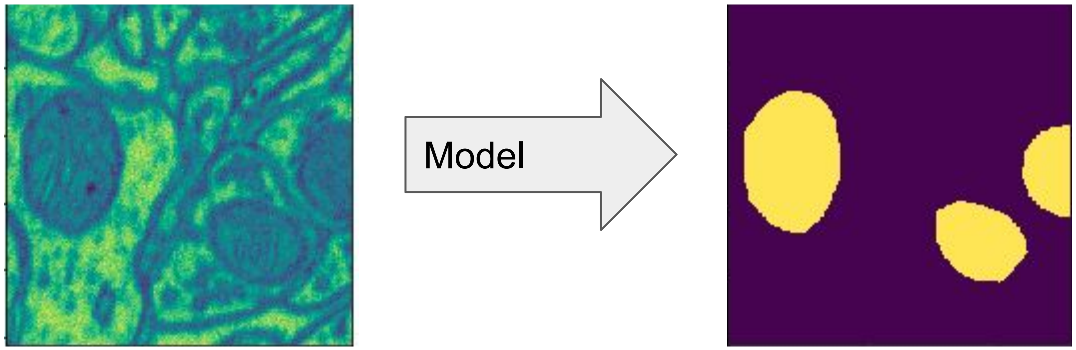
November 2019

Client Problem

- Segmenting images by hand is time consuming and laborious
- Microscopy data can be generated faster than it can be analyzed
- Segmentation requires years of experience

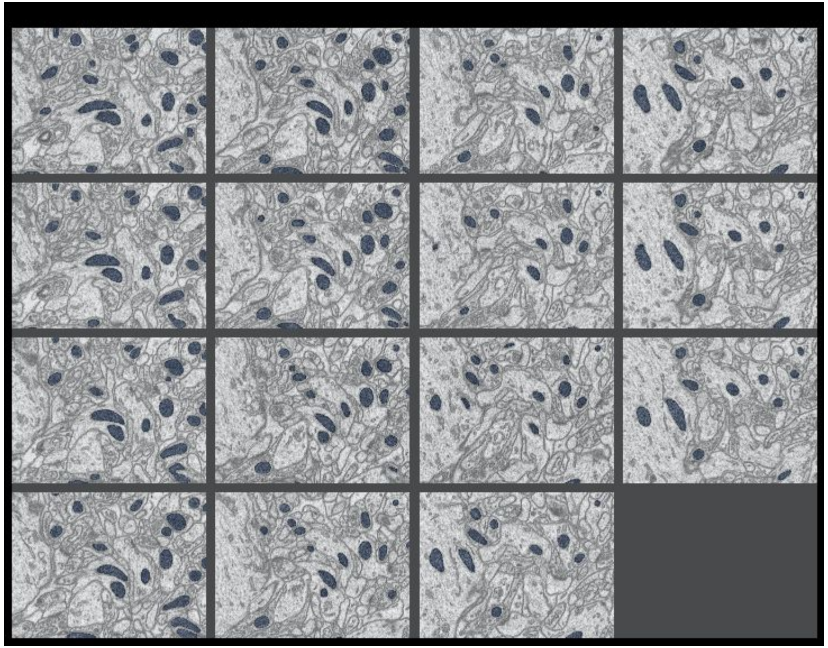
The Data Science Problem

- We would like to generate a binary image from an input image
- A pixel labeled ‘1’ is in the category of interest
- A pixel labeled ‘0’ is not in the category of interest



Data Wrangling

- Data took the form of four multi-frame tif files
- 165 images and ground-truths for the training set
- 165 images and ground-truths for the test set
- Both sets are from the same microscope run, but separated by thousands of image slices

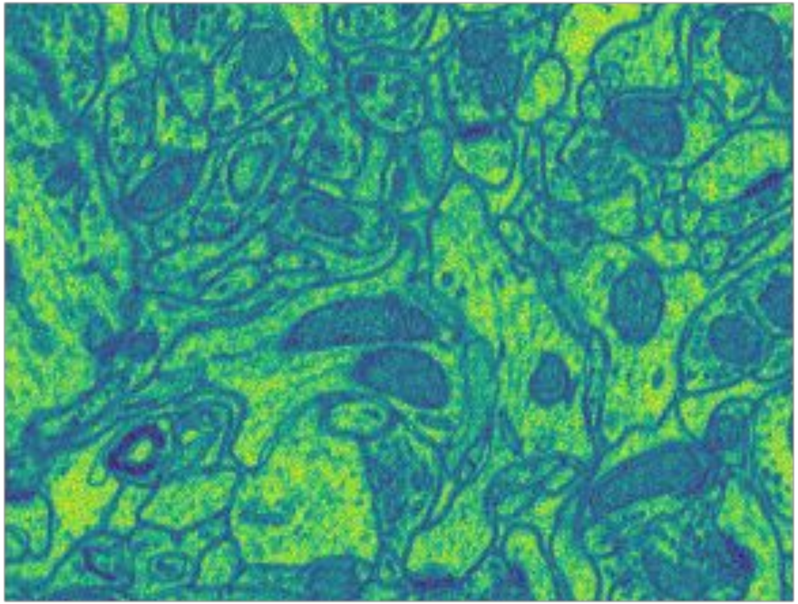


Exploratory Analysis

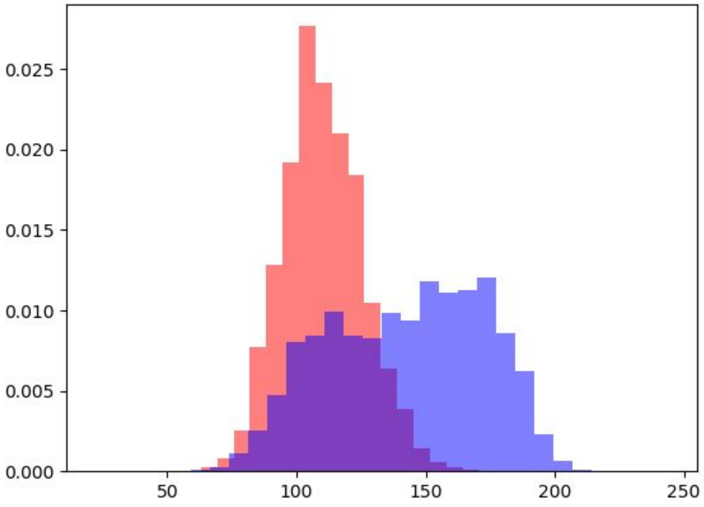
- Each voxel represents about 5 nm^3 of volume

- Large images, 768x1024 pixels

- On average, mitochondria take up about 13 million nm^3 or a cube about 48 pixels on a side



Exploratory Analysis



- The pixel intensity of mitochondria (red) is lower than on average than non-mitochondria (blue)
- There is a large overlap in intensity between the two regions
- Pixel intensity alone can't be used for segmentation, shape and pattern must also play a role

The Machine Learning Model - Convolutional Neural Network

- Convolutional Neural Nets are a class of supervised learning algorithms that use neural nets and convolutions
- They work well for problems with short range order, like images
- A convolution operation acts on a set of nearby pixels with a filter to search for patterns.

1	25	23	21	25	23	5	1
6	25	0	0	0	0	25	6
25	0	84	82	86	82	0	22
0	81	0	86	81	0	83	0
0	81	82	81	79	89	84	0
24	0	83	0	0	82	0	1
6	25	0	81	81	0	26	25
1	6	26	0	0	25	6	6

The Machine Learning Model - Convolutional Neural Network

- This filter looks for dark horizontal lines

1	1	1
-1	-1	-1
1	1	1

1	25	23	21	25	23	5	1
6	25	0	0	0	0	25	6
25	0	84	82	86	82	0	22
0	81	0	86	81	0	83	0
0	81	82	81	79	89	84	0
24	0	83	0	0	82	0	1
6	25	0	81	81	0	26	25
1	6	26	0	0	25	6	6

The Machine Learning Model - Convolutional Neural Network

- This filter looks for dark horizontal lines

1	1	1
-3	-3	-3
1	1	1

- Multiplying this group of pixels gives -7, a low score

1	25	23	21	25	23	5	1
6	25	0	0	0	0	25	6
25	0	84	82	86	82	0	22
0	81	0	86	81	0	83	0
0	81	82	81	79	89	84	0
24	0	83	0	0	82	0	1
6	25	0	81	81	0	26	25
1	6	26	0	0	25	6	6

The Machine Learning Model - Convolutional Neural Network

- This filter looks for dark horizontal lines

1	1	1
-3	-3	-3
1	1	1

- Multiplying this group of pixels gives 321, a high score

1	25	23	21	25	23	5	1
6	25	0	0	0	0	25	6
25	0	84	82	86	82	0	22
0	81	0	86	81	0	83	0
0	81	82	81	79	89	84	0
24	0	83	0	0	82	0	1
6	25	0	81	81	0	26	25
1	6	26	0	0	25	6	6

The Machine Learning Model - Convolutional Neural Network

- This filter looks for dark horizontal lines

1	1	1
-3	-3	-3
1	1	1

- Each filter passes the entire image

1	25	23	21	25	23	5	1
6	25	0	0	0	0	25	6
25	0	84	82	86	82	0	22
0	81	0	86	81	0	83	0
0	81	82	81	79	89	84	0
24	0	83	0	0	82	0	1
6	25	0	81	81	0	26	25
1	6	26	0	0	25	6	6

The Machine Learning Model - Convolutional Neural Network

- This filter looks for dark horizontal lines

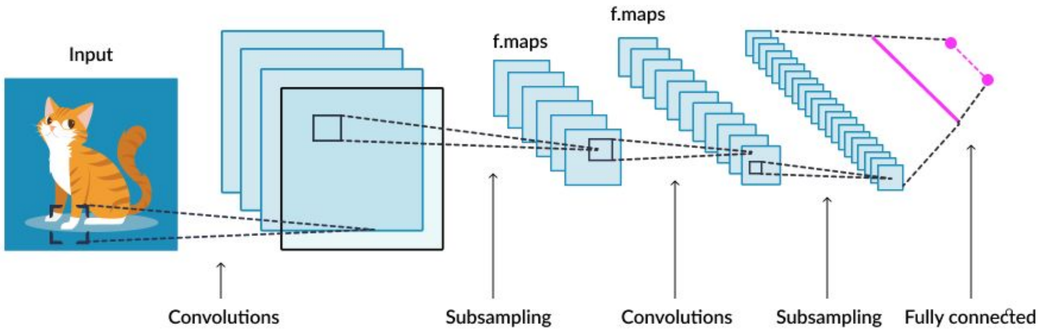
1	1	1
-3	-3	-3
1	1	1

- Each filter passes the entire image
- This picks out specific features in the image for each filter

1	25	23	21	25	23	5	1
6	25	0	0	0	0	25	6
25	0	84	82	86	82	0	22
0	81	0	86	81	0	83	0
0	81	82	81	79	89	84	0
24	0	83	0	0	82	0	1
6	25	0	81	81	0	26	25
1	6	26	0	0	25	6	6

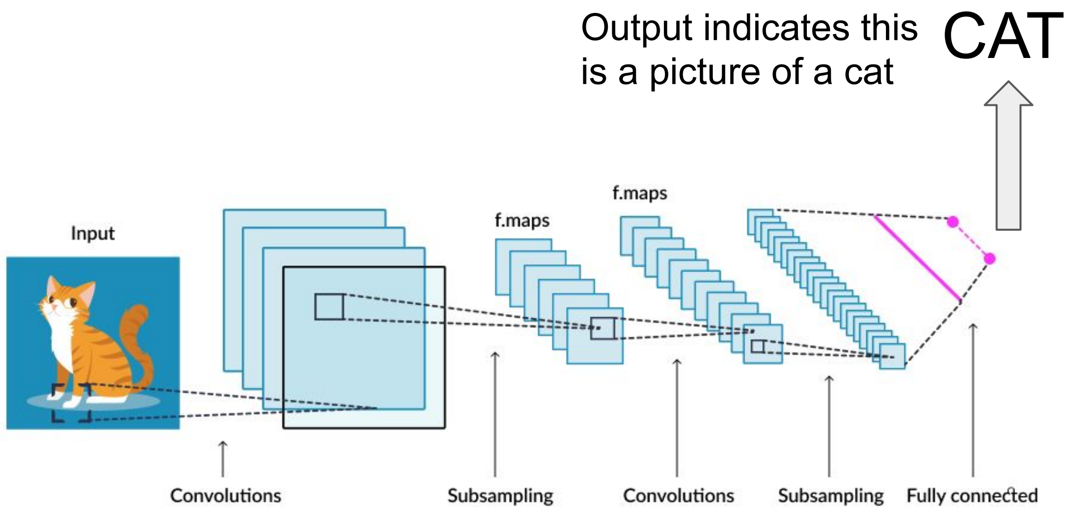
Convolutional Neural Nets

- Convolutional neural nets use a convolution operation instead of fully connected layers
- The weights that the model learns in this case are the values for the many filters involved in the convolutions
- Each layer of the net searches for more complicated patterns



Convolutional Neural Nets

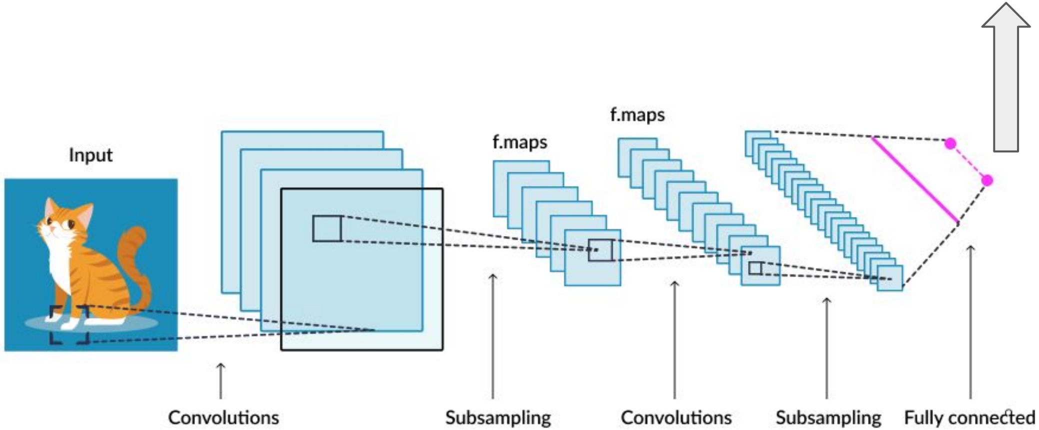
- Many CNN’s will follow a pattern of convolution followed by downsampling, increasing the number of filters with each step
- This would end with a large number of filters representing a few pixels, with an output to a fully connected layer



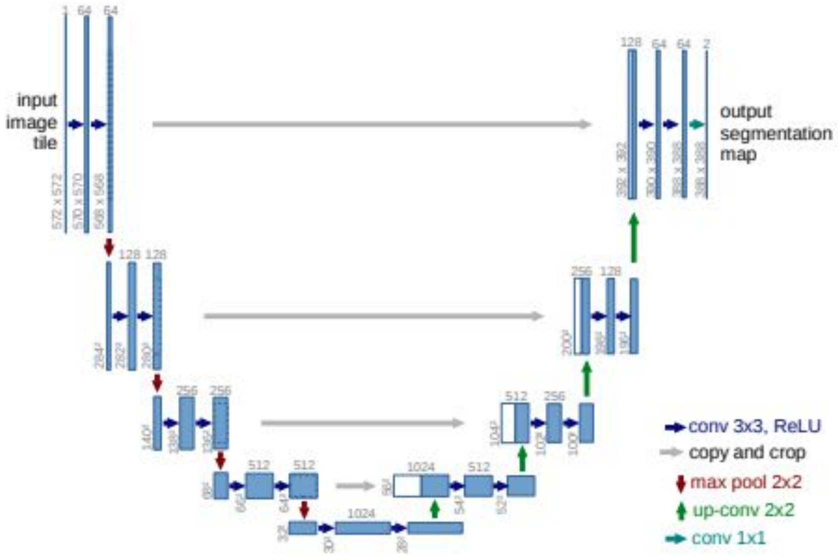
Convolutional Neural Nets

Output is picture showing which pixels are part of the cat

- Many CNN’s will follow a pattern of convolution followed by downsampling, increasing the number of filters with each step
- This would end with a large number of filters representing a few pixels, with an output to a fully connected layer
- But we want the output to be a binarized image



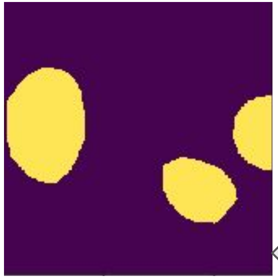
U-Net

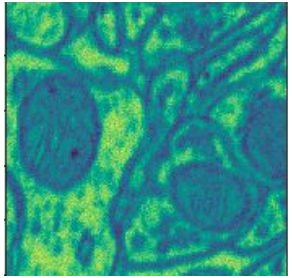


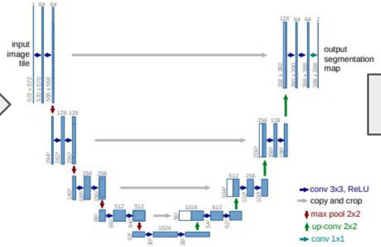
- The solution is U-Net, which takes a the usual series of convolution and downsample operations, and then reverses them
- The upsampling layers concatenate the output of each downsampling layer with the input of an upsampling layer
- The final output sends each pixel to a sigmoid activation function so that the result is a segmentation map of the original resolution

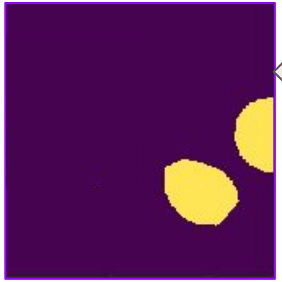
U-Net

- Experts have annotated a series of microscopy images
- The neural net attempts to create a segmentation mask
- The predicted segmentation is compared to the expert segmentation and a loss function is calculated



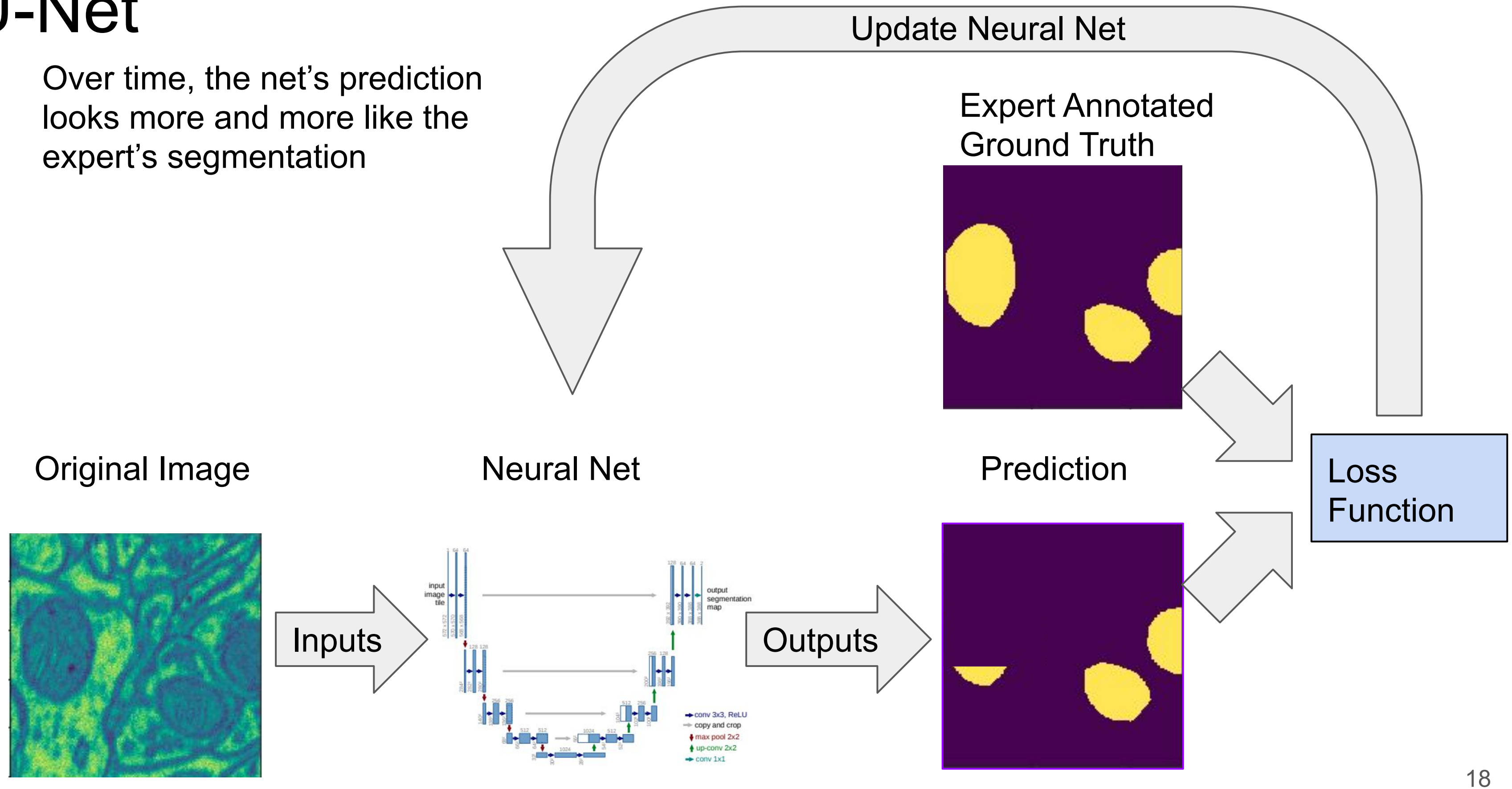






U-Net

- Over time, the net's prediction looks more and more like the expert's segmentation

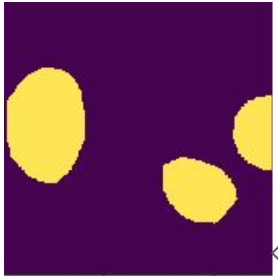


U-Net

- Over time, the net's prediction looks more and more like the expert's segmentation
- Eventually it should match very closely

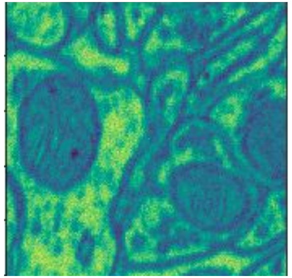
Update Neural Net

Expert Annotated Ground Truth



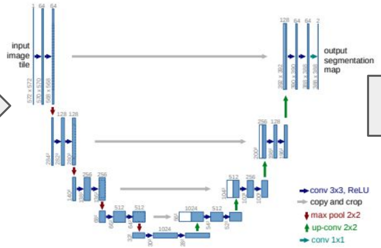
Loss Function

Original Image



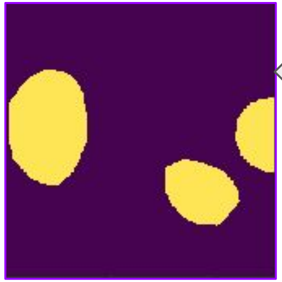
Inputs

Neural Net

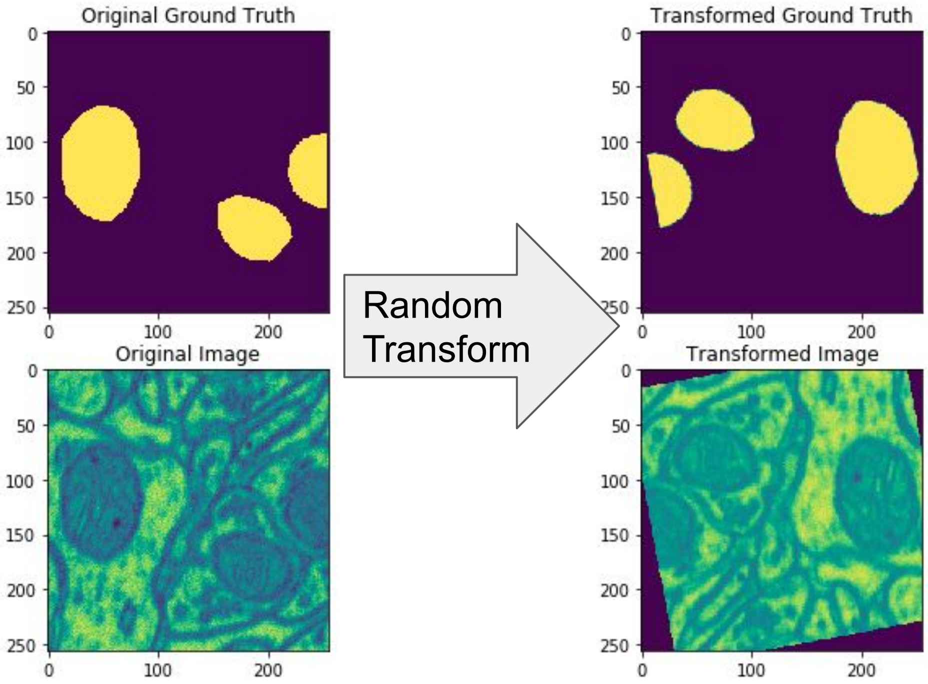


Outputs

Prediction



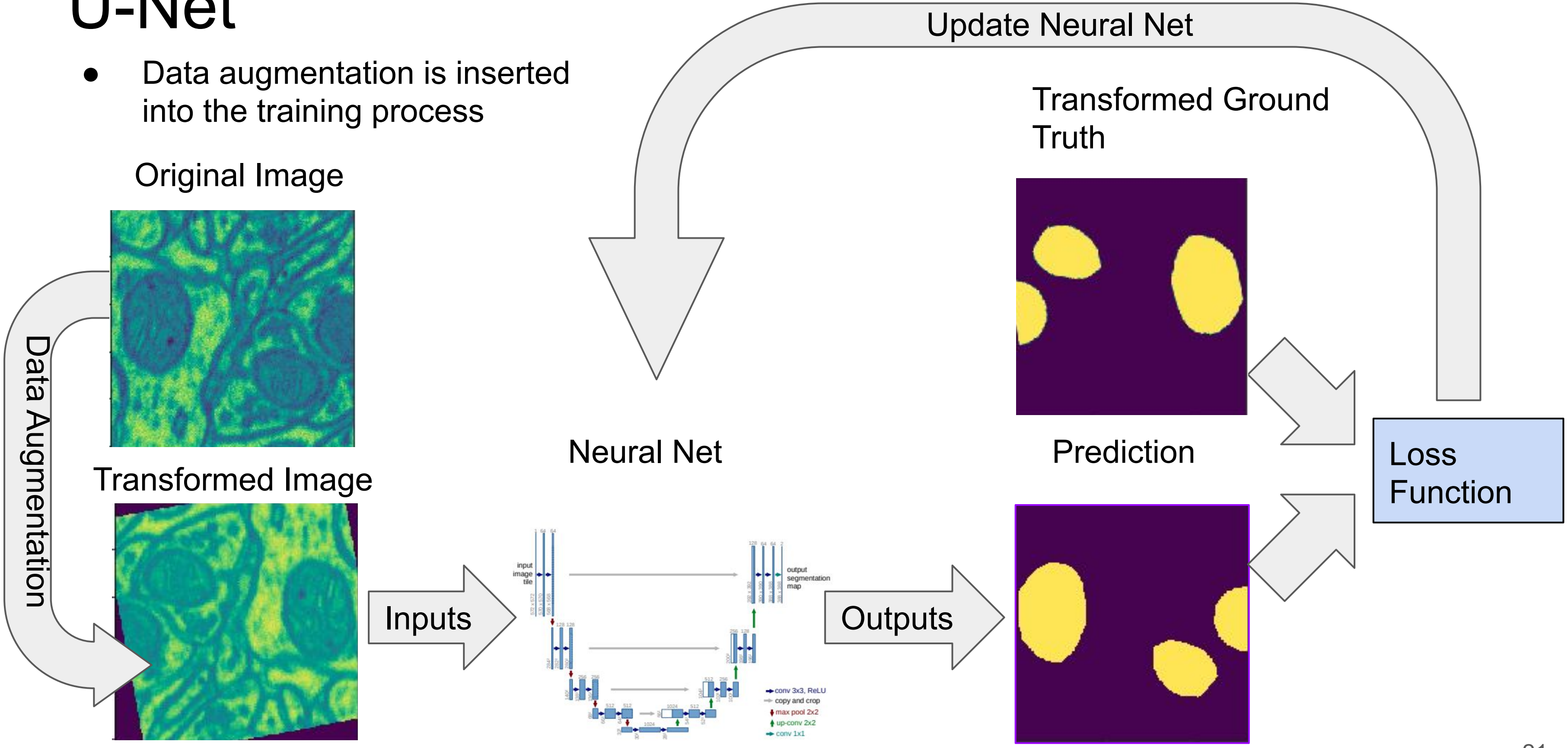
Data Augmentation



- Another important technique used here is data augmentation
- A lot of times data is more scarce than what is necessary to get a good fit.
- We can create more data by applying random transformations such as rotation, flips, random noise, etc.
- In this case it is very important to transform the ground-truth images exactly the same way as the original image
- It improves both the training loss and leads to better performance on unseen data

U-Net

- Data augmentation is inserted into the training process



Hyper-parameter tuning

- Neural Nets have an exceptional number of hyper-parameters
- The long training time can also mean that it is difficult to explore hyperparameters in a gridsearch fashion
- In the end, systematic searches of parameters must be combined with an understanding of different hyperparameter properties to reach an optimal solution

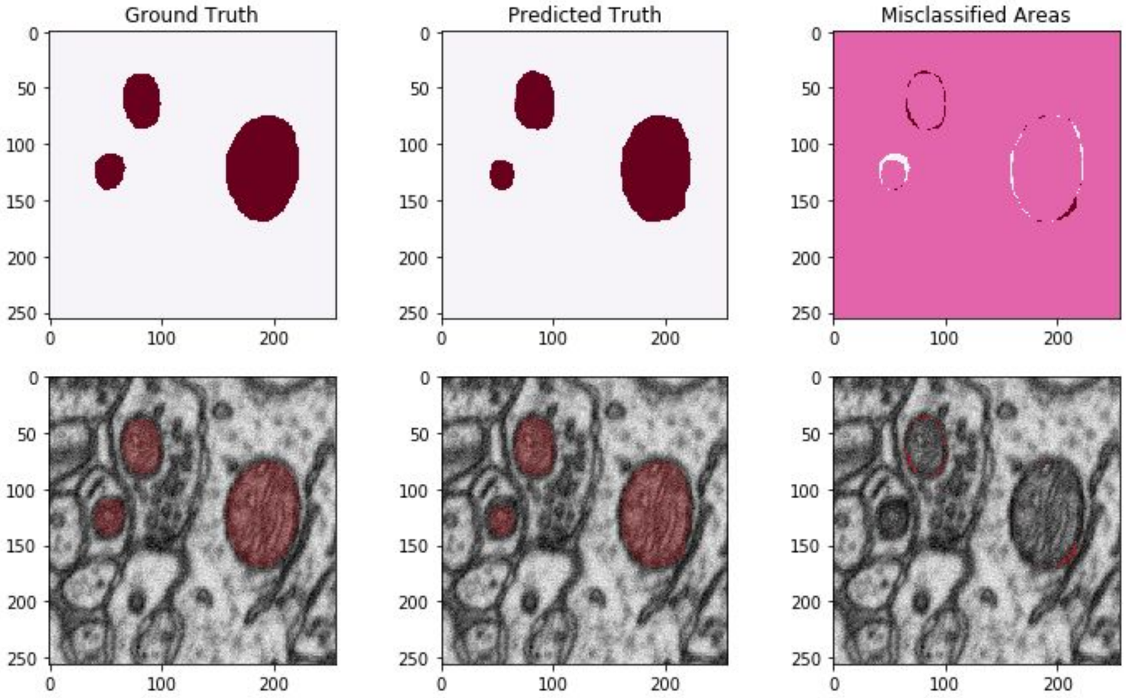
Evaluation: The Dice Coefficient

- The Dice coefficient measures the overlapping pixels between the predicted and true values, then divides by the total number of values that are true
- This is a common measure for how well two binary images match
- It lies between 0 and 1
- This can easily be turned into a loss function so that :

  Dice Loss = log(1 - D.C.)

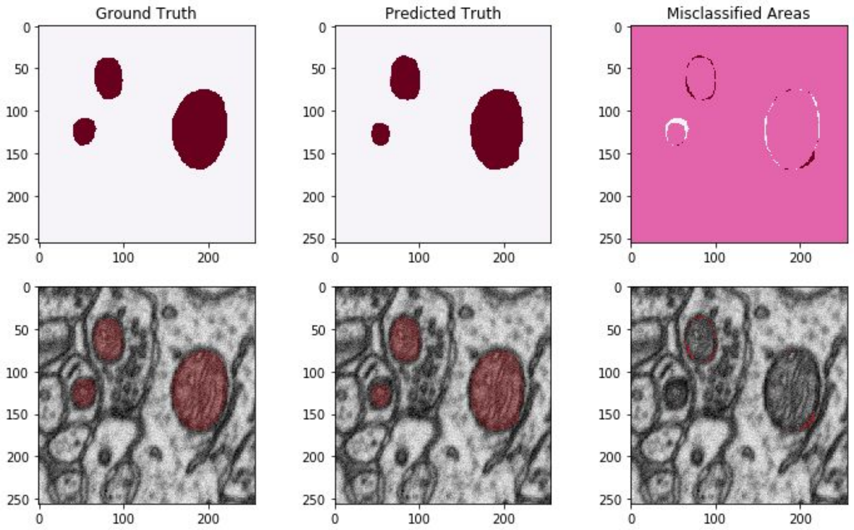
$$Dice\ Coefficient = \frac{2 \cdot \Sigma X \cap Y}{\Sigma X \cup Y}$$

Results - Segmented Images



Dice Coefficient - 0.91

Results - Segmented Images



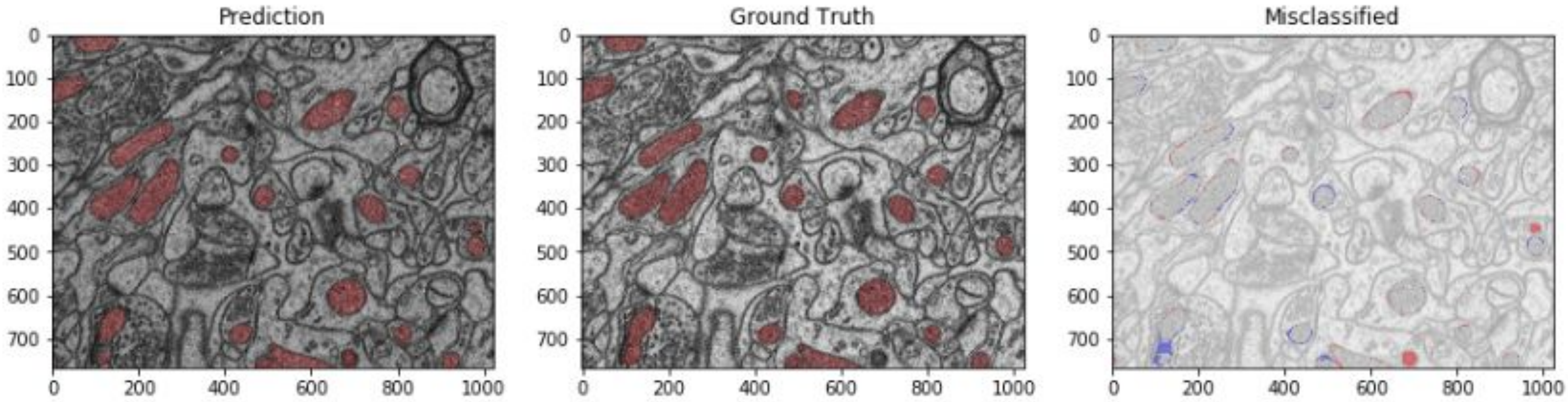
Dice Coefficient - 0.91

- After hyper-parameter tuning and data augmentation, a dice coefficient of 0.91 was achieved on the test set
- Inspection of various individual images demonstrate that the very edges of mitochondria are difficult to correctly categorize
- From a practical standpoint, this is not a huge issue, as it may not be important to outline the mitochondria perfectly so much as to correctly label as many mitochondria as possible

Results - Segmented Images

- This image was created by scanning a smaller window across the large image to generate many predicted values that were averaged together

- For this image the only small areas of false negatives (blue) and false positives (red), the rest are on the edges of correctly classified mitochondria



Client Recommendations - Data acquisition

- Acquiring more data is essential to improve the model, as the current network was trained from only one microscope run

- Standard practices could be adopted to insure uniformity of images, for example standard magnification settings, or at least recording the magnification

- With proper data collection, the technique can easily be extended to any number of biological image segmentation tasks

Client Recommendations - Metrics and explanations for researchers

- The Dice coefficient is not particularly useful for deciding whether or not the algorithm produces false negatives or false positives

- A method of counting the number of mitochondria correctly identified

- Translate into precision and recall of individual mitochondria

- Particularly important when application tests for presence or absence of disease, ethical and legal concerns require a quantification of the error in the algorithm that fits in with standard practices

Client Recommendations - Metrics and explanations for researchers

- Applications may also require that researcher or doctor can ask to see how the algorithm arrived at its answers
- The filters in the net most responsible for the identification can be generated
- This would constitute a type of ‘explainable AI’
- Significant investment in this aspect of the algorithm may be necessary to gain trust with doctors and researchers who are used to relying on their intuition

Future Work

- Three dimensional convolutions
  - Make use of truly 3D data
  - Would have less training examples, but may reveal better patterns in each example
- Focus on data augmentation
  - Not all transformations are independent
  - Some transformations clip pixels, and therefore some sections of the image are underrepresented
- More computing power
  - GPUs
  - Google Cloud
  - AWS
  - Microsoft Azure

References

- Dataset: CVLab SEM dataset: https://cvlab.epfl.ch/data/data-em/
- Convolutional neural net cat example: missinglink.ai
- U-Net: U-Net- Convolutional Networks for Biomedical Image Segmentation, Ronneberger et al, 2015, https://arxiv.org/pdf/1505.04597.pdf
- U-Net tutorial: https://github.com/tensorflow/models/tree/master/samples/outreach/blogs/segmentation_blogpost
- Explainable AI: https://www.darpa.mil/attachments/XAIProgramUpdate.pdf