Leveraging Convolutional Neural Networks to Advance the Identification of Schizophrenia in Structural MRI.

Project Statement

- Use MRI images of various degrees of schizophrenia to train a neural network that can classify whether a patient is likely to be developing schizophrenia or not.
- “There's no single test for schizophrenia and the condition is usually diagnosed after assessment by a specialist in mental health.” -NHS UK

Structural and diffusion MRI based schizophrenia classification using 2D pretrained and 3D naive Convolutional Neural Networks

Mengjiao Hu [a b], Xing Qian [b], Siwei Liu [b], Amelia Jialing Koh [b], Kang Sim [c d], Xudong Jiang [e 1], Cuntai Guan [f 1], Juan Helen Zhou [b g h i 1]

Front. Psychiatry, 03 February 2020
Sec. Schizophrenia
Volume 11 - 2020 | https://doi.org/10.3389/fpsyt.2020.00016

Identifying Schizophrenia Using Structural MRI With a Deep Learning Algorithm

Jihoon Oh[1], Baek-Lok Oh[2], Kyong-Uk Lee[3], Jeong-Ho Chae[1*] and Kyongsik Yun[4,5*]

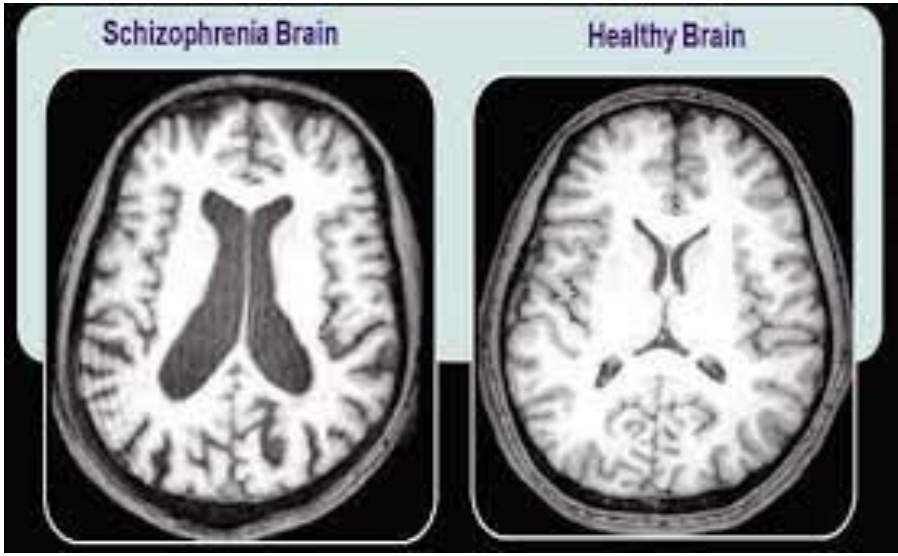
[1] Department of Psychiatry, Seoul St. Mary's Hospital, College of Medicine, The Catholic University of Korea, Seoul, South Korea

“A meta-analysis of MRI findings in schizophrenia”

M E Shenton, C C Dickey, M Frumin, R W McCarley (2001)

- **Ventricular enlargement (80%)**
- **Frontal lobe abnormalities (59%)**
- **Parietal lobe abnormalities (60%)**
- **Subcortical abnormalities**
  - **Cavum septi pellucidi (92%)**
  - **Basal ganglia (68%),**
  - **Corpus callosum (63%)**
  - Thalamus (42%)

https://pubmed.ncbi.nlm.nih.gov/11343862/



Project Roadmap

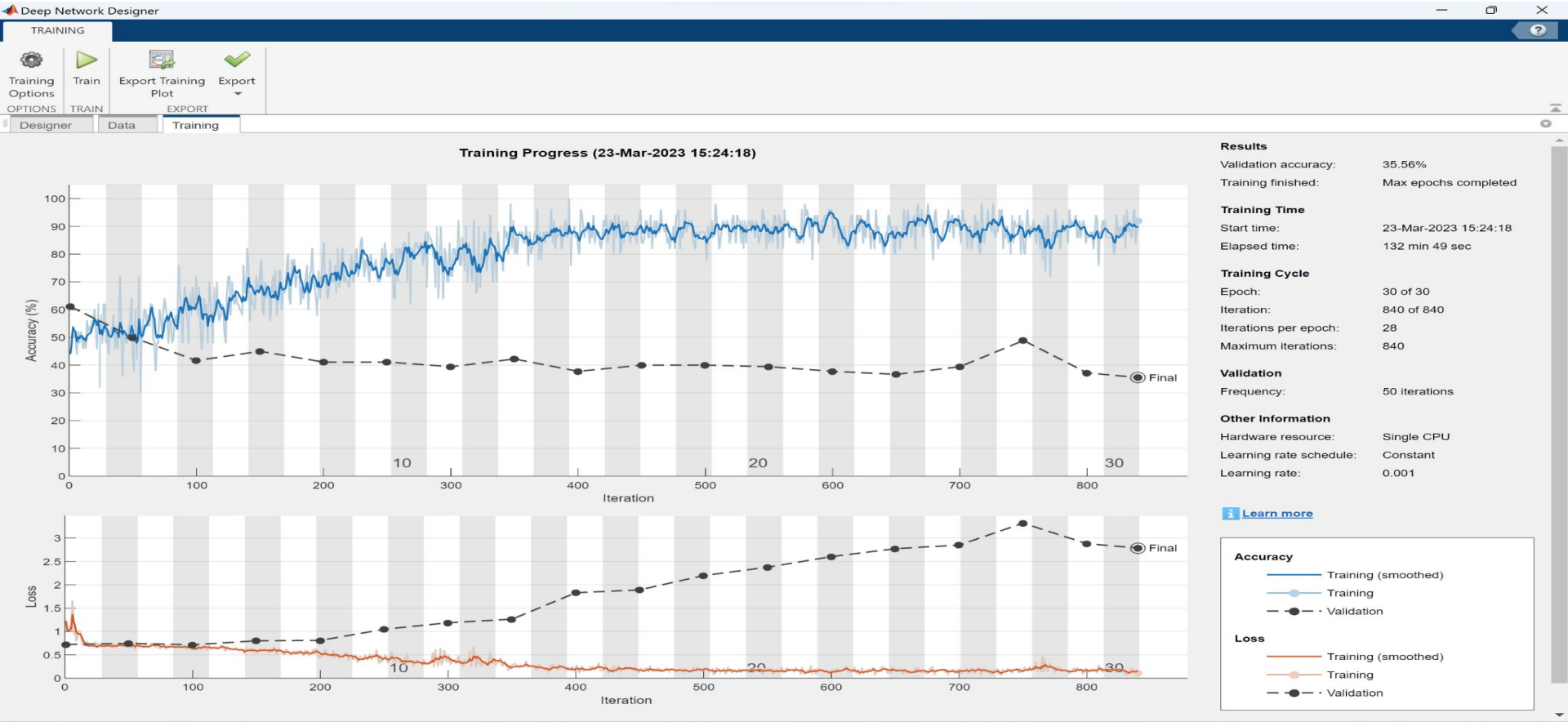
Complete:

- Taking volume of MRI and turning it into slices
  - While preserving the 16 bit MRI images as 16 bit slices
- First GoogleNet Training with 4 MRI images
- Automate Unzipping
- Individual Image min-max normalization
- Train GoogleNet with All Schizconnect Data

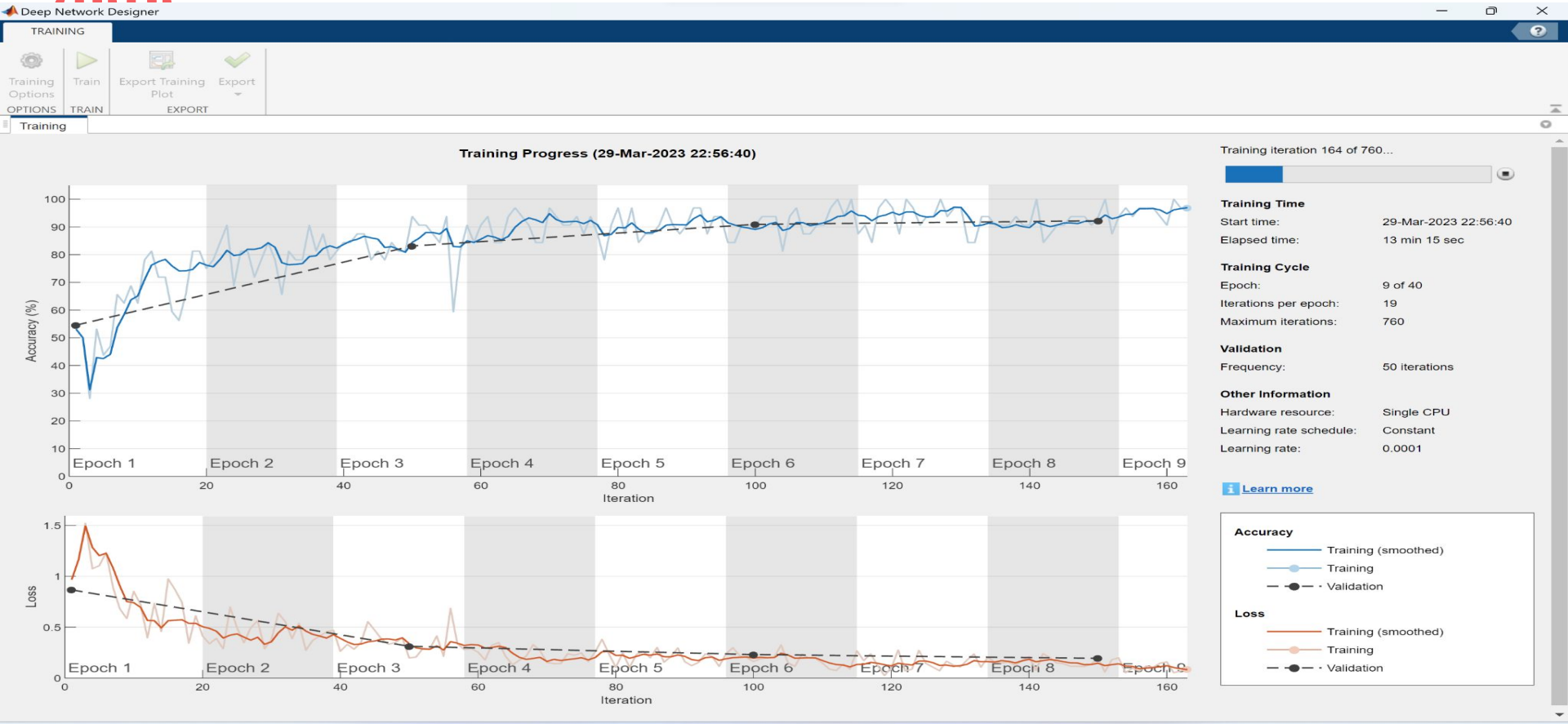
Working:

- Filtering out “Bad Slices”
  - Starting with Loosest Restriction Possible
- Fine tuning model to improve training, visualize wrongly classified images.
- Trying new Loss Functions
  - Weighted, Masked, Class Activation Map

From Overfitting to Healthier Results



2 Subjects/Group here, so now let's use all the data



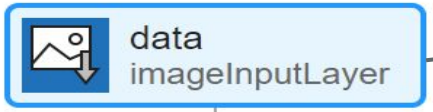
Automation of Unzipping: (.tsv) ➳ Excel ➳ Matlab

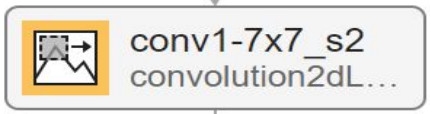
```
XelCell = readcell("MCIC Participant.xlsx"); %Loads Data
%Use cell 2 mat after indexing to actually extract data from the cell!!

%%
%Match ID with the diagnoses, and thus the folder we need to save

%Id = column 2 ; Diagnoses Col 5;
[nRows, nCol] = size(XelCell); %NumRows and columns

for i=2:nRows %loop through patients, row 1= labels so we start at row 2.
    %Below lines access cell indices we need for patient and diagnoses
    disp(i)
    thisPatientID = XelCell(i,2);
    thisDiagnoses = XelCell(i,5);
    %Below lines cell indices into characters, one more cast for string
    PatientIDtwo = cell2mat(thisPatientID);
    Diagnosestwo = cell2mat(thisDiagnoses);
    %Type cast from char to string
    stringPatient = string(PatientIDtwo);
    stringDiagnoses = string(Diagnosestwo);
```

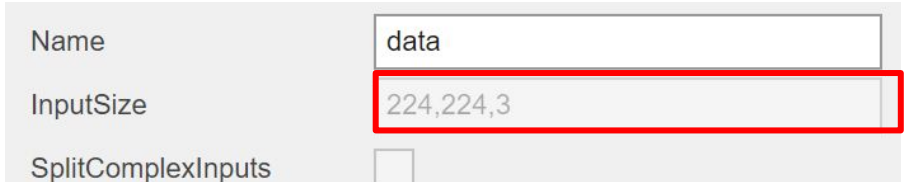
Cd, pwd, and movefile functions

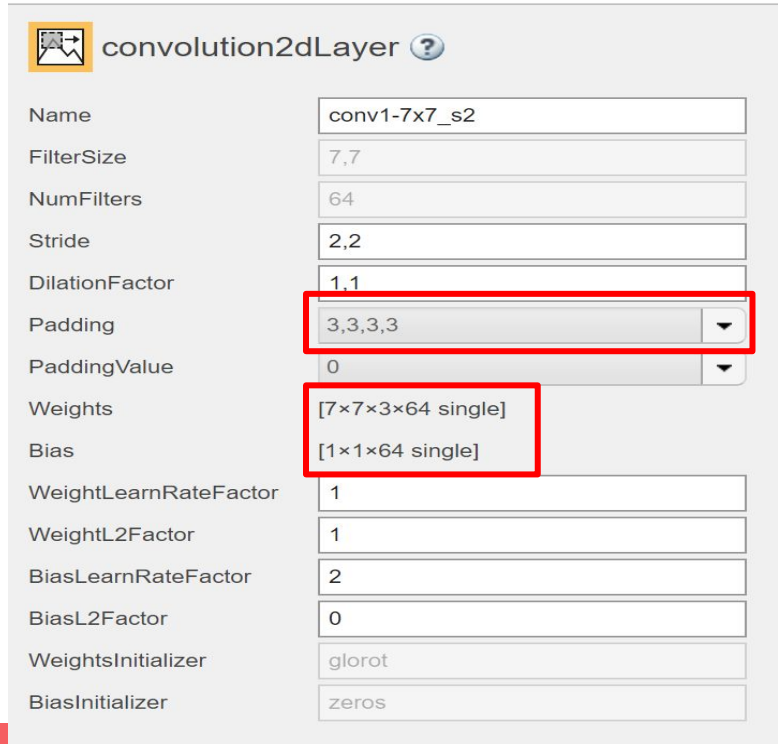
```
    %Unzipping code

    currentPath = pwd; %String of current path
    booleanTest = ~isempty( dir ( fullfile (currentPath, '*T1w.nii.gz' ) ) ); %Boolean value for if target filetype+name i
    %If contains T1w images...
    if booleanTest==1
    %Doesn't include end index
        if extractBefore(stringDiagnoses,3)=="Sc" %If schizophrenic
            movefile *T1w.nii.gz D:\AyaObtin\Project\allZips_MCIC\MCIC_SchizZip;
        elseif extractBefore(stringDiagnoses,3) == "No"%If no known disorder, not just else incase of error
            movefile *T1w.nii.gz D:\AyaObtin\Project\allZips_MCIC\MCIC_HZip;
        end
    end

%Restart folder back at original after every loop
    cd 'D:\AyaObtin\Project\Failed Unzip\MCICShare'
end
```

Turning GoogleNet to GrayScale... (A Look at the Initial Layers)

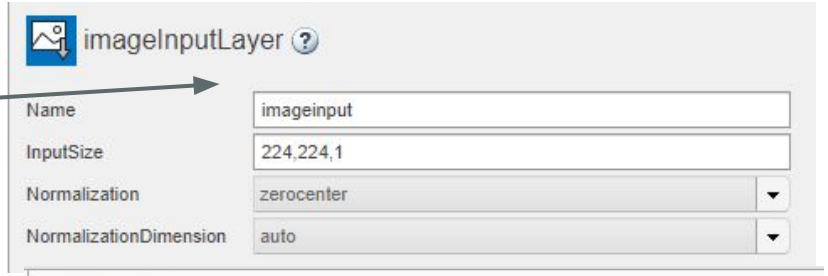


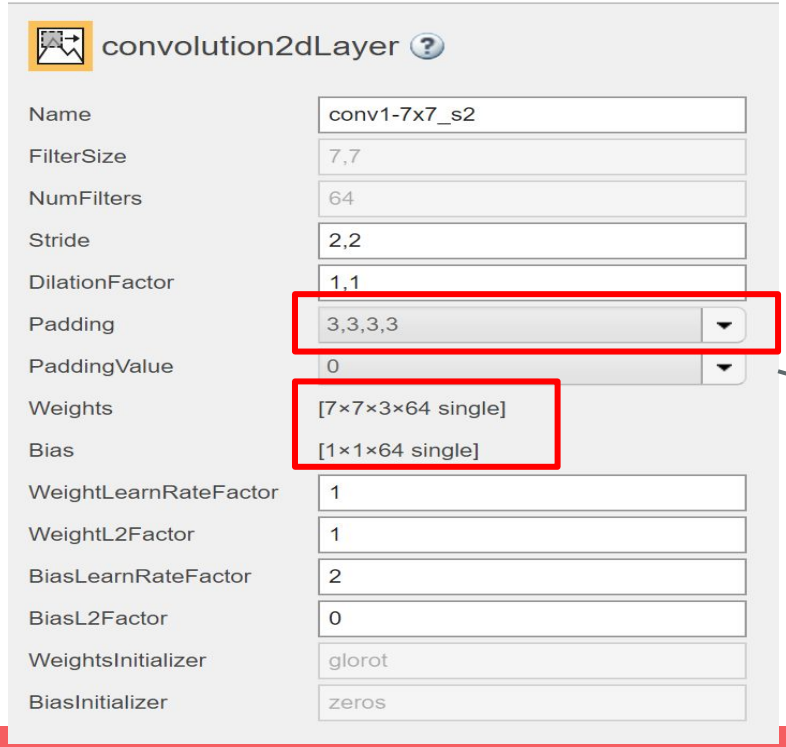


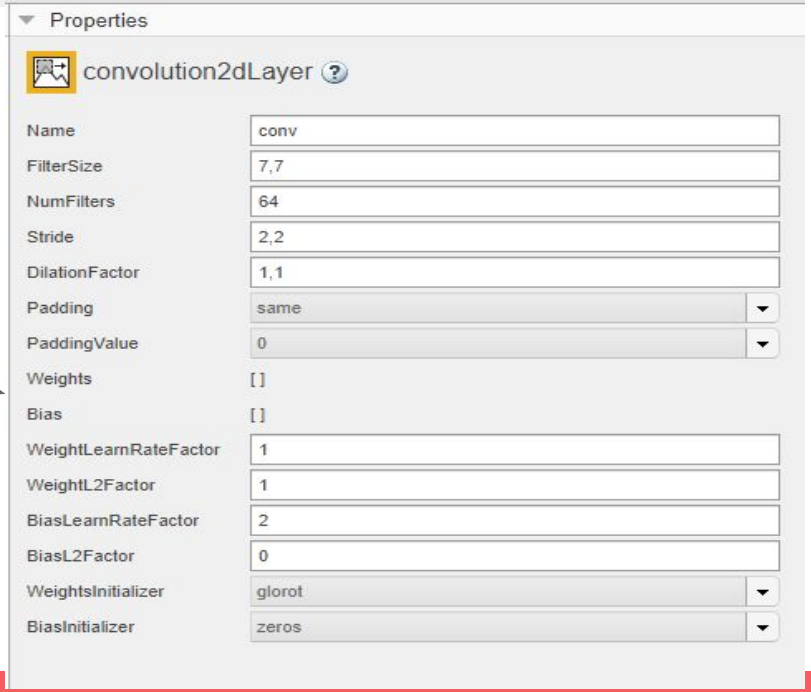




GoogleNet Before (RGB) and After (Gray):







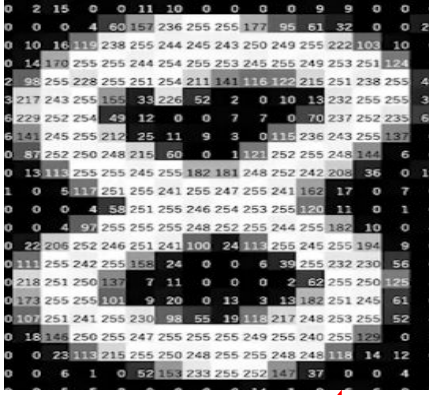
Why are “Non-Standard” file Types Weird to Work With?

```
TestniiDatastore = imageDatastore('sub-01_*','FileExtensions','.nii','ReadFcn',@(x)
niftiread(x));
testimg2 = readimage(TestniiDatastore,2); %readImage works the same as before,
(dataStorevariable,imgIndex)
thisslice = testimg2(:,:,5); % (:,:)= Every row & col pixel ; n = slice number we are
visualizing (n=5 here)

augmentedImageDatastore(testniiDatastore,'ColorPreprocessing','gray2rgb')

imshow(thisslice,[]) %We can use imshow() on .nii files with this code line found
online (thisslice,[])
```

The highlighted lines exemplify why there can be issues when training:

- **imshow(image) ~= imshow(image,[])**
- **thisSlice(:,:) ~= aug_dataStore(1)(:,:,1)**

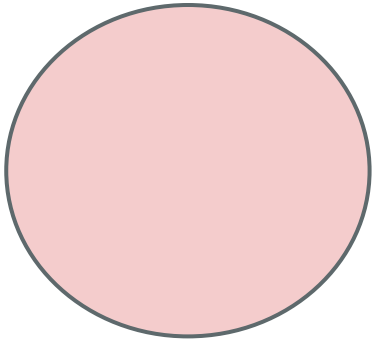
All About Pixels

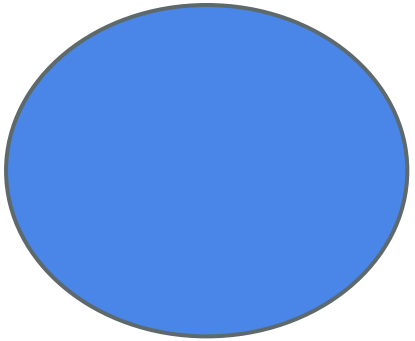


- Colored/RGB versus Gray Images?
  - Gray Images are 2D matrices, while colored images are 3D matrices
    - Gray = (Height x Width, inside = 1 grayscale value)
    - Color = (Height x Width, inside = [Red,Green,Blue])
- Bits?
  - A bit is the “color” value stored within an image (0,255,100)
  - An 8 bit image can display 16.7 million colors, while 16 bits can display up to 281 trillion colors.
    - 8 bits range from (0-255)= 2^8
    - 16 bits from (0-65536) = 2^16
    - Decimals from 0-1!!

Why Normalize Pixels?

Let's say our goal is to classify circles





For Grayscale, we have intensity; In RGB, we have luminance

Pixel Normalization and MRI-Specific Reasons for it

- Different pulse sequences and scan parameters give different results
- Different scanners could have been used as a whole
- Different environments/rooms for different days/machines
- Pixel Intensity is *NOT* a important feature in the first place!

The Two Types of Normalization:

- Sample Based vs. Individual Image Based

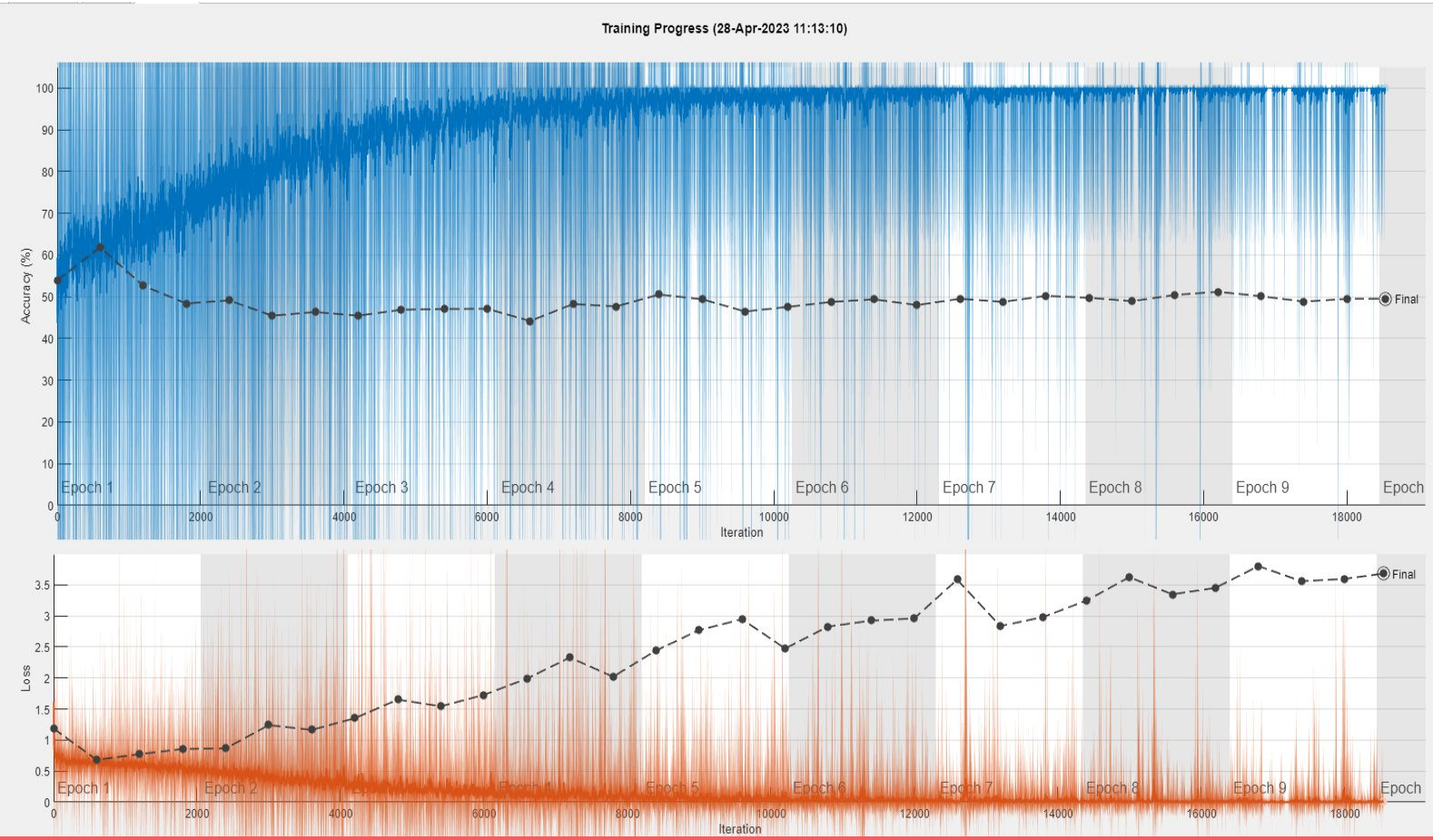
$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Min-Max Image Based Normalization:

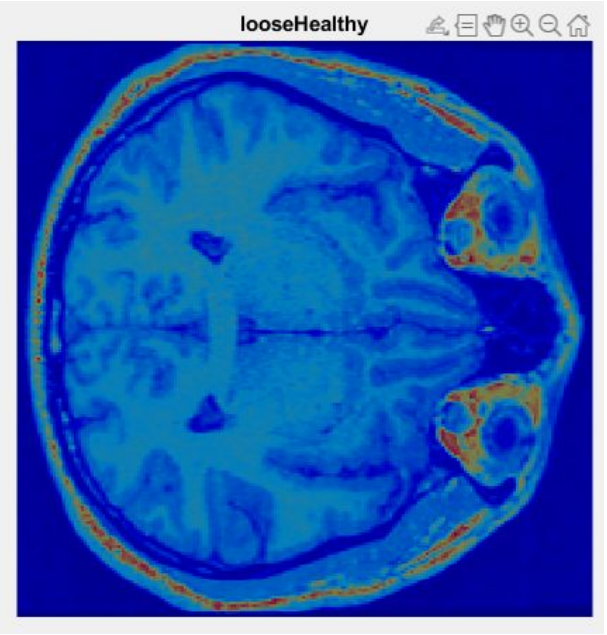
```
1
2   vol = niftiread("sub-A00018403_ses-20110101_acq-mprage_run-01_echo-02_T1w.nii"); %Load volume
3   signSlice = vol(:,:,70); %Take one slice from volume
4
5   blankzero = int16(NaN(size(signSlice,1),size(signSlice,2))); %Create a blank matrix for typecasting slices
6   blankzero = signSlice; %Set all pixels in blank to our slice
7   doubleCopy=double(blankzero); %typecast original blank to a double for .tif saving
8
9   Vmax = max(doubleCopy,[],"all"); %Max value
10  Vmin = min(doubleCopy,[],"all"); %min value
11  primeImage = (doubleCopy-Vmin)./(Vmax-Vmin); %normalize all pixels in slice between 0-1
12
```

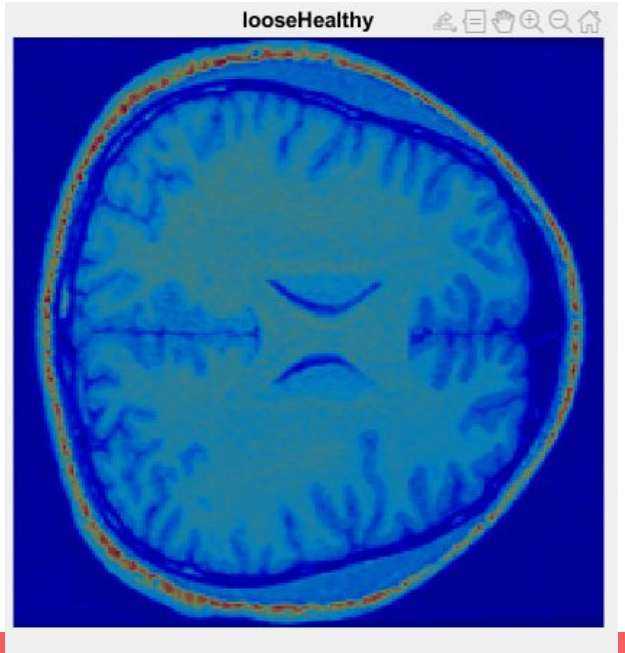
- Another Syntax for finding Max&Min pixels (should be 1 int):
  - `min(signSlice_final(:))`
  - `max(signSlice_final(:))`

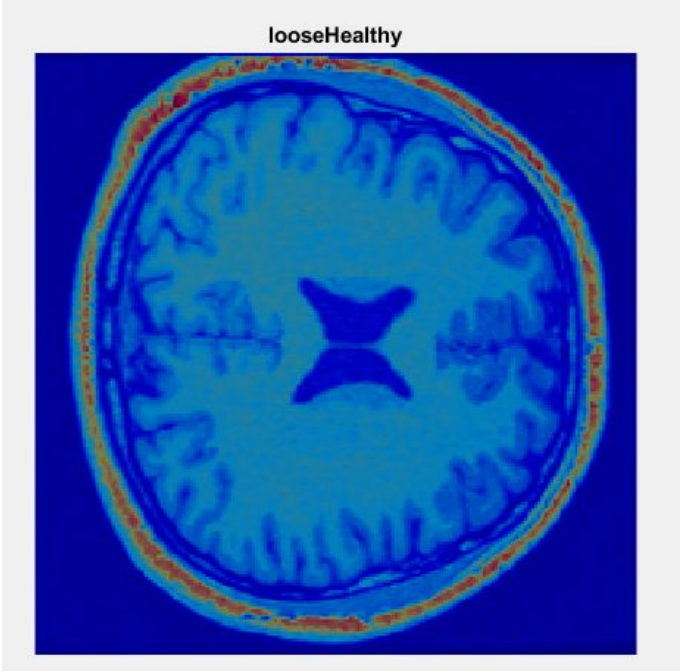
Overfitting Again... Why?



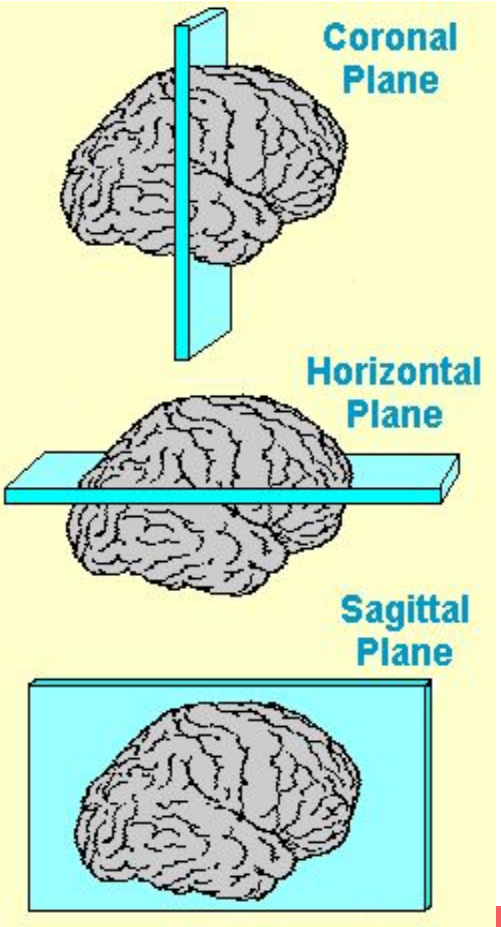
The Data Says Why!







Slice Orientation



—> Fixed y axis per slice; volume(: , i , :)

—> Fixed z axis per slice; volume(: , : , i)

—> Fixed x axis per slice; volume (i , : , :)

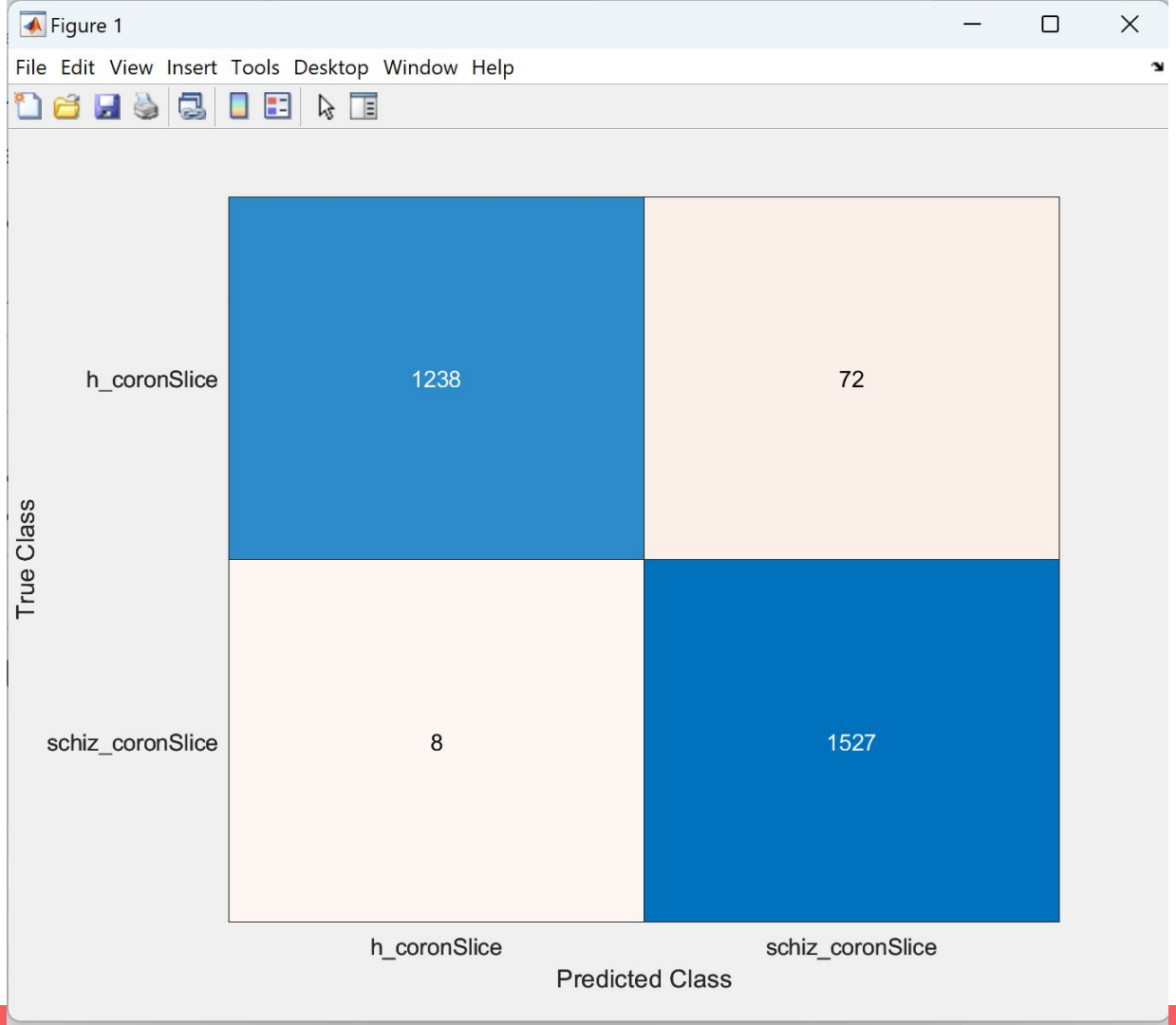
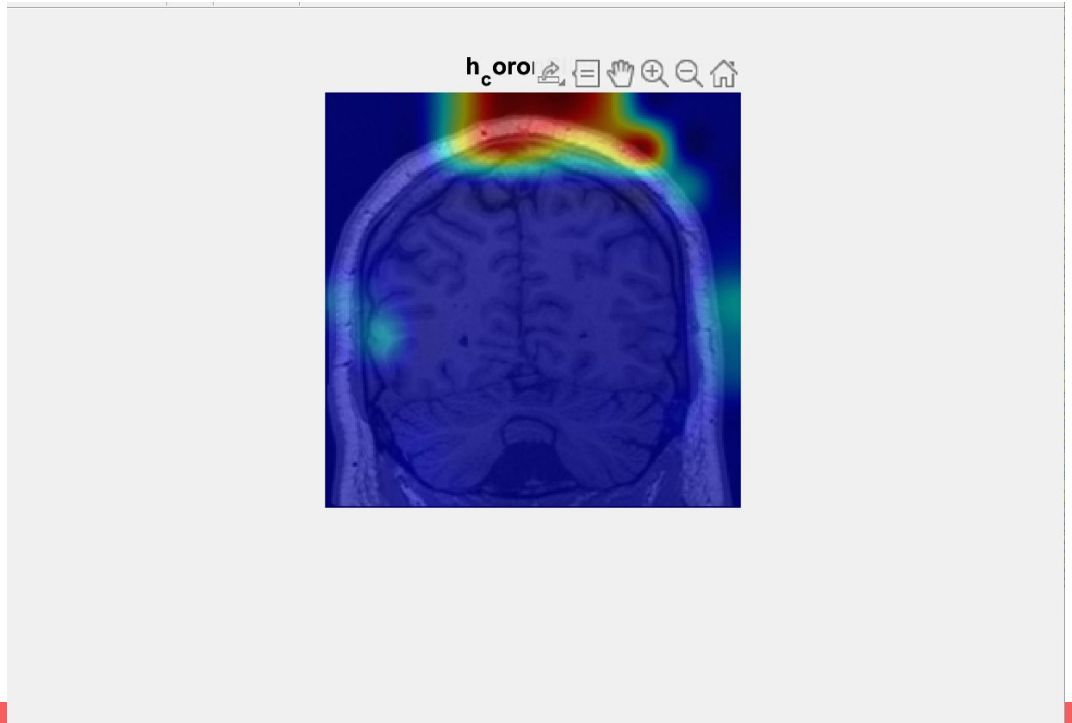
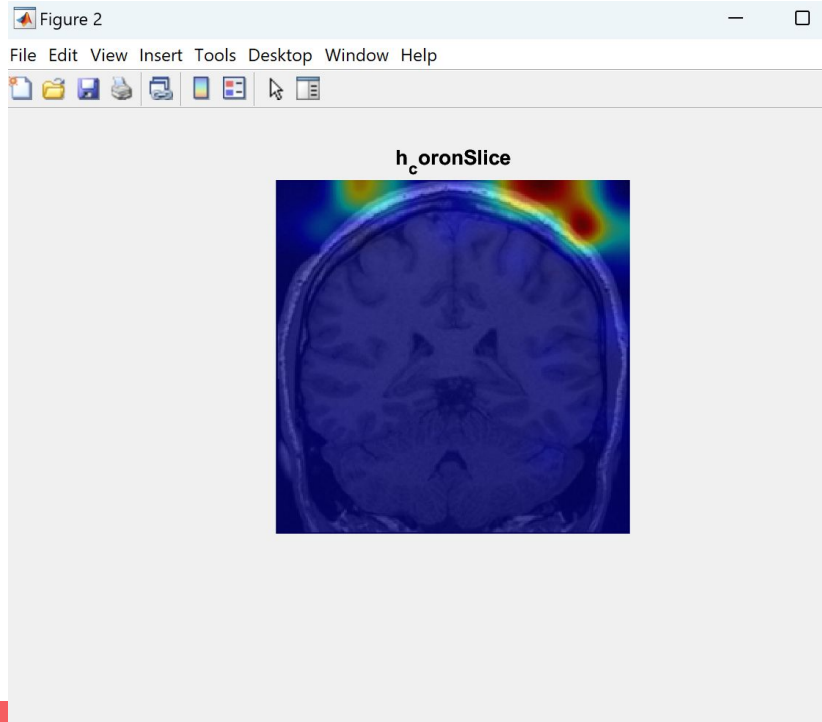


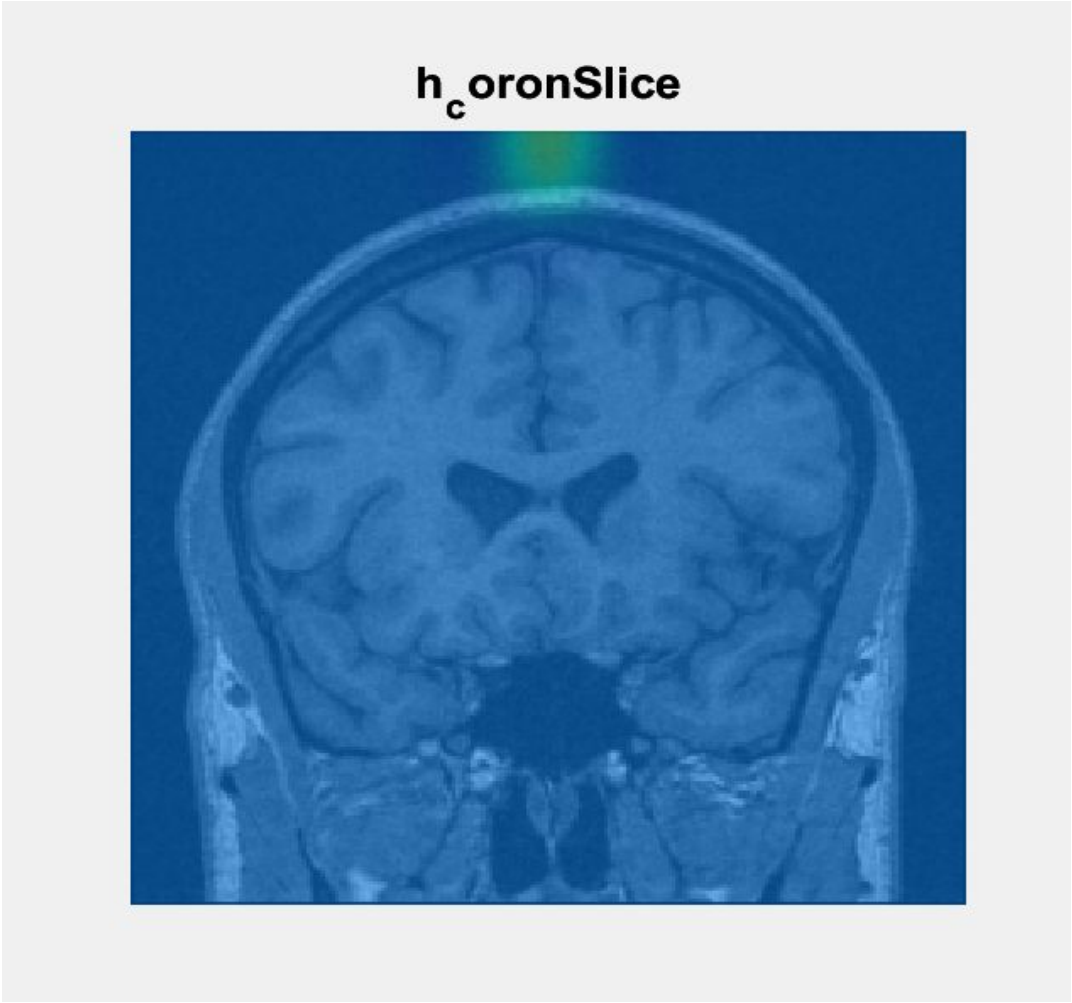
Figure 1
File Edit View Insert Tools Desktop Window Help
h_coronSlice
schiz_coronSlice
True Class
1238
72
8
1527
h_coronSlice
schiz_coronSlice
Predicted Class

Wrongly Classified Images

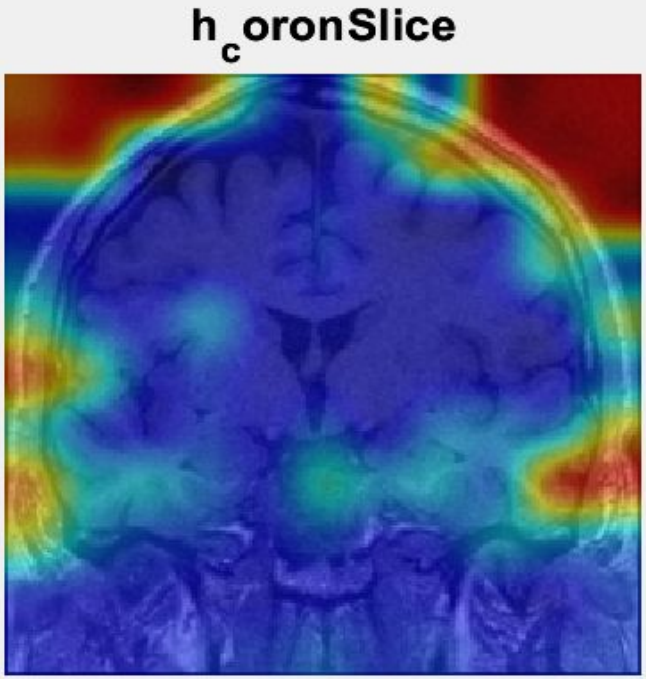


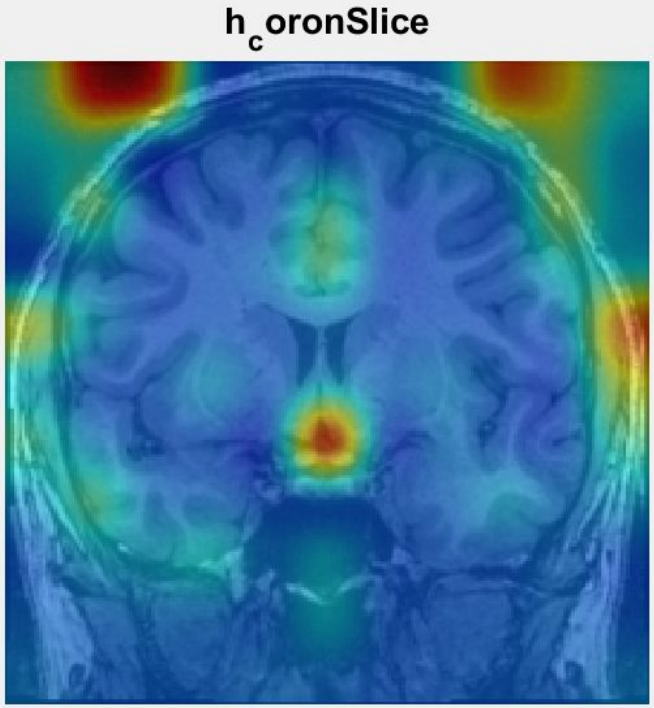


Incorrect

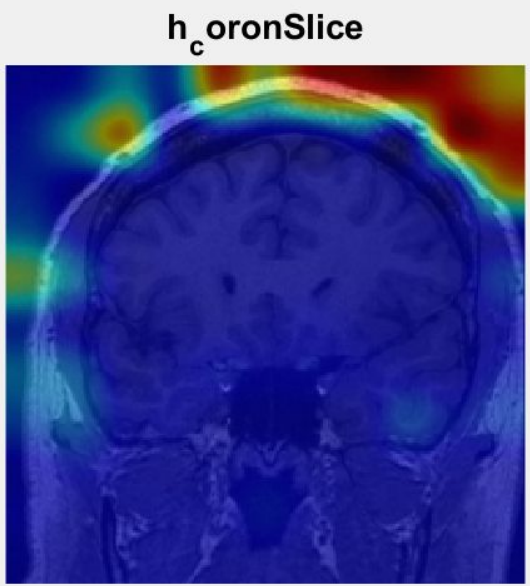


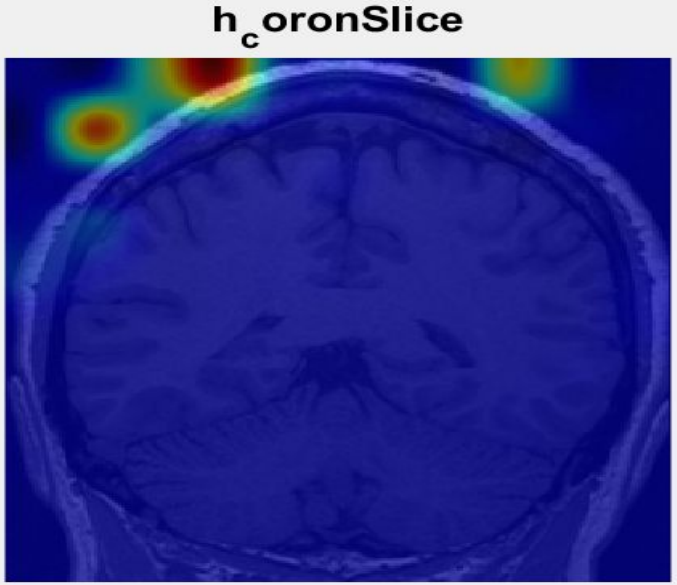
Correctly Classified:



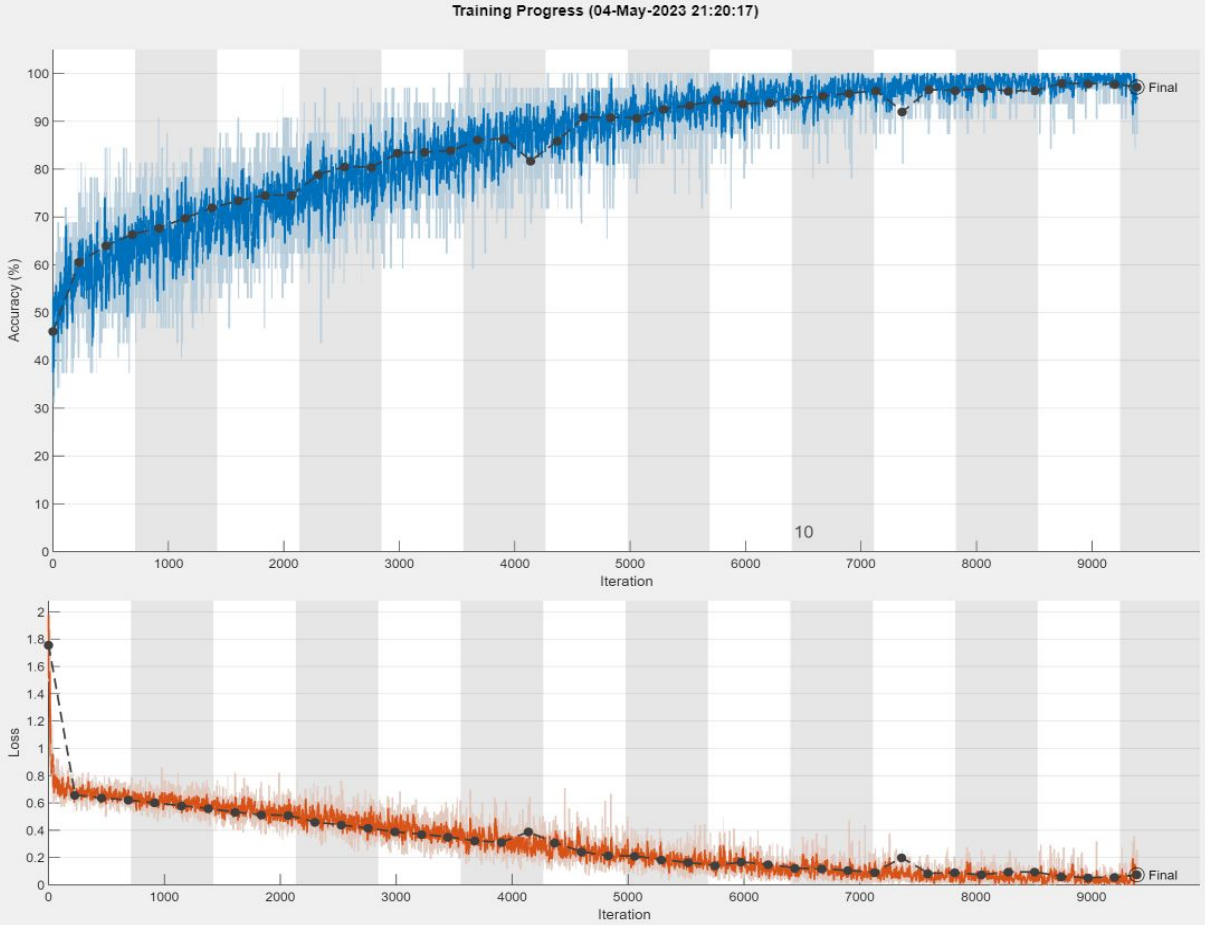


Correctly Classified:





Healthier Results



Results

Validation accuracy:	97.15%
Training finished:	Stopped manually

Training Time

Start time:	04-May-2023 21:20:17
Elapsed time:	669 min 30 sec

Training Cycle

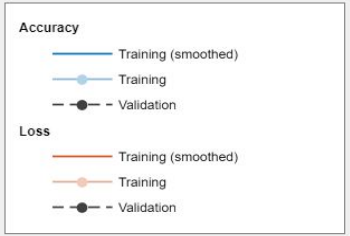
Epoch:	14 of 30
Iteration:	9391 of 21330
Iterations per epoch:	711
Maximum iterations:	21330

Validation

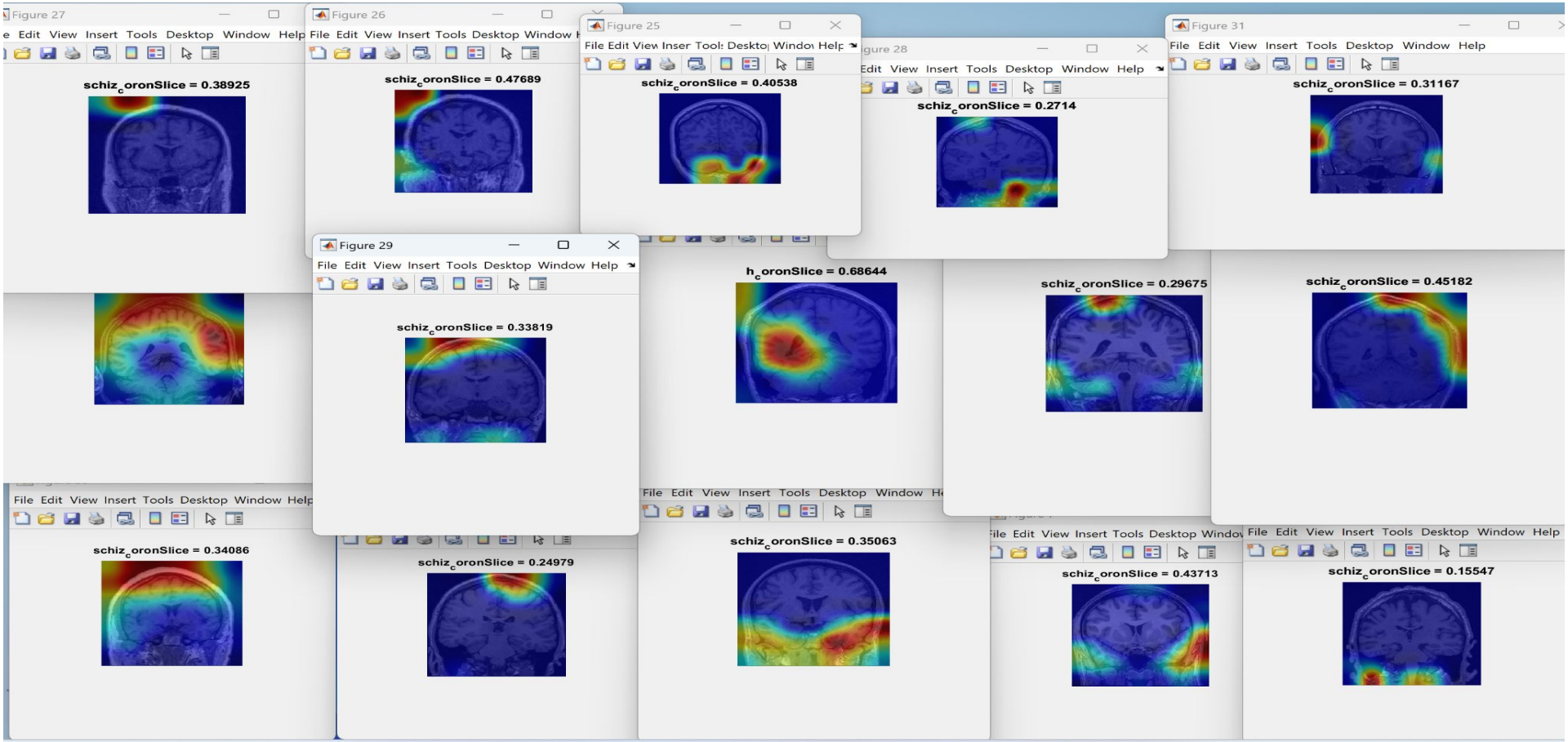
Frequency:	230 iterations

Other Information

Hardware resource:	Single CPU
Learning rate schedule:	Constant
Learning rate:	0.0001



GradCAM is meaningful... Wrong im (Title = Pred, score)



First 25 Correct GradCAM

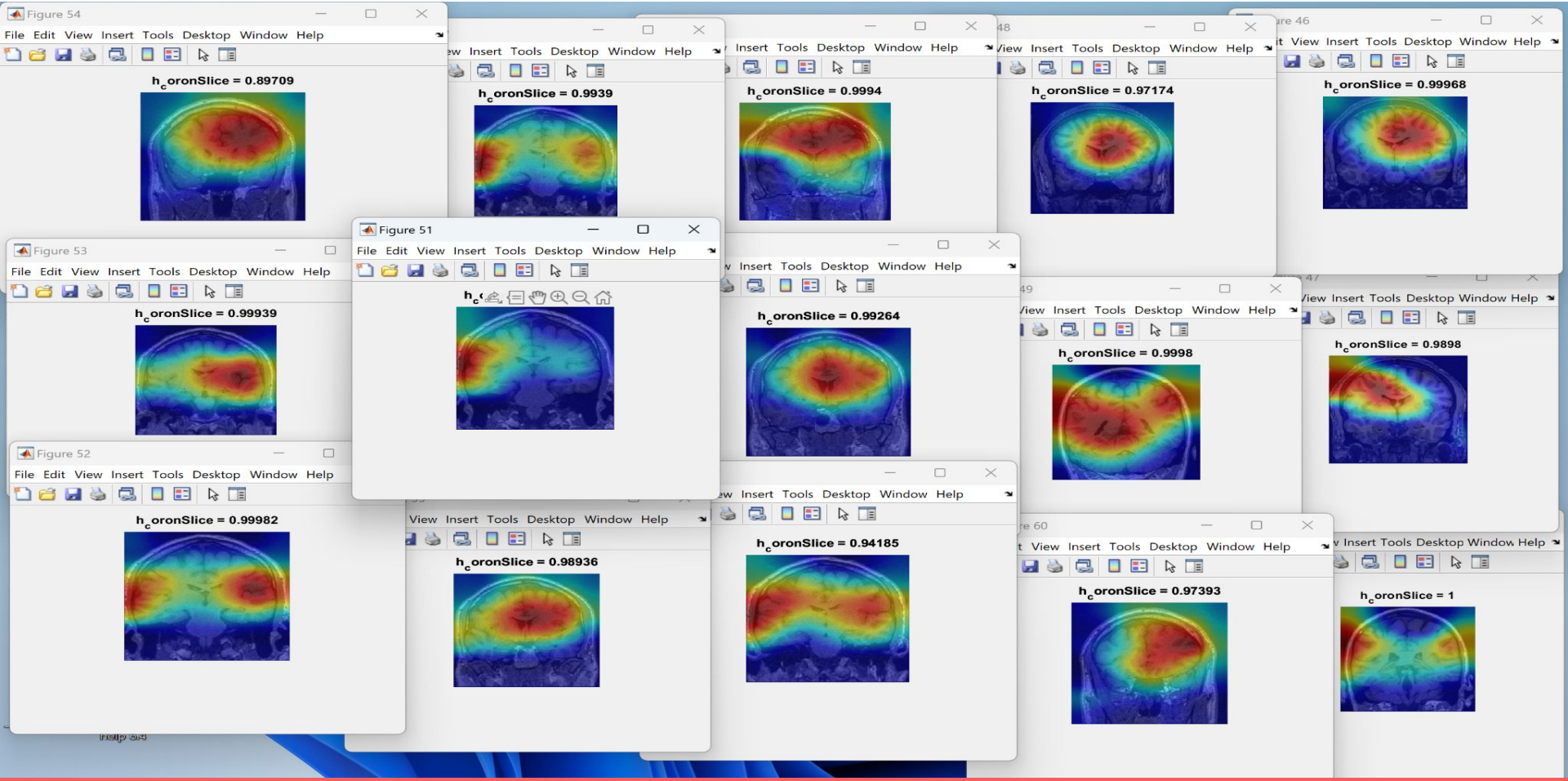
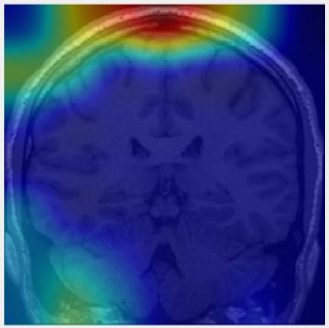


Figure 3

File Edit View Insert Tools Desktop Window Help

$h_{coronSlice} = 0.074332$



Tools Desktop Window Help

$h_{coronSlice} = 0.26118$

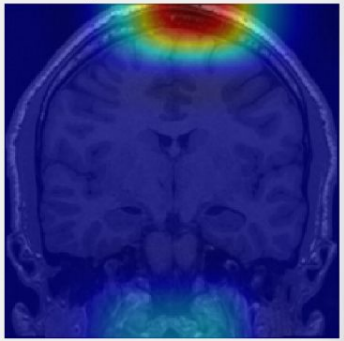
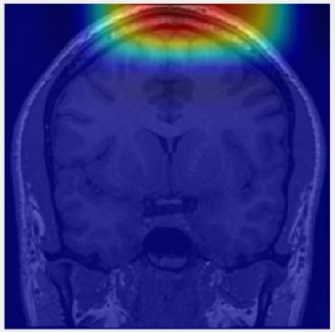


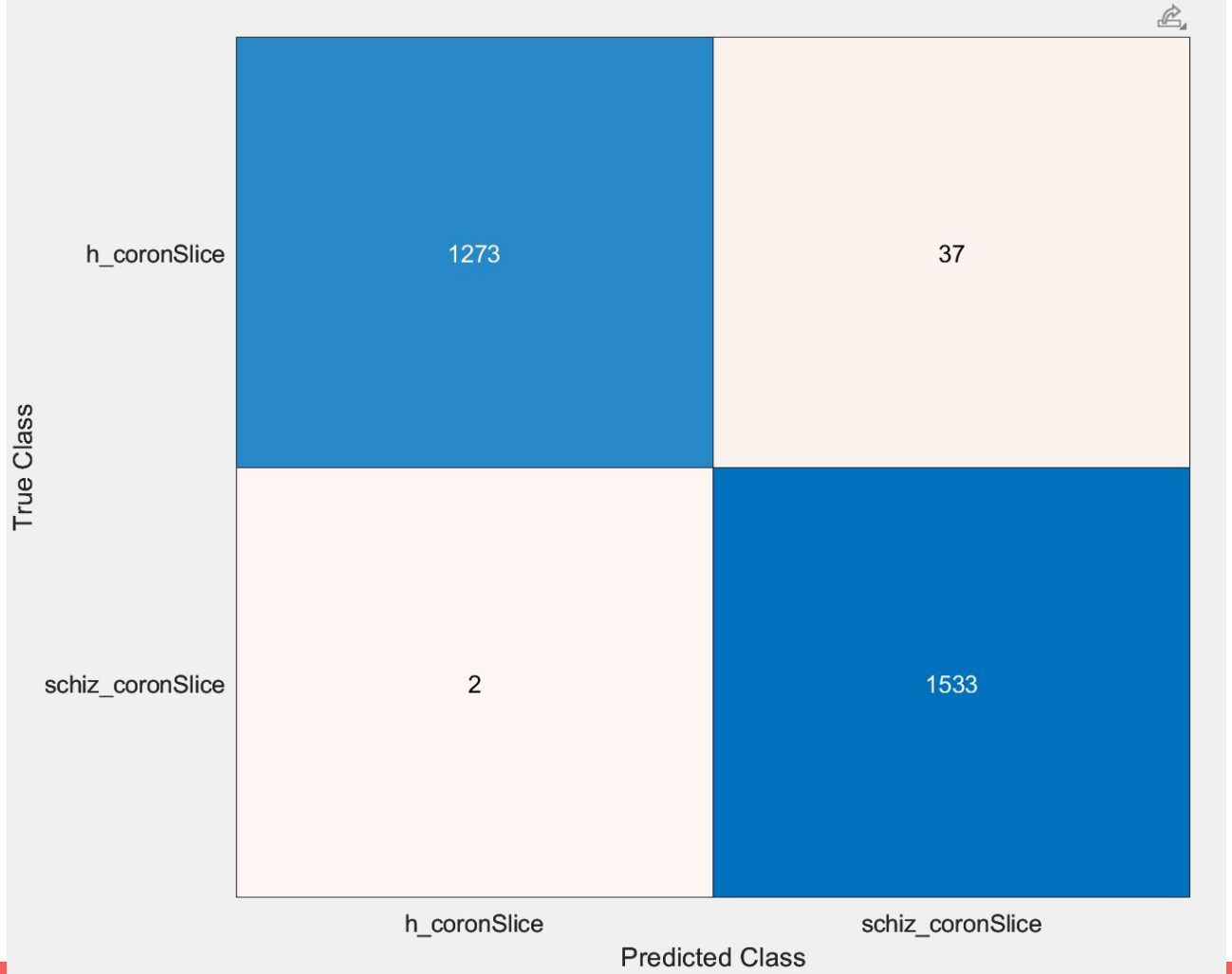
Figure 5

File Edit View Insert Tools Desktop Window Help

$h_{coronSlice} = 0.20569$



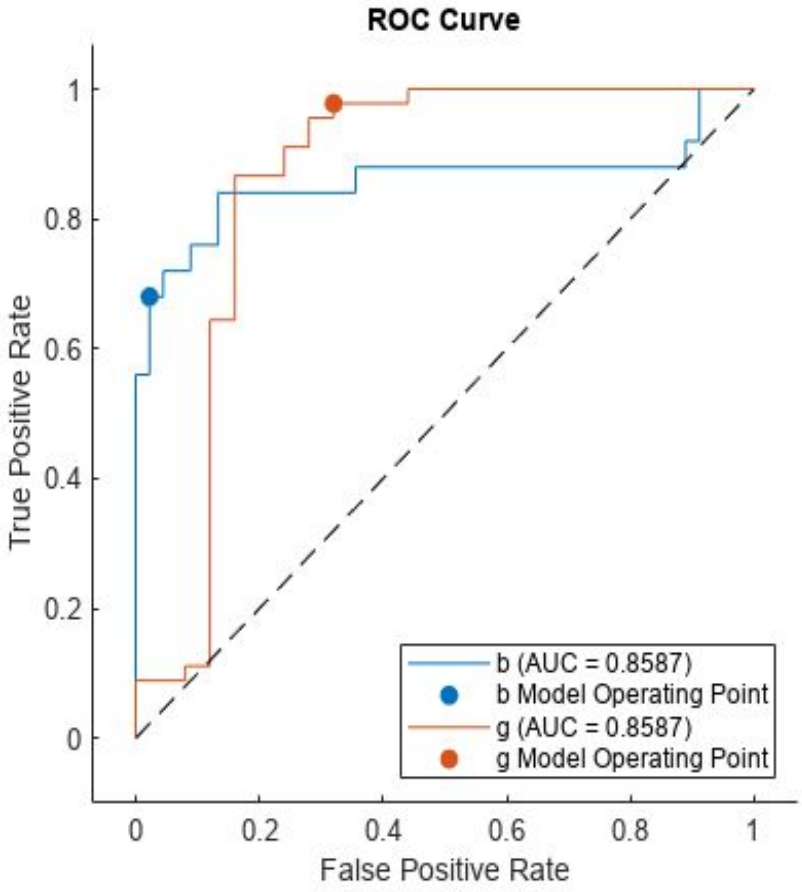
		Predicted	
		Negative (**N**) **-**	Positive (**P**) **+**
Actual	Negative **-**	True Negatives (T**N**)	False Positives (F**P**) **Type I error**
	Positive **+**	False Negatives (F**N**) **Type II error**	True Positives (T**P**)


h_coronSlice
schiz_coronSlice
True Class
1273
37
2
1533
h_coronSlice
schiz_coronSlice
Predicted Class

Ways to Quantify Classification Accuracy

- **Recall/Sensitivity**:
  - True-pred-Schiz / (True Schiz + False-Pred-Healthy)
- **True Positive Rate (Precision):**
  - True Schiz / (True Schiz + False-Pred-Schiz)
- **True Negative Rate (Specificity):**
  - True Healthy / (True Healthy + False-Pred-Schiz)
- **ROC curve (AUC)**
  - Threshold point: Value along X axis of ROC curve (False Positive)

We'll definitely see AUC in the papers we read, and most also show a table of sensitivity and specificity.



Outside of ROC Curves: F1 score (Binary Classification)

$$F_1 = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} = \frac{\mathrm{TP}}{\mathrm{TP} + \frac{1}{2}(\mathrm{FP} + \mathrm{FN})}$$

TP = number of true positives

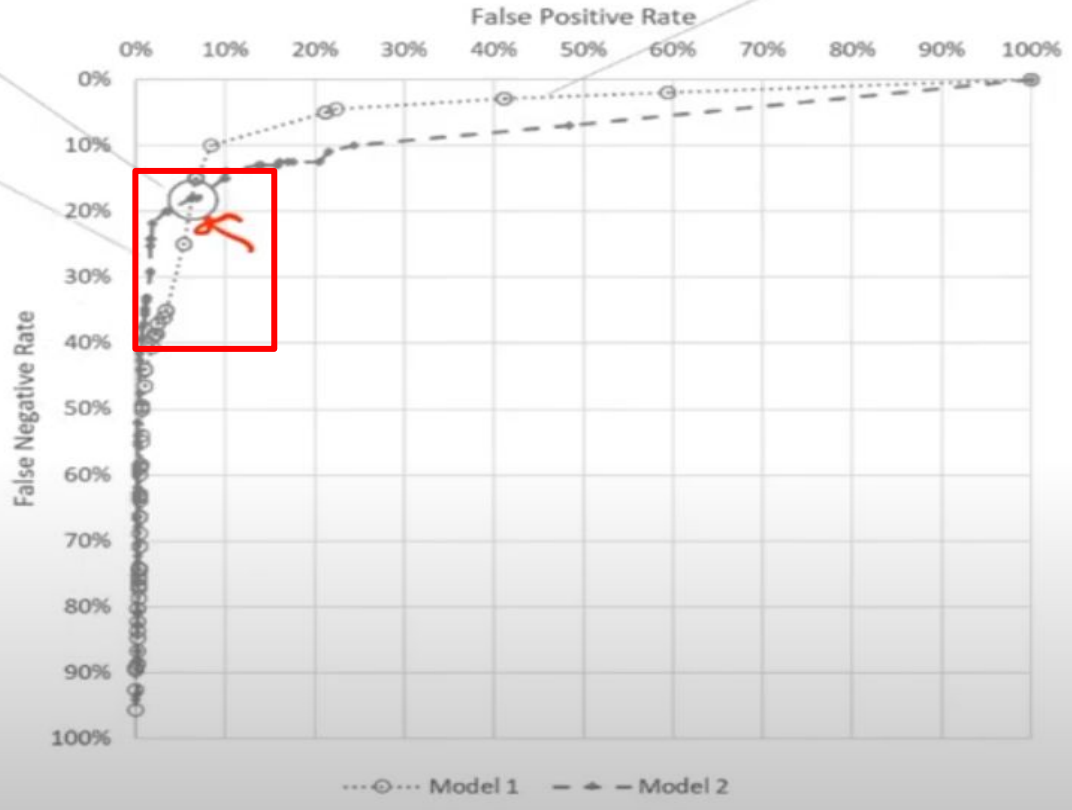
FP = number of false positives

FN = number of false negatives

ROC Curves Cross

Model 1 better at low FNR

Model 2 better at low FPR



- M1 AUC = 97
- M2 AUC = 94

How Accuracy Metrics can Look in a Paper:

SP = Specificity

SE = Sensitivity

AUC = Area under ROC Curve (range: 0-1)

Table 3. Single modal (sMRI) independent testing results on the IMH dataset.

		ACC	SP	SE	ROC-AUC
Handcrafted feature-based machine learning	Linear SVM	54.02%	85.53%	37.84%	0.72
	Nonlinear SVM (RBF kernel)	42.41%	97.37%	14.19%	0.73
Deep feature based on pretrained 2D CNN	VGG16	66.52%	59.21%	70.27%	0.71
	Xception	65.62%	36.84%	80.41%	0.70
	Resnet101	64.73%	69.74%	62.16%	0.73
	Densenet121	70.09%	42.11%	84.46%	0.73
	Inception_V3	67.86%	36.84%	83.78%	0.71
	Inception_resnet_V2	65.18%	48.68%	73.65%	0.70
Naive 3D CNN models	Sequential_1	62.95%	75.00%	56.76%	0.73
	Sequential_2	54.46%	93.42%	34.46%	0.72
	Sequential_3	70.09%	46.05%	82.43%	0.72
	Inception_1	68.30%	68.42%	68.24%	0.74
	Inception_2	66.96%	65.79%	67.57%	0.71
	Inception_resnet_1	**70.98%**	63.16%	75.00%	0.75
	Inception_resnet_2	66.96%	72.37%	64.19%	0.61

Fold N0	00	01	02	03	04	Average
Accuracy	94.59%	93.04%	95.12%	95.01%	94.11%	94.37%

Table 3: The accuracy of our model in each experiment and the average accuracy of all five experiments.

$$\text{Accuracy} = \frac{TP}{TP+FN}.$$

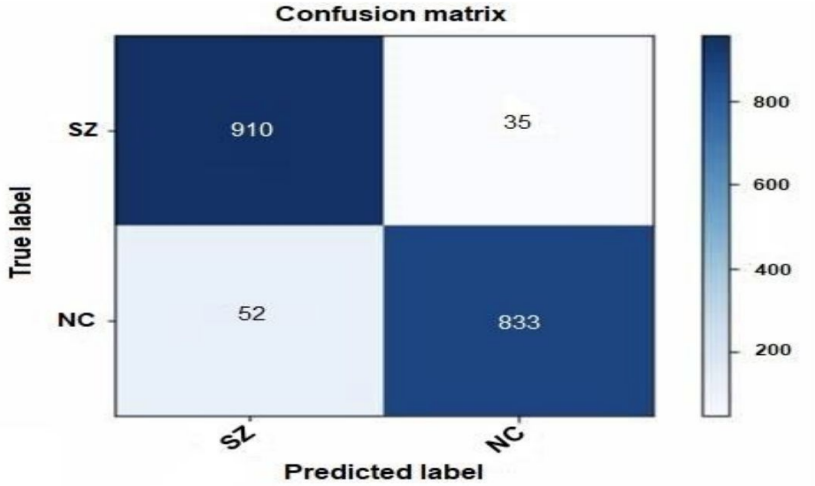


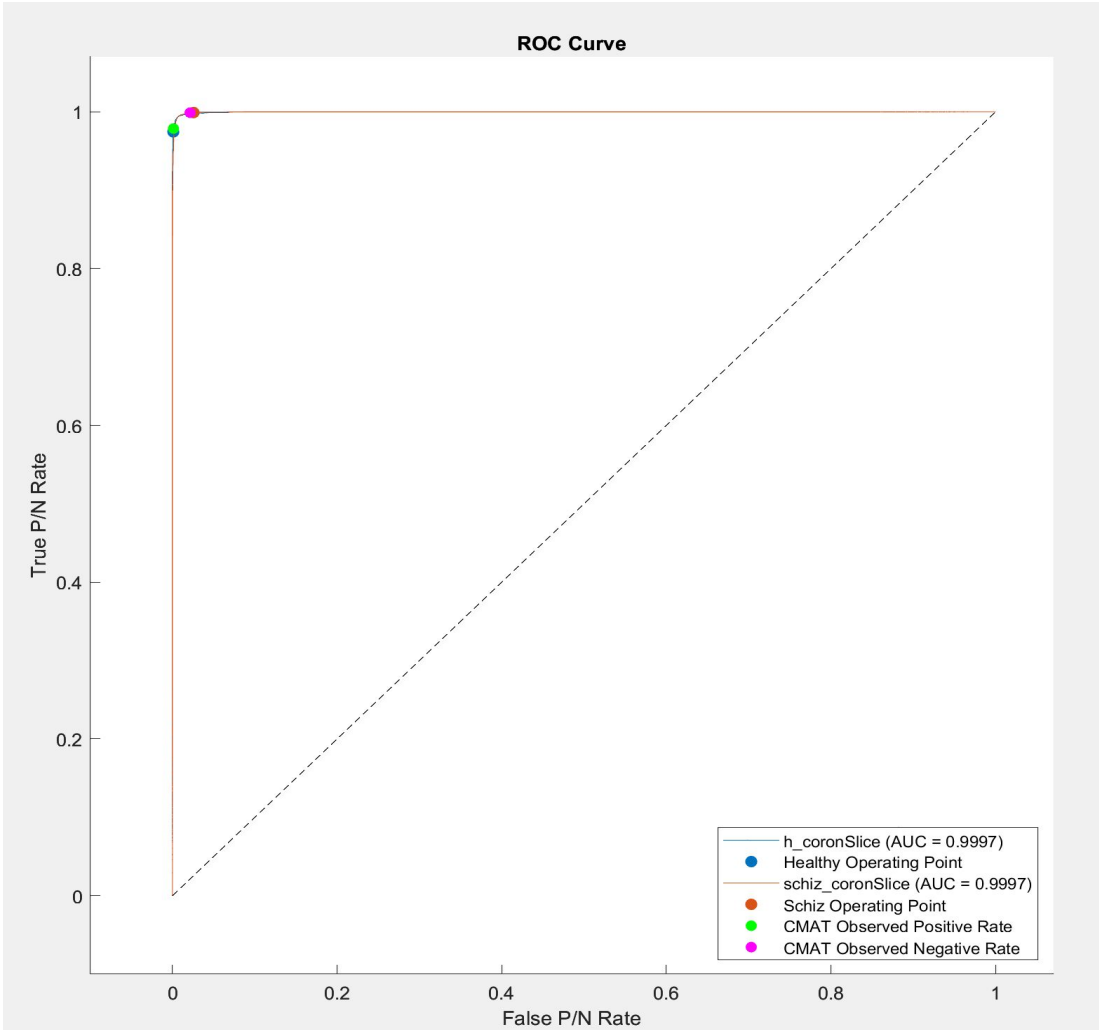
Figure 17: The confusion matrix for testing 1830 images in our trained model

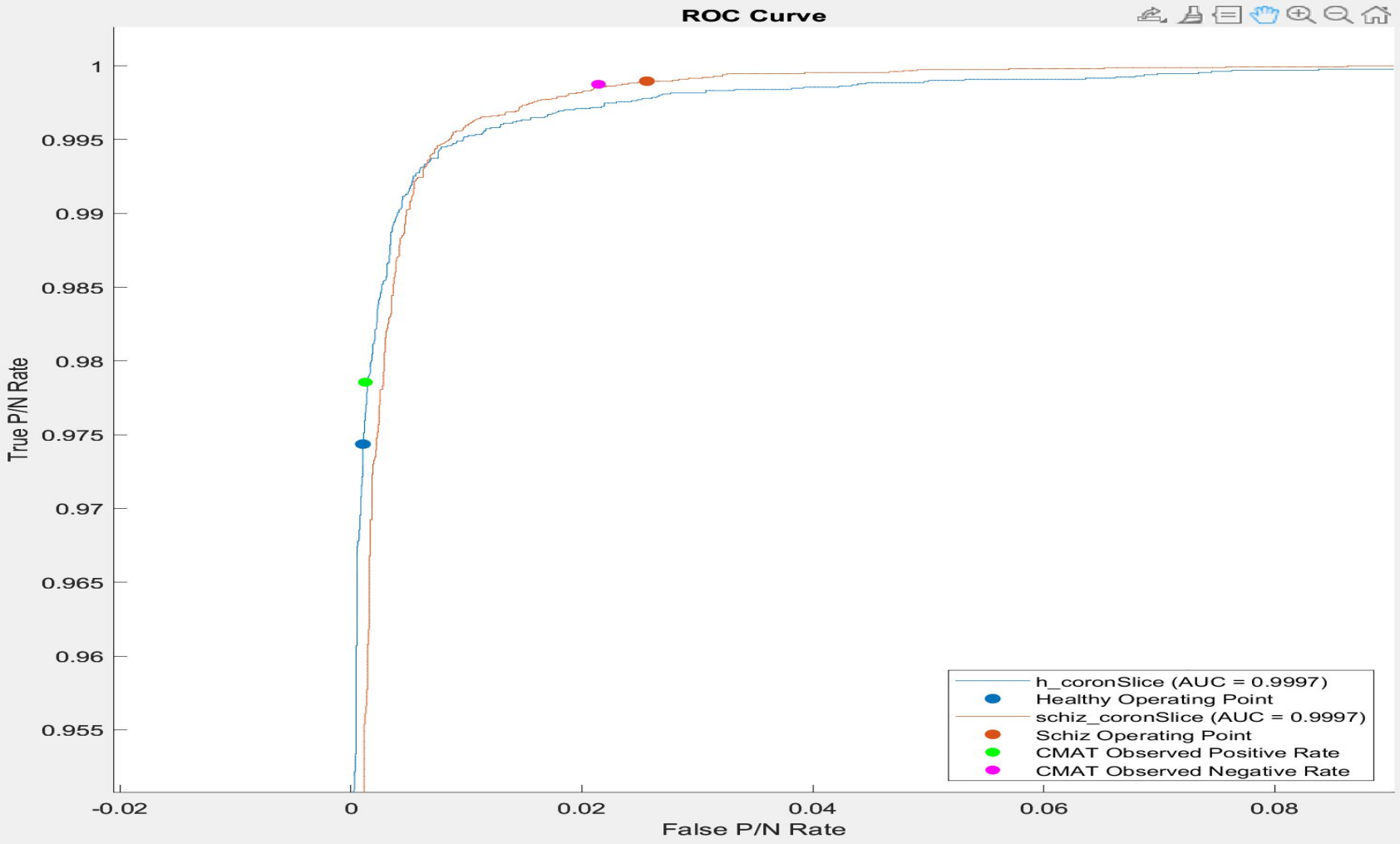
Reference	Dataset Modality	Method	SZvsNC
Patel et al. [9]	MRI	SAE+SVM	88.4%
Lu et al.[47]	fMRI	SAE+SVM	90%
Zeng et al. [13]	MRI+fMRI	SAE+SVM	85%
Cetin et al.[11]	MEG+fMRI	SVM	86.86%
Cabral et al.[48]	fMRI	R-CNN	57.89%
Yan et al.[10]	fMRI	FNC	82%
Vu et al. [36]	fMRI	DL-CNN	87.11%
Our Experiment	fMRI	DL-CNN	94.37%

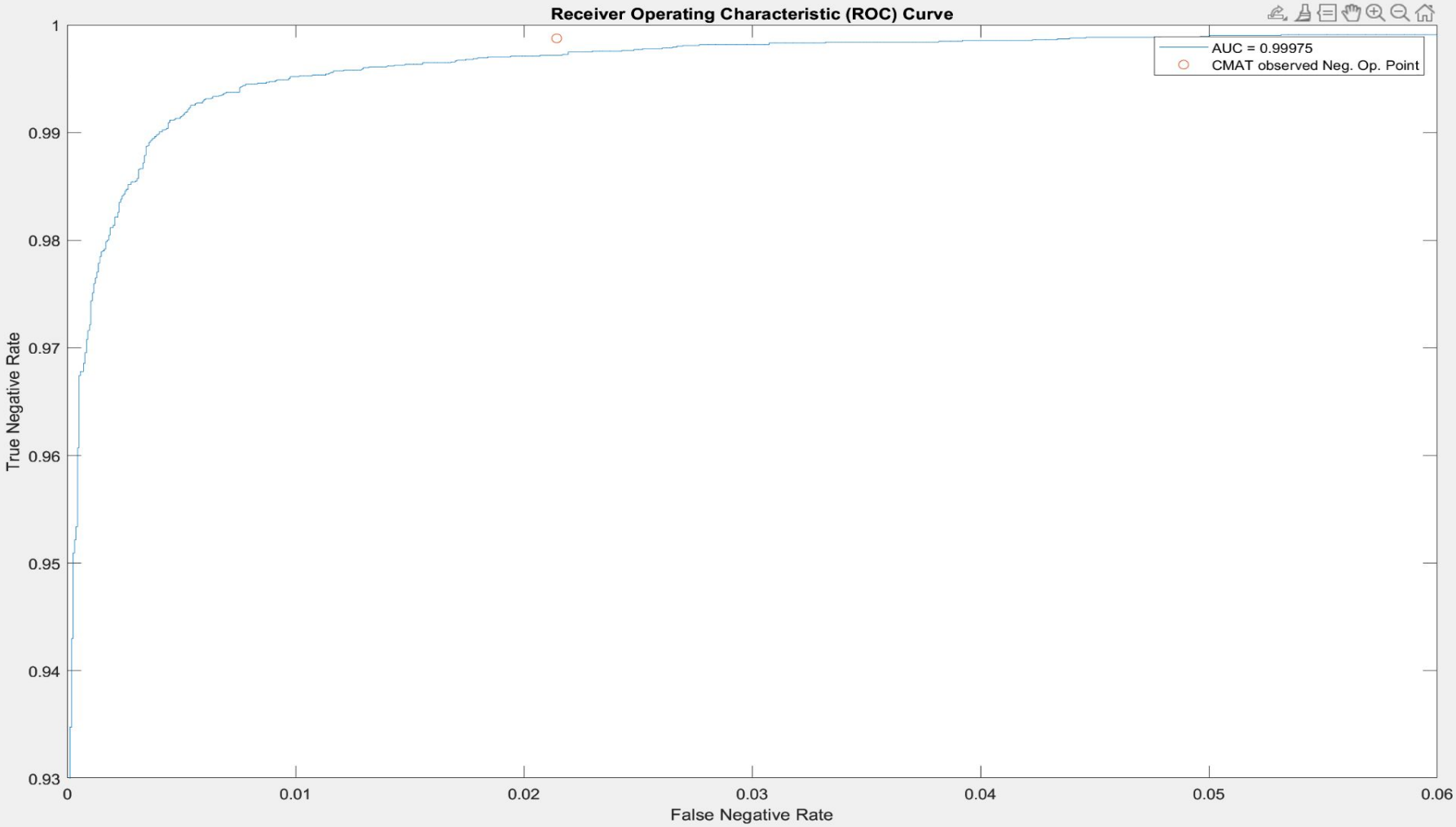
Table 4: Based on the accuracy of the model, our results are compared with previous work.

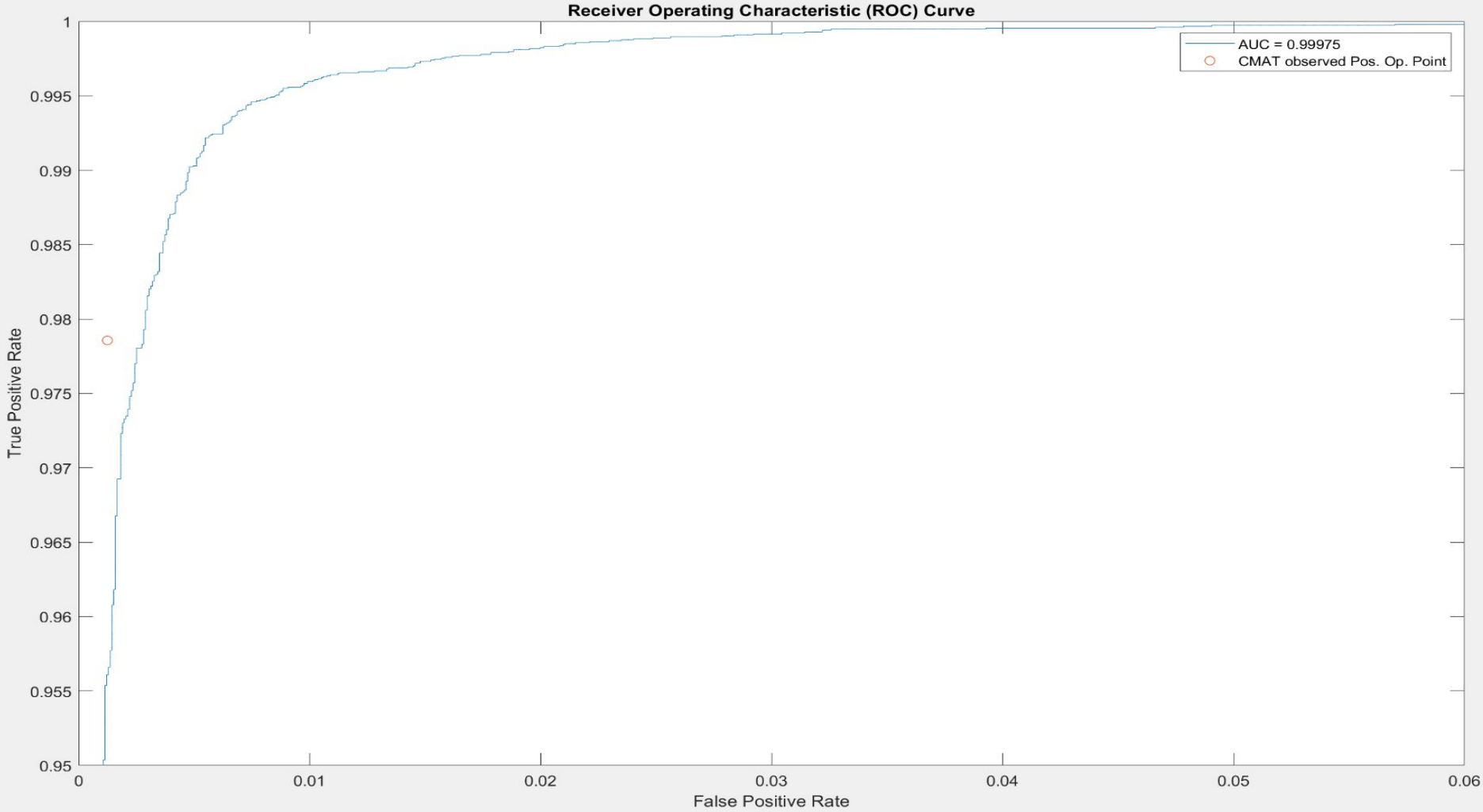
Why use “ROCMetrics()” over “PerfCurve()”?

- **Observed Confusion Matrix Rates Actually Fall on the ROCMetrics Curve, but they don’t on a PerfCurve ROC**
- Thresholding Pattern Used Avoids “Double Classification Problem”




ROC Curve
True P/N Rate
False P/N Rate
1
0.995
0.99
0.985
0.98
0.975
0.97
0.965
0.96
0.955
-0.02
0
0.02
0.04
0.06
0.08
h_coronSlice (AUC = 0.9997)
Healthy Operating Point
schiz_coronSlice (AUC = 0.9997)
Schiz Operating Point
CMAT Observed Positive Rate
CMAT Observed Negative Rate


Receiver Operating Characteristic (ROC) Curve
AUC = 0.99975
CMAT observed Neg. Op. Point
True Negative Rate
False Negative Rate
1
0.99
0.98
0.97
0.96
0.95
0.94
0.93
0
0.01
0.02
0.03
0.04
0.05
0.06


Receiver Operating Characteristic (ROC) Curve
AUC = 0.99975
CMAT observed Pos. Op. Point
True Positive Rate
False Positive Rate
1
0.995
0.99
0.985
0.98
0.975
0.97
0.965
0.96
0.955
0.95
0
0.01
0.02
0.03
0.04
0.05
0.06

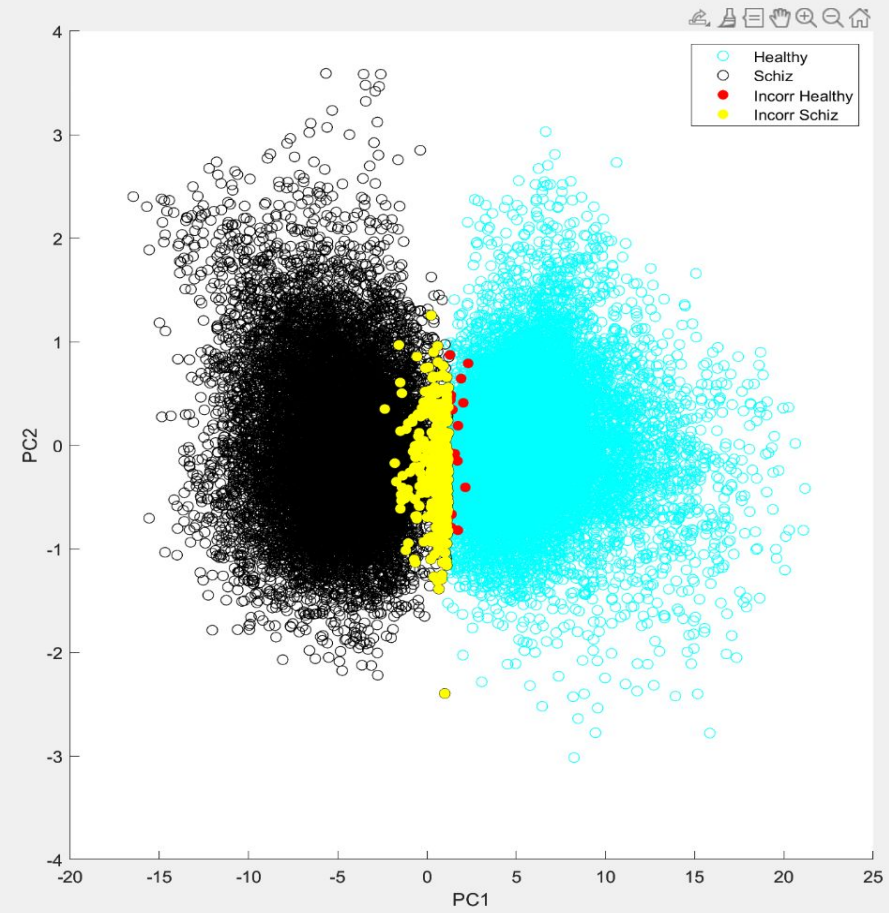
metrics =

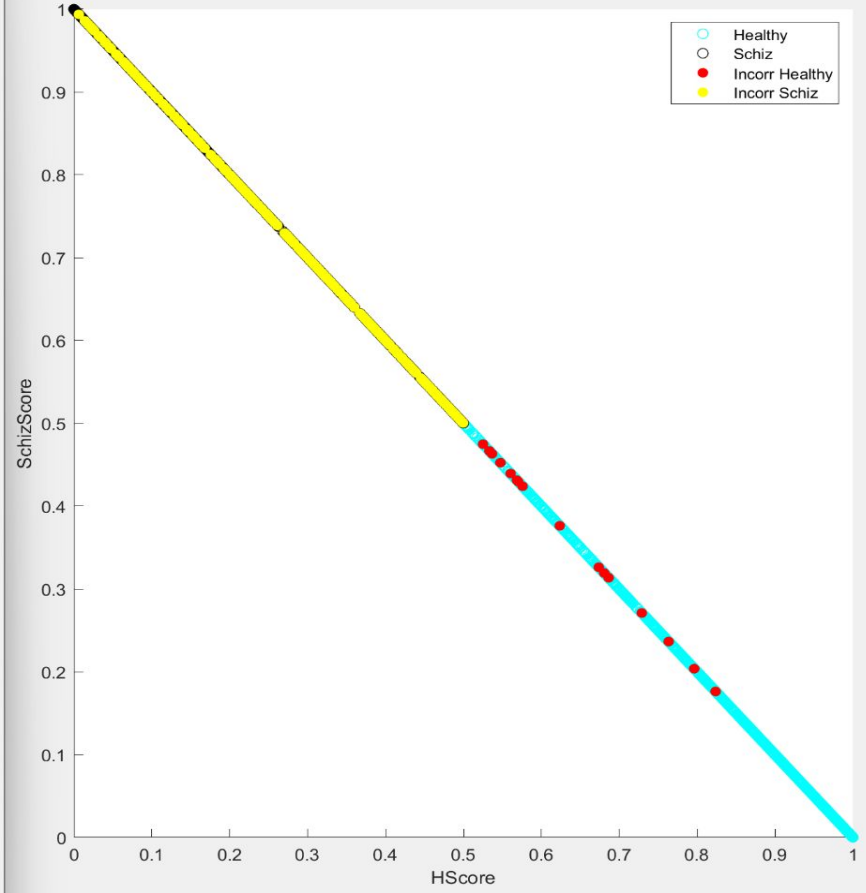
42994×4 table

ClassName	Threshold	FalsePositiveRate	TruePositiveRate
h_coronSlice	1	0	0
h_coronSlice	1	0	0.026398
h_coronSlice	1	0	0.031205
h_coronSlice	1	0	0.034028
h_coronSlice	1	0	0.039063
:	:	:	:
schiz_coronSlice	-1	0.95933	1
schiz_coronSlice	-1	0.96094	1
schiz_coronSlice	-1	0.96597	1
schiz_coronSlice	-1	0.9688	1
schiz_coronSlice	-1	1	1

Display all 42994 rows.

Plotting Raw Scores vs. PCA'd FCN Activations





So Which Images are Classified Incorrectly?

incorrH_fileNames

16x108 char

```
val =

    'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036520_ses-20040101_acq-mprage_run-03_T1w_Slice61.tif'
    'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036520_ses-20040101_acq-mprage_run-03_T1w_Slice62.tif'
    'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036520_ses-20040101_acq-mprage_run-03_T1w_Slice63.tif'
    'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036520_ses-20040101_acq-mprage_run-03_T1w_Slice64.tif'
    'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036520_ses-20040101_acq-mprage_run-03_T1w_Slice65.tif'
    'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036520_ses-20040101_acq-mprage_run-03_T1w_Slice66.tif'
    'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036520_ses-20040101_acq-mprage_run-03_T1w_Slice67.tif'
    'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036520_ses-20040101_acq-mprage_run-03_T1w_Slice68.tif'
    'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036520_ses-20040101_acq-mprage_run-03_T1w_Slice69.tif'
    'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036520_ses-20040101_acq-mprage_run-03_T1w_Slice70.tif'
    'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036520_ses-20040101_acq-mprage_run-03_T1w_Slice71.tif'
    'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036520_ses-20040101_acq-mprage_run-03_T1w_Slice72.tif'
    'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036520_ses-20040101_acq-mprage_run-03_T1w_Slice73.tif'
    'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036520_ses-20040101_acq-mprage_run-03_T1w_Slice74.tif'
    'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036520_ses-20040101_acq-mprage_run-03_T1w_Slice75.tif'
    'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036520_ses-20040101_acq-mprage_run-03_T1w_Slice76.tif'
```

'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036123_ses-20050101_acq-mprage_run-03_T1w_Slice82.tif'
'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036123_ses-20050101_acq-mprage_run-03_T1w_Slice83.tif'
'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036123_ses-20050101_acq-mprage_run-03_T1w_Slice84.tif'
'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036123_ses-20050101_acq-mprage_run-03_T1w_Slice85.tif'
'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036123_ses-20050101_acq-mprage_run-03_T1w_Slice86.tif'
'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036123_ses-20050101_acq-mprage_run-03_T1w_Slice87.tif'
'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036123_ses-20050101_acq-mprage_run-03_T1w_Slice88.tif'
'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036123_ses-20050101_acq-mprage_run-03_T1w_Slice89.tif'
'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036123_ses-20050101_acq-mprage_run-03_T1w_Slice90.tif'
'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036128_ses-20050101_acq-mprage_run-01_T1w_Slice40.tif'
'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036128_ses-20050101_acq-mprage_run-01_T1w_Slice41.tif'
'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036128_ses-20050101_acq-mprage_run-01_T1w_Slice42.tif'
'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036128_ses-20050101_acq-mprage_run-01_T1w_Slice43.tif'
'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036128_ses-20050101_acq-mprage_run-01_T1w_Slice44.tif'
'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036128_ses-20050101_acq-mprage_run-01_T1w_Slice45.tif'
'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036128_ses-20050101_acq-mprage_run-01_T1w_Slice46.tif'
'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036128_ses-20050101_acq-mprage_run-01_T1w_Slice47.tif'
'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036128_ses-20050101_acq-mprage_run-01_T1w_Slice48.tif'
'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036128_ses-20050101_acq-mprage_run-01_T1w_Slice49.tif'
'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036128_ses-20050101_acq-mprage_run-01_T1w_Slice50.tif'
'E:\loose_coronTraining\CoronSlices\h_coronSlice\sub-A00036128_ses-20050101_acq-mprage_run-01_T1w_Slice51.tif'

Figure 5

File Edit View Insert Tools Desktop Window Help

h_coronSlice; BadScore= 0.99995; h_activ= 5.655 sz_activ= -4.3525

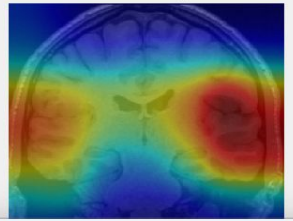


Figure 35

File Edit View Insert Tools Desktop Window Help

schiz_coronSlice = 0.99993; h_activ= -5.2107 sz_activ= 4.3652

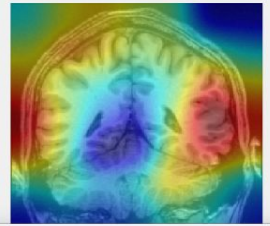


Figure 10

File Edit View Insert Tools Desktop Window Help

h_coronSlice; BadScore= 0.99991; h_activ= 5.0644 sz_activ= -4.2185

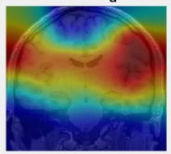


Figure 6

le Edit View Insert Tools Desktop Window Help

h_coronSlice; BadScore= 0.99999; h_activ= 6.8702 sz_activ= -5.3153

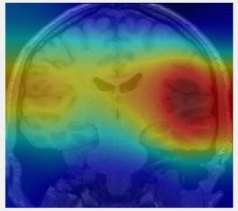
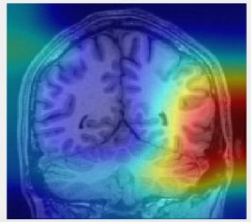


Figure 31

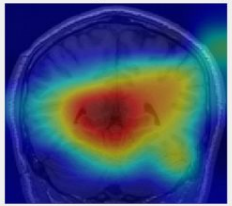
File Edit View Insert Tools Desktop Window Help

schiz_coronSlice = 0.99734; h_activ= -3.2003 sz_activ= 2.7273



Edit View Insert Tools Desktop Window Help

h_coronSlice = 0.72275; h_activ= -0.1197 sz_activ= -1.0778



h_coronSlice; BadScore= 1; h_activ= 6.8495 sz_activ= -5.4917

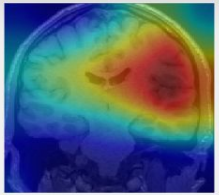


Figure 38

Edit View Insert Tools Desktop Window Help

schiz_coronSlice = 0.9996; h_activ= -3.8936 sz_activ= 3.9187

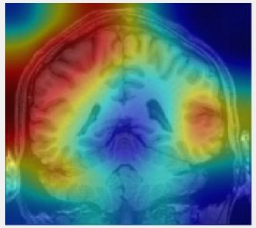
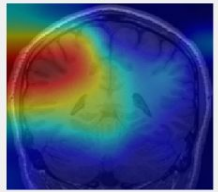


Figure 19

Edit View Insert Tools Desktop Window Help

h_coronSlice = 0.98121; h_activ= 1.7965 sz_activ= -2.159



Project Roadmap

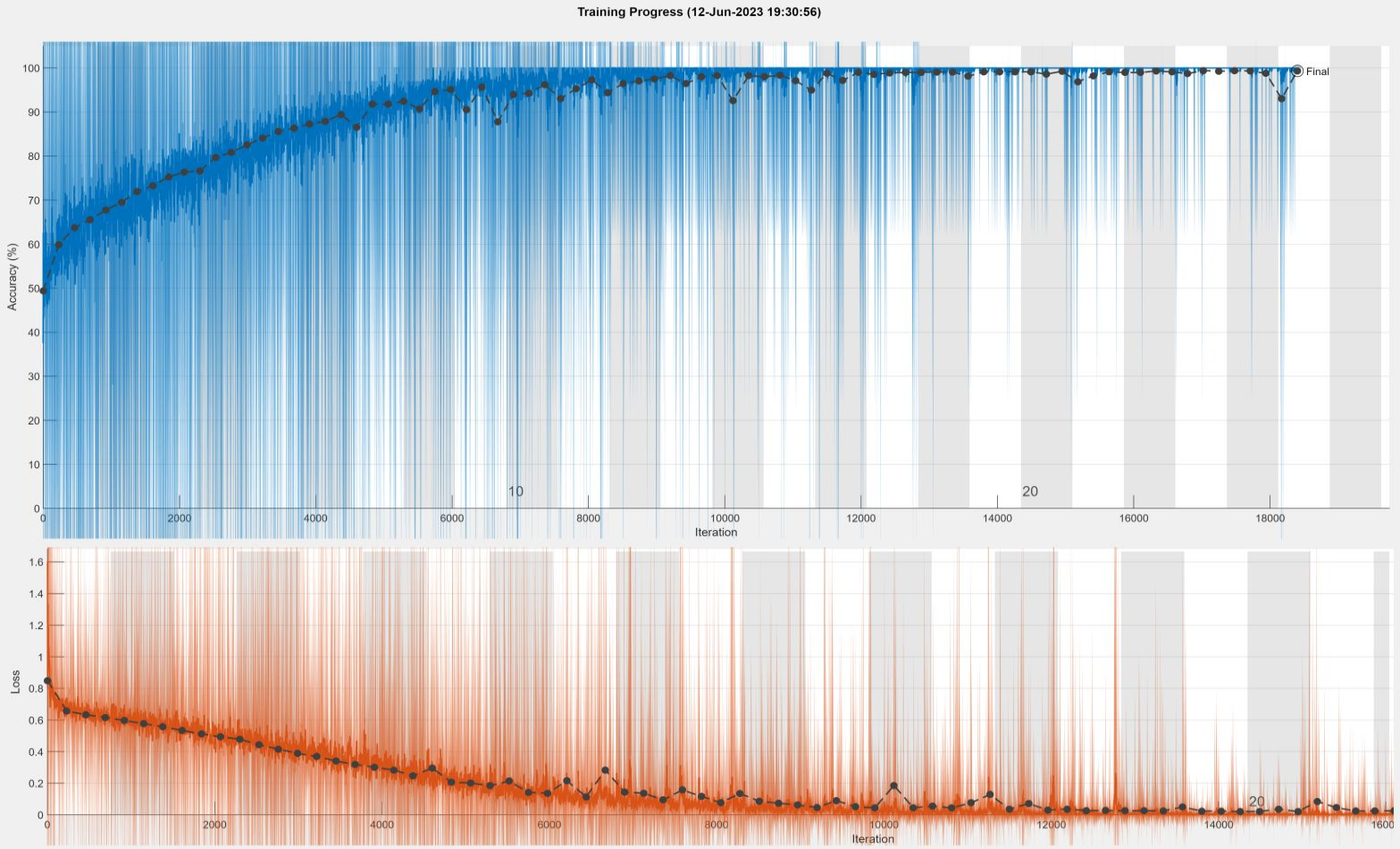
Complete:

- Taking volume of MRI and turning it into slices
  - While preserving the 16 bit MRI images as 16 bit slices
- First GoogleNet Training with 4 MRI images
- Automate Unzipping
- Individual Image min-max normalization
- Train GoogleNet with All Schizconnect Data
- ROC Curve/Area

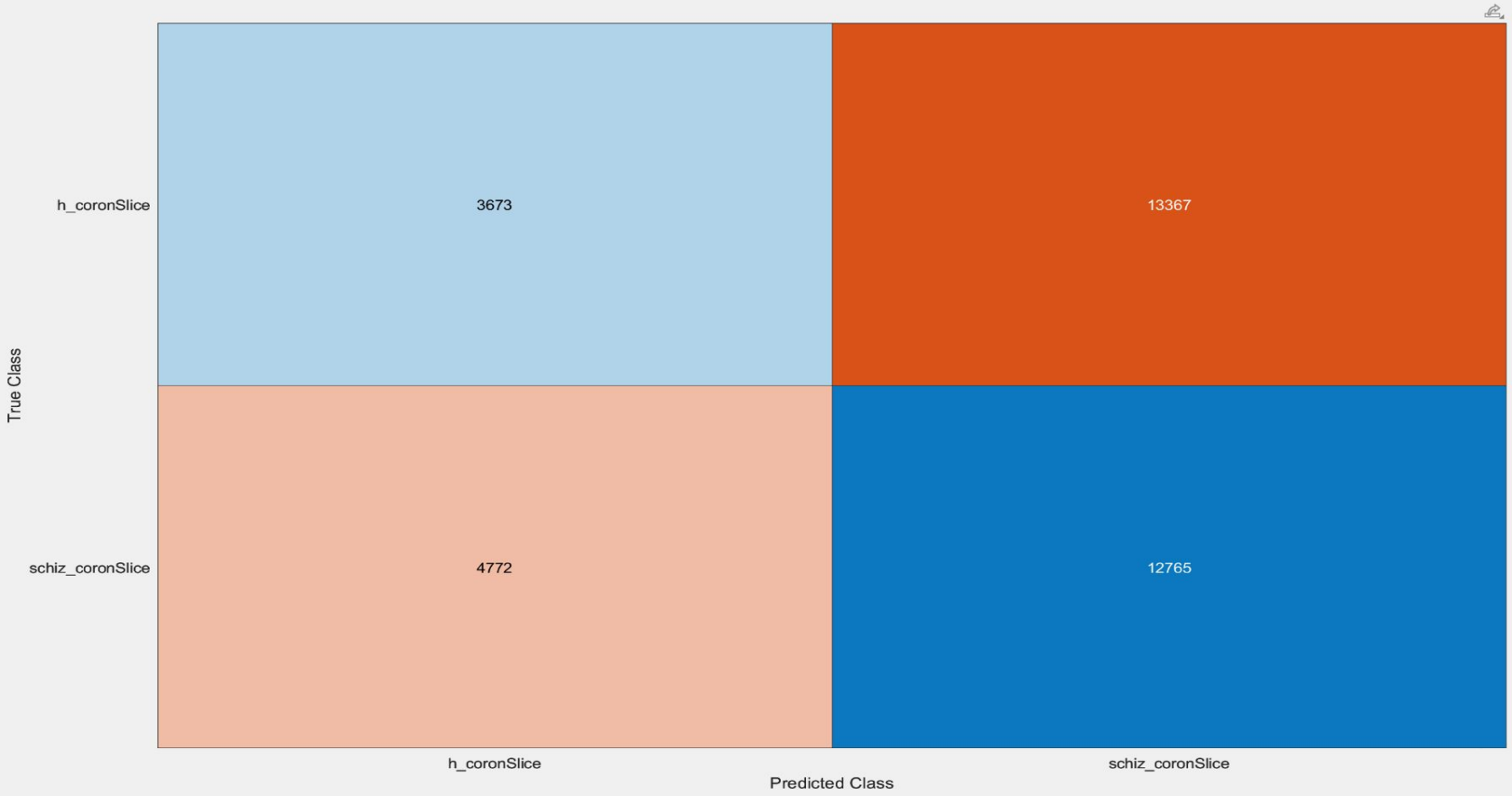
Working:

- ***Assess Network Generalizability***
- Instead of simply scoring every image/slice, let's give each patient an average schiz/healthy score
- Average GradCam
  - Per Slice number
  - Per Participant
  - Per Class
- After everything, DeepDream Images

Re-Run of Training for Confirmation:

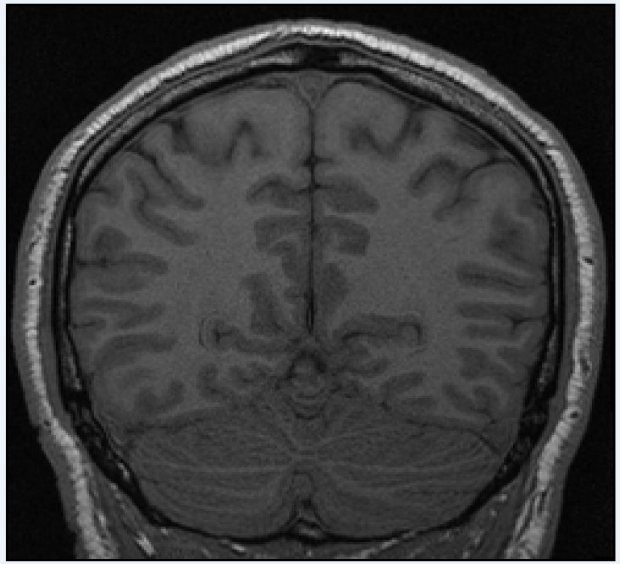


But on New Data (no more “splitLabel” for Test), network coinflips

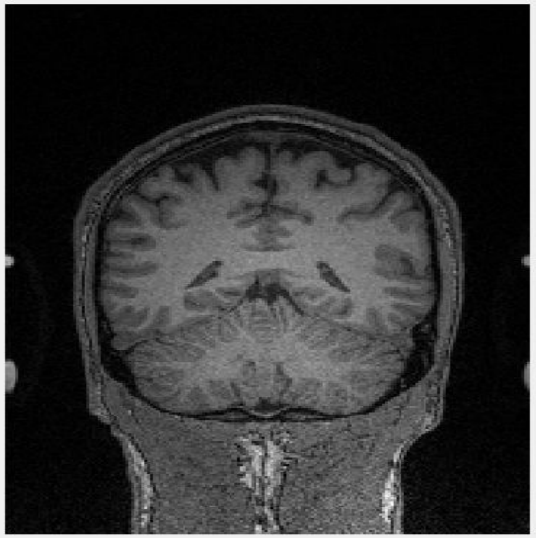


So why could the Network be failing to Generalize?

Training Images:



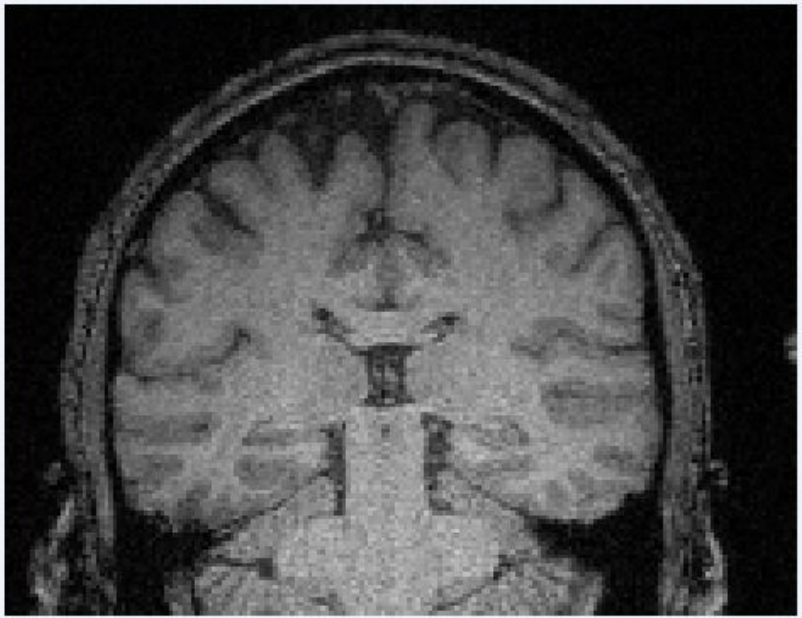
New Study Test Images:



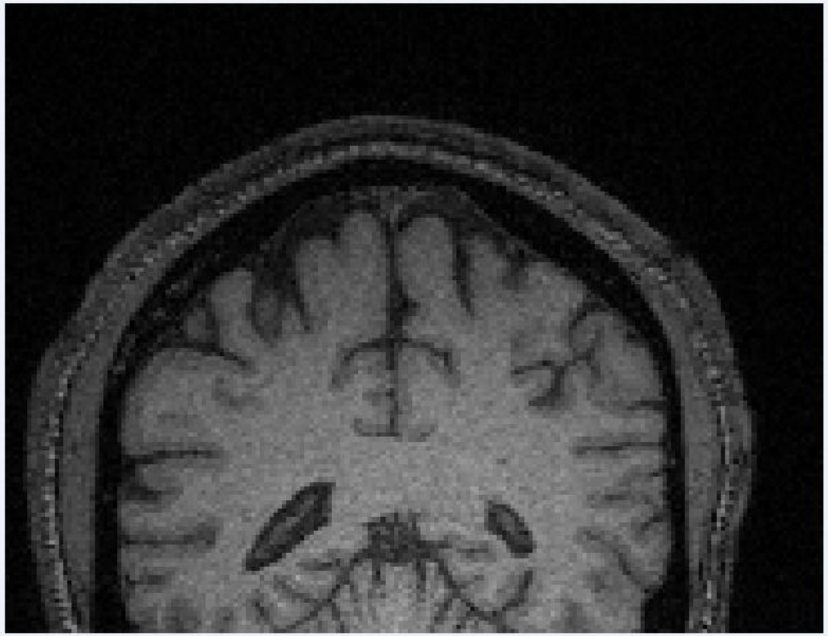
Also very plausible to switch what's training & test data to see if performance improves on test data

After Simple Cropping

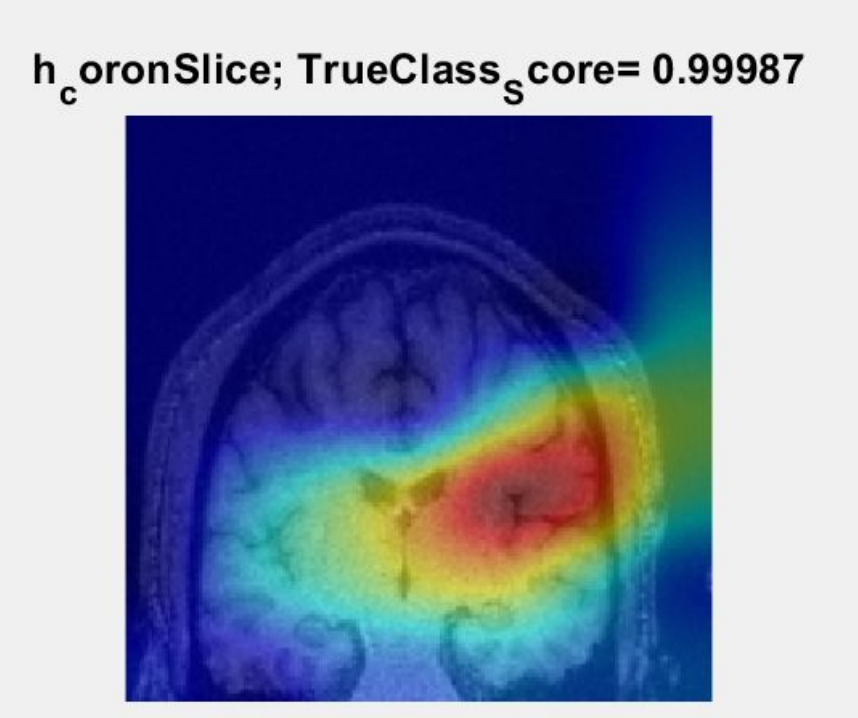
A decent crop:

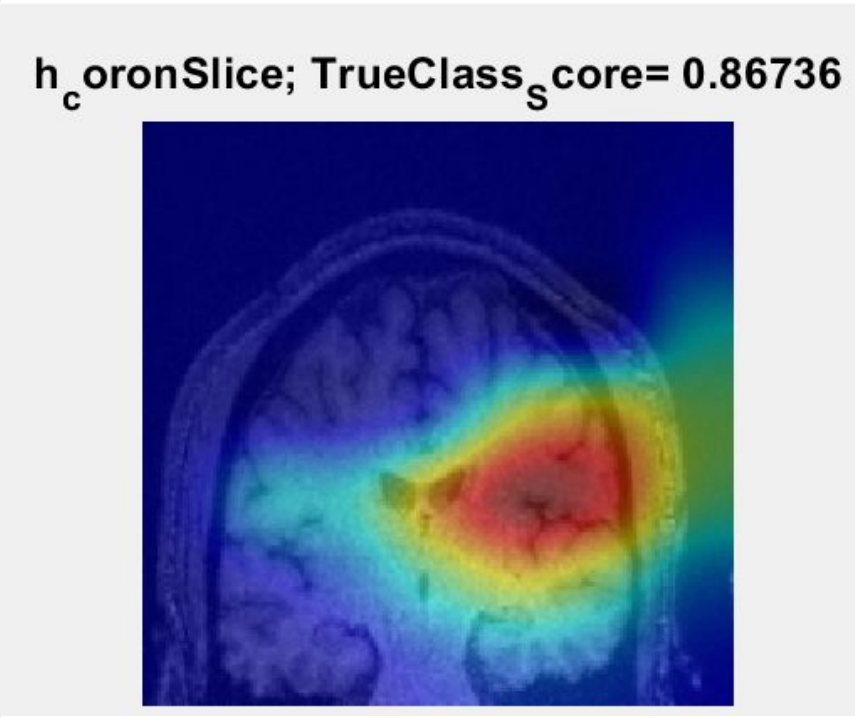


A not so Decent Crop:

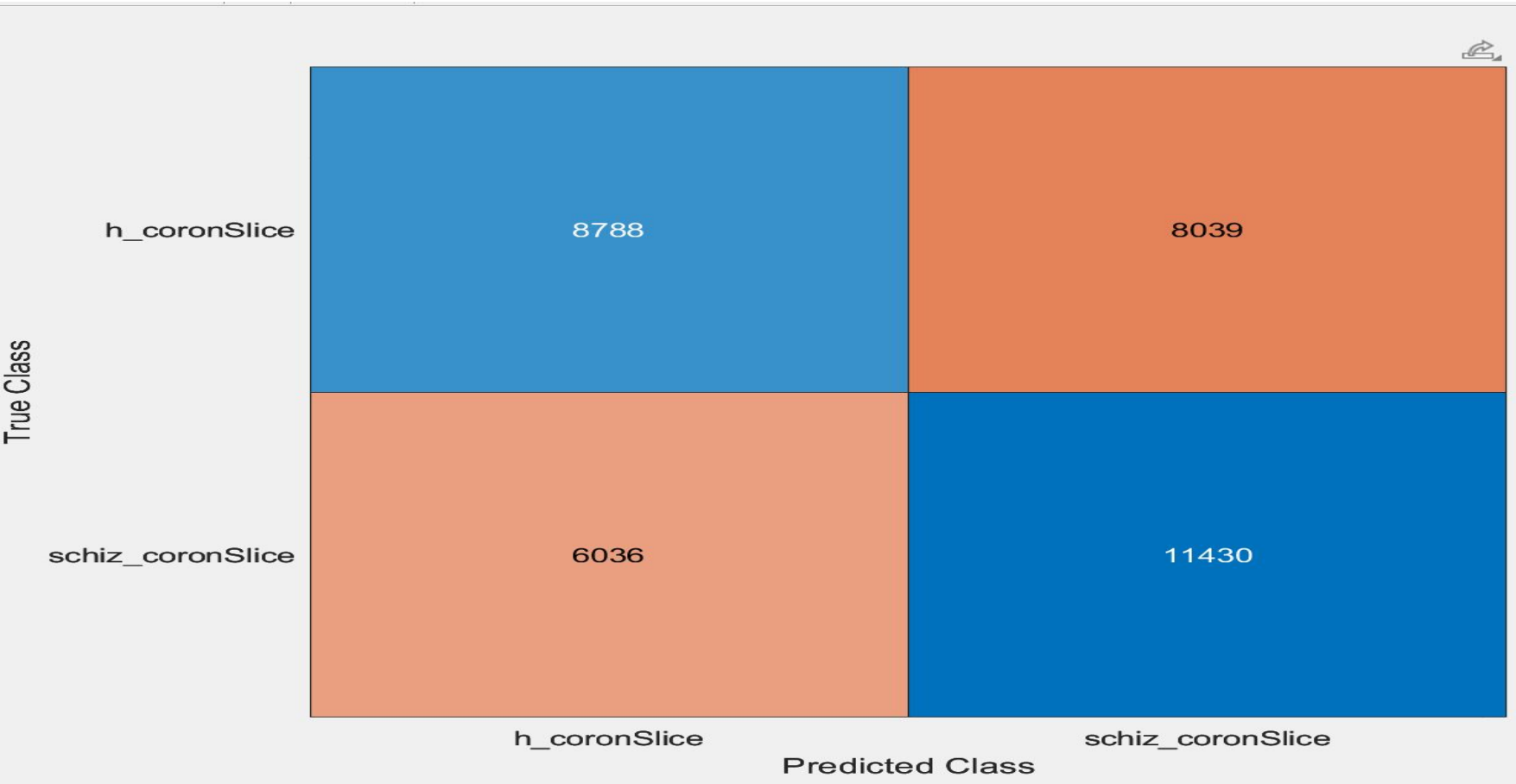


But After Cropping, Heatmaps are in the Cortex Again!

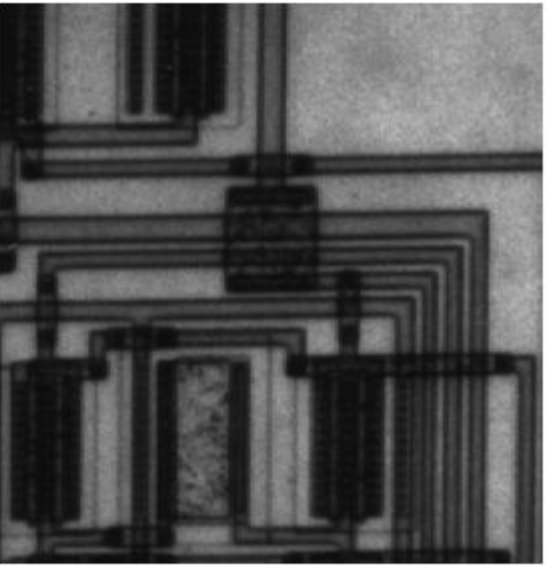


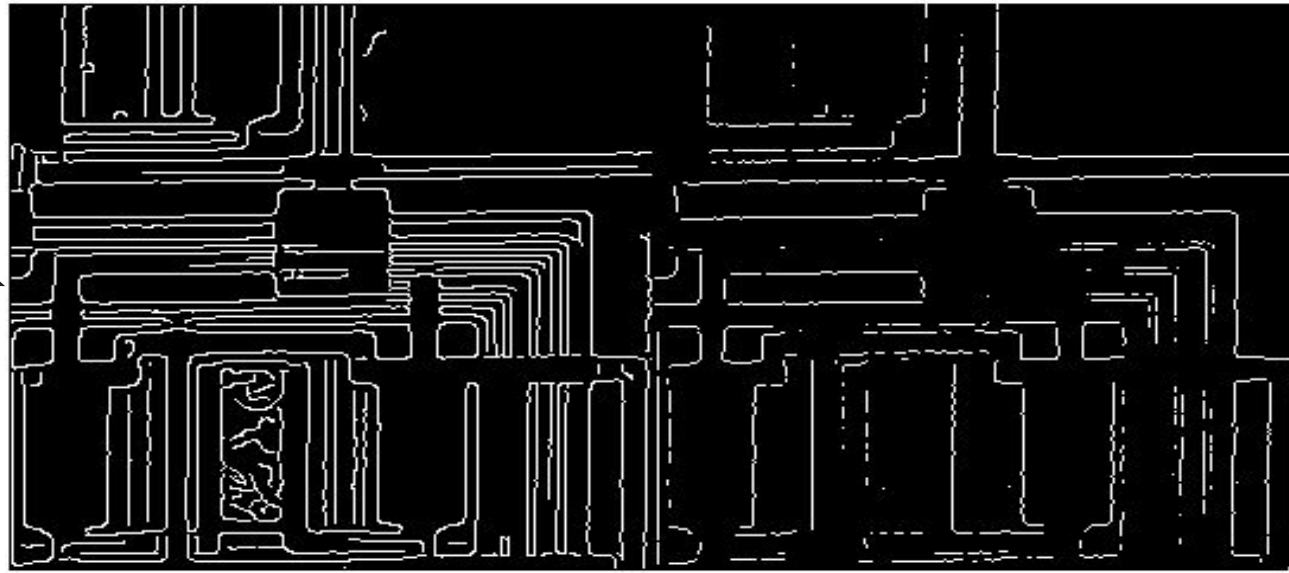


After Crop: Diagonal = +4000; Off-diag = -4000



Now we need Edge Detection





Find edges using the Canny method.

```
BW1 = edge(I,'Canny');
```

Find edges using the Prewitt method.

```
BW2 = edge(I,'Prewitt');
```

Goal will then be to crop out nonzero values as a new image - just the skull / cortex

Image Region Analyzer

EXPLORE

Load Image

LOAD IMAGE

Exclude Border

Fill Holes

Filter

ADD/REMOVE

Choose Properties

PROPERTIES

Export

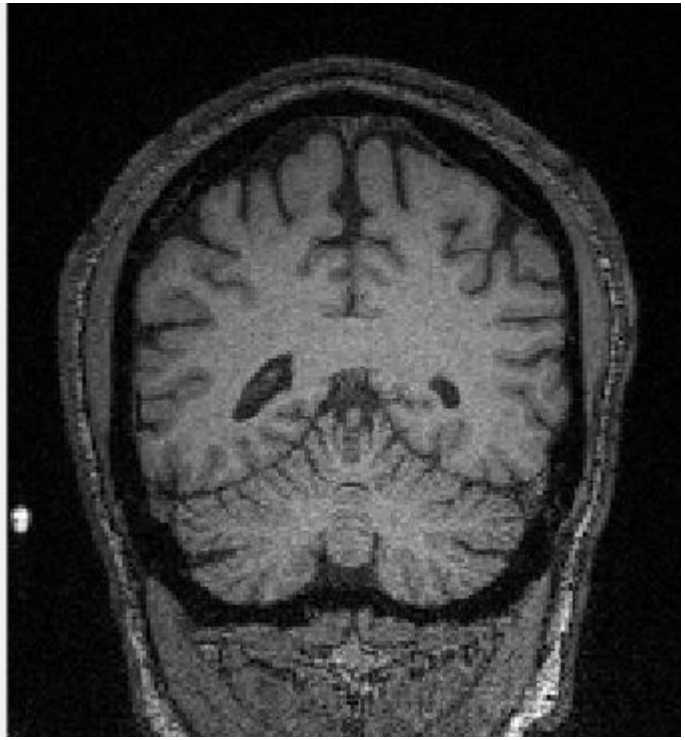
EXPORT

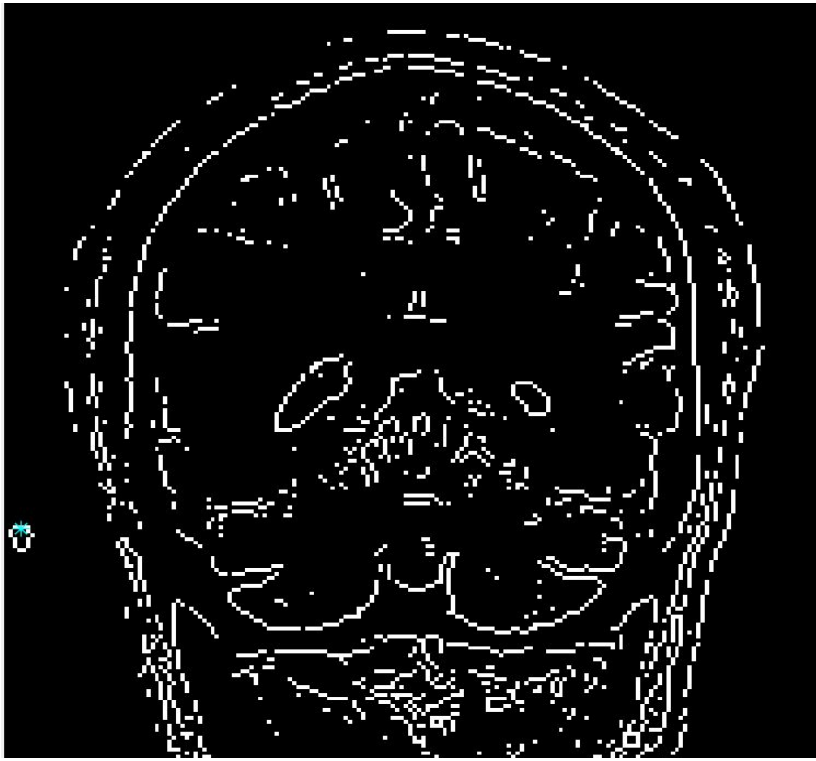
Analyze Regions

Region Properties

Area	Eccentricity	EquivDiameter	EulerN
95	0.9969	10.9981	
88	0.9043	10.5851	
66	0.9966	9.1670	
57	0.9497	8.5191	
43	0.9400	7.3993	
38	0.9066	6.9558	
33	0.9986	6.4820	
32	0.9979	6.3831	
30	0.7849	6.1804	
30	0.9114	6.1804	
29	0.8848	6.0765	
28	0.9614	5.9708	
28	0.9140	5.9708	
26	0.9114	5.7536	
25	0.9918	5.6419	
23	0.9621	5.4115	
23	0.9767	5.4115	
22	0.7852	5.2926	
21	0.9912	5.1709	
21	0.9969	5.1709	
21	0.9973	5.1709	
20	0.9977	5.0463	
20	0.9933	5.0463	
20	0.9167	5.0463	
20	0.9949	5.0463	
18	0.8954	4.7873	
17	0.9800	4.6524	
17	0.9766	4.6524	
17	0.6860	4.6524	

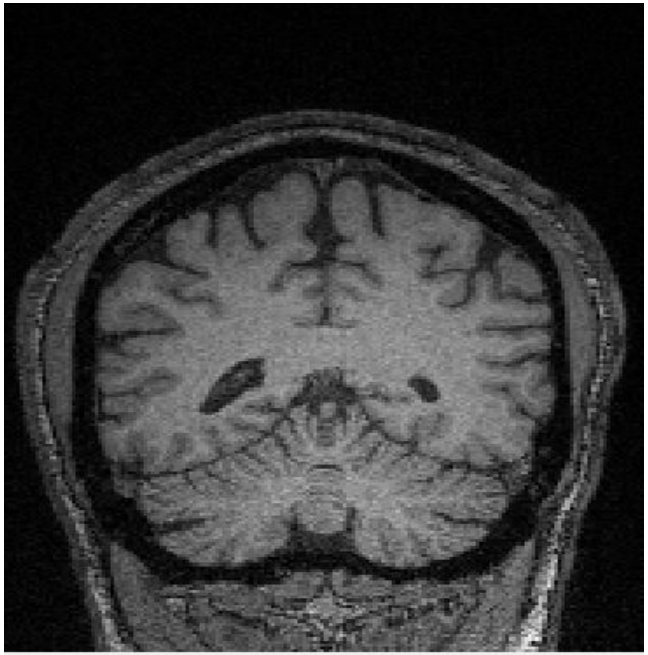
Auto-Generated Bounding Boxes aren't Perfect, though

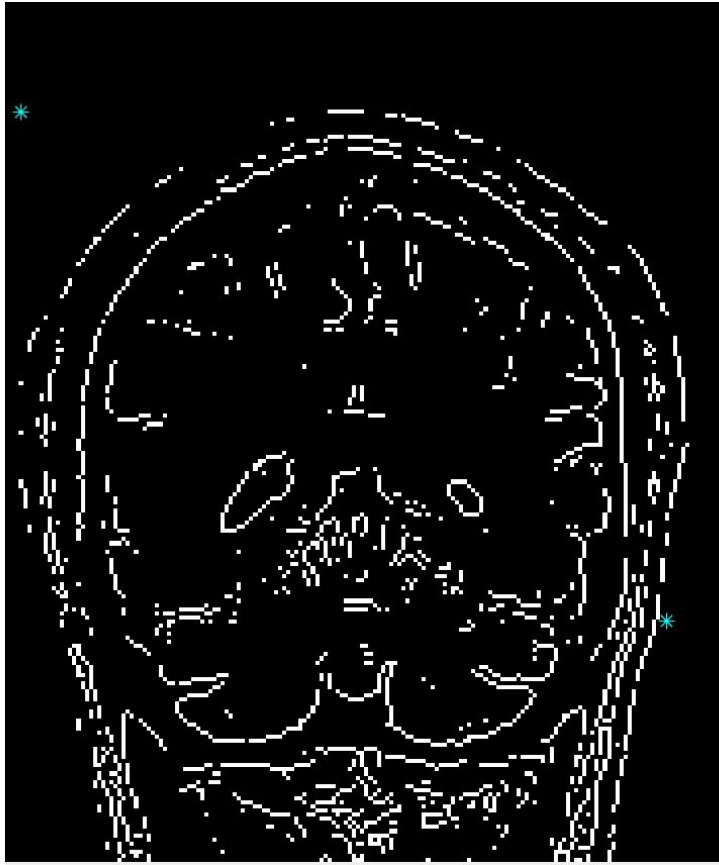




```
tester = regionprops(rotIm,'BoundingBox');
bbox = tester.BoundingBox;
```

A quick, but Potentially Unsatisfactory fix...



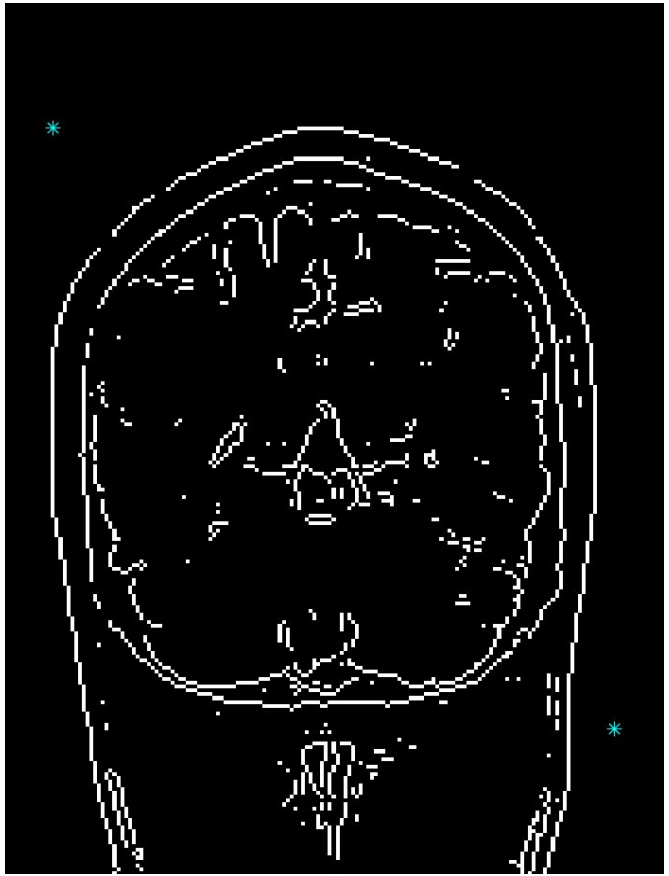


```
quickCrop = imcrop(corona_primeSlice,[10,10,240,180]);
```

```
edgeDetect = edge(rotIm);
```

```
[row, col] = find(edgeDetect);
```

Goal is to Keep Aspect Ratio 3:4, Some Boxes Better Than Others



Project Roadmap

Complete:

- Taking volume of MRI and turning it into slices
  - While preserving the 16 bit MRI images as 16 bit slices
- First GoogleNet Training with 4 MRI images
- Automate Unzipping
- Individual Image min-max normalization
- Train GoogleNet with All Schizconnect Data
- ROC Curve/Area

Working:

- *Assess Network Generalizability*
  - **Edge detection for cropping**
  - **Test on new Data**
- Instead of simply scoring every image/slice, let's give each patient an average schiz/healthy score
- Average GradCam, Occlusion Sens, etc.
  - Per Slice number
  - Per Participant
  - Per Class
- After everything, DeepDream Images