This readme accompanies the code for the project described in: IMPROVED\_MEDICAL\_IMAGE\_CLASSIFICATION\_THROUGH\_TEST\_BASED\_LEARNING.pdf

To run the code, first unzip data.zip

Run 269\_alz\_project\_GenerateStartingModel.ipynb to train initial .h5 model

Then run 269\_alz\_project to evaluate

Other files are functions used in the main code:

data/test: test data folder containing reserved test data for each Alzheimer’s case

data/train: train data folder containing reserved training data for each Alzheimer’s case

val20\_epochs10\_START\_testacc82.h5: model file used for experiments after training initial .h5 model

data\_selector:

python function

input: data - nx3 numpy array where the first column contains the image distances from the hyperplane, the second column contains the image class identification bool index values, and the third column contains its overall index number in the data folder

outputs: easy, medium, hard - nx3 numpy arrays where the first column contains the “easy”, “medium”, or “hard” image distances from the hyperplane, the second column contains the “easy”, “medium”, or “hard” image class identification bool index values, and the third column contains the “easy”, “medium”, or “hard” image’s overall index number in the data folder

description: this function divides the results from the support vector matrix into three classifications of data based on their distances from the hyperplane. The “medium” classified images are made up of those within one standard deviation from the average distance value from the hyperplane. The “easy” classified images are made up of those outside one standard deviation from the average distance value from the hyperplane, and who’s distance values are greater than the average distance value from the hyperplane. The “hard” classified images are made up of those outside one standard deviation from the average distance value from the hyperplane, and who’s distance values are less than the average distance value from the hyperplane.

evaluate\_most\_confused\_labels:

python function

input: conf\_arr - 4x4 np array containing prediction errors from the model evaluation

output: most\_confused - 1x2 array of 1x2 arrays containing the index values of the two most confused classes in the confusion matrix

description: this function evaluates the test results in the confusion matrix to determine which two classes are confused the most by the model. It searches for where the maximum values in the non-diagonal fields of the matrix are located and returns the class numbers associated with the highest errors.

Fix\_labels:

python function

input: list\_labels - nx1 array of lists containing four binary entries where 0s indicate the image is not associated with the class index value and 1s indicate the image is associated with the class index value.

output: array\_labels - nx4 numpy array containing the same four binary entries associated with each image index in the list.

description: this function is simply a formatting helper function that converts the format of the labels stored by the support vector matrix into one usable by the keras.fit function call.