Assignment 4

Bharat Mallala (bmallala)
Jyothi Pranavi (jyodevin)
Harshit Krishnakumar (harkrish)

K nearest Neighbors:

Design decisions- iterated o multiple value of k

K- value	Accuracy
K= 16	71.26%
K = 17	71.47%
K= 18	71.68%
K =15	71.47%

The algorithm has the best accuracy for k = 18. The program takes about 2.5 minutes to run for testing. The training phase of this algorithm is just storing all of the train data in the model file. For every image in the test data, we have calculated the Euclidean distance to every image in the train data and has summed over all the value to get the error.

Recommended- Based on the data set the optimal number of k nearest neighbors varies. Optimally for any data set it is best not to take a value which is too small or too big.

Adaboost:

Design decision-

Decision Stumps	Accuracy
70	62.48
100	65.005

The training part for this algorithm takes about 20 mins for 70 decision stumps. The training phase includes randomly picking two pixels from the 192 and comparing them for the hypothesis part and building weak classifiers based on the error calculated after run. These values are stored in the model file. In the testing phase each image is classified based on all the weak classifiers and a label which has the highest majority is assigned to the image.

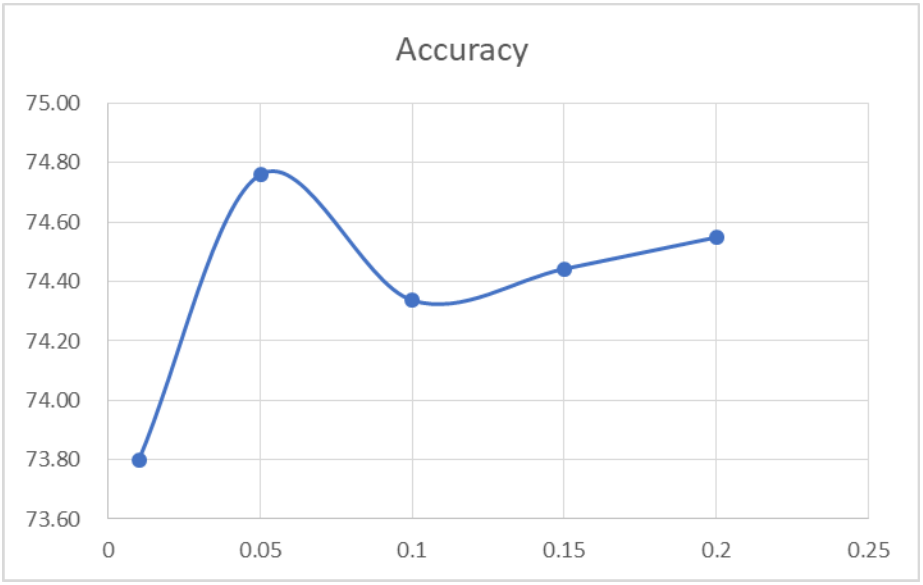
Recommended- The hypothesis taken for the decision stumps depend on the data set we are dealing with. For this data set we have randomly chosen 2-pixel values and compared the values. The ideal number of decision stumps would be between 70 and 100.

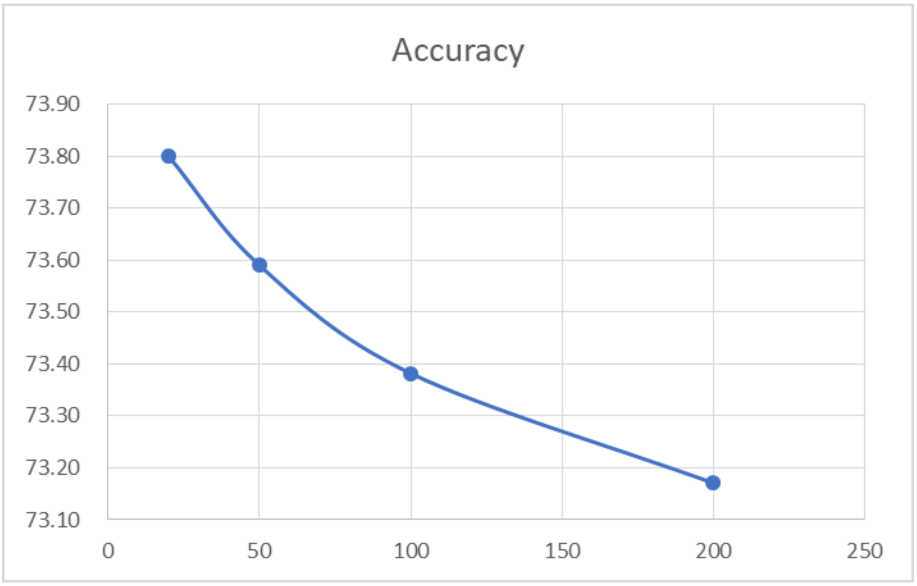
Neural Network:

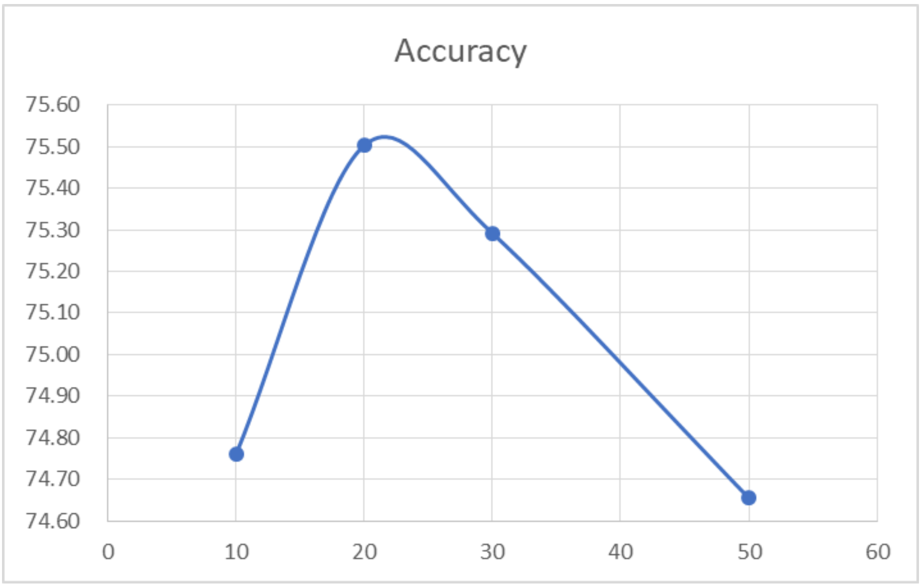
We wrote Neural Network code using Python Numpy library. The structure of the neural network is: 1 hidden layer and 1 output layer. In order to avoid NAN's the input was normalized, and after every step, the weights are rounded to nearest 10 digit floating points, this is in order to avoid very low numbers like E-100.

Further, initialization of weights were done using Xavier Initialization, which recommends the weights to have a variance of 2/sqrt(number of inputs * number of outputs to neuron). This makes sure that the weights are not too huge or too small when initialized.

We tried multiple configurations for neural networks, the results are plotted as a graph.







Number of units	Accuracy
20	73.80
50	73.59
100	73.38
200	73.17

eta	Accuracy
0.01	73.80
0.05	74.76
0.1	74.34
0.15	74.44
0.2	74.55

Epochs	Accuracy
10	74.76
20	75.50
30	75.29
50	74.66

Decay	Accuracy
0.6	75.08
0.9	74.87

Recommended- From the experiments, we would recommend the following neural network model: 1 hidden layer with 20 units, Learning Rate of 0.05, 20 epochs. We tried using decay to improve on the accuracy, but the accuracy without learning rate decay turned out to be better than the one with decay.

Best Model:

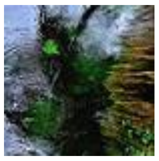
Based on the above three algorithms, the neural network algorithm work best with an accuracy of 75.50.

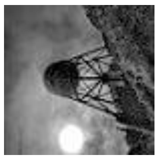
Sample images:

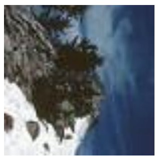
Correctly classified:

These images were correctly classified by the neural net algorithm. All these are images of landscapes, with some references in the image (for example 'sky') which contributes towards prediction.

Wrongly classified:







These images are misclassified by the neural net algorithm. These images have a bit of ambiguity in them. For example one of the image is the reflection of something in water while the other is a black and white image, which makes them really hard for the classifier to predict correctly.