AI assignment 4

- Identify a few parameters (like number of neurons, number of layers, etc), and conduct experiments to study their effect on the final classification accuracy
- Present neatly-organized tables or graphs showing accuracies and running times as a function of the parameters you choose
- Which classifiers and which parameters would you recommend to a potential client
- How does performance vary depending on the training dataset size, i.e. if you use just a fraction of the training data?
- Show a few sample images that were classified correctly and incorrectly. Do you see any patterns to the errors?

K-nearest neighbors:

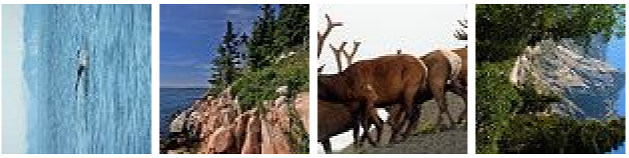
Parameters:

- K
- Mini-batch (take random samples)

Exp #	**K**	**# of training examples**	**Time (in sec)**	**Accuracy**
1	10	~36000	993.2	70.3%
2	20	~36000	1005.01	70.5%
3	10	~5000	141.18	67.4%
4	20	~5000	227.11	67.4%

Some wrong classifications:

Some correct classifications:



We found that some images that were classified incorrectly were actually difficult to identify from naked eyes. In other cases, the landscape (probably the identifying feature) was not very conspicuous.

ADA-boost:

Parameters:

- Multivariate classification:
  - 4 separate binary classifiers were used, each for 0,90,180 and 270 degree orientation. The one that had the highest weight for positive classification was chosen as the final orientation.
- Number of decision stumps:
  - The initial no. of stumps to be chosen from was 192*192 (comparing every pixel with another) that proved to be too high as choosing the best classifiers from that number took a lot of time for the algorithm
  - Then, the number was cut down by half (by comparing every pixel pair only once)
  - Later, only pixels in top two rows were compared with bottom two and similarly left two columns with right two which reduced the no. of stumps significantly
  - It was noticed that just comparing top two rows with bottom two rows gave more accuracy and made the algorithm faster, hence 2304 was chosen as the no. of weak classifiers from which the best 20 classifiers were chosen
- Number of iterations
  - Reducing the no. of classifiers enabled the algorithm to have more iterations (more weak learners)
  - Starting from 3, the iterations were increased step by step to 20

Exp #	No. of total decision stumps	# of training examples	# of iterations (weak learners chosen)	Training time	Accuracy
1	36672	36000	3	180m	27%
2	36672	1000	3	4m	38%
3	18336	1000	3	1.5m	41%
4	2608	1000	20	1.5m	28%
5	2304	36000	20	80m	54%

- Which classifiers and which parameters would you recommend to a potential client

    - Rather than having a huge no. of classifiers to be chosen from, a less no. of classifiers which are more effective would make the algorithm faster. E.g. than comparing pixel to pixel, a better approach would have been to divide the image into rectangles of random sizes and using a classifier that would split the rectangle using a random threshold. This would reduce the no. of classifiers but increase the decision making power of the same.
    - Although the no. of iterations was chosen by experimenting, I would recommend the no. of iterations to be high but not so high that it would cause overfitting. No. of iterations beyond 50 causes the model to lose its accuracy.

- Training set size
    - The size of the training data, 36000 did cause the model to take significantly high time to train. A smaller fraction of around 1000 samples was much faster with acceptable accuracy.
    - A very small fraction of data causes the model to overfit, giving 100% accuracy. A slightly bigger fraction, 1000 samples gives accuracy a little better than the entire training data.

Some wrong classifications:

Some correct classifications:

Neural Net:

We implemented neural net models with a single hidden layer.The number of nodes in the input layer were 192, these coincide with the number of 64 pixels * 3(R,G,B) values of an image. We identified the epochs, learning rate and number of neurons in the hidden layer as the most significant factors that could affect our model.The output neurons were fixed at 4 in accordance with 4 labels. The activation functions used on the hidden and output layers were sigmoid and tanh.

Exp #	**Topology**	**# of training examples**	**# of epochs**	**Learning rate**	**Training time (in sec)**	**Accuracy**

1	192, 10, 4	~36000	10	0.15	~15 minutes	46%
2	192, 10, 4	~36000	50	0.15	~45 minutes	64%
3	192, 10, 4	~36000	100	0.15	~2 hours	50.69%
4	192, 5, 4	~36000	10	0.15	~15 minute	65.53%
5	192, 5, 4	~36000	10	0.015	~15 minute	69.7%
6	192, 5, 4	~36000	50	0.015	~15 minute	72.1%

Some wrong classifications:

Some correct classifications:

We found that neural networks gave us the best accuracy of 72%. The specifications for the model were :

1) Learning_rate = 0.015
2) Epochs = 50
3) Number of hidden neurons = 5