

Classification

Name	Roll No
Noor Ahmed	15-4265

1. Classification Algorithms used

- K-Nearest Neighbour Classifier
- Decision Tree

2. Preprocessing

- Simple Scaling between 0-1
- Normalisation followed by scaling (increases classification error)
- Deskewing
- Sobel + Moment + Custom Filters(Brightness levels + HOG) = Extracted Features (Accuracy not satisfactory)

3. Result

KNN:

Performance of KNN on Image Pixels alone (No Preprocessing):

Time taken to 'train/fit' the model is negligible so it wasn't added

```
For k = 1 Accuracy was: 0.982524271845, Time Taken: 20.46740698814392
For k = 2 Accuracy was: 0.993124271845, Time Taken: 14.699619041171265
For k = 3 Accuracy was: 0.98380970814, Time Taken: 14.408888035282026
For k = 4 Accuracy was: 0.98291621359, Time Taken: 14.749805927276611
For k = 5 Accuracy was: 0.98291621359, Time Taken: 15.235103845596313
For k = 6 Accuracy was: 0.98380970814, Time Taken: 15.741915225982666
For k = 7 Accuracy was: 0.98213592233, Time Taken: 14.012870982855229
For k = 8 Accuracy was: 0.980970972786, Time Taken: 11.667812061389014
For k = 9 Accuracy was: 0.98058524212, Time Taken: 11.648888958557129
```

matplotlib.pyplot.plot at 0x1106a7480



Experiments that followed use K = 6 as accuracy was good and time taken was minimum for that accuracy

Performance of KNN on Scaled Image Intensity Values:

Average time to predict increased by 2.1s for scaled train and test data

0.9833

Performance of KNN on Normalised Image:

0.9724

(Normalized Image yields lesser accurate results as it changes the distribution of the pixel values)

Performance of KNN on Scaled Image + Extracted Features:

	Sobel (Horizontal Derivative of Image)Values	Classification based on edge detected features (The motivation behind this was bold written digits in train set)	Image Fourier
Accuracy	0.97203	Canny => 0.9605	Error due to complex number calculation

Performance of KNN on Scaled Image + Deskewed Image:

0.99339805825242722

0.98990291262135921 (with image derivative)

Decision Tree:

Performance of Decision Trees on Scaled Image:

0.93126213592233009

Performance of Decision Trees on Scaled Image (Parametrized):

	Criterion	Max_Depth	Accuracy
1	entropy	400	0.93126213592233009
2	entropy	30	0.92970873786407771
3	gini	30	0.92388349514563106

Performance of Decision Trees on Scaled Image with PCA: (Parameters from previous best experiment chosen)

```

Dimensions= 50 0.502912621359
Dimensions= 100 0.70640776699
Dimensions= 150 0.741747572816
Dimensions= 200 0.737087378641
Dimensions= 250 0.742524271845
Dimensions= 300 0.747961165049
Dimensions= 350 0.734757281553

```

Reduced dimensions reduced time complexity although it did not yield accurate results

4. Discussions and Conclusion

KNN gave accurate results for data that was preprocessed. Although time taken for train data of almost 12k examples took: 5.179606914520264 seconds. This may be much larger for larger datasets, not taking into account the time taken to process the test set also. But for a small dataset deskewing with KNN model gave really good results predicting wrong only 10 examples out of all the test examples.

Image processing techniques did not help that much although they couldve been utilised better. Some techniques i wished to implement in my solution was Histogram of oriented gradients. Instead canny edge detector was used which yielded unsatisfactory results as stated above.

Decision trees predicitions were slightly over fit to the training data. Reducing max_depth and dimensions of the training example helped the process but still yielded unsatisfactory results due to underfitting and ambiguous examples.