Pattern Recognition

Assignment #5

Non Parametric Methods

Professor:
Hema A. Murthy

Nirav Bhavsar (CS17S016)
Sidharth Aggarwal (CS17S012)

1.1 Non Parametric Methods

In this assignment we have dealt with the non parametric methods for classification of different types of data. The data which is used in the assignment are Image Data, Speech and Hand Written Data set. Further we have performed different methods which are as follows :-

1. Bayesian classifier on Parzen Window Using Gaussian Kernel
2. Fisher Discriminant based Classifier
3. Perceptron based Classifier
4. Support Vector Machine Based Classifier
5. Multi-layer Feed-forward Neural Network based Classifier

For the classifation we have used the Bayesian classifier in which we have taken the densities of the data to be gaussian:-

$$f(x) = 2\pi^{-\frac{d}{2}} \det \Sigma^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x-\mu)^{\mathrm{T}}\Sigma^{-1}(x-\mu)\right)$$

1.2 Bayes classifier on Parzen window method using Gaussian kernel

For this we have applied the Parzen window using Gaussian kernel in which the volume is fixed with constant variance of h = 0.5 choosen by observing below graph, but the number of points coming within the volume changes. So as it the gaussian kernel so we have the partial points also in consideration. It is applied on the Image Data set. Further after getting the probabilities we have applied the bayesian classifier for classifying the whole image into one class by voting and by taking max average probability.

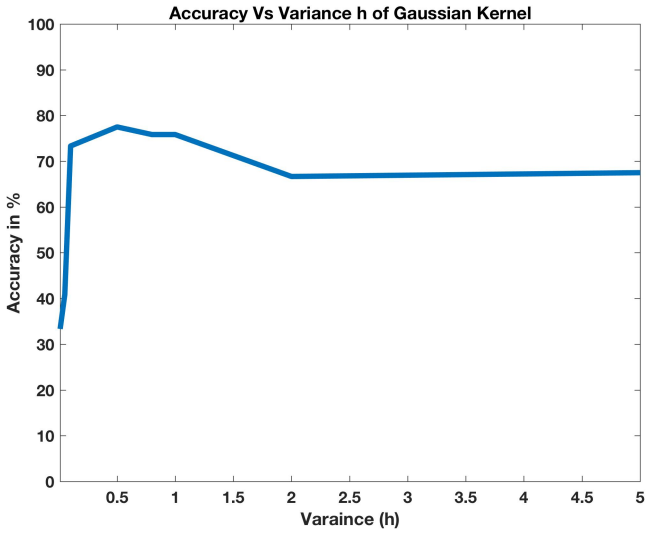


Figure 1.1: Comparision of Accuracy Vs Variance of Gaussian Kernel

Parzen– Raw Dataset for H = 0.5 with Accuracy 75%

Predicted \ Actual Class	Highway	Street	Buildings
Highway	32	0	9
Street	7	38	11
Buildings	1	2	20

Parzen– Normalized Dataset for H = 0.5 with Accuracy 83.33%

Predicted \ Actual Class	Highway	Street	Buildings
Highway	34	1	3
Street	5	32	3
Buildings	1	7	34

Parzen– Within Class Norm. Dataset for H = 0.5 with Acc. 98.33%

Predicted \ Actual Class	Highway	Street	Buildings
Highway	39	0	1
Street	0	40	0
Buildings	1	0	39

Figure 1.2: Confusion Matrix of of Parzen Window for h = 0.5

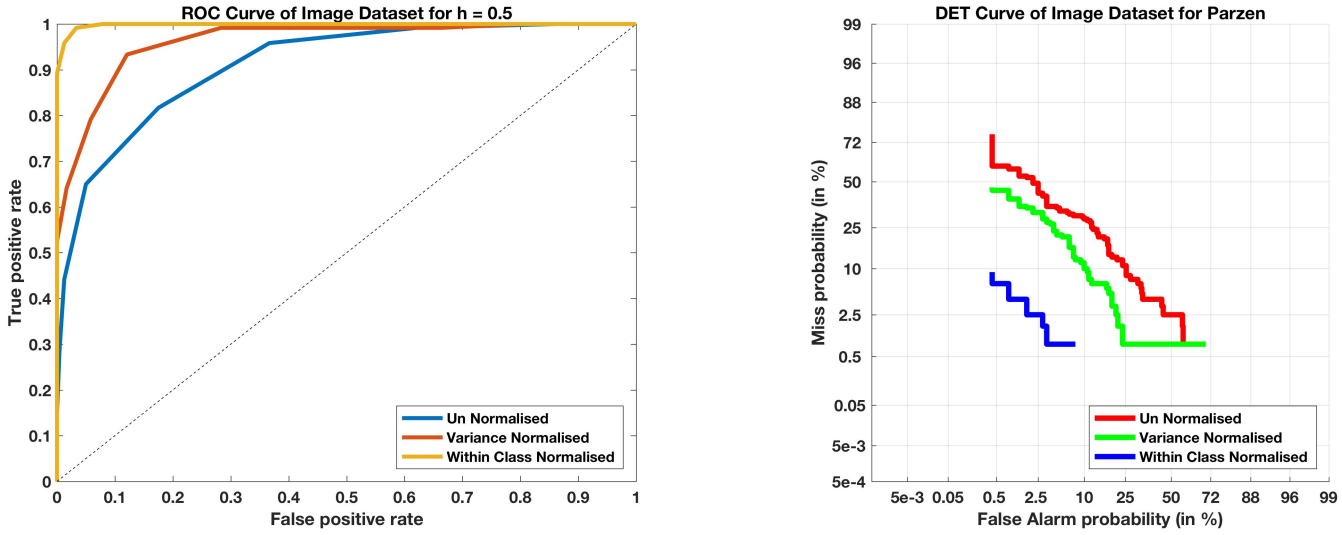


Figure 1.3: ROC & DET of Parzen Window for h = 0.5

Observations :-
From the graph of Accuracy Vs Variance we observed that at h = 0.5 we are getting maximium accuray. Hence after choosing h = 0.5 we applied parzen window using gaussian kerneland we observed that we were getting 75% accuray. And when we did variance normalized on whole data set we got 83.33% accuray. And we were able to get 98.33% when we did within class variance normalisition.

1.3 Fisher discriminant based classifier

Fisher discriminant is a technique which projects the point in the lower dimension taking in consideration that the data is linearly separable in the input space and by minimising within class scatter and maximising between class scatter. We worked on the Image Data set which have 3 classes so we can represent the data point in (c-1) 2 dimensional space. Then to classify points projected in the lower dimension we have applied the Bayesian clssifer.

We have made the confusion matrix which tells the accuracy of the identification done for every Image class. The ROC plot tells the accuracy of the image classification by plotting the graph between TPR and FPR. DET graph tell the accuracy of the model by comparing the Miss & False Alarm.

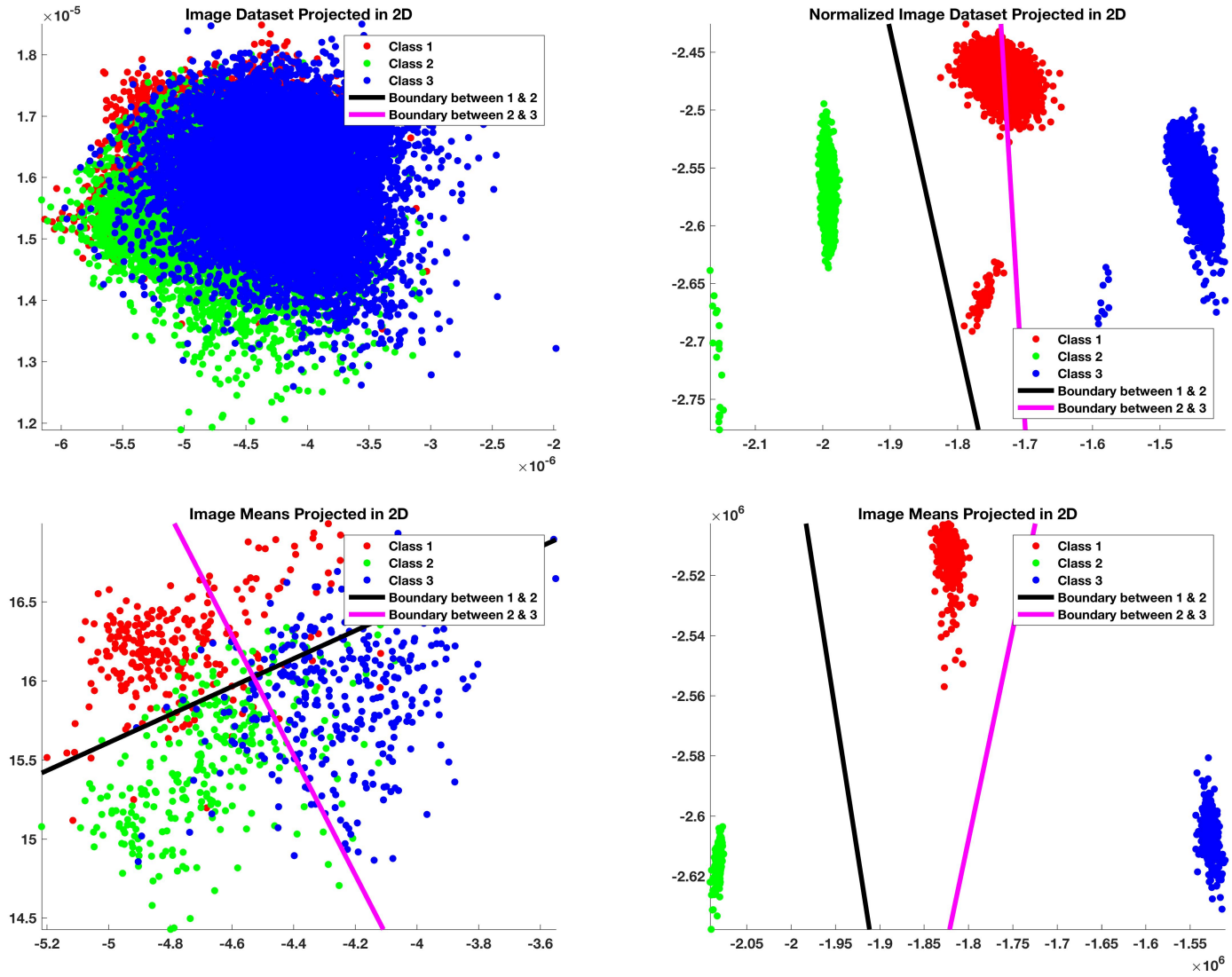


Figure 1.4: Scatter plot of Image Dataset and Image Means for Normalised and Raw dataset

LDA with Whole Var. Normalized Data Accuracy 74.1%

		Actual Class		
		Highway	Street	Buildings
Predicted	Highway	35	7	7
	Street	4	28	7
	Buildings	1	5	26

LDA with Raw Data Accuracy 75.8%

		Actual Class		
		Highway	Street	Buildings
Predicted	Highway	29	2	2
	Street	5	30	6
	Buildings	6	8	32

LDA with Within Class Var. Normalized Data Accuracy 100%

		Actual Class		
		Highway	Street	Buildings
Predicted	Highway	40	0	0
	Street	0	40	0
	Buildings	0	0	40

Figure 1.5: Confusion Matrix of LDA

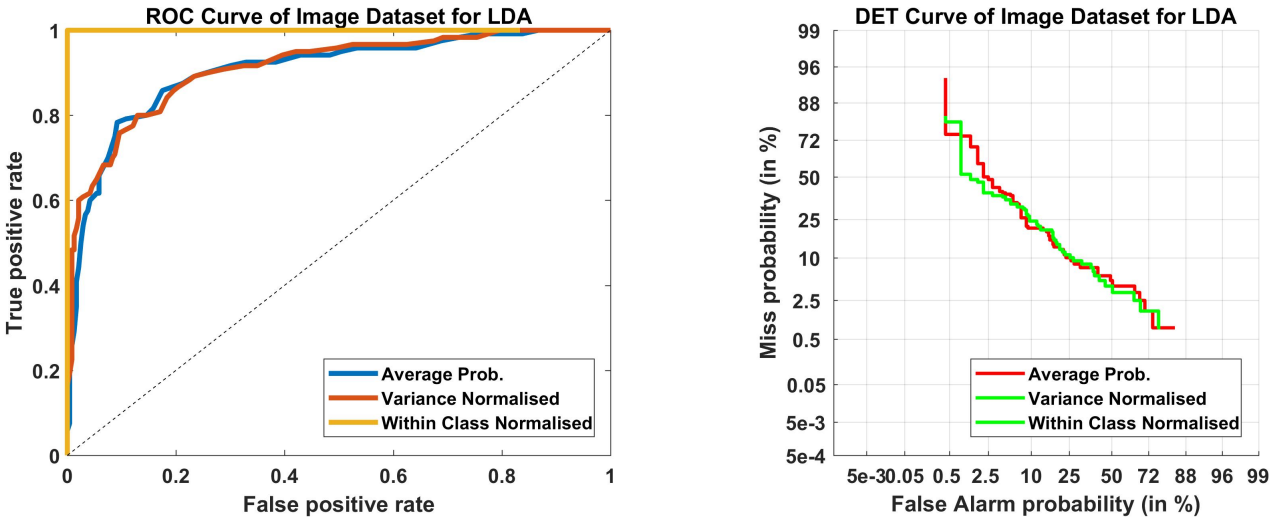


Figure 1.6: ROC & DET of LDA

Observations :-
By performing LDA we projected image data in 2 dimension and then performed Bayesian Classifier on those projected data and we were getting 75.8% accuray. But when we did within class variance normalisation we were getting 100% accuracy, which we can observe from the scatter plot also that the project image data set is linearly seperable.

1.4 Perceptron based classifier

Perceptron is the technique which gives us linear hyperplane in the input space keeping in consideration that all points need to be correctly classified. We have applied perceptron on the Image Data set. So, for this we have taken a random weight vector initially and keep changing the weight according to the missclassified points using gradient descent technique.

We have made the confusion matrix which tells the accuracy of the identification done for every Image class. The ROC plot tells the accuracy of the image classification by plotting the graph between TPR and FPR. DET graph tell the accuracy of the model by comparing the Miss & False Alarm.

Perceptron with Whole Var. Normalized Data Accuracy 74.1%

		Actual Class		
		Highway	Street	Buildings
Predicted	Highway	35	13	5
	Street	1	20	1
	Buildings	4	7	34

Perceptron with Within Class Normalized Data Accuracy 82.5%

		Actual Class		
		Highway	Street	Buildings
Predicted	Highway	36	7	4
	Street	4	30	3
	Buildings	0	3	33

Figure 1.7: Confusion Matrix of Perceptron based classifier

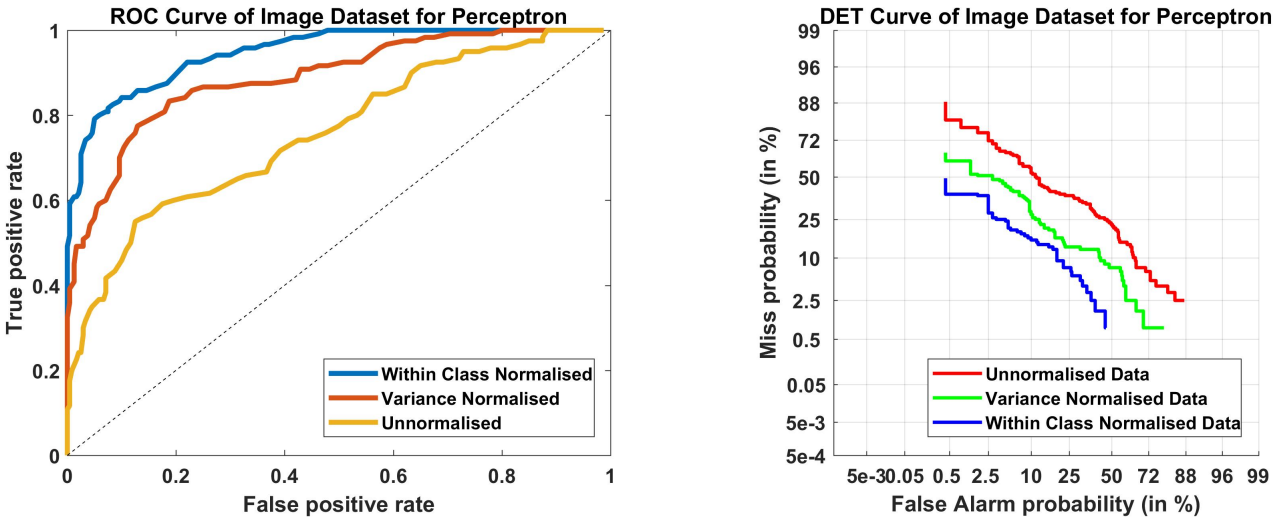


Figure 1.8: ROC & DET of Perceptron based classifier

Observations :-
Perceptron gives us a linear hyperplane, hence it will not be able to distinguish the variances between the class in image dataset and we were getting 44.5% accuray. And whe we did variance normalized on whole data set we got 74% accuray. But when we did within class variance normalisation we were getting 82.5% accuracy.

1.5 Support vector machine based classifier

SVM is technique of the maximising the margin between the classes with the help of the support vectors. So in this we have used the libsvm library to implement the SVM. In this we have used the linear kernel and gaussian kernel for decinding seperating plane. We have implemented the SVM on the Speech and Hand written Data set. In this we have converted all the sequence of speech and handwritten dataset to equal fixed length before applying to the SVM.

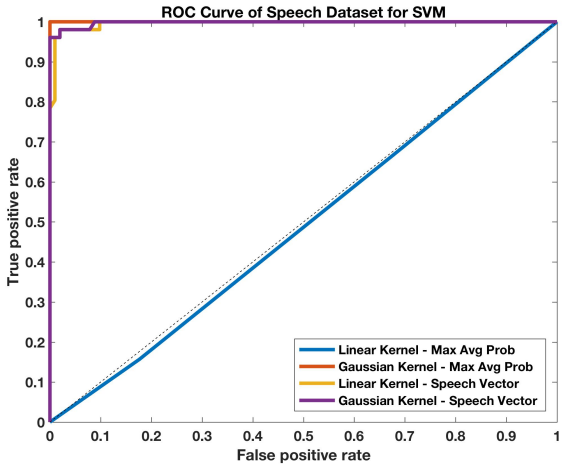
Speech - Linear Kernel & Speech Vector with Acc. 96.07%

		Actual Class		
		Digit 1	Digit o	Digit z
Predicted	Digit 1	16	1	0
	Digit o	1	16	0
	Digit z	0	0	17

Speech - Gaussian Kernel & Max Avg Prob with Acc. 100%

		Actual Class		
		Digit 1	Digit o	Digit z
Predicted	Digit 1	17	0	0
	Digit o	0	17	0
	Digit z	0	0	17

Figure 1.9: Confusion Matrix of Speech Dataset for SVM



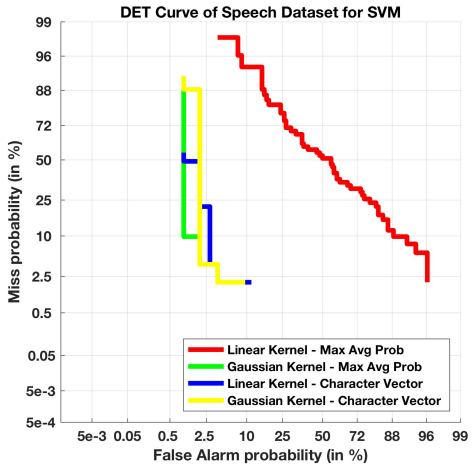


Figure 1.10: ROC & DET of Speech Dataset for SVM

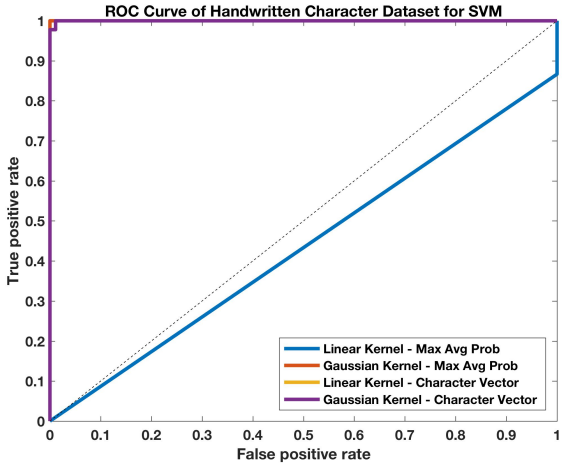
Handwritten - Linear Kernel & Character Vector with Acc. 97.77%

		Actual Class		
		Letter ai	Letter bA	Letter lA
Predicted	Letter ai	15	0	0
	Letter bA	0	14	0
	Letter lA	0	1	15

Handwritten - Gaussian Kernel & Max Avg Probability with Acc. 100%

		Actual Class		
		Letter ai	Letter bA	Letter lA
Predicted	Letter ai	15	0	0
	Letter bA	0	15	0
	Letter lA	0	0	15

Figure 1.11: Confusion Matrix of Handwritten Letters for SVM



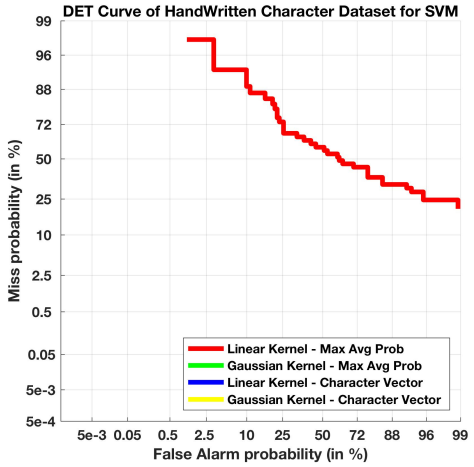


Figure 1.12: ROC & DET of Handwritten Letters for SVM

Observations :-

- For Speech Dataset,when we used linear kernel and then after taking maximum average probability we got 33% accuracy as these speechs are not linearly seperable, but when we coverted

each speech file into one vecter then we got 96% accuracy. And when we used gaussian kernel we got 100% accuracy as gaussian will be able to detect varainces between classes of speech.

- For Handwritten Character Dataset, we first did mean and variance normalisation and then extracted features and then we used linear kernel and then after taking maximum average probability we got 30% accuracy as these characters are not linearly seperable, but when we coverted each character file into one vecter then we got 97.77% accuracy. And when we used gaussian kernel we got 100% accuracy as gaussian will be able to detect varainces between classes of handwritten character.

1.6 Multi-layer feed-forward neural network based classifier

Feedforward neural network is a technique which works on the idea that if the machine makes mistake in the output(i.e.Error) then the error is corrected by backpropogating the error and adjusting the weight accordingly. It keeps on improving the model by correcting the error. In this we have taken 10 hidden layers. To implement the neural network we have the used the inbuilt function of matlab. We have applied the neural network on the Speech and Handwritten data set. In this we have converted all the sequence of speech and handwritten dataset to equal fixed length before applying to the NN.

Speech - NN & Speech Vector with Acc. 96.07%

		Actual Class		
		Digit 1	Digit o	Digit z
Predicted	Digit 1	13	1	0
	Digit o	0	20	1
	Digit z	0	0	16

Speech - NN & Max Avg Prob with Acc. 100%

		Actual Class		
		Digit 1	Digit o	Digit z
Predicted	Digit 1	17	0	0
	Digit o	0	17	0
	Digit z	0	0	17

Figure 1.13: Confusion Matrix of Speech Dataset for NN

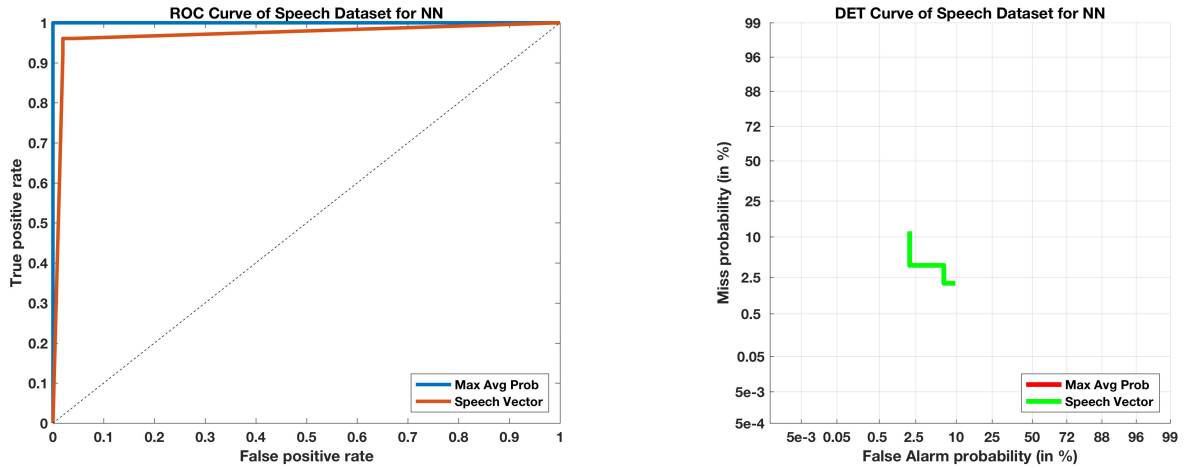


Figure 1.14: ROC & DET of Speech Dataset for NN

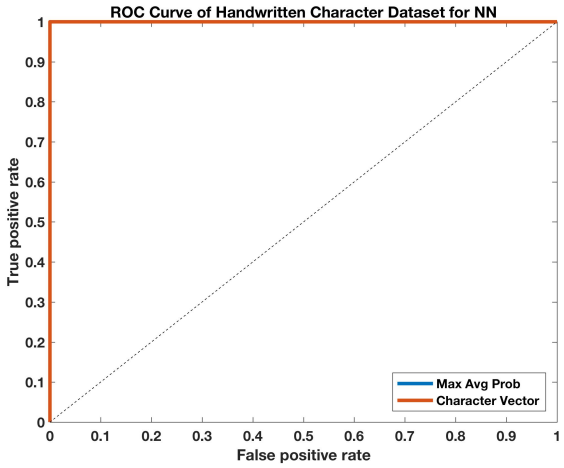
Handwritten - NN & Max Avg Probability with Acc. 100%

Predicted \ Actual Class	Letter ai	Letter bA	Letter lA
Letter ai	15	0	0
Letter bA	0	15	0
Letter lA	0	0	15

Handwritten - NN & Character Vector with Acc. 100%

Predicted \ Actual Class	Letter ai	Letter bA	Letter lA
Letter ai	14	0	0
Letter bA	0	11	0
Letter lA	0	0	20

Figure 1.15: Confusion Matrix of Handwritten Letters for NN



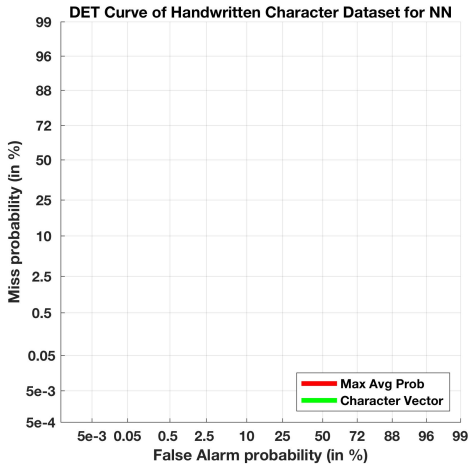


Figure 1.16: ROC & DET of Handwritten Letters for NN

Observations :-

- For Speech Dataset,when we used 10 hidden layer neural network and then after taking maximum average probability we got 96% accuracy, but when we coverted each speech file into one vecter then we got 100% accuracy.

- For Handwritten Character Dataset, we first did mean and variance normalisation and then extracted features and then used 10 hidden layer neural network for training model, and then after taking maximum average probability we got 100% accuracy, and when we converted each speech file into one vecter then we also got 100% accuracy.