# Pattern Recognition Assignment V

Muhammad Arsath K F(CS16S035) and Srihari Maruthachalam (CS16S024) Group 17

November 23, 2017

# 1 Classification using Parzen window

## 1.1 Classification on Cubic Parzen window method with unnormalized features

### Intuition

The idea is to classify the images using Parzen window with the unnormalized features. Here, we pick the neighbouring data points from the train data which are covered in the Parzen window and classify with voting technique.

| | | Target Classes | | |
|---|---|---|---|---|
| | | **1** | **2** | **3** |
| Predicted Classes | **1** | **55** | 12 | 0 |
| | **2** | 6 | **60** | 1 |
| | **3** | 47 | 41 | **98** |

Table 1: Confusion matrix on Cubic Parzen window method using unnormalized features when h = 1

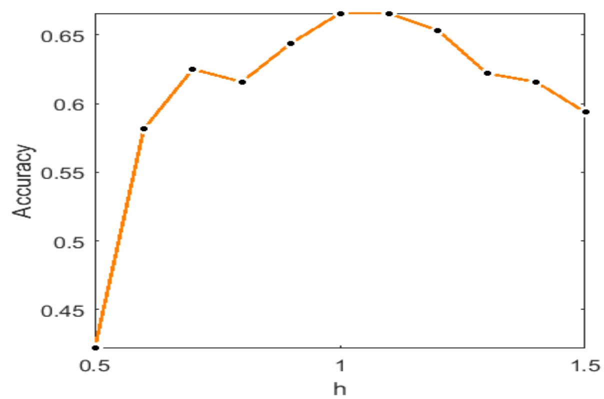



Figure 1: Classification accuracy on Cubic Parzen window method using unnormalized features for various h values

### Inference

- As we increase the width, accuracy of the model gets improved. At h = 1, we are getting 67% accuracy.
- Beyond that, the accuracy of the model started declining. *This is expected, as more number of data points from non target class falls inside the hyper cube.*

## 1.2 Classification on Cubic Parzen window method with normalized features

### Intuition

Every dimension in the feature vector is normalized. The idea is to classify the images using Parzen window with the normalized features. Here, we pick the neighbouring data points from the train data which are covered by the Parzen window and classify with voting technique.

| | | Target Classes | | |
|---|---|---|---|---|
| | | **1** | **2** | **3** |
| Predicted Classes | **1** | **88** | 19 | 2 |
| | **2** | 5 | **75** | 8 |
| | **3** | 15 | 19 | **89** |

Table 2: Confusion matrix on Cubic Parzen window method using normalized features when h = 2.335

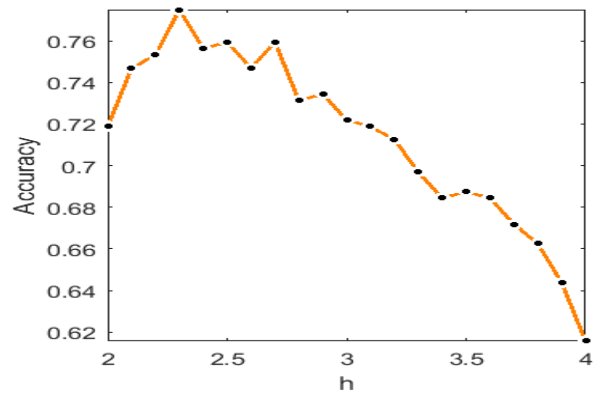



Figure 2: Classification accuracy on Cubic Parzen window method using normalized features for various h values

**Inference**

- As we increase the width, accuracy of the model gets improved. At h = 2.335, we are getting 78% accuracy.
- Beyond that, accuracy of the model started declining. *This is expected, as more number of data points from non target class falls inside the hyper cube.*
- Another inference is that normalized features performs better than that of unnormalized feature vectors.

## 1.3 Classification on Parzen window method using Gaussian Kernel

**Intuition**

Every dimension in feature vector is normalized. The idea is to classify the images using Parzen window with Gaussian Kernel. Here, we predict the test data using likelihood of each class. Images are classified based on voting technique.

| | | Target Classes | | |
|---|---|---|---|---|
| | | **1** | **2** | **3** |
| Predicted Classes | **1** | **94** | 5 | 0 |
| | **2** | 12 | **95** | 11 |
| | **3** | 2 | 13 | **88** |

Table 3: Confusion matrix on Parzen window method using Gaussian Kernel when h = 0.02

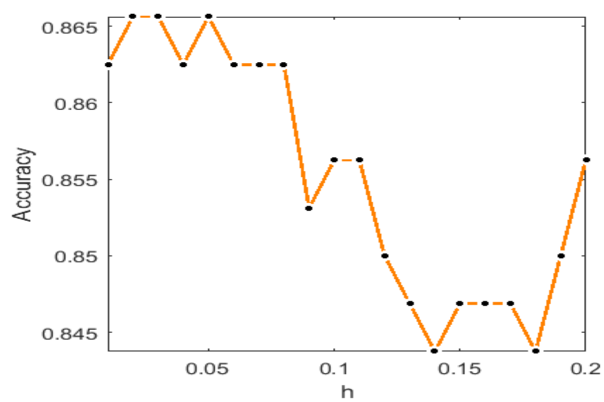



Figure 3: Classification accuracy on Parzen window method using Gaussian Kernel for various h values

**Inference**

- For Gaussian kernel, Accuracy of the model varies from 84% to 86%. Highest accuracy is obtained at h = 0.02 with accuracy of 86.6%.
- we inferred that Gaussian Kernel based Parzen window classifier outperforms both normalized and unnormalized Cubic Parzen window based classifier.

### 1.4 Performance analysis among various Parzen Window Methods

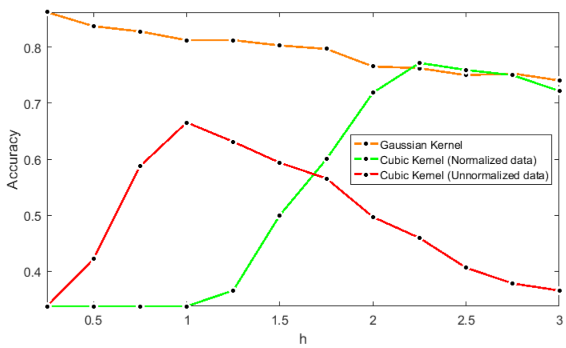



Figure 4: Accuracy obtained for various Parzen Window Methods

#### Inference

- As we can see that, Gaussian kernel based Parzen window classifier performs well for the values of h in the range of 0.1 to 2.
- As we increase h value in Cubic kernel based Parzen window classifier, performance is comparable with the Gaussian kernel based Parzen window classifier in the case of normalized feature vectors.

## 2 Fisher Linear Discriminant Analysis(LDA)

### Intuition

We computed the within-class covariance matrix and between class covariance matrix. Then we found the linear projection matrix using Fisher's LDA and projected all the feature vectors using the projection matrix. The classification model is built for the projected feature vectors and classification is performed.

### 2.1 Classification using k Nearest Neighbours(k NN)

| | | Target Classes | | |
|---|---|---|---|---|
| | | **1** | **2** | **3** |
| Predicted Classes | **1** | **92** | 5 | 0 |
| | **2** | 9 | **96** | 13 |
| | **3** | 7 | 12 | **86** |

Table 4: Confusion matrix on k NN classifier with k = 5 after LDA

#### Intuition

To perform classical k NN based classification on the projected feature vectors.

#### Inference

- Model is performing well with small values of k(say k = 2).
- Accuracy of the model is stable even after increasing the k value.
- We achieved the best accuracy of 86% with minimal k value at k = 5.

### 2.2 Classification using Bayesian Classifier

| | | Target Classes | | |
|---|---|---|---|---|
| | | **1** | **2** | **3** |
| Predicted Classes | **1** | **81** | 1 | 0 |
| | **2** | 15 | **99** | 12 |
| | **3** | 12 | 13 | **87** |

Table 5: Confusion matrix on Bayes' classifier after LDA

#### Intuition

To perform Bayes' classification with full covariance on the projected feature vectors.

#### Inference

- We achieved the best accuracy of 84% using Bayes' classifier.

## Inference

- It is quite surprising that k NN classifier gives comparatively better accuracy than that of Bayes' classifier. Still, improvement is very minimal.
- Performance of Parzen window with Gaussian Kernel is almost same as that of both k NN based and Bayes' classifier after LDA projection.
- We also inferred that irrespective of the classifier, performance of LDA is stable.
- Normalizing the feature vectors doesn't give impact on the performance of the model.

# 3 Speech and Online Character Recognition

## Preprocessing of Speech

Sequential patterns are learned properly in Recurrent Neural Networks or Hidden Markov Models. Support Vector Machine or Neural Networks fails to work in the same situation as they can't capture sequential patterns. As we got varying length sequence for speech, we can't incorporate feature vectors to classify using Support Vector Machine or Neural Networks. In order to make the feature vectors usable for SVM or Neural Networks, we have to preprocess the speech data. The steps followed for preprocessing are,

- Build an Universal Background Model(UBM) from the development data.
- Divide the feature vectors of each class into 100 segments.
- Adapt the new means for each segments of speech data.
- Repeat the adaptation process for all the classes.
- Adapted means will act as train data for SVM or Neural Networks.
- With the UBM from development data, adapt new means for the test speech data.
- Adapted means will act as test data for SVM or Neural Networks.

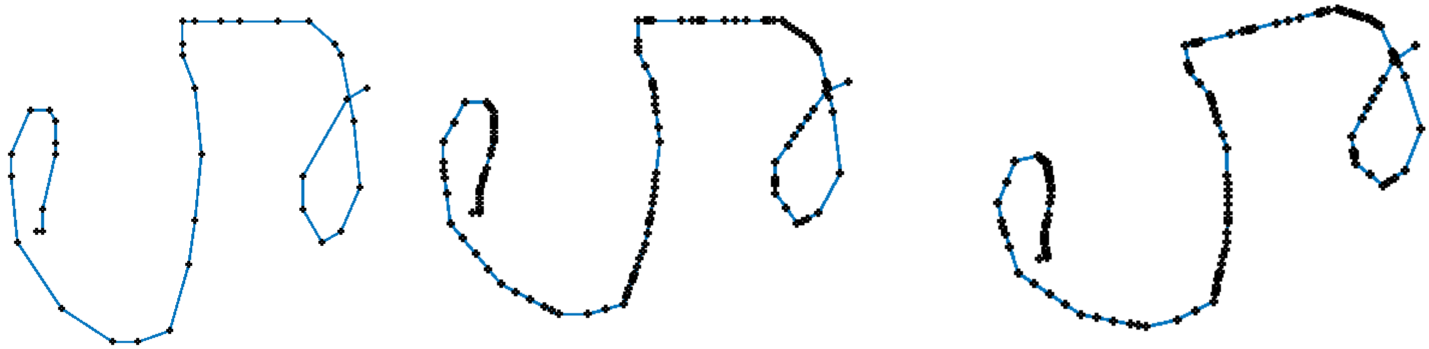
## Preprocessing for Online Character Recognition(OCR)

### Extrapolation of points

As we have the online characters of varying length, we can't use it in SVM or Neural Networks directly. We had an intuition of extrapolating the points in the character. All the feature vectors are extrapolated to the length of the longest feature vector.

### Generating more Online Character

We had an intituiton of generating more number of character data by rotating each characters. As a result, we obtained ample amount of train data.



(a) Given Character

(b) Extrapolation of points in the character

(c) Generation of Characters using Rotation Matrix

Figure 5: Prepocessing of Online Character

### Feature Extraction for OCR

We extracted Histogram of Oriented Gradients(HOG) features from online characters.

## 4 Perceptron Classification

### Online Character Recognition

| | | Target Classes | | |
|---|---|---|---|---|
| | | **1** | **2** | **3** |
| Predicted Classes | **1** | **74** | 1 | 7 |
| | **2** | 2 | **51** | 24 |
| | **3** | 2 | 9 | **37** |

Table 6: Confusion Matrix of Perceptron Based Classifier for OCR

#### Intuition

The idea is to classify Online Character Recognition using perceptron. To learn weights of perceptron using gradient descent weight update formula. The weight parameters are learned till we get the maximum classification accuracy.

#### Inference

- We found that learning rate of 5e-07 works best for the given dataset.
- We obtained 78% accuracy for Online Character Recognition.

### Image Classification

| | | Target Classes | | |
|---|---|---|---|---|
| | | **1** | **2** | **3** |
| Predicted Classes | **1** | **52** | 36 | 1 |
| | **2** | 1 | **12** | 1 |
| | **3** | 0 | 1 | **54** |

Table 7: Confusion Matrix of Perceptron Based Classifier for Image Dataset

#### Intuition

The idea is to classify Images using perceptron. To learn weights of perceptron using gradient descent weight update formula. The weight parameters are learned till we get maximum classification accuracy.

#### Inference

- We found that learning rate of 5e-07 works best for the given dataset.
- We obtained 74% accuracy for Online Character Recognition.

## Receiver Operating Curve for OCR and Image Dataset

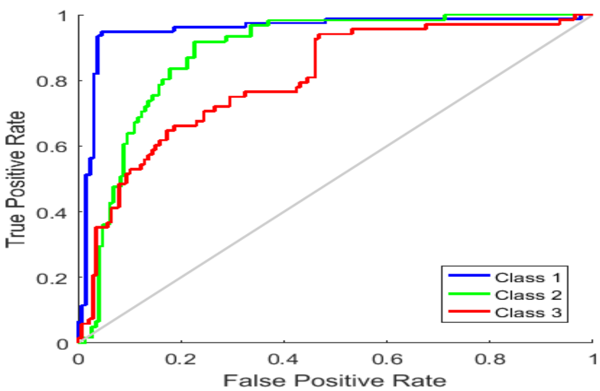



Figure 6: Receiver Operating Curve for OCR

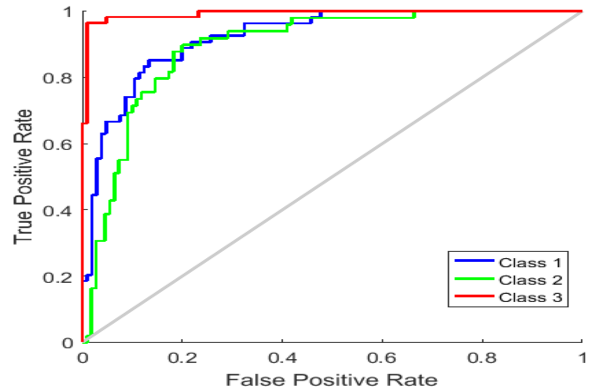



Figure 7: Receiver Operating Curve for Image Dataset

## Inference

- We found that mountain images are classified as coast class to very large extent. We inspected the images visually and inferred the following.
- Some of the mountain images contains water and coast images contains rocks/mountains.

Figure 8: Coast image with mountain

Figure 9: Mountain image with coast

- We inferred that the color feature plays a vital role in image classification. For mountain and coast images, 'blue' is prominent color.

Figure 10: Coast image with high intensity of color 'Blue'

Figure 11: Mountain image with high intensity of color 'Blue'

# 5 Support Vector Machine(SVM) Classification

### Online Character Recognition

| | | Target Classes | | |
|---|---|---|---|---|
| | | **1** | **2** | **3** |
| Predicted Classes | **1** | **0** | 0 | 0 |
| | **2** | 11 | **27** | 3 |
| | **3** | 19 | 2 | **26** |

Table 8: Confusion Matrix for OCR with radial basis kernel based SVM with $\nu = 0.087$

#### Intuition

The idea is to classify Online Character Recognition using $\nu$ SVM. We tried linear SVM, polynomial kernel based SVM, sigmoidal kernel based SVM and radial basis kernel based SVM.

#### Inference

- We obtained 60% accuracy in linear SVM using $\nu = 0.043$; 60% accuracy in polynomial kernel based SVM using $\nu = 0.103$; 61% accuracy in radial basis kernel based SVM using $\nu = 0.087$ and 60% accuracy in sigmoidal kernel based SVM using $\nu = 0.59$.

### Speech Classification

| | | Target Classes | | |
|---|---|---|---|---|
| | | **1** | **2** | **3** |
| Predicted Classes | **1** | **8** | 3 | 6 |
| | **2** | 6 | **10** | 4 |
| | **3** | 3 | 4 | **7** |

Table 9: Confusion Matrix for Speech with radial basis kernel based SVM with $\nu = 0.017$

#### Intuition

The idea is to classify speech using $\nu$ SVM. We tried linear SVM, polynomial kernel based SVM, sigmoidal kernel based SVM and radial basis kernel based SVM.

#### Inference

- We obtained 49% accuracy in linear SVM using $\nu = 0.051$; 41% accuracy in polynomial kernel based SVM using $\nu = 0.003$; 49% accuracy in radial basis kernel based SVM using $\nu = 0.017$ and 51% accuracy in sigmoidal kernel based SVM using $\nu = 0.519$.

## 5.1 Performance analysis among various Kernel Methods

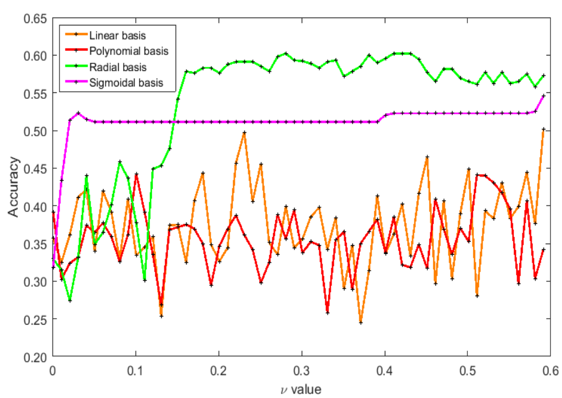



Figure 12: Classification accuracy on OCR with various SVMs and $\nu$ values

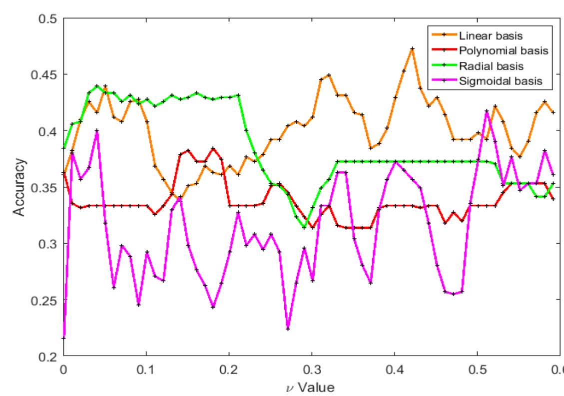



Figure 13: Classification accuracy on Speech with various SVMs and $\nu$ values

# 6 Multi Layer Feed Forward Neural Network

### Online Character Recognition

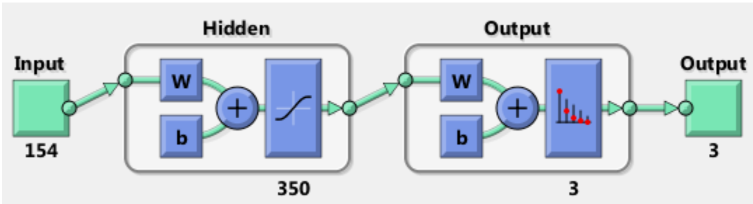



Figure 14: Neural Network Architecture for Online Character Recognition

| | | Target Classes | | |
|---|---|---|---|---|
| | | **1** | **2** | **3** |
| Predicted Classes | **1** | **67** | 0 | 0 |
| | **2** | 0 | **69** | 0 |
| | **3** | 0 | 0 | **71** |

Table 10: Confusion Matrix for OCR with MLFFNN

### Intuition

To build Multi Layer Feed Forward Neural Network(MLFFNN) with 1 hidden layer and 350 nodes. We use sigmoidal activation function in the hidden layer and softmax in output layer. We used cross entropy as error function.

### Inference

- The learning rate $\eta$ = 5e-07 works well for OCR.
- We obtained cent percent accuracy for OCR using MLFFNN.

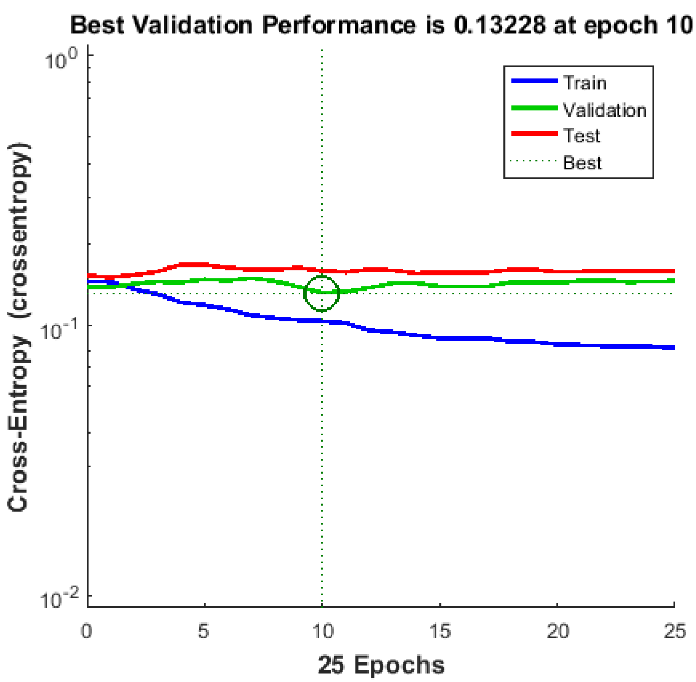



Figure 15: Performance of MLFFNN for OCR

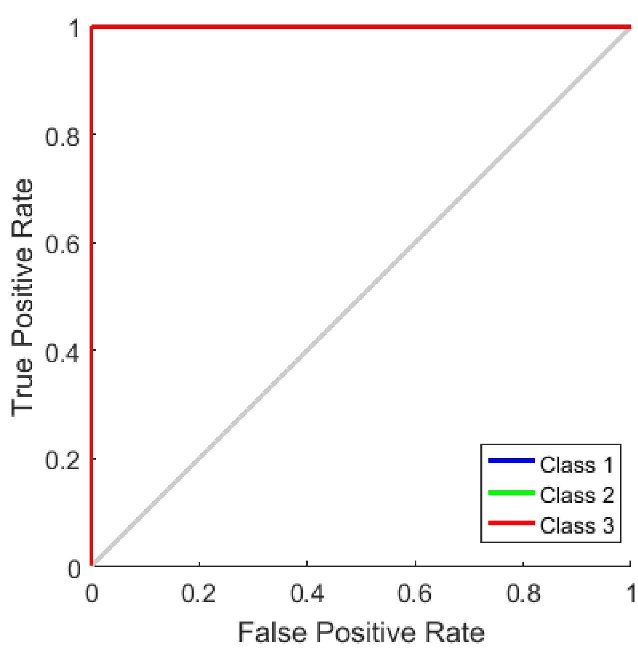



Figure 16: ROC for OCR using MLFFNN

### Speech Classification

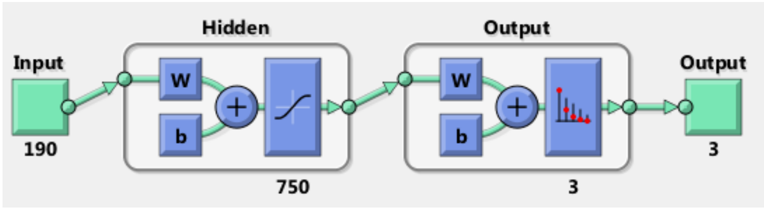



Figure 17: Neural Network Architecture for Speech Classification

| | | Target Classes | | |
|---|---|---|---|---|
| | | **1** | **2** | **3** |
| Predicted Classes | **1** | **32** | 22 | 11 |
| | **2** | 24 | **47** | 15 |
| | **3** | 26 | 13 | **55** |

Table 11: Confusion Matrix for Speech Classification with MLFFNN

#### Intuition

To build Multi Layer Feed Forward Neural Network(MLFFNN) with 1 hidden layer and 750 nodes. We use sigmoidal activation function in the hidden layer and softmax in output layer. We used cross entropy as error function.

#### Inference

- The learning rate $\eta$ = 5e-07 works well for OCR.
- We obtained 55% accuracy for Speech classification using MLFFNN.
- From the performance graph, we can see that MLFFNN over-fitted when we increase the Epoch counts.

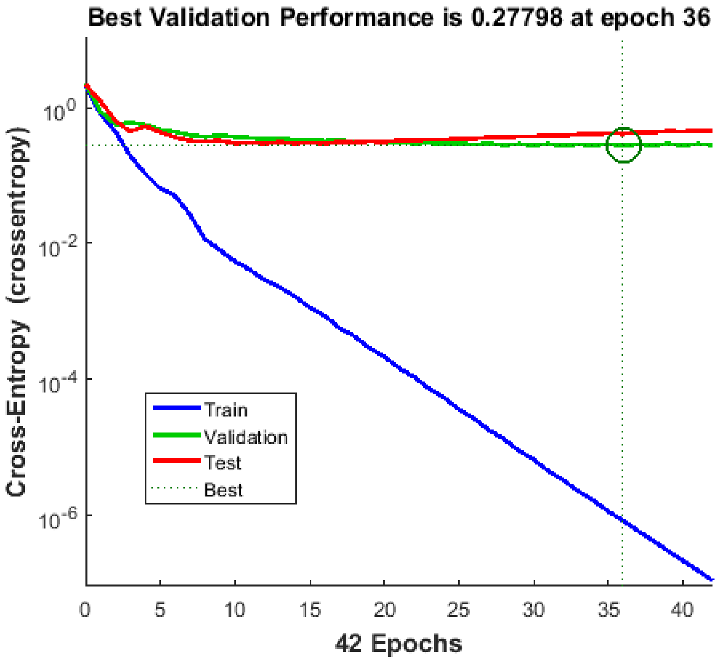



Figure 18: Performance of MLFFNN for Speech Classification

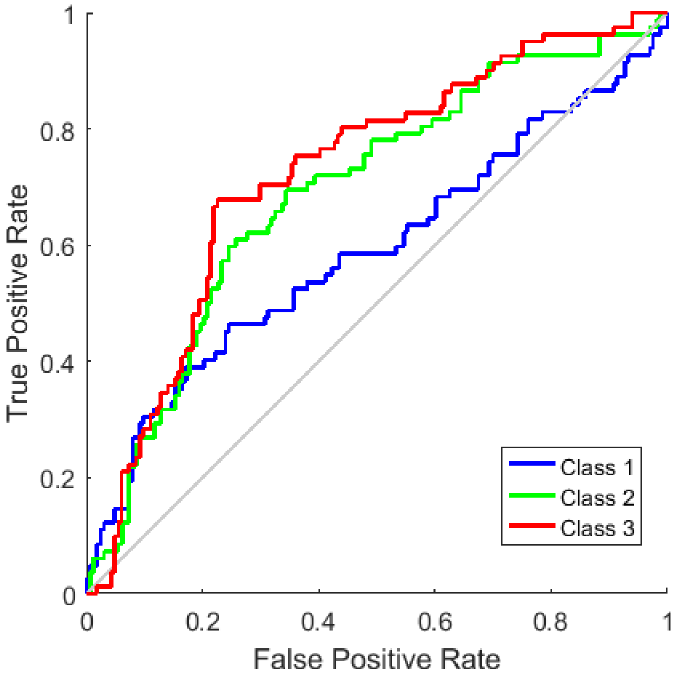



Figure 19: ROC for Speech Classification using MLFFNN

# 7 References


- Bishop, Christopher M., "Pattern Recognition and Machine Learning", Information Science and Statistics, 2006.
- R.O.Duda, P.E.Hart and D.G.Stork, "Pattern Classification", John Wiley, 2001.
- M. Pastor, A. Toselli, and E. Vidal, "Writing speed normalization for on-line handwritten text recognition", Eighth International Conference on Document Analysis and Recognition, 2005.
- Martin, A and R. Doddington, George and Kamm, Terri and Ordowski, Mark and Przybocki, Mark, "The DET curve in assessment of decision task performance", Fifth European Conference on Speech Communication and Technology, EUROSPEECH 1997.
- https://www.mathworks.com/
- NIST DETWare