Report: Assignment 4

Submitted by: Arulkumar S, Shitanshu Kusmakar
CS15S023, ED15F003

25-Nov-2015

1 Introduction

1.1 Problem Definition

In our previous assignments we have focused upon the use of probability distributions having given functional forms governed by parameters, whose values are determined from the data set. This approach was termed as parametric approach. However, the approach has an inherent limitation, that the chosen density function can be a poor model of the given data, which might lead to compromise in the prediction accuracy of the system. In contrast, non-parametric approach doesn't make any assumptions about the probability distribution of the data and assumes parameters based on the training data. However, non-parametric approaches can be computationally slower than parametric approaches, owing to the fact that all training data is stored in non-parametric form. Nearest neighbour method and parzen windows are such examples of the non-parametric approach.

Linear models of the form as shown in equation 1 corresponds to a two class model and can be readily learned using error function minimization.

$$y(x_n) = W^T \phi(x_n) + b \tag{1}$$

In Fisher linear discriminant analysis (FLDA) the data is first projected to a lower dimensional space such that the class separation is maximum. The direction that maximises the separation between the class is computed using the scatter matrices (between-class and within-class scatter matrix). Reducing the dimensionality of the data improves the computational efficiency of the pattern classification algorithm. FLDA is generally used as a dimensional reduction technique for pre-processing of the data for pattern classification. Another, approach called the perceptron algorithm is based upon determining the weight parameters for training data such that the number of misclassified patterns are minimised using the gradient of the error function. The weights are updated using one of class representation such that the target value $t = +1$ for patterns in class C_1 will have $W^T\phi(x) \geq 0$ and target value $t = -1$ for pattern in class C_2 will have $W^T\phi(x) \leq 0$ for a appropriate choice of the activation function. However, the accuracy of the algorithm is based upon linear separability of the data. Algorithms such as support vector machines (SVM) work by finding the hyperplane that maximizes the class separation in the data. SVM can be viewed as an optimization problem. For, pattern which are correctly classified $t_n y(x_n) \geq 0$ for all n, so that the distance of a point x_n from the decision surface is given by $\frac{t_n y(x_n)}{\|W\|}$. The parameters W and b are optimized to get the maximum distance. SVM's can result in better classification accuracy

by incorporating a slack term in optimization such that some training error will be permitted and thus allowing greater generalization to the data.

In addition Gaussian mixture model (GMM), hidden markov model (HMM) and DTW algorithms are also implemented on multi class data set and performance of all the algorithms are reported using ROC and DET curve for the respective systems.

2 Approach to the problem

2.1 What is the given information

Parametric Methods: A Gaussian mixture model has to be build to fit multivariate normal distributions. A gaussian mixture distribution can be written as a linear superposition of Gaussians in the form as shown in equation 2.

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x/\mu_k, \Sigma_k) \tag{2}$$

where, x is the d-component feature vector, μ is the mean vector with d-components, Σ is the $d \times d$ covariance matrix, and π_k is the mixing parameter.

Also, the HMM and DTW algorithm has to be implemented on the given data sets and performance measures for all the pattern recognition systems has to be calculated.

Non-Parametric Methods: The classifiers based on Bayesian approach use the equation 3 to estimate the posterior probability.

$$P(w_i/x) = \frac{P(\frac{x}{w_i})P(w_i)}{\Sigma_j P(\frac{x}{w_j})P(w_j)} \tag{3}$$

to find the posterior we need $P(\frac{x}{w_i})$ and $P(w_i)$. The parzen window method estimates these probabilities by putting a window (with defined width) using a kernel function (hypersphere and Gaussian). Then, a window function is placed on every datum and the number of observations falling inside the window are determined. The probability density estimate or the parzen window estimate is defined as shown in equation 4.

$$P(x) = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{V} k\Big(\frac{x - x_n}{h}\Big) \tag{4}$$

where, V is the volume of the window and $k(\cdot)$ is the kernel function.

In, parzen window method we fix the volume and find K from the data. However, if we fix K and determine V, it is called k-nearest neighbour method. The posterior probability for each class in k-nearest neighbour method is determined as shown in equation 5.

$$P(\frac{w_i}{x}) = \frac{k_i}{k} \tag{5}$$

where, k_i are the points belonging to class w_i.

Another approach to classification can be viewed in terms of dimensionality reduction using FLDA. In, FLDA the data is projected along the direction of maximum class separation. Variance within the classes and among the classes is used to calculate the fisher criterion function as shown in equation 6.

$$J(w) = \frac{w^T S_B w}{w^T S_w w} \tag{6}$$

where, S_B, and S_w is the between class and within class covariance matrix.

The perceptron algorithm on the other hand, makes an assumption that the data is linearly separable. Where, the weights are updated using the gradient of the error function as shown in equation 7.

$$w^{\tau+1} = w^{\tau} + \eta \bigtriangledown E_p(w) \tag{7}$$

However, another linear classification algorithm based on calculating the hyperplanes separating the two data is based on maximizing the margin separating the hyperplanes and is termed as support vector machines. The optimization problem for SVM can be written as shown in equation 8.

$$argminC \sum_{n=1}^{N} \epsilon_n + \frac{1}{2}\|w\|^2 \tag{8}$$

where, C is the regularization term and ϵ is the slack term.

2.2 Approach

The approach to the problem was based on following things

- The data was divided randomly into training and test set. 70% of the data was selected as training set and 30% as test.

- A k-component Gaussian mixture model was trained using the training data. The parameter k was determined empirically, and it was set to 10.

- The initialization of the parameters of the GMM were done using k-means clustering. The figure 1 (a) shows the convergence of the k-means algorithm based on calculating the squared error.

- The parameters are then updated using the EM algorithm and log likelihood of the observed data is calculated in each step. The log likelihood is as shown in equation 9.

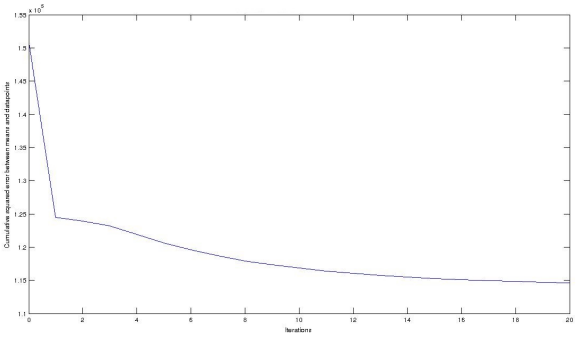
$$lnp(X/\mu, \Sigma, \pi) = \sum_{n=1}^{N} ln\{\Sigma_{k=1}^{K} \pi_k \mathcal{N}(x_n/\mu_k, \Sigma_k)\} \tag{9}$$

- The convergence of the algorithm is decided by fixing the number of iterations for the EM step. The figure 1 (b) shows the convergence of the algorithm based on log likelihood calculation.

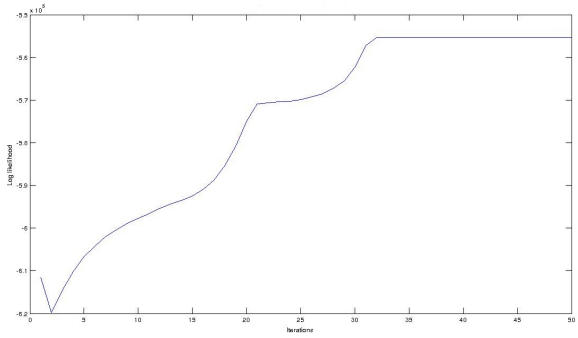
- The testing is then performed using the GMM parameters (π_k, μ_k, Σ_k) for every image. The feature vectors corresponding to a particular image were assigned to individual classes based on maximum likelihood and the class for an image is determined using mode of classes for the feature vectors.

- The ROC and DET curves along with the performance measures for each data set are reported.

- The implementation of the DTW is followed by the preparation of the data into sequences represented by the number of classes using k-means clustering.



(a) Convergence of k-means



(b) Convergence of EM algorithm

Figure 1: Convergence of k-means and EM algorithm

Non-parametric methods :

- A Bayesian classifier is built using the kernel estimator method with hypersphere as a region and also a Gaussian kernel. The parameter h is chosen empirically.

- Similarly, a Bayesian classifier is built using k-nearest neighbour, where the number of datum points k in the region is fixed. The parameter k is chosen empirically. For, any test datum we find its k nearest neighbour and define the class based on the class of the majority of the points enclosed in the region as shown by equation 5.

- In approach based on using FLDA based classifier, we first reduce the dimension using FLDA such that the data is projected on a direction which has maximum class separability. After using FLDA we perform classification using k-nearest neighbours. The steps in FLDA are as follows

  - d-dimensional mean vectors are calculated for every class.
  - The between class S_B and within class S_w scatter matrices are calculated.
  - We, compute the eigen vectors and the eigen values for the matrix $S_w^{-1}S_B$.
  - The $d \times k$ projection matrix W is computed using the eigen vectors corresponding to the top k eigen values.
  - The input features are then projected to a new subspace using the projection matrix W as shown in equation 10.

$$Y = X \times W \tag{10}$$

- For, perceptron and SVM based classifier the data is first processed for every image by converting all the features into a row vector. The transformed features are then normalized to have a zero mean and standard deviation of 1.

- The perceptron based classifier is a supervised learning algorithm, where weights for every data point is updated in every iteration until all the patterns are classified with the required accuracy. For, i^{th} feature in the training data, the output is calculated as shown in equation 11, where $f(\cdot)$ is the activation function. The activation function used is sigmoid function and the learning rate

$\eta = 0.0025$ is chosen heuristically. The weights will be updated as shown in equation 7. The classification models for all classes are developed using $1vsrest$ training.

$$y_i = f[w^T \cdot x_j] \tag{11}$$

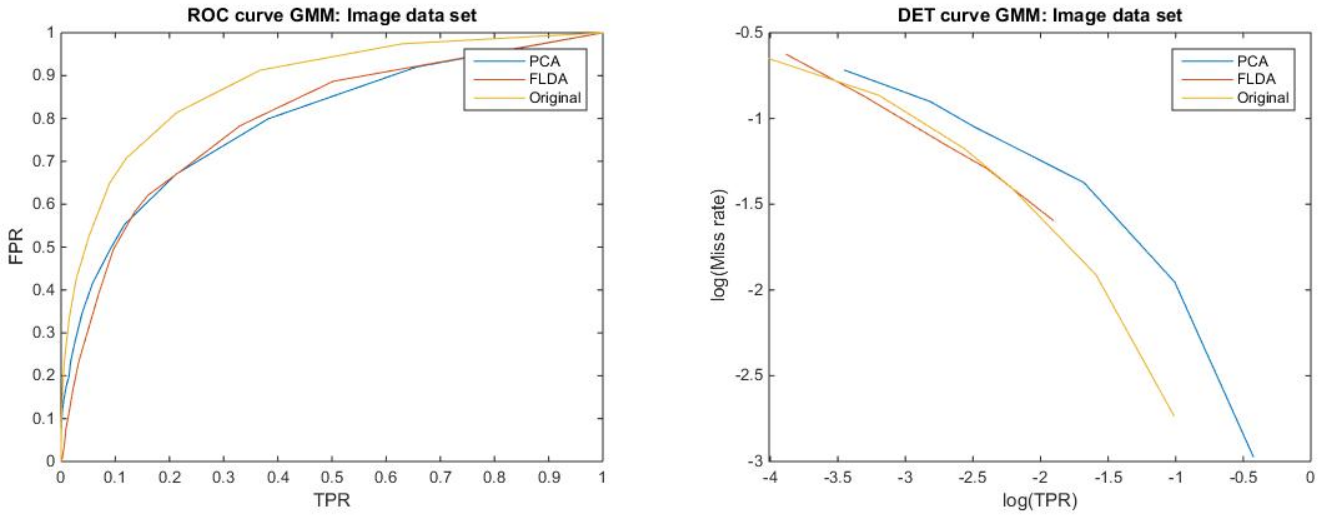
- The support vector machine classifier (SVM) is implemented using the $libSVM$ library. The training is performed using $1vsrest$ method and a linear kernel is used. ROC and DET curves are reported using the generated classification models. Also, the models are also validated based on the number of support vectors for the model.

3 Results & Observations

3.1 GMM: Image Data Set (All classes)

Gaussian mixture model with 10-components is implemented on the image data set. The figure 2 shows the ROC curve for the GMM. The figure 2 (a) shows the performance of the GMM based classifier and figure 2 (b) shows the DET curve for the GMM algoirthm obtained on the image dataset. The plots are shown as a PCA, FLDA and original image data to show the performance of the GMM algorithm for three different cases. On, application of PCA and FLDA the overall classification accuracy is reduced. PCA and FLDA performs better when the data is of very large dimensionality as the image data is only 23 dimensions, application of PCA and FLDA doesn't result in performance improvement.

An overall accuracy from $55 - 60\%$ is achieved using the Gaussian mixture model on the original image data. As, can be seen from the confusion matrix Table 1 that there is a significant class overlap between $(class5Opencountry, class7forest)$, $(class8street, class3insidecity)$ and $(class2highway, class1coast)$. On verifying the images it is observed that the above mentioned set of images have significantly similar content $e.g$ highway and coast both images have lots of blue patches representing sky. Thus, resulting in indiscriminatry features for the set of images.



(a) ROC curve for all classes: Image data set

(b) DET curve for all classes: Image data set

Figure 2: ROC and DET curves for image data set for GMM

Table 1 and 2 shows the confusion matrix and the performance measure for the image data set for 10-component Gaussian mixture model.

Table 1: Confusion matrix. GMM image data set. All classes.

Class	**Coast-Pred**	**Highway-Pred**	**Inside City-Pred**	**Mountain-Pred**	**Open country-Pred**	**Tall Building-Pred**	**Forest-Pred**	**Street-Pred**
Coast-Trgt	73	21	0	1	7	2	1	3
Highway-Trgt	15	36	4	3	1	2	0	17
Inside city-Trgt	5	1	47	1	2	6	0	30
Mountain-Trgt	7	15	1	48	7	5	13	16
Open country-Trgt	19	10	0	3	36	1	52	2
Tall Building-Trgt	8	7	13	5	0	56	2	16
Forest-Trgt	0	1	0	2	7	0	82	16
Street-Trgt	0	1	7	3	0	4	1	72

Table 2: Table showing the performance measures for the classifier. GMM image data set.

Measure	**Coast**	**Highway**	**Inside City**	**Mountain**	**Open country**	**Tall Building**	**Forest**	**Street**
Sensitivity	0.68	0.46	0.51	0.43	0.30	0.52	0.84	0.82
Specificity	0.93	0.93	0.96	0.97	0.96	0.97	0.90	0.87
Precision	0.57	0.40	0.65	0.73	0.60	0.74	0.54	0.44
F1-score	0.62	0.42	0.57	0.54	0.40	0.61	0.66	0.58
Overall Accuracy	0.56							

3.2 HMM: Digits and hand written dataset

Continuous density HMM were performed for the digit and the hand written dataset using the HTK library. The features for the hand written dataset were extracted as shown in equation 12- 13.

$$\acute{x_i} = \frac{x_i - x_{max}}{x_{max} - x_{min}} \tag{12}$$

$$\acute{y_i} = \frac{y_i - y_{max}}{y_{max} - y_{min}} \tag{13}$$

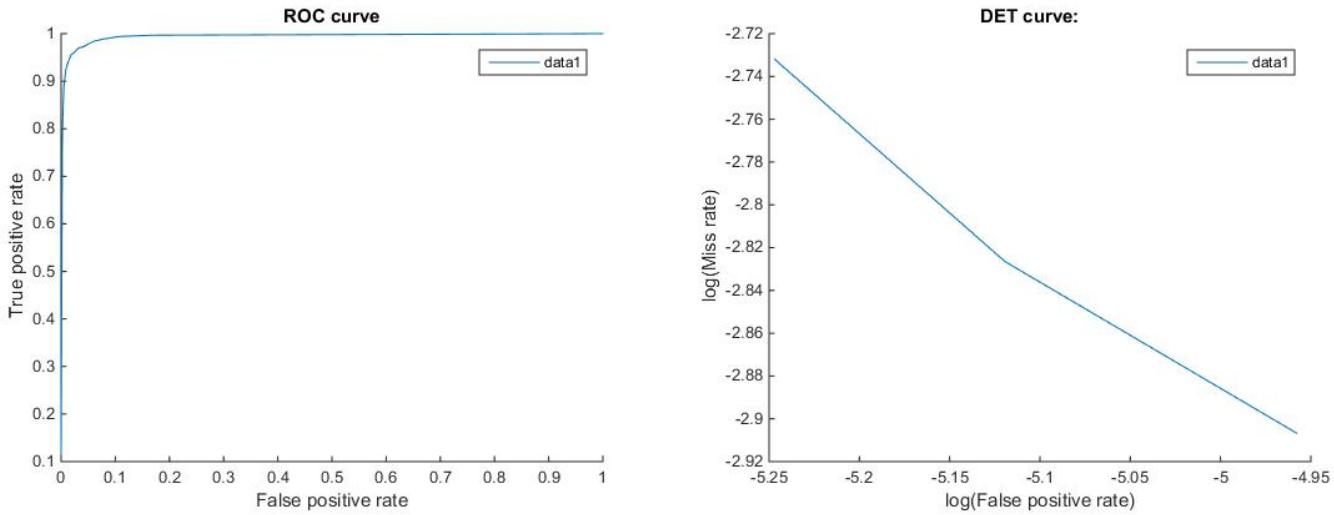
We trained the HMM for approximately 12 iterations, we were able to get a classification accuracy of 70% with digit data set and an accuracy of 60% after normalizing the data as shown above with hand written data set. It was observed that the normalization did not show improvement in classification accuracy of digit data set.

3.3 DTW: Mandi dataset

A discrete time warping algorithm was implemented on the mandi dataset. An overall classification accuracy of $10 - 15\%$ is obtained.

3.4 Parzen method: Image data set

The figure 4 shows the ROC and DET curve for the parzen window based classifier with hypersphere as a region, when FLDA was used to separate the data. An overall accuracy of 93% was obtained.



(a) ROC curve for all classes: Image data set

(b) DET curve for all classes: Image data set

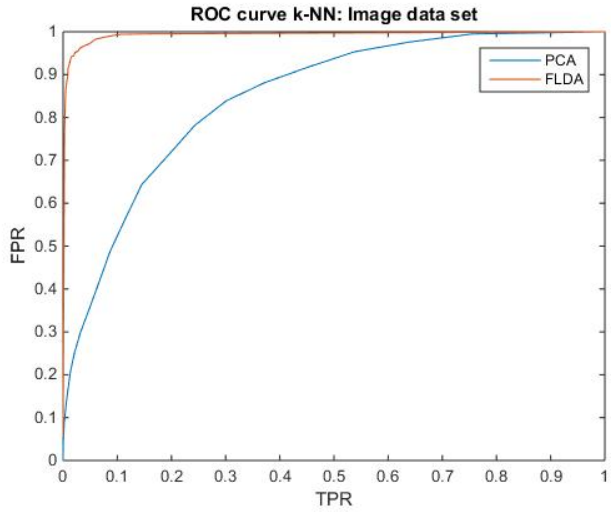
Figure 3: ROC and DET curves for image data set for GMM

3.5 k-nearest neighbour method: Image data set

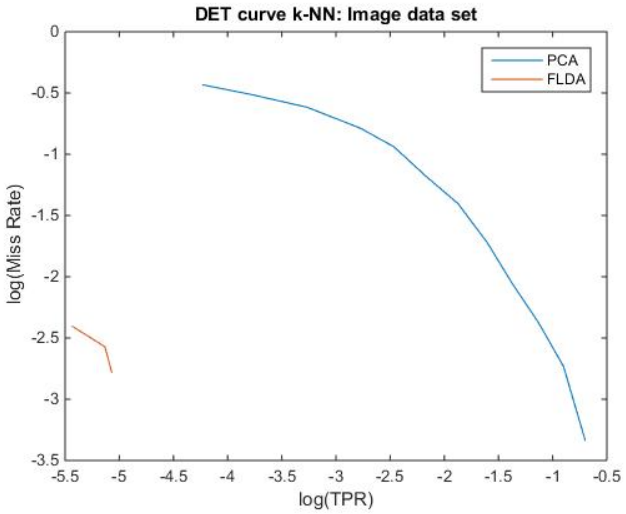
The figure 4 shows the ROC and DET curve for the k-nearest neighbour method based classifier with k=150, when PCA & FLDA was used to separate the data. An overall accuracy of 92% was obtained.

3.6 Perceptron based classifier: Image dataset

The figure 5 shows the ROC and the DET curve for the perceptron based classifier. An overall classification accuracy of 54% was achieved using the perceptron based classifier. The perceptron was trained as per one vs all classification rule. The convergence of the algorithm was based on minimising the



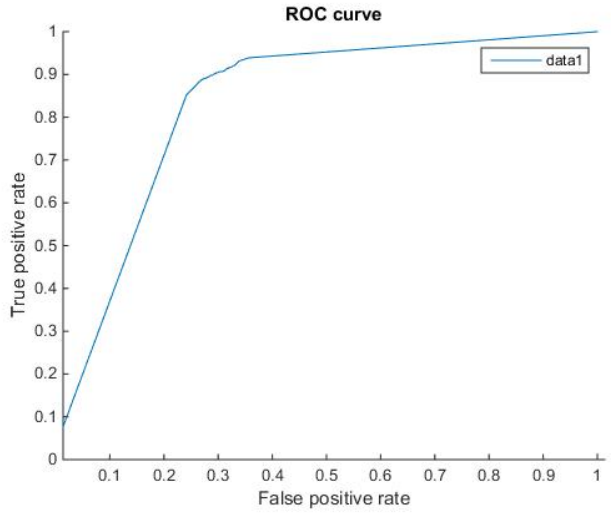
(a) ROC curve for all classes: Image data set



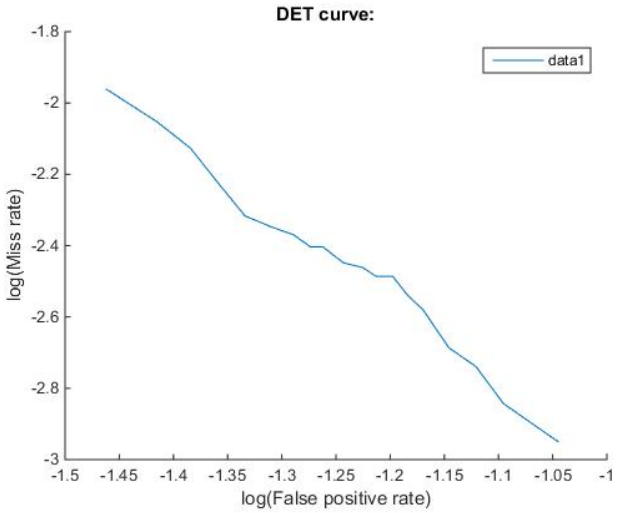
(b) DET curve for all classes: Image data set

Figure 4: ROC and DET curves for image data set for GMM

misclassification rate and the threshold for the same was set at 5%. It can be seen from the confusion matrix 3 that some classes specially tall building and inside city are misclassified. The reason for low classification accuracy can be accounted due to the non-linear separability of the overlapping classes. The perceptron algorithm is a linear classification model. Thus, for classes which are non-linearly separable, perceptron based classifier might result in lower classification accuracy 4.



(a) ROC curve for all classes: Image data set



(b) DET curve for all classes: Image data set

Figure 5: ROC and DET curves for image data set. Algorithm Perceptron

Table 3: Confusion matrix. Perceptron: image data set. All classes.

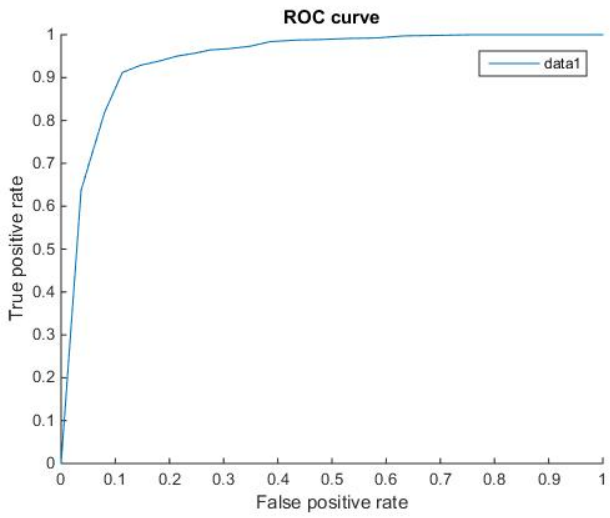
Class	**Highway-Pred**	**Coast-Pred**	**Inside City-Pred**	**Mountain-Pred**	**Open country-Pred**	**Tall Building-Pred**	**Forest-Pred**	**Street-Pred**
Highway-Trgt	31	14	1	10	11	9	0	2
Coast-Trgt	5	61	3	14	17	4	2	2
Inside city-Trgt	2	5	43	11	5	15	5	6
Mountain-Trgt	1	13	5	56	14	17	3	3
Open country-Trgt	0	11	1	16	80	8	6	1
Tall Building-Trgt	2	13	6	19	11	49	3	4
Forest-Trgt	0	0	5	6	15	2	69	1
Street-Trgt	1	3	18	12	2	3	3	46

Table 4: Table showing the performance measures for the classifier. Perceptron image data set.

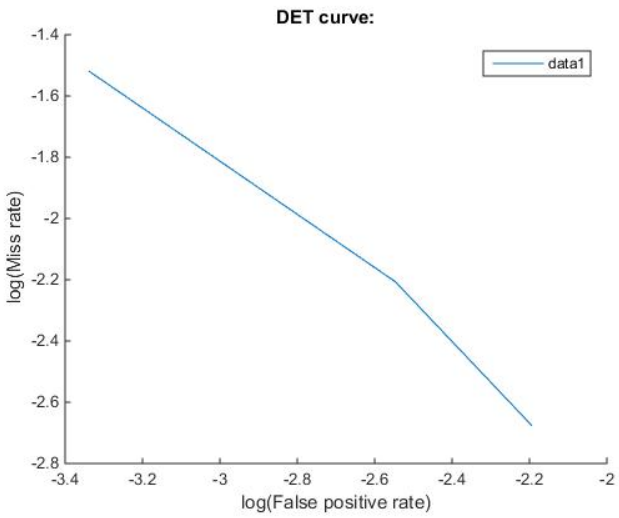
Measure	**Highway**	**Coast**	**Inside City**	**Mountain**	**Open country**	**Tall Building**	**Forest**	**Street**
Sensitivity	0.40	0.56	0.46	0.50	0.65	0.45	0.70	0.50
Specificity	0.98	0.91	0.94	0.87	0.89	0.91	0.96	0.97
Precision	0.73	0.50	0.52	0.38	0.51	0.45	0.75	0.70
F1-score	0.51	0.53	0.49	0.43	0.57	0.45	0.73	0.60
Overall Accuracy	0.54							

3.7 SVM based classifier: Image dataset

The figure 6 shows the ROC and the DET curve for the SVM based classifier. An overall classification accuracy of 75% was achieved using the SVM based classifier. The table 5 and table 6 shows the confusion matrix and the performance measures for the image data set using SVM based classifier. It can be seen from the table 5 that the classification of overlapping classes such as open country and forest is much better using SVM. The reason for the same can be attributed to SVM's ability to learn non-linear boundaries and the slack term parameter ϵ which allows for certain misclassification error for overlapping classes. Thus, the overall classification accuracy is improved using SVM. Table 7 shows the number of support vectors for the models generated using SVM's.



(a) ROC curve for all classes: Image data set



(b) DET curve for all classes: Image data set

Figure 6: ROC and DET curves for image data set for GMM

Table 5: Confusion matrix. SVM: image data set. All classes.

Class	**Coast-Pred**	**Highway-Pred**	**Inside City-Pred**	**Mountain-Pred**	**Open country-Pred**	**Tall Building-Pred**	**Forest-Pred**	**Street-Pred**
Coast-Trgt	68	4	1	1	3	0	0	1
Highway-Trgt	6	88	1	3	7	1	2	0
Inside city-Trgt	6	2	51	0	1	16	3	13
Mountain-Trgt	3	8	0	65	5	9	16	6
Open country-Trgt	1	22	1	1	67	3	24	4
Tall Building-Trgt	0	1	4	2	0	95	3	2
Forest-Trgt	0	0	0	0	0	0	98	0
Street-Trgt	3	0	2	6	1	4	4	68

4 Conclusion

As seen from the results the classifier based on mixture models and markov model is a good algorithm for performing classification tasks involving multi class data. We performed various experiments based on

Table 6: Table showing the performance measures for the classifier. SVM: Image data set.

Measure	**Coast**	**Highway**	**Inside City**	**Mountain**	**Open country**	**Tall Building**	**Forest**	**Street**
Sensitivity	0.87	0.81	0.55	0.58	0.54	0.88	1	0.77
Specificity	0.97	0.94	0.98	0.98	0.97	0.95	0.92	0.96
Precision	0.78	0.70	0.85	0.83	0.80	0.74	0.65	0.72
F1-score	0.82	0.75	0.67	0.68	0.64	0.80	0.79	0.74
Overall Accuracy	0.75							

Table 7: Table showing the number of support vectors.

Data set	**Support Vectors**	**Percentage**
Highway	175	40%
Coast	229	46%
Inside city	204	44%
Mountain	268	52%
Open country	250	46%
Tall Building	219	44%
Forest	110	23%
Street	172	38%

which we observed that mixture models based on full covariance matrix perform better classification. Similar results were obtained for markov models. Using the mixture model classifier built for full covariance, we achieved a classification accuracy of greater than 70% in classification of image data set and greater than 90% for spiral and digit data set. However, the performance for hand written character data set was low owing to the inherent class overlap in the data. An overall accuracy 20% with GMM and 33% with HMM is achieved. On the video data set based on the optical flow features extracted from the videos, we could achieve an overall classification accuracy of 80% with GMM and 67% with HMM.