REPORT
SMAI Project

- Mohit Aggarwal 201101164
- Yashasvi Girdhar 201101146

PROBLEM
In this project, we implement the face recognition algorithm. Recognition is carried out in modes: Identification and Verification. We are given a dataset, we divide it into train set and test set and then compute the ROC curves and accuracies using 4 cross validation.

DATASETS

Yale Dataset
This dataset contains face images that are captured under controlled conditions (illumination, pose and expression) and cropped to contain the face only. A perl script is written to select only those images which have their Azimuth and Elevation angles are between -25 to +25. The images are of dimension 192 X 168 which are resized to 80 X 80. Finally 760 images (20 images belonging to each class) are selected for training and testing.

CMU-PIE Dataset
This dataset contains face image captured under various poses, illuminations and expressions (hence PIE). There are total 2856 images of dimension 32 X 32 pixels in this dataset.

SMAI Students Dataset
This dataset contains face images detected from the photographs taken in a selected set of classes. These faces may be of different sizes, pose and illumination. The images are of dimension 80 X 80.

ALGORITHM
The algorithm works by computing the Principal components (PCA) of the covariance matrix of the training set. Top k eigenvectors of the covariance matrix are chosen. Each training image is then reduced to k dimensions after projecting it on the k eigen faces. We then store the weight vector(train image in reduced dimension) of each train image and use K-nn to classify the test images. We also compute the ROC curves.

When we get a test image, we subtract the mean image (of the training set) from the test image, reduce it to k dimensions by projecting it on k eigen faces and then use k-nn to classify this image.

Reconstruction of Image

We reduce the test image to k dimensions to get the k dimension weight vector of the image. We then multiply each value in the weight vector with the corresponding eigen face and then take the linear combination of k vectors to get the reconstructed test image. Then use imshow function in Matlab to display the image.

RESULTS

Experiment 1 : Yale Dataset

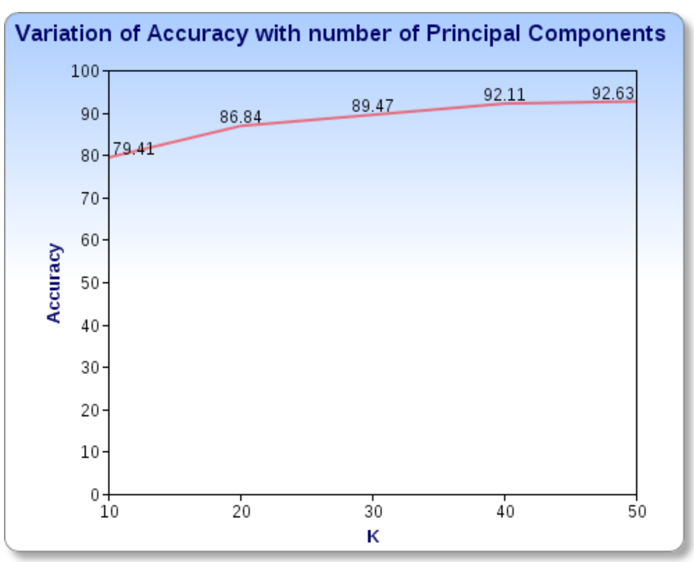
4 Cross Validation (Number of Eigen faces = 50)

Train Set	Test Set	Correct	Total	Accuracy
1,2,3	4	177	190	93.16
1,2,4	3	182	190	95.79
1,3,4	2	189	190	99.47
2,3,4	1	183	190	96.32

Average accuracy = 96.185 %

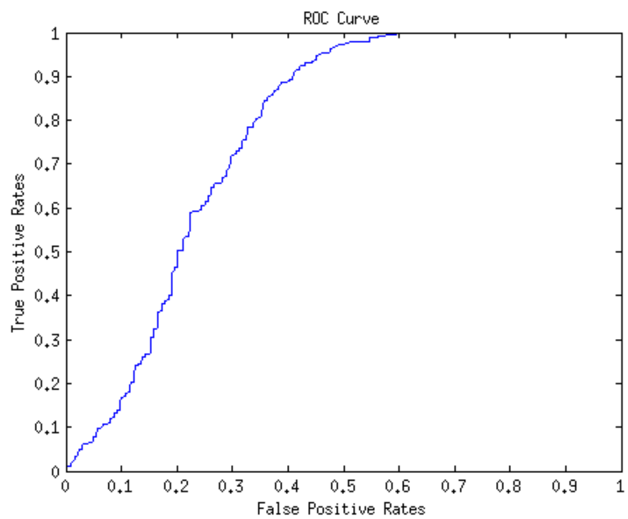
Variation of Accuracy with number of Principal Components (Eigen Faces)

Here I have taken the 1st , 2nd and 3rd sets as the training sets and the 4th set as the test set.



As we can see in the graph, that as 'K' increases, the accuracy increases. This is because K is the number of eigen faces on which the image is projected so as to get the weight vector in reduced dimensions. If we choose more number of eigen faces we'll able to preserve more information (features) about the image.

ROC Curve



Experiment 2: CMU-PIE Dataset

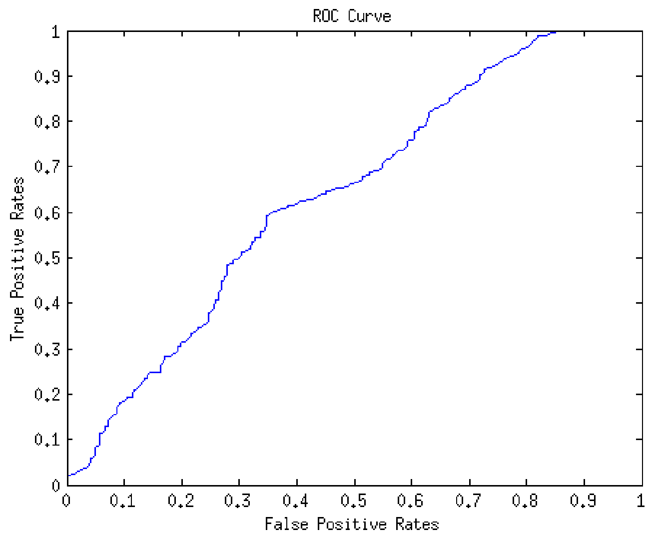
4 Cross Validation (Number of Eigen faces = 50)

Train Sets	Test Set	Correct	Total	Accuracy
1,2,3	4	714	714	100
1,2,4	3	714	714	100
1,3,4	2	714	714	100
2,3,4	1	714	714	100

Average accuracy = 100 %

Principal Component Analysis (PCA) Method is efficient because it preserves maximum information about the images in lesser number of dimensions. And then we apply K-nn which classifies the test images in a very efficient manner, however it makes the program slow. However this method is more efficient than the previous assignments, because we are applying k-nn on images with reduced dimensions that have captured maximum information about the images.

ROC Curve



Experiment 3 : SVM

For the svm classifier, **libsvm** is used. The classifier has been trained on the Yale Dataset. By default, svm uses binary classification. For the multi-class classification, we have used a python script (trans_class.py, available online) that transforms data to multi-class sets.

So, basically we have used three commands:

> *./trans_class.py training_file testing_file*

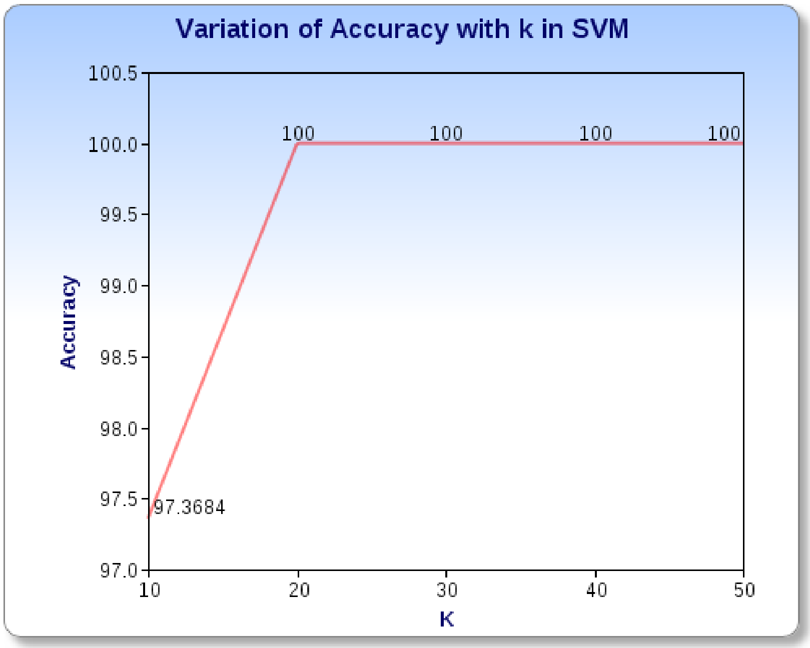
- It generates three temporary files: "tmp_train" and "tmp_test" are multi-class sets, and "tmp_class" contains the mapping information.

> *svm-train tmp_train*

- It creates the model file tmp_train.model.

-> *svm-predict tmp_test tmp_train.model o*

- It gives the accuracy and stores the labels assigned to the test images, in the file ‘o’.



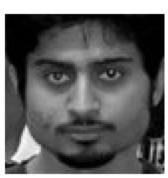
The svm classifier gives better accuracy as compared to k-nn classifier.

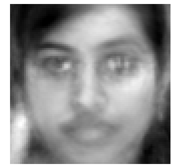
Experiment 4

Here we consider a variety of images (both known, unknown faces and non-faces) of the same size as that of the Eigen faces computed in the previous experiment. We then compute the representation of these images using the Eigen Face basis and reconstruct the image from the representation.

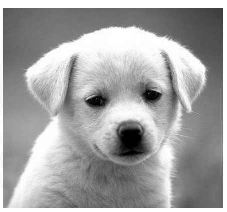
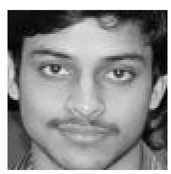
Train Set - SMAI Students Dataset

Reconstruction of known face images (from Class)

Original Image	**Reconstructed Image**
	

Reconstruction of unknown face images (which do not belong to any class in Training Set)

Original Image	Closest Image (Min Euclidean distance)	Reconstructed Image
		
		
		

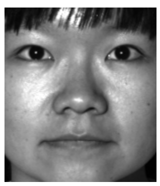
The reconstructed images of the known-faces are much more face like, as compared to the reconstructed images of the Unknown-faces. This is because when we choose a known face, we already have images in the train set belonging to that class, due to which it is able to reconstruct an image which resembles more of a human face.

The reconstruction of the face images is not faithful when the original image is not face-like.

So, what we infer from the results is that if we have a test image from the known classes, the results are very much accurate as compared to the case when the test image doesn't belong to the known classes.

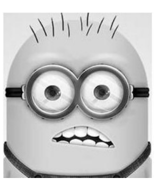
Train Set - Yale Dataset

Reconstruction of known face images (from Class)

Original Image	Reconstructed Image
	
	

Here the reconstruction of the images is very accurate because the eigen faces have been trained on a train set which contains face images with proper pose, alignment and illumination. As a result, the eigen faces preserve maximum information about the face features and reconstruct a test image which resembles very much like a face.

Reconstruction of non-faces

Original Image	Closest Image (Min Euclidean distance)	Reconstructed Image
		
		

The reconstructed images of non-faces are completely blurred. The reason for this is because face images form a subspace of image space and the eigen faces are determined in this subspace. Since the testing images are outside this subspace, their reconstruction is a complete failure.

Experiment 5 : SMAI Student Dataset

4 Cross Validation (Number of Eigen faces = 50)

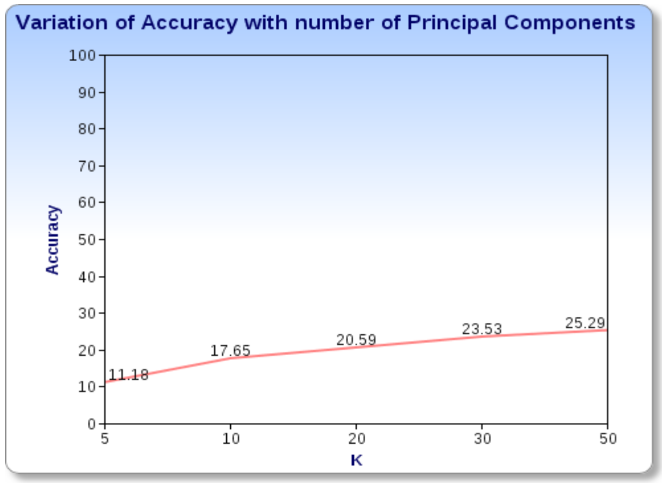
Train Sets	**Test Set**	**Correct**	**Total**	**Accuracy**
1,2,3	4	38	170	22.35
1,2,4	3	32	170	18.82
1,3,4	2	33	170	19.41
2,3,4	1	43	170	25.29

Average accuracy = 21.46 %

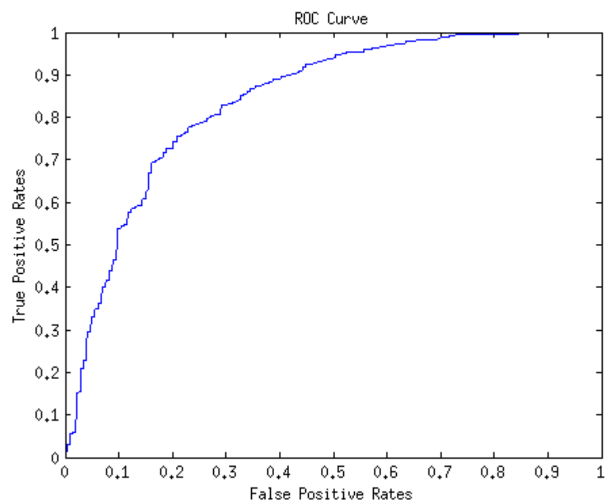
Hold One Out Method
In this method, we take one image as the test image and rest all images as the train images. This method is applied for each image in the dataset and then the average accuracy is computed.
Average accuracy = 24.5%

Here the accuracy is very less because the images used in the dataset are random images which are captured under different poses, illuminations and expressions. The face images are not properly aligned and contain other things apart from the face, due to which the features are not properly extracted and the classifier is not able to classify test images with good accuracy. Also there is less number of available training set for each label.

Variation of Accuracy with number of Principal Components (Eigen Faces)
Here I have taken the 2nd, 3rd and 4th sets as the training sets and the 1st set as the test set.



ROC Curve



Interesting Results

- Computing the principal components using first 50 eigen vectors (with largest eigen values) gives an accuracy of approx. 96%. However if I remove the first 3 eigen vectors, and compute the principal components, then it gives an accuracy of 100%. This is probably because the first 3 eigen vectors are noisy.
- Increasing the number of eigen faces after 50, does not significantly change the accuracy. Most of the information of the training dataset is preserved in 50 eigen faces. So its sufficient to reduce the images to 50 dimensions.
- If I do not normalise the eigen faces, then accuracy drops from 96% to 10%.
