Face Detection and Recognition

GROUP – 11

Project Report by:

Akash Waghela (13064)

Rajat Jain (10510569)

Harshita Shrivastava (13298)

Proposal: Face Detection and Recognition from User created Database of faces.

The Report describes various ways to Detect Facial features and recognize them versus a given Database. Inference are drawn on the basis of obtained results and are tried to explain. The following is used to achieve the same.

Used **Open CV** for Extracting Facial Features and Recognizing a given face.

Implementation Language Python

Face Detection:

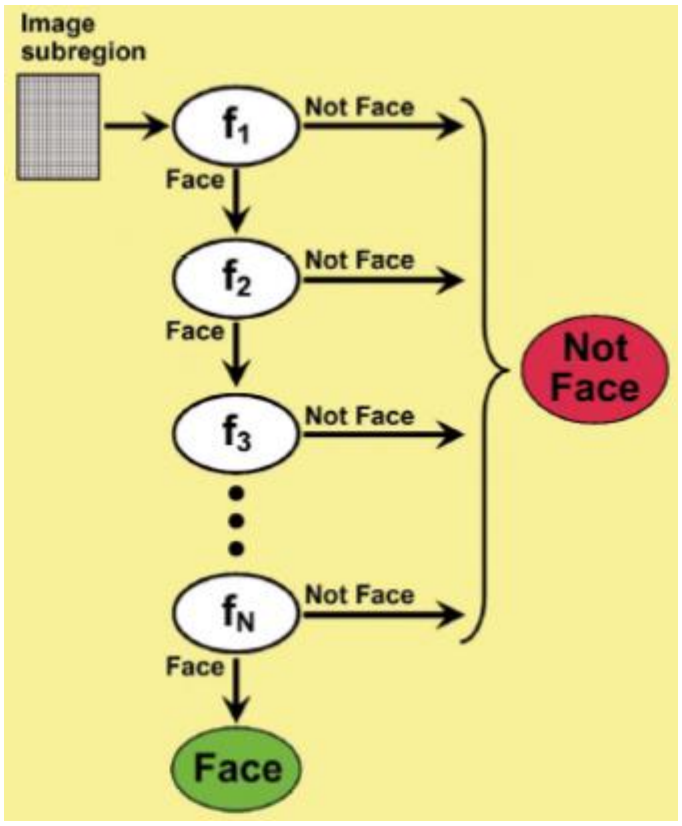
Initially we labelled 200 images for faces and marked their facial features like eyes, ears, nose and hairline.

For Training[1]:

200 Positive and Negative Images with associated label.

Output:

A Harrcascade file that has features associated to a face. As shown below.



Cascade of Classifiers i.e. Instead of applying all the features at a time it applies a subset of features one at a time if the window fails at any stage it is discarded else we continue to process till it passes all feature. The frame that passes all the features is a face.

[1] Source https://www.cs.auckland.ac.nz/~m.rezaei/Tutorials/Haar-Training.zip contains 200 facial images and 200 non-facial images.

Results

	Total Number of Faces	Number of Detected Face	Comment
	40	26	Without Hairline
	140	89	Side+Frontal Face
	400	384	All Frontal Face
Average	580	499	0.860344828

Face Recognition:

A Face is detected using the Harrcascade file made in previous step and then recognized by one of the methods described below against a database.

For Testing

cv2.createLBPHFaceRecognizer()

cv2.createEigenFaceRecognizer()

cv2.createFisherFaceRecognizer()

Methods used and tested

LBPH (Local Binary Patterns Histograms):

The idea is to not look at the whole image as a high-dimensional vector, but describe only local features of an object. The features you extract this way will have a low-dimensionality implicitly.

The Local Binary Patterns methodology has its roots in 2D texture analysis. The basic idea of Local Binary Patterns is to summarize the local structure in an image by comparing each pixel with its neighborhood. Take a pixel as center and threshold its neighbors against. If the intensity of the center pixel is greater-equal its neighbor, then denote it with 1 and 0 if not. You'll end up with a binary number for each pixel, just like 11001111. So with 8 surrounding pixels you'll end up with 2^8 possible combinations, called **Local Binary Patterns**.

Eigen Faces:

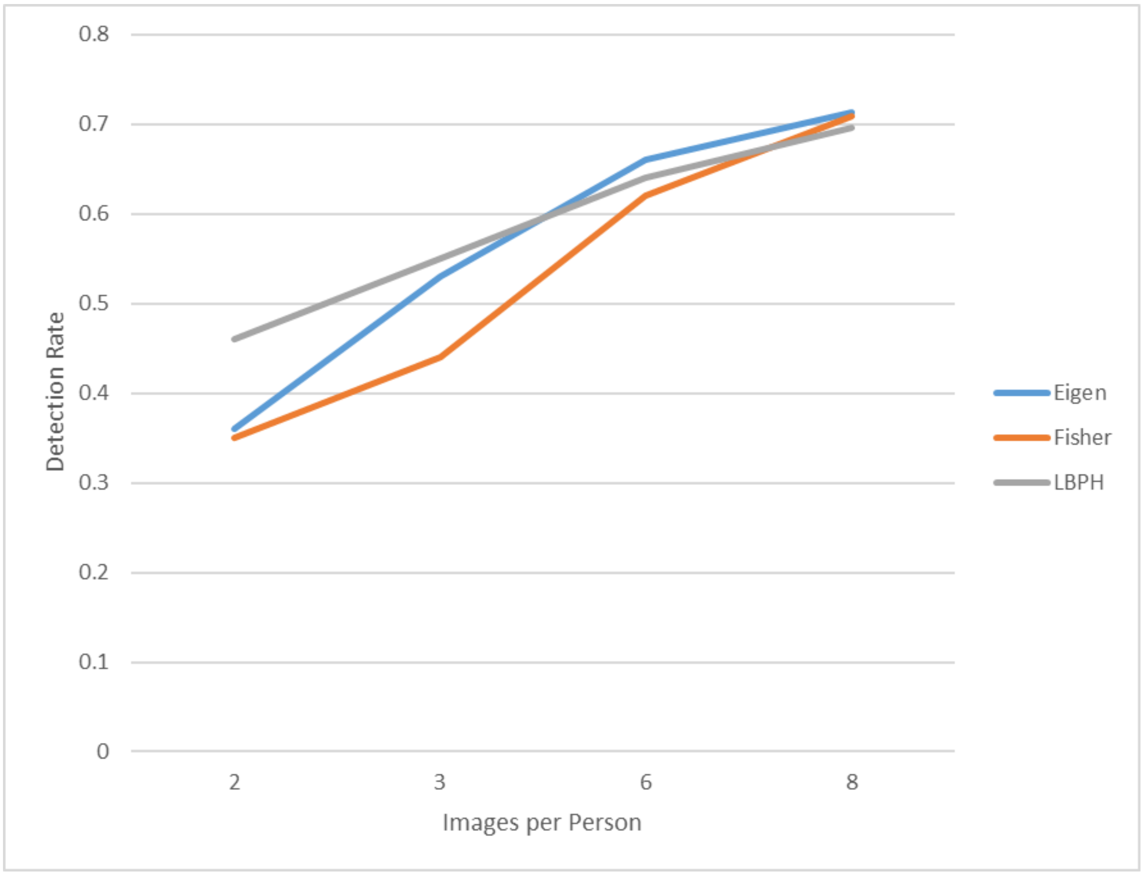
Eigen Faces is a very popular method to detect faces. It uses the fact that that every dimension does not provide equal information about a feature. Hence, it uses PCA for dimensionality reduction and constructs an average face of a given label form a pool of images of that label. While doing a query for a face it converts it into an Eigen face and finds a weight vector that closely generates the given Eigen face. Then the label of the closest weight vector is given to the query.

Fisher Faces:

It uses LDA (Linear Discriminant Analysis), when LDA is used to find the subspace representation of a set of face images, the resulting basis vectors defining that space are known as Fisherfaces.

Results:

Method Used	Number Of frames	Number of Correct Labels	Accuracy	Training Size
LBPH	86	40	0.465116279	2 images per person
EigenFaces	82	30	0.365853659	
FisherFaces	80	28	0.35	
LBPH	100	55	0.55	3 images per person
EigenFaces	100	53	0.53	
FisherFaces	80	35	0.4375	
LBPH	100	64	0.64	6 images per person
EigenFaces	100	66	0.66	
FisherFaces	100	62	0.62	
LBPH	79	55	0.696202532	8 images per person
EigenFaces	77	55	0.714285714	
FisherFaces	79	56	0.708860759	



Comments:

Eigen and Fisher require generally larger number of faces (10-15) for making a better average face and thus perform lower than LBPH which saturates much faster with increasing number of images per person.

Ensemble Voting Algorithm can be used on this three method to predict the label of a query. If the label of both Fisher and Eigen match then give it that label else go to LBPH for deciding which label to give, as it has higher accuracy and dominates in the given region of interest i.e. the database used by us.

Ideally, the features of face should not change for being a good biometric. But since the face changes slightly with age, the facial growth can be modelled over time by simulation to avoid the problem due to ageing

Side Face Detection, can also be done easily by marking special features like eyes, nose and mouth instead of complete face as these are always present in side face.