Project 2

Jatin Chinchkar jchinch@ncsu.edu

Data set

CMU/MIT Frontal Faces(16)

Data processing

1. The dataset has two separate files for training and testing purpose as well it has face and non-face images.
2. I have stored the dataset into the project directory and accessed and stored the respective data using OS library from Python into four lists.
3. I have worked on the images of size 10x10 and in greyscale format.

1. Haar like features

Haar like features are mainly used for classificatino of face data. The haar like features are difference of the sum of pixels with in the rectangle. The haar like features indicate the presence of characteristic features in the image.

2. Approach for algorithm

The algorithm was implemented into following steps

1 - Calulcating integral image of a given image.

2 - Calculating all possible haar like features of the images.

3 - Finding the threshold value of all features which would classify the face, non face data having accuracy more than 0.50.

4 - Implement the Adaboost Algorithm to fetch the top features along with their weights, feature value.

5 - Use the top haar features and extract those features from test data and classify ac-

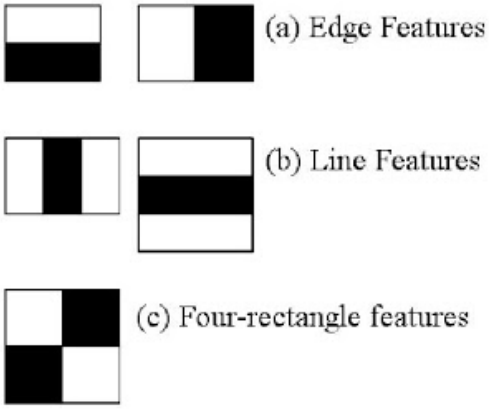
cording to the the result from Adaboost.

3. Calculating integral image

The integral images are calulcated by summing the above and left pixel values for each pixel (x,y). This method helps in faster calculation of haar like features.

4. Calculating Haar like features

The haar like features are difference of sum of them pixel with in the rectangle of the haar feature. I have used 5 haar feature types. The haar features are calculated using integral image just by (A+D-C-B) where A,D are start and end point area values of the image and C,D are remaing corner area values.(Image reference from http://www.willberger.org/cascade-haar-explained/)



5. Finding threshold

For calculating threshold of each haar like feature the value of the face and non face data for the same features are extracted. and using linesearch an optimal feature value is selected which would result in the best classification of data using that feature value.

6. Implementing Adaboost

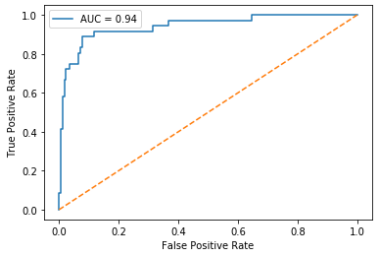
The Adaboost is implemented according to the steps provided in the slides. First weights for each image is assigned by value 1/(number of training example). Then the error is calculate by adding up the weights of data which are incorrectly classified for every feature value. Then minimum value error is chosen as the best classifier for that iteration and alpha value is calculate using the error for the same feature. Weights for the data is updated using labels, currently classification labels, alpha,and current weights. The weights are normalised to keep the summation of weights equal to 1. All the top ten features along with the feature value, alpha, classification of the data for the respective classifiers are store in order to implement testing of the mode.

7. Testing Model

Once we have the top 10 features we will calculate the feature values for the features in test face and test non face data. Then we will compare the feature values of test data with the feature values of weak classifier. Then the final classification is done by combining the weak classifiers together.

8. Results

The training accuracy for the given data(1000 face, 1000 non face) set was obtained upto 0.9015 where as the test accuracy on 800(400 face, 400 non face) images was obtain upto 0.8975. Below is the ROC curve for the model.



9. Top ten haar features visalization