Face Detection Algorithms

Face detection is a computer technology that determines the locations and sizes of human faces in arbitrary (digital) images. It detects facial features and ignores anything else, such as buildings, trees and bodies. Face detection can be regarded as a more general case of face localization. In face localization, the task is to find the locations and sizes of a known number of faces (usually one).

A face detector has to tell whether an image of arbitrary size contains a human face and if so, where it is. One natural framework for considering this problem is that of binary classification, in which a classifier is constructed to minimize the misclassification risk. Since no objective distribution can describe the actual prior probability for a given image to have a face, the algorithm must minimize both the false negative and false positive rates in order to achieve an acceptable performance.

This task requires an accurate numerical description of what sets human faces apart from other objects. It turns out that these characteristics can be extracted with a remarkable committee learning algorithm called Adaboost, which relies on a committee of weak classifiers to form a strong one through a voting mechanism. A classifier is weak if, in general, it cannot meet a predefined classification target in error terms.

An operational algorithm must also work with a reasonable computational budget.

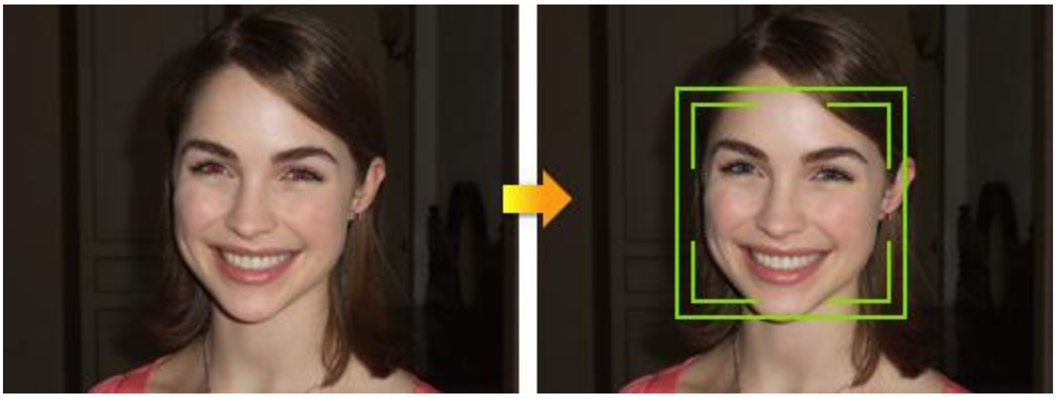


Fig 1: A basic Face Detection problem

Different Types of Face Recognition Algorithms

Principal Component Analysis (PCA)

Principal Component Analysis commonly uses the Eigen faces in which the probe and gallery images must be the same size as well as normalized to line up the eyes and mouth of the subjects whining the images. Approach is then used to reduce the dimension of data by the means of image compression basics and provides most effective low dimensional structure of facial pattern. This reduction drops the useless information and decomposes the face structure into orthogonal (uncorrelated) components known as Eigen faces. Each face image is represented as weighted sum feature vector of Eigen faces which are stored in 1-D array. A probe image is compared against the gallery image by measuring the distance between their respective feature vectors then matching result has been disclosed. The main advantage of this technique is that it can reduce the data needed to identify the individual to 1/1000th of the data presented.

Linear Discriminant Analysis

It is difficult to analysed human facial components like features. All of component has a different discrimination for identifying a person. Linear discriminant group images of the same class and separates images of different classes of the images. Follows these steps:

- First, we acquire training set composed of a relatively large set of images with diverse facial characteristics.
- Second, appropriate selection of training set support to find out final result. So that we considered that images have already normalized to m 3 n arrays only face region.
- Each image and sub image, starting with the two dimensional m 3 n array of intensity values I(x, y), then we construct the lexicographic vector expansion f P Rm3n.
- Third, by defining one class for all instances of the same person's face and the faces of different subjects as being in different classes for all subjects in the training set, we establish a framework for performing a cluster separation analysis in the feature space. Having labelled all instances in the training set and having defined all the classes, we compute the within and between-class scatter matrices as follows:

$$S_w^{(V)} = \sum_{i=1}^{L} \Pr(C_i)\Sigma_i\,,$$

$$S_b^{(V)} = \sum_{i=1}^{L} \Pr(C_i)(\mu - \mu_i)(\mu - \mu_i)^T.$$

Support Vector Machine

The Support Vector Machine is based on VC theory of statistical learning. It is implement structural risk minimization. Basically, it was introduced as per a binary classifier. It estimates the support vectors through concluding a hyperplane. Support Vectors increase the distance or margin between the

hyperplane and the closest points.
Suppose a set of N points and Xi ∈ Rn, i=1, 2, 3,.. , N. Each point belongs to one of the two classes i.e. Yi ∈ {−1,1}.

$$f(x) = \sum_{i=1}^{l} \alpha_i Y_i \boldsymbol{X}_i . \boldsymbol{X} + b$$

This quadratic equation f(x) decides the Classification of a new point data in the above equation.

Independent Compontent Analysis

Generalization View of the PCA is known as ICA. Use for increases the second order and higher order dependencies in the input and determines a set of statistically independent variables or basis vectors.
Here we are using architecture which finds statistically independent basis images.

Follow the basic steps for ICA:

- Collect Xi of n dimensional data set X, i = 1, 2, 3, ….., M.
- Mean correct all the points: calculate mean Mx and substract it from each data point, Xi − Mx
- Calculate the covariance matrix: C= (Xi – Mx) (Xi –Mx) T
- The ICA of X factorizes the covariance matrix into the following form: C = FΔFt where Δ is a diagonal real positive matrix. F converts into the original data X into Z.

Singular Valued Decomposition

Basic steps for SVD as below:
- Obtain a training set S with N face images of known individuals.
- Compute the mean face f of S

- Forms a matrix A with the computed f.
- Calculate the SVD of A
- For each known individual, compute the coordinate vector Xi.
- For a new input image f to be identified, calculate its coordinate vector x,
- Apply the Max-MIN Normalization to Bound the value obtained.

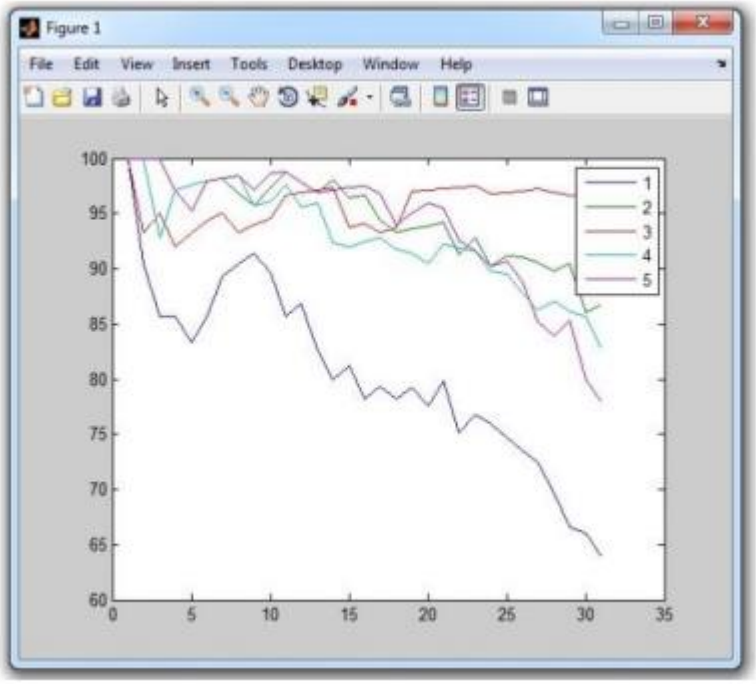
Comparison of Algorithms by running on AT&T and IFD database

(Source: International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 5, May 2015)

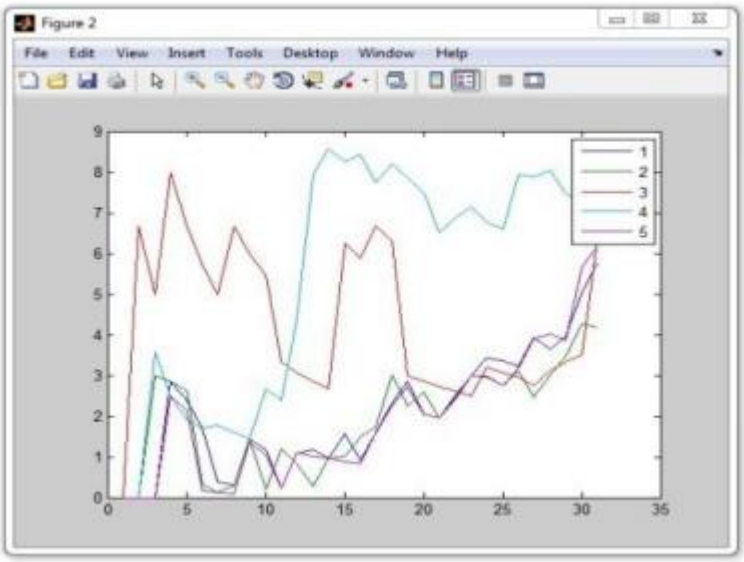
For comparison selection of the algorithms was made one by one and then the recognition rate, failure rate, training time and recognition time information was generated for all the algorithms separately.
This information was used to plot the graphs comparing the above mentioned rate and time performances of the five algorithms under study.

Graphs

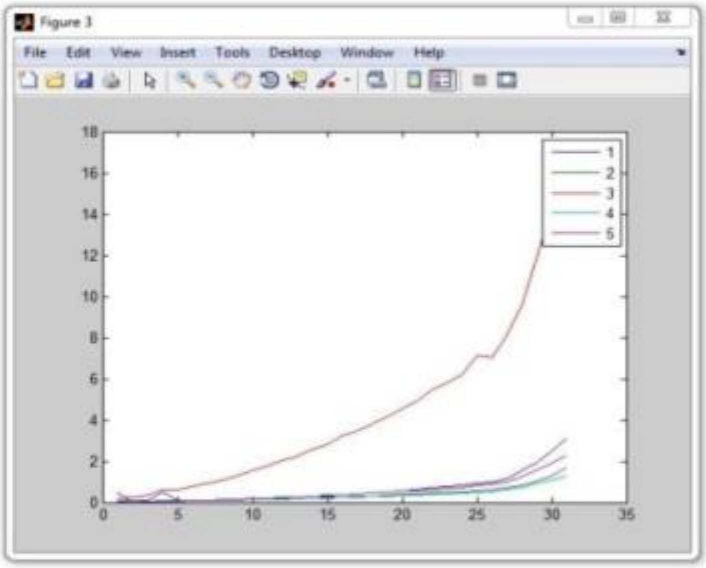
(1.PCA 2.LDA 3.SVD 4.SVM 5. ICA)



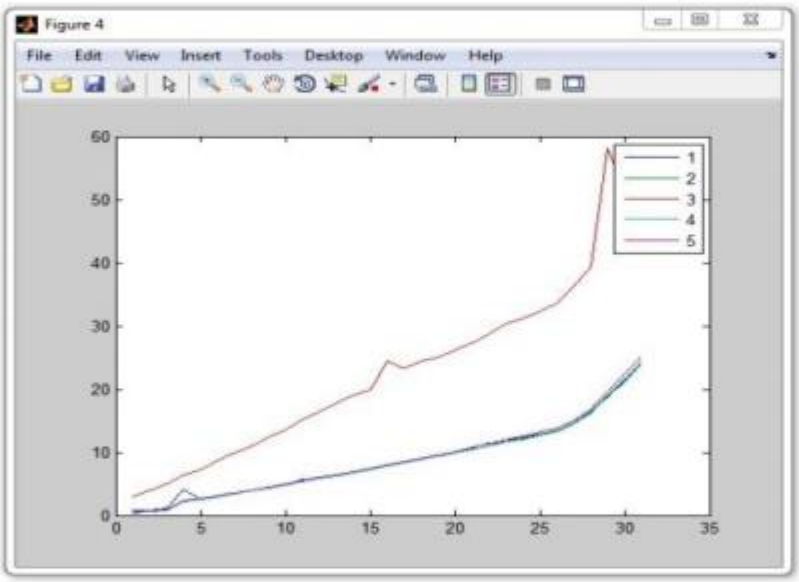
The recognition rates of the five algorithms when applied on both databases



Comparison of failure rate when five algorithm applied on both databases



The comparison of training rate when the five algorithms were applied on both the databases



The comparison of recognition time of the five algorithm when applied on both the databases

Results

Following results were achieved after experimental analysis of the performance of above five algorithms.

- According to comparison of recognition rate employed on five algorithms. The highest accuracy rate is 97.5% (SVD) and lowest is PCA accuracy rate
  is 93.7% but performance of LDA & SVM was same at 95.3%. ICA performed with an average recognition rate of 80%.

- The second graph shows failure rate of algorithms when number of image were added to database. We analysed that PCA performed very well on less
  number of images but with increased number of images failure rate grew. ICA performed opposite to PCA. SVD and LDA varied in terms of performance
  but resultant is fixed. Most of the fluctuation appeared in SVM when images were added on database, failure rate increases but after that decreased slowly.
  Comparing all of them SVD and LDA's performance was very good in terms of failure rate.

- Comparison of algorithms on the basis of training time, we observed that SVD performance was best and that of SVM slowest. PCA, LDA and ICA training time was average in comparison with SVD and SVM.

- For the Last one i.e. recognition time it was observed that time consumed by SVD was more than four times than that of other algorithms.

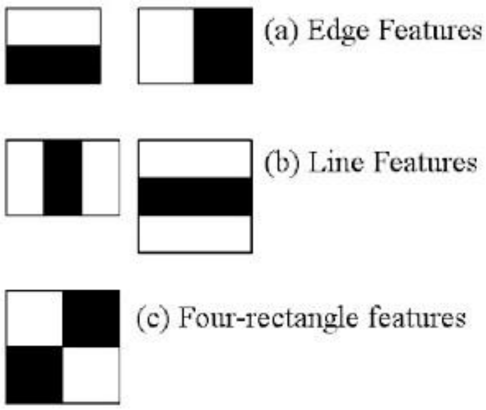
Face Detection using Haar-Cascade Classifier

(OpenCV algorithm used for Task Completion)

Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in

2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.

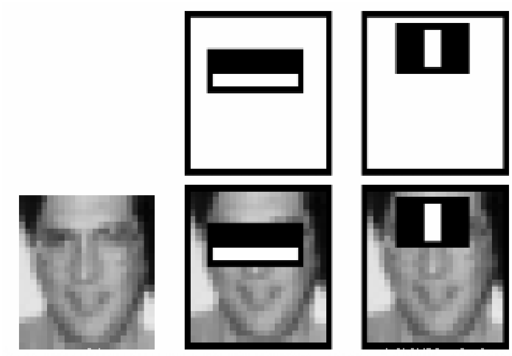
Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. For this, haar features shown in below image are used. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting sum of pixels under white rectangle from sum of pixels under black rectangle.



Image

Now all possible sizes and locations of each kernel is used to calculate plenty of features. For each feature calculation, we need to find sum of pixels under white and black rectangles. To solve this, they introduced the integral images. It simplifies calculation of sum of pixels, how large may be the number of pixels, to an operation involving just four pixels.

But among all these features we calculated, most of them are irrelevant. For example, consider the image below. Top row shows two good features. The first feature selected seems to focus on the property that the region of the eyes is often darker than the region of the nose and cheeks. The second feature selected relies on the property that the eyes are darker than the bridge of the nose. But the same windows applying on cheeks or any other place is irrelevant. The best features out of 160000+ features are selected by using Adaboost.



Image

For this, we apply each and every feature on all the training images. For each feature, it finds the best threshold which will classify the faces to positive and negative. But obviously, there will be errors or misclassifications. We select the features with minimum error rate, which means they are the features that best classifies the face and non-face images.

Final classifier is a weighted sum of these weak classifiers. It is called weak because it alone can't classify the image, but together with others forms a strong classifier. The paper says even 200 features provide detection with 95% accuracy. Their final setup had around 6000 features.

In an image, most of the image region is non-face region. So it is a better idea to have a simple method to check if a window is not a face region. If it is not, discard it in a single shot. Don't process it again. Instead focus on region where there can be a face. This way, we can find more time to check a possible face region.

For this they introduced the concept of Cascade of Classifiers. Instead of applying all the 6000 features on a window, group the features into different stages of classifiers and apply one-by-one. If a window fails the first stage, discard it. We don't consider remaining features on it. If it passes, apply the second stage of features and continue the process. The window which passes all stages is a face region.

Authors' detector had 6000+ features with 38 stages with 1, 10, 25, 25 and 50 features in first five stages. (Two features in the above image is actually obtained as the best two features from Adaboost). According to authors, on an average, 10 features out of 6000+ are evaluated per sub-window.

OpenCV already contains many pre-trained classifiers for face, eyes, smile etc. Those XML files are stored in opencv/data/haarcascades/ folder. Let's create face and eye detector with OpenCV. **(Note: The given task in questionnaire is performed using the Frontal Face Haar Cascade xml file)**

References

- OpenCV tutorials
- http://www.ipol.im/pub/art/2014/104/
- International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 5, May 2015
- Face Detection Home Page (Website)