ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 1, January 2018

Face Detection and Recognition from Live Video Stream

[1] Dr. S Govinda Rao, [2] G Anil Kumar, [3]Y Manoj Kumar
[1] Professor in CSE, [2][3]Assistant Professor in CSE
[1][2][3] GokarajuRangaraju Institute of Engineering and Technology

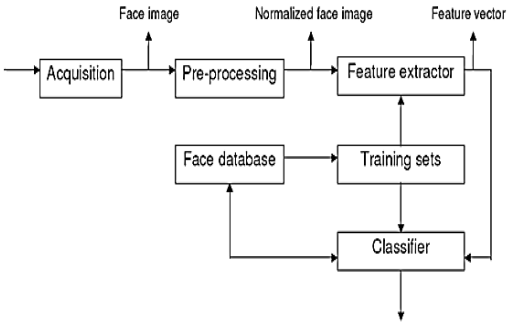
***Abstract* - Face recognition" is a very active area in the computer vision and Biometric fields as it has been studied vigorously for 25 years and is finally producing applications in security ,robotics ,human-computer interfaces ,digital cameras and entertainment. "Face recognition" generally involves two stages. Face detection, where a photo is searched to find related face, then image processing cleans up the facial image for easier recognition. Since 2002, Face Detection can be performed fairly reliably such as openCV face detector, working in roughly 90-95% of clear photos of a person looking forward at the camera .The openCV libraries makes it fairly easy to detect a frontal face new in an image or from a video feed. Face Recognition is the process, where that detected and processed face is compared to a database of known faces to decide who that person is. Under face recognition, we can then compare the detected image to a database of real identity of that person**

1. INTRODUCTION

The human face possesses superior expressive ability and provides one of the most powerful and natural means of communicating motivational and affective state. We use facial expressions not only to express our emotions, but also to provide important communicative cues during social interaction, such as our level of interest, our desire to take a speaking turn and continuous feedback signaling understanding of the information conveyed. Facial expression constitutes 55 percent of the effect of a communicated message and is hence a major modality in human communication [1]. This project is supposed to detect the faces visible in the camera source and on training the images, the faces are stored in a folder with the name given to it by the user. Whenever the faces that are stored in the source folder are displayed on the camera source, on clicking the identify button, the images are compared and the matched name is displayed on the screen

Face Recognition Process

Regardless of the algorithm used, facial recognition is accomplished in a five step process.



1. Image acquisition

Image acquisition can be accomplished by digitally scanning an existing photograph or by using an electro-optical camera to acquire a live picture of a subject. Video can also be used as a source of facial images. The most existing facial recognition systems consist of a single camera. The recognition rate is relatively low when face images are of various pose and expression and different illumination. With increasing of the pose angle, the recognition rate decreases. The recognition rate decreases greatly when the pose angle is larger than 30 degrees. Different illumination is not a problem for some algorithms like LDA that can still recognize faces with different illumination, but this is not true for PCA. To overcome this problem, we can generate the face images with frontal view (or little rotation), moderate facial expression, and same illumination if PCA algorithm is being used [2].

2. Image Preprocessing:

Face recognition algorithms have to deal with significant amounts of illumination variations between gallery and probe images. For this reason, image preprocessing algorithm that compensates for illumination variations in images is used prior to recognition. The images used are gray scaled.

Histogram equalization is used here to enhance important features by modifying the contrast of the image, reducing the noise and thus improving the quality of an image and improving face recognition. It is usually done on too dark or too bright images.

The idea behind image enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interest in an image. Images are

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 1, January 2018

enhanced to improve the recognition performance of the system.

3. Face Detection:

Face detection is a computer technology that determines the locations and sizes of human faces in arbitrary images. It detects facial features and ignores anything else, such as buildings, trees and bodies. Face detection can be regarded as a specific case of object-class recognition, a major task in computer vision. Software is employed to detect the location of any faces in the acquired image. Generalized patterns of what a face "looks like" are employed to pick out the faces.

The method devised by Viola and Jones, that is used here, uses Haar-like features. Even for a small image, the number of Haar-like features is very large, for a 24×24 pixel window one can generate more than 180000 features.

AdaBoost is used to train a classifier, which allows for a feature selection. The final classifier only uses a few hundred Haar-like features. Yet, it achieves a very good hit rate with a relatively low false detection rate.

4. Feature Extraction

This module is responsible for composing a feature vector that is well enough to represent the face image. Its goal is to extract the relevant data from the captured sample. Feature extraction is divided into two categories, the holistic feature category and the local features category. Local feature based approaches try to automatically locate specific facial features such as eyes, nose and mouth based on known distances between them. The holistic feature category deals with the input face image as a whole.

Different methods are used to extract the identifying features of a face. The most popular method is called Principle Components Analysis (PCA), which is commonly referred to as the eigen face method. Another method used here is called Linear Discriminant Analysis (LDA), which is referred to as the fisher face method. Both LDA and PCA algorithms belong to the holistic feature category [3].Template generation is the result of the feature extraction process. A template is a reduced set of data that represents the unique features of an enrollee's face consisting of weights for each image in the database. The projected space can be seen as a feature space where each component is seen as a feature [4].

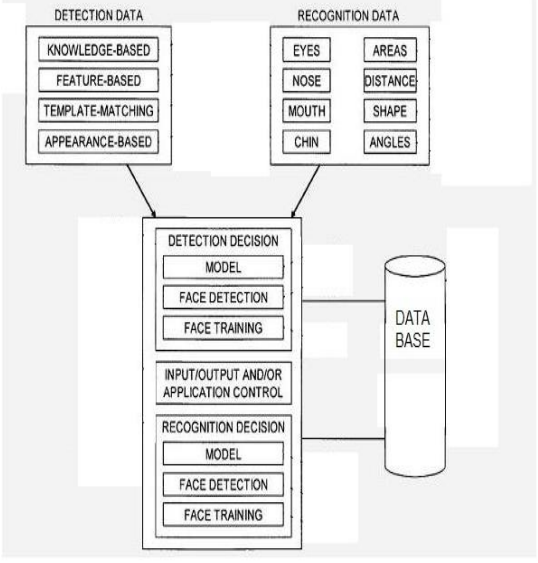
5. Declaring a match

The Last step is to compare the template generated in step four with those in a database of known faces. In an identification application, the biometric device reads a sample and compares that sample against every record or template in the database, this process returns a match or a candidate list of potential matches that are close to the generated templates in the database. In a verification application, the generated template is only compared with one template in the database that of the claimed identity, which is faster.

Closest match is found by using the Euclidean distance which finds the minimum difference between the weights of the input image and the set of weights of all images in the database [5].

II. METHODOLOGY

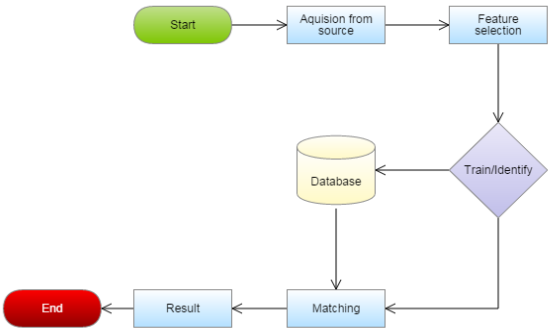
Real Time Face detection and recognition implementing Eigen faces algorithm using openCV from a live video feed. Eigen faces is a computer vision problem that uses a set of training images to develop data matrix that can be used to project, detect or compare against input images. OpenCV is widely used computer vision and machine learning library compatible with multiple programming languages.The project intends to use Java with OpenCV to realize the solution by diminishing the effect of external environment on the user input image. Below block diagram gives us a brief outline of the flow of the steps used in the project implementation. The detection data contains estimated angles and other statistic values that are required to identify any human face. In the recognition data, we have all the stored faces which are used for comparison of the faces whenever required [6].



ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 1, January 2018

In certain embodiments, the input/output control application determines an input interface feature and/or characteristic based on a decision signal from the decision application and/or decision application. In one embodiment, the input/output control application determines an alert output characteristic based on a decision signal from the decision application. For example, where the personal computing device is a cellular telephone, upon an incoming call, the device may sense whether the user is viewing its display. If the user's presence is detected, the device may only provide a visual alert via the device's display. If the user's presence is not detected, the device may initiate an audible alert, e.g., ringtone, to alert the user about the incoming call. In this instance, the device may only apply face detection to determine whether any face is present and/or any person is viewing the device's display.



Above diagramis a graphical representation of workflows of stepwise activities and actions with support for choice iteration and concurrency. These are intended to model both computational and organizational processes (i.e. workflows).

When the source is open, the feature extraction is done from the available source. From the extracted features, the image is either stored in the database or compared for existence. If the comparison is done, on success it displays the name of the detected face else shows an error [7].

III. RESULTS AND DISCUSSIONS

The interface for the face detection and recognition project is designed using the Net Beans IDE. This interface serves the to make program operation more intuitive and interactive thus making it easier to learn and use[7].

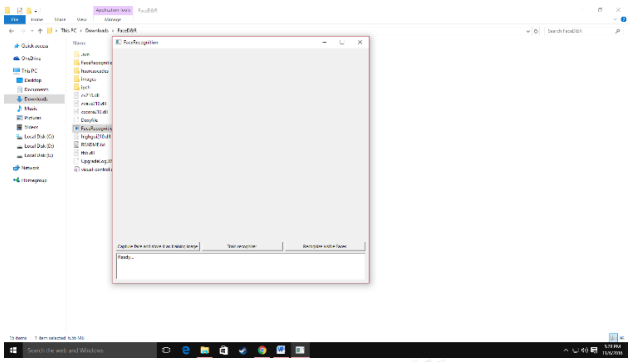


Figure. Interface design preview

5.2. Importing the cascades

Cascading is a particular case of ensemblelearning based on the concatenation of several Classifiers, using all information collected from the output from a given classifier as additional information for the next classifier in the cascade. Unlike voting or stacking ensembles, which are multi expert systems, cascading is a multistage one [8].

Cascading Classifiers are trained with several hundred "positive" sample views of a particular object and arbitrary "negative" images of the same size. After the classifier is trained it can be applied to a region of an image and detect the object in question. To search for the object in the entire frame, the search window can be moved across the image and check every location for the classifier. This process is most commonly used in image processing for object detection and tracking, primarily facial detection and recognition.

Classifiers successfully serve the requirement of implementing the program on low CPU systems, such as cameras and phones. Cascade classifiers are available in OpenCV, with pre-trained cascades for frontal faces and upper body. Training a new cascade in OpenCV is also possible with either haar_training or train_cascades methods. This can be used for rapid object detection of more specific targets, including non-human objects with Haar-like features. These XML files are stored in openCV/ data/ haar cascades/ folder. These files are to be imported into our project file to detect faces. The import statement is:

```
importorg.opencv.objdetect.CascadeClassifier;
CascadeClassifierfaceDetector = new CascadeClassifier
(FaceDetection.class.
getResource("haarcascade_frontalface_alt.xml").
getPath().substring(1));
```

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 1, January 2018

haarcascade_eye.xml	3/8/2015 9:36 AM	XML Document	495 KB
haarcascade_eye_tree_eyeglasses.xml	3/8/2015 9:36 AM	XML Document	1,070 KB
haarcascade_frontalface_alt.xml	3/8/2015 9:36 AM	XML Document	899 KB
haarcascade_frontalface_alt_tree.xml	3/8/2015 9:36 AM	XML Document	3,560 KB
haarcascade_frontalface_alt2.xml	3/8/2015 9:36 AM	XML Document	818 KB
haarcascade_frontalface_default.xml	3/8/2015 9:36 AM	XML Document	1,226 KB
haarcascade_fullbody.xml	3/8/2015 9:36 AM	XML Document	622 KB
haarcascade_lefteye_2splits.xml	3/8/2015 9:36 AM	XML Document	316 KB
haarcascade_lowerbody.xml	3/8/2015 9:36 AM	XML Document	520 KB
haarcascade_profileface.xml	3/8/2015 9:36 AM	XML Document	1,100 KB
haarcascade_righteye_2splits.xml	3/8/2015 9:36 AM	XML Document	317 KB
haarcascade_smile.xml	3/8/2015 9:36 AM	XML Document	276 KB
haarcascade_upperbody.xml	3/8/2015 9:36 AM	XML Document	1,022 KB

Figure. List of available cascades [9]

5.3. Face Detection Module

Face detection and recognition includes many complementary parts where each part is a complement to the other. Depending on regular system each part can work individually. Face detection is a computer technology that is based on learning algorithms to allocate human faces in digital images. Face detection also refers to the psychological process by which humans locate and attend to faces in a visual scene.

Face detection takes images/video sequences as input and locates face areas within these images. This is done by separating face areas from non-face background regions. Then the possible human eye regions are detected by testing all the valley regions in the gray-level image. Feature extraction simplifies face region normalization where detected face is aligned to coordinate framework to reduce the large variances introduced by different face scales, poses and lightning effect caused due to uneven illumination and the shirring effect due to head movement. The accurate locations of feature points sampling the shape of facial features provide input parameters for the face identification.

Face identification generates the final output of complete face-recognition system: the identity of the given face image. Based on normalized face image and facial feature locations derived from previous stages, a feature vector is generated from given face and compared with a database of known faces.

A reliable face-detection approach based on the genetic algorithm and the Eigen-face technique [9].

Firstly, the possible human eye regions are detected by testing all the valley regions in the gray-level image. Then the genetic algorithm is used to generate all the possible face regions which include the eyebrows, the iris, the nostril and the mouth corners.

Each possible face candidates is normalized to reduce lightning effect caused due to uneven illumination and the shirring effect due to head movement. The fitness value of each candidate is measured based on its projection on the Eigen-faces. After a number of iterations, all the face candidates with a high fitness value are selected for further verification. At this stage, the face symmetry is measured.

```
webSource.retrieve(frame);

Graphics g =jPanel1.getGraphics();

faceDetector.detectMultiScale(frame, faceDetections);

for (Rect rect: faceDetections.toArray()){

    Core.rectangle(frame, new Point(rect.x,rect.y),new Point(rect.x + rect.width, rect.y + rect.height),

        new Scalar(0,255,0));
```

Different facial features is verified for each face candidate.

Fig 5.3.1 Code Snippet for detection

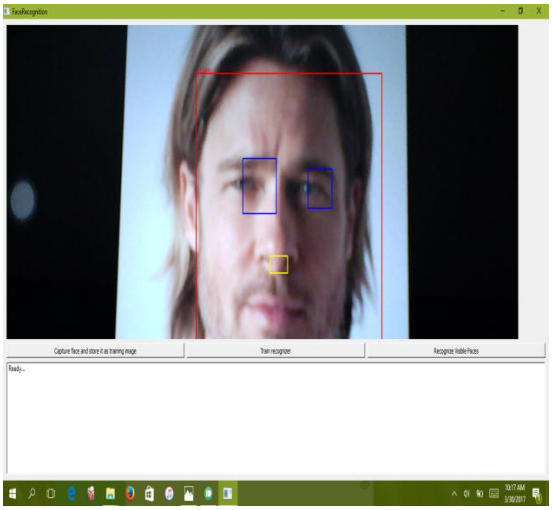


Figure Application detecting face

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 1, January 2018

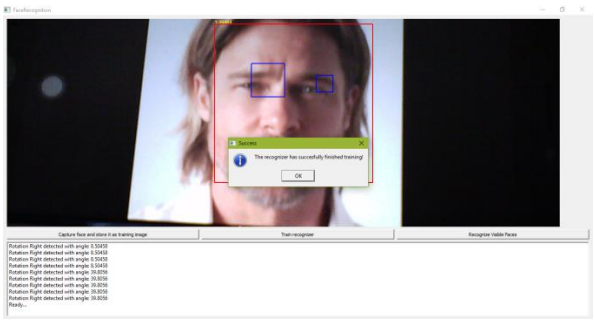


Figure Detecting the faces with Rotation

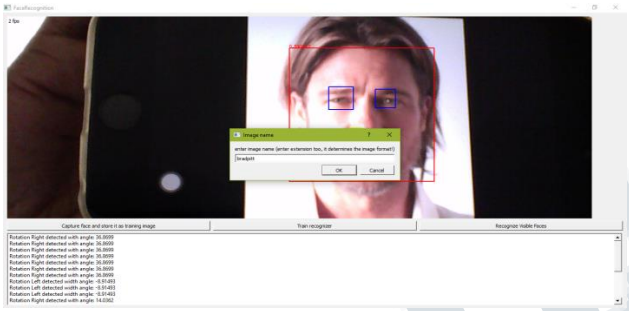


Figure Application prompting to enter the file name with extension

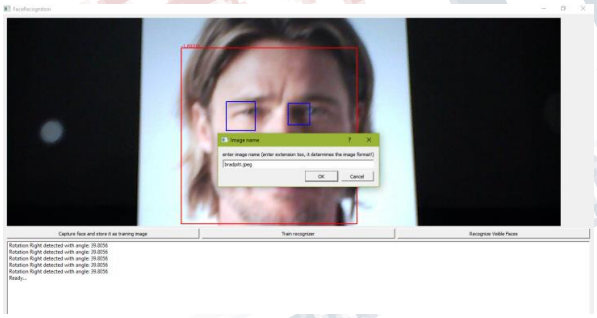


Figure. Pop up on successful Training of the face

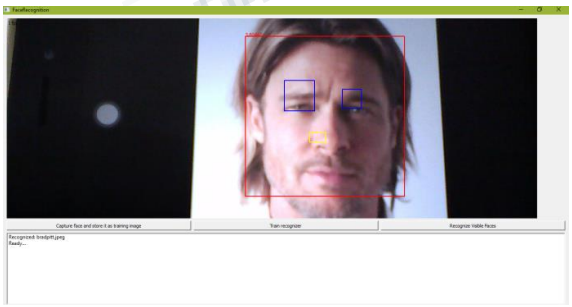


Fig displaying the name of the face

The facial recognition module works with a camera and the Intellect facial detection tool. Firstly, the facial detection tool detects the dat Presence of a face within a video frame and captures its image. The face recognition module has two possible operating modes: identification and verification.

In identification mode, the parameters of a captured face are compared to all facial templates in the database. This helps determine, for example, whether or not the person is blacklisted or a welcomed VIP customer for the particular facility.

In verification mode, the face of an access card holder or a person using another sort of ID to gain access through a turnstile or restricted access door is compared with the rightful pass owner's photo stored in the database. This helps verify that the person requesting access is actually the person he/she claims to be.

The module's settings let you define the levels of similarity (expressed in percentage) that form the boundary limits of what are known as similarity zones. The following three zones can be configured: red (high similarity), yellow (medium similarity), and green (low similarity). If similarity is high, the recognized face is saved to the facial database along with the date and time of recognition, camera ID, and the degree of similarity. To simplify monitoring, the similarity level is color coded on operator's displays.

In addition to recognition, the module lets you delete existing records from the template database. You can also create new records containing images, their parameters and personal such as full name, department, relevant comments, etc. A reference image can be any
Digital photo imported into the database, or any image captured by cameras in operation. The module also lets you verify images' compliance with international standards for automatic personal identification in biometric systems (ISO 197945)

IV. CONCLUSION

The face recognition and detection algorithms were thoroughly studied taking a number of test images and varying the conditions and variables. All the work mentioned above involved real time data. The success rate depends upon certain external factors like brightness, angle with respect to camera, etc. The overall success rate when performed in ideal conditions is 95%. Face

Recognition is a technology just reaching sufficient maturity for it to experience a rapid growth in its practical applications .Face Recognition technology can be utilized to build an automated attendance system that makes counting and identifying students much easier and convenient with robustness and reliability .Further Face Recognition can be used in Anti-theft smart car security, in which one alarm signal could be given to make an alarm or "call" the police and the host soundlessly with the help of other modules in the system prototype. Face Recognition technology can automatically spot shoplifters in a crowd. Verification systems for physical and electronic access security are available today, but the future holds the promise of passive customization and automated surveillance systems enabled by Face Recognition.

REFERENCES

[1] M. Turk, A. Pentland, Eigen faces for Recognition, Journal of Cognitive Neurosicence, Vol. 3, No. 1, Win. 1991, pp. 71-86

[2] Discriminant analysis for recognition of human face images Kamran Etemad and Rama Chellappa

[3] MPCA: Multilinear Principal Component Analysis of Tensor Objects, Haiping Lu, Student Member, IEEE, Konstantinos N. (Kostas) Plataniotis, Senior Member, IEEE, and Anastasios N. Venetsanopoulos, Fellow, IEEE

[4] Face detection ,Inseong Kim, JoonHyung Shim, and Jinkyu Yang

[5] X. Liu and T. Chen. Video-based face recognition using adaptive hidden Markov models. In CVPR, pages I: 340–345. IEEE Computer Society, 2003.

[6] M. Pantic and L. J. M. Rothkrantz. Automatic analysis of facial expressions: The state of the art. I E EE Transaction on Pattern Analysis and MachinesIntel ligence, 22(12):1424–1445, 2000.

[7] R. Brunelli and T. Poggio, "Face Recognition: Features versus Templates", IEEE Trans. on PAMI, 1993, (15)10:1042-1052

[8] R. Brunelli, Template Matching Techniques in Computer Vision: Theory and Practice, Wiley, ISBN 978-0-470-51706-2, 2009.

[9]Duhn, S. von; Ko, M. J.; Yin, L.; Hung, T.; Wei, X. (1 September 2007). "Three-View Surveillance Video Based Face Modeling for Recognition". pp. 1–6. doi:10.1109/BCC.2007.4430529 – via IEEE Xplore.

[10] Socolinsky, Diego A.; Selinger, Andrea (1 January 2004). "Thermal Face Recognition in an Operational Scenario". IEEE Computer Society. pp. 1012–1019 – via ACM Digital Library