International Research Journal of Engineering and Management Studies (IRJEMS)
Volume: 03 Issue: 05 | May -2019 ISSN: 2395-0126

# Smart Access SystemUsing Face Recognition

Kunal Jagdale, Shubham Mishra, Tanmay Jagtap, Prof. D.G.Kannade
kunal.jagdale15@vit.edu, shubham.mishra15@vit.edu, tanmay.jagtap15@vit.edu, dyaneshwar.kannade@vit.edu
Department of Electronics Engineering
Vishwakarma Institute of Technology, Pune

**Abstract- Identifying a person with an image has been popularized through the mass media. However, it is less robust to fingerprint or retina scanning. This report describes the face detection and recognition mini-project undertaken for the visual perception and autonomy module at Plymouth university. It reports the technologies available in the Open-Computer-Vision (OpenCV) library and methodology to implement them using Python. For face detection, Haar-Cascades were used and for face recognition Eigenfaces, Fisherfaces and Local binary pattern histograms were used. The methodology is described including flow charts for each stage of the system. Next, the results are shownincludingplotsandscreen-shotsfollowedbyadiscussionofencounteredchallenges. Thereport isconcludedwiththeauthors'opinionontheprojectand possibleapplications.**

**Key Words – Haar-Cascade, Face-Recognition, Computer Vision, Face detection.**

## 1. Introduction

The following document is a report on the project for Robotic visual perception and autonomy. It involved building a system for face detection and face recognition using several classifiers available in theopencomputervisionlibrary(OpenCV). Face recognitionisanon-invasiveidentificationsystemand fasterthanothersystemssincemultiplefacescanbeanalyzedatthesametime.Thedifferencebetween facedetectionandidentificationis,facedetectionistoidentifyafacefromanimageandlocatetheface. Face recognition is making the decision" whose face is it?", using an image database. In this project bothareaccomplishedusingdifferenttechniquesandaredescribedbelow.Thereportbeginswithabrief historyoffacerecognition.Thisisfollowedbytheexplanationof HAAR-cascades,Eigenface,Fisherface and Local binary pattern histogram (LBPH) algorithms. Next, the methodology and the results of the projectaredescribed.Adiscussionregardingthechallengesandtheresolutionsaredescribed.Finally,a conclusionisprovidedontheprosandconsofeachalgorithmandpossibleimplementations.

## 2. The History of FaceRecognition

Face recognition began as early as 1977 with the first automated system being introduced By Kanade usingafeaturevectorofhumanfaces.[1]In1983,SirovichandKirbyintroducedtheprincipalcomponent analysis(PCA) for feature extraction. Using PCA, Turk and Pentland Eigenface were developed in 1991 and is considered a major milestone in technology. Local binary pattern analysis for texture recognition was introduced in 1994 and is improved upon for facial recognition later by incorporating Histograms(LBPH) [2].In1996Fisherfacewasdevelopedusinglineardiscriminantanalysis(LDA) fordimensionalreductionandcanidentifyfacesindifferentilluminationconditions,whichwasanissue in Eigenface method. Viola and Jones introduced a face detection technique using HAAR cascades andADABoost.In2007,Afacerecognitiontechniquewas developedbyNaruniecandSkarbekusing GaborJetsthataresimilartomammalianeyes [6].InThisproject,HAARcascadesareusedforface detectionandEigenface,FisherfaceandLBPHareusedfor facerecognition.

## 3.Face Detection using Haar-Cascades

AHaarwaveletisamathematicalfictionthatproducessquare-shapedwaveswithabeginningandanend andusedtocreateboxshapedpatternstorecognizesignals

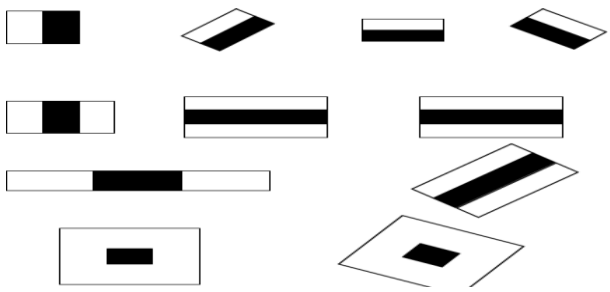
withsuddentransformations.Bycombiningseveral wavelets,acascadecanbecreatedthatcanidentifyedges,lines and circles with different colour intensities.

*Figure 1: Several Haar-like-features matched to the features of authors face.*

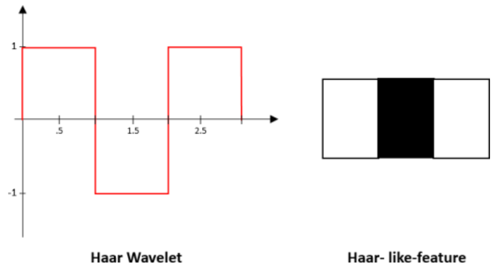
These sets are used in Viola Jones face detection technique in 2001 and since then more patterns are introduced for object detection.

To analyze an image using Haar cascades, a scale is selected smaller than the target image. It is then placed on the image, and the average of the values of pixels in each section is taken. If the difference between two values pass a given threshold, it is considered a match. Face detection on a human face is performed by matching a combination of different Haar-like-features.

For example, forehead,eyebrows andeyescontrastaswellasthenosewitheyesasshownbelow infigureAsingleclassifierisnotaccurate enough.Severalclassifiersarecombinedastoprovideanaccuratefacedetectionsystem. [3]



2.a



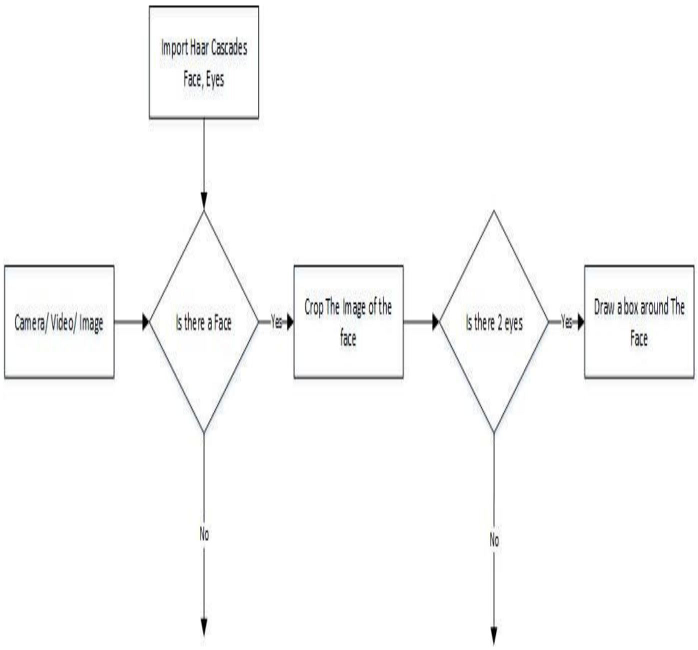

2.b 2.c

*Figure 2a: Different Haar Features*

*Figure 2b: A Haar wavelet*

*Figure 2c: Resulting Haar-like features.*

In this project, a similar method is used effectively to by identifying faces and eyes in combination resulting better face detection. Similarly, in viola Jones method,




several classifies were combined to create stronger classifiers. ADA boost is a machine learning algorithm that tests out several week classifiersonaselectedlocationandchoosethemostsuitable.Itcanalsoreversethedirectionofthe classifier and get better results if necessary.Furthermore, Weight-update-steps can be updatedonly on misses to get better performance. The cascade is scaled by 1.25 and re-

*Figure 3: Haar-cascade flow chart*

iterated in order to find differentsizedfaces.Runningthecascadeonanimageusingconventionalloopstakesalargeamountof computingpowerandtime.ViolaJonesusedasummedareatable(anintegralimage)tocomputethe matchesfast.Firstdevelopedin1984,itbecamepopularafter2001whenViolaJonesimplemented Haar-cascades for face detection.[4] Using an integral image enables matching features with a single pass over theimage.

## 4. Algorithm

- Face Detection and Data Gathering
- Train the Recognizer
- Face Recognition

The below block diagram resumes those phases:

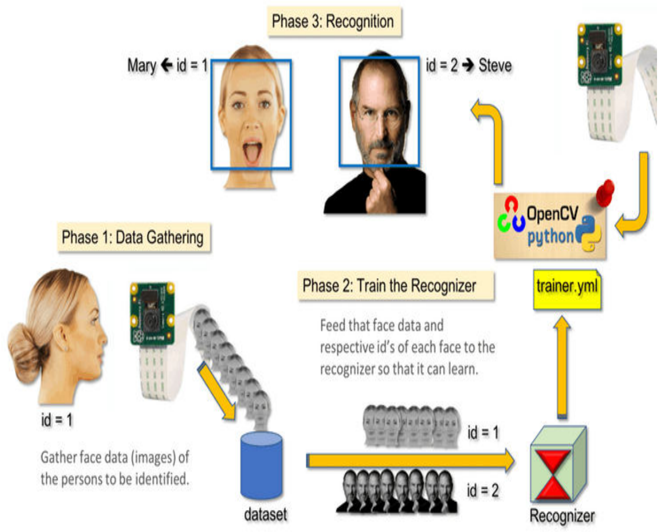



*Figure 4: Algorithm of Face Recognition*

## 5. Proposed Work

Below are the methodology and descriptions of the applications used for data gathering, face detection, training and face recognition.

The project was coded in Python using a mixture of IDLE and PYCharm IDEs.

Face recognition is different of face detection:

- **Face Detection**: it has the objective of finding the faces (location and size) in an image and probably extract them to be used by the face recognition algorithm.
- **Face Recognition**: with the facial images already extracted, cropped, resized and usually converted to grayscale, the face recognition algorithm is responsible

for finding characteristics which best describe the image.

**FaceDetection**

First stage was creating a face detection system using Haar-cascades. Although, training is required for creating new Haar-cascades, OpenCV has a robust set of Haar-cascades that was used for the project. Usingface-cascadesalonecausedrandomobjectstobeidentifiedandeyecascadeswereincorporatedto obtain stable face detection. The flowchart of the detection system can be seen in figure.
Face andeye classifier objects are created using classifier class in OpenCV through the **cv2.CascadeClassifter()** and loading the respective XML files. A camera object is created using the **cv2.VideoCapture()** to capture images. By using the **CascadeClassifter.detectMultiScale()** object of various sizes are matched and location is returned. Using the location data, the face is cropped for further verification. Eye cascade is used to verify there are two eyes in the cropped face. If satisfied a marker is placed around the face to illustrate a face is detected in the location.

**Face RecognitionProcess**

For this project three algorithms are implemented independently. These are Eigenface, Fisherface and Linear binary pattern histograms respectively. All three can be implemented using OpenCV libraries. There are three stages for the face recognition as follows:

1. Collecting imagesIDs
2. ExtractinguniquefeaturesclassifyingthemandstoringinXMLfiles
3. MatchingfeaturesofaninputimagetothefeaturesinthesavedXMLfilesandpredictidentity.

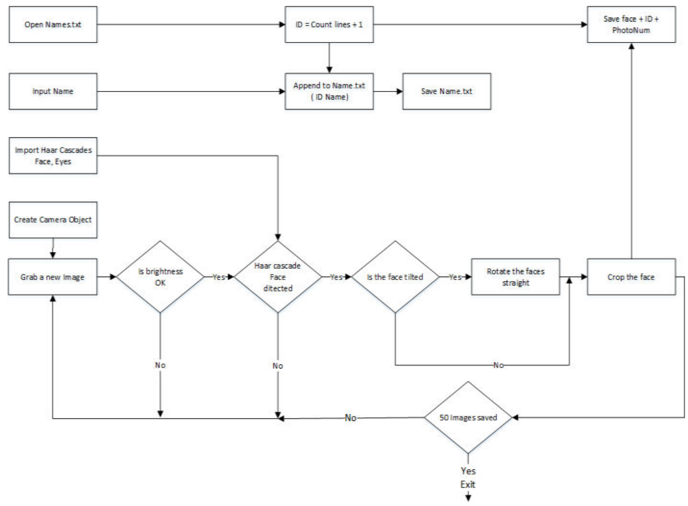
## 6. LBPH Algorithm

1. **Parameters**: the LBPH uses 4 parameters:

- **Radius**: the radius is used to build the circular local binary pattern and represents the radius around the central pixel. It is usually set to 1.
- **Neighbors**: the number of sample points to build the circular local binary pattern. Keep in mind: the more sample points you include, the higher the computational cost. It is usually set to 8.
- **Grid X**: the number of cells in the horizontal direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8.
- **Grid Y**: the number of cells in the vertical direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8.

**2. Training the Algorithm**: First, we need to train the algorithm. To do so, we need to use a dataset with the facial images of the people we want to recognize. We need to also set an ID (it may be a number or the name of the person) for each image, so the algorithm will use this information to recognize an input image and give you an output. Images of the same person must have the same ID. With the training set already constructed, let's see the LBPH computational steps. [10]

**3. Applying the LBP operation**: The first computational step of the LBPH is to create an




intermediate image that describes the original image in a better way, by highlighting the facial characteristics. To do so, the algorithm uses a concept of a sliding window, based on the parameter's radius and neighbors [5].

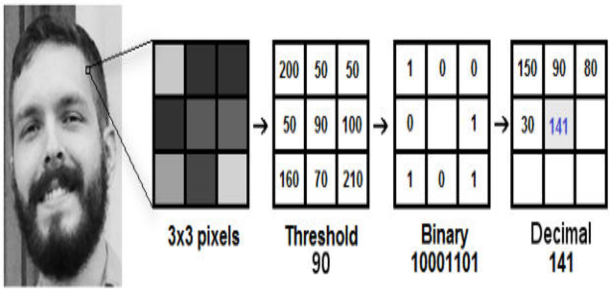
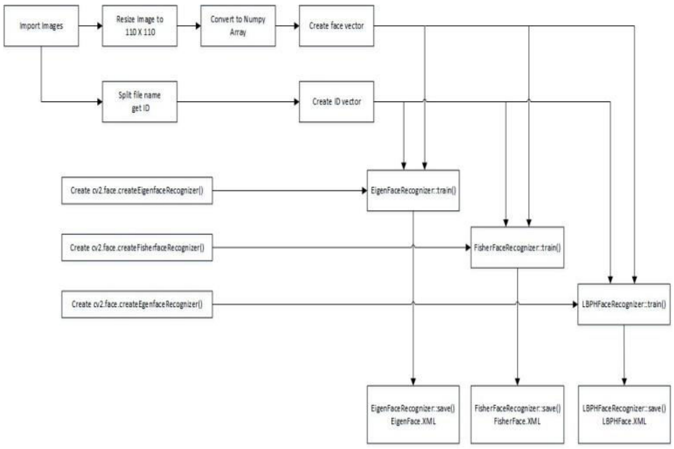



*Figure 5: LBHF Algorithm [9]*

Based on the image above, let's break it into several small steps so we can understand it easily:Suppose we have a facial image in grayscale.

- We can get part of this image as a window of 3x3 pixels.
- It can also be represented as a 3x3 matrix containing the intensity of each pixel (0~255).
- Then, we need to take the central value of the matrix to be used as the threshold.
- This value will be used to define the new values from the 8 neighbors. [8]

*Figure 6: Flowchart for Image Collection*




- For each neighbor of the central value (threshold), we set a new binary value. We set 1 for values equal or higher than the threshold and 0 for values lower than the threshold.
- Now, the matrix will contain only binary values (ignoring the central value). We need to concatenate each binary value from each position from the matrix line by line into a new binary value (e.g. 10001101). Note: some authors use other approaches to concatenate the binary values (e.g. clockwise direction), but the final result will be the same.
- Then, we convert this binary value to a decimal value and set it to the central value of the matrix, which is actually a pixel from the original image.[11]

### Collecting the imagedata

*Figure 7: The Flowchart for the image collection.*

Collecting classification images is usually done manually using a photo editing software to crop and resize photos. [9] Furthermore, PCA and LDA requires the same number of pixels in all the images for the correct operation. This time consuming and a laborious task is automated through an application to collect50imageswithdifferentexpressions.Theapplicationdetectssuitableexpressionsbetween300ms, straightensanyexistingtiltandsavethem.TheFlowchartfor theapplicationisshowninfigure 7.

Application starts with a request for a name to be entered to be stored with the ID in a text file. The facedetectionsystemstartsthefirsthalf.[12]However,beforethecapturingbegins,theapplication checkfor the brightness levels and will capture only if the face is well illuminated. Furthermore, after the face is detected,thepositionoftheeyesisanalyzed.Iftheheadistilted,theapplicationautomaticallycorrects the orientation. These two additions were made considering the requirements for Eigenface algorithm. The Image is then cropped and saved using the ID as a filename to be identified later. A loop runs thisprogramuntil50viableimagesarecollectedfromtheperson.Thisapplicationmadedatacollection efficient.

## 7. Results

*Figure 8: 200 image training data in grayscale*

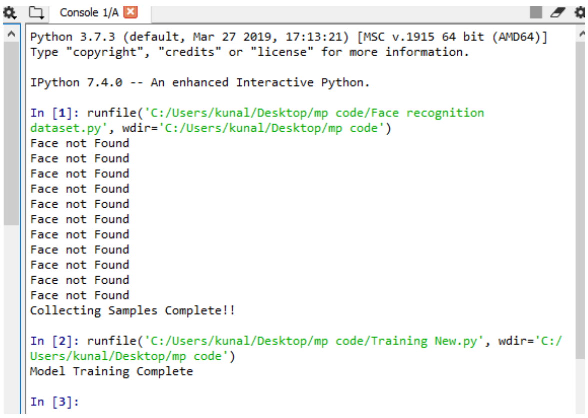



*Fig 9: Samples collected and Training model complete*


*Figure 10: No Capture if face not found*

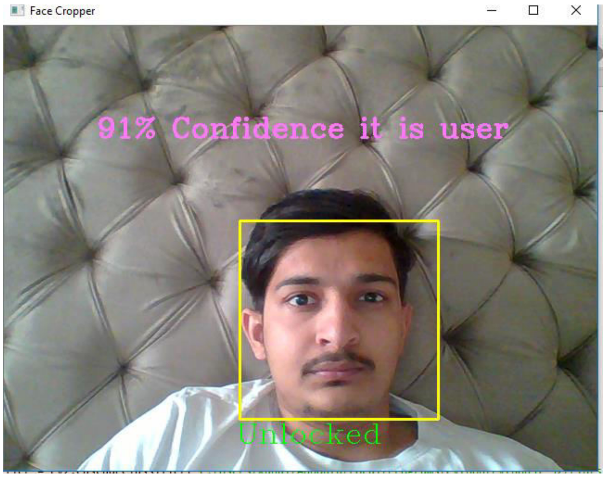



*Figure 11: Face labeled for the detected personby the test-trained data.*

## 8. Conclusion

This paper describes the mini-project for visual perception and autonomy module. Next, it explains the technologies used in the project and the methodology used. Finally, it shows the results, discuss the challengesandhowtheywereresolvedfollowedbyadiscussion.UsingHaar-cascadesforfacedetection worked extremely well even when subjects wore spectacles. Real time video speed was satisfactory as well devoid of noticeable frame lag. Considering all factors, LBPH combined with Haar-cascades can be implemented as a cost effective face recognition platform. An example is a system to identify known troublemakers in a mall or a supermarket to provide the owner a warning to keep him alert or for automaticattendancetakinginaclass.

## 9. References

[1] TakeoKanade.*Computerrecognitionofhumanfaces*,volume47.BirkhäuserBasel,1977.

[2] LawrenceSirovichandMichaelKirby.Low-dimensionalprocedureforthecharacterizationofhuman faces. *Josa a*, 4(3):519–524,1987.

[3] M.TurkandA.Pentland.Eigenfacesforrecognition.*JournalofCognitiveNeuroscience*,3(1):71–86, Jan1991.

[4] DongchenHeandLiWang.Textureunit,texturespectrum,andtextureanalysis.*IEEETransactions on Geoscience and Remote Sensing*, 28(4):509–512, Jul1990.

[5] X.Wang,T.X.Han,andS.Yan.Anhog-lbphumandetectorwithpartialocclusionhandling.In *2009 IEEE 12th International Conference on Computer Vision*, pages 32–39, Sept 2009.

[6] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, Jul1997.

[7] P.ViolaandM.Jones.Rapidobjectdetectionusingaboostedcascadeofsimplefeatures.In*Proceed- ings of the 2001 IEEE Computer Society Conference on Computer Vision and PatternRecognition. CVPR 2001*, volume 1, pages I–511–I–518 vol.1,2001.

[8] John G Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *JOSA A*, 2(7):1160–1169,1985.

[9] SMarĉelja. Mathematicaldescriptionoftheresponsesofsimplecorticalcells.*JOSA*,70(11):1297– 1300,1980.

[10] RainerLienhartandJochenMaydt.Anextendedsetofhaar-likefeaturesforrapidobjectdetection. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 1, pages I–I. IEEE,2002.

[11] John P Lewis. Fast template matching. In *Vision interface*, volume 95, pages 15–19,1995.

[12] MengXiaoHeandHelenYang.Microarraydimensionreduction,2009.