ATTENDANCE SYSTEM USING FACE DETECTION USING PYTHON-ML

# A PROJECT REPORT

***Submitted by***

**DHANUSHKUMAR R(210701051)**

**ABHISEK K(2010701524)**

**ROHITH Y(2010701519)**

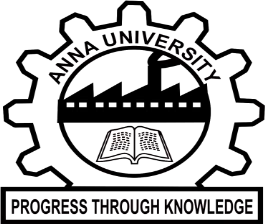
***in partial fulfillment for the award of the degree of***

**BACHELOR OF ENGINEERING IN**

**COMPUTER SCIENCE**

**RAJALAKSHMI ENGINEERING COLLEGE**

**THANDALAM**





**ANNA UNIVERSITY CHENNAI 600 025 DECEMBER-2022**

**BONAFIDE CERTIFICATE**

This is to certify that this project report titled **“ATTENDANCE SYSTEM USING FACE DETECTION USING PYTHON-ML**” is the bonafide work of **“DHANUSHKUMAR.R(210701051)** , and **ABHISEK(210701524), ROHITH Y(210701519)”** who carried out the project work under my supervision.

**SIGNATURE**

**DR.VINOTHKUMAR S ,**

Assistant Professor (SS)

Department of Computer Science and Engineering Rajalakshmi Engineering College

Chennai - 602 105

This project report is submitted via viva voce examination to be held on

… at Rajalakshmi Engineering College, Thandalam.

**EXTERNAL EXAMINER INTERNAL EXAMINER**

# ACKNOWLEDGEMENT

First and foremost, I acknowledge the amazing Grace of God Almighty, who blessed my efforts and enabled me to complete this thesis in good health, mind, and spirit.

I am grateful to my Chairman **Mr.S.Meganathan**, Chairperson **Dr.Thangam Meganathan**, Vice Chairman **Mr.M.Abhay Shankar** for their enthusiastic motivation, which inspired me a lot when I worked to complete this project work. I also express our gratitude to our principal **Dr.S.N.Murugesan** who helped us in providing the required facilities in completing the project.

I would like to thank our Head of Department **Dr. P. KUMAR** for his guidance and encouragement for completion of project.

I would like to thank **MRS. SUSMITA MISHRA, M.TECH,** our supervisor for constantly guiding us and motivating us throughout the course of the project. We express our gratitude to our parents and friends for extending their full support to us.

**ABSTRACT**

Machine learning has been gaining momentum over last decades: self-driving cars, efficient web search, speech and image recognition. The successful results gradually propagate into our daily live. Machine learning is a class of artificial intelligence methods, which allows the computer to operate in a self-learning mode, without being explicitly programmed. It is a very interesting and complex topic, which could drive the future of technology. Face detection is an important step in face recognition and emotion recognition, which is one of the more representative and classic application in computer vision. Face is one of the physiological bio-metrics based on stable features. Face detection by computer systems has become a major field of interest. Face detection algorithms are used in wide range of applications, such as security control, video retrieving, biometric signal processing, human computer interface, emotion detection, face recognition and image database management. Face detection is a challenging mission because faces in the images are all uncontrolled. E.g. illumination condition, vary pose, different facial expressions.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO** | **TITLE** | **PAGE NO.** |
|  | **ACKNOWLEDGEMENT** | **iii** |
|  | **ABSTRACT** | **iv** |
|  | **LIST OF FIGURES** | **vii** |
|  | **LIST OF TABLES** | **viii** |
|  | **LIST OF SYMBOLS** | **ix** |
|  | **LIST OF ABBREVIATIONS** | **x** |
| **1** | **INTRODUCTION** | **1** |
| 1.1 | INTRODUCTION | **1** |
| 1.2 | SCOPE OF WORK | **2** |
| 1.3 | PROBLEM STATEMENT | **2** |
| 1.4 | AIM AND OBJECTIVE OF THE PROJECT | **2** |
| **2** | **LITERATURE SURVEY** | **3** |
| **3** | **SYSTEM SPECIFICATION** | **4** |
| 3.1 | REQUIREMENT SPECIFIACTION | **4** |
| 3.1.1 | HARDWARE REQUIREMENTS | **4** |
| 3.1.2 | SOFTWARE REQUIREMENTS | **5** |
| 3.2 | PROPOSED SYSTEM | **5** |
| **4** | **SYSTEM DESIGN** | **6** |
| 4.1 | SYSTEM ARCHITECTURE | **6** |
| **5** | **RESULT AND DISCUSSION** | **7** |
| 5.1 | DATACOLLECTION | **7** |
| 5.2 | DATA PREPROCESSING | **8** |
| 5.3 | ALGORITHMS AND OUTPUTS | **10** |
| **6** | **CONCLUSON AND FUTURE WORK** | **17** |
| 6.1 | CONCLUSION | **17** |
| 6.2 | FUTURE WORK | **17** |
|  | **REFERENCES** |  |
|  | **APPENDIX** |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **NAME OF FIGURES** | **PAGE NO.** |
| 3.1.1 | Proposed System | 5 |
| 4.1 | System Architecture | 6 |
| 5.1.1 | Datasets of Diabetes Prediction | 7 |
| 5.2.1 | Preprocessing of Data | 8 |
| 5.3.1 | Output Screenshot | 16 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **TABLE**  **NO.** | **TABLE NAME** | **PAGE**  **NO.** |
| 4 | Comparison Of Algorithms Based On Performance –  Python | 15 |

**LIST OF SYMBOLS**

Dataset

Denotes the dataset used for both training and testing the model using different algorithms.

Process

This denotes various process involved in the development of proposed system

This arrow indicates the flow from one process to the another process.

This indicates the Stages in the proposed system

**,**

It indicates start and the end stage of the

**,** process.

It denotes direction of flow between different stages.

**ABBREVIATIONS**

|  |  |
| --- | --- |
| ML : | Machine Learning |
| SVM : | Support Vector Machine |

**CHAPTER 1 INTRODUCTION**

* 1. **INTRODUCTION**

Machine learning has been gaining momentum over last decades: self-driving cars, efficient web search, speech and image recognition. The successful results gradually propagate into our daily live. Machine learning is a class of artificial intelligence methods, which allows the computer to operate in a self-learning mode, without being explicitly programmed. It is a very interesting and complex topic, which could drive the future of technology. Face detection is an important step in face recognition and emotion recognition, which is one of the more representative and classic application in computer vision. Face is one of the physiological bio-metrics based on stable features. Face detection by computer systems has become a major field of interest. Face detection algorithms are used in wide range of applications, such as security control, video retrieving, biometric signal processing, human computer interface, emotion detection, face recognition and image database management. Face detection is a challenging mission because faces in the images are all uncontrolled. E.g. illumination condition, vary pose, different facial expressions.

* 1. **SCOPE OF WORK:**

The most useful area in which face recognition is important is the biometrics that is used for authentication process which makes the work mor easier. Face recognition is one of the widely used technologies or systems in which it has the potential to perform tasks such as to have records provided in by the dataset in many areas such as the school and colleges attendance systems, it can also be helpful in catching the thieves or the terrorist, can be helpful in the security of common people.

* 1. **PROBLEM STATEMENT:**

Problem Statement The main aim or objective of this paper is to provide or develop a system that will use the camera of the computer or the system that would detect and recognize the person’s face or the face of the individual using the tool in OpenCV called as the Open Face and python programming language in deep learning domain

* 1. **AIM AND OBJECTIVE:**

T Face recognition can be used by the government to verify the voters list, find missing persons, find the population or census, immigration process, also provide security over internet scams protecting Ecommerce and highly used in the medicine and healthcare range. This brings in a very high demand or a real time face recognition system for several uses for the people and government. Providing such excellent systems there would be ease in several activities.

**CHAPTER 2**

# LITERATURE SURVEY

This section is a basic overview of the major techniques used in the face recognition system that apply mostly to the front face of the human being. The methods include neural networks, hidden Markov model, face matching done geometrically and template matching. Eigenface is one of the most widely used methods in face recognition and detection which are broadly called as the principle components in mathematical terms. The eigenvectors are ordered to represent different amounts of the variations in the faces. Neural networks are highly used in the face recognition and detection systems. An ANN (artificial neural network) Was used in face recognition which contained a single layer Which shows adaptiveness in crucial face recognition systems. The face verification is done using a double layer of WISARD in neural networks. Graph matching is other option for face recognition. The object as well as the face recognition can be formulated using graph matching performed by optimization of a matching function. Hidden Markov Models is the way by which stochastic modeling of nonstationary vector time series based on HMM model applied to

Athe human face recognition wherein the faces gets divided into parts such as the eyes, nose, ears, etc The face recognition and correct matching is 87% correct as it always gives out the best and right choice of face detection through stored dataset. Or else the relevant model reveals the identity of the face. The geometrical feature matching is the technique which is based on the geometrical shapes of the face.

The geometrical face configuration has sufficient dataset for face detection and recognition system. This is one of the commonly used methods of the face recognition and detection. This system apparently gives satisfactory results. Template matching is one of the techniques through which the test image is represented as a two- dimensional array of values which can be compared using Euclidean distance with single template representing the whole face. This method can also use more than one face template from different points of view to represent an individual face.

**CHAPTER 3**

**SYSTEM SPECIFICATION**

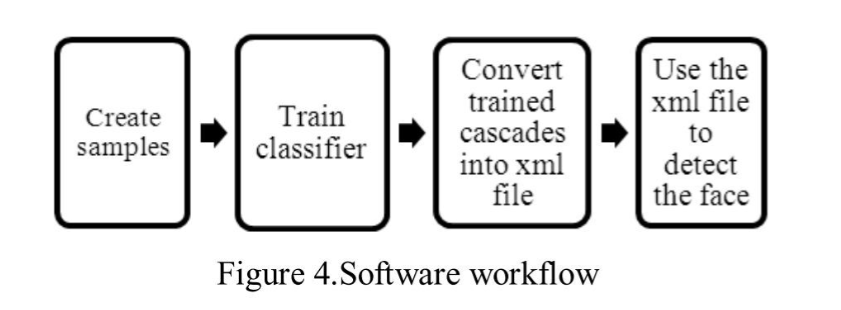
* 1. **REQUIREMENT SPECIFICATION**
     1. **HARDWARE REQUIREMENTS**
* Processors - 11th Gen Intel(R) Core(TM) i5
* Speed - 2.40GHz
* RAM - 2 GB
* Storage - 20 GB
  + 1. **SOFTWARE REQUIREMENTS**
* Operating system - Windows 11 Home
* IDE used - Visual Studio Code

- Google Colab

* Python Libraries- Numpy, pandas, OpenCv,Face\_dection

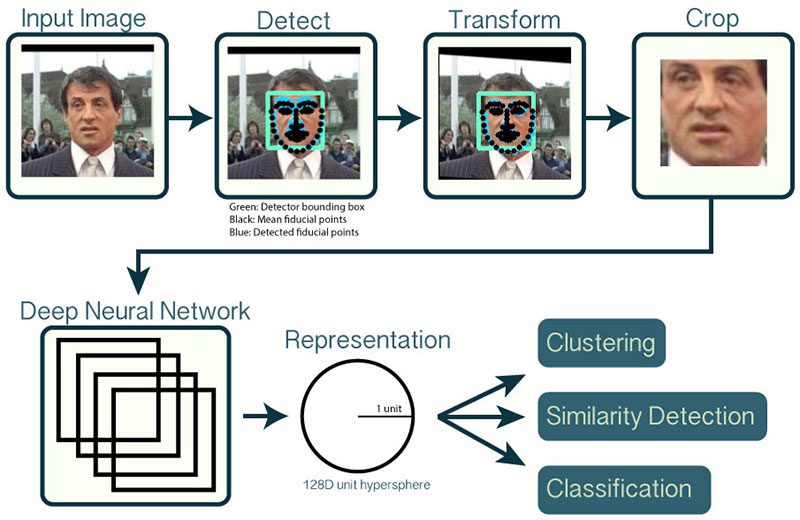
**3.1 PROPOSED SYSTEM**

In order to create this system first we will have to make the datasets. When the image quality becomes favourable different procedures will take place in the face recognition system the tasks are performed using the python queries “python encode\_faces.py”. The input will be taken from the dataset which will be received in the “encodings.py”. There will be precision formatting in the system wherein face embedding for each face will occur. Secondly a file “recognize\_faces\_images.py” will contain all the required methods and the techniques for the process of identification of the face of the person from the given image of the dataset. The given file will be executed by the python command “python recognize\_faces\_image.py-encodings”. We can resize or turn the image for approximity with the goal for getting the desired output. The present classifier along with OpenCV libraries will enhance the outcome or results in the face recognition system.



# CHAPTER 4

**SYSTEM DESIGN SYSTEM ARCHITECTURE :**



**Fig 2: System Architecture**

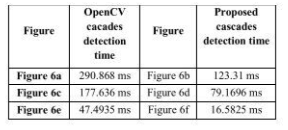
# CHAPTER 5

**RESULT AND DISCUSSION**

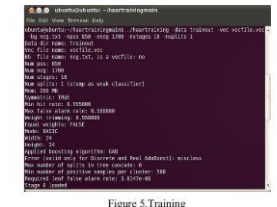
* 1. **DATASET COLLECTION**

**Dataset Description**

* + - Facial detection and eye detection using OpenCV typically involve using computer vision techniques to identify and locate faces and eyes within images or video streams. OpenCV is an open-source computer vision library that provides a wide range of tools and algorithms for image processing and object detection.

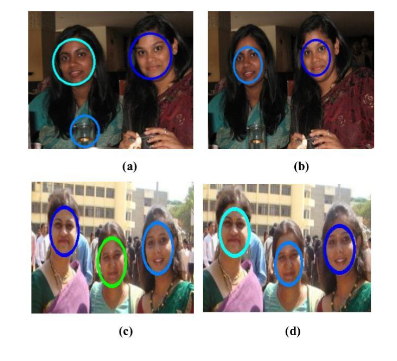


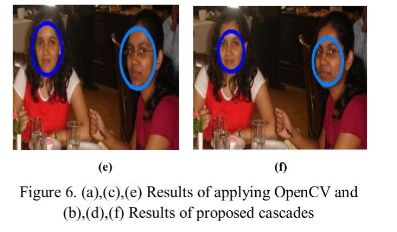
**Fig 3:dataset**

****

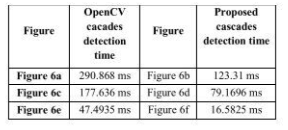
* 1. **FACE-DETECTION MODEL**

It Contains a pretrained Caffe deep learning model provided by OpenCV to detect faces. This model detects and localizes face in an image A Serialized facial embeddings file. Embeddings have been computed for every face in the dataset





TIME TAKEN FOR OPENCV TO DETECTION FACE OF A,B,C,D



Our Linear Support Vector Machine (SVM) model. This is a machine learning model rather than a deep learning model and it is responsible for actually recognizing faces.

# Algorithms and outputs:

First ,The concept of OpenCV was put forth by Gary Brad ski which had the ability to perform on multilevel framework. OpenCV has a number of significant abilities as well as utilities appears from the outset. The OpenCV helps in recognizing the frontal face of the person and also creates XML documents for several areas such as the parts of the body. Deep learning evolved lately in the process of the recognition systems. Hence deep learning along with the recognition together work as the deep metric learning systems. In short deep learning in face detection and recognition will broadly work on two areas the first one being accepting the solidary input image or any other relevant picture and the second being giving the best outputs or the results of the image of the picture. We would be using dlib facial recognition framework that would be the easy way to organize the face evaluation. The two main significant libraries used in the system are dlib and face recognition. Python being a very powerful programming languages and one of the programming languages that are being used all over the world has proven to give best results in the face recognition and detection systems. Together face recognition and detection becomes very easy and fruitful with the help of the python programming language and OpenCV.

### OpenCV for Face Detection in Images

We will build a detector to identify the human face in a **photo** from Unsplash. Make sure to the picture to your working directory and rename it to input\_image before coding along.

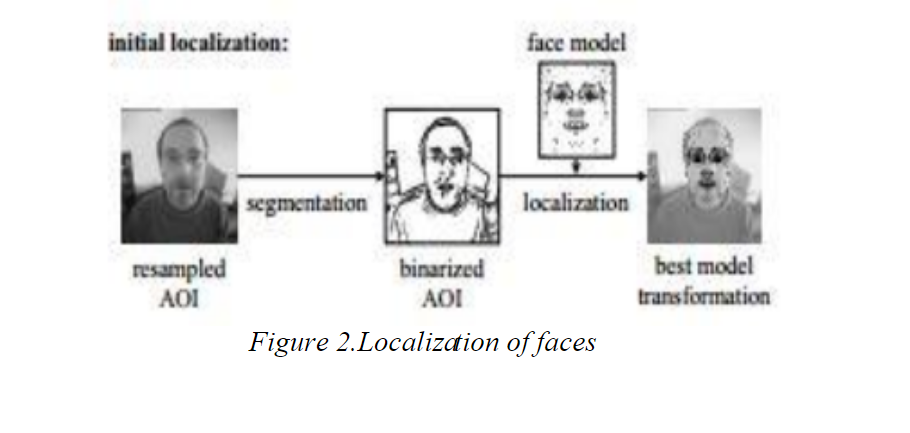
Now, let’s import OpenCV and enter the input image path with the following lines of code

import cv2

imagePath = 'input\_image.jpg'

Then, we need to read the image with OpenCV’s imread() function, Notice that this is a 3-dimensional array. The array’s values represent the picture’s height, width, and channels respectively. Since this is a color image, there are three channels used to depict it - blue, green, and red (BGR). Note that while the conventional sequence used to represent images is RGB (Red, Blue, Green), the OpenCV library uses the opposite layout (Blue, Green, Red).

To improve computational efficiency, we first need to convert this image to grayscale before performing face detection on it: Notice that we are using a file called haarcascade\_frontalface\_default.xml. This classifier is designed specifically for frontal faces in visual input.



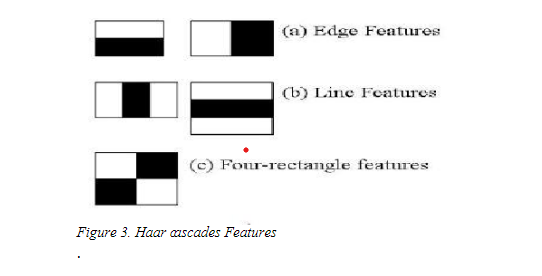
OpenCV also provides other pre-trained models to detect different objects within an image - such as a person’s eyes, smile, upper body, and even a vehicle’s license plate. You can learn more about the different classifiers built into OpenCV by examining the library’s [**GitHub repository**](https://github.com/opencv/opencv/tree/master/data/haarcascades).We can now perform face detection on the grayscale image using the classifier we just loaded:

The cascade classifier applies a sliding window through the image to detect faces in it. You can think of these windows as rectangles.

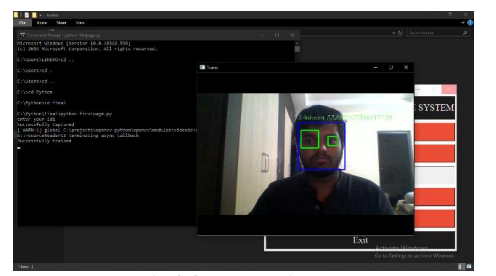
Initially, the classifier will capture a large number of false positives. These are eliminated using the minNeighbors parameter, which specifies the number of neighboring rectangles that need to be identified for an object to be considered a valid detection.

To summarize, passing a small value like 0 or 1 to this parameter would result in a high number of false positives, whereas a large number could lead to losing out on many true positives.

The trick here is to find a tradeoff that allows us to eliminate false positives while also accurately identifying true positives.



**OUTPUT**:



**Fig 12: Output Screenshot**

**CHAPTER 6**

**CONCLUSION AND FUTURE WORK**

**6.1 CONCLUSION**

Face recognition systems are currently associated with many top technological companies and industries making the work of face recognition easier. The use of python programming and OpenCV makes it an easier and handy tool or system which can be made by anyone according to their requirement. The proposed system discussed in this project will be helpful for many as it is user friendly and cost\_ efficient system. Hence by the use of python and OpenCV the face recognition system can be designed for various purposes

**6.2 FUTURE WORK**

The main aim of this project was to design and implement Facial detection and eye detection using OpenCV are essential components of many computer vision applications, and there are several directions for future work and improvement in this area. Here are some potential future work areas:

approach Facial Expression Recognition: Expand the system to not only detect faces but also recognize facial expressions, which can be useful in emotion analysis and human-computer interaction. Face Recognition: Extend the system to perform face recognition in addition to detection. This can be used for tasks like access control, identity verification, and personalized user experiences.

**REFERENCES**

[1]. Aljumah, A.A., Ahamad, M.G., Siddiqui, M.K., 2013. Application of data mining: Diabetes health care in young and old patients. Journal of King Saud University - Computer and Information Sciences 25, 127–136. doi:10.1016/j.jksuci.2012.10.003.

[2]. Arora, R., Suman, 2012. Comparative Analysis of Classification Algorithms on Different Datasets using WEKA. International Journal of Computer Applications 54, 21–25. doi:10.5120/8626-2492.

[3]. Bamnote, M.P., G.R., 2014. Design of Classifier for Detection of Diabetes Mellitus Using Genetic Programming. Advances in Intelligent Systems and Computing 1, 763–770. doi:10.1007/978-3319-11933-5.

[4]. Choubey, D.K., Paul, S., Kumar, S., Kumar, S., 2017. Classification of Pima indian diabetes dataset using naive bayes with genetic algorithm as an attribute selection, in: Communication and Computing Systems: Proceedings of the International Conference on Communication and Computing System (ICCCS 2016), pp. 451– 455.

[5]. Dhomse Kanchan B., M.K.M., 2016. Study of Machine Learning Algorithms for Special Disease Prediction using Principal of Component Analysis, in: 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication, IEEE. pp. 5–10.

**APPENDIX**

import cv2  
  
trainedDataset = cv2.CascadeClassifier('haarcascade\_frontalface\_default.xml')  
eye\_cascade = cv2.CascadeClassifier('haarcascade\_eye.xml')  
  
video = cv2.VideoCapture(0)  
  
while True:  
 ret, img = video.read()  
 gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)  
 faces = trainedDataset.detectMultiScale(gray)  
  
 for (x, y, w, h) in faces:  
 cv2.rectangle(img, (x, y), (x + w, y + h), (0, 0, 255), 2)  
 face\_gray = gray[y:y + h, x:x + w]  
 face\_color = img[y:y + h, x:x + w]  
 print(faces)  
  
 eyes = eye\_cascade.detectMultiScale(face\_gray)  
 for (ex, ey, ew, eh) in eyes:  
 cv2.rectangle(face\_color, (ex, ey), (ex + ew, ey + eh), (0, 255, 0), 2)  
  
 cv2.imshow('img', img)  
 k = cv2.waitKey(30) & 0xff  
 if k == 113: # Press 'q' key to exit  
 break  
  
video.release()  
cv2.destroyAllWindows()