Face Detection and Recognition using OpenCV

Instructor: Dr. Jinan Fiaidhi Urvil Ashokkumar Patel Jhanvi Dilipkumar Upadhyay

Professor at Lakehead University Student Id: 1131703 Student Id: 1159217

[upatel19@lakeheadu.ca](mailto:upatel19@lakeheadu.ca) [jupadhya@lakeheadu.ca](mailto:jupadhya@lakeheadu.ca)

M.Sc. Computer Science M.Sc. Computer Science

Lakehead University Lakehead University

1. **ABSTRACT**

In today’s era, face detection and recognition from an image or video is a popular topic in biometric research. The growing interest in computer vision over the past decade brings new technologies like face detection and recognition, which grabs the attention of today’s IT professionals. Face detection and recognition is a popular area of research in computer vision, especially in image analysis and algorithm-based understanding. Face recognition technology is widely used in real-time video surveillance systems. Our main aim is to detect and recognize human faces and further extend that idea into an automated attendance system. It will add the student attendance in the excel sheet automatically once his/her face is recognized. In this paper, we represent a methodology of how our projects work. For that, we will use some of the basic python libraries like OpenCV, NumPy and we are also going to use a Haar-like classifier and local binary pattern histogram algorithm to extract the facial features from digital images.

1. **KEYWORDS**

Face Detection, Face Recognition, Haar like Feature, OpenCV, NumPy, LBPH algorithm, Feature extraction

1. **INTRODUCTION**

Facial recognition is the method by which the identity of a human being can be identified using an individual face. The art of recognizing the human face is quite challenging because of the human face expression, age, change in hairstyle, etc. Although there will be many methods, have been proposed to recognize the human face, but in this project, we will use the OpenCV library, which is developed explicitly for real-time computer vision applications. This system contains three modules which are face detection, training, and recognition, by applying various algorithms. This method is helpful in many fields such as the military, security, schools, colleges and universities, airlines, banking, online web applications, gaming, etc. this system uses a robust python algorithm through which the detection and recognition of face are straightforward and efficient [[1]](#b1).

Face recognition has some unique characteristics that other biometrics do not have. Facial images can be captured from some specific distance, and no particular action is required for authentication. It is a touchless mode of attendance system, which will be very helpful in the current covid pandemic. While in other ways of attendance like a fingerprint in which a person has to put his/her finger to register their attendance. Due to such types of characteristics, face recognition technology is widely used in many organizations, schools, and colleges.

1. **PROBLEM DEFINITION**

Maintaining the attendance records with day-to-day activities is a challenging task. Traditionally, in schools and colleges, the attendance is taken manually by using an attendance sheet, which is time-consuming and will add the extra workload on the teacher. Moreover, it is challenging to verify one by one student in a large classroom whether the authenticated students are actually responding or not. The use of face recognition for the purpose of attendance marking is the intelligent way of an attendance management system. Face recognition is a more accurate and faster technique among other techniques and reduces the chance of proxy attendance [[2]](#b2).

1. **FACE DETECTION & FACE RECOGNITION**

**Face Detection:** Face detection is an artificial intelligence-based computer technology used to identify only human faces in digital images. It simply means that the face detection system can identify human face is present or not- it cannot identify that particular person. There will be other objects presents in an image like road, trees, bungalow, etc., but the primary aim of face detection algorithms is to determine whether there is any human face in an image or not [[3]](#b3).

**How does face detection works?**

There are various libraries, algorithms that helps to identify a human face in an image. Typically, the first thing that the face detection algorithm will look for is the eyes because eyes are the most accessible features to identify. Once eyes are detected, the algorithm might attempt to detect facial regions, including eyebrows, the mouth, nose, etc. [[4]](#b4).

**Face Recognition:** Facial recognition is the process of identifying or confirming an individual’s identity using their face. It can be used to identify people in photos, videos, or in real-time. So, face recognition is the task of identifying an already detected face as a known or unknown face, and in more advanced cases, telling exactly whose face it is.

It has two sets of tasks:

1. **Face Identification:** Given an face image that belongs to a person in a database and we need to tell whose image it is or specifically recognize a face in an image and give decision whether the face is correctly recognize or not.
2. **Face Verification:** Given Face image that might not belong to database and we need to authenticate whether a correct face is subjected to the database or not.
3. **METHODOLOGY**

There are mainly five steps for face recognition.

1. Image Acquisition
2. Image Processing
3. Face detection
4. Feature Extraction
5. Recognize face

This is the actual block diagram of our system, which shows how the actual process flow works—starting from the high-quality camera that captures an image of a person and converts that image from RGB to grayscale. The next step is to apply the various algorithms to extract the facial features from that grayscale image, and then it will be compared with the known database faces by using a unique student ID, and once the match is found, the attendance is automatically marked in the excel sheet.

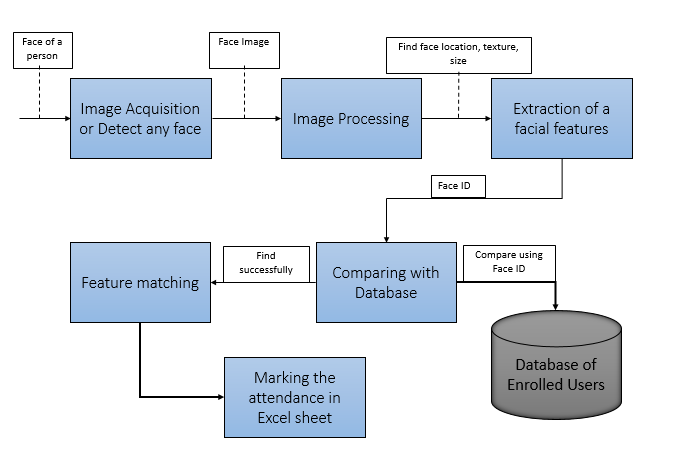


Figure 1. Block Diagram

1. **Image Acquisition**

As the name suggests, Acquisition means to acquire or capture anything. Image acquisition can be accomplished by capturing real-time images of any object using a high-optical camera and generates images of sufficient quality and resolution. Image acquisition can be made by digitally scanning an existing photograph or by using high-quality sensors like the camera to capture a live picture of a subject. Here, it is one condition that the person's face should be at the perfect angle; no various poses or expressions are allowed while capturing the image. With increasing of the pose angle, the recognition rate decreases, and it will be challenging to recognize the human face. A high-quality image is necessary for detecting any facial characteristics, which can be used in the further face recognition process.

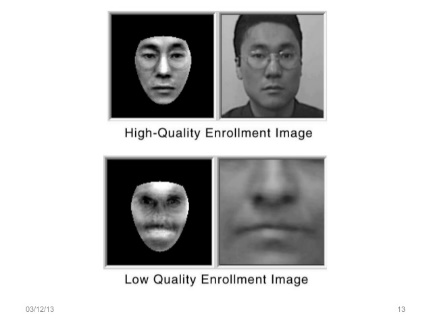
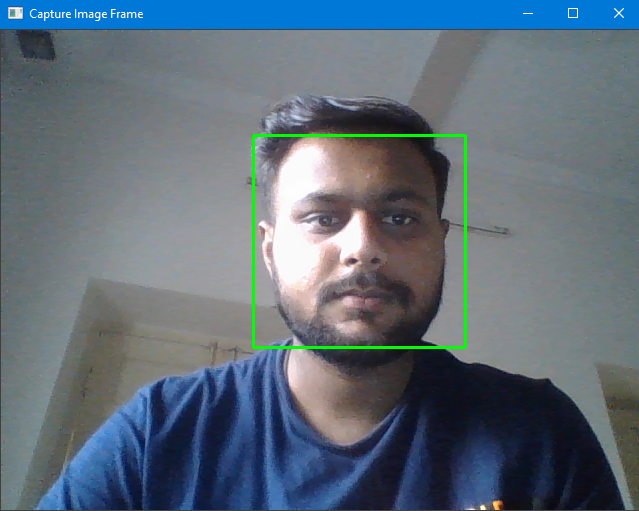


Figure 2. Image Acquisition [[5]](#b5)

1. **Image Processing**

Image processing is the process of performing certain operations on an image to improve its quality or to extract features of that image. This can be used for further analysis and decision-making; when the camera detects



any face and converts it into an image, then that image will be first cropped and converted from RGB to grayscale because it is very easy to detect faces in grayscale images.

1. **Face Detection**

Face detection is a computer technology used to identify the locations and sizes of human faces in arbitrary images. It will only detect facial features and ignores everything else like roads, trees, bungalows, etc. So, face detection can be performed by using the classifier. Classifier means algorithm, which helps us to distinguish whether the given image has a face or not. OpenCV provides some pre-trained classifiers such as Haar Cascade, LBP (Local Binary Pattern), PCA (Principal Component Analysis), etc. Here, we will use Haar cascade and LBP classifier to detect and recognize the human face easily [[6]](#b6).

1. **Feature Extraction**

So, the main goal of this module is to extracting some relevant feature that is well enough to identify as the face. To recognize any human face, software defines nodal points. There are about 80 nodal points on a human face. So, with the help of the nodal points, software identify that the particular image has a human face or not. Here are some nodal points:

* Distance between the eyes
* Width of the nose
* Length of Jaw line
* Chin

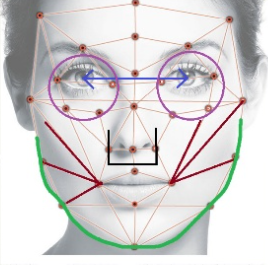


Figure 3. Feature Extraction [[7]](#b7)

1. **Face Matching**

When the facial feature is extracted and landmarks, face position is fed into the software, the software generates a unique feature vector for each face in the numeric form. The feature vector is used to search through the entire database of enrolled users during the face-recognizing process [[8]](#b8). Once the matching face is found, then the particular student attendance is automatically marked in the excel sheet.

In the automatic attendance system, registered student faces are trained and stored in the database. Therefore, when the camera captures the face of any student, it will compare it with the known face database by using the unique student ID, and this process will return matches or potential matches close to the image in the database. Then, the attendance is automatically marked in the excel worksheet.

1. **CLASSIFIER IN FACE DETECTION**

Classifier is a one type of algorithms which is prepared on a huge number of faces and non-faces pictures to figure out how to distinguish human face from an images. OpenCV provides two pre trained classifier which we can use for face identification [[9]](#b9). The two classifier are:

* Haar cascade Classifier
* LBP (Local Binary Pattern) Classifier
* **Haar Cascade Classifier**

Object Detection using Haar feature-based cascade classifiers is an effective method proposed by Paul Viola and Michael Jones in the 2001 paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" [[10]](#b10). The Haar classifier is an AI based methodology, which are trained from a lot of positive images (with faces) and negative images (without faces).

1. **Haar Feature Selection:**

The first step is to collect Haar Features. There are mainly three types of Haar features as Edge Features, Line Features, and Center-surround Features. These features will apply particularly in the rectangle area in an identification window, sum up the pixel intensities in each region, and calculate the difference between these sums [[11]](#b11).

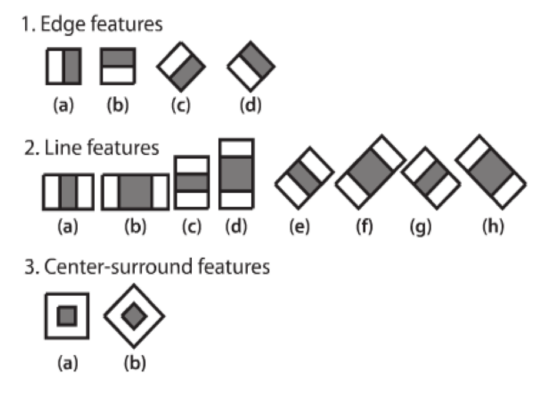


Figure 4. Haar Features [[12]](#b12)

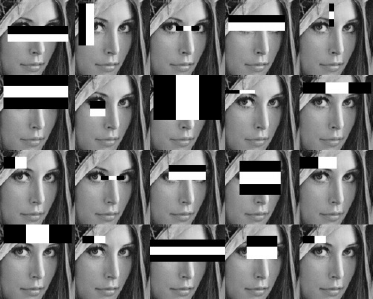


Figure 5. Apply Haar Features on image [[13]](#b13)

1. **Making Integral Images:**

Integral images are used to make this process fast. Most of the calculated features are irrelevant. So, to select the best features out of so many features, we can achieved by Adaboost training.

1. **Adaboost Training:**

Adaboost classifier builds a strong classifier by combining multiple poorly performing classifiers to get high accuracy strong classifier. It combines multiple classifiers to increase the accuracy of classifiers [[14]](#b14). In this classifier, first, apply every feature on all training images. For each feature, it will find the best threshold, which will find the facial features in an image. After each classification, weights of misclassified images are increased, and again the same process is done. In the Adaboost training algorithm, it will first take a 24x24 window from a training image and apply all the features to it. And in an image, most of the image region is non-face, then that particular window is discarded, and don’t process it again. This way, we can find more time to check a possible face region.

* **LBP (Local Binary Pattern) Cascade Classifier**

Local Binary Pattern (LBP) is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number [[15]](#b15). LBP is one of the efficient algorithms for texture classification, and it is further combined with Histogram, which improves the detection performance on datasets. The Local Binary Pattern Histogram (LBPH) algorithm is a simple solution to the face recognition problem, which can recognize both front face and side face.

**Steps of LBPH algorithm:**

1. **Parameters:** The LBPH algorithm uses four parameters [[15]](#b15):

* **Radius:** The radius used for building the Circular Local Binary Pattern. It is usually set to 1.
* **Neighbours:** The number of sample points to build the circular local binary pattern. 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.

1. **Training the algorithm:**

We need to train the algorithm. To do that, we need a facial image dataset and unique ID for each image that we want to recognize, so the algorithm will use that ID to recognize an input image with the images stored in the database and give the output according to the match. One condition here is that images of the same person must have the same ID.

1. **Applying the LBP operation:**

The first computational step of the LBPH algorithm is to generate a new image that describes the original image in a better way by highlighting the facial features or characteristics. For that, the algorithm uses the concept of sliding window protocol. The below image shows this procedure:

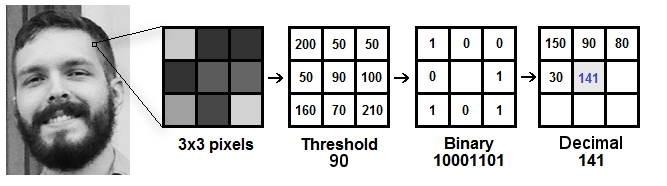


Figure 6. LBPH algorithm [[15]](#b15)

As shown in the image, the algorithm will first select a window of 3x3 pixels. Then, the central value of the matrix to be used as the threshold. By using this value, it will define new values from the 8 neighbors. For each neighbor, we set a new binary value by using the central value. For each neighbours, the condition is: put 1 when central value <= neighbour value and put 0 when central value > neighbour value. Now, the matrix will contain the binary values. We need to concatenate the binary value and create a new binary value. Then, we convert this binary value to a decimal value in a clockwise direction and set it to the central value of the matrix. At the end of this procedure, we have a new image that represents the original image in a better way by highlighting the facial features or characteristics.

1. **Extracting the Histograms:**

Now, using the image generated in the last step, we can use the Grid X and Grid Y parameters to divide the image into multiple grids.

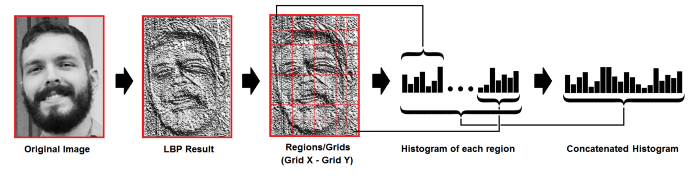


Figure 7. Extracting Histogram [[15]](#b15)

Based on the above image, we can extract the histogram of each region and the final histogram represents the characteristics of the original image.

1. **HAAR CLASSIFIER VS LBP CLASSIFIER**

|  |  |  |
| --- | --- | --- |
| **Algorithm** | **Advantages** | **Disadvantages** |
| **Haar Classifier** | * High detection accuracy * Low false positive rate | * Less accurate on black faces * Longer training time |
| **LBP Classifier** | * Computationally simple and fast * Shorter training time | * Less accurate * High false positive rate |

1. **WHY OpenCV?**

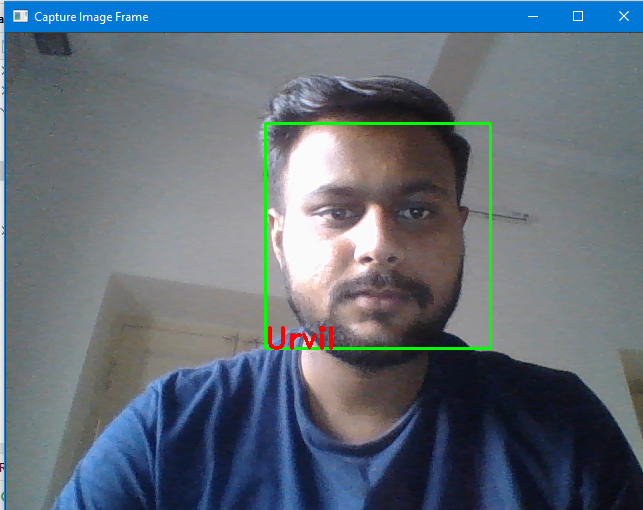
An OpenCV is an open-source and cross-platform library using which we can develop real-time computer vision applications. OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. The library has more than 2500 optimized algorithms. It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV is written in C/C++ [[16]](#b16).

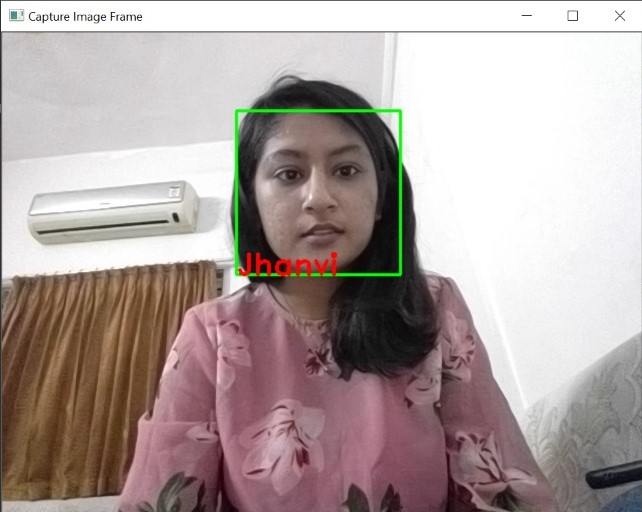
1. **WHY NumPy?**

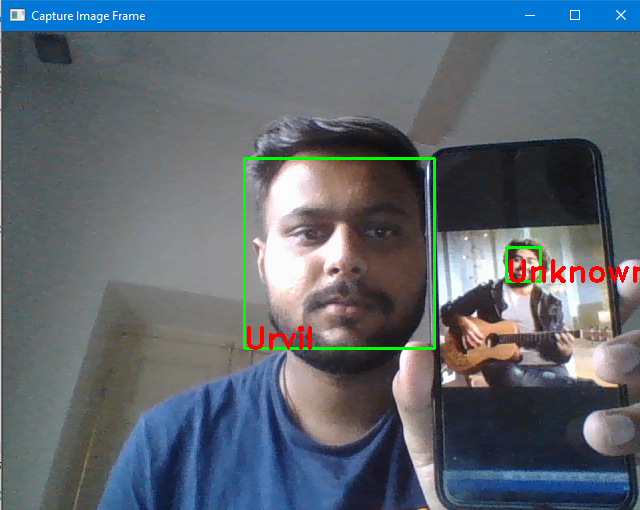
NumPy, which stands for Numerical Python, is the fundamental package for scientific computing in Python [[17]](#b17). NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on the arrays [[18]](#b18).

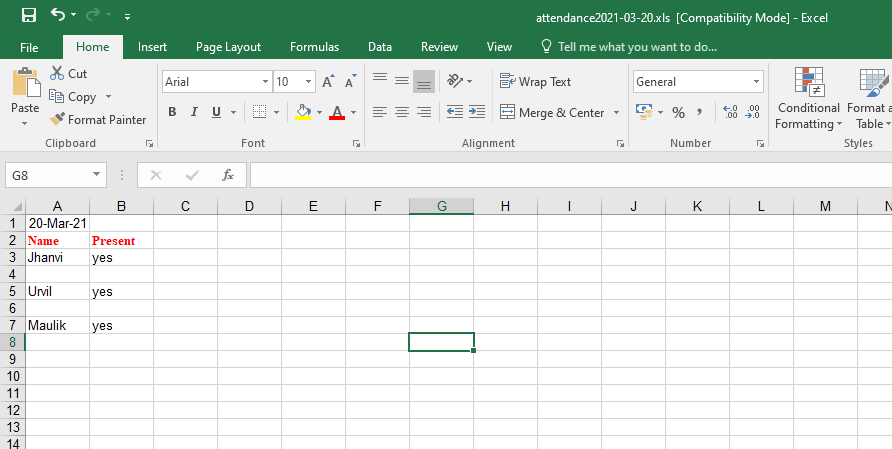
* Using NumPy, we can perform the following operations [[18]](#b18):
* Mathematical and logical calculations on arrays.
* Fourier transforms and routines for shape manipulation.
* It has built-in functions for linear algebra and random number generation.
* It has also used for scientific calculations.
* Faster than python list.
* It is fast because it is associated with C programming.

1. **EXPERIMENTAL RESULTS**

****







1. **ADVANTAGES AND DISADVANTAGES**

The advantages of a face recognition system include public security, fast processing speed, best result, automation of identification, real-time face recognition in schools, colleges, offices, smartphone unlock, and many more in our day-to-day life [[1]](#b1).

Few disadvantages in this system include the funding, high-quality cameras are required, poor image quality may create problems while recognizing any face, perfect size of the image is necessary because it becomes difficult to identify the face in small images. Moreover, various facial expressions, face angles can limit this system [[1]](#b1).

1. **CONCLUSIONS**

In this paper, we have developed a system for face detection and recognition using OpenCV. Face recognition system is currently associated with many top companies and industries making their employees attendance. With the use of python programming and OpenCV, it becomes easy to detect and recognize human faces. This system is used to detect and recognize human faces. The images of a person which we have to recognize will be already stored in databased and trained by using the LBPH algorithm. Moreover, we used the Haar cascade classifier to detect facial features from an image easily. This paper contains a detailed discussion about how LBPH and Haar cascade algorithm works in detecting and recognizing any human face.

1. **REFERENCES**

[[1]](#t1) Tejashree Dhawle, Urvashi Ukey, and Rakshandha Choudante. “Face Detection and Recognition using OpenCV and Python” International Research Journal of Engineering and Technology (IRJET) [Online] Available at: <https://www.irjet.net/archives/V7/i10/IRJET-V7I10219.pdf>

[[2]](#t2) Bhatti KL, Mughal L, Khuhawar FY, Memon SA. Smart Attendance Management System Using Face Recognition. EAI Endorsed Transactions on Creative Technologies. 2018 Oct 29;5(17).

[[3]](#t3) Divyansh Dwivedi, “Face Detection For Beginners” [Online]. Available at: <https://towardsdatascience.com/face-detection-for-beginners-e58e8f21aad9>

[[4]](#t4) FACEFIRST. “What is face detection and how face detection works” [Online]. Available at: <https://www.facefirst.com/blog/face-detection-vs-face-recognition/>

[[5]](#t5) Figure 2. “Image Acquisition” [Online]. Available at: <https://www.slideshare.net/Nikyaa7/automatic-attendance-system-using-facial-recognition>

[[6]](#t6) Ali Tarhini, “Face Recognition: An Introduction”: [https://alitarhini.wordpress.com/2010/12/05/face-recognition-an-introduction/#:~:text=Image%20acquisition%3A&text=The%20most%20existing%20facial%20recognition,angle%2C%20the%20recognition%20rate%20decreases](https://alitarhini.wordpress.com/2010/12/05/face-recognition-an-introduction/)

[[7]](#t7) Figure 3. “Feature Extraction” [Online]. Available at: <https://www.slideshare.net/vishalgarg20/presentation1face-recognition-technology-282152013-49726636>

[[8]](#t8) PATHPARTNER. “What is facial recognition technology- and how it works?” [Online]. Available at: <https://www.pathpartnertech.com/what-is-facial-recognition-technology-and-how-it-works/>

[[9]](#t9) J. Manikandan, S. Lakshmi Prathyusha, P. Sai Kumar, Y. Jaya Chandra and M. Umaditya Hanuman. “Face Detection and Recognition using Open CV Based on Fisher Faces Algorithm”. International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8 Issue-5, January 2020

[[10]](#t10) OpenCV “Face Detection using Haar Cascades” [Online]. Available at: <https://docs.opencv.org/3.4/d2/d99/tutorial_js_face_detection.html>

[[11]](#t11) Mrs. Madhuram.M, B. Prithvi Kumar, Lakshman Sridhar, Nishanth Prem, and Venkatesh Prasad. “Face Detection and Recognition Using OpenCV” International Research Journal of Engineering and Technology (IRJET) Volume: 05 Issue: 10 | Oct 2018 [Online]. Available at: <https://www.irjet.net/archives/V5/i10/IRJET-V5I1089.pdf>

[[12]](#t12) Vincent Tabora. “Face Detection Using OpenCV With Haar Cascade Classifiers” [Online]. Available at: <https://becominghuman.ai/face-detection-using-opencv-with-haar-cascade-classifiers-941dbb25177>

[[13]](#t13) “Training Haar Cascade” [Online]. Available at: <https://memememememememe.me/post/training-haar-cascades/>

[[14]](#t14) Avinash Navlani “AdaBoost Classifier in Python” [Online]. Available at: <https://www.datacamp.com/community/tutorials/adaboost-classifier-python>

[[15]](#t15) Kelvin Salton do Prado. “Face Recognition: Understanding LBPH Algorithm” [Online]. Available at: <https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b>

[[16]](#t16) OpenCV [Online]. Available at: <https://opencv.org/about/>

[[17]](#t17) NumPy [Online]. Available at: <https://en.wikipedia.org/wiki/NumPy>

[[18]](#t18) NumPy [Online]. Available at: <https://numpy.org/devdocs/user/whatisnumpy.html>