A Project Report

on

FACE RECOGNITION ON UNCONDITIONAL IMAGES

Submitted in partial fulfillment of the requirement for the award of the degree of

Bachelor of Technology in Computer Science and Engineering

Under The Supervision of
Mr Gokul Rajan V
Assistant Professor

Department of Computer Science and Engineering

Submitted By

19SCSE1010902 - Priyanka Yadav
19SCSE1140009 - Sahil Sourav Sharma

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GALGOTIAS UNIVERSITY, GREATER NOIDA, INDIA

DECEMBER - 2023

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled **"FACE RECOGNITION ON UNCONDITIONAL IMAGES"** in partial fulfillment of the requirements for the award of the B.TECH submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of month, Year to Month and Year, under the supervision of Gokul Rajan V ,Assistant Professor, Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

Priyanka Yadav,19SCSE1010902

Sahil Sourav Sharma, 19SCSE1140009

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Mr. Gokul Rajan V

Assistance Professor

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of Priyanka Yadav: 19SCSE1010902 has been held on __________________ and his/her work is recommended for the award of B.Tech.

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date: November, 2023

Place: Greater Noida

Acknowledgement

I’d like to thank the following people who have helped us to shoulder this exploration: Our administrators, Mr Gokul Rajan V, for his enthusiasm for the design. for his support, stimulant, and tolerance; School of Computer and Engineering, Galgotias University, for input throughout this Design.

Abstract

Humans can quickly identify faces without much effort. It's an easy task for us, but it's a difficult task for a computer. There are various complexities such as low resolution, occlusion, lighting variations, etc. These factors greatly affect the accuracy of the computer to recognize the face more effectively. First, it is necessary to understand the difference between face detection and face recognition. Face detection and image or video recognition is a popular research subject in biometrics. Real-time face recognition is an exciting field and a rapidly growing challenge. A framework for using facial recognition application authentication.

The main goal of facial recognition systems is to ensure that everyone has a unique face. We know that every person has a unique fingerprint, so every face has unique features. Here we use human facial features. It can store the facial features of many people and identify them based on their facial features. Face recognition and facial recognition is a difficult and challenging task. For a facial recognition system to be authentic, it must be accurate and precise. The facial recognition technology took pictures using cameras and compared them with pictures stored in a database. If the captured image matches one of the saved images, matching faces will be displayed. If no match is found, an unassigned face will be displayed.

Contents

List of Tables

List of Figures

Acronyms

CHAPTER-1

INTRODUCTION

Computer vision is one of the most fascinating and challenging tasks in artificial intelligence. Computer vision serves as the link between computer software and the visual elements we see around us. It allows computer software to understand and learn about the visuals in its environment. The main goal is to identify and understand images and offer new images that are more useful to us in various areas of life [1], [2]. The term ``OpenCV" stands for ``open source computer vision." The architecture consists of software, databases, and plugins that are pre-programmed to support the integration of computer vision applications [3]. the most used toolsets with a large group of developers. It is known for its size in which it creates real use cases for industrial use. OpenCV follows C/C++, Python, Java programming languages and can be used to create computer vision software for desktop and smartphone platforms such as Windows, Linux, macOS, Android and iOS. The latest versions are OpenCV-4.5.2 and OpenCV-3.4.14. It is free and open source and easy to use and install. It is designed for numerical productivity with a strong emphasis on real-time applications. The first version was in the C programming language, but its success increased with the release of version 2.0, which had a C++ implementation [2]. used to create new functions. OpenCV can be downloaded for free from http://opencv.org. This platform includes the latest distribution update (version: 4.5.2) as well as older iterations. Photos must be in BGR or grayscale format to be displayed or saved via OpenCV. Otherwise, adverse results could occur [1]. Face detection is a form of computer vision that helps detect and visualize facial features in captured images or videos in real time. This type of object detection technique detects occurrences of semantic artifacts of a given class (such as people, cars, and houses) in digital images and videos. Facial recognition has become increasingly important as technology has advanced, especially in fields such as photography, defense, and marketing [4], [5].

Recognition is a modern field of study that has attracted the interest of researchers because it is easy to use using OpenCV-based Python. Facial recognition technologies have various applications in public safety, entertainment, human-machine contact, and social networking, such as automatically recommending brands in Facebook images. It has also been seen in attendance control of educational and non-educational institutions, tax offices, voter registration and other fields [6], [7] In this post, we highlight the important role of OpenCV in face detection and face recognition, what Algorithm can be used in OpenCV for face detection and face recognition, then list OpenCV modules and explain Python-based OpenCV and mention applications for OpenCV.

1.2 Formulation of problem

1.2.1 Background of the study

Face detection is a subfield of computer vision that involves the identification and localization of human faces in digital images or videos. The goal of face detection is to create algorithms that can accurately detect and recognize faces in a variety of environments, including images with complex backgrounds, changing lighting conditions, and different facial expressions.

The development of face detection technology has been driven by a variety of applications, including security and surveillance systems, facial recognition software, and digital photography. For example, facial detection is used in security systems to identify people entering restricted areas, in facial recognition software to authenticate users or compare faces in a database, and in digital photography to automatically focus on faces and adjust exposure settings.

Early approaches to face detection relied on simple image processing techniques such as edge detection and template matching. However, these techniques were not robust to changes in lighting, pose, and facial expression. In the 1990s, researchers began developing more sophisticated algorithms based on machine learning techniques such as neural networks and support vector machines.

One of the most popular face detection algorithms is the Viola-Jones algorithm, also known as the Haar Cascade classifier. This algorithm was introduced in 2001 by Paul Viola and Michael Jones and is based on the AdaBoost machine learning algorithm. The Haar Cascade classifier uses a set of Haar-like features to detect faces in images and is known for its high accuracy and speed.

In recent years, deep learning techniques such as convolutional neural networks (CNNs) have shown promising results in face detection and recognition. These techniques have enabled the development of more sophisticated face detection systems that can handle complex scenarios such as occlusions and different poses.

Overall, face detection technology has advanced significantly over the past few decades and has become an important tool in various applications. The continued development of this technology is expected to have a significant impact on areas such as security, healthcare and entertainment.

1.2.2. Objective of the study

The aim of the study on unconditional image face detection is to explore the capabilities of OpenCV in detecting human faces in images and videos with high accuracy and speed. The objective is to investigate the effectiveness of the Haar Cascade classifier in detecting human faces in digital images or videos. The Haar Cascade classifier is a machine learning approach that uses a set of Haar-like features to detect faces in an image [1]. The aim of the study is to evaluate the performance of the Haar Cascade algorithm in terms of accuracy and speed and compare it with other face detection algorithms. In addition, the study can explore potential applications of Haar Cascade face detection in various fields such as security, surveillance, and digital photography [2]. OpenCV provides a wide range of tools and techniques for face detection, including Haar cascades, local binary patterns, and deep learning approaches such as convolutional neural networks (CNNs) [1]. The aim of the study is to evaluate the performance of these techniques and compare their accuracy and speed in face detection in different scenarios. In addition, the study can explore potential applications of face detection using OpenCV in various fields such as security, surveillance, and digital photography.

1.2.3 Tools & Technology used

1) Any operating system that will support OpenCV and Python
2) Python
3) OpenCV-Python
4) Haar Cascades Data File
5) i3 or higher core processor (CPU)/ 2.1 GHz or higher
6) Photo/images for testing

Here we used an MSI laptop (15-bs1xx) with 10th generation CORE i5 Intel processor 1.8 GHz with 8 GB memory and Windows 11 Home. We used Pycharm IDE with Python 3.9.1 and

OpenCV 4.5.1.48 installed on my system. Also the VS Code was used to run the python file with ease.

CHAPTER-2

LITERATURE SURVEY

2. Assessment of Literature Reviews

OpenCV is an image and video processing library used for image and video analysis, among other things. We review the recent literature on face detection and recognition using OpenCV.

2.1 Face Detection

Alcantara etal.(22) proposed a system for real- time mortal head shadowing and discovery in videotape using the OpenCV library. The proposed system will use a Haar- suchlike classifier to descry the head, Haar Training to train the system, and use the CMT object tracking algorithm to track the head, where the result shows 68 delicacy of CMT for tracking a person's head and 83 delicacy for detecting a person's head.

Gupta(7) proposes a real- time and still image emotion discovery system. So, before trying to descry feelings, it must first identify faces in still prints or vids in real time using OpenCV's Haar classifier. Once the face has been linked, it can be cropped and anatomized to identify other facial milestones. The datasets are also edited using facial milestones and distributed into eight feelings using SVM, a machine learning algorithm. They attained an delicacy of about93.7 percent using SVM.

These facial milestones can be acclimated to increase delicacy. exploration by Lee etal.(4) aims to break the classic problem of face discovery in different lighting conditions and produce an intelligent and effective mortal face discovery system using Visual Studio 2015 software frame and OpenCV technology. Through trial, they demonstrated that the print processing approach used in their paper completed face recognition under colorful lighting conditions, a truly significant advance in face recognition technology.

Gupta etal.(16) proposed a system to ameliorate the traditional attendance systems of universities and avoid wasting time in traditional attendance counting grounded on image processing styles. The primary function of the pupil attendance layout structure is to take, incorporate and manage attendance notes for the pupil, to automatically estimate the number of presents and absent depending on the content and relish of the class, and also produce an automated document or spreadsheet..

They used OpenCV library, Haar- Cascade for face discovery and LBPH for face recognition; also individual pupil training took place and eventually the device created a spreadsheet that handed the number of scholars present in the classroom with a live image or videotape.

Das etal.(19) offers a simpler approach to face mask discovery using several introductory machine literacy packages similar as Scikit, OpenCV, TensorFlow, and Keras. The proposed procedure directly scans the face in the image and also determines whether it's wearing a mask or not. It can track the face and mask in stir as a shadowing charge artist. On two separate data sets, the system achieves an delicacy of over to95.77 percent and94.58 percent, independently. They probe optimized parameter values for the periodical CNN model to rightly descry the actuality of masks without driving overfitting.

Hoque etal.(5) designed software that can identify mortal faces or faces from live videotape streaming. The heart of the control scheme is an ATMega328p Arduino Uno microcontroller withPan-Tilt and OpenCV. Hair – Cascade, Camshift, Housedorff gap, AdaBoost, Viola Jones,etc. are used to identify mortal faces. They used the Haar Classifier waterfall algorithm to classify faces.

Mehariya etal.(20) proposed a system to overcome the problem of scholars not attending lectures in university classrooms or any place taking staff attendance. Since absenteeism is a waste of structure, homemade absence operation is a tedious task and a waste of time as the traditional absence counting system can be cheated. They used OpenCV to find the number of scholars in a class and created an algorithm to give the stylish residency rate.

The proposed system would be different from the conventional approach. The Multiscales discovery procedure is used to count the number of scholars in a class and the residency rate is used to assign a complex classroom.

The first system of classifying an individual in an image is the Multiscale discovery system. It doesn't consider particulars. This count is stored in the cast fire database and the residency rate is used for comprehensive class distribution. This reduces the destruction of space and allows us to prepare the schedule efficiently.

Factors similar as class scale, classroom projectors and dynamic allocation capability are taken into account. Google Cloud Database(Firebase) is used to maintain and store council/ university lines. It includes particular records, registries, schedules and attendance records.

Sriratana etal.(11) created a particular identifier scheme by combining the Viola and Jones algorithms with the OpenCV library and Python enciphered on a jeer Pi regulator board. The frame successfully demonstrates all intended endpoints, including cost-effective deployment, simple installation, and real- time discovery.

During tests of 150 samples, only 8- 9 percent of crimes were discovered, which proves the high delicacy and performance of the system.

Patel etal.(23) proposed a system that senses whether the motorist is asleep or not while driving a auto or other large vehicle, and if the motorist is suspected to be sleepy, the system cautions the motorist to wake up and stop driving. One of the reasons for collisions on public roads is motorist doziness while driving. It's necessary to develop an effective approach to prognosticate doziness once the motorist becomes drowsy.

This can help reduce the large number of injuries that do. This system would help reduce the number of sleepy motorist injuries. In the OpenCV setting, he used real- time image processing using a vision device and the fashion of facial expressions and eye blinks.

2.2 Face Recognition

Bojko and co. [21] compared the effectiveness of two major libraries in computer vision (Dlib and OpenCV) and proposed two basic face recognition systems for these two libraries. Based on its results, it demonstrates that the OpenCV library is more efficient and has better face detection and recognition results than the Dlib library. It also implies that OpenCV is most suitable for developing recognition software for the IoT framework.

Sarkar and Sikka [9] investigate and evaluate various classifiers used in face embedding classification. They also focus on a Python-based face recognition pipeline that can be used to create a face recognition framework on compact, low-power hardware devices. The mentioned technique uses appropriate models and structures, resulting in superior efficiency without the need for powerful hardware. On the LFW dataset, the proposed approach achieves an accuracy value of 99.4.

Sharma [18] proposed a facial recognition system for specialized applications, including entry and security, shopping, and criminal identity. The identification method will focus on face recognition, which will be face detection, feature extraction and classification, and real-time face recognition. They used Haar-like for face detection and LBPH for face recognition, all done using OpenCV in Python environment. Kivy is used to create user interfaces to make the proposed system executable through multiple platforms.

James and Nettikadan [12] presented a real-time monitoring system inside school buses. By using image processing, using the image to identify the student with the camera. This machine watches videos inside the bus and recognizes students and their gestures. The proposed system recognizes the faces of the students and maintains the count of their numbers. If absolutely necessary, the system can even sound a warning to attract public attention using technologies from OpenCV and their implementations in Python. Face detection was done using HaarCascades classifier and face recognition was done using Eigenfaces and Lbph. This eliminates most of the disadvantages of manual attendance systems, such as simple handling of attendance records.

Balachandran et al. [13] proposed an effective AI face recognition application. The VGGFace framework was used for the neural network. The application is divided into two phases: training and recognition. The training phase involves adding new faces to the schema, while the identification phase involves determining the identity of the face. The program will run on multiple cores without any problems.

Apoorva. et al. [15] proposes a technique for robust real-time face recognition. Haar-cascade is one of the face recognition algorithms. They tracked faces using Haar-like classifiers on the OpenCV website. Face detection has a good level of accuracy. Since the computation time is short, the proposed method successfully identifies more than one face, which is useful for quickly searching for suspects.

Srivastava et al. [17] created an attendance system by combining facial recognition technologies with an algorithm (OpenCV). By simply keeping records of arrival and departure times, this program can simplify the process of automating attendance and help faculties gain access to student details.

Soomro et al. [6] created a standalone authentication program using face recognition method with (NI VISION, LabVIEW, NI MyRIO, OpenCV). The developed framework should identify and allow authenticated individuals to remember internal system weaknesses in a real-time context, especially those triggered by a pause. The entire framework consists of hardware and software, including parallel processing methods and modules such as the NI MyRIO FPGA.

Sharma et al. [24] propose a recognition system that can benefit a blind individual. This paper used a hand gesture identification method and a face recognition system to perform many tasks. Dynamic images are from dynamic video and are interpreted by algorithms. In the Hand Motion scheme, the skin color recognition was performed in YCbCr colors, and the convex point of the defect feature based on the hand was used to identify various characteristics such as fingertips and angles between fingers. Depending on the recognized gesture, various actions can be performed, such as turning on the fan or lighting. OpenCV, Haar-Cascade and LBPH are used for mask detection and face identification.

Salihbasic and Orehovacki [14] outline and describe the entire process of creating an Android application that recognizes a person's gender, age, and face in depth. Face detection and recognition techniques and development software used in Android mobile application development are defined and discussed. The software solution explains how to use the OpenCV library and shows the actual results of the smartphone application using photos.

Zhu and Cheng [25] present. OpenCV-based Efficient Location Tracking Algorithm (EATA) is used for face recognition in smart door lock system. A specialized application has been created that enables device operation and location display. Automated door monitoring system using raspberry pi python, USB camera and OpenCV is provided for monitoring and security purposes. The system state database was created at some point in time. The gadget is very cheap, easy to set up and easy to use. The attitude tracking algorithm is extremely accurate and works well.

Table 1 - Comparison table

Ref	Aim	classifier	accuracy	Result/purpose
[21] 2018	SVM	OpenCV more accurate than dlib	83% head detect	The OpenCV library is more productive, and has improved facial recognition and detection accuracy.
[22] 2018	Head Detection and Tracking	Haar-like CMT Cascade	68% tracking	The proposed system successfully detected the head of a human using OpenCV libraries, specifically using Haar-like attribute detection.
[7] 2018	real-time recognition Facial emotion	Haar SVM	93.7%	The findings suggest that with today's computing power, user-independent, completely automated real-time coding of facial expressions in a continuous video stream is a goal that can be achieved.
[4] 2018	Face Detection under Different Lighting	Haar	80%	The experiment demonstrated that the picture processing system has facial recognition in various lighting conditions.
[18] 2019	Designing of Face Recognition System	Haar-like LBPH	80%	The system is tested by more than 150 people and has a reliability of approximately 80%. It is measured with multiple cameras in various settings, and lighting conditions, and the findings are about the

				same. A Logitech C90 USB webcam is used here.
[16] 2020	Automated Attendance System	KNN HaarCascade LBPH	97%	Prevent students from marking fake attendance for other students. It would also save faculty resources in universities by eliminating the need for them to take attendance of students who are present in class, and it will be able to send monthly attendance reports to students' parents by email.
[5] 2020	Autonomous Face Detection System from Real-time Video Streaming	Haar Cascade	83%	Recognize human faces with some kind of camera and issue an alarm with a buzzer and an automatic-on light bulb that makes it noticeable from a long distance.
[24] 2019	A Face Recognition and Static Hand Gesture System for the Blind	Haar Cascade LBPH	hand gesture recognized is 95.2% Facial recognition is 92%.	The developed system will function as a virtual assistant for a blind individual using hand gestures and face recognition.
[11] 2018	Personal Identifier application	Cascade Haar	90% Just 8-9 percent of errors were discovered after analyzing 150 samples.	The proposed application helps companies calculate work attendance and detect cases of fraud in work attendance compared to old work attendance monitoring methods.

Based on Table 1, we compared the accuracies of OpenCV techniques and classifiers used in a number of face detection and face recognition studies investigated in different areas of computer vision. It has been found that OpenCV can be used in various fields such as face detection, face recognition, and facial expression recognition [26], [27]. It has also been found that we can use OpenCV in the field of security, such as criminal identification [28]. Additionally, others have used OpenCV to automate the attendance of students or staff in institutions. Meanwhile, other researchers have used OpenCV to reduce traffic accidents on the roads.

Unconditional image face recognition refers to the process of detecting and identifying faces in images that are not constrained by specific environmental or contextual factors. A review of the literature on this topic reveals a significant amount of research and development in the field of computer vision, with many methods and techniques being proposed and evaluated for face recognition in unconditional images.

One of the first approaches to face recognition in unconditional images was the use of eigenfaces, introduced by Turk and Pentland in 1991. This method involves representing a face as a linear combination of principal components, or "eigenfaces", which are derived from a training set of faces. The method has since been refined and extended in various ways, including the use of nonlinear manifolds and deep neural networks.

Another approach to face recognition in unconditioned images is to use feature-based methods, which involve extracting relevant features from the image and using them to classify the face. Examples of features include texture, shape, and color. These methods have proven effective in some cases, but can suffer from feature variability, which occurs when the same face appears differently due to changes in flash, position, or other factor.

In recent years, deep learning techniques, especially convolutional neural networks (CNNs), have emerged as a state-of-the-art approach to face recognition in unconditional images. CNNs are able to automatically learn hierarchical features from raw image data and can be trained using large datasets. Many variants of CNNs have been proposed for face recognition, including conjoint networks, triple networks, and attention-based models.

Despite significant progress in face recognition from unconditional images, there are still many challenges that remain, including handling occlusions, variations in position and illumination, and recognizing faces of different ages and ethnicities. In addition, there are concerns about the potential misuse of facial recognition technology and its impact on privacy and civil liberties.

Overall, face recognition in unconditional images is a highly active research area with many promising approaches and open research questions.

2.1 History and overview of face detection

Face detection has received much attention in recent years due to its applications in computer and human interaction. Face detection is a subset of image processing. Image processing is primarily a technique for compressing, enhancing or extracting valuable information from images. Face recognition technology can identify one or more faces in an image and remove unwanted background noise. Basically, a face recognition algorithm has to categorize images into two groups based on whether they contain a face or not. The goal of the face detection algorithm is to thoroughly examine the image, identify the existence of faces in the image, and remove the background from the image. Face detection errors are divided into two types: false negative and false positive. A false positive occurs when a face is identified in an image that does not contain any faces. A false negative occurs when the algorithm rejects the existence of anything in the image. The detection rate is the ratio of the number of faces identified by humans to the number of faces correctly detected by the system. The detection rate of the face detection algorithm should be as high as possible.

2.2 Face Detection

The scope of the Haar Cascade face detection study may include evaluating the performance of the Haar Cascade algorithm in detecting human faces in digital images or videos. The study may involve testing the algorithm on different datasets with different image qualities, lighting conditions and facial expressions. The study can also explore the potential applications of Haar Cascade face detection in various fields such as security, surveillance and digital photography.

However, there are some limitations to the study of face detection using Haar Cascade. One limitation is the sensitivity of the algorithm to changes in the training data. The accuracy of the Haar Cascade algorithm depends on the quality and quantity of training data used to train the classifier. Another limitation is the algorithm's performance in detecting faces under occlusions such as glasses, hats, or other objects that cover parts of the face. The Haar cascade algorithm may not be able to accurately detect faces in such scenarios.

In addition, the study may be limited by the computational resources available for testing the algorithm. The Haar Cascade algorithm can be computationally intensive, especially when working with large datasets or high-resolution images. Therefore, the study may be limited by the computing power and memory of the computer used to test the algorithm.

Overall, the scope and limitations of the study on face detection using Haar Cascade depends on the research questions, objectives and available resources. These factors are important to consider when designing and conducting a Haar Cascade face detection study.

2.3. Face Recognition

Facial recognition is the most advanced and fastest biometric technology in the world. It uses the most visible part of the human body, the face, in a non-obtrusive way. According to global data, most individuals are unaware of the facial recognition process taking place on them, making it one of the least invasive procedures with the least delay. A face recognition algorithm examines many facial features in an input image. This biometric has been widely, and perhaps excessively, praised as a great method for identifying potential dangers such as terrorists, fraudsters, and so on, but it has yet to gain widespread, high-level adoption. Face biometric technology is expected to overtake fingerprint biometrics as the most common method of user identification and authentication in the near future [8], [9], [10].

2.4. OpenCV Library

It is a massive open-source library for image processing, machine learning and computer vision. OpenCV is compatible with a wide range of programming languages, including Python, C++ and Java. It will analyze photos and videos to recognize artifacts, faces and even human handwriting.

When paired with many other libraries, such as Numpy, a high-performance lathe library, you get good performance; this means that all services that can be performed in Numpy can also be integrated with OpenCV. It is written in C++ and has a C++ interface as the main interface, but also has less robust but still detailed legacy language training. Both the latest technologies and algorithms are visible in the C++ GUI. Python, Java and MATLAB/OCTAVE [3] bindings are available. Wrappers have been created in various programming languages to encourage wider adoption. The JavaScript plugins for the OpenCV feature variant are published as OpenCV.js in version 3.4, which can be used on web platforms. Officially introduced in 1999, the OpenCV project was originally a research program by Intel to support CPU-intensive applications [11]. OpenCV is a popular platform for implementing face detection and recognition algorithms. Below are some frequently used OpenCV algorithms.

2.5 Comparison of Haar Cascade with other face detection algorithm

Haar Cascade is a popular and widely used face detection technique, but it is not a complete face recognition system. Face recognition involves not only detecting the presence of a face, but also identifying an individual in a facial image.

Here are some comparisons of Haar Cascade with other facial recognition techniques:

Cascade Haar vs. Viola-Jones: The Viola-Jones algorithm is a similar face detection algorithm that also uses Haar-like features and machine learning. However, it is less accurate than the Haar cascade, especially when detecting faces at different angles and scales.

Approaches based on Haar cascade vs. Deep Learning: Face recognition systems based on Deep Learning, such as Convolutional Neural Networks (CNNs), can achieve higher accuracy than

Haar cascade in face detection and individual recognition. However, they require much larger training datasets and computational resources.

Haar Cascade vs. Eigenfaces: Eigenfaces is a popular face recognition technique that uses Principal Component Analysis (PCA) to extract features from face images. It is more accurate than Haar Cascade in face recognition, but requires more computing resources and is less efficient.

Haar cascade vs. local binary patterns (LBP): LBP is another feature extraction technique used in face recognition. It is faster than the Haar cascade, but less accurate in detecting faces, especially in the presence of noise or occlusion.

In conclusion, Haar Cascade is a fast and efficient face detection algorithm that works well in many scenarios. However, for more accurate and robust face recognition, deep learning-based approaches such as CNNs are preferable, although they require more computing resources and larger training datasets.

CHAPTER-3

METHODOLOGY/FUNCTIONALITY

1. Face Detection:

The first task that we perform is detecting faces in the image(photograph) or video stream. Now we know the exact coordinates/location of the face, so we extract this face for further processing.

2. Feature Extraction:

Now we can see that we have cropped the face from the image, so we are extracting specific features from it. Here we look at how to use face embedding to extract these facial features. As we know, a neural network takes an image of a person's face as input and outputs a vector that represents the most important features of the face! In machine learning, this vector is called nothing but embedding, so we call this vector face embedding. Now how will this help in recognizing different people's faces?

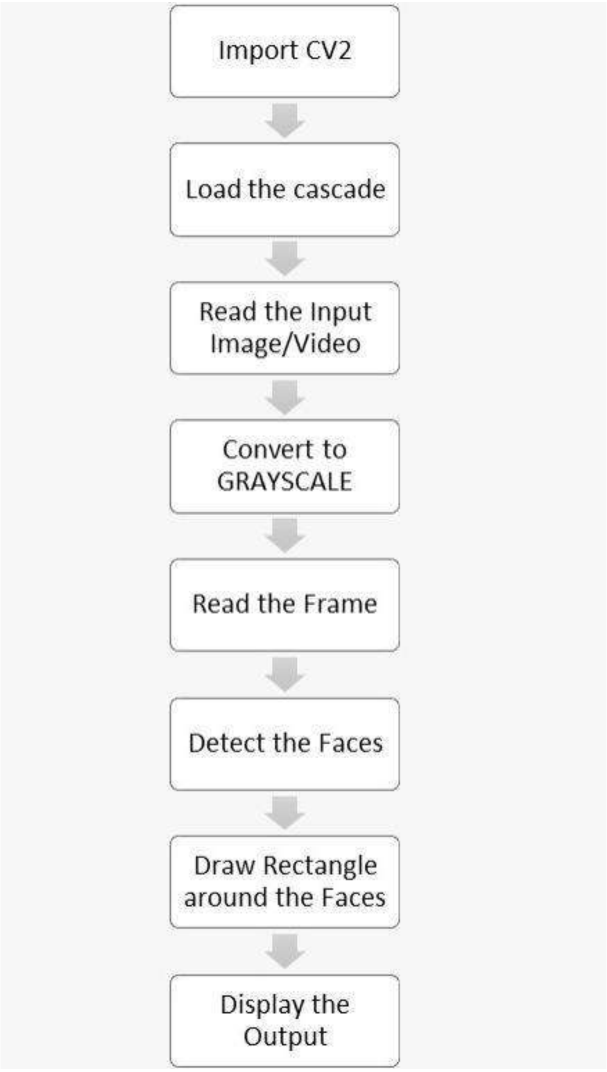
When we train a neural network, the network learns to output similar vectors for faces that look similar. Consider an example, if I have multiple images of faces at different time-lapses, it is obvious that some features may change, but not too much. So in this problem the vectors associated with the surfaces are similar or we can say they are very close in the vector space.

Up to this point we have seen how this network works, let's see how to use this network on our own data. Here we feed all the images in our data to this pre-trained network to get the relevant embeddings and save those embeddings to a file for the next step.

3. Comparing Faces:

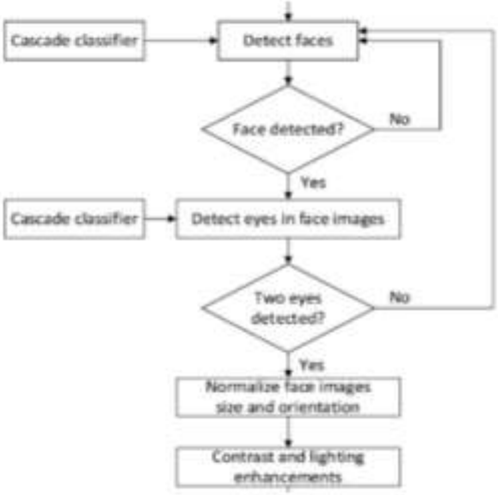
We have a face insert for each face in our data stored in a file, the next step is to recognize a new image that is not in our data. So the first step is to calculate the face embedding for the image using the same mesh we used earlier and then compare that embedding to the rest of the embeddings we have. We recognize a face if the generated embedding is closer or similar to any other embedding.

IMPLEMENTATION FLOW



3.1 Haar Cascade

Haar Cascade is an efficient method for object detection. It is a machine learning based method in which a cascade of actions is learned from a large number of positive and negative images. He gets used to seeing things in different frames [12]. Giant. 1 shows a view of the Haar cascade classifier.


Cascade classifier
Detect faces
Face detected?
No
Yes
Cascade classifier
Detect eyes in face images
Two eyes detected?
No
Yes
Normalize face images size and orientation
Contrast and lighting enhancements

Performance Evaluation of Haar cascade Algorithm:

Performance evaluation is an essential step in evaluating the effectiveness of any face detection algorithm, including Haar Cascade. There are several metrics that can be used to evaluate the performance of Haar Cascade, including accuracy, precision, recall, and F1 score.

Accuracy: Accuracy measures the proportion of correctly identified faces among all faces in an image or video. It is calculated as the number of true positives plus true negatives divided by the total number of faces.

Accuracy: Accuracy measures the proportion of true positives among all detected faces. It is calculated as the number of true positives divided by the total number of faces detected.

Recall: Recall measures the proportion of true positives among all real faces. It is calculated as the number of true positives divided by the total number of faces.

F1 Score: The F1 score is the harmonic mean of precision and recall. It is a measure of the overall performance of an algorithm and is calculated as twice the product of precision and recall divided by their sum.

A dataset of images or videos with ground truth labels is required to evaluate the performance of Haar Cascade. The dataset should contain a number of images with different lighting conditions, orientations and scales to test the robustness of the algorithm.

The performance of the Haar Cascade algorithm can be evaluated by comparing the detected faces with the ground truth. Accuracy, precision, recall and F1 scores can be calculated and used to compare the performance of Haar Cascade with other face detection algorithms.

In general, Haar Cascade has been shown to be an accurate and efficient face detection algorithm, especially for real-time applications. However, its performance can be affected by various factors such as the quality of training data and algorithm parameters. Therefore, it is important to evaluate the performance of the algorithm in different scenarios to ensure its reliability.

Summary of the study of face detection using haar cascade

Haar Cascade face detection is a popular computer vision technique that allows computers to identify and locate human faces within images or video frames. The Haar cascade algorithm is based on Haar-like features that are used to detect various patterns in an image, including the presence of a face. During the detection process, the algorithm uses a sliding window to scan the entire image and pre-trained classifiers evaluate the Haar-like features. If the features match the face pattern, the algorithm marks the area as a potential face. In this way, multiple windows are scanned and the algorithm outputs the areas that are most likely to contain a face.

Performance evaluation is an essential step in evaluating the effectiveness of any face detection algorithm, including Haar Cascade. The performance of the Haar Cascade algorithm can be evaluated by comparing the detected faces with the ground truth. Accuracy, precision, recall and F1 scores can be calculated and used to compare the performance of Haar Cascade with other face detection algorithms.

In general, Haar Cascade has been shown to be an accurate and efficient face detection algorithm, especially for real-time applications. However, its performance can be affected by various factors such as the quality of training data and algorithm parameters. Therefore, it is important to evaluate the performance of the algorithm in different scenarios to ensure its reliability.

3.2 LBP (Local Binary Pattern)

It is a simple but powerful texture operator that labels pixels in an image by thresholding pixel neighborhoods and processes the result as a binary number. The LBP texture operator has become a common approach in various applications due to its discriminating power and computational simplicity. It can be seen as a unifying solution to the historically divergent statistical and structural models of texture analysis. The robustness of the LBP operator to monotonous changes in grayscale caused by, for example, changes in illumination is probably its most important property in real-world applications. Another key feature is its computational simplicity, which allows it to 6 analyze images in difficult scenarios in real time [13].

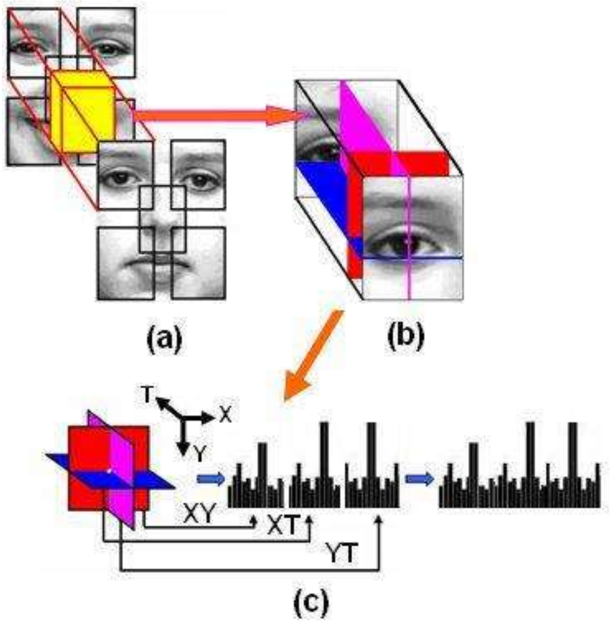


Fig. 3 - Description of facial expressions with local binary patterns [13]

3.3 EigenFaces

It is a tactic that uses PCA (Principal Component Analysis) to minimize dimensions and find the strongest vectors to distribute face images into existing face spaces. The primary goal of PCA is to identify the best vectors to explain the distribution of face images in image space to face space. According to the distribution of eigenvalues, the m eigenvector is used to construct the quantity of the principal component. The eigenvector and eigenvalue are calculated from the covariance matrix of the qualified face image. The eigenvector is sorted by eigenvalue (highest to lowest) and the M first eigenvectors are chosen to form the main variable [1], [14].

3.4 FisherFaces

It is a facial recognition system that has been shown by several researchers to be able to accurately identify faces. FisherFace is a computational model that combines the PCA (Principal Component Analysis) computational model with Fisher's Linear Discriminant (FLD). PCA is used to minimize the input data to simplify and speed up FLD operation. On the other hand, FLD is used to generate a distribution matrix that helps in classification and identification. Using PCA and FLD computational models, a series of FisherFaces is generated. This face recognition process consists of four main steps: face identification, PCA estimation, computation and classification [15]

3.5 LBPH (Local Binary Pattern Histogram)

LBP is a highly efficient texture operator. This compares the threshold value of each neighboring pixel to the value of the middle pixel. It considers results in the context of binary numbers. LBP is a common technique in various applications due to its discriminating power and simplicity. LBP was first identified in 1994. Since then, it seems to have evolved into a more efficient texture classification algorithm. It was later found that combining LBP with histograms of directed gradient descriptors increases its accuracy on the same data set. LBP has additional features such as monotonic grayscale enhancements and statistical simplicity, which enables it to interpret images in real-time applications [1]. Giant. 4 shows the LBPH algorithm for face recognition.

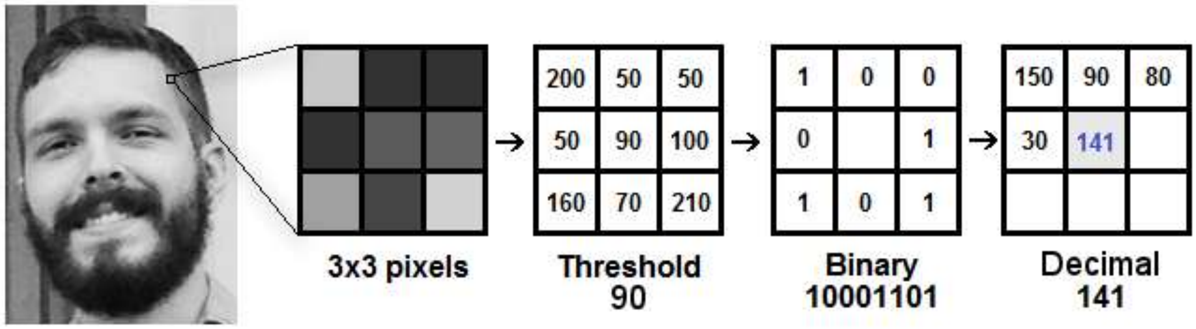


Fig. 4 - LBPH Algorithm for Face Recognition [1]

3.6 YOLO

Yolo is short for (you only look once). It is the latest real-time object detection system that uses a single neural network to process the entire image [16]. This network divides the image into parts and estimates bounding boxes and probabilities for each part [17]. Estimated probabilities are used to weight these bounding boxes. The testing phase examines the entire image, so the global meaning of the image drives its predictions. It can detect objects from videos or images [18] [19]. Giant. 5 shows the Yolo process.

3.7 Faster R-CNN

It is a popular object detection architecture proposed in 2015 by Ross_Girshick and is one of the most famous object detection architectures that uses convolutional neural networks. Implementing a region design network makes (Faster R-CNN) easier and faster (RPN). The RPN is a fully convolutional network that has been trained side-by-side to predict object boundaries and object ratings at each detection. Since RPN is so fundamental to (Faster-R-CNN) and remains one of the most powerful entity detection frameworks open to researchers, most of this piece would focus on RPN architecture and the concepts of anchor boxes and non-maximal suppression [18]. . Giant. 6 shows the Faster_R-CNN step.

4. RESEARCH AND ANALYSIS

5. Modules of OpenCV

5.1 Main Modules

Core modules are the main components of OpenCV and are necessarily part of packages. They are called core modules because they contain basic functions, including image recognition, filtering, and transformation [8], [21].

5.2 Extra Modules

Extra modules are not part of the OpenCV version by default. Additional machine vision functions such as text detection [21] are applied to these modules.

The main modules are described below:

● Core: Has most of the core features of OpenCV.

● Imgproc: Image processing tools such as transformation, manipulation and filtering are used.

● Imgcodecs: Features for reading/writing images are included.

● Videoio: Video read/write functions are included.

● Highgui: Used to create a GUI to display the output.

● Video: Motion detection and monitoring.

● calib3d: This package contains calibration and 3D reconstruction functions for translation estimation between multiple images.

● features2d: Item identification and categorization algorithms using keypoint detection and descriptor extraction algorithms included in this library.

● Objdetect: used for object detection.

● Dnn: Classification and detection of objects among other objects.

● Ml: Used for regression and classification and encompasses the vast majority of machine learning.

● Flann: Supports optimized algorithms for finding high-dimensional attributes in large datasets using nearest neighbor search. Fast Library for Approximate Nearest Neighbors (FLANN) is short for Quick Library for Estimated Nearest Neighbors (FLANN)[22].

● Photography: Removes noise and creates HD images, for computer vision related to photography, among others.

● Stitching: Used for stitching the image.

● Shape: Solves problems like shape transformation, matching and distance.

● Superres: Deal with enhancement and resolution algorithms.

● Videostab: They have video stabilization algorithms.

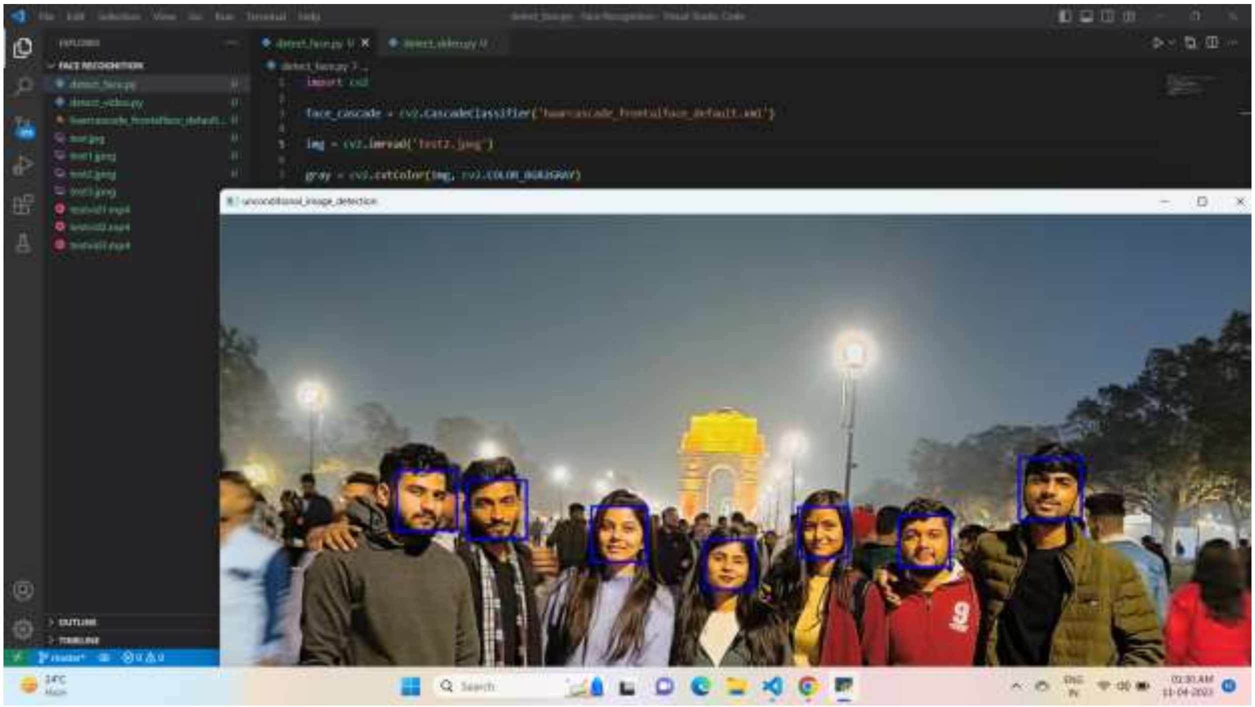
● See: creating a 3D view window for widgets.

6. OpenCV based on Python

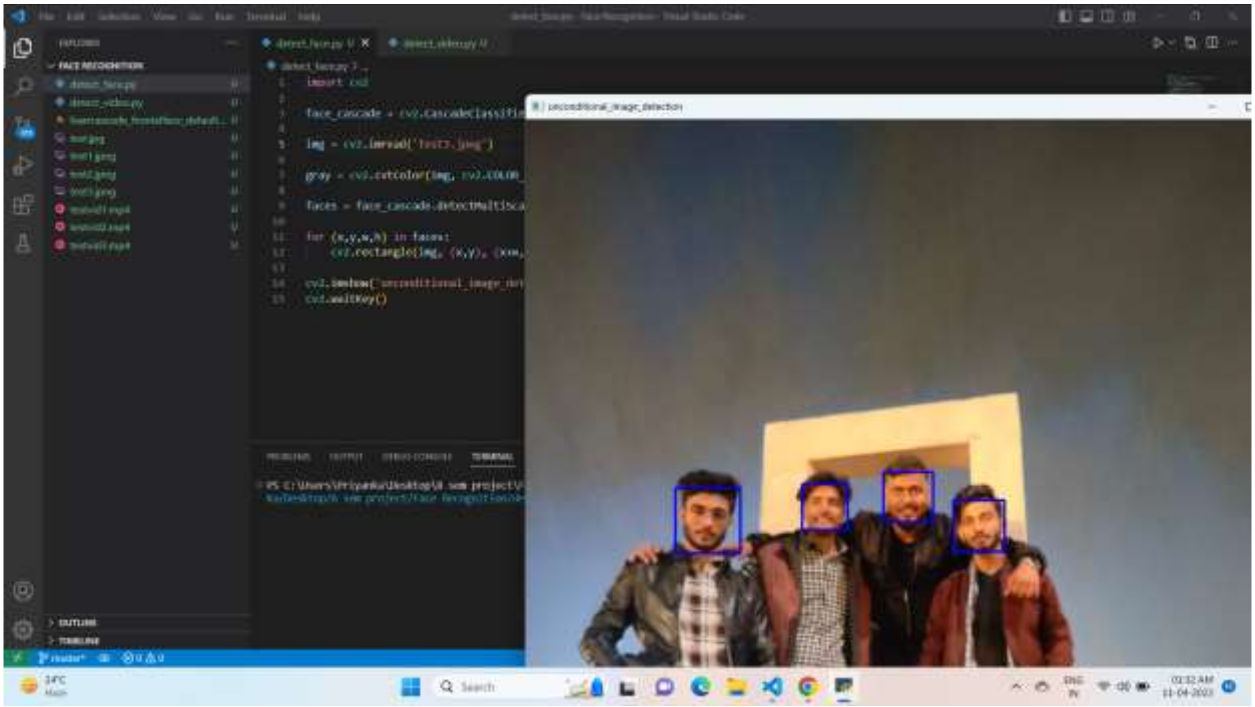
Python was created by Guido van Rossum, who focused on writing clear and easy-to-understand code. With shorter code, the programmer will express the same ideas in fewer characters. Python is slower than some programming languages like C/C++. C/C++ can be effectively extended with Python, but another valuable aspect of Python is that it is generalizable and flexible. This feature helps in implementing C++ codes, which we can then transform into Python modules for ease of use[3]. These two things offer big advantages: first, the code is as fast as C++ code (as it is), and second, Python is easy to program. Python bindings for OpenCV work like this. The Python implementation was built on top of the original C++ code. This is where Numpy's post helps. Numpy is a highly structured library that often offers Python-level statistical capabilities. It does things in a MATLAB-like way. Conversion to Numpy arrays is done and array processing is done on all OpenCV structs. This is very good because any functions that can be done in Numpy can now be combined with OpenCV, making you more efficient. SciPy, which also supports NumPy, can be used along with this. Thus, Python with OpenCV offers a good way to quickly create computer vision projects [11].

CHAPTER - 4

RESULTS AND DISCUSSION

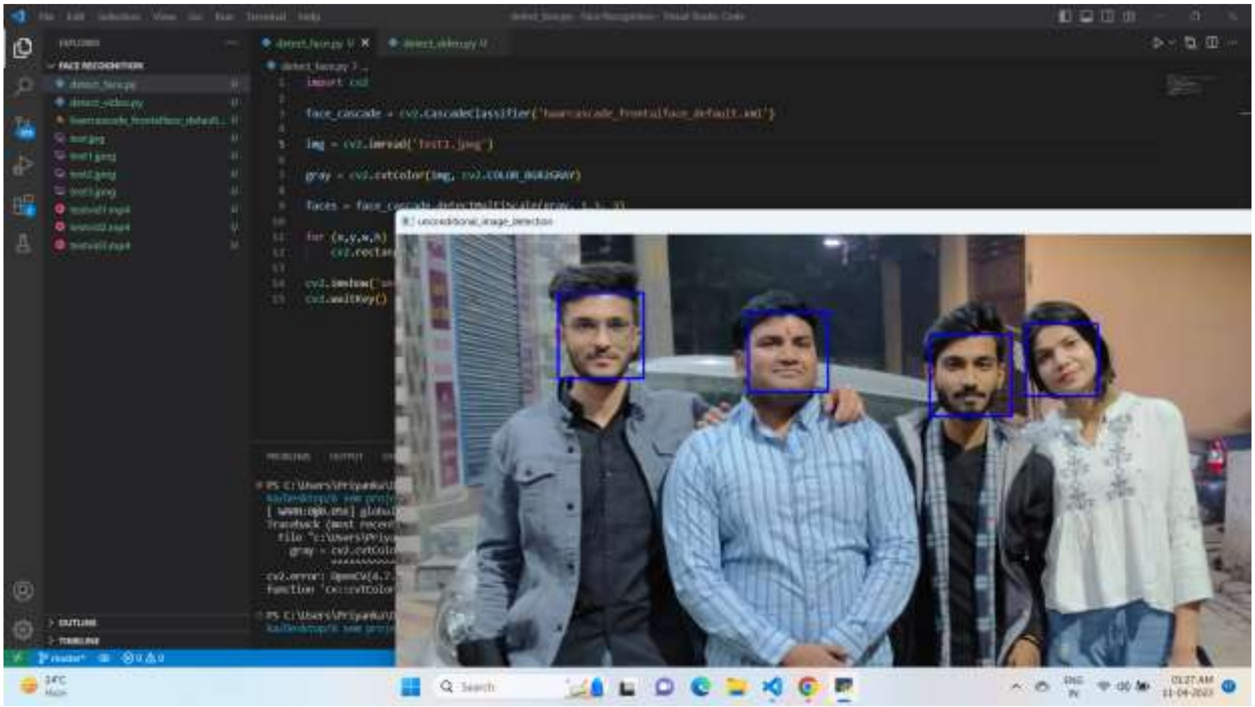


```
import cv2

face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

img = cv2.imread('test2.jpeg')

gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```



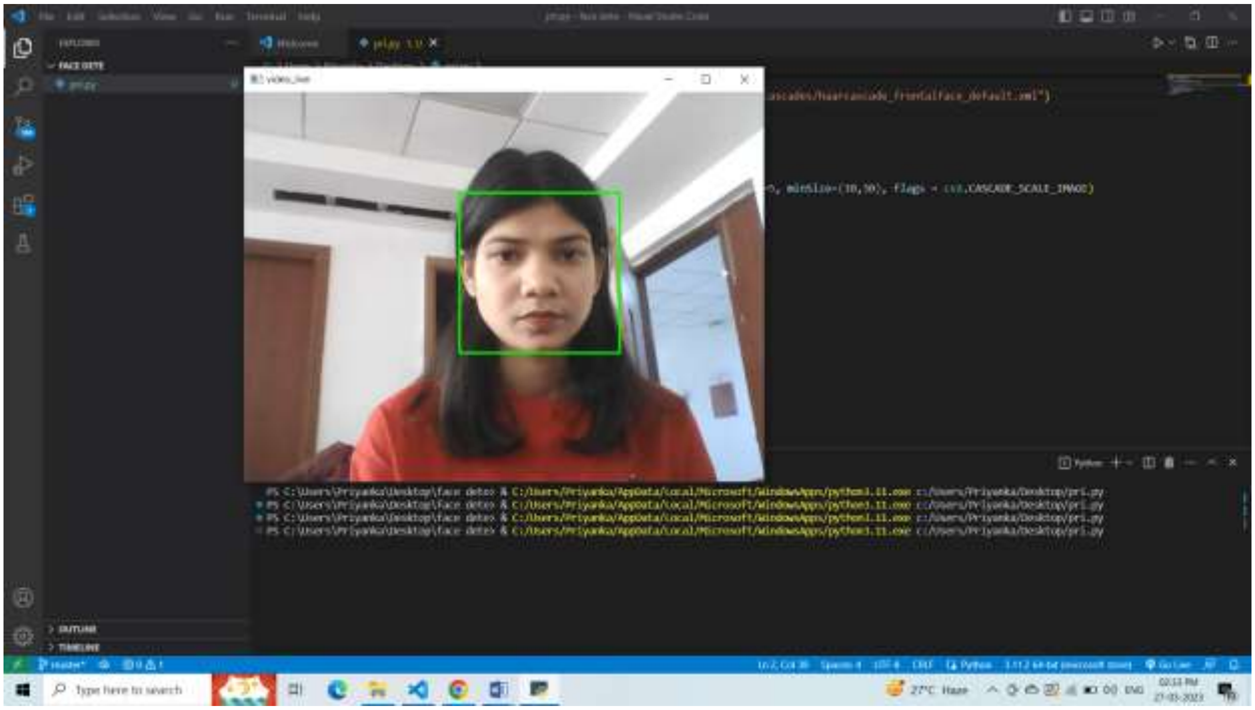
```
import cv2

face_cascade = cv2.CascadeClassifie

img = cv2.imread('test3.jpeg')

gray = cv2.cvtColor(img, cv2.COLOR_

faces = face_cascade.detectMultiSca

for (x,y,w,h) in faces:
    cv2.rectangle(img, (x,y), (x+w,

cv2.imshow('unconditional_image_det
cv2.waitKey()
```

```
PS C:\Users\Priyanka\Desktop\8 sem project\
ka/Desktop/8 sem project/Face Recognition/
```



```
import cv2

face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

img = cv2.imread('test1.jpeg')

gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

faces = face_cascade.detectMultiScale(gray, 1.1, 3)

for (x,y,w,h)
    cv2.rectan

cv2.imshow('un
cv2.waitKey()
```

```
PS C:\Users\Priyanka\
ka/Desktop/8 sem proj
[ WARN:0@0.056] global
Traceback (most recent
  File "c:\Users\Priya
    gray = cv2.cvtColo
           ^^^^^^^^^^^
cv2.error: OpenCV(4.7.
function 'cv::cvtColor

PS C:\Users\Priyanka\
ka/Desktop/8 sem proj
```



```
scades/haarcascade_frontalface_default.xml")
```

```
, minSize=(30,30), flags = cv2.CASCADE_SCALE_IMAGE)
```

```
PS C:\Users\Priyanka\Desktop\face dete> & C:/Users/Priyanka/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/Priyanka/Desktop/pri.py
PS C:\Users\Priyanka\Desktop\face dete> & C:/Users/Priyanka/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/Priyanka/Desktop/pri.py
PS C:\Users\Priyanka\Desktop\face dete> & C:/Users/Priyanka/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/Priyanka/Desktop/pri.py
PS C:\Users\Priyanka\Desktop\face dete> & C:/Users/Priyanka/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/Priyanka/Desktop/pri.py
```

CHAPTER - 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

Computer vision is a subfield of artificial intelligence where computers are trained to process images and extract important features from images or videos. Open Computer Vision (OpenCV), a python library written in C++, provides various functions for computer vision applications. Computer vision applications are object detection, face recognition, medical diagnosis, etc. In this post, we highlight the important role of OpenCV in face detection and face recognition. We illustrate popular algorithms in OpenCV that are used for face detection and face recognition. Then list OpenCV modules and explain Python based OpenCV and mention applications for

OpenCV. Finally, we evaluate and compare recent literature reviews that use OpenCV for human face detection and recognition in various fields to improve human life.

5.2 Future Scope

1. To prevent ATM fraud in India, it is recommended to prepare a database of all ATM customers in banks in India and deploy high resolution cameras and facial recognition software in all ATMs. So whenever a user enters an ATM, their photo will be taken after comparing it with a stored photo from the database to allow access.

2. Duplicate voters are reported in India. To prevent this, it is recommended to prepare a database of all voters, of course, of all electoral districts. Then, at the time of voting, the voting page's resolution camera and facial recognition device will 100% accept the subject's face and generate recognition for voting if a match is found.

3. Passport and visa verification can also be done using facial recognition technology as explained above.

4. Driving license verification can also be done using facial recognition technology as mentioned earlier.

5. To identify and verify terrorists at airports, railway stations and malls, facial recognition technology in India will be the best choice compared to other biometric technologies, as other technologies cannot be useful in crowded places.

6. Facial technology can be deployed in Ministry of Defense and all other important places for better security.

7. This technology can also be used effectively in various important exams like SSC, HSC, Medical, Engineering, MCA, MBA, B-Pharmacy, Nursing Courses etc. The examinee can be identified and authenticated using face recognition technique.

8. In all government and private offices this system can be deployed for identification, authentication and attendance.

9. It can also be deployed in police stations to identify and verify criminals.

10. It can also be deployed in vaults and lockers in banks to verify access control and identify authentic users.

11. The current barcode system could be completely replaced by facial recognition technology as it is a better option for access and security as the barcode could be stolen by anyone. The scope of the Haar Cascade face detection study may include evaluating the performance of the Haar Cascade algorithm in detecting human faces in digital images or videos. The study may involve testing the algorithm on different datasets with different image qualities, lighting conditions and facial expressions. The study can also explore the potential applications of Haar Cascade face detection in various fields such as security, surveillance and digital photography.

However, there are some limitations to the study of face detection using Haar Cascade. One limitation is the sensitivity of the algorithm to changes in the training data. The accuracy of the Haar Cascade algorithm depends on the quality and quantity of training data used to train the classifier. Another limitation is the algorithm's performance in detecting faces under occlusions such as glasses, hats, or other objects that cover parts of the face. The Haar cascade algorithm may not be able to accurately detect faces in such scenarios.

In addition, the study may be limited by the computational resources available for testing the algorithm. The Haar Cascade algorithm can be computationally intensive, especially when working with large datasets or high-resolution images. Therefore, the study may be limited by the computing power and memory of the computer used to test the algorithm.

Overall, the scope and limitations of the study on face detection using Haar Cascade depends on the research questions, objectives and available resources. These factors are important to consider when designing and conducting a Haar Cascade face detection study.

Recommendation for future research:

Here are some recommendations for future research on face detection projects:

Improving the Accuracy of Face Detection Algorithms: While current face detection algorithms have high accuracy, there is still room for improvement. Future research could focus on developing more sophisticated algorithms that can accurately detect faces in challenging scenarios such as low light conditions or occlusion.

Increase the speed of face detection algorithms: Face detection algorithms can be computationally intensive, especially when processing large datasets or high-resolution images. Future research could focus on developing algorithms that are faster and more efficient and enable real-time face detection in video streams.

Develop face detection algorithms for specific applications: Face detection algorithms can be tailored for specific applications such as healthcare, security or entertainment. Future research could focus on developing specialized algorithms for these applications that can accurately detect faces in specific scenarios.

Improve the robustness of face detection algorithms: Face detection algorithms can be vulnerable to attacks such as adversarial examples or spoofing attacks. Future research could focus on developing algorithms that are more resistant to these attacks and ensure the security and reliability of face detection systems.

Explore the Ethical and Legal Implications of Facial Detection: Facial detection technology raises important ethical and legal questions about privacy, surveillance, and bias. Future research could focus on exploring these implications and developing guidelines for the responsible use of face detection technology.

Overall, future research on face detection projects should focus on improving the accuracy, speed, and robustness of face detection algorithms while considering the ethical and legal implications of the technology.

5.3 Complexity

Face recognition is a complex process that involves several steps, including image capture, feature extraction, and matching. The computational complexity of each step may vary depending on the specific algorithm used and the size and quality of the input data.

Facial recognition algorithms typically use machine learning techniques such as neural networks to analyze and identify facial features. Training these models can be time-consuming and resource-intensive, requiring large datasets and considerable computing power.

After training, the algorithm must process the input images and extract relevant features such as facial landmarks, skin texture, and color. This feature extraction step can be computationally intensive, especially when working with high-resolution images or video streams.

Finally, the algorithm must compare the extracted features to be a database of known faces and determine the best match. This matching process can also be computationally expensive, especially when searching large databases or working with multiple frames or real-time video frames.

Overall, the complexity of face recognition can be quite high, especially for real-time applications or when working with large databases. However, advances in machine learning and computer vision continue to improve the speed and accuracy of these algorithms, making them increasingly useful in a variety of applications.

References

[1] Jagtap, A.M., Kangale, V., Unune, K., & Gosavi, P. (2019, February). A study of LBPH, Eigenface, Fisherface and Haar-like features for face recognition using OpenCV. In 2019 International Conference on Intelligent Sustainable Systems (ICISS) (pp. 219-224). IEEE

[2] Sigut, J., Castro, M., Arnay, R., & Sigut, M. (2020). OpenCV basics: a mobile application to support the teaching of computer vision concepts. IEEE Transactions on Education, 63(4), 328-335

[3] Adusumalli, H., Kalyani, D., Sri, R.K., Pratapteja, M., & Rao, P.P. (2021, February). Face mask detection using OpenCV. In 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV) (pp. 1304-1309). IEEE

[4] Mostafa, S.A., Mustapha, A., Gunasekaran, S.S., Ahmad, M.S., Mohammed, M.A., Parwekar, P., & Kadry, S. (2021). Agent architecture for autonomous UAV flight control in object classification and recognition missions. Soft Computing, 1-14

[5] Hoque, M.A., Islam, T., Ahmed, T., & Amin, A. (2020, March). An Autonomous Face Detection System from Real-Time Video Streaming to Provide an Intelligence Security System. In 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS) (pp. 261-265). IEEE

[6] Soomro, Z.A., Memon, T.D., Naz, F., & Ali, A. (2020, January). FPGA based Real Time Face Authorization System for Electronic Voting System. In 2020 3rd International Conference on Computing, Mathematical and Engineering Technologies (iCoMET) (pp. 1-6). IEEE

[7] Gupta, S. (2018, January). Real-time facial emotion recognition and static images. In 2018 2nd International Conference on Innovative Systems and Control (ICISC) (pp. 553-560). IEEE

[8] Dino, H., Abdulrazzaq, M.B., Zeebaree, S.R., Sallow, A.B., Zebari, R.R., Shukur, H.M., & Haji, L.M. (2020). Facial expression recognition based on hybrid feature extraction techniques with different classifiers. TEST Engineering & Management, 83, 22319-22329

[9] Kashinath, S.A., Mostafa, S.A., Mustapha, A., Mahdin, H., Lim, D., Mahmoud, M.A., ... & Yang, T.J. (2021). An overview of data fusion methods for real-time and multi-sensor traffic flow analysis. IEEE Approach

[10] Abdulrazaq, M.B., Mahmood, M.R., Zeebaree, S.R., Abdulwahab, M.H., Zebari, R.R., & Sallow, A.B. (2021, February). Analytical evaluation of the performance of supervised classifiers in facial expression recognition based on relief selection F. In Journal of Physics: Conference Series (Vol. 1804, No. 1, pp. 012055). IOP Publishing

[11] Sriratana, W., Mukma, S., Tammarugwattana, N., & Sirisantisamrid, K. (2018, July). An OpenCVPython application for declaring a personal identifier. In 2018 International Conference on Engineering, Applied Sciences and Technology (ICEAST) (pp. 1-4). IEEE

[12] James, C., & Nettikadan, D. (2019, April). Student monitoring system for school bus using facial recognition. In 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI) (pp. 659-663). IEEE

[13] Balachandran, B., Saad, K.F., Patel, K., & Mekhiel, N. (2019, December). A parallel computer for face recognition using artificial intelligence. In 2019 14th International Conference on Computational Engineering and Systems (ICCES) (pp. 158-162). IEEE

[14] Salihbašić, A., & Orehovački, T. (2019, May). Android application development for gender, age and face recognition using opencv. In 2019 42nd International Convention on Information and Communication Technologies, Electronics and Microelectronics (MIPRO) (pp. 1635-1640). IEEE

[15] Apoorva, P., Impana, H.C., Siri, S.L., Varshitha, M.R., & Ramesh, B. (2019, March). Automatic identification of criminals using facial recognition using open computer vision classifiers. In 2019 3rd International Conference on Computational Methodologies and Communication (ICCMC) (pp. 775-778). IEEE

[16] Gupta, N., Sharma, P., Deep, V., & Shukla, V.K. (2020, June). Automated attendance system using OpenCV. In 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO) (pp. 1226-1230). IEEE

[17] Srivastava, M., Kumar, A., Dixit, A., & Kumar, A. (2020, February). A real-time attendance system using facial recognition technology. In 2020 Power Electronics & IoT Applications in Renewable Energy and its Control (PARC) International Conference (pp. 370-373). IEEE

[18] Sharma, V.K. (2019, May). Design of a facial recognition system. In 2019 International Conference on Intelligent Computing and Control Systems (ICCS) (pp. 459-461). IEEE

[19] Das, A., Ansari, M.W., & Basak, R. (2020, December). Covid-19 face mask detection using TensorFlow, Keras and OpenCV. In 2020 IEEE 17th International Conference of India Council (INDICON) (pp. 1–5). IEEE

[20] Mehariya, J., Gupta, C., Pai, N., Koul, S., & Gadakh, P. (2020, July). Counting students using OpenCV and integrating with Firebase for classroom allocation. In 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC) (pp. 624-629). IEEE

[21] Bojko, N., Basystiuk, O., & Shakhovska, N. (2018, August). Performance evaluation and comparison of face recognition software based on dlib and opencv library. In 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP) (pp. 478-482). IEEE

[22] Alcantara, G. K. L., Evangelista, I. D. J., Malinao, J. V. B., Ong, O. B., Rivera, R. S. D., & Ambata, E. L. U. (2018). Head detection and tracking using OpenCV. In 2018 IEEE 10th

International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM) (pp. 1-5). IEEE

[23] Patel, R., Patel, M., & Patel, J. (2018, April). A real-time somnolence detection system in OpenCV for drivers. In 2018 Second International Conference on Innovative Communication and Computing Technologies (ICICCT) (pp. 407-410). IEEE

[24] Sharma, S., & Jain, S. (2019, March). A static hand and face gesture recognition system for the blind. In 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN) (pp. 534-539). IEEE.

[25] Zhu, Z., & Cheng, Y. (2020). Application of OpenCV-based Face Recognition Position Tracking Algorithm in Smart Door Lock. Computer Communications, 154, 390-397

[26] Chandan, G., Jain, A., & Jain, H. (2018, July). Real-time object detection and tracking using Deep Learning and OpenCV. In 2018 International Conference on Invention Research in Computer Applications (ICIRCA) (pp. 1305-1308). IEEE

[27] Khan, M., Chakraborty, S., Astya, R., & Khepra, S. (2019, October). Face detection and recognition using OpenCV. In 2019 International Conference on Computing, Communication and Intelligent Systems (ICCCIS) (pp. 116-119). IEEE

[28] Palekar, R.R., Parab, S.U., Parikh, D.P., & Kamble, V.N. (2017, April). Real-time license plate detection using openCV and tesseract. In 2017 intern