2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS) | 979-8-3503-9737-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICACCS57279.2023.10113066

An approach for Face Detection and Face Recognition using OpenCV and Face Recognition Libraries in Python

Ainampudi Kumari Sirivarshitha
Department of CSE
Koneru Lakshmaiah Education Foundation
Vaddeswaram, AP, India
sirivarshitha115@gmail.com

Kadavakollu Sravani
Department of CSE
Koneru Lakshmaiah Education Foundation
Vaddeswaram, AP, India
kadavakollusravani2001@gmail.com

Kothamasu Santhi Priya
Department of CSE
Koneru Lakshmaiah Education Foundation
Vaddeswaram, AP, India
k.santhipriya2002@gmail.com

Vasantha Bhavani
Department of CSE
Koneru Lakshmaiah Education Foundation
Vaddeswaram, AP, India
vasanthabhavani@kluniversity.in

***Abstract*—With the extraordinary growth in images and video data sets, there is a mind-boggling want for programmed un- derstanding and evaluation of data with the assistance of smart frameworks, since physically it is a long way off. Individuals, unlike robots, have a limited capacity to distinguish unexpected expressions. As a result, the programmed face proximity frame- work is important in face identification, appearance recognition, head-present evaluation, human-PC cooperation, and other ap- plications. Software that uses facial recognition for face detectionand identification is regarded as biometric. This study converts the mathematical aspects of a person's face into a face print, which is then stored in a database to verify an individual's identification. A deep learning system compares a digital image or an image taken quickly to a previously stored image(whichis saved in the database). The face has a significant functionin interpersonal communication for identifying oneself. Face recognition technology determines the size and placement of a human face in a digital picture. Facial recognition software hasa wide range of uses in the consumer market and in the security and surveillance sectors. The COVID pandemic has brought facial recognition into greater focus lately than ever before. Face detection and recognition play a vital part in security systemsthat people need to interact with without making physical contact. The pattern of online exam proctoring is employing face detection and recognition. Facial recognition is used in the airline sector to enable rapid, accurate identification and verification at every stage of the passenger trip. In this research, we focused on image quality because it is the major drawback in existing algorithms and used OPEN CV, Face Recognition, and designed algorithms using libraries in python. This study discusses a method for facial recognition along with its implementation and applications.**

***Keywords*—Face Detection, Face Recognition, Face Alignment , Feature Extraction, Python, OpenCV Library, face Recognition Library**

I. INTRODUCTION

Face detection has sparked a lot of attention since it is frequently employed in vision systems for computers for autonomous control and communication. A face-detection system is a technique to identify a face from a picture with several faces featured in that picture. The concept of feature vectors serves as the foundation for most face recognition theories. Research is needed in the areas of face detection, emotion identification, face tracking, and posture estimation. The task of identifying the face in a single photograph is presented. Face detection is tough since faces are not static and change in size, shape, color, and other aspects. When the provided image is blurry, obscured by anything else, has poor lighting, is not facing the camera, etc., face identification becomes a more difficult process. A feature vector is created by converting the facial picture into a string of integers. The content of a feature vector for a face photograph includes properties such as head level, face breadth, average face color, lips width, and nose height. The face-recognition technique evaluates the input feature vector of a face photograph with a large range of feature vectors in a collection of data to identify a person's identification. The Python environment is mostly used to develop the facial recognition system in this study. Face generation is accomplished using selfies. The image is cropped to keep the face's portion, then it is saved to the database. If the resemblance score is higher than the real threshold, compare the incoming facial images with those kept in the data collection. The application will display the identification of faces by picture. Otherwise, a bogus notice will be shown by the system. The facial recognition system has been completed, and the trial results show that it is effective. The facial image is finally captured using the laptop Webcam, and the comparison outcome is likewise accurate.

II. LITERATURE REVIEW

Although face recognition is simple for humans, it is more challenging for computers since the human body is not rigid, meaning that changes will occasionally occur in it. Face detection is the first and most crucial issue for face recognition. The first step in automatic face recognition is face detection. To address this issue, we may develop a set of various facial picture locations for a person. Face images may differ due to differences in facial expressions, even when the expression and position are the same. Lighting may also cause variations in face images when the same expression and position are used. This study focuses on the lateral face identification system, which can detect faces while having a low proportion of false positives. According to M. Yang and his team in[1], there are four categories of face recognition techniques: knowledge-based, template-based, feature invariant, and appearance-based.

A. Knowledge based Method

This approach looks for relationships between facial features relationships based on prior information. It relies on stored knowledge of what makes up a human face and is used for localizing human faces.

B. Feature Invariant Approach

These techniques, which are also utilized for face local-ization, are used to find faces in a variety of situations, for example, low or high light densities, poses from various angles, and combinations of these conditions. This approach is used to resolve various facial poses.

C. Template based Method

Both face localization and face detection may be accomplished using the template-matching approach. This approach keeps a number of common patterns of the human face, and those patterns can be used to describe the full face or a particular facial feature. For detection, a connection must be determined between the previously recorded pattern and the input picture.

D. Appearance based method

The model of the picture is learned using a series of training photos provided in this manner rather than templates. These models are then used in the face-detection process.

Yang and Huang used a face detection strategy in [2]; in this technique, a set of specifications are described as a knowledge-based approach, and it contains rules on the face of an input picture before employing face scanning to locate every conceivable face. At the highest level, for example, a collection of rules is developed to characterize a human face, while at the lowest level, rules for facial features are offered. For face identification from pictures, many rules and layers of rules are devised. It is possible to generate multi-layer picture hierarchies by averaging and sub-sampling. Sung and Piggo developed a face identification strategy as a distribution-based system [3], and an object class was taught using both positive and negative examples. A distribution-based system consists of two halves: a multi-layer perceptron classifier and distribution-based models for face-nonface patterns. Each picture is vectored in 361 dimensions after each face example has been processed and normalized as a 19*19-pixel image. Images are now grouped into six clusters for faces and six clusters for non-faces using a modified K-means method.. There are two distances between a cluster and an input picture. Using a modified K-means approach, images are now categorized into a 6-face and a 6-non-face cluster. There are two distances between an input image and a cluster. Calculations, normalized Mahalanobis distance, and Euclidian coordinates distance. Finally, a multi- layer perceptron network is employed to use a twelve-pair to separate face- and non-face-containing window patterns. Each face cluster and non-face cluster in terms of distance. In [4] a method to express static color pictures in a tensor- based format was proposed by Lajevardi and Wu. In a colored picture, it is utilized to identify facial emotions. It has 68.8 percent accuracy in color images of various resolutions. In 1977, [5] Ekman and Friesen suggested a reliable technique for facial activity coding. The Face Action Coding System attempts to analyze facial emotion by monitoring the shifting behavior of the facial muscles. The average detection rate is 7.8 percent higher with this approach. The program uses 46 Action Points (AP), which represent human facial behavior. In [6], Divya Meena and Ravi Sharan worked on the viola Jones algorithm to detect face and principal component analysis for face recognition when the image is not clear and occluded by any other thing and not proper lightning, not facing the camera, etc. In [7], Mohammad Abu-Lebdeh and his team presented a new architecture for EPC-based mobile video surveillance applications. The design enables quick development and deployment of new applications while also providing assured and differentiated QoS, which is not feasible in other networks. A proof-of-concept prototype was successfully developed; as evidenced by the performance measurement study, the delays experienced by adopting the suggested architecture were reduced. Geetha and her team [8], used SVM and Eigenface algorithms to improve the accuracy of existing face recognition systems. They used a method similar to Eigenface is utilized to extract facial features from facial vectors, and the datasets are trained using the Support Vector Machine (SVM) algorithm for face classification and detection, which guarantees that face recognition is faster and that it may be utilized for online exam monitoring. In [9], Menq-Jiun Wu and his team achieved face recognition using the idea of feature vectors. Jayanth Vadlapati and his team in [10], using face recognition modules from python's huge collection of libraries, are able to train the model to recognize people while wearing masks.

III. PROPOSED APPROACH

A. Python

Python is presently the most widely used pro- gramming language. Python is straightforward but flexible, which is why programmers adore it. Python is capable of tackling complicated jobs despite being simple. Python has so far been utilized for the back end. Python is still a fairly dependable programming language for image recognition. Image

recognition is well supported by the Python library. Scikit-Learn Machine Learning, which is renowned for its capacity to handle facial recognition and motion detection, is one of the most potent and effective Python modules.

B. OpenCV Library

Using the OpenCV library is the most well-liked and most likely the easiest approach to recognizing faces using Python. OpenCV, which was first created in C/C++, now has Python bindings. To find faces in a photograph, it employs machine learning techniques. Faces are extremely intricate, consisting of countless minute patterns and characteristics that must match. The facial recognition algorithms, also known as classifiers, divide the identification of the face into thousands of manageable, bite-sized jobs. A face may have 5000 classifiers or more, and each one of them must match in order for a face to be recognized. We may have to perform millions of calculations because each block has at least 5,000 tests, if not more. Cascades are used by OpenCV to address this. The OpenCV data needed to detect objects is included in a collection of XML files called cascades. The code performs the work for us after we initialize it with the desired cascade. Given how frequently faces are detected, OpenCV has a plethora of algorithms for recognizing everything from heads to hands to toes.

C. Face Recognition Library

The face recognition library contains numerous methods (functions) for dealing with faces in pictures, one of which is known as face positions, which will locate the location of the face inside a certain image and we will pass the image file that we used in the previous line of code.

The world's most basic face recognition library allows us to recognize and operate with faces from Python or the command line. The model was developed using the most powerful facial recognition technology available from dlib, and it has an accuracy of 99.38% on the labeled faces in the wild benchmark. In addition, a simple face recognition command line application is supplied, allowing us to perform face identification on a folder of images directly from the command line.

D. Face Detection

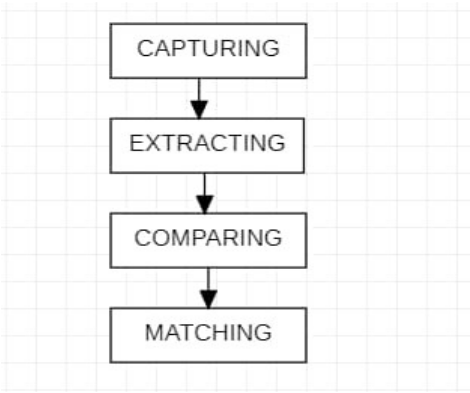

Fig. 1. Work Flow Diagram

Face detection is a computer technique that recognizes human faces in digital images and is utilized in many different applications. Face detection finds many faces in an image together with the essential facial characteristics that go with them, including emotional state, or uses the expressions on the faces to identify age, gender, and emotions. Many face-related technologies, such as face identification or verification, typically start with face detection. Work flow diagram is shown in Fig. 1.

Face detection algorithms can be classified into two or more of the four categories in which these approaches are classified. These are the classifications:

1) Knowledge-Based: To identify faces, this method relies on a set of rules and is based on human knowledge. An example would be that the nose, eyes, and mouth of a face must all be at specific angles and distances from one another.

2) Feature-Based: Using structural elements of the face, the feature-based technique locates faces. It is used to distinguish between facial and non-facial regions after being trained as a classifier.

3) Template Matching: The Template Matching approach locates or detects faces by correlating input photos and pre- defined or parameterized face templates. Examples include theeyes, face shape, nose, and lips of a human face. Additionally,a face model can be constructed solely from edges using edge detection.

4) Appearance-Based: In order to identify face models, theappearance-based technique uses a collection of delegate training face photos. The appearance-based method outperforms other modes of performance. In order to identify the pertinent features of face photos, appearance-based methods typically use statistical analysis and machine learning approaches.

E. Face Alignment

Face alignment is crucial in the majority of face analysis procedures. The practice of identifying the geometric composition of faces in digital photos and attempting to create a canonical alignment of the face based on translation, scale, and rotation is known as face alignment. It focuses on detecting a few key locations on human faces in photos or videos.

There are several types of facial alignment. Several strategies try to impose a (pre-defined) 3D model and then change the input picture so that the landmarks on the input face match the landmarks on the 3D model. Other, simpler systems (such as the one described in this blog article) depend solely on facial landmarks (especially the eye regions) to generate a normalized rotation, translation, and scale representation of the face. Block Diagram of Face Detection and Recognition is shown in Fig. 2.

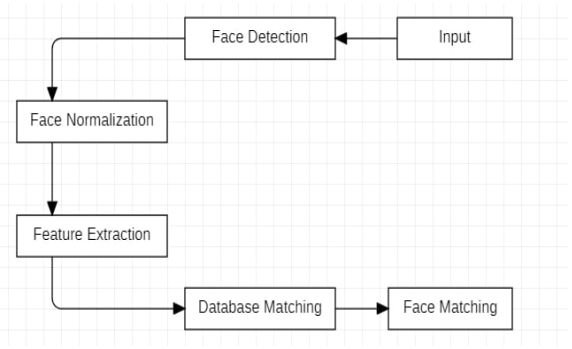

Fig. 2. Block Diagram of Face Detection and Recognition

F. Feature Extraction

In various pattern recognition tasks, dimension reduction techniques have demonstrated significant benefits, and face processing is no exception. Typically, we are able to compress the input data, which lowers the storage requirements. Due to less noise, which is present in the majority of natural photos, the classification results have also improved in other instances. One of the most used methods for dimensionality reduction is principal component analysis. Feature extraction significantly reduces the amount of data while still obtaining the optimal feature from those enormous data. Following the face detection step, the face extraction procedure is completed to produce accurate data that can be used to discriminate between the faces of various people.

The technique of extracting facial feature components from a human face photograph includes elements like the lips, nose, and eyes. For the start of processing techniques like face tracking, facial emotion recognition, or face recognition, facial feature extraction is crucial. To process features like shapes, edges, or movements in a digital picture or video, a feature extraction procedure is used. The extraction of facial features is a fundamental step in computerized visual observation and human face recognition. Identifying facial characteristics is crucial in a variety of applications, including human-computer interaction, facial animation, and face identification.

G. Face Recognition

Facial recognition is a method of recognizing or verifying an individual's identity by utilizing their face. Face recognition algorithms can distinguish people in photographs, films, and in real-time.

In order to identify certain, recognizable features on a person's face, face recognition systems employ computer algorithms. Then, a mathematical representation of these details, such as the separation between the eyes or the contour of the chin, is created and compared to information on other faces gathered in a face recognition database.

The steps involved in face recognition are detection, alignment, feature extraction, and task recognition. Face recognition has been implemented at border crossings, during Olympic Games activities, and in airports. Face recognition technology may also be used in private locations like shops and sporting venues; however, these locations may be subject to various regulations. Flow of Image Recognition and Matching is shown in Fig. 3.

IV. Result Discussion

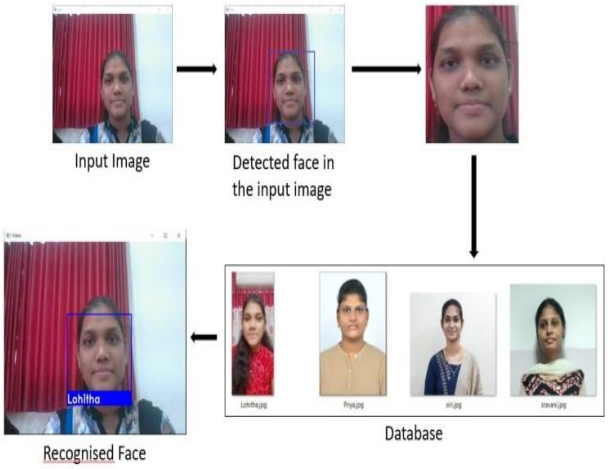

Fig. 3. Flow of Image Recognition and Matching

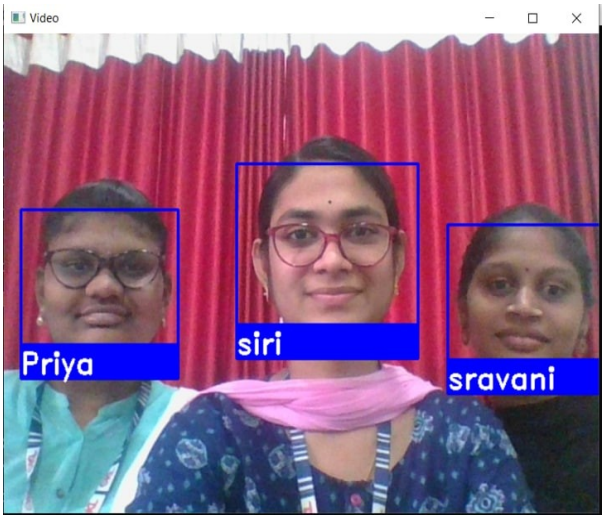

Fig. 4. Successful Multiple Faces Recognition

Most often, the accuracy of the algorithm is used to gauge how effective a face recognition system is. The algorithm's capacity to identify the face input and provide the percentage of the match is how accuracy is determined. The algorithm's presentation of the closest match percentage is crucial. In this model, the algorithm takes the image of the person and detects the face, and recognizes and displays the name from the images it has. As a result, this facial recognition technology may be used in place of passwords to get access to computers. This method has been tested on a larger database with more photos and has an 80% accuracy rate. When face recognition technology becomes more widely available, public concern about unwarranted stops and searches may become less of an

issue for police. Succesful Multiple Faces Recognition is shown in Fig. 4.

V. CONCLUSION

In this report, face detection and recognition using Python environment by using OpenCV and face recognition we have come to the conclusion that, the OpenCV library is becoming more and more successful and performs better when it comes to locating and identifying faces. It further suggests that building recognition apps at the IOT stage with OpenCV is a wiser choice. An algorithm for both Face Detection and Face recognition is derived, which helps to identify multiple faces present in image. Used image resolution to get high quality images which helps to detect the face within seconds.

Security is one of the sectors that use facial recognition that employs facial recognition technology the most. Software busi- nesses are utilizing facial recognition technology to make it easier for consumers to use their technology. Facial recognition is a very powerful technique that may assist law enforcement in identifying criminals. This technology may be improved to be utilized in several contexts, such as ATMs, accessing private information, or handling other delicate materials. Other security mechanisms like passwords and keys can become outdated as a result of this. Face recognition would scan your face, run it through a system, and charge the account that you've already registered instead of requiring you to visit a kiosk to purchase a ticket for a fee. This may significantly simplify the procedure and improve traffic flow. Further study will aim to improve the accuracy of face detection and recognition.

VI. FUTURE SCOPE

Developing identification programs using OpenCV during the IOT stage is a preferable choice since face recognition is a sophisticated technology that may assist public safety in detecting suspects. This technique might be improved to be utilized in a variety of circumstances, including ATMs, accessing private data, and handling other delicate materials. Other security mechanisms, such as passwords and keys, may become outdated as a result. Instead of requiring you to visit a kiosk to purchase a ticket, facial recognition would scan your face, run it through a system, and charge the already registered account. This might considerably simplify the procedure and improve traffic flow.

REFERENCES

[1] Lajevardi, S. M., & Wu, H. R. (2012). Facial expression recognition in perceptual color space. IEEE Transactions on Image Processing, 21(8), 3721-3733.

[2] Yang, G., & Huang, T. S. (1994). Human face detection in a complex background. Pattern recognition, 27(1), 53-63.

[3] Yang, M. H., Abuja, N., & Kriegman, D. (2000). Face detection using mixtures of linear subspaces. In Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on (pp. 70-76). IEEE.

[4] Sung, K. K., & Poggio, T. (1998). Example-based learning for view-based human face detection. IEEE Transactions on pattern analysis and machine intelligence, 20(1), 39-51.

[5] Sung, K. K. (1996). Learning and example selection for object andpattern detection.

[6] Divya Meena, Ravi Sharan(2016), An Approach to Face Detection and Recognition.

[7] Maliha Khan, Sudeshna Chakraborty, Rani Astya, Shaveta Khepra(2019), Face Detection and Recognition Using OpenCV.

[8] M.Geetha, R.S.Latha, S.K.Nivetha, S.Hariprasath, S.Gowtham, C.S.Deepak(2021), Design of face detection and recognition system to monitor students during online examinations using Machine Learning algorithms.

[9] Menq-Jiun Wu, Yeong-Chin Chen, Yi-Sheng Liao, Jun-An Chen, Hao-Han Lin(2021), Face-recognition System Design and Manufacture.

[10] Jayanth vadlapati, S Senthil Velan, Ewin Varghese(2021),Facial Recog-nition using the OpenCV Libraries of Python for the Pictures of Human Faces Wearing Face Masks during the COVID-19 Pandemic.