Motion Detection and Multiple Faces Identification using Webcam

Varad Kalkhaire[1], Shubham Dhanne[2], Ashutosh Khatavkar[3], Shantanu Bhumkar[4]

Prof. R. V. Agawane (BE, ME)[5]

[1-4] Student, Department of Computer Engineering, KJCOEMR, Pune

[5]Professor, Computer Department, KJCOEMR, Pune

---***---

Abstract – As technology is marking the highest peak, all things are being automated using Machine Learning and Deep Learning. By using Machine Learning models, we can detect the movements, actions, multiple faces, their recognition and many more in snap of time. Just like that, this project is implemented using Machine Learning models, like OpenCV for detecting motions of an object and Face Recognition for multiple faces recognition to provide the advanced security. In this paper, we are going to see the project named as "Motion Detection and Multiple Face Identification using Webcam" were detection of motion and multiple faces will be done with the help of a system Webcam or Security Cameras. This is implemented using Python's OpenCV a Computer Vision module and Cascade Classifier. This research paper will guide, how this machine learning project helps to detect and verify the motion and faces of people more efficiently and provide advanced security to an organization or to a home.

Key Words- Machine Learning (ML), Artificial Intelligence (AI), MYSQL, Haar-Cascade Classifier, Gaussian-Blur, Face Recognition, MySQL Database.

1. INTRODUCTION

The title "Motion Detection and Multiple Faces Identification using Webcam" tells us much more about this system. The "Motion Detection" were the actions made by an object is going to be observed. And the "Multiple Faces Identification" were detection and recognition of multiple faces is done at same time. Both model use Webcam / Security Camera for detecting motion and multiple faces in live video capturing, and a database is used to recognize multiple faces and storing of newly detected face.

There are many approaches for motion detection in a continuous video stream. All of them are based on comparing of the current video frame with one from the previous frames or with something that called as background. In this paper, there are four approaches are used and comparison is made to find out a best detector for an effective motion detection [5]. For object detection and counting, OpenCV includes a number of useful techniques [2]. The detection algorithm uses the advantage of background subtraction and fed in data to detect even the slightest movement, this system makes use of a webcam to scan a premise and detect movement of any sort [3]. The main part in the system is difference between two continuous frames [4] and alert system according to detected motion.

Similarly, Face Recognition in a real-time setting is an exciting area and a rapidly growing challenge [1] like Motion Detection, Face detection and picture or video recognition is a popular subject of research on biometrics. Face recognition is a challenging term and making facial recognition automatically is a big, exciting problem and there are many different approaches to solve this problem because it must detect the multiple faces (multi-face). Especially in uncontrolled real-life scenarios, faces will be seen from various sides and not always facing forward, which makes classification problems more difficult to solve [8]. In this article, we intend to implement the Haar-Classifier for Face detection and tracking based on the Haar-Features. Features can be eye colour, eye shape, skin contour, face shape, nose shape, lips shape and colour, lengths between lips and nose or between both eyes and eyebrows style and shape, etc., are going to capture using Webcam and also stored on a database to recognize a single face or multiple faces (multi-face).

The rest of the article is structured as follows: given below is the architectural design of the system and main objectives present in the system, then in next section we give the descriptive information of our system, focusing on Gaussian-Blur and Haar-Cascade Classifier modules. In Section III the result of the system is reported and afterward the conclusion and future scope about the project is given.

A. System Design

Below in Figure 1, is the system architecture diagram, where video footage will be captured by using Webcam and then it will be sent to system for processing and finding out

for any movement of objects or for any facial feature for recognizing the detected faces with help of database.
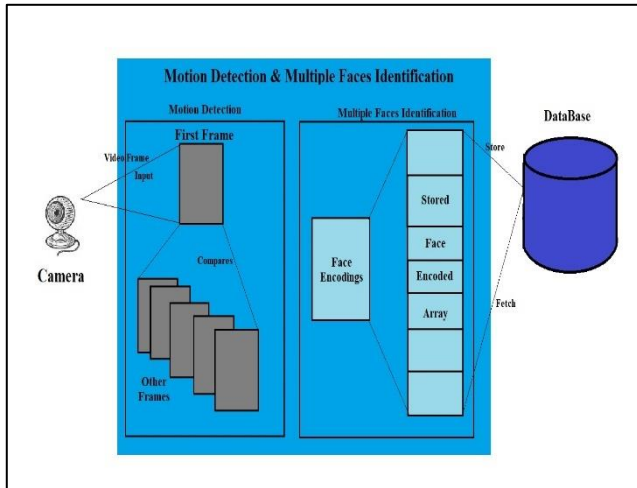

Fig -1. System Architecture

The above system is divided into four parts as:

1) Webcam
2) Motion Detection
3) Face Recognition
4) Database

1) Webcam:

Webcam is used to capture the video footage in real-time to detect the motion and to perform verification on multiple faces by system itself. Webcam's video footage is going to process to find the insights from it.

2) Motion Detection:

The detection of Motion is done using the OpenCV's Gaussian-Blur method. In Gaussian-Blur, the image is convolved with a Gaussian filter instead of the box filter. The Gaussian filter is a low-pass filter that removes the high-frequency components are reduced [6]. Which helps to detect the motions of an object easily and alert the system.

3) Face Recognition:

After no motion is detected then system will start automatically to detect multiple faces and their features to recognize. For this, we use Haar-Cascade Classifier to get the detection. It is a machine learning based approach in which a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images [7]. The Face Recognition module is used for recognizing the human face.

4) Database:

Database is used for storing and fetching the features of faces. Features like eye shape and colour, face shape, skin contour, nose shape, etc., For the database, we are using MySQL to create the database for fetching and storing the facial features with their names.
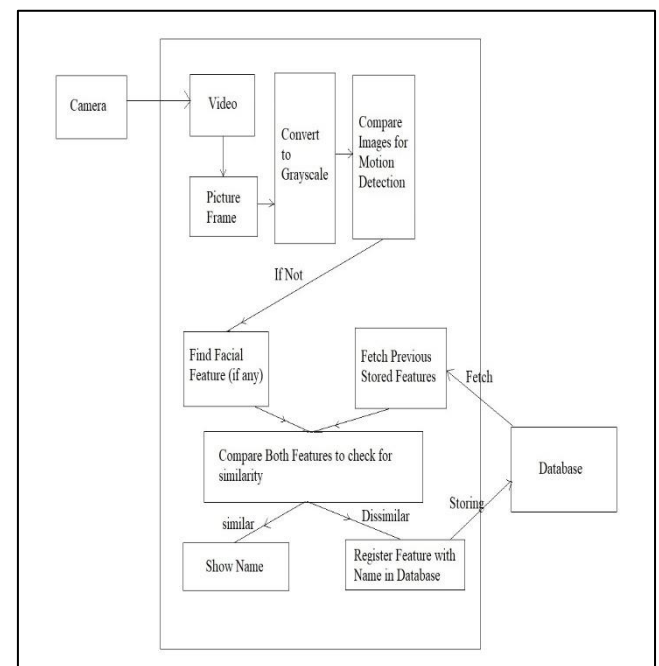
2. PROPOSED SYSTEM

Fig -2. Data Flow Diagram

In above Figure 2, as soon as Webcam captures the video footage, it will be sent to system for further process of that footage were pictures frames will be taken out i.e., first frame will be base frame and others frames will be used for comparisons. Before performing the comparisons, the picture frames will be converted into grayscale for smoothness and for reducing the noise present in the frames. This all above process will be done using Gaussian-Blur method; which supports comparison of frames, where the first frame is compared with other frame for detecting the motion made by an object. In Motion Detection system, it divides the picture frames into pixels and starts finding the motions by comparing both frames. If difference is found, then there is some kind of motion detected otherwise no motion / movement took place.

Similarly, it will start scanning for facial feature using the Haar-Cascade Classifier module, to detect multiple faces. Then start checking whether the facial features present in database matches the features captured by camera; if matched

then fetch the name of person whose face is captured by Web Camera. Haar-Cascade Classifier, can be used to detect the multiple faces.

Haar-Cascade Classifier, is an OpenCV module, which provides the extraction of features. The extracted features will be compared with previously stored features present in the database. Haar-Cascade Classifier, uses lots of negative and positive images to find the features from given picture frames. It detects the features using black and white colour code i.e., 0 or 1 in binary value for determination and detection of the multiple faces. With the help of black and white colour code of pixel, it is lot easier to detect multiple faces from the given input picture frames. At the same time, the Face Recognition module is used to fetch the previously stored features on database for the comparison of features to identify the faces. If the features matches then the face is identified else face will be new one and must be entered in the database.

This project is built to get the better and efficient results of face detection and face identifications within seconds of time by using the camera of the computer. It uses real time camera footage, so that system can start to detect multiple faces and try to recognize it. This system uses Haar-like features like rectangle, square or circular shape or line for locating face locations. The Face Detection process is done using Haar-like feature like rectangle to locate the face in the picture frame. Whereas Face Recognition module is use for the extraction of the features from the picture frame, which is encoded for better comparison. If the comparison matched with the current facial features (which are also encoded), then the system will fetch the name of face from the database to display it; otherwise, the system will display the name as "Unknown" and will also upload it into the database for registering it later.

For this whole system, two tables in databases are required for storing the human facial features and name respectively. One table is required for storing the new facial features for the first time, known as 'unknown' table, whereas second table named 'known' is required for storing the facial features after completing the registration of faces from unknown table.

The above three systems Motion Detection, Face Detection and Face Recognition are combined together to form a one system, which run simultaneously to detect motions and multiple faces along with verifications of faces. This system also uses the Webcam or Camera or CCTV Cameras for giving the inputs to the models present in the system. This system is much more dependent on Camera hardware configuration and the advanced configured system for better execution. This whole project uses Machine

Learning models to detect faces, motions and facial verification of human faces quickly and more efficiently.

3. RESULT

The results of the project are in the form of an active video frame, which is a continuous stream of video captured using the Webcam. In this stream of video, the motion detection, multiple face detection and multiple face recognition is done simultaneously.
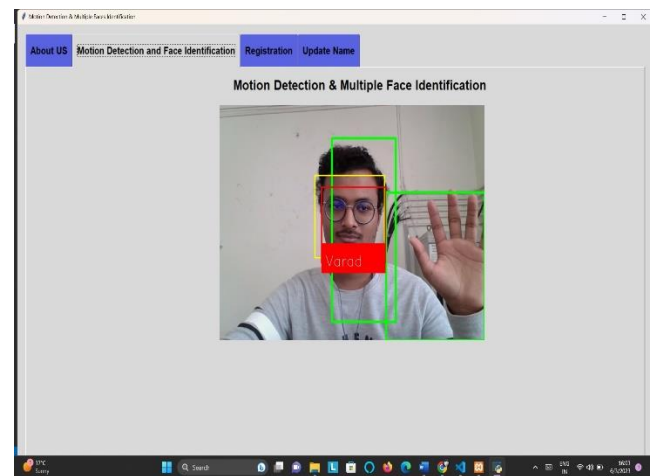

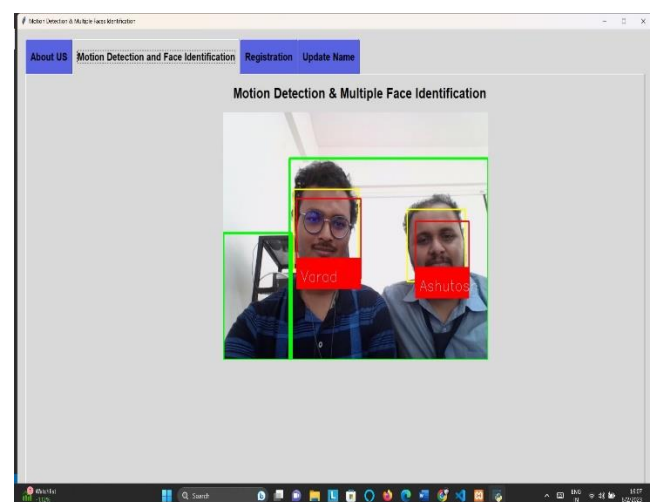Fig -3. Motion Detection and Face Identification

Fig -4. Motion Detection and Multiple Faces Identification

In above Figure 3 and 4, the Green Boxes in the result shows how the use of Gaussian-Blur model helps the motion detection model to execute efficiently and find the even the smallest movements or actions made by human or an object.

Gaussian-Blur makes the motion detection model faster by using grey scale image processing.

The Yellow boxes in the result shows the execution output of multiple face detection. By using Haar-Cascade Classifier model, it is quite easy to detect multiple faces at the same time without taking more execution time.

Also, the Red Boxes in result shows, how efficiently and effectively the model recognizes multiple faces by fetching their names from the database without getting any error and reducing the execution time. The Face Recognition module helps to recognize the human faces using the Webcam rather than the Sensors. It uses the Face Recognition Module which helps to recognize the faces in better way.

4. CONCLUSION AND FUTURE SCOPE

According to the results, we conclude that the Gaussian-Blur, Haar-Cascade Classifier and Face Recognition model helps to build the entire project effectively.

The model detects and recognizes multiple faces simultaneously by using the Webcam of the computer. This project successfully shows how the model helps to capture the multiple faces, their detection and verification using the Haar-Cascade features and Face Recognitions Module. All the detection and recognition are done by the system with much fewer human interfaces, which makes this system automated and can give more advanced security.

Even when detecting multiple faces, the model also detects the motion of the objects and alerts the system about the movements captured by the Web Camera which will be much more helpful for providing advanced security to an organization or to a home.

This model can be used in "Mental Hospital" for keeping track of mentally-ill patients, and keeping their facial data in the database to track them even into a crowded area using the Security or Web Camera. It also helps in keeping the facial data of the patient's family to avoid cases of fraud.

This model can also be used in the Airline industry for keeping track of bags, people and boarding people. By using the recognition model, we can recognize the people at the check-in, which will help speed up the check-in process while ensuring security protocols and measures are followed, and also make the duties easier for the security personnel and airline staff.

5. ACKNOWLEDGMENT

6. REFERENCES

[1] M. Khan, S. Chakraborty, R. Astya and S. Khepra, "Face Detection and Recognition Using OpenCV," 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), Greater Noida, India, 2019, pp. 116-119, doi: 10.1109/ICCCIS48478.2019.8974493.

[2] Shubham Mishra, Mrs. Versha Verma, Dr. Nikhat Akhtar, Shivam Chaturvedi, Dr. Yusuf Perwej, " An Intelligent Motion Detection Using OpenCV, International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Print ISSN : 2395-1990, Online ISSN : 2394-4099, Volume 9, Issue 2, pp.51-63, March-April-2022.

[3] N. Pooja, B. S. Bhaskar, J. Aman, and C. Sarika, "Motion detection of webcam using frame differencing method," Research Journal of Engineering and Technology, vol. 12, no. 2, pp. 32–38, 2021.

[4] Ankita Rameshwar Mahajan and Vinod Agrawal, "Movement Detection Using OpenCV", International Research Journal of Modernization in Engineering Technology and Science, vol. 04, no. 05, 2022.

[5] C. Y. Yong, R. Sudirman and K. M. Chew, "Motion Detection and Analysis with Four Different Detectors," 2011 Third International Conference on Computational Intelligence, Modelling & Simulation, Langkawi, Malaysia, 2011, pp. 46-50, doi: 10.1109/CIMSim.2011.18.

[6] Gaussian-Blur: https://www.tutorialspoint.com/opencv/opencv_gaussian_blur.htm

[7] OpenCV Haar-Cascade Classifier: https://docs.opencv.org/3.4/d2/d99/tutorial_js_face_detection.html