Savitribai Phule Pune University

A
Project Report
On

Motion Detection and Multiple Faces Identification

Submitted by

Mr. VARAD KALKHAIRE (B190724251)
Mr. ASHUTOSH KHATAVKAR (B190724262)
Mr. SHUBHAM DHANNE (B190724224)
Mr. SHANTANU BHUMKAR (B190724213)

Under the Guidance of

Prof. Rohini V. Agawane
K J's Educational Institutes
K J College of Engineering and Management Research
Department of Computer Engineering
2022-2023

K J's Educational Institutes
K J College of Engineering and Management Research
Department of Computer Engineering

CERTIFICATE

This is to certify those following students

Mr. VARAD KALKHAIRE (B190724251)
Mr. ASHUTOSH KHATAVKAR (B190724262)
Mr. SHUBHAM DHANNE (B190724224)
Mr. SHANTANU BHUMKAR (B190724213)
Of K J College of Engineering and Management Research has
Submitted the Project Report entitled

"**Motion Detection and Multiple Faces Identification**"

They have satisfactorily completed and submitted Project Report as
Prescribed by Savitribai Phule Pune University for Fourth Year Engineering
(Computer Engineering) for the Academic Year 2022-2023

Place: Pune
Date:

Prof. Rohini V. Agawane **Project Guide**	**Prof. Aparna S. Hambarde** **Project Co-ordinator**	**Prof. Mahesh P. Wankhede** **External Examiner**

Dr. Nikita Kulkarni **Head of Department**	**Dr. Suhas Khot** **Principal**

Acknowledgements

First and foremost, I would like to thank my guide, Prof. **Rohini V. Agawane**, for his/her guidance and support. I will forever remain grateful for the constant support and guidance extended by guide, in making this report. Through our many discussions, she helped me to form and solidify ideas. The invaluable discussions I had with her, the penetrating questions she has put to me and the constant motivation, has all led to the development of this project.

I wish to express my sincere thanks to the Head of department, Prof. **Dr. Nikita Kulkarni**, as well as our Principal **Dr. Suhas Khot,** also grateful thanks to our Project coordinator **Prof. Aparna S. Hambarde** and the departmental staff members for their support.

I am also thankful to my family members for encouragement and support

I would also like to thank to my friends for listening to my ideas, asking questions and providing feedback and suggestions for improving my ideas.

Mr. VARAD KALKHAIRE
Mr. ASHUTOSH KHATAVKAR
Mr. SHUBHAM DHANNE
Mr. SHANTANU BHUMKAR

TABLE OF CONTENTS

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.

The Surveillance Cameras are the currently best-known security due to storing of live video stream into physical drive. By using Machine Learning and Deep Learning models and use of Database, we can make the surveillance camera security more advanced. In this project we built the system "Motion Detection & Multiple Faces Identification using Webcam", were we uses Webcam as Surveillance Camera and Machine Learning models like OpenCV, and Face Recognition for detecting motion and recognizing of multiple faces. For detecting motion of an object, the Gaussian-Blur method is used and for detecting and recognizing multiple faces with help of previously stored Facial data on Database. Haar-Cascade Classifier method is used. For storing Facial data, a MySQL database is used to get efficiency in recognizing faces. This system helps to provide an automated, and advanced security with less human interface by using the system's Webcam.

According to Physics, when an object is motionless and has no speed, then it is considered to be at rest, and just the opposite is when an object is not at complete rest and has some movement or speed in some direction, then it is considered to be in motion. In this article, we will try to detect it.

Motion detection has many real-life implementations or usages where it can prove its worthiness, such as for invigilation of online exams using a webcam as a security guard, etc.

In this project, we will try to implement a script through which we will detect motion using the Web-Camera of the desktop or laptop. The idea is that we will take two frames of the videos and try to find differences between them. If there is some kind of difference between the two frames, then it is clear that there is some kind of movement of an object in front of the camera, which creates the difference. Also, we can use the idea of face detection to detect facial expressions like smile, sad, crying, depression, frustration, etc or detecting multiple faces in one single video frames and recognizing the people names according to their faces stored in database.

Abbreviations

AI	Artificial Intelligence
ML	Machine Learning
CNN	Convolutional Neural Network
RGB	(Red, Green, Blue) Color Combination
GPU	Graphical Processing Unit
CPU	Central Processing Unit

List of Figures

List of Tables

CHAPTER 1:
INTRODUCTION

According to Physics, when an object is motionless and has no speed, then it is considered to be at rest, and just the opposite is when an object is not at complete rest and has some movement or speed in some direction, then it is considered to be in motion. Motion detection has many real-life implementations or usages where it can prove its worthiness, such as for invigilation of online exams using a webcam as a security guard, etc.,

In this project, we will try to implement a script through which we will detect motion using the Web-Camera of the desktop or laptop. The idea is that we will take two frames of the videos and try to find differences between them. If there is some kind of difference between the two frames, then it is clear that there is some kind of movement of an object in front of the camera, which creates the difference. Also, we can use the idea of face detection to detect facial expressions like smile, sad, crying, depression, frustration, etc., or detecting multiple faces in one single video frames and recognizing the people names according to their faces stored in database.

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. For object detection and counting, OpenCV includes a number of useful techniques. 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. The main part in the system is difference between two continuous frames and alert system according to detected motion.

Similarly, Face Recognition in a real-time setting is an exciting area and a rapidly growing challenge 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. 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).

Motion detection and multiple faces identification are two important techniques used in computer vision and surveillance systems. They play a crucial role in various applications such as security monitoring, video analysis, and automated tracking systems. These techniques enable machines to detect and track moving objects, as well as identify and recognize multiple human faces in a given scene.

Motion detection refers to the process of identifying changes in the position of objects over time. It involves analyzing consecutive frames of a video stream or image sequence to detect regions that exhibit significant movement. Motion detection algorithms typically compare pixel intensities between consecutive frames and use thresholding or other mathematical techniques to determine regions of interest. These regions can then be further processed to extract valuable information or trigger specific actions.

There are several methods for motion detection, including background subtraction, optical flow, and frame differencing. Background subtraction involves creating a model of the background scene and subtracting it from the current frame to identify moving objects. Optical flow calculates the apparent motion of pixels between frames, allowing the detection of object movement. Frame differencing compares the pixel values of consecutive frames and highlights

areas with significant differences as potential motion regions.

Multiple faces identification, on the other hand, focuses on the detection and recognition of multiple human faces in an image or video stream. It involves analyzing the characteristics of facial features, such as the shape, texture, and spatial relationships between different parts of the face, to identify and differentiate individuals.

Multiple faces identification techniques often employ machine learning algorithms, specifically deep learning models like convolutional neural networks (CNNs) and recurrent neural networks (RNNs). These models are trained on large datasets of labeled facial images to learn the patterns and representations necessary for accurate face detection and recognition.

The process of multiple faces identification typically involves several steps, including face detection, alignment, feature extraction, and matching. Face detection algorithms locate the presence and location of faces in an image or video frame. Face alignment techniques ensure that the detected faces are correctly aligned to a standardized pose for accurate analysis. Feature extraction methods extract discriminative features from the aligned faces, such as the position of facial landmarks or the values of specific facial descriptors. Finally, face matching algorithms compare the extracted features with a database of known faces to identify and recognize specific individuals.

Motion detection and multiple faces identification have numerous applications, including security systems, surveillance cameras, crowd analysis, human-computer interaction, and biometric authentication. They have greatly advanced the field of computer vision and are continually being improved to enhance accuracy, speed, and robustness in real-world scenarios. These technologies have the potential to significantly improve the efficiency and effectiveness of various industries and contribute to the development of intelligent systems capable of perceiving and understanding the visual world.

Nowadays, security becomes most important factor. When it comes to security everyone never compromises. They will always seek for security to protect their family members, home, company, etc., As technology become most interested topic, we can use technology to provide security. By using technology, we can use Voice Recognition, Fingerprints, Retina Scan, or Surveillance Camera System which can give best security. But the Surveillance Camera Security is the more strong and safest security, where it captures video frames and store them

in hard drive, which can be accessed when there where a security breech.

As technology advances, the newly generated models of Artificial Intelligence and Machine Learning, which gives more accurate result than the traditional one. By using these models, we can provide more accurate, automated, and the more advanced security. The Machine Learning models like OpenCV, and Face Recognition, which are used to detect motion and recognize the faces respectively. By using those models, we can provide the advanced security known as "Motion Detection and Multiple Faces Identification using Webcam" to home or organization. We can also use Surveillance Cameras rather than Webcam for better results. Also, by using these models, we are going to automate the system to detect and recognize faces on its own by taking help of MySQL database.

CHAPTER 2:

LITERATURE SURVEY

M. Khan, S. Chakraborty, R. Astya and S. Khepra. In this paper, the authors used Haar-Cascade Classifier to detect the multiple faces in single frame of a live video stream along, and Eigen Faces Recognizer method for recognition of faces using training dataset present in a hard drive. They use the training dataset in the Eigen Faces Recognizer to recognize the faces in live video stream. The Eigen Faces Recognizer takes lots of images to train the model and then starts to recognize the faces from it, which takes lots of physical storage and requires high performance computer to execute it effectively.

Shubham Mishra, Mrs. Versha Verma, Dr. Nikhat Akhtar, Shivam Chaturvedi, Dr. Yusuf Perwej. This paper studies, the authors compared the first frame from live video stream of Webcam with authors frames using Gaussian-Blur method of an OpenCV. They use Gaussian-Blur method to reduce noise from the frames to make it easier for Image Segmentation and Background Subtraction methods. Image Segmentation method is used to divide the image into segments into its own set of properties with help of Thresholding Segmentation, to find the foreground objects and the background objects. Then Background Subtraction is used to compare background image with current image. The comparison is check with threshold value, were if threshold value is greater then motion is detected else no motion is taken. Here, it requires Threshold for both Image Segmentation and Background Subtraction method, which must be choose correctly and carefully otherwise, will rise an error in the system and overall result.

Ankita Rameshwar Mahajan and Vinod Agrawal. In the cited paper, the authors used Background Subtraction method to detect the motion of an object by comparing current image frame with background image frame, where the Threshold value is taken to compare the images frames for detecting motion in live video stream. Where the Threshold value is much more important and must be choose correctly and carefully to avoid error in detecting the motions.

Diyasa G, Fauzi A, Idhom M, Setiawan A. In this research paper, the authors used Haar-

Cascade Classifier method to detect multiple faces using Haar-like Features, which makes it easier to detect multiple faces by converting colour RGB image (Red, Green, Blue, colour combination of an image) to Grey Level colour. The converted Grey Level images are in the binary value (0 or 1) so it helps to detect the multiple faces efficiently. Also, they used Convolutional Neural Network (CNN) a Deep Neural Network model to recognize the faces by taking the training dataset to train the model and recognize the faces in an image captured by the Webcam on basis of trained model. The CNN uses a lot of images for training, where it reduces the image size by taking kernel for filtering the image and a max pool layer to reduce the dimensions of image. This makes it easier to recognize faces more efficiently than any other method. The CNN is a Deep Neural Network, which is used to process the images and give the result according to the validation dataset and requires lots of computation power and a higher physical storage, high performance computer, and a high-end Graphical Processing Unit (GPU). Lots of computation power is required to run CNN model and only on a image frame not on the live video stream.

R. Rizal Isnanto, Adian Fatchur Rochim, Dania Eridani, Guntur Dwi Cahyono. This study aims to build a face recognition prototype that can recognize multiple face objects within one frame. The proposed method uses a local binary pattern histogram and Haar cascade classifier on low-resolution images. The lowest data resolution used in this study was 76 × 76 pixels and the highest was 156 × 156 pixels. The face images were preprocessed using the histogram equalization and median filtering. The face recognition prototype proposed successfully recognized four face objects in one frame. The results obtained were comparable for local and real-time stream video data for testing. The RR obtained with the local data test was 99.67%, which indicates better performance in recognizing 75 frames for each object, compared to the 92.67% RR for the realtime data stream. In comparison to the results obtained in previous works, it can be concluded that the proposed method yields the highest RR of 99.67%.

Vijay Kumar Sharma. The paper study that Face Recognition is the most popular and trending technology in the present era. It is an effective way to provide vision to a machine for better interaction with humans. The way of living will be reflected if machines can read our faces. The face recognition system will move the world in a new dimension. It will be

beneficial in many ways to find the identity and security. In this paper, a face recognition system is proposed for advanced applications such as access and security, payments, criminal identifications etc. The process of identification will be based on face recognition which is further divided into three steps: detection of face, extractions of the features and classification, and real time recognition. Detection of face is recognized as the essential step of our system. It is used to extract a face in a frame, which is based on the Viola-Jones object detection algorithm that uses AdaBoost classifier with Haar and LBP features. Local Binary Patterns (LBP) is utilized to extract the unique features of the face like eyes, nose, and mouth in the feature extraction phase. The facial image is correlated with the images available in the database for the classification. The system is implemented in Python using OpenCV library. Kivy is used to create a user interface and also to build executables for different platforms.

Abdullah Suleiman Alhaydan, Dr. Mafawez Alharbi. The authors said that the traditional security systems store all what cameras capture, this increases the load on storage devices and makes security employees do large effort while monitoring continuously all what is displayed on screens, or it makes security employees get back later to all video that has been stored and revise them. Therefore, it is necessary to have an application which is capable to set off an alarm when motion happened in front of any camera, this can be implemented by grabbing a frame of a real-time video every short period of time, for example every second, then comparing this frame with a previous frame, if there is any motion in front of the camera, the frames must be different and an event has been happened, then the application can set off an alarm or send an SMS which let the security employees pay attention to the event, as well as the application can store the images which contain the events or start recording video to document this events.

Developing an application that can set off alarm or send email when it detects motion. In real-time video, where each video frame is compared to a previous frame. If there is any motion, the sequent frames must be different. The program takes a picture and convert from RGB to gray to minimize the size then enhance the picture to like in real then show it to the user to see the different.

Anna Liza A.Ramos, Dania May P.Aguila Anne, Catlyne B.Karunungan, Jon-Jon B.Patiño, Vincent L.Polintan. The paper studied that the face makeup is applied to cover

unwanted marked on the face in order to improve a person's appearance. However, this is used in criminal activities since makeup can disguise the true identity of the person. This concern serves as the basis by several studies to discover the technique and applied different methods to optimize the detection and improve accuracy. This study aims to conduct an experiment by applying new methods and techniques in order to provide new results which will increase the scope to consider. Furthermore, this study utilized best performing algorithms: the Haar-Cascade classifier for face detection, Color Feature Extraction for detection of face makeup and Local Binary Pattern Histogram algorithm for extraction and recognition of the features which applied in the experiments considering condition from fluorescent light and sunlight, angle variation of 0 degree, 45% upward and downward and distance of 0.5 meter and 1 meter. The study collected an over-all of 3000 images which will serve as the training datasets. In result, the experiment marked an average accuracy of 88.75% comprises all the methods applied. Specifically, the result showed an average result of both sunlight and fluorescent result with no makeup marked an average score of 90% while the result with makeup marked a score of 87.5%. Moreover, the results showed a higher recognition in near distance of 0.5 meters, the fluorescent light recorded an accuracy result of 90% and 0% degree marked a score of 100%.

Tejas Saraf, Ketan Shukla, Harish Balkhande, Ajinkya Deshmukh. The authors implemented their system for authentication of user to access the automatic door in an organization. They said that the Authentication is one of the significant issues in the era of the information system. Among other things, human face recognition (HFR) is one of the known techniques which can be used for user authentication. As an important branch of biometric verification, HFR has been widely used in many applications, such as video monitoring/surveillance system, human-computer interaction. This project proposes a method for automatic door access system using face recognition technique by using python programming and from OpenCV library Haar cascade method. Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones. This is the standalone security device has been developed by using Raspberry Pi electronic development board and operated on Battery power supply, wireless internet connectivity by using USB modem. Automatic e-mail notification has been achieved by sending security alert mail to the user e-mail id.

CHAPTER 3:

PROPOSED SYSTEM

3.1. BLOCK DIAGRAM

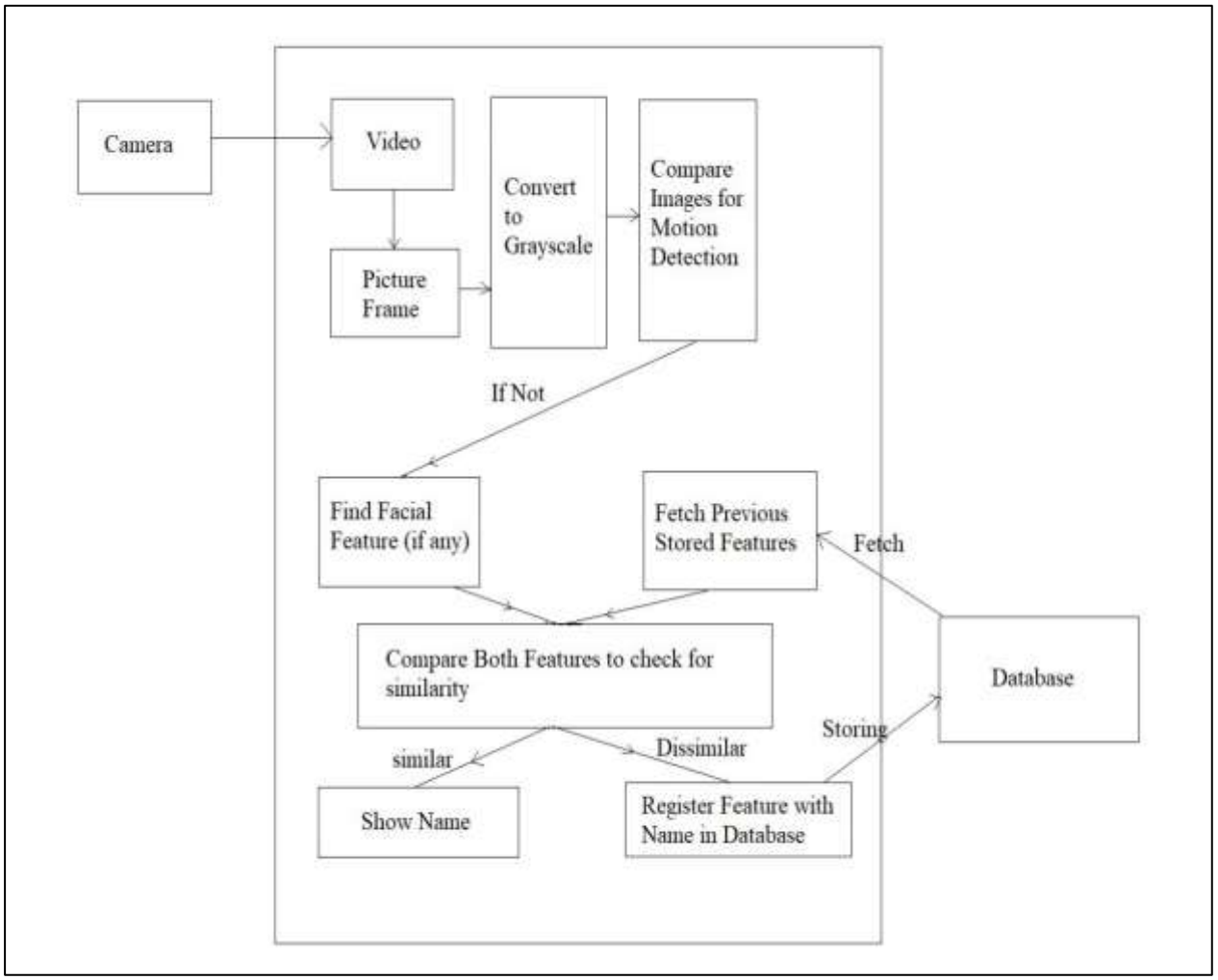

Fig. 3.1. Block Diagram of System

3.1.1 WORKING

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.

3.2. MODULES

3.2.1. Gaussian-Blur

In Gaussian Blur operation, 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. You can perform this operation on an image using the GaussianBlur() method of the imgproc class. Following is the syntax of this method −

GaussianBlur(src, dst, ksize, sigmaX)

This method accepts the following parameters −

src − A Mat object representing the source (input image) for this operation.

dst − A Mat object representing the destination (output image) for this operation.

ksize − A Size object representing the size of the kernel.

sigmaX − A variable of the type double representing the Gaussian kernel standard deviation in X direction.

3.2.2. Haar- Cascade Classifier

Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is

then used to detect objects in other images.

Here we will work with face detection. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. For this, Haar-like features shown in the below image are used. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting sum of pixels under the white rectangle from sum of pixels under the black rectangle.

Now, all possible sizes and locations of each kernel are used to calculate lots of features. (Just imagine how much computation it needs? Even a 24x24 window results over 160000 features). For each feature calculation, we need to find the sum of the pixels under white and black rectangles. To solve this, they introduced the integral image. However large your image, it reduces the calculations for a given pixel to an operation involving just four pixels. Nice, isn't it? It makes things super-fast.

3.2.3 Face Recognition

Facial Recognition is a category of biometric software that maps an individual's facial features mathematically and stores the data as a faceprint. The software uses Deep Learning algorithms to compare a live capture or digital image to the stored faceprint in order to verify an individual's identity.

It uses machine learning algorithms to search for faces within a picture. Faces are very complicated, made of thousands of small patterns and features that must be matched. The face recognition algorithms break the task of identifying the face into thousands of smaller, bite-sized tasks, each of which is easy to solve, known as classifiers.

A face may have 5000 or more classifiers, all of which must match for a face to be detected. Since there are at least 5,000 or more tests per block, you might have millions of calculations to do, which makes it a difficult process. To solve this, OpenCV uses cascades.

The OpenCV cascade breaks the problem of detecting faces into multiple stages. It performs a detailed test for each block. The algorithm may have 30 to 50 of these stages or cascades, and it will only detect a face if all stages pass.

The cascades are a bunch of XML files that contain OpenCV data used to detect objects. You initialize your code with the cascade you want, and then it does the work for you. Since face detection is such a common case, OpenCV comes with a number of built-in cascades for detecting everything from faces to eyes to hands to legs.

3.3. ALGORITHM/ STEPS

Step 1: Import Libraries:

- Import the necessary libraries, including OpenCV, Haar-Cascade Classifier, and Face Recognition.

Step 2: Initialize Webcam:

- Initialize the webcam using OpenCV and set up parameters such as resolution and frame rate.

Step 3: Motion Detection:

- Capture the current frame from the webcam.
- Convert the frame to grayscale for efficient processing.
- Calculate the absolute difference between the current frame and the previous frame to detect motion.
- Apply thresholding or other filtering techniques to distinguish significant motion regions from noise.
- Extract motion regions of interest (ROIs) or bounding boxes for further analysis or tracking.

Step 4: Face Detection:

- Load a pre-trained Haar-Cascade classifier for face detection.
- Apply the classifier to detect faces in the captured frame.
- Obtain the coordinates or bounding boxes of the detected faces.

Step 5: Multiple Faces Identification:

- Load a pre-trained face recognition model or create a face recognition instance.
- Align the detected faces to a standardized pose for accurate analysis.
- Extract facial embeddings or descriptors from the aligned faces using the face recognition model.
- Compare the extracted embeddings with a database of known face embeddings using the face recognition model.
- Assign identities to the matched faces based on the database information.

Step 6: Visualization and Output:

- Draw bounding boxes or overlays around the detected faces and motion regions on the video frame using OpenCV.
- Display the processed frame with the overlays on a screen or user interface using OpenCV.
- Optionally, trigger actions or alerts based on the detected motion or identified faces.

Step 7: Loop and Continuously Process Frames:

- Repeat steps 3-6 for each subsequent frame captured from the webcam to enable real-time monitoring and analysis.
- Update the previous frame to the current frame for motion detection in the next iteration.

CHAPTER 4:

SOFTWARE REQUIREMENT SPECIFICATION

4.1 PROBLEM STATEMENT

Developing an efficient and robust system for motion detection and multiple faces identification in real-time video streams.

Description:

Motion detection and multiple faces identification are essential components of modern surveillance and computer vision systems. However, there are still several challenges that need to be addressed to develop a system that can accurately detect and track moving objects while identifying multiple human faces in a given scene. The following problem statement outlines the key challenges:

1. **Motion Detection:**

- Designing algorithms that can accurately detect and distinguish between relevant motion (e.g., objects of interest) and irrelevant motion (e.g., background noise, lighting changes).
- Developing techniques that can handle various environmental conditions, such as low lighting, shadows, occlusions, and complex backgrounds.
- Ensuring real-time performance and efficiency, especially when processing high-resolution video streams or multiple camera feeds simultaneously.
- Minimizing false positives and false negatives by optimizing thresholding techniques and incorporating advanced filtering mechanisms.

2. **Multiple Faces Identification:**

- Designing robust face detection algorithms that can accurately locate multiple faces in different poses, orientations, and scales.
- Developing methods to handle occlusions, variations in facial expressions, and changes in lighting conditions.

- Extracting discriminative facial features that can reliably represent individuals, while considering variations in appearance and capturing key identity cues.
- Developing efficient matching algorithms that can accurately compare the extracted features against a large database of known faces.
- Ensuring scalability and real-time performance, especially in scenarios with a large number of faces or multiple camera feeds.

3. **Integration and System-level Challenges:**

- Developing an integrated system that seamlessly combines motion detection and multiple faces identification to provide comprehensive surveillance and analysis capabilities.
- Addressing computational and resource constraints to ensure real-time performance on various hardware platforms.
- Developing user-friendly interfaces and visualization tools to present the detected motion and identified faces in a clear and intuitive manner.
- Evaluating the system's performance under different real-world scenarios and benchmarking against existing state-of-the-art methods.
- Solving these challenges will contribute to the development of robust and efficient systems for motion detection and multiple faces identification, enabling enhanced security, video analysis, and automated tracking capabilities in various domains, including surveillance, biometric authentication, and human-computer interaction.

4.2 EXTERNAL INTERFACE REQUIREMENT

4.2.1. User Interface

Application-Based “Motion Detection & Multiple Faces Identification using Webcam”.

4.2.2. Hardware Interfaces:

RAM: 8GB (min)

Storage: 500 GB (HDD/SSD)

Processor: Intel i5 (or above) or AMD Ryzen 5 (or above)

Camera: System’s Web-Camera

4.2.3. Software Interfaces

Operating System: Windows 10 (or above) or Ubuntu

IDE: Visual Studio

Programming Languages: Python, MySQL

4.3 NON-FUNCTIONAL REQUIREMENT

4.3.1. Performance Requirements

The performance of the functions and every module must be well. The overall performance of the software will enable the users to work efficiently. Performance of encryption of data should be fast. Performance of the providing virtual environment should be fast.

4.3.2. Safety Requirement

The application is designed in modules where errors can be detected and fixed easily.This makes it easier to install and update new functionality if required.

4.3.3. Software Quality Attributes

Our software has many qualities attribute that are given below: -

1. **Adaptability**: This software is adaptable by all users.
2. **Availability**: This software is freely available to all users. The availability of the software is easy for everyone.
3. **Maintainability**: After the deployment of the project if any error occurs then it can be easily maintained by the software developer.
4. **Reliability**: The performance of the software is better which will increase the reliability of the Software.
5. **User Friendliness**: Since, the software is a GUI application; the output generated is much user friendly in its behavior.
6. **Integrity**: Integrity refers to the extent to which access to software or data by unauthorized persons can be controlled.
7. **Security**: Users are authenticated using many security phases so reliable security is provided.
8. **Testability**: The software will be tested considering all the aspects.

CHAPTER 5:

DESIGN

5.1. DATA FLOW DIAGRAM

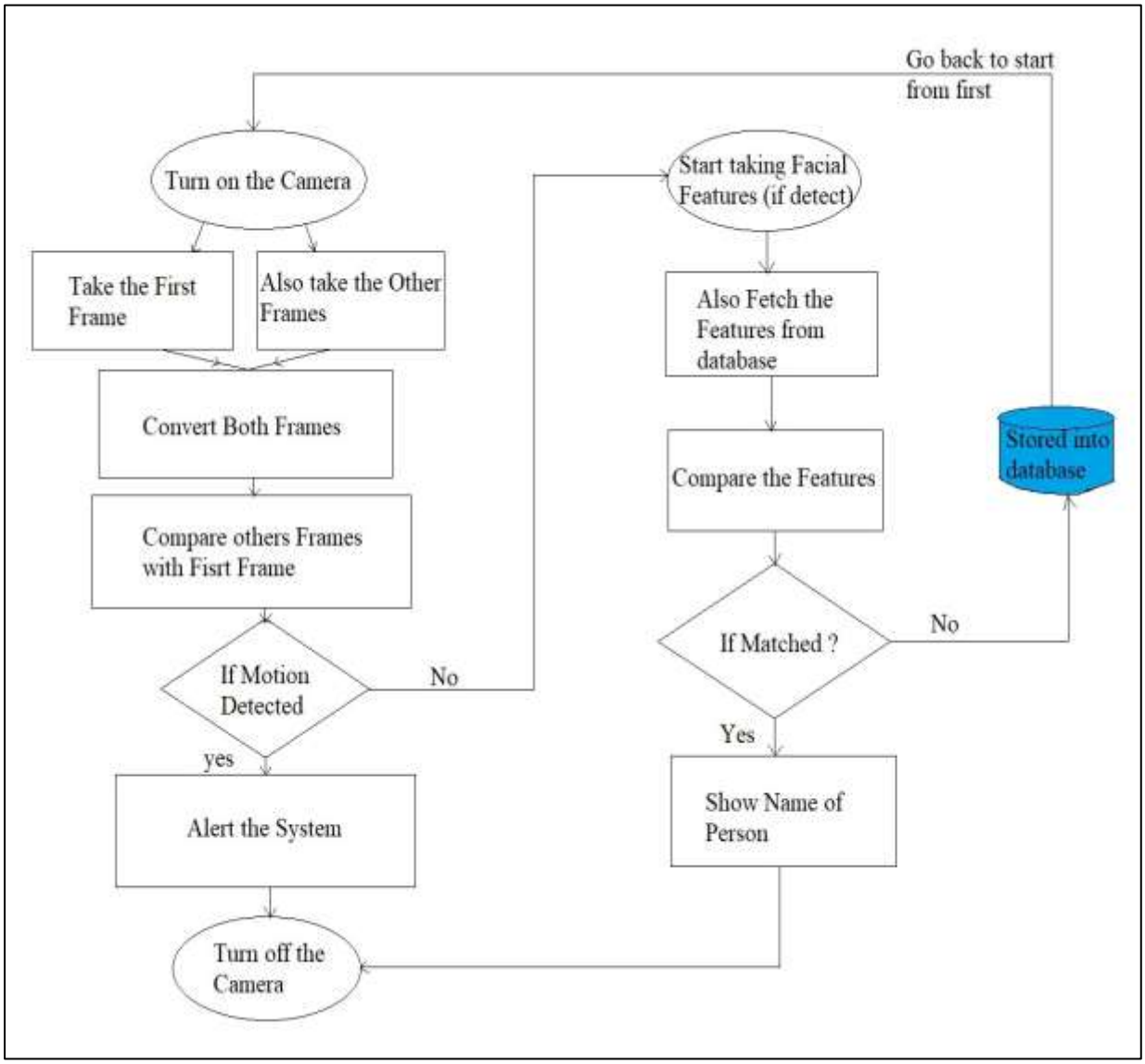

Fig. 5.1. Data Flow Diagram

5.2. USE CASE DIAGRAM

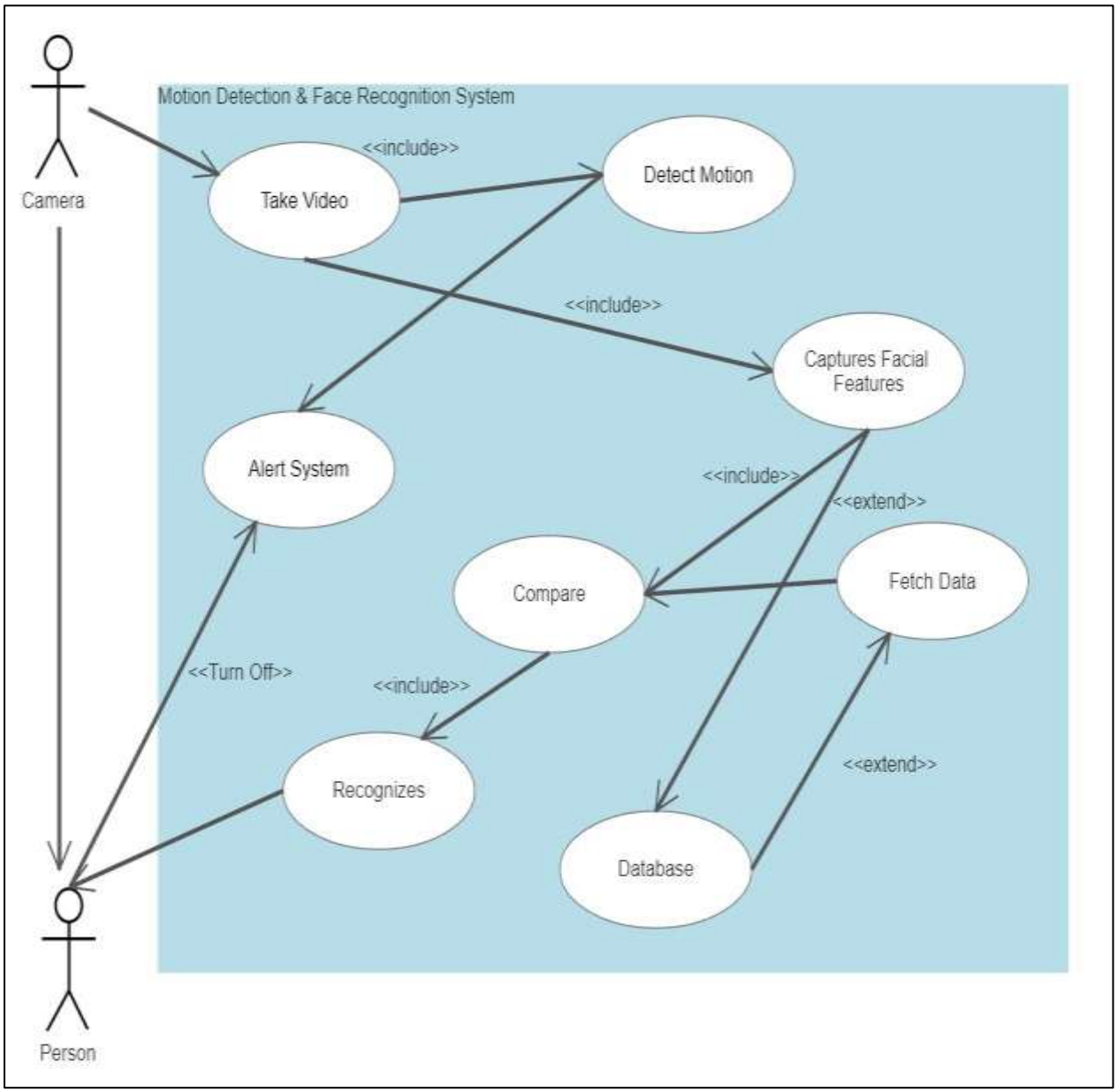

Fig. 5.2. Use Case Diagram

5.3. SEQUENCE DIAGRAM

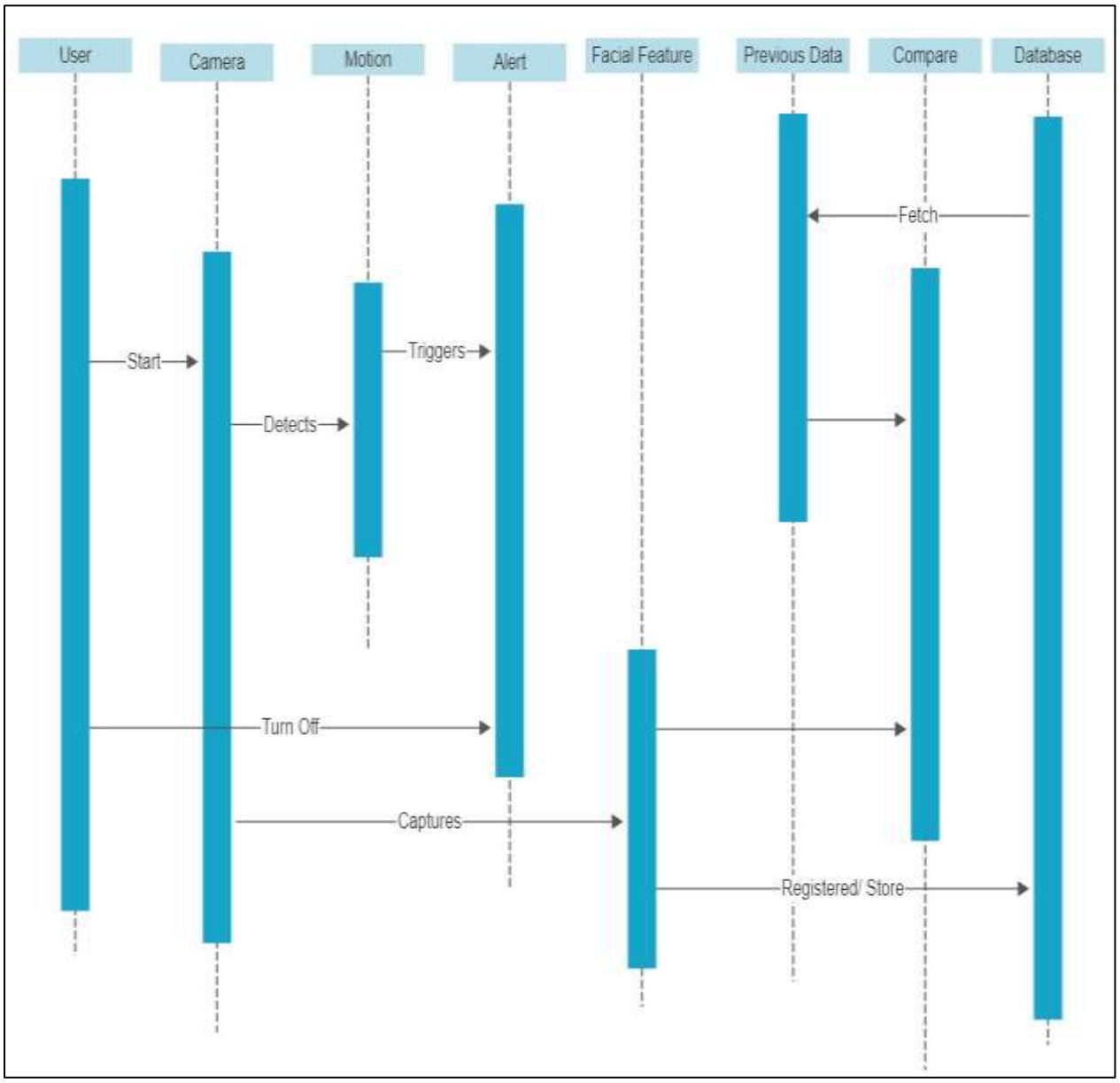

Fig. 5.3. Sequence Diagram

5.4. ARCHITECTURE DIAGRAM

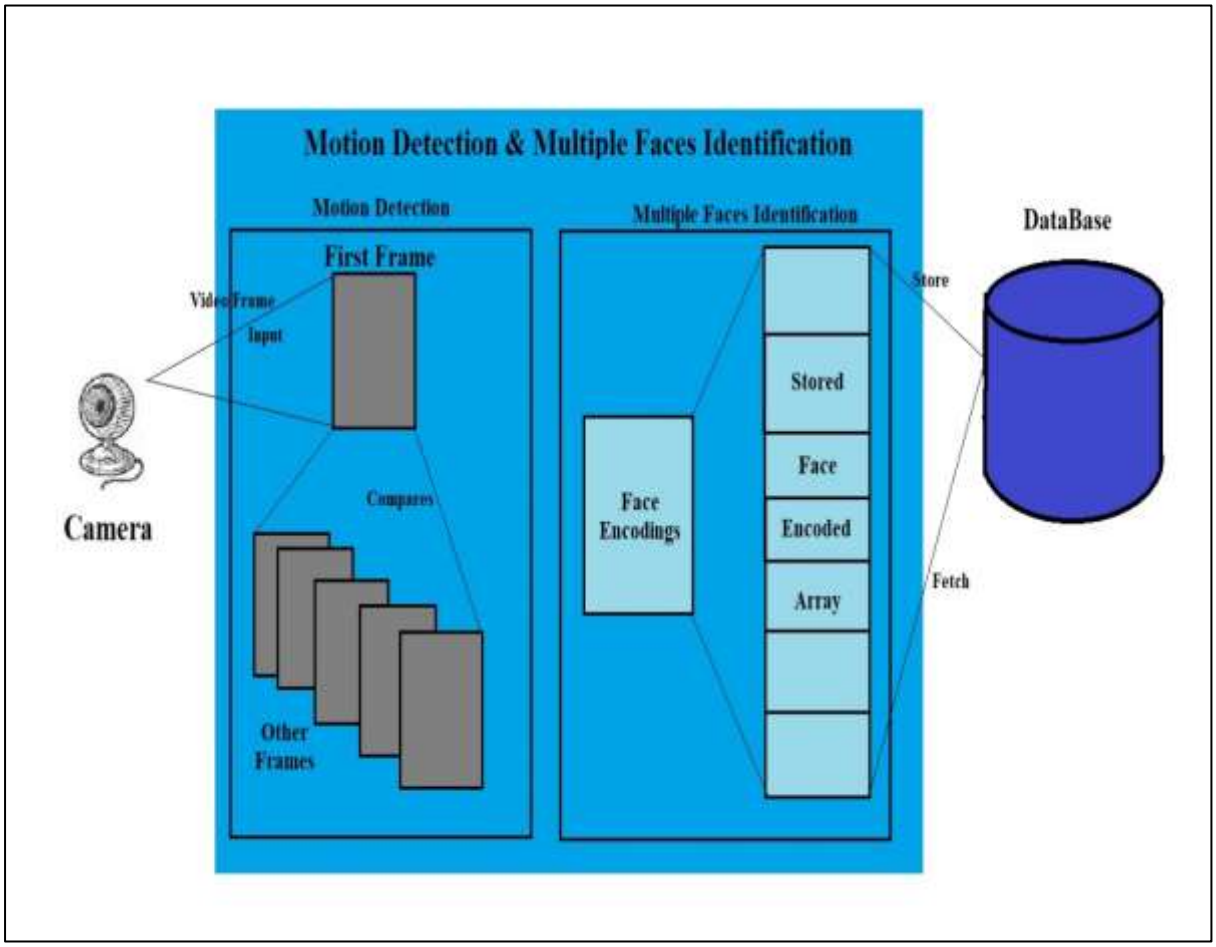

Fig. 5.4. System Architecture

5.5. ER DIAGRAM

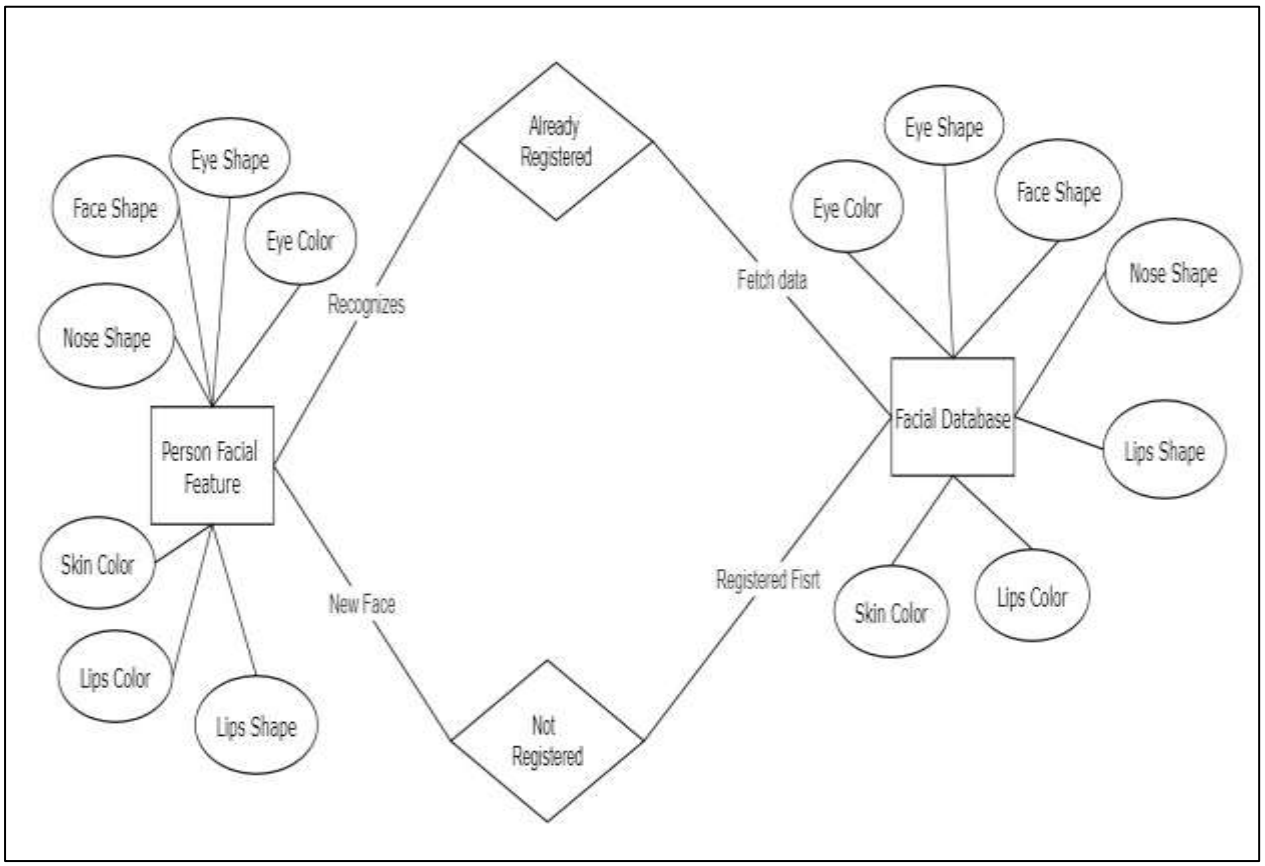

Fig. 5.5. ER Diagram

CHAPTER 6:

ADVANTAGES & DISADVANTAGES

6.1. ADVANTAGES

1. **Enhanced Security**: Motion detection allows for the automatic detection of any movement or activity in the monitored area, enabling immediate alerts and response to potential security threats. Multiple faces identification adds an additional layer of security by recognizing and identifying specific individuals.

2. **Real-Time Monitoring**: Webcam-based motion detection and multiple faces identification systems provide real-time monitoring of the captured video streams, allowing for immediate action or intervention when needed.

3. **Cost-Effectiveness**: Utilizing webcams for motion detection and multiple faces identification is a cost-effective solution compared to dedicated surveillance hardware. Webcams are readily available, affordable, and easy to install, making them accessible for both personal and small-scale commercial applications.

4. **Convenience and Accessibility**: Webcams are commonly integrated into laptops, desktop computers, and mobile devices, making them easily accessible for various applications. They can be easily positioned and adjusted, providing flexibility in monitoring specific areas of interest.

5. **Non-Intrusive**: Webcams are non-intrusive in nature and do not require physical contact or interaction with the subjects being monitored. This makes them suitable for applications where privacy or discretion is a concern, such as in public spaces or personal environments.

6.2. DISADVANTAGES

1. **Limited Field of View**: Webcams typically have a fixed field of view, which may limit the coverage area for motion detection and face identification. This can be a challenge when monitoring larger spaces or areas that require wide coverage.

2. **Image Quality and Resolution**: The image quality and resolution of webcams may vary, and they may not always capture clear and detailed images. This can impact the accuracy and reliability of motion detection and face identification algorithms, especially in low-light conditions or when dealing with small or distant objects.

3. **Processing Limitations**: Webcams often have limited processing capabilities, which can affect the performance of motion detection and face identification algorithms. Complex algorithms and real-time processing may require additional computational resources or dedicated hardware accelerators to achieve optimal performance.

4. **Sensitivity to Environmental Factors**: Webcams are sensitive to environmental factors such as lighting conditions, shadows, reflections, and noise. These factors can introduce false positives or false negatives in motion detection and face identification, reducing the overall accuracy and reliability of the system.

5. **Privacy Concerns**: As motion detection and multiple faces identification involve capturing and analyzing video footage, privacy concerns may arise. Proper measures should be implemented to ensure compliance with privacy regulations and protect the privacy of individuals being monitored.

CHAPTER 7:
APPLICATIONS

7.1 APPLICATIONS OF THE PROJECT:

Motion detection and multiple faces identification have a wide range of applications across various industries. Here are some key areas where these technologies are commonly applied:

1. **Security and Surveillance**: Motion detection and multiple faces identification are extensively used in security and surveillance systems. They help detect suspicious activities, monitor restricted areas, and identify individuals for access control purposes.

2. **Home Automation**: Motion detection is employed in smart home automation systems to trigger actions based on movement. For example, lights can be automatically turned on when someone enters a room, or security cameras can be activated when motion is detected outside the house.

3. **Traffic Monitoring and Management**: Motion detection is crucial in traffic monitoring and management systems. It helps detect vehicle movement, identify congestion points, and optimize traffic flow. Multiple faces identification can also be used in intelligent transportation systems for license plate recognition and identifying traffic violators.

4. **Gaming and Virtual Reality**: Motion detection is widely used in gaming and virtual reality applications. It enables users to interact with the virtual environment through body movements and gestures, providing a more immersive and engaging experience.

5. **Healthcare and Assisted Living**: Motion detection and multiple faces identification technologies have applications in healthcare settings. They can be used for patient monitoring, fall detection, activity tracking, and identifying individuals with specific health conditions. These technologies are particularly useful in assisted living facilities and home healthcare.

6. **Retail and Marketing**: Multiple faces identification is employed in retail and marketing environments to gather demographic information about customers, track customer behavior, and personalize marketing campaigns. It enables targeted advertisements, product recommendations, and enhanced customer experiences.

7. **Human-Computer Interaction**: Motion detection and multiple faces identification are utilized in human-computer interaction systems. They allow users to control devices and interfaces through gestures, facial expressions, and body movements, eliminating the need for physical input devices.

8. **Augmented Reality**: Motion detection and multiple faces identification technologies are utilized in augmented reality applications. They enable virtual objects to interact with the real world based on user movements and gestures, creating immersive and interactive augmented reality experiences.

9. **Sports Analysis**: Motion detection is widely used in sports analysis for tracking and analyzing the movements of athletes. It helps in performance evaluation, injury prevention, and strategic decision-making in sports such as soccer, basketball, and athletics.

10. **Crowd Management**: Motion detection and multiple faces identification technologies are employed in crowd management systems for events, public spaces, and transportation hubs. They help in monitoring crowd movements, identifying overcrowded areas, and ensuring public safety.

These are just a few examples of the diverse applications of motion detection and multiple faces identification technologies. As the technologies continue to advance, they are likely to find new and innovative applications across various industries.

CHAPTER 8:

RESULT

7.1 SCREEN SHOT

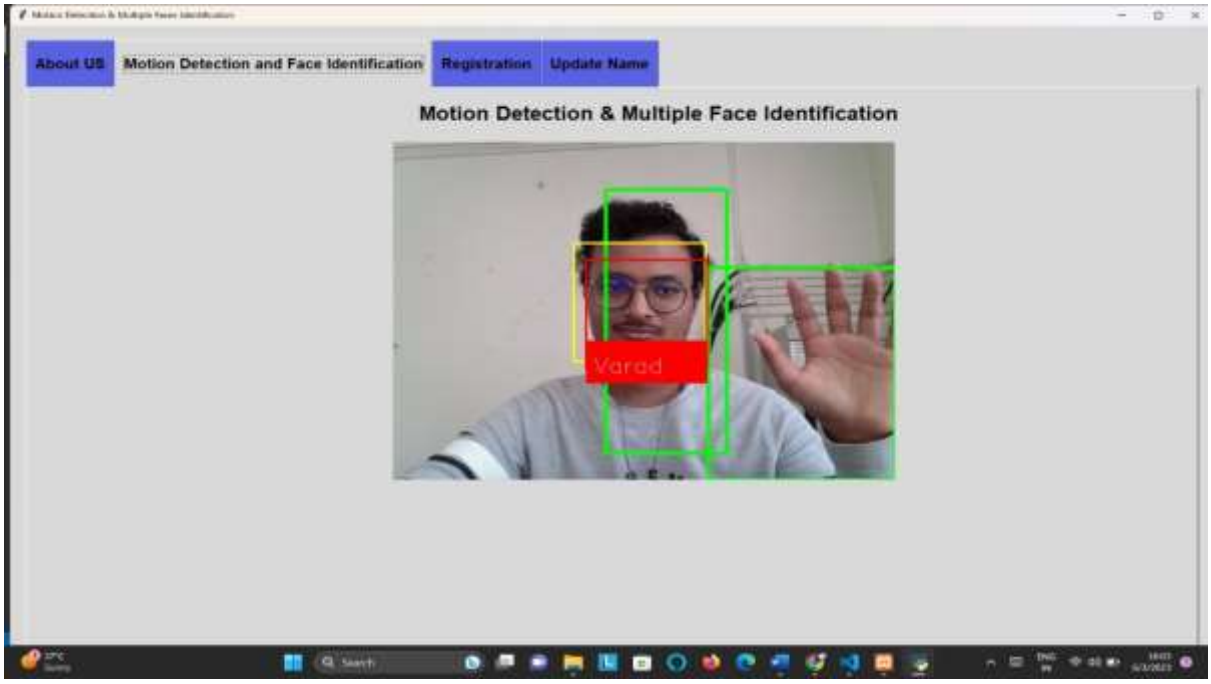

Fig. 7.1. Motion Detection and Face Identification

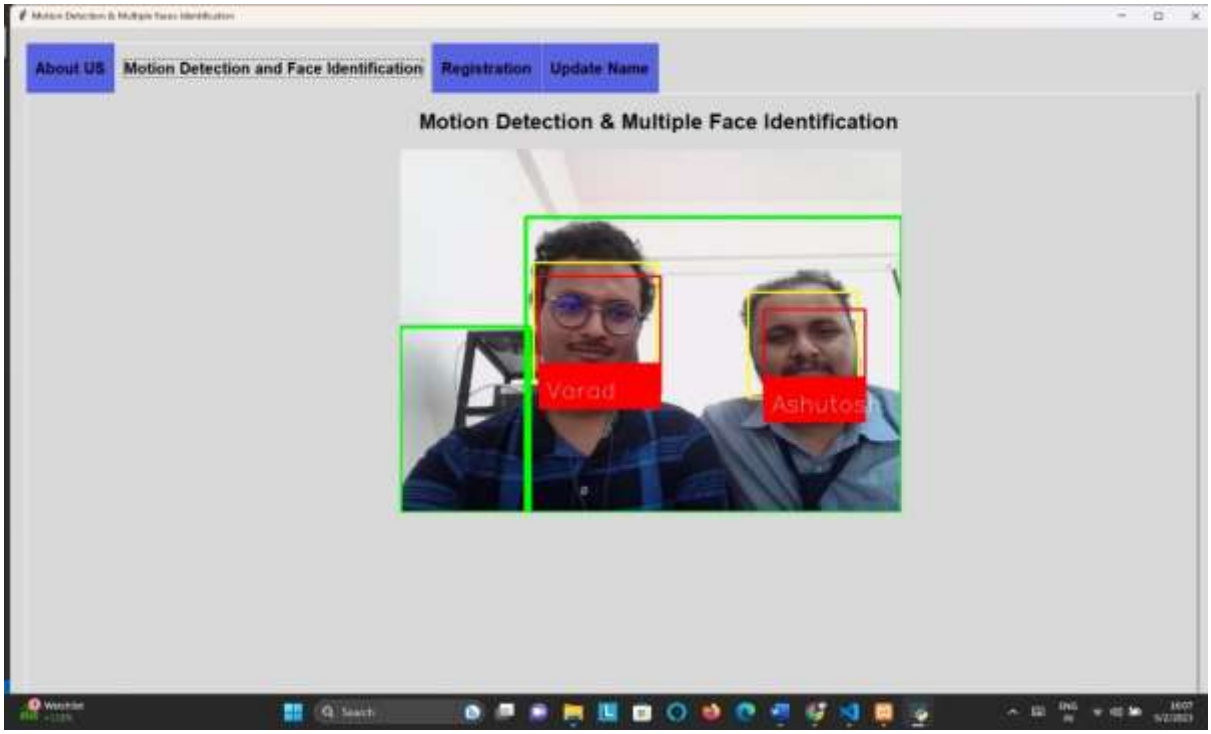

Fig. 7.2. Motion Detection and Multiple Faces Identification

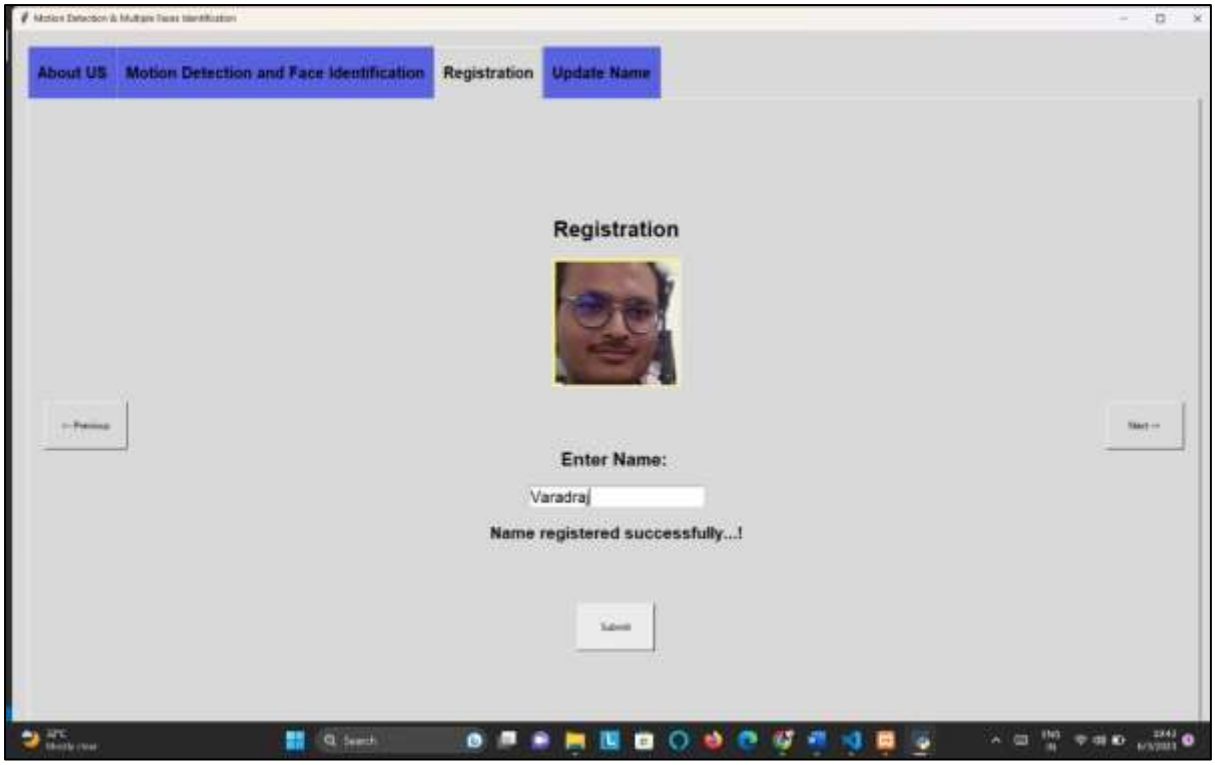

Fig. 7.3. Face Registration

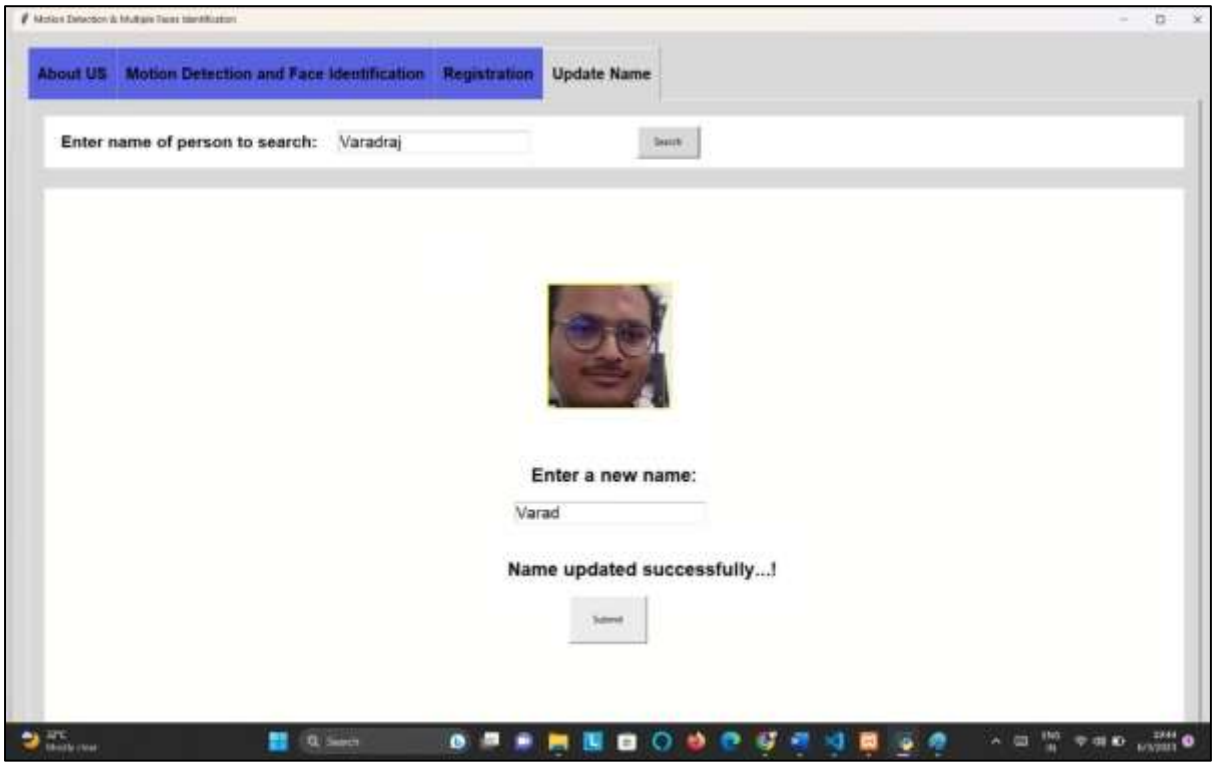

Fig. 7.4. Face Name Upgardation

7.2 RESULT

In above Figure 7.1 and 7.2, 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 faster execution of motion detection model possible 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.

The Figure 7.3 shows, the registration of newly detected face by the Haar-Cascade Classifier model. It shows the newly detected faces stored in the local storage and from the unknown table too. This helps to user to register a name of newly detected faces into the known table of database. Also, it checks whether the person's name is unique or not i.e., if person's name is presented in the database, then show message as "Name already present in Database. Enter their name with surname or nickname". Else, registered the face along with face encodings from unknown table. Afterward, delete the data of registered person from unknown table.

The Figure 7.4 shows, the upgradation of person's new name into the known table. In this, the person whose name need to update must be searched into known table to check whether name is present in database or not. If not then show message as "Name not present in database and enter a valid name" or show the person's face to enter newly updated name. This helps user to manipulate the model easily and more efficiently.

Overall, the project is built to provide the advanced security by taking use of Surveillance Camera and automated the face recognition and motion detection model using the Machine

Learning and Deep learning models. The “Motion Detection and Multiple Faces Identification using Webcam” uses Deep Learning models like Gaussian-Blur, Haar-Cascade Classifier, and Face Recognition which helps to make this system more automated and advanced. Thus, using such Deep Learning models we can provide the advanced security to Home or to an organization.

CHAPTER 9:

TEST CASES:

8.1 TEST CASES FOR DATABASE:

TC_ID	Test Case Objective	Pre-requisites	Steps	Input	Expected Result	Actual Result	Status
TC_01	To check the database is accessing or not	“imaged” database should be available	Connect to database using MySQL connector	(‘localhost”, ”root”, “”, “ImageDB”)	Database must be accessed	Database connection accessed	Pass
TC_02	Inserting data into unknown table	Unknown table should be available	Add face encodings and image file name into unknown table	(face_enc, Img_name)	Data must be inserted	Data inserted successfully	Pass
TC_03	Fetching data from unknown table	Unknown table should be available	Fetch data from unknown table using MySQL DML select query	Select face_enc and name from unknown	Data should be fetched	Data fetched successfully	Pass
TC_04	Inserting data into known table	Known table should be available	Add face encoding from unknown table and person’s name	(face_enc, name)	Data must be inserted	Data insert successfully	Pass
TC_05	After successfully insertion of data in known table data from unknown table should be deleted	Known and unknown table should be available	Delete data from unknown table after successful insertion of data from unknown to known table	Delete all from unknown where img_name = current_img_name	Data should be deleted	Data deleted successful	Pass
TC_06	Fetching data from known table	Known table should be available	Fetch data from known table using MySQL DML select query	Select face_enc and person’s name from known	Data should be fetched	Data fetched successfully	Pass

Table 8.1. Test Cases for Database

8.2 TEST CASES FOR THE WEBCAM:

TC_ID	Test Case Objective	Pre-requisites	Steps	Input	Expected Result	Actual Result	Status
TC_01	On start of the model the Webcam should be accessed	System's Webcam should be available	Access the system Webcam when model start	NA	System's Webcam must be accessed	System's Webcam accessed successfully	Pass
TC_02	Webcam should capture the live video stream	Webcam should be available and must be accessible	Start live video stream using Webcam	NA	Live video stream should be started	Live video stream had started	Pass
TC_03	Webcam live video stream should stop on model termination	Webcam should be available and must be accessible	Stop live video stream on model termination	NA	Live video stream should be stop on model termination	Live video stream had stopped	Pass

Table 8.2. Test Cases for Webcam

8.3 TEST CASES FOR MODEL:

TC_ID	Test Case Objective	Pre-requisites	Steps	Input	Expected Result	Actual Result	Status
TC_01	Model should be started	Model must be available with code	Start model	NA	Model should be start	Model had started	Pass
TC_02	Fetch data from known and unknown tables	Known and Unknown table should be available in ImageDB database	Fetch data from known and unknown table and store in local variables	MySQL DML select query for fetching data from tables	Data should be fetched	Data fetched successfully	Pass
TC_03	Start live video stream using System's Webcam	System's Webcam should be available and must be accessible	Start live video stream using Webcam	NA	Live video stream should start	Live video stream had started	Pass
TC_04	Access video stream into frame	System's Webcam should be available and must be accessible	Accept frames of live video stream	Webcam's live video stream	Frames should be accepted using live video stream	Frames was accepted using live video stream	Pass
TC_05	Convert RGB frames into grey level frames	Frames should be available	Conversion of frames from RGB color to grey scale level color	RGB frames	Conversion should be done	Conversion done successfully	Pass
TC_06	Start detecting the Motion from Grey level frames	Grey level frame should be available	Start detecting motion using grey level converted frames	Grey Level frames	Motion should be detected	Motion is detected	Pass
TC_07	Start Haar-Cascade Model	Haar-Cascade model should be available	Start the Haar-Cascade model for detecting faces	Live video stream from Webcam	Haar-Cascade Classifier model should be started	Haar-Cascade Classifier had started	Pass
TC_08	Start detecting the multiple	Haar-Cascade model should be started	Start detecting	Harr-Cascade model and	Multiple faces must be detected	Multiple faces are detected	Pass

	faces using Haar-Cascade Model		multiple faces	live video stream from Webcam			
TC_09	Store the newly detected face in unknown table and also in local drive	Unknown table and physical drive should be available	Store newly detected face in database and in local storage	Newly detected face from Haar-Cascade Classifier model	Newly detected face must be stored in local storage as well as in unknown table	Newly detected face stored in local storage as well as in unknown table successfully	Pass
TC_10	Start the Face Recognition model	Face Recognition model should be available	Start the Face Recognition model	Live video stream from Webcam	Face Recognition model should start	Face Recognition model had started	Pass
TC_11	Access the known face encodings fetched from known table from database	Known face encodings list and known table should be accessed	Access the known face encodings list	Known table face encodings	List of known face encodings should be access	List of known face encodings had accessed	Pass
TC_12	Start recognition of multiple faces	Known face encodings list, multiple faces detected by Haar-Cascade Classifier and live video stream using Webcam	Start recognizing of multiple faces	Known face encodings list, multiple faces detected by Haar-Cascade Classifier model and Live video stream from Webcam	Multiple Faces Recognition must be done by the model	The model has recognized multiple faces	Pass
TC_13	Update detected movement time in csv file and terminate model	Time of movement.csv file	Update time of detected movement by Gaussian-Blur method	Start and end time of movement	Time of detected movement should be updated and model has Terminated too.	Time of movement detected by Gaussian-Blur model has updated and Model has	Pass

						terminated too.	
TC_14	Add the newly Detected face	Newly detected face file, textbox to entry name should be available	Add name to newly detected face	Enter name in textbox	Name of newly detected face should be entered	Name of newly detected face had entered	Pass
TC_15	Check whether the enter name is present in database	Known table should be available	Check whether name is duplicate	Name entered in textbox	If available show message "Name already present in database. Enter their name with surname or nickname"	Message shown as "name already present in database. Enter their name with surname or nickname"	Pass
TC_16	Register name with face encodings in known table if name is unique and show message	Known table, face encodings of newly detected face from unknown table and message label	Registered name with face encodings in the known table	Newly detected face encodings from unknown table and unique name	Insert newly detected face encodings with their name and show message "Name registered successfully"	Newly detected face encodings with their name registered and message shown as "Name registered successfully"	Pass
TC_17	Enter name to search in database for updating name	Name textbox to search name in table and known table	Enter name to search in the known table	Person's name	Person's name should be entered	Person's name has entered	Pass
TC_18	is name not present in database then show message	Message label, person's name and known table	Check whether name is present in database	Person's name	If not available show message "Name not present in database. Enter a valid name"	Message shown as "Name not present in database. Enter a valid name" for invalid name	Pass
TC_19	If present show the face to enter the updated name	Face label and new name textbox	Show face of search name to update his/her name	Person's name	Name present in database and accept the new update	Updated name of person has accepted by the model	Pass

					name of person		
TC_20	Check whether name is present in database and show message	Known table, message label and person's updated name	Check whether name is duplicate	Person's updated name	If present then show message "Name already present in database. Enter their name with surname or nickname"	Message shown for present name in database as "Name already present in database. Enter their name with surname or nickname"	Pass
TC_21	If not update name in database and show message	Known table, message label and person's updated name	Update person's unique name in known table	Person's updated name	Update person's new name using MySQL DML update query and show message as "Name updated successfully"	Message shown as "Name updated successfully" for uniquely updated person's name	Pass

Table 8.3. Test Case for Model

CHAPTER 10:

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.

Motion detection and multiple faces identification using webcams offer numerous advantages and opportunities for enhanced security, real-time monitoring, cost-effectiveness, convenience, and accessibility. These techniques leverage the capabilities of webcams to detect and track motion in a monitored area and identify multiple human faces in a video stream. However, there are also certain limitations and challenges that need to be considered.

Webcams may have a limited field of view and varying image quality and resolution, which can impact the accuracy and reliability of the systems. Processing limitations of webcams can affect the performance of motion detection and face identification algorithms, requiring additional computational resources for optimal results. Moreover, webcams are sensitive to environmental factors such as lighting conditions, shadows, reflections, and noise, which can introduce false positives or false negatives.

Despite these limitations, motion detection and multiple faces identification using webcams can be an effective and accessible solution for various applications. They provide an affordable and non-intrusive means of monitoring and enhancing security in both personal and small-

scale commercial settings. With proper consideration of privacy concerns and the implementation of suitable measures, these systems can be deployed to improve surveillance, biometric authentication, and human-computer interaction.

Continued advancements in computer vision algorithms, hardware capabilities, and image processing techniques can further enhance the accuracy, efficiency, and reliability of motion detection and multiple faces identification using webcams. As technology evolves, these systems have the potential to become even more robust and capable, enabling advanced functionalities and contributing to the development of intelligent visual monitoring solutions.

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.

This project can be used in future application like:

1. **Improved Accuracy and Robustness**: Future advancements in computer vision algorithms, such as deep learning models, can lead to improved accuracy and robustness in motion detection and face identification using webcams. More sophisticated techniques can handle challenging scenarios, such as complex backgrounds, occlusions, and varying lighting conditions, resulting in more reliable and precise results.

2. **Integration with IoT and Edge Computing**: The integration of motion detection and multiple faces identification with Internet of Things (IoT) devices and edge computing platforms can enable distributed and decentralized surveillance systems. This integration can enhance scalability, reduce latency, and allow for real-time analysis and response in diverse environments.

3. **Multi-Camera Systems**: Future developments may involve utilizing multiple webcams

or camera arrays for motion detection and multiple faces identification. This can provide a broader field of view, improved coverage, and more accurate tracking capabilities. The fusion of information from multiple cameras can enhance the overall performance of the system.

4. **Enhanced Privacy and Ethical Considerations**: As privacy concerns become increasingly important, future research may focus on developing privacy-preserving techniques for motion detection and face identification. This could involve the use of advanced encryption methods, anonymization of data, or the development of algorithms that extract relevant information while minimizing the risk of privacy breaches.

5. **Real-Time Action and Decision Making**: Advancements in machine learning and AI algorithms can enable motion detection and face identification systems to make real-time decisions and trigger automated actions. For example, the system can automatically alert security personnel, adjust surveillance parameters, or initiate specific responses based on detected motion or identified individuals.

6. **Behavioral Analysis**: Future research may involve incorporating behavioral analysis into motion detection and multiple faces identification systems. By analyzing the movement patterns and behaviors of individuals, it becomes possible to detect suspicious or abnormal activities, leading to more proactive and intelligent surveillance systems.

7. **Augmented Reality and Virtual Reality Integration**: Integration with augmented reality (AR) and virtual reality (VR) technologies can open up new possibilities for motion detection and face identification. This integration can enable immersive experiences, interactive monitoring, and enhanced situational awareness for security and surveillance applications.

8. **Cross-Domain Applications**: Motion detection and multiple faces identification using webcams can find applications beyond traditional security and surveillance. They can be integrated into various domains, such as retail analytics, smart homes, healthcare, and human-computer interaction, to enhance user experiences, improve efficiency, and enable personalized services.

Overall, the future of motion detection and multiple faces identification using webcams holds great potential for advancements in accuracy, privacy, real-time decision-making, and integration with emerging technologies. Continued research and development in these areas will contribute to the evolution of intelligent surveillance systems and open up new opportunities for a wide range of applications.

CHAPTER 11:
REFERENCES

[1] Recognition using OpenCV" in Institution of Electrical & Electronics Engineers, paper presentation, 2019.

[2] Shizhen Huang, Haonan Luo, "Attendance System Based on dynamic face Detection" in Institution of Electrical & Electronics Engineers, paper presentation, 2020.

[3] Chen Zhaoyang, Gao Haolin, Wang Kun , "A Motion-Based Object Detection Method," in Institution of Electrical & Electronics Engineers, paper presentation, 2020.

[4] Stan Z. Li, Juwei Lu, "Generalizing Capacity of Face Database for Face Recognition," in Institution of Electrical & Electronics Engineers, paper presentation, 2020.

[5] Jean-Christophe Terrillon, Arnaud Pilpré, Yoshinori Niwa and Kazuhiko Yamamoto, "A Real-Time System for Robust Multiple Face Detection, Tracking and Hand Posture Recognition in Color Video sequences", in Institution of Electrical & Electronics Engineers, paper presentation, 2020.

[6] "Python (Programming Language)", https://en.wikipedia.org/wiki/Python_(programming_language), article on Python Programming Language, 2020.

[7] "Gaussian-Blur Python Module", https://www.tutorialspoint.com/opencv/opencv_gaussian_blur.htm , article on Gaussian-Blur Python Module, 2020.

[8] "Cascade-Classifier module in Python", https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html , article on Cascade-Classifier module in Python OpenCV, 2004.

[9] Vijay Kumar Sharma, "Designing of Face Recognition System", in Institution of Electrical & Electronics Engineers, paper presentation, 2019.

[10] Nikhil Singh, Praveen Kumar, Priya Akhoury, Rohit Kumar, M. Ramasubramanian, "Motion Detection Application Using Web Camera", in International Journal of Modern Engineering Research (IJMER), paper presentation, 2019.

[11] S. V. Tathe, A. S. Narote, S. P. Narote, "Human Face Detection and Recognition in Videos", in Intl. Conference on Advances in Computing, Communications and Informatics (ICACCI), paper presentation, 2016.

[12] Chunxia Yang, Yunyang Yan and Quanyin Zhu, "The Face Database Development of Science and Technology Expects Based on Web Mining", in Institution of Electrical & Electronics Engineers, paper presentation, 2012.

[13] Yidong Ma, "Effective methods for lightweight image-based and video-based face recognition", in Institution of Electrical & Electronics Engineers, paper presentation, 2016.