International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 4
DOI: https://doi.org/10.17762/ijritcc.v11i4.6452
***Article Received**: 02 February 2023 **Revised**: 12 March 2023 **Accepted**: 28 March 2023*

Dual-Level Secured Autonomous Bank Locker System

Shanmugasundaram M, Bagubali A, Rajesh R, , Hemprasad Yashwant Patil, Dhanabal R, Karthikeyan A, Ann Mary Andrews, and Akshit Nagpal
School of Electronics Engineering
Vellore Institue of Technology
Vellore, India
mssundaramvit@gmail.com

Abstract— The proposed development intends to establish an autonomous bank locker using industry-standard innovative locker technologies to deliver more flexible and reasonably priced semi-autonomous bank security mechanisms with minimal human intervention. In this design, there are two layers of locker security. The system proposed in this effort is a better security system regarding the number of security tiers. Its primary base is facial recognition. The first level is implemented by asking the user to input a passkey. A matrix keypad and Python programming are both employed. The user is then authorized to continue to the subsequent stage if a match is confirmed to exist. The second level was implemented using Python programming, OpenCV software, and face detection and identification techniques. To make Windows compatible with third-party apps Putty and Xming, the Raspberry Pi was linked to the laptop using IEEE 802.3 Ethernet and X11 forwarding on the UBUNTU operating system. IEEE 802.11 USB Wi-Fi was used to connect devices to the Wi-Fi network. The HAAR OpenCV standard has been used for face detection because of its better Face Acceptance and Rejection Ratio. The EIGENFACES OpenCV standard is employed for face recognition due to its efficacy, robustness, and simplicity.

Keywords- Face Detection, Haar Like Feature, AT&T Lab Data Base, Eigenfaces, Principal Component Analysis, Raspberry Pi Module, OpenCV

I. INTRODUCTION

This paper implies establishing a two-level bank locker system to verify, monitor, and control the security of bank lockers. Its structure makes it incredibly trustworthy, multi-level, and successful in attaining its objective. To make bank locker systems safer and more secure, increasing the percentage of security measures that can be applied in real-world situations with the hardware currently in use is necessary [1]. The proposal's initial setup involves asking each user for their password and building a live database of those with access to the locker. Both of these pieces of information are stored in a database. The positive database, which comprises images of the authorized person's face, is trained using negative images to increase facial recognition accuracy [2]. Obtaining the user's password and comparing it to the password that the authorized party has recorded is the objective of the first security level. The 4x4 matrix keypad and Python programming are both employed. If a match is discovered, the user can proceed to the second security level. Otherwise, users probably couldn't use the locker [3]. The second security level performs face detection and recognition to verify if the user's face matches the real-time database already stored in the system. The user's face must be validated for the locker to unlock. The second level also has an email-generating system that sends the authorized person an email if the user crosses the first level and descends to the second level to alert them. The implementation of the second level involves Python programming and OpenCV technologies.

Since most recognition systems are PC-based, creating a portable system is another challenge. A PC's portability, however, is limited by its compactness and power consumption. Face recognition is restricted to a few applications, which is cumbersome [4]. Using an embedded system is one technique to overcome the limitations of a P.C. The image capture, and recognition algorithm's design method is investigated in the embedded system. The most refined aspect was utilizing the Raspberry Pi board module and its peripherals. The Raspberry Pi is integrated with a Broadcom BCM 2835 System-on-Chip (SoC). SoC features an ARM 1176JZF-S 700 MHz CPU, a Video Core IV GPU, and 256 MB of RAM that can be expanded to 512 MB. There is no internal hard disc or solid-state drive; an S.D. card is used for booting and permanent storage [5]. There are various tools for the principal programming language, Python. As Haar-like properties, three elements are necessary for face identification here. The second aspect is the Eigenfaces face recognition system and the AT&T LABS database. The system can then be effectively integrated into the Raspberry Pi module. Because face detection and identification technology are difficult to deceive, those techniques are used. In this work, even complex sensors are not necessary. It is the main focus of this work.

Security systems have recently become more crucial and attentive. Security for bank lockers is vital for a variety of reasons. One of the reasons is that it shields expensive items that are extremely difficult to acquire, such as jewelry, hard currency, and title documents. The number of crimes has risen during the past several years. The issues with bank lockers are also present with them. More awareness and caution should be added to bank security systems. Unauthorized entry to bank lockers has occurred recently in a few instances. As a result, the system has to be more trustworthy, effective, and multi-level. The issue of security levels is a challenge for the current security systems. The smaller number of security layers is simple for the thieves to spoof. Banks locker systems use a hard-wired lock and key mechanism that locksmiths can trick. Locker systems with biometric scanners and password entry methods are launched in this digital age. These are ineffective, though, because they can be tricked if someone has the password or is wearing a wearable synthetic finger glove.

The categories of the biometric methods are shown in Fig. 1. Fear and disease may cause behavioral characteristics to change. Compared to other techniques, face detection, and recognition systems are more straightforward, accurate, and non-intrusive [6]. Two steps can be taken to accomplish this strategy. Face detection comes first, and face recognition comes afterward. It is necessary to compare a single face picture with many input images during face recognition to distinguish between face and nonface zones during face detection. The existence of eyeglasses and a beard, facial expression, occlusion (someone blocking the visual), image orientation (change in rotation), imaging circumstance (lightning and lens qualities), and other issues might arise while taking an image from an RPi-like posture [7]. Face recognition is the most secure method since it is non-intrusive, distinctive, stable, and has a low false recognition rate [8]. Therefore, we rely on facial detection and identification technology, which is difficult to track. Even complex sensors are not necessary.

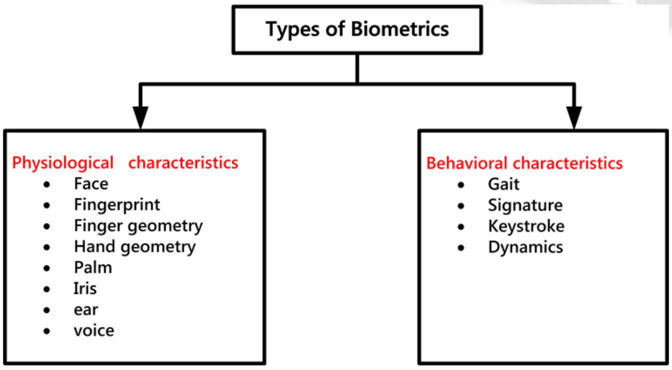

Figure 1. Types of Biometrics

II. PROPOSED APPROACH

The suggested system should operate in real time with a low false-positive rate. Fig. 2 depicts a block diagram of real-time face detection and recognition system incorporating different components. The hardware parts include a Raspberry Pi board, a Pi camera module, a servo motor, a matrix keypad, a push button, and a piezoelectric buzzer. The software includes OpenCV libraries and Python programming tools. The developed platform captures the photos, saves them in a real-time database, and then authenticates the user by matching their face to those stored in existing databases. The design approach of the algorithm for image capture and recognition is examined in embedded systems, with the Raspberry Pi board module and its peripherals emerging as the best. There are various tools for the principal programming language, Python. In this study, the suggested face detection and recognition system would be quick and have a high face detection success rate. The first is the Haar-like feature for face detection, the second is the Eigenfaces technique for face identification, and the third is our database or the AT&T labs database. The proposed system was then implemented on the Raspberry Pi module. Many metrics, including the false rejection and acceptance rates, are determined.

The proposed work prerequisites include establishing a real-time database of allowed locker openers and obtaining the password of each approved user. These particulars are both kept in the system. For accurate face recognition, the positive database containing images of the authorized person's face is trained with negative images [9]. The first level of security verifies the user's password against the saved password of the authorized individual. Python programming and a 4x4 matrix keypad are employed. The user can only continue to the second security level if a match exists. Otherwise, locker access is denied. The second security level performs facial detection and recognition to determine if the user's face matches the system's real-time database. The locker will only unlock if the user's face is recognized. Once the user crosses the first level and descends to the second, an alert email will be sent to the authorized person. The second-level approach uses Python programming and OpenCV technologies.

A. Face detection using the Haar algorithm

Existing face detection systems offer both advantages and disadvantages. They have employed various techniques, including skin tones, contours, neural networks, and the more advanced usage of templates and filters. These algorithms have a high computational cost. Due to the numerous differences in form and colouring of the human face, they are all time-consuming and challenging to execute [10]. Viola and Jones have successfully established a Haar Classifiers algorithm capable of detecting any object, including human faces [11].

1) Cascade Classifiers: Haar-like features form the basis for object recognition via the Haar classifier. Instead of using the intensity values of a pixel, these features employ the difference in contrast values between rectangular groupings of neighbouring pixels. The contrast differences between pixel groups are applied to evaluate a region's relative brightness and darkness. Haar-like features consist of two or three neighbouring groups with a relative contrast variance. As seen in Fig. 3, Haar-like characteristics are widely used to detect an image. By adjusting the size of the pixel group being analyzed, it is simple to scale the Haar features. It permits the use of features to detect objects of varying sizes.

2) Integral image: Superficial rectangular characteristics of a picture are computed using an intermediate image representation known as the integral image. The integral image is an array comprising the sums of the intensity values of the pixels immediately to the left and directly above the pixel at location (x, y) inclusive.

Consequently, if A[x, y] is the original image and A.I. [x, y] is the integral image, then the integral image is computed according to equation (1).

$$AI[x,y] = \sum_{x<x',y<y'} AI[x',y'] \qquad (1)$$

The 45-degree rotating features, such as the line feature in Fig. 2(e), require an additional intermediate representation known as the rotated integral or the sum auxiliary image. The rotated integral image can be obtained by adding the pixel intensity values at a 45° angle to the left and above for the x value and below for the y value. Consequently, if A[x, y] is the original image and A.R. [x, y] is the rotated integral image, then the integral image is obtained using equation (2).

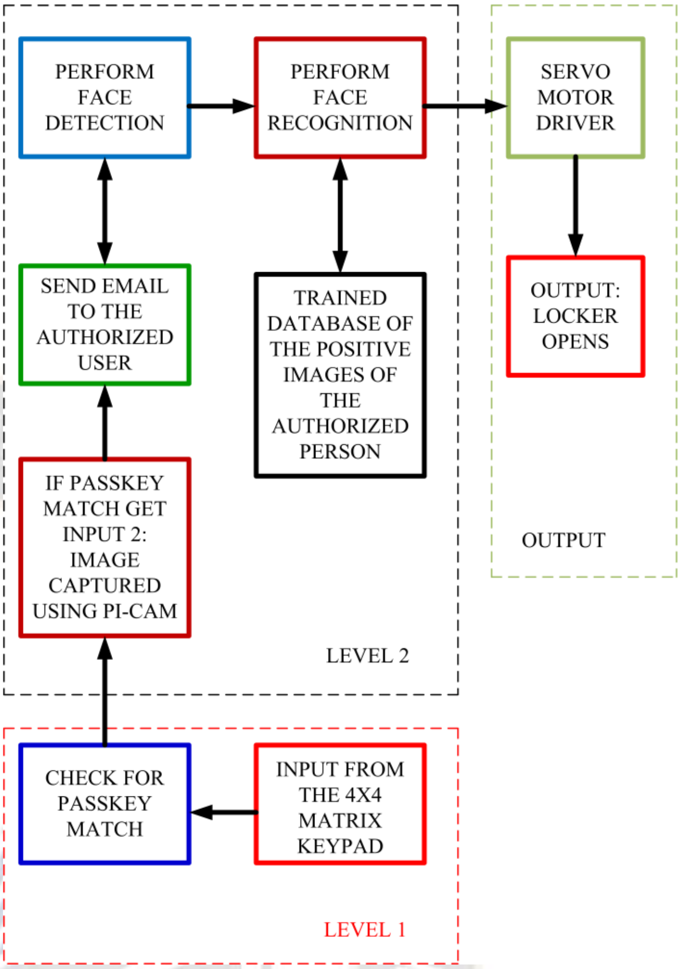
PERFORM FACE DETECTION
PERFORM FACE RECOGNITION
SERVO MOTOR DRIVER
SEND EMAIL TO THE AUTHORIZED USER
TRAINED DATABASE OF THE POSITIVE IMAGES OF THE AUTHORIZED PERSON
OUTPUT: LOCKER OPENS
IF PASSKEY MATCH GET INPUT 2: IMAGE CAPTURED USING PI-CAM
OUTPUT
LEVEL 2
CHECK FOR PASSKEY MATCH
INPUT FROM THE 4X4 MATRIX KEYPAD
LEVEL 1

Figure 2. Block Diagram of the System

$$AI[x,y] = \sum_{x'<x,\ x'<x-|y-y'|} AI[x',y'] \qquad (2)$$

It simply requires two steps, one for each array, to compute integral image arrays. Using the suitable integral image and the difference between six to eight array elements composing two or three linked rectangles, a feature of any scale may be calculated. Therefore, computing a feature is incredibly quick and effective. It also implies that computing features of varying sizes involve the same effort as calculating a feature consisting of only two or three pixels. Detecting objects of varying sizes need the same amount of work and time as detecting things of comparable sizes, as scaling requires no additional effort.

3) Classifier cascaded: Although computing a feature is incredibly efficient and quick, it is impracticable to calculate all 180,000 features inside a 24 x 24 sub-image. Fortunately, only a tiny percentage of these features are required to assess whether or not a sub-image may include the requested item. While evaluating sub-images, just a few attributes that describe an object are employed to reduce as many sub-images as feasible. The objective is to delete around 50 percent of the sub-images that do not include the object. This procedure is repeated, with the number of features employed to analyze the sub-image increasing at each iteration. The cascade of the classifiers enables just the most probable sub-images to be examined for all Haar features that differentiate an object. It also permits the variation of a classifier's accuracy. Reducing the number of steps increases both the false alarm rate and the positive hit rate.

4) Training classifier for facial features: Haar Classifier cascades must first be trained to detect human face features like the lips, eyes, and nose. A moderate AdaBoost method and Haar feature techniques should be implemented to train the classifiers. Open Computer Vision Library is an open-source library established by Intel to facilitate the development of computer vision-related programs (OpenCV). The OpenCV library is intended for applications in HCI, robotics, biometrics, image processing, and other areas where visualization plays a significant role; it also includes implementing Haar classifier detection and training. For classifier training, two sets of images are required. One set has an image or scene that does not include a detectable object, such as a facial characteristic. This collection of images is known as negative images. The second group of images, the positive images, depict at least one instance of the object. The placement of the objects within the positive images is indicated by the image name, the pixel in the upper-left corner, and the object's height and width. 5,000 negative images with at least one-megapixel resolution were employed for training face training features. These images included commonplace things, such as paperclips and views of woods and mountains. To achieve the most robust face feature detection feasible, the initial positive set of images must reflect individual diversity, including ethnicity, gender, and age. The Facial Recognition Technology (FERET) database of the National Institute of Standards and Technology is an excellent resource for these images. This collection comprises over 10,000 photographs of over 1,000 individuals in various lighting circumstances, positions, and perspectives.

1,500 images were employed to train each face feature. These images were shot at a frontal angle of 0 to 45 degrees. It offers the variance necessary for identification if the head is slightly rotated. One classifier was trained for the eyes, another for the nose, and a third for the lips. Once the classifiers were trained, they were applied to another batch of images from the FERET database to detect face traits. The classifiers, except the mouth classifier, have a high detection rate. As mentioned, the percentage of false positives is likewise rather large.

B. Face recognition using the Eigenfaces developed using Principal Component Analysis

Eigenfaces-based face recognition involves the extraction of the face's central features and the formation of eigenvectors. The training set (database) pictures are projected onto the primary eigenvectors, and the projection values are calculated [12]. During the recognition phase, the projection value of the input image is also determined, and the distance from the known projection values is computed to determine the individual's identity. Principal Component Analysis (PCA) may be used to build a set of Eigenfaces from a massive collection of images displaying various human faces. Eigenfaces may be viewed as a collection of "standardized face constituents" obtained through statistical analysis of many images of faces.

1) Eigenvalue and Eigenvector: Employing large matrices might be expensive in processing time. A computation may need hundreds or thousands of iterations of large matrices. Moreover, the behaviour of matrices would be difficult to investigate without essential mathematical tools. Eigenvalues and eigenvectors are mathematical tools with applications in Linear Algebra and differential equations, calculus, and many other fields. Eigenvalue and eigenvector stem from the German term Eigen, which means "proper" or "distinctive." An eigenvalue of a square matrix is a scalar often indicated by the Greek letter λ, whereas an eigenvector is a nonzero vector marked by the tiny letter x. All eigenvalues and eigenvectors for a given square matrix A fulfill the equation (3).

$$Ax = \lambda x \tag{3}$$

2) Face recognition using the Eigenfaces developed using Principal Component Analysis: Eigenfaces-based face recognition involves extracting the face's central features and forming eigenvectors. The training set (database) pictures are projected onto the primary eigenvectors, and the projection values are calculated [GuangShun ¬et al. (2011)]. During the recognition phase, the projection value of the input image is also determined, and the distance from the known projection values is computed to determine the individual's identity. Principal Component Analysis (PCA) may be used to build a set of Eigenfaces from a massive collection of images displaying various human faces. Eigenfaces may be viewed as a collection of "standardized face constituents" obtained through statistical analysis of many images of faces.

a) . Eigenvalue and Eigenvector: Employing large matrices might be expensive in processing time. A computation may need hundreds or thousands of iterations of large matrices. Moreover, the behaviour of matrices would be difficult to investigate without essential mathematical tools. Eigenvalues and eigenvectors are mathematical tools with applications in Linear Algebra and differential equations, calculus, and many other fields. Eigenvalue and eigenvector stem from the German term Eigen, which means "proper" or "distinctive." An eigenvalue of a square matrix is a scalar often indicated by the Greek letter λ, whereas an eigenvector is a nonzero vector marked by the tiny letter x. All eigenvalues and eigenvectors for a given square matrix A fulfill the equation (3).

In other terms, an eigenvector of a matrix is a vector whose product with the matrix is always an integer multiple of the vector itself. This integer represents the eigenvalue related to the eigenvector.

Eigenvectors have the following characteristics:

- They are only computable for square matrices
- An m x m matrix has m eigenvectors and their associated eigenvalues.
- All eigenvectors are parallel or at right angles to one another.

Since every eigenvector is linked with an eigenvalue, x, and λ that correspond to one another are commonly referred to as Eigepairs. Eigenspace is a space that contains all eigenvectors with the same eigenvalue. Eigenfaces are a collection of eigenvectors obtained from the covariance matrix of the probability distribution of the high-dimensional vector space of conceivable human faces.

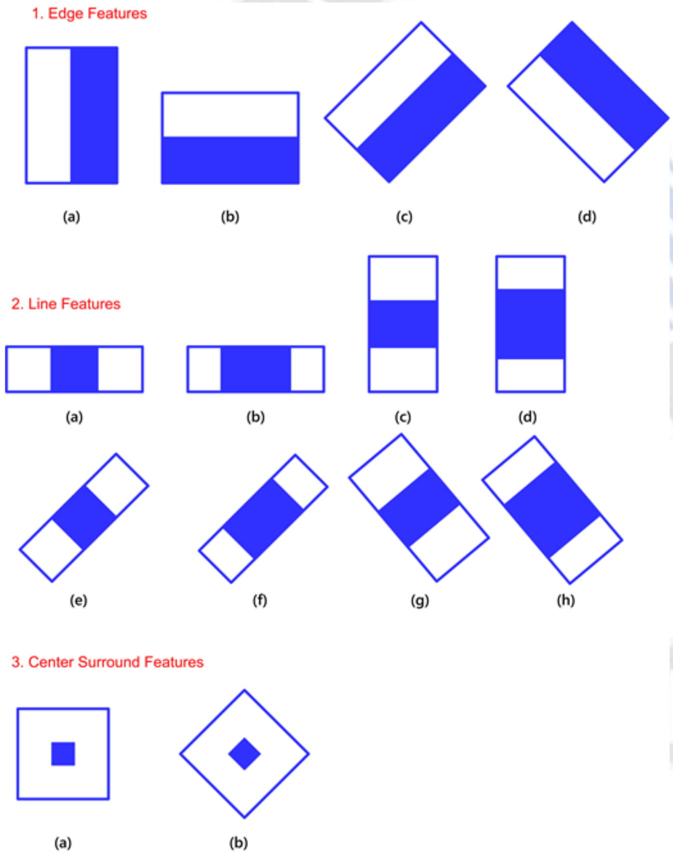

Figure 3. Common Haar Features

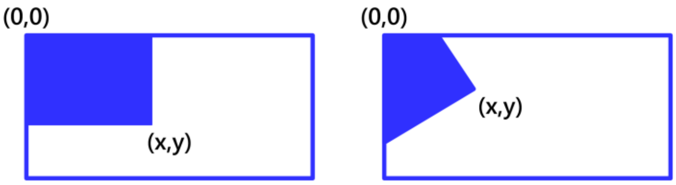

Figure 4. Summed Area of Integral Image and Rotated Integral Image

b) Principal Component Analysis (PCA): PCA is a statistical method that may be used to minimize a dataset. It is a linear transformation that selects a new coordinate system for the data set so that the variance by any projection is the most significant.

The first most significant variation resides on the first axis (called the first principle component), the second most crucial variance on the second axis, and so on. Retaining lower-order main components and disregarding higher-order ones, PCA can minimize the dimensionality of a dataset while retaining the qualities that contribute most to its variance. The concept is that low-order components frequently include the "most crucial" parts of the data. Facial recognition aims to classify input signals (image data) into distinct groups (persons). The input signals are very noisy (e.g., the noise results from varying lighting conditions, postures, etc.), yet the input images are not entirely random. Despite their variances, all input signals include patterns. Such patterns, which may be noticed in all signals, might include certain items (eyes, nose, and mouth) in any face and the relative distances between these features in facial recognition. These characteristics are called Eigenfaces (or principal components) in facial identification. Principal Component Analysis is a statistical method that can extract them from the original visual data (PCA). Through PCA, each original image of the training set may be transformed into its associated Eigen face original image. It is possible to rebuild the original images precisely using all the Eigenfaces recovered from the original images. However, employing only a portion of the Eigenfaces is also possible. The reconstructed picture is afterward a close approximation of the original image.

Nevertheless, losses caused by ignoring some Eigenfaces can be mitigated. It happens by choosing only the most critical features (Eigenfaces). The omission of Eigenfaces is necessary due to the scarcity of computational resources. Thus the purpose of PCA is to reduce the large dimensionality of the face space (observed variables) to the smaller intrinsic dimensionality of features pace (independent variables), which are needed to describe the data economically. It is the case when there is a strong correlation between observed variables. The eyes and mouths of many digital images of human faces captured under identical lighting are aligned to form a collection of Eigenfaces. They are then resampled at the precise pixel resolution (m x n) and represented as m x n-dimensional vectors whose components are the pixel values. The eigenvectors of the facial image statistical vector distribution covariance matrix are then retrieved. Because eigenvectors share the same vector space as face pictures, they may be regarded as m x n-pixel face images; thus, Eigenfaces. The primary Eigen face resembles a nondescript androgynous average human face. Some subsequent Eigenfaces can correlate to generic characteristics like left-right and top-bottom asymmetry or the presence or absence of a beard. Other Eigenfaces are challenging to classify and appear odd. Eigenfaces can be added when suitably weighted to approximate a grayscale representation of a human face. Eigenfaces provide a technique for applying data compression to faces for identification purposes. A few eigenvector terms are required to resemble most people's faces reasonably. Not only is it feasible to extract the face from Eigenfaces given a set of weights, but the inverse is also achievable. This alternative method involves extracting the weights from Eigenfaces and the face to be identified. These weights reveal the degree to which the questioned face varies from "typical" faces represented by the Eigenfaces. Using these weights, one may therefore determine two crucial facts:

- Determine if the image in the issue contains a face. In this instance, the weights of the image and face images diverge too significantly (i.e., images from which we know are faces).
- The image is most likely not a face. Comparable faces (images) exhibit similar traits (Eigenfaces) to similar degrees (weights). The images might be grouped into clusters if weights are extracted from all accessible photos. Thus, any photos with comparable weights are likely to face.

c) Database: Section III describes the hardware configuration. Then, we must create a database comprising the facial pictures of approved individuals. When a person is ready to snap a photo, they press the button to capture the image. Enter the following in the terminal window, "Sudo python capture-positives.py." This code's explanation has already been provided above. After executing this script, we will click the button and capture images under varying lighting circumstances and facial angles to construct a database with a high rejection rate and excellent recognition capability. The accuracy is better for a more significant number of images. This script will capture images using the box hardware and save them to the training/positive subdirectory (which will be created by the script if it does not exist). When a face is identified, it is converted to grayscale, cropped, and scaled before being saved to the positive database. For OpenCV to function with the Eigenfaces Algorithm, the images of the authorized individual who may unlock the lock must be trained. The work consists of an extensive collection of face photos optimized for training the face recognition algorithm. It is the database of faces that AT&T Laboratories Cambridge published in the mid-1990s.

d) The AT&T Face database, also known as the ORL Database of Faces, has ten photographs of each of forty separate individuals. For some participants, photos were captured several times, with varied lighting, facial emotions (open/closed eyes, smiling/not smiling), and facial characteristics (glasses/no glasses). All the photographs were taken against a uniformly black background, with the people facing the front (with tolerance for some side movement). This collection of negative pictures represents faces prohibited from opening the locker. After this, we must run the script 'train.py,' which will use the 'HaarFaces' and 'Eigenfaces' algorithm via the XML file named 'Haar cascades frontal faces alt.xml', which is a part of the OpenCV library. It will generate the authorized user's 'Positive,' 'Negative,' and 'Mean' images. The command entered at the terminal was: "Sudo python train.py".

III. Hardware analysis

A. *Rpi Controller*

The proposed concept is implemented at the portable hardware level. The design approach of the algorithm for picture capture and recognition is analyzed by implementing it in embedded systems. It identified that the Raspberry Pi board module and its peripherals are the best. The high-performance Broadcom BCM2835 SoC chip is here to power the world's most miniature fully-functional computer and to construct an ever-expanding power system. The 700 MHz processor is equipped with ARM11 Architecture to use the computer entirely. A Dual-Core Video-Core IV Multimedia co-processor was working as a GPU, which offers the Raspberry Pi the ability to play any game. Four USB 3.0 ports and a high-output HDMI port may function as a home entertainment system or support an H.D. monitor. 512MB of memory and a specialized operating system (RASPBIAN) bring the chip to life. With GPIO Pins and Python, this is a small chip's most acceptable conceivable combination. Due to its existing tools ' interoperability, Python has been used to program Raspberry Pi.

B. *Rpi Camera Module*

For facial identification, a high-resolution image is essential. It was not achievable with a standard webcam that would have been less expensive. Therefore, we opted for the more expensive Raspberry Pi camera, which offered superior performance and resolution. The camera comprises a compact circuit board (25mm by 20mm by 9mm) that connects via a flexible ribbon cable to the Camera Serial Interface (CSI) bus socket on the Raspberry Pi. The camera's image sensor features a five-megapixel native resolution and a fixed-focus lens. The camera's software allows still photos with a maximum resolution of 2592x1944 and video resolutions of 1080p30, 720p60, and 640x480p60/90. Three apps are available, including raspistill, raspivid, and raspistillyuv. The raspistill and raspistillyuv aim to take photos, while raspivid is intended to capture video. All apps are command-line oriented and were developed using the MMAL API, which runs on OpenMAX. The MMAL API provides a system that is simpler to use than OpenMAX. MMAL is a Broadcom-unique API exclusive to the Video core four platforms. The applications utilize up to four OpenMAX

***Article Received**: 02 February 2023 **Revised**: 12 March 2023 **Accepted**: 28 March 2023*

(MMAL) components: camera, preview, encoder, and null sink. All programs utilize the camera component, whereas raspistill uses the Image Encode component, raspivid uses the Video Encode component, and raspistillyuv delivers its YUV or RGB output directly to the file from the camera component. The preview display is optional and can be directed to the entire screen or a selected rectangular region. If the preview is disabled, the preview frames are "absorbed" by the null sink component. Even if preview frames are not required for display, the camera must make them for exposure and white balance calculations. It is also possible to ignore the filename option, in which case the preview is presented but no file is created, or to send all output to stdout. You may get command-line help by entering the application's name on the command line.

C. Servo Motor

In the locker system, a servo motor controls the lock's latch. It is linked to GPIO PIN 18 to receive the raspberry pi's pulse. A servo motor is a rotary actuator that highly precision regulates angular position, velocity, and acceleration. It comprises an appropriate motor connected to a position feedback sensor. In addition, it needs a somewhat complex controller. Often a module is built expressly for use with servomotors. As its position encoder, simple servomotors may employ as resistive potentiometers. Close competition with stepper motors is only utilized at the most basic and economic levels. In the potentiometer's track, they experience wear and electrical noise. Although it is feasible to distinguish their position signal to generate a speed signal electronically, PID controllers that utilize such a speed signal often require a more precise encoder. Throughout our project, we have utilized servo motors with three controllers that allow us to control their motion using the Raspberry Pi's output manually. The image Processing Mechanism has been utilized to regulate a mechanical locking mechanism.

The project is designed to provide a safe locker system with two layers of protection. As shown in Fig.5, the flowchart indicates that level 1 requires the user to input the passphrase using the matrix keypad. If all of the characters of the submitted password match the password remembered by the authorized user, then access to level 2 is granted. Requests to access the locker are denied if no match is found. The picture frame is collected, and an email is forwarded to the authorized user at the second level. Face detection is conducted on the image that was recorded. If a face is recognized, the picture is then trained and processed. It is performed to extract the facial characteristics, as shown in Fig.6. The face is then compared to the database. If a good appearance is achieved, the locker will open. Otherwise, the locker will not open. Fig. 7 and Fig.8 depict the hardware setup of the proposed approach.

IV. SYSTEM OPERATION

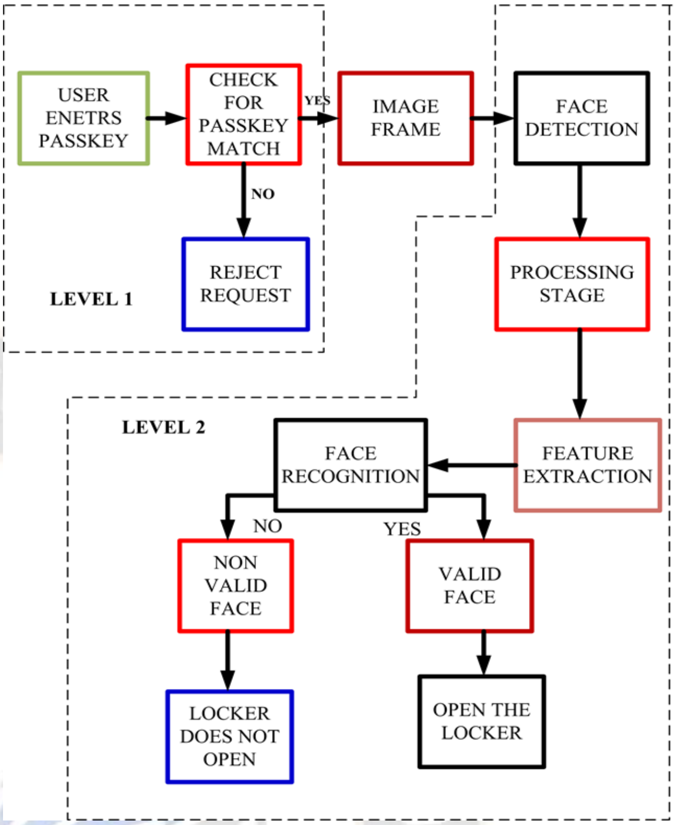

Figure 5. Flow Chart of the System

Figure 6. The output obtained after Training the Database

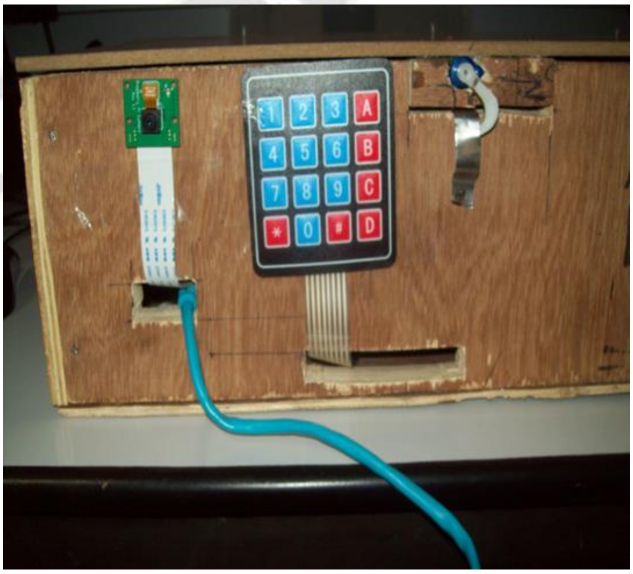

Figure 7. Final Assembly of the Hardware

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 4
DOI: https://doi.org/10.17762/ijritcc.v11i4.6452
***Article Received**: 02 February 2023 **Revised**: 12 March 2023 **Accepted**: 28 March 2023*

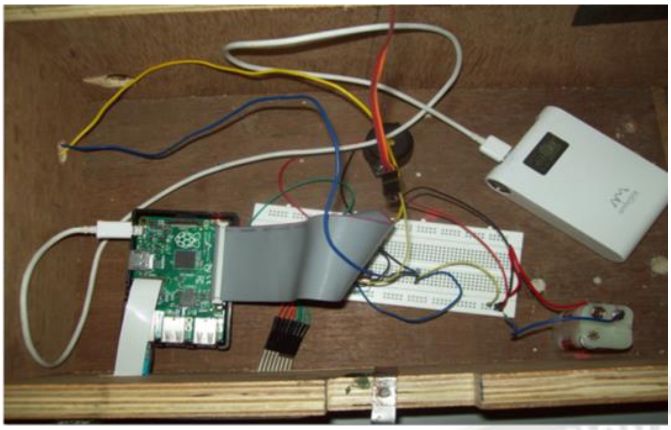

Figure 8. Internal Circuit of the System

V. Results and Discussion

The project's work's logic may be provided using specific logic tables. A "0" in the tables denotes an error, whereas a "1" denotes a valid response. The following are the tables:

TABLE I. Logic table for passkey match

Passkey Saved By Authorized User	Passkey Entered By The User	Output
1	0	0 (Request rejected)
1	1	1 (Move to level 2)

Table 1 describes the password authentication process. If the user-entered password matches the saved password, the user will be allowed to the next level. Otherwise, the system will reject user access. Logic table for passkey match

TABLE II. Logic table for Face match

Face Of The Authorized User	Face Of The User	Output
1	0	0 (Request rejected)
1	1	1 (Locker Opens)

Table 2 summarizes the level of the facial recognition process. After succeeding in the level 1 password security process, the user will be allowed for facial authentication. In this, the captured user face by the camera will be compared with the stored database. If a match is found, the lock will open for access to the locker by the user. Otherwise, access will be denied.

TABLE III. Overall Logic table

Passkey Status	Captured Face For Recognition	Output
0	0	0 (Request rejected)
0	1	0 (Request rejected)
1	0	0 (Request rejected)
1	1	1 (Locker opens)

Table 3 summarizes the dual-layered security process of the bank locking system. The four combinations enumerate the various security state of the user. The system will allow the user to access the locker if he has succeeded in the password and face recognition levels. In all other cases, he will be denied access. Different problems plague the security measures in place for locker rooms. All those problems will be attempted to prevent by the suggested security system. Table 4 describes the advantage of the proposed approach over the existing methods.

TABLE IV. Comparative study of the existing and proposed systems

S.No	Existing System	Proposed System
1.	The existing systems have only one level of security.	The number of security levels is higher in the proposed system.
2.	In these systems, the cameras are used only for monitoring.	In this system, the camera is not only used for monitoring but also takes a control action by taking a snapshot on a real-time basis.
3.	These systems are designed only to protect the locker rooms from robbery. They require a manual appearance for information security.	This system will not only protect the locker from external activity but also will try to catch robbers by taking the snapshot and mailing it to the security officials, which can be used as a piece of evidence in future

VI. Conclusion

Security for bank locker rooms is a growing concern worldwide. This work provides two strategies for gaining access to a secured locker. With the implementation of two layers of security, this article aims to create a bank locker system that is more secure. A person permitted to use the locker must maintain track of a passkey for the locker and build a positive database of images of themselves taken with an RPi camera module. The user is also prompted for the email address they want to receive alerts. Making the user input the passkey is the primary goal of the first level. Only when the passkey matches may it help the user to go to the second level. In the second level, the user's face is captured, and an alert email is sent to the email address of the authorized person. The identified face is now against the learned data to determine the recognition. The locker is only accessible if the passkey and the face match. It guarantees a high degree of security during the peak of criminal activity. A buzzer device can also be installed within the locker to notify the bank of unauthorized access. For bank locker rooms requiring a high level of security, the real-world replica of the prototype may be created with minimal design costs and low operational costs. Soon, different degrees of protection can be added for monitoring and management.

References

[1] Bansal A; Mehta K.; Arora S. (2012): Face Recognition Using PCA and LDA Algorithm, Second International Conference

on Advanced Computing & Communication Technologies, pp. 251-254.

[2] Beumer G M;, Tao Q; Bazen A M; Veldhuis R N J, (2006): A landmark paper in face recognition, 7th International Conference on Automatic Face and Gesture Recognition (FGR06), pp. 66.-78.

[3] GuangShun Shi; BiJia Lan; Liang Huang; XiaoYong Peng; Jia Feng Ma; Qian Liang, (2011): Research of face recognition under active infrared lighting based on embedded system, The First Asian Conference on Pattern Recognition, pp. 535-539.

[4] Haibin Lv; Di Lu; Limin Yan. (2021): Face Detection and Recognition Algorithm in Digital Image Based on Computer Vision Sensor, Journal of Sensors, 2021, pp. 1-16.

[5] Huang Chenxi; Yuan Zhenguo. (2020): Face Detection and Recognition Based on Visual Attention Mechanism Guidance Model in Unrestricted Posture, Scientific Programming, 2020, pp. 1–10.

[6] Jiarui Zhou; Lai Jiang; Zhen Ji; Linlin Shen. Haar-like features based eye detection algorithm and its implementation on TI TMS320DM6446 platform, (2009): IEEE International Workshop on Imaging Systems and Techniques, pp. 89-93.

[7] Murugappan M; Mutawa A. (2021): Facial geometric feature extraction based emotional expression classification using machine learning algorithms, PLoS ONE, 16 (2), pp.1-20.

[8] Senthilkumar G; Gopalakrishnan K;, Sathish Kumar V, (2014): Embedded Image Capturing System using Raspberry Pi system, International Journal of Emerging Trends & Technology in Computer Science, 3 (2), pp. 213–215.

[9] Verma A; A Multi-Layer Bank Security System, (2013): International Conference on Green Computing, Communication and Conservation of Energy (ICGCE), pp. 914-917.

[10] Wang J.; Wang B.; Zheng Y; Liu W. (2019): Research and Implementation on Face Detection Approach Based on Cascaded Convolutional Neural Networks, (2017): International Conference on Vision, Image and Signal Processing (ICVISP), pp. 34-39.

[11] Xiong T; Wang S. (2019): Intelligent farm management and control system based on Raspberry Pi, IEEE 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), pp. 1286-1290.

[12] Yadav M.; Koul R.; Suneja K, (2020): FPGA Based Hardware Design of PCA for Face Recognition, 7th International Conference on Signal Processing and Integrated Networks (SPIN), pp. 642-646.