

Smart Raspberry Pi Bank Safety Deposit box With Facial Recognition: Fintech Case Study

Jay Patel ¹ and Sabah Mohammed ²

¹Lakehead University

²Affiliation not available

October 30, 2023

Abstract

Bank safety deposit box is one of the secure storage for the clients to store their valuables such as important property or business paper, gold or even money but is in some case scenario traditional safe deposit are not safe enough, there have been many robberies from this safe box because of the code lock can be hacked and there no alarming system in this kind of safe boxes. In this paper I proposed a smart secure safe box with facial reorganization unlock feature using raspberry pi and uno Arduino which can used to secure your documents or valuables in a bank or at your home. In this I have used raspberry pi and pi camera to perform the face unlock and to send security alert through email. Secondly, I have used Arduino and hall sensor to trigger an alarm when the door is open unauthorized or kept open.

Smart Raspberry Pi Bank Safety Deposit box With Facial Reconition: Fintech Case Study

Jay Patel
Masters in computer science
Lakehead University
Thunder Bay, Canada
jpatel27@lakeheadu.ca

Dr. Sabah Mohmmmed
COMP5800 YB Supervisor
MSc. Computer Science
Lakehead University, Thunder Bay
Canada
mohammed@lakeheadu.ca

Abstract—Internet of things (IOT) is immensely growing technology not only in the field of home automation and security but also in myriad fields such as health care, drones, automotive and so on. One of the main aspects of smart home system in how secure your devices are. Bank safety deposit box is one of the secure storage for the clients to store their valuables such as important property or business paper, gold or even money but is in some case scenario traditional safe deposit are not safe enough, there have been many robberies from this safe box because of the code lock can be hacked and there no alarming system in this kind of safe boxes. In this paper I proposed a smart secure safe box with facial reorganization unlock feature using raspberry pi and uno Arduino which can used to secure your documents or valuables in a bank or at your home. In this I have used raspberry pi and pi camera to perform the face unlock and to send security alert through email. Secondly, I have used Arduino and hall sensor to trigger an alarm when the door is open unauthorized or kept open.

Keywords—IOT, Security, Home Automation, Safe box, Raspberry pi, Uno Arduino, Pi camera, Hall Sensor

I. INTRODUCTION

The Internet of things (IOT) has gained so much popularity in last few years. So basically, IOT is a network of interconnecting devices such as sensors and actuators to transfer the data to the network without any human to human or human to machine interaction. One of the main applications of IOT is home automation and security such as smart lights, smart fridge, smart locks, smart thermostat and so on which works independently without any human interaction and communicate with each other. Nowadays fintech companies such as financial organization or a bank are focus so much on security and to avoid any fraudulent activity. Some fintech companies around the world uses face recognition techniques for securing their mobile banking and not only that they use high tech face recognition cameras in bank which analysis the face of every person entering the bank and check in their databases to detect whether the person is a robber or not. There are various image recognition techniques such for such purpose are image classification, object detection, facial recognition, semantic segmentation and so on. So, I created a smart facial unlock safe box system which can be used in a bank or at home. This safe box can be unlock by recognizing the trusted face of the owner and it is also secure with the safety features such as 24 hour live footage from the safe box's camera, if nobody is home it detects any person near the safe box and send the picture to the user's email, and also triggers the alarm if somebody tries to open the safe box's door or the door is kept open.

Furthermore, let me introduce more about the hardware, software and working of the system. The reason behind choosing Raspberry pi with pi camera as a major component is that it is the smallest micro controller which gives as much power as actual computer and which can make the safe box small and portable. Next, I have used Arduino because it is portable and simple to handle small sensors. I have used Haar Cascade classifiers for the face detection which have many pre-trained classifiers for facial detection, person detection, upper body detection, eyes detection and so on. So now let me give you a shallow idea how the facial recognition system works. First of all pre-processing part takes place which will take 30 images of a person and stored in local directory, which is sent to our cascade classifier for the feature extraction. To recognize the face as per shown in Fig 1, when the person is in the front of the pi camera which would be attached on the top of the safe box, the pi camera detect the face using cascade classifiers and matches the face with the trusted images stored in the local directory and if the face matches then the servo motor will turn to unlock which I am using as a lock. Moreover, if there is nobody at home, the user will activate the safety mode in which the pi will be running continuous 24 hour and showing the live footage on users phone and if somebody enter in the room or near the safe it will detect the person and send the image of the person as email notification to user.



Fig 1. Face Recognition Process

II. PROBLEM DEFINITION

Safes and safe boxes are been used since many years to store valuable things such as important documents, gold or money. The problem with traditional safe boxes is that they all just have same kind of manual locking mechanism and there have been many hacks to break this kind of safe open and due to which there are many chances of robbery. Another weak point here is that this kind of safe don't have any alarm system or there is no mechanism that the owner can know who tried to break their safe. One more problem with these safes are convenience, every time owner has to enter the code to open the safe. So, the idea here is to build a smart secure face unlock safe box. This safe is convenient and safe because user just have to stand in from of the safe and safe will automatically opens once it recognises the trusted person. Another safety feature here is that if no body is home and if someone tries to open the safe or enter the

room where the safe is located then it will send picture of that person to the owner of the safe. And lastly to make this safe more secure I have implemented an alarm system which will trigger when the someone tries to open safe door forcefully or the door is kept open.

III. RELATED WORK

In this section I would like to discuss about some research work done related to my topic. This research work [1] which is highly related to my topic and I am following this research to implement my work. So, this research was conducted by Rahul Bhatt, Hiren Kanzariya, Kewal Thakkar and Prof. Sridhar Iyer. They build a bank vault security system which will automatically traps the burglar if he tries to breach bank's security system. They are using two door security system in which as soon as the person enters the first door the security camera on the second door will activate and which has image processing capability and which will detect a person's face and if the person is an authorized person then he will get an OTP on his mobile which will be the password for the second door. But if there is some unauthorized person trying to enter the camera will recognize it and trigger the alarm and close the first door so he gets trapped there. For hardware they have used Arduino Uno, Arduino camera module, Fingerprint module, Keypad, LCD Display and a DC motor. For face recognition they have used Arduino camera image sensor which is a VGA camera which can capture a whole frame, sampling and resolution of 8 bits and the image can be captured up to 30 frames per second, they have used OpenCv pretrained frontal face Haar cascade model for face detection. The major advantage of their system is which two door face verification system it is very difficult for an intruder to enter the bank and likely very less chances of robbery but on the other hand there would be also a disadvantage like it would be very time consuming for the clients to enter the bank as they have to pass a security system and enter the OTP each and every time.

In this section I would like to discuss about some research work [2] done related to my topic. The first research work is Raspberry Pi Based Face Recognition System For Door Unlocking conducted by Akshay N. Patil, Rohit B. Ranavare, Dayasagar V. Ballal. They used OpenCV library which will be formulated as given images to identify or verify one or more person with the help of stored database of faces. They have used PCA algorithm which detects the face and extracts the features from it. In their system once the features are extracted, their system matches the feature to the stored database images and if the features are matched the system will decide if it is a match or no match. Additionally, they have used SIM300 GSM module which will send the alert to the authorized person that somebody entered the door or some unauthorized person tries to enter. The tools and technology they used are Raspberry Pi, Pi camera, Python and GSM module. Now let me explain their working, so the whole project is divided into two parts one is for capturing and creating the database and another part will capture the image which is used to identify and compare the images in the database. So first of all, the camera module comes in picture which is using Pi camera which will capture an image and send the captured image to the Raspberry Pi module. The Raspberry Pi module compares the image taken by Pi camera with the Eigen face image. First time when the camera

captures the image to create a database it captures six types of the images to create the database in the directory and this database will be compared with the live captured images. The third module is GSM module which will send a message to the authorized person after comparison output is positive or negative. If the face matched successfully the message will send "Person Identified Door Open" otherwise it will send message that unknown person trying to enter the door. Their future work is to use this system as high surveillance camera or for the attendance system in the classroom.

The third research work [3] which I have referred is a Face Recognition On Bag Locking Mechanism proposed by Mohamad Hafis Izran, Nurul Hawani Idris and Shafishuhaza Sahlan. Their idea here is to build a bag with locking mechanism which can be unlocked using face recognition. Their system consists of four modules which are detection, alignment, feature extraction and matching, where localization and normalization are face detection and alignment which are processed before facial feature extraction and matching is performed. Now let me explain their working and methodology, their algorithm is based on information theory approach which decomposes face images into small set of characteristic feature image which is called 'Eigenfaces', which are the main components of initial training set of face images. The recognition is done by projecting a new image in the subspace spanned by the Eigenfaces which is also called face space and then face is classified by comparing its position in the face space with the position of trusted faces. In the hardware the main component is Raspberry Pi which is used as the processing part of the face recognition algorithm and it is powered by a 5V and 2A battery. They have also used an electromagnetic lock which is capable of providing 50N attractive force to prevent unauthorized person to open the case. Lastly I would like to explain how their training process is working, so first the training data is created which includes pictures of faces that are not allowed to open the bag, they are using the AT&T research images for their database. For face recognition they are capturing 35 positive images which are stored in local directory. So, when the camera is launched it will detect the face and recognize the positive image then the lock will unlock otherwise it will remain closed.

The fourth research work [4] I have referred is a Face Detection Based ATM Security System using Embedded Linux Platform conducted by Jignesh J. Patoliya and Miral M. Desai. The idea here is to build a smart ATM security system based on embedded Linux platform. The system is implemented on Raspberry Pi with the OpenCV computer vision software for image processing. They have implemented high level security mechanism which is provided by continuous monitoring by camera to detect the face and if the face is detected successfully it will let the user to use ATM services and if the face is not detected repeatedly two times then it will lock the door of the ATM cabin for security purpose and send the message through GSM module to bank authorities. Now let me explain how their face detection methodology works, the face detection algorithm contains different methods to detect face from which the primary method is Haar based algorithm and Local binary Pattern based algorithm. The face is detected from the image using Haar cascade feature in which the cascade function is trained from many positive and negative images.

IV. METHODOLOGY

In this section I am going to explain different models and methods which are used in project in details with their working. So for face detection I have used Haar cascade classifier model in while I have used frontal face model to detect the faces from video frames. And the second main model which I have used is LBPH which stands for Local Binary Patterns Histogram which is used for recognizing the detected face. It is basically used for face matching feature. To send email notification I have used SMTP protocol. Let me explain these methods in details.

Haar Cascade Frontal Face Model

Haar cascade [5] is a classifier in which the cascade function is trained by superimposing the positive image over a set of negative images. So as shown in Fig 2, there are two features used in Haar cascade, one is Haar feature and another is Integral images. So, the first step is to collect the haar features as you can see in the picture below there are three kind of feature extraction, edge feature, line features and center-surround features. A haar feature [6] consider a rectangular region in the specify location of the direction window, it sums up the pixel intensity in particular region and calculate the pixel between the sums. Integral images are used to make this process a lot faster, so amongst the features are collected most of them are irrelevant so, the job of integral images is to find which ones are relevant from the irrelevant once and it does it by superimposing the positive image over a set of negative image. In the below image we can see how haar cascade detects the face. So, as you can see below the features are eyes, nose, mouth in a rectangular region. So, through this we can identify that how the face looks different from any other things such as car or an animal.

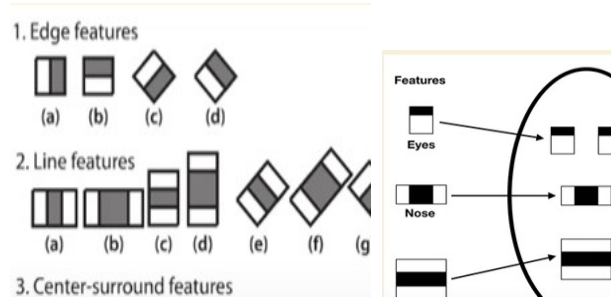


Fig 2. Haar Cascade Frontal face method

LBPH (Local Binary Patterns Histogram)

LBPH [7] is a very popular face recognition model, and it is a significant improvement over Eigenfaces and Fisherfaces as both of these are affected by light as they try to look at the facial image as a whole. LBPH, however, tries to find the local structure of the image as it compares each pixel to its neighboring pixels. So, once we have identified a face and then we use LBPH recognizer to recognize whose the face is. So, let me first explain how LPBH works so as shown in Fig 3, you can see in the image below LPBH model takes the values of every pixel and create a matrix with the center pixel. So next the LPBH converts the pixel value matrix into binary value matrix, so if the value of the

center pixel is lower then the value of neighboring pixel then it will take 0 or if the value greater then it will take 1 as shown in below image. After we got all the binary value it will convert it into decimal value to create a histogram as shown in Fig 4.

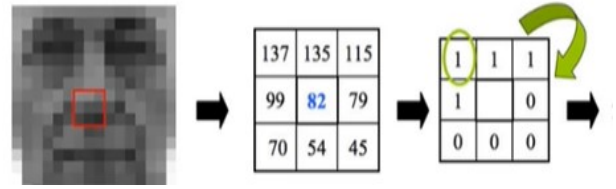


Fig 3. LBPH Pixel Matrix



Fig 4. LBPH Histogram

SMTP(Simple Mail Transfer Protocol)

SMTP is a part of application layer of TCP/IP, it uses the process called store and forward through which it moves your email on and across the network. SMTP contains set of codes that makes communication of email message between email servers simple. In python 'smtplib' library is used to use SMTP services. There are several other libraries used with SMTP in python such as 'MIMEMultipart', 'MIMEText', 'MIMEImage' which are used for the text and image part.

Hall Effect Sensor and Magnet

Hall effect sensor [8] is a sensor which will activated by an external magnetic field. Magnetic field have two main characteristic flux density and polarity (North and South pole). The output signal from the hall sensor depends on the magnetic field density across the sensor. When the magnetic flux density around the sensor exceeds a certain pre-set threshold, the sensor detects it and generates an output voltage called the Hall Voltage, V_H . Hall Effect Sensors consist of a thin piece of rectangular p-type semiconductor material like gallium arsenide (GaAs), indium antimonide (InSb) and indium arsenide (InAs) which passes a continuous current through it. When the sensor is places near magnetic field, the magnetic field apply a force on semiconductor which reflects the charge carrier, electrons and holes to the other side of semiconductor slab. The entire working of hall effect sensor is shown in Fig 5.

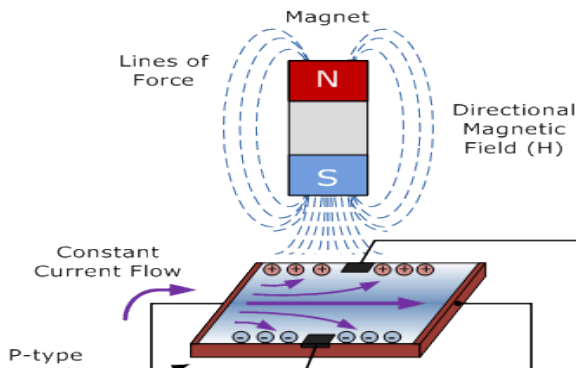


Fig 5. Hall Sensor And Magnetic Field

V. PROTOTYPING

In this section I would like to explain the prototyping of my whole system and also explain the different components used in the project. So I am using two different microcontroller hardware which are Raspberry Pi and Arduino to implement the whole system. The main part of face detection and locking system is implemented on Raspberry Pi and attached hall sensor and buzzer to Arduino to make the system more secure. Raspberry Pi and Arduino are working independently on separate power supply. Following are the components used in the project.

A. Components

- Raspberry Pi
- Uno Arduino
- Pi Camera
- Servo motor
- Hall sensor and magnet
- Buzzer
- Jumper Wires

Raspberry Pi Connection and code explanation

In the project raspberry pi is one of the main components as most of the part of the project was implemented on it. As shown in Fig 6 I have used Pi camera for face detection and recognition and servo motor as a safe lock which will unlock when the camera detects the face successfully. The figure below shows the circuit diagram of raspberry pi. The Pi camera is attached in the camera slot of raspberry pi. The servo motor have three pins one is VCC which supply power to it, other is ground and the third pin is digital pin. As shown in the figure below the VCC of servo motor is connected to 3V, the ground is connected to ground and the digital pin is connected in GPIO21 pin of raspberry pi.

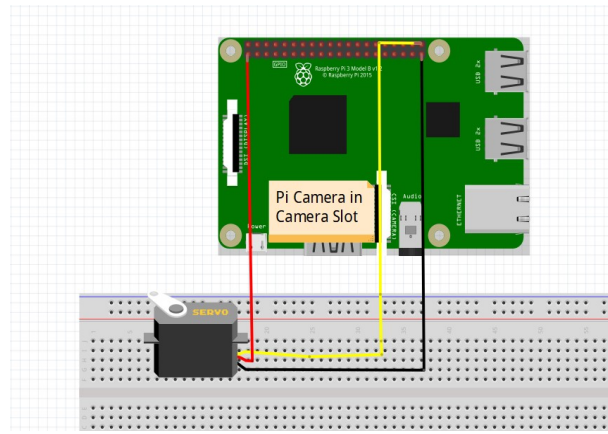


Fig 6. Raspberry Pi Circuit Diagram

➤ Code Explanation

1) DETECTION METHOD

Firstly, I am writing the detection python code to detects the face using haar cascade frontal face model. And then taking the 30 gray scale images and store them in my working directory for further recognizing. Step by step explanation of detection method is as following.

- Initializing camera module, resolution, framerate and raw capture size.
- Using Haar cascade model for frontal object detection.
- Creating the directories to store trusted faces.
- Taking video from pi camera as input and 30 frames as a platform to detect face in RGB mode after converting it to gray scale it will only display a square around the face in the given frame in entire video and including the square portion downside the screen to show the face in gray scale.
- Storing the 30 gray scale images in the image directory of the person.
- The below Fig 7 shows the main code segment for detection method.
- The Fig 8 shows the output of detection method.

```
count = 1
for frame in cam.capture_continuous(capture, format="bgr", use_video_port=True):
    #storing the 30 gray scale images in the image directory of the person
    if count > 30:
        break
    frame = frame.array
    gray_scale = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = front_face_cascade.detectMultiScale(gray_scale, scaleFactor = 1.1)
    #after converting it to gray scale it will only display a square around
    for (x, y, w, h) in faces:
        RGray = gray_scale[y:y+h, x:x+w]
        fname = dirct_name + "/" + name + str(count) + ".jpg"
        cv2.imwrite(fname, RGray)
        cv2.imshow("face", RGray)
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
        count += 1
```

Fig 7. Detection method code segment

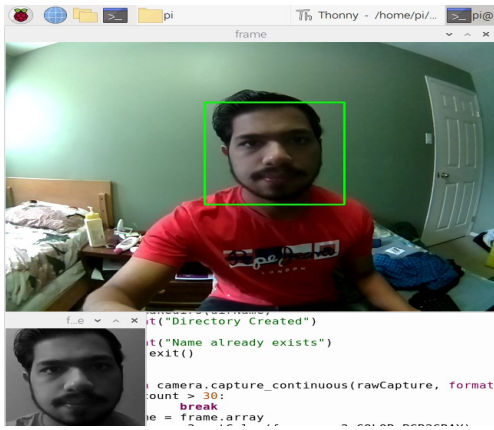


Fig 8. Face Detection And Gray Scale Output

2) RECOGNIZER METHOD

In recognizer method I am taking the 30 images stored in image directories and creating a label file which will store names of the person and also using LBPH(Local Binary Patterns Histogram method) which I will explain in depth in next methodology. Here I have used LBPH model to recognize the detected face and create a trainer.yml file which will store the recognized faces data. The steps for recognizer model are as following.

- Using haarcascade frontal face detection pretrained model and LBPH(Local Binary Patterns Histogram) from OpenCV as a recognizer.
- Scan into each image directory and if the images are present, image is converted into NumPy array.
- Once it takes the image it creates a label file and store all the names of the image in the label file.
- And then it performs the face detection again to make sure that we have the right images and then prepare the training data.
- Initiating training and save the trainer.yml file in the local working directory.

3) TRAINING METHOD

In this method the I will connect my hardware lock (servo motor) with raspberry pi, so if there is trusted face in front of camera the lock will unlock.

So in this method first of off the pi camera will launch and it start recording and convert the image is image array and perform the face detection using haar cascade model and once the face is detected it will check in your recognizer which is trainer.yml, so if the face match successfully and the confidence is less then 70 then the lock will unlock. The steps of this method are as following.

- Setting up servo motor and assigning the pin.
- Creating PWM channel for servo at 50Hz frequency.
- Initializing the camera setting resolution and frame rates and converting raw capture to PiRGBArray.

- Using haarcascade for face detection and LBPH recognizer for recognizing the faces and loading the trainer.yml file.
- Launch the Pi camera and starts recording video frames and convert the video frames in gray scale. Compare the faces from live video frames to our trainer.yml.
- And if the confidence is less than 70 it will detect the face and create a rectangle with text unlock and confidence. If the face is detected correctly then the servo motor will turn from lock to unlock.
- The below Fig 9 shows the main code segment for training method.
- The Fig 10 shows the output of Training method.

```
for frame in cam.capture_continuous(capture, format="bgr", use_video_port=True):
    frame = frame.array
    gray_scale = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = front_face_cascade.detectMultiScale(gray_scale, scaleFactor = 1.1)
    for (x, y, w, h) in faces:
        RGray = gray_scale[y:y+h, x:x+w]
        #compare the faces from live video frames to our trainer.yml
        id, confid = lbph_recognizer.predict(RGray)

        #if the confidence is less than 70 it will detect the face and unlock
        if confid <= 70:
            try:
                while 1:
                    n.ChannelOutputWrite(12.5)
```

Fig 9. Training method code segment

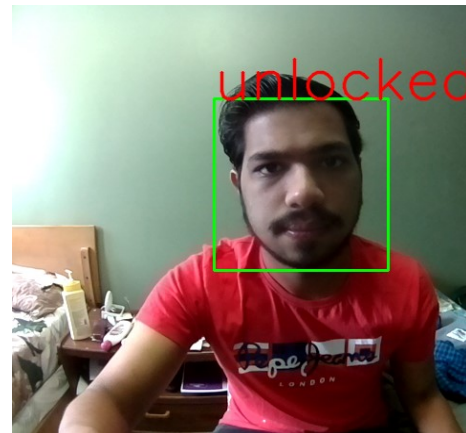


Fig 10. Face Recognition Unlock

4) DETECTS PERSON AND SEND EMAIL NOTIFICATION

In this section I would like to discuss how the security system of the safe box works to prevent thieves and robbers. In order to make my system more secure I have implemented a python code on raspberry pi using pi camera which will detect any person who comes in front of pi camera and send the picture of the that person to the owner through email. To detect the person, I have used three models, facial recognition which detect the face, upperbody recognition which detects upper human body and last fullbody recognition which detects whole human body. All these three

models are a Haar cascade models. Following is set by set explanation.

- Initializing camera module, resolution, framerate and raw capture size.
- Using Haar cascade model for frontal object detection.
- Pi camera is continuously recording the frames and checking for object, here object is a face.
- If the object is detected, then it capture that frame and will execute the send email method which will send the email of the captured frame.
- The below Fig 11 shows the main code segment for send email method.
- The Fig 12 shows the output of send email method.

```
def check_for_objects():
    global epoch
    while True:
        try:
            #if the object is detected, then it capture that frame and will
            frame, obj = v.camera.get_object(obj_clsfr)
            if obj and (time.time() - epoch) > email_interval:
                epoch = time.time()
                print ("Trying to send email")
                sendEmail(frame)
                print ("Done")
        except:
```

Fig 11. Send email method code segment

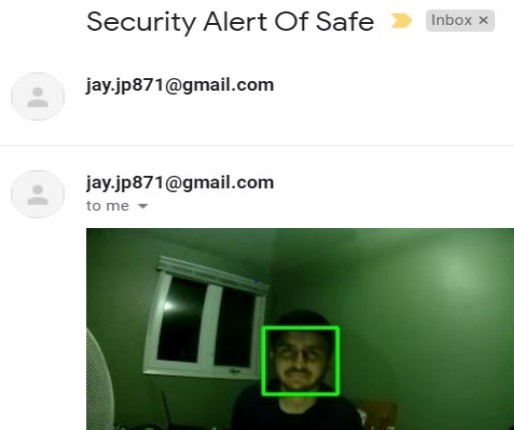


Fig 12. Send Email Alert

Arduino and Hall Sensor connection

In order to make the safe system more secure I have implemented a alarm system using Arduino, Hall sensor and a buzzer as shown in Fig 14. This system would be attached on the door of safe box and there would be a magnet on the frame of the the box, so once we activate this system the door is closed and the hall sensor and magnets are together and the hall sensor is high due to magnetic field but if somebody tries to open the door the magnet will get away from hall sensor and hall sensor will turn to low which will trigger the buzzer on. As shown in figure below the hall sensor have three pin from which one is ground, second is VCC which is connected to 5V of Arduino and third is digital pin which is connected to the pin number 3 of Arduino. On the other hand the buzzer have two pin one

connects to the ground and the digital pin is connected to the pin 7 of Arduino. Here I have used 'Magnetstate' to check if the sensor detects the magnetic field or not. If the hall sensor detects the magnetic field then the buzzer will be off and when the sensor don't detects the magnetic field the buzzer will trigger. The below Fig 13 shows the main code segment for hall sensor.

```
void loop()
{ // if the hall sensor detects the magnet the buzzer wil
  if(magnetState(hall_pin) && !flag)
  {
    flag = true;
    digitalWrite(buzzer, LOW);
  }
  // if the hall sensor dont detects the magnet the buzze
  else if(!magnetState(hall_pin) && flag)
  {
    flag = false;
    digitalWrite(buzzer, HIGH);
  }
```

Fig 13. Alarm method code segment

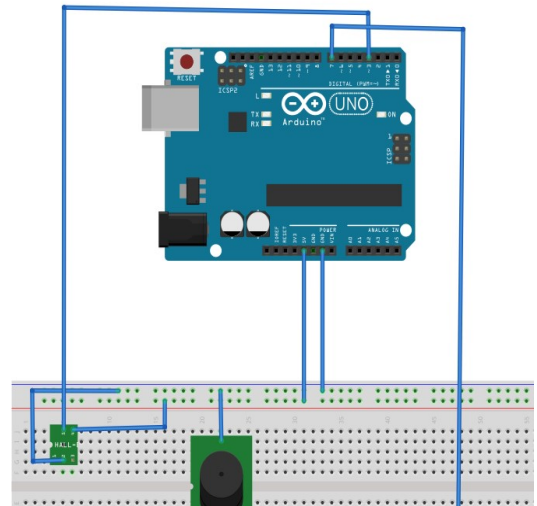


Fig 14. Arduino And Hall Sensor Circuit Diagram

VI. CONCLUSION

The Smart Raspberry Pi Bank Safety Deposit box is a fully secure safe box which can be used in banks or any fintech organization to store valuables. Facial detection and whole separate security system make this deposit box more secure and very difficult to break. And as this system uses microcontroller as its processor, it is very portable and can be moved anywhere. Hence it can be concluded that the proposed system has achieved the required goals and objectives. This system and be alerted in future to make it more secure and useful implementing fingerprint sensor and adding a SMS alert system which owner will received when the alarm triggers. Apart from fintech this proposed system can also be used for personal lockers.

REFERENCES

- [1] Rahul Bhatt, Hiren Kanzariya, Kewal Thakkar and Prof. Sridhar Iyer (2018). 3 Tier Bank Vault Security
- [2] Shrutika V. Deshmukh and Prof Dr. U. A. Kshirsagar (2017). Implementation of Human Face DetectionSystem for Door Security using Raspberry Pi .
- [3] Mohamad Hafis Izran Ishak, Nurul Hawani Idris and Shafishuhaza Sahlan (2019). Face Recognition on Bag Locking Mechanism
- [4] Jignesh J. Patoliya and Miral M. Desai (2017). Face Detection based ATM Security System using Embedded Linux Platform
- [5] Vandna Singh1, Dr. Vinod Shokeen, Bhupendra Singh, Face detection by Haar Cascade classifier with simple and complex background images using open cv, 2013.
- [6] Phillip Ian Wilson, Dr. John Fernandez, Facial Feature Detection Using Haar Clasifiers, 2015
- [7] Mohannad A. Abuzneid, Ausif Mahmood, Enhanced human face recognition using LBPH descriptor, 2016.
- [8] Umar Hassan, Azeem Iqbal, and Muhammad Sabieh Anwar (2019). Flux and rate of change of flux with a Hall sensor and solenoids