Access Control Of Door using Face Recognition And Home Security Alert Using Raspberry Pi And Internet

Gorantla Sai Gopi, Palla Priyanka, K. Bharathi, J.Rajasekhar, Krishnaveni Kommuri

***ABSTRACT:** In the present age, the Internet of things (IOT) has entered a brilliant period of quick development. The Internet of things is an idea that expects to expand the advantages of the ordinary Internet steady network, remote control capacity, information sharing, etc. to products in the physical world. Ordinary things are getting associated with the Internet. This idea can be utilized to manage the security concerned issues in a financially way. In this paperwork a framework is being created to interface any entryway with the Internet, so the entrance control framework can be controlled from anyplace on the planet. For a situation that one isn't at home and a visitor is at his entryway steps then the approved individual will be told about the guest by means of IFTTT app, which pings up a message to your from anyplace and the framework IFTTT snap a photo of the guest and keep a record by sending a connection through Email. On the off chance that the approved individual needs to give a message the guest, it tends to be sent effectively through the IFTTT app and it will show up in a screen on the front essence of the entryway. The entryway lock can be controlled through the IFTTT. With the assistance of this framework, a proof of the guest can be kept as a record if any crisis case or situation happens.*

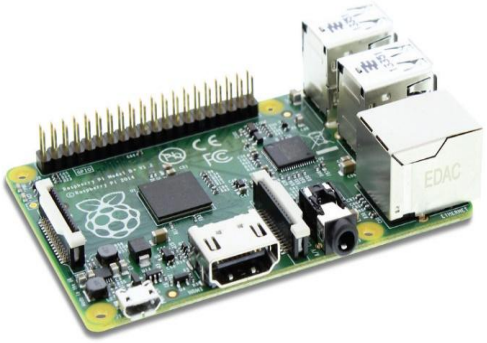
I. INTRODUCTION:

In this advanced world, wrongdoing has turned out to be current as well, this present time a ton of occurrences happens like theft, taking undesirable passageway happens suddenly. So, the security does matters in this day by day life. Individuals dependably stay occupied in their everyday work additionally needs to guarantee the wellbeing of their darling things. Here and there they neglect to take care of their vital things like keys, wallet, Visas and so forth. Without these, they can't get to their home or wherever they need. To incidents, numerous researchers came energetically to restrict them. Like Neelam Majgaonkar, Ruhina Hodekar, Priyanka Bandagale student, Assistant professor information Technology, Finolex Academy of Management and Technology, Tatnagiri, India. Introduced an Automatic Door Locking system but it is not cost effective and Md.Nasimuzzanman, Md. Shiblee Nooman, Srijon Saker proposed in their paper. Which is a totally a cost-efficient but the work process cannot be explained to the people of no knowledge about this equipment's, that accurately as the work with the wifi- Dongle, Wifi camera, complicates the work process with the Raspberry Pi.

II. COMPONENTS USED IN THE PROJECT:

Raspberry Pi:

Raspberry Pi is a mini computer of a credit card sized which is powered by Broadcom BCM2837 SoC with a 1.2 GHz 64-bit quad core ARM Cortex-A53 processor, but the clock speed is same as the V1.1 which is 900 MHz CPU.



Camera:

The camera can be used of any model which supports the Raspberry Pi, but the high definition camera is preferred to get the capture quality of the image or the computer vision.

Revised Manuscript Received on April 07, 2019.

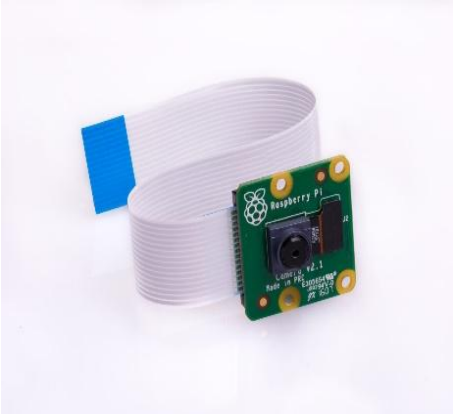
Gorantla Sai Gopi, UG Student, Department of Electronics and Computer Engineering, Koneru Lakshmaiah Education Foundation, Guntur, India.

Palla Priyanka, UG Student, Department of Electronics and Computer Engineering, Koneru Lakshmaiah Education Foundation, Guntur, India.

K. Bharathi, Associate Professor, Department of Electronics and Computer Engineering, Koneru Lakshmaiah Education Foundation, Guntur, India.

J.Rajasekhar, Associate Professor, Department of Electronics and Computer Engineering, Koneru Lakshmaiah Education Foundation, Guntur, India.

Krishnaveni Kommuri, Associate Professor, Department of Electronics and Computer Engineering, Koneru Lakshmaiah Education Foundation, Guntur, India.



Servo motor:
A servomotor is a rotational actuator or straight actuator that takes into consideration exact control of precise or direct position, speed and quickening. It comprises of an appropriate engine coupled to a sensor for position input.

Buzzer:
A buzzer is a signaling audio device. Which may be mechanical or electrotechnical. Now this typical buzzer is being used like alarm, timers and confirmation of user.

OpenCV:
It is for the most part gone for giving library of programming capacities at constant PC vision. Initially created by Intel, it was later bolstered by Willow Garage then Itseez (which was later gained by Intel). The library is cross-stage and free for use under the open-source BSD permit.

III. PROPOSED SYSTEM

Using IoT in the Security is emerging in the paradigm in which resources are being used and rapidly getting into the limelight.Now the face recognition which comes in the modern security which is being featured in a vast area and is being studied extensively. This application is now being used in almost all the technologies which are being invented and which are being used right now in the time being. Now the prototype works with respect to the Raspberry Pi which is being used in almost all the projects with face recognition complexities. Now the Principle of the component analysis comes to consideration when the hardware interacts with the different home appliance such as camera and servo motor using the Raspberry Pi. Now the face recognition is done with the code which is developed using the help of python using the libraries of OpenCV (Computer vision). Now this library is used in the in IDE which is used to run the python code in the Raspbian OS which specially developed for Raspberry Pi. This OpenCV is used in this case only for the communication between two parties, which are hardware and the code. In this project the camera is now linked to raspberry Pi.The working of the project can explain as the home security automation. The camera which is set near the door of a house will be monitoring the vision up till which it can be monitored by the camera. Then, if the human interaction is made such as if the, if the humans face comes in front of the camera then the code which is being run by the OpenCV, checks the data base of the PI and if the checking goes success then the door opens. If the checking fails i.e., if the face is not trained before the project is set then the buzzer will be set to on state. After that the notification will be sent to the Gmail of the house's owner, through the IFTTT app which is connected to the Gmail of the house owner.

IV. SOURCE CODES

```
# Import OpenCV2 for image processing
import cv2
# Start capturing video
cam = cv2.VideoCapture(0)
if (cam.isOpened() == False):
  print("Unable to read camera feed")
face_detector = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
# For each person, one face id
face_id = raw_input("enter id")
# Initialize sample face image
count = 0
# Start looping
while(True):
    # Capture video frame
    _,img = cam.read()
    # Convert frame to grayscale
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    cv2.imshow('frame1', gray)
    # Detect frames of different sizes, list of faces rectangles
    faces = face_detector.detectMultiScale(gray, 1.1, 5)
    # Loops for each faces
    for (x,y,w,h) in faces:
        # Crop the image frame into rectangle
        cv2.rectangle(img, (x,y), (x+w,y+h), (255,0,0), 2)
        # Increment sample face image
        count += 1
        # Save the captured image into the datasets folder
        cv2.imwrite("dataset/User." + str(face_id) + '.' + str(count) + ".jpg", gray[y:y+h,x:x+w])
        # Display the video frame, with bounded rectangle on the person's face
        cv2.imshow('frame', img)
    # To stop taking video, press 'q' for at least 100ms
    if cv2.waitKey(100) & 0xFF == ord('q'):
        break
    # If image taken reach 100, stop taking video
    elif count>50:
        break
# Stop video
vid_cam.release()
# Close all started windows
cv2.destroyAllWindows()
```


Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Source code for face datasets

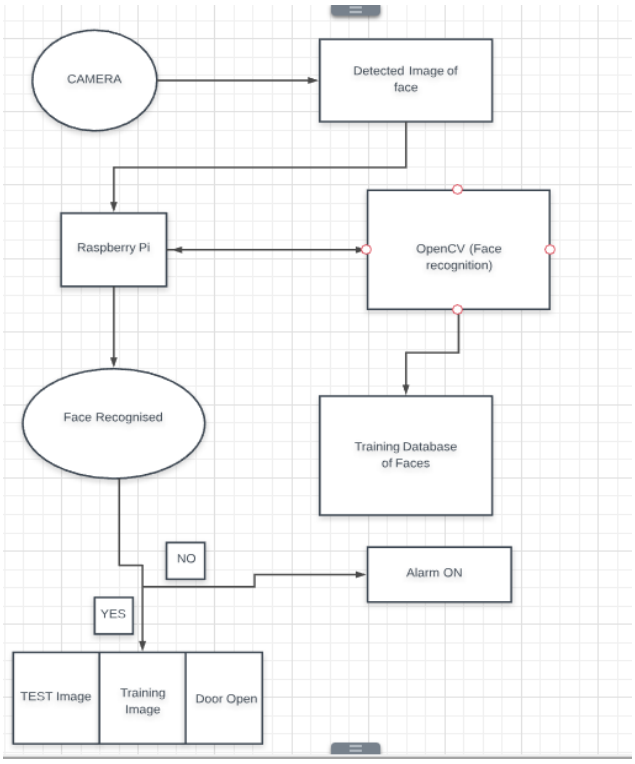
```
# Import OpenCV2 for image processing
import cv2
import time
# Import numpy for matrices calculations
import numpy as np
import RPi.GPIO as GPIO
import time
import urllib2
GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)
GPIO.setup(12, GPIO.OUT)
GPIO.setup(40, GPIO.OUT)
p = GPIO.PWM(12, 50)
p.start(7.5)
# Create Local Binary Patterns Histograms for face recognization
recognizer = cv2.face.LBPHFaceRecognizer_create()
# Load the trained mode
recognizer.read('/home/pi/full compled Raspberry-Face-Recognition-master/trainer/trainer.yml')
# Load prebuilt model for Frontal Face
cascadePath = "/home/pi/full compled Raspberry-Face-Recognition-master/haarcascade_frontalface_default.xml"
# Create classifier from prebuilt model
faceCascade = cv2.CascadeClassifier(cascadePath);
# Set the font style
font = cv2.FONT_HERSHEY_SIMPLEX
# Initialize and start the video frame capture
cam = cv2.VideoCapture(0)
# Loop
while True:
    # Read the video frame
    ret, im =cam.read()
    # Convert the captured frame into grayscale
    gray = cv2.cvtColor(im,cv2.COLOR_BGR2GRAY)
    # Get all face from the video frame
    faces = faceCascade.detectMultiScale(gray, 1.3,5)
    # For each face in faces
    for(x,y,w,h) in faces:
        # Create rectangle around the face
        #cv2.rectangle(im, (x-20,y-20), (x+w+20,y+h+20), (0,255,0), 4)
        cv2.rectangle(im, (x,y), (x+w,y+h), (255,0,0), 2)
        cv2.imshow('im',im)
        # Recognize the face belongs to which ID
        qwe = recognizer.predict(gray[y:y+h,x:x+w])
        print qwe
        ty="unkown"
```

Source code for face recognition

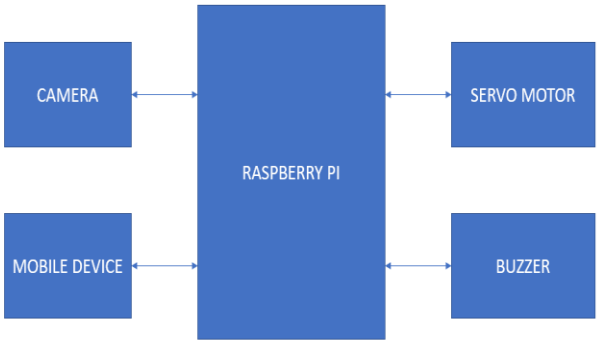
```
# Import OpenCV2 for image processing
# Import os for file path
import cv2, os
# Import numpy for matrix calculation
import numpy as np
# Import Python Image Library (PIL)
from PIL import Image
# Cr.eate Local Binary Patterns Histograms for face recognization
recognizer = cv2.face.LBPHFaceRecognizer_create()
# Using prebuilt frontal face training model, for face detection
detector = cv2.CascadeClassifier("/home/pi/full compled Raspberry-Face-Recognition-master/haarcascade
# Create method to get the images and label data
def getImagesAndLabels(path):
    # Get all file path
    imagePaths = [os.path.join(path,f) for f in os.listdir(path)]
    # Initialize empty face sample
    faceSamples=[]
    # Initialize empty id
    ids = []
    # Loop all the file path
    for imagePath in imagePaths:
        # Get the image and convert it to grayscale
        PIL_img = Image.open(imagePath).convert('L')
        # PIL image to numpy array
        img_numpy = np.array(PIL_img,'uint8')
        # Get the image id
        id = int(os.path.split(imagePath)[-1].split(".")[1])
        print(id)
        # Get the face from the training images
        faces = detector.detectMultiScale(img_numpy)
        # Loop for each face, append to their respective ID
        for (x,y,w,h) in faces:
            # Add the image to face samples
            faceSamples.append(img_numpy[y:y+h,x:x+w])
            # Add the ID to IDs
            ids.append(id)
    # Pass the face array and IDs array
    return faceSamples,ids
# Get the faces and IDs
faces,ids = getImagesAndLabels('/home/pi/full compled Raspberry-Face-Recognition-master/dataset')
# Train the model using the faces and IDs
recognizer.train(faces, np.array(ids))
```

Source code for training

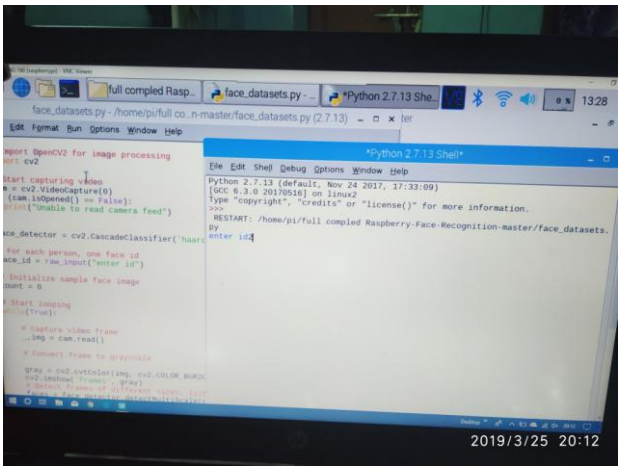
V.FLOW CHART



VI. BLOCK DIAGRAM

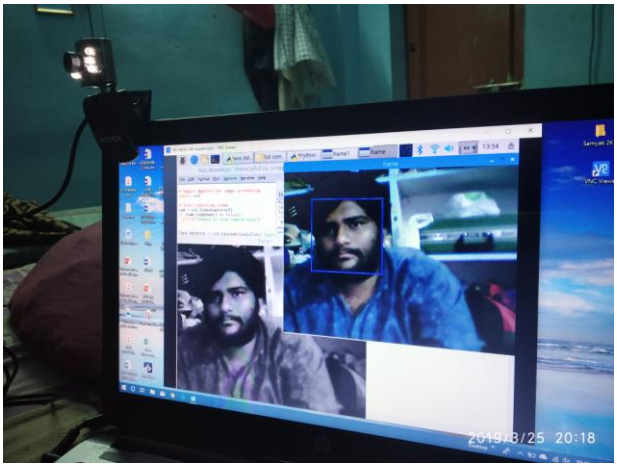


VII. OUTPUTS



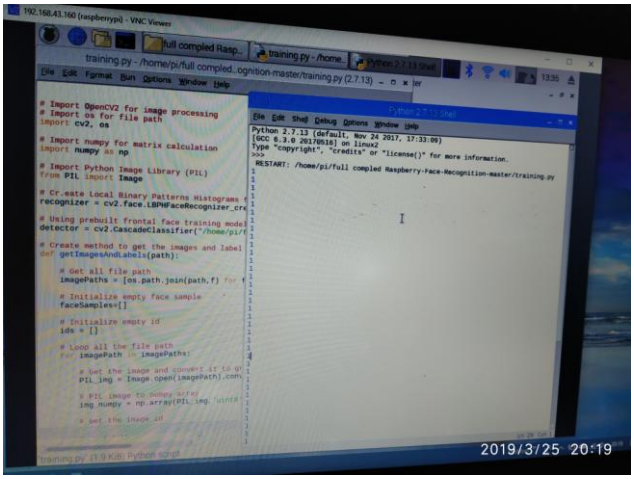
This is the output of giving ID for face recognition.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

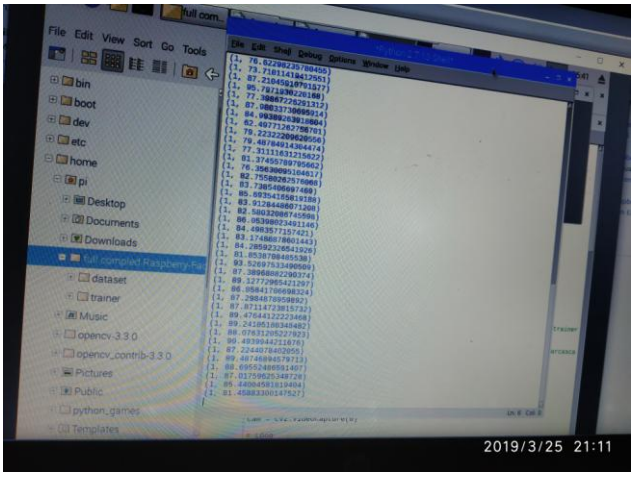


This is the output of users face recognition.

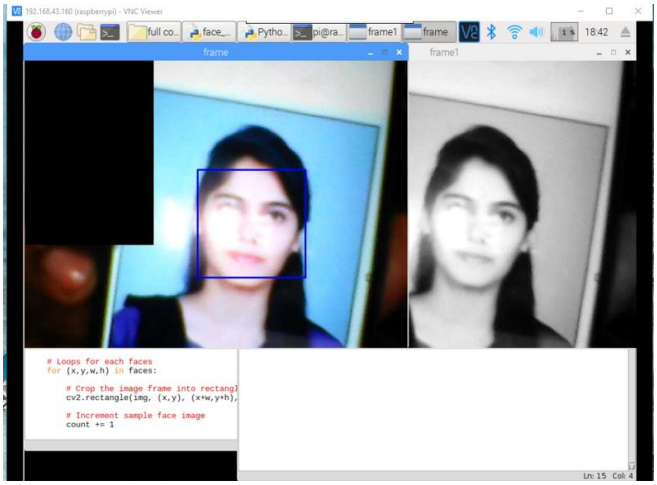
Here in the recognition process the camera will take upto 51 samples of photos and stores it in the database.



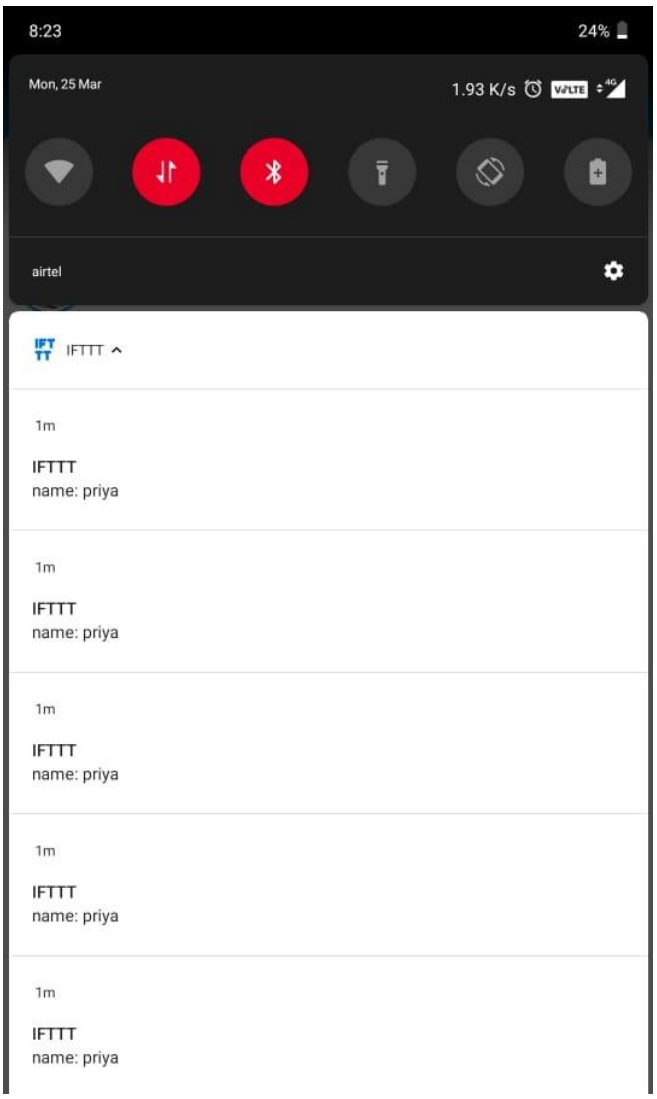
This is the output for training of the given users ID.



This is the output for reading of user ID data.



This is the output of the users face recognition.



This is the output of the users name to the mobile which is sent from IFTTT.

VIII. CONCLUSION AND FUTURE SCOPE:

We designed the System which reduces human efforts and provide security. Proposed system is cheap, reliable and components are easily available. It is also portable and easily upgradable. System provides Security locks for door, comfort, security and energy efficiency for user. Raspberry Pi operates and controls motion detector and cameras for capturing the image of the guest. The future scope of this propject is to add an extra module to the ongoing project which is GSM module, with the help of this GSM module we can send a alert message in terms of a password.

REFERENCES

1. Internet of Things in Industries: A Survey. Li Da Xu, Shancang Li, Wu He, Li Da Xu, NOVEMBER 2014
2. https://www.raspberrypi.org/blog
3. Real-Time Indian Traffic Sign Detection using Raspberry Pi S Maruthi,N Radhakrishnan 19th November 2017.
4. Face recognition system on Raspberry Pi Rihards Fuksis, Modris Greitans,Arturs Kadiks, Olegs Nikisins, Latvia.
5. A brief introduction to OpenCV David Abram, Mario Cifrek, Hrvoje Dzapo, Tomislav Pribanic,Ivan Culjak May 21-25,2012, Croatia
6. Robust Fingerprint Authentication Using Local Structural Similarity. Nalini K. Ratha, Vinayaka D. Pandit. , Ruud M. Bolle Vaibhav Vaish Applications of Computer Vision, 2000, Fifth IEEE Workshop, Dec. 2000.
7. Random Password Generation of OTP System using Changed Location and Angle of Fingerprint Features. KyungJun Kim, HyunShik Na, ByungRae Cha Computer and Information Technology, 2008.
8. A Python-based Software Tool for Power System Analysis. Federico Milano, July 2013.
9. IoT based smart water tank with Android application, Subodh S. Ingleshwar,Anjali A. Patil, Priyen P. Shah International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud).
10. Raspberry Pi based Interactive Home Automation System, Lovely Goyal,Anant Vaibhav,Sarthak Jain 2014 International Conference on Reliability, Optimization and Information Technology – ICROIT 2014, Feb 6-8 2014.