Project Report

Home Automation System

Required Software and Hardware:

- Raspberry pi
- Webcam
- Sensor(VL53L0X)
- LAN cable
- Tight VNC Viewer
- Putty
- Android Studio
- Nmap
- Online Web services(AWS)

Implementation:

This section consists of steps that are followed to implement the system. Firstly, the raspberry pi is setup with webcam, sensor and is connected to the network through LAN cable. The next step is to find out the IP address that is assigned to the raspberry pi. This can be accomplished by using command "nmap -sn (ipaddressofthenetwork)- nmap -sn 10.0.0.0/24". Now the connection is established through putty and Tight vnc viewer. Open the putty session with ip address of raspberry pi with login as: pi and password: raspberry. Then type the command "tightvncserver" that gives the hostname to communicate with the raspberry pi console. Once entered into the raspberry window, there is a need to install several packages to get the required functionality.

The packages that need to be install are: Python, opencv, boto, avconv, fswebcam, speech_recognition, pydub, pyaudio, vlc player, Amazon polly

Now, git clone the repositories for calculating the distance and face recognition purposes.

`https://github.com/johnbryanmoore/VL53L0X_rasp_python`
https://github.com/af001/pi-detector

Add the test images to the collection in the AWS. Open an account in the AWS which can be used as server (Ec2 instance). Get a public DNS which can be used as username to log into the cloud server. Generate a private key for authentication and store it safely for future references. Open a new putty session and enter the username as "ec2-user@public DNS" and browse the private key file under SSH-> Auth.

The execution starts with capturing the test images for a particular person and storing them into the home collection of the AWS(Face Recognition) with "add_image.py" program. The continuous transfer of images from raspberry pi to the server and from the server to the android application will flows depending on the frame rate specified by the user. At a point of time, a test image is taken through the webcam, a distance between the hand the sensor is calculated using VL53L0X_exmaple.py program which will be stored in a variable and face recognition is applied through facematch.py program that displays the name of the user if the face matches, else it displays the message saying that it is not detected. If the match is observed and the distance calculated is in the same limits that are defined to the person then the image, along with

a file that contains the details of the user who matches with the test image, and the distance calculated are sent to the cloud server using SCP or sockets. Once the server receives them it tries to send them to the android app. By running the python code that is written in the server the details will be sent to the android app and depending on the functionality defined in the android application the output will be displayed. The IP address in the android start page will be the ipv4 address of EC2 instance that is created for server purposes and the port number will be available in the code written in the server. By clicking on "connect to the server", the images will be displayed in the continuous manner. The display details show the matched user details and it prompts with an option for authorizing or non-authorizing the test person. The user needs to get the approval from the owner of the application to complete the entire authentication process. This will be achieved by catching the return value in cloud server that is sent from the android app. The received value will be written into a file, which will be sent back to the raspberry pi by following polling or by implementing non-polling process.

Depending on the received status the audio file will be executed, and the voice result will be outputted. This voice implementation is achieved through "polly" service offered by AWS.

*All codes are attached in the same repository for reference.

The network overhead of the system is observed by using "bmon" and "tshark". For these purposes install bmon and tshark. The opencv acts as a software for face recognition on the system whereas the Amazon API is used as an external software for recognition purposes. There are slight differences between these two methods which will be discussed a head in the report. A new feature for the project is adding a audio note which may contain a password to spell for authentication. This can be achieved by installing speech_recognition and pydub.

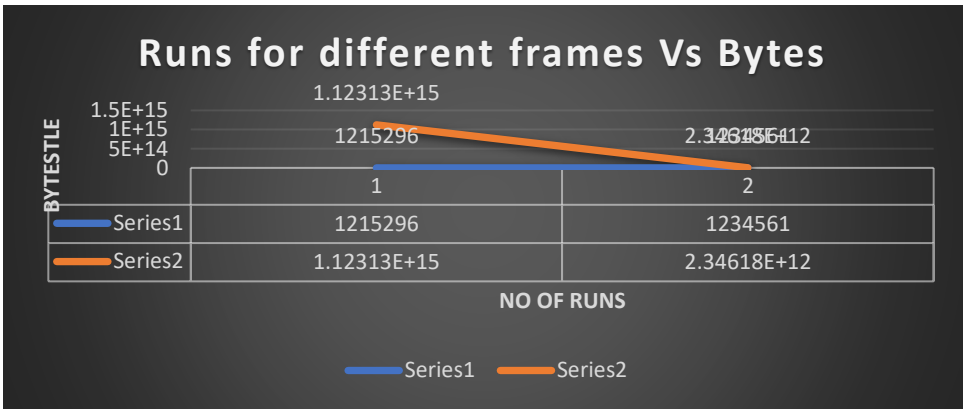
A new user can be added to the database with the permission of the owner using add_image.py program that takes image and name of the person as input.

A new feature is developed which is voice recognition. The voice recognition is implemented by using the google services and speech recognition package of python. If a audio jack is available, the voice can be generated and can be tested , whereas there is no audio jack with me, I have taken a predefined set of audio files are taken and are inputted to the program convert.py and is transferred to text format and is compared with the passwords that are stored in the databases. The audio files should be in .wav format in order to apply the program to it.

The entire process is shown through a small DIY tutorial. (Find the video in the repository).

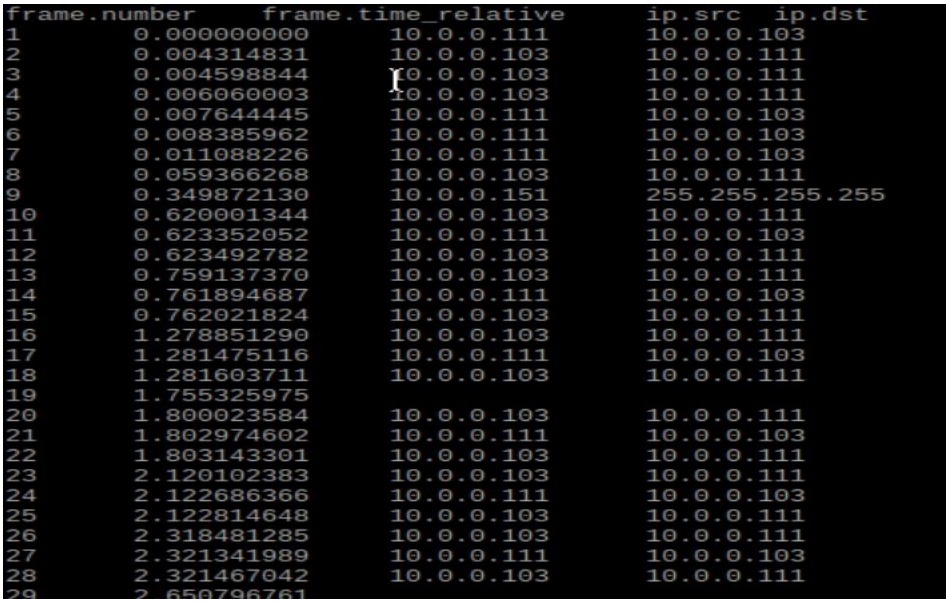
System Performance:

- Number of bytes that are transferred from pi to server and server to android and in reverse direction for one entire frame cycles is calculated. This total varies from one run to other and it totally depends on the number of frames for each cycle and the quality of image transferred at that point of time. For example, in one run, the number of bytes transferred for 5 frames in the entire process is 1215296bytes.



- The factors that affect network performance may be Bandwidth, Throughput, latency, Jitter, Error rate. In this project, I am using tshark and bmon processes to understand the bandwidth role in network performance. Install tshark and bmon in raspberry pi before calculating the bandwidth.
-

- From the above image, one can infer the number of bytes that are send per second from receiver and as well as transmitter. The MTU of the packets, overhead of the network everything can be found out using the bmon.
- Using tshark, the time taken to transfer the frames from source to destination can be figured out.s



frame.number	frame.time_relative	ip.src	ip.dst
1	0.000000000	10.0.0.111	10.0.0.103
2	0.004314831	10.0.0.103	10.0.0.111
3	0.004598844	10.0.0.103	10.0.0.111
4	0.006060003	10.0.0.103	10.0.0.111
5	0.007644445	10.0.0.111	10.0.0.103
6	0.008385962	10.0.0.111	10.0.0.103
7	0.011088226	10.0.0.111	10.0.0.103
8	0.059366268	10.0.0.103	10.0.0.111
9	0.349872130	10.0.0.151	255.255.255.255
10	0.620001344	10.0.0.103	10.0.0.111
11	0.623352052	10.0.0.111	10.0.0.103
12	0.623492782	10.0.0.103	10.0.0.111
13	0.759137370	10.0.0.103	10.0.0.111
14	0.761894687	10.0.0.111	10.0.0.103
15	0.762021824	10.0.0.103	10.0.0.111
16	1.278851290	10.0.0.103	10.0.0.111
17	1.281475116	10.0.0.111	10.0.0.103
18	1.281603711	10.0.0.103	10.0.0.111
19	1.755325975		
20	1.800023584	10.0.0.103	10.0.0.111
21	1.802974602	10.0.0.111	10.0.0.103
22	1.803143301	10.0.0.103	10.0.0.111
23	2.120102383	10.0.0.103	10.0.0.111
24	2.122686366	10.0.0.111	10.0.0.103
25	2.122814648	10.0.0.103	10.0.0.111
26	2.318481285	10.0.0.103	10.0.0.111
27	2.321341989	10.0.0.111	10.0.0.103
28	2.321467042	10.0.0.103	10.0.0.111
29	2.650796761		

- Differences between face recognition on raspberry pi and in internet through API.

  Face Recognition in Raspberry Pi:

  OpenCV is used for face recognition in this implementation.

  Advantages:

  Adding new user is easy and fast.

  As the methods and process presents in the system, there is no need of updating any test images to the remote servers that decreases the system processing and time.

  The cost for hardware and software is also less.

  Disadvantages:

  The efficiency of result is less than the other methods.

  The number of times the correct output results is also very less than the built in API's. This is because the OpenCV creates a rectangle box for performing face recognition. So, when a person stands out of range for camera then the box may not contain the exact face, which results in errors. So, this dependency on the size of the input image is main disadvantage of the opencv.

  Face Recognition in internet through API:

  In this project, we are using Amazon AWS web service for the purpose. The "Rekognition" service of the AWS is used for the purpose.

  Advantages:

  The results obtained from the API are more efficient than the OpenCV.

  Like OpenCV, there is no dependency on the size of the input image for recognition process.

  One can maintain their own database with defined users.

  Disadvantages:

  As to maintain the database in some remote cloud server, the processing time and memory used to process is more.

  Adding a user takes a little time and a bit difficult than the OpenCV.

- The number of frames that can be sent from pi to android can be changed for different runs. For each cycle, the time taken for all the frames to reach the destination is calculated. By the trial and error method, the fastest and slowest frame rate can be found out. The highest frame rate can also be found out. The highest in my case is 15 frames/cycle. As the number of frames increases per cycle, the more time it will take for the processing. To meet the frame rate with desired time, the quality of the images should be decreased.

- Non-polling is implemented by using sockets. A socket is exclusively used for receiving the authentication status file. The connection will be opened till the server sends an authentication file and once it is reached the socket gets closed.

- Differences between polling and non-polling:

  Polling:

  The result obtained from this process is used for other methods to obtain different functionalities.

  The entire processor is dedicated to one task.

  The raspberry pi needs to continuously pull the server for authentication file which may consume the memory and processing time of the raspberry pi.

  The number of pull requests are more for one single file.

  If there is any polling with some interval is applied, even though the pi receives the authentication file, it needs to wait till the interval time for going to the next step.

  In my implementation, the pi searches its directory for authentication file, if it receives the file from server it breaks the polling process and goes to the next step. For next cycle, it again checks for the authentication file in its directory, as it contains the previous authentication file it may miss assumes that it is the one that is looking for and doesn't perform any polling process. To avoid this condition, before performing the next cycle, the authentication file must be deleted in pi as well as in server, which is a tedious process.

  Non-polling:

  There is no need of any wait time by raspberry pi to obtain the authentication file.
  Only the connection needs to be created by pi and the rest of all the work is performed by the server itself.
  If the server missed sending the file, it may lead to deadlock.
  The memory utilized by the pi for this propose is less than the polling process.

What will you learn?

From start point till the end everything is a learning in this project.

- One can get clear understanding on what is raspberry pi and how is it useful and how does it work.
- How a webcam can be connected to raspberry pi to capture images or videos.
- How to find out ip address of a device when connected remotely.

- About what is VL53L0X sensor and how it is useful
- How to use online web services for different purposes like AWS Rekognition for face recognition, EC2 instance for storing the data remotely.
- Socket programming that helps in sending and receiving the data.
- How to perform audio recognition
- How to develop an android application
- Coding skills can be developed.
- One can learn how to analysed the network performance based on the data rate used.
- Will learn about the concepts of polling and non-polling and their advantages as well as disadvantages.
- How to convert speech to text and from text to speech.
- Throughout the implementation there are several packages, functions, methods, processes that are used One can go deep into them to understand their purpose.

Conclusion:

The system designed is good for authentication of the home. Even the system has few tedious process that needs to be performed by the owner to authenticate the user properly, the authentication is quite efficient and accurate.