

Smart Security Camera

CS 480I/580I Project

Rohit Dhuri

Computer Science

Binghamton University

New York

Rdhuri1@gmail.com

ABSTRACT

Traditional security cameras have been around for decades. In all these years these cameras have not made any significant technological progress. Atleast the ones which are available for cheap come into this segment. This project aims to create a device that can add significant features to a security camera. So that the 'smart' keyword suits it. The device will come bundled with an android application through which you can get notified about the guests in several ways. On which, you can also choose to monitor through the live feed. This provides a lot more simplified information to the user than a typical security camera. The luxury of having an android application which is connected to your camera gives this device an edge over the typical security cameras.

KEYWORDS

Raspberry Pi, android, face recognition, notifications, video streaming.

1 Introduction

The project aims to create a device which will constantly be aware of who is in front of the camera and try to recognize them. Also notify the respective owner about the activities. Here the camera needs to be connected to a secure network so that it can't be hacked. This project will be passed on a Raspberry Pi. Once we start the script on the Raspberry Pi we can disconnect all the cables or devices connected to it except for the power cable and set up the device anywhere we like. The device will keep monitoring its environment. Whenever something moves in front of it a record is mad into the database. After which if it is able to find a face, it will try to recognize it. If the face is unknown the user will be notified and a log would be made and same for if a registered user is detected. User can login into the app and choose to get or stop notifications and view or not view the live video stream from the camera.

2 System Design and Implementation

The features in android application can be used only after you register with the system or log into the system. Google firebase has been used for authentication in this project. After logging in, the user can view an interface with a dedicated button for all the features. Then the user can click on the 'Get notifications' button to start receiving notifications from the device. After this, if any activity takes place in front of the camera an entry will be added to the movement log. Which the user can view using the movement log button. If there is a person in front of the camera, the user will be notified of the same. The user view to monitor the live stream base on this information. Any number of users can be registered as known users.

2.1 Face Recognition

The face recognition script is written in python3 which uses the 'face_recognition' library to sense number of faces in an image and then identify them. There can be any number of registered users in the system. So, the algorithm first gets an instance of the camera and then loads the registered faces. It then tries to find and identify the faces in each frame. Upon finding a face it makes an entry into the firebase realtime database. This python script runs in the Raspbian operating system which is installed on the Raspberry Pi. The python script also makes us of several other libraries like the picamera, GPIO, os, sys and time. There is also a feature in the same script which starts the streaming. When a variable names 'stream' is updated in the database. The realtime video stream is a http stream which is recorded using the raspivid tool.

2.2 Google Firebase

All data is stored in the firebase realtime database. Whenever a entry has to made in the movement or the faces log a record is created in the subdirectory of the corresponding log. Every entry is associated with a timestamp. With every write into the faces directory

the user is notified using an android application notification.



The stream variable is used to specify whether to live stream video or continue monitoring. As video streaming and monitoring at the same time is not possible due to only one camera resource. The process is forked into a child process which streams the live feed and then relaunches to original script when stream is set to 'off'.

2.3 Android Application

The android application was created in android studio using java. The application is developed for android version 'O'. It has been tested on the android virtual device emulator and then finally on the actual device.

```
public class MainActivity extends AppCompatActivity {

    private Button logout, face_log, movement_log, get_not, stop_not, live_feed;
    private EditText name;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        FirebaseDatabase.getInstance().getReference().child("App").child("stream").setValue("off");

        logout = findViewById(R.id.logout);
        face_log = findViewById(R.id.face_log);
        movement_log = findViewById(R.id.movement_log);
        get_not = findViewById(R.id.get_notifications);
        stop_not = findViewById(R.id.stop_notifications);
        live_feed = findViewById(R.id.live_feed);

        logout.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                FirebaseAuth.getInstance().signOut();
                Toast.makeText(MainActivity.this, "logged out!", Toast.LENGTH_SHORT).show();
                stopService(new Intent(MainActivity.this, MyService.class));
                startActivity(new Intent(MainActivity.this, StartActivity.class));
            }
        });
    }
}
```

Figure 1: MainActivity.java

The android project is connected to the firebase authentication and realtime database. When the user open the application for the first time they will be asked to either register or login. After registration or login they land on the main page/home page from where they can access all the features of the application.

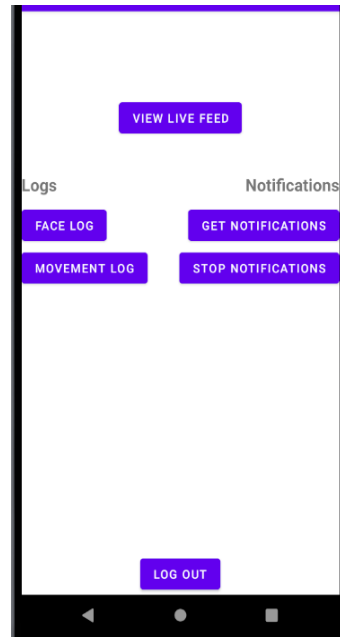


Figure 2: Home Page

After clicking the get notifications a background service will generate notifications based on writes in the database, even when the user is not using the application or when the application is not in the foreground they will continue receiving notifications.

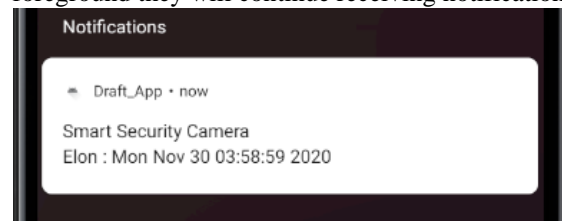


Figure 3: Notification example

3 Used Software

I have used Java for developing the android application, python3 for the face recognition script and also for data retrieval from the sensors. Thonny was used as an IDE for python programming on the Raspberry Pi. Python script is retrieving the data from the PIR motion sensor and the camera sensor and also the work of face recognition. The PIR motion sensor triggers the face recognition script to start. The raspivid tool is used for video streaming along with the vlc software on the Raspberry Pi.

4 Evaluation

This application serves to the aim of providing the user with a powerful tool for monitoring their

privacy. It will also help the users in case of a burglary, theft or any crime. The device comes with features like face recognition, instant notifications and live video streaming which are not available in mainstream security cameras.

5 Conclusion

The project is built on the idea of providing additional security features to the user to avoid any loss in case of a theft or burglary. The face recognition algorithm proves to be a key feature which recognizes people and tells the user about the guest.

REFERENCES

- [1] <https://raspberrypi-projects.com/pi/pi-hardware/raspberry-pi-camera/streaming-video-using-vlc-player>
- [2] <https://pypi.org/project/face-recognition/>
- [3] <https://firebase.google.com/docs/database>