HOME AUTOMATION USING ARDUINO BY FACIAL RECOGNITIONAND BLUETOOTH MODULE

Subhasish Das,16BCE0435,VIT
Kusumakar Kashyap,16BCE0351,VIT
Vishwas Bhanawat,16BCE0429,VIT
Divyanshi,16BCE0340,VIT
Naresh.K,VIT,Faculty

ABSTRACT: Technology is a evergoing process. We must be able to design a product using the technology that will be beneficial to the lives of others is a huge contribution to the community. This project gives the implementation and design of a feasible yet flexible and secure cell phone based home automation system. This design is based on aArduino BT board and the home appliances are connected to the input/ output ports of this board via various components. The communication between the cellular phone and the Arduino board is via Electromagnetic waves. It is designed to be low cost and scalable allowing variety of devices to be controlled while keping its basic functionality same. Protection by password is being used to only allow authorization of users so that they can access the appliances at home.

Privacy and Security are two universal rights and, to ensure that in our daily life we are secure, a lot of research is going on in the field of home security, and IoT is the turning point for the industry, where we connect everyday objects to share data for our betterment. Facial recognition is a well-established process in which the face is detected and identified out of the image.

INTRODUCTION

In today's world of connectivity and smart devices There is an urgent need to modify our existing day to day objects in today's world of connectivity and smart devices and make them smart. It is not the era when we can blindly trust the old and conventional security measures, specifically speaking is our door locks. We need to eliminate its existing drawbacks and add extra functionality to change and modernize any object

Anyone can open a conventional door lock by duplicating or stealing the key major drawbacks in a common door lock. Its simply impossible if we want our friends and family to enter our house, without being actually present over there. Thus we should seek to eliminate this problem. We need to modify the door to simply convert this normal door lock into a smart lock, which can open the door whenever we turn up in front of the gate or when we want it to open up for someone else without being physically present. It is the era where devices can interact with its users and at the same time ensure of their safety and keep improvising themselves.

Advanced knowledge of micro controllers and interfaces, as the Raspberry Pi computing device is used and interfaced with different drivers along with application development to develop a desktop application are the major concepts used to design and model this access control system. Live high quality HD Camera is connected with the display using the same processor to provide the functionalities. By capturing the photo and processing through the app which detects face out of the image and sends it over the Microsoft FACE API for recognition. Additionally differentIoT protocols and APIs have been used to make the device smart.

code is generated by an intensive study of opencv platform and its inbuilt libraries , which does correct and reliable facial recognition with new and efficient use of hardware. In this progressive world people are scared about the safety of their possessions, information and themselves. a profound impact is expected on the security industry with the model of Smart Door. It is somewhat anticipated as the time has come to make all daily life objects interconnected and interactive. This model will be a major contribution to the field of Home Security.

Recent trends have seen a dawn of work being done in fields of Computer Science like Artificial Intelligence, Machine Learning, Neural Networks, IOT, Big Data Analytics all with a common goal to make innovation make life easier and make optimum utilization of science and technology.

There has been a need that is felt in digitalizing conventional security tools and thereby exhaustive work has been done on making daily life locks smart by introducing locks movable with the help of stepper motors and/or adding a digital number pad to take inputs from user or adding Infra Red or Bluetooth modules to operate these devices.

An intensive study of literatures implementing Smart Locks had been done and literatures implementing Door Locks with the help of GSM phones and Stepper motors have been studied. Also literatures regarding smart display have been thoroughly reviewed. The fault in existing models is complexity of system and unnecessarily relying on extra components. Our model is unique with its one of a kind combination of functionalities offered and the simplicity of the model. A major difference is in the overhead reduction by the application as it detects the face out of the image and sends it directly to application program interfaced with our application, which has not been provided in any existing model and the efficient use of solenoids, which also eliminates use of stepper motors. So, we have avoided the use of unnecessary components like stepper motors and drivers as done in existing models and also we have given newer and unprecedented features of facial recognition as an access point control system with a combination of relay module with solenoid to open the gate and unique and interactive User Interface. Also rather than using a low quality Raspberry Pi Interfaced Camera we have used cellular camera of specific requirements to do efficient and reliable facial recognition.

The objectives is to implement a working model of a smart door and to give solutions to the problem faced by people in day to day incidents of burglary or losing the key and also to promote and ignite the work being done on IOT systems and implementing it with the help of key research areas of Neural Networks and IoT APIs and protocols.

This project uses the phone camera to recognize and authenticate the users in the database. Once authenticated the user can command any smart device connected to a bot to work as specified.

This model is allowing people to add more functionality to it and thus induce more research work in the field of AI, Machine Learning, IoT and lot more.

METHODOLOGY

We can segregate the whole system on basis of two major functionalities:

1) Accessing door on basis of recognition: the HD camera captures the photo on pressing the door bell. Then the application developed detects the face out of the image and crops it out and sends it over to the Microsoft Face API interfaced to the application through Microsoft Azure cloud set up. The face is identified and recognized from a pre-saved database of facial images on cloud. User gets an audio of “welcome USER NAME” if the face is matched. The processor controlling the relay module opens up the door, which can be observed by movement of solenoid.

2) Adding users to a database to be recognized from: We need to click on add user button to add users. On doing this, the cameraattached captures the image of user sitting in front and asks for name as input. It then adds it to the database of images over the cloud from which the face will be recognized.

CODE:

TRAINING OF FACES

```
private booleantrainFaces() {
if (images.isEmpty())
return true;
if (mTrainFacesTask != null &&mTrainFacesTask.getStatus() != AsyncTask.Status.FINISHED) {
Log.i(TAG, "Training Faces is still running");
return false;
    }
    Mat imagesMatrix = new Mat((int) images.get(0).total(), images.size(), images.get(0).type());
for (inti = 0; i<images.size(); i++)
images.get(i).copyTo(imagesMatrix.col(i));
Log.i(TAG, "Images height: " + imagesMatrix.height() + " Width: " + imagesMatrix.width() + " total: " + imagesMatrix.total());


if (useEigenfaces) {
Log.i(TAG, "Training faces");
showToast("Training " + getResources().getString(R.string.eigenfaces), Toast.LENGTH_SHORT);

mTrainFacesTask = new NativeMethods.TrainFacesTask(imagesMatrix, trainFacesTaskCallback);
    } else {
Log.i(TAG, "Training faces");
showToast("Training " + getResources().getString(R.string.fisherfaces), Toast.LENGTH_SHORT);

        Set<String>uniqueLabelsSet = new HashSet<>(imagesLabels);uniqueLabels = uniqueLabelsSet.toArray(new
String[uniqueLabelsSet.size()]);
int[] classesNumbers = new int[uniqueLabels.length];
for (inti = 0; i<classesNumbers.length; i++)
classesNumbers[i] = i + 1;
int[] classes = new int[imagesLabels.size()];
for (inti = 0; i<imagesLabels.size(); i++) {
            String label = imagesLabels.get(i);
for (intj = 0; j <uniqueLabels.length; j++) {
if (label.equals(uniqueLabels[j])) {
                    classes[i] = classesNumbers[j];
                    break;
                }
            }
        }


Mat vectorClasses = new Mat(classes.length, 1, CvType.CV_32S);vectorClasses.put(0, 0, classes);
mTrainFacesTask = new NativeMethods.TrainFacesTask(imagesMatrix, vectorClasses, trainFacesTaskCallback);
    }
mTrainFacesTask.execute();

return true;
}
```

DETECTION AND DISTANCE MEASURE

```
public void onMeasureDistComplete(Bundle bundle) {
if (bundle == null) {
showToast("Failed to measure distance", Toast.LENGTH_LONG);
return;
    }

float minDist = bundle.getFloat(NativeMethods.MeasureDistTask.MIN_DIST_FLOAT);
if (minDist != -1) {
intminIndex = bundle.getInt(NativeMethods.MeasureDistTask.MIN_DIST_INDEX_INT);
float faceDist = bundle.getFloat(NativeMethods.MeasureDistTask.DIST_FACE_FLOAT);
if (imagesLabels.size() >minIndex) { // Just to be sure
Log.i(TAG, "dist[" + minIndex + "]: " + minDist + ", face dist: " + faceDist + ", label: " + imagesLabels.get(minIndex));

            String minDistString = String.format(Locale.US, "%.4f", minDist);
            String faceDistString = String.format(Locale.US, "%.4f", faceDist);

if (faceDist<faceThreshold&&minDist<distanceThreshold) // 1. Near face space and near a face class
showToast("Face detected: " + imagesLabels.get(minIndex) + ". Distance: " + minDistString, Toast.LENGTH_LONG);
else if (faceDist<faceThreshold) // 2. Near face space but not near a known face class
showToast("Unknown face. Face distance: " + faceDistString + ". Closest Distance: " + minDistString, Toast.LENGTH_LONG);
else if (minDist<distanceThreshold) // 3. Distant from face space and near a face class
showToast("False recognition. Face distance: " + faceDistString + ". Closest Distance: " + minDistString,
Toast.LENGTH_LONG);
else // 4. Distant from face space and not near a known face class.
showToast("Image is not a face. Face distance: " + faceDistString + ". Closest Distance: " + minDistString,
Toast.LENGTH_LONG);
        }
    } else {
Log.w(TAG, "Array is null");
if (useEigenfaces || uniqueLabels == null || uniqueLabels.length>1)
showToast("Keep training...", Toast.LENGTH_SHORT);
else
showToast("Try Different Position", Toast.LENGTH_SHORT);
    }
}
```

REFERANCES

[1] Jie-Ci Yang et. all An Intelligent Automated Door Control System Based on a Smart Camera

[2] S. NazeemBasha et all An Intelligent Door System using Raspberry Pi and Amazon Web Services IoT

[3] ChaitanyaRane Password Based Door Locking System Using GSM

[4] PratikshaMisalet.all Door Locking/Unlocking using SMS Technology with GSM/GPRS technology

[5] Ushie James Ogrietall DESIGN AND CONSTRUCTION OF DOOR LOCKING SECURITY SYSTEM USING GSM

[6] Facial Recognition Application – Kristian Lauszushttps://github.com/Lauszus/FaceRecognitionApp