A project report on

SMARTPHONE CONTROLLED ROBOT USING ARDUINO AND ANDROID

Submitted in partial fulfilment for the award of the degree of

Bachelor of Technology (IT)

by

ROHAN SINHA (14BIT0297)

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Information Technology & Engineering (SITE)

April, 2018

SMARTPHONE CONTROLLED ROBOT USING ARDUINO AND ANDROID

Submitted in partial fulfilment for the award of the degree of

Bachelor of Technology (IT)

by

ROHAN SINHA (14BIT0297)

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Information Technology & Engineering (SITE)

April, 2018

DECLARATION

I hereby declare that the thesis entitled “SMARTPHONE CONTROLLED ROBOT USING ARDUINO AND ANDROID” submitted by me, for the award of the degree of Bachelor of Technology (IT) VIT is a record of bonafide work carried out by me under the supervision of Prof. Shynu P.G.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Vellore

Date: 09.04.2018 Signature of the Candidate

CERTIFICATE

This is to certify that the thesis entitled "SMARTPHONE CONTROLLED ROBOT USING ARDUINO AND ANDROID**"** submitted by ROHAN SINHA (14BIT0297) School of information Technology & Engineering (SITE) VIT, for the award of the degree of B.Tech. in Information Technology is a record of bonafide work carried out by his under my supervision.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The Project report fulfils the requirements and regulations of VIT and in my opinion meets the necessary standards for submission.

-

Signature of the Guide

Signature of the HOD

Internal Examiner

External Examiner

ABSTRACT

Every face has numerous, distinguishable **landmarks**, the different peaks and valleys that make up facial features. FaceIt defines these landmarks as **nodal points**. Each human face has approximately 80 nodal points. Some of these measured by the software are:

- Distance between the eyes
- Width of the nose
- Depth of the eye sockets
- The shape of the cheekbones
- The length of the jaw line

These nodal points are measured creating a numerical code, called a **faceprint**, representing the face in the database.

In the past, facial recognition software has relied on a 2D image to compare or identify another 2D image from the database. To be effective and accurate, the image captured needed to be of a face that was looking almost directly at the camera, with little variance of light or facial expression from the image in the database. This created quite a problem.

The project is a Smartphone Controlled Arduino Robot and Facial Recognition System that can be used for many purposes like, Military Recon. The robot will be controlled with three different ways - Buttons, Accelerometer and Voice. All these ways will be applied using Android Applications. The applications will be called **SmartBot**, **AcceleroBot** and **VoiceBot**. The robot will also have a camera which will be used to take images in any situation (like hostage situation). These images will be used for facial recognition which will help the military/police in recognizing the criminals and also help them in understanding the threat. There are many algorithms for facial recognition in the market today like Artificial Neural Networks, Convolution Neural Network and also hybrid neural networks (Combination of 2 techniques). I will be using one of these techniques for my problem.

ACKNOWLEDGEMENT

It is my pleasure to express with deep sense of gratitude to Prof. Shynu P.G., Assistant Professor, SITE, Vellore Institute of Technology, for his constant guidance, continual encouragement, and understanding; more than all, he taught me patience in my endeavour. My association with him is not confined to academics only, but it is a great opportunity on my part of work with an intellectual and expert in the field of information technology.

I would like to express my gratitude to Dr. G. Viswanathan, Chancellor, VIT University, Dr. S. Narayan, Pro-Vice Chancellor, VIT University, Vellore Dr. Anand Samuel, Vice Chancellor, VIT University, and Dr. Aswani Kumar , School of Information Technology, for providing with a environment to work in and for his inspiration during the tenure of the course.

In jubilant mood, I express ingeniously my whole-hearted thanks to Prof. Shynu P. and all teaching staff and members working as limbs of our university for their not-self-centred enthusiasm coupled with timely encouragements showered on me with zeal, which prompted the acquirement of the requisite knowledge to finalize my course study successfully.

I would like to thank my parents for their support. It is indeed a pleasure to thank my friends who persuaded and encouraged me to take up and complete this task. Last but not the least, I express my gratitude and appreciation to all those who have helped me directly or indirectly toward the successful completion of this project.

Place: Vellore Name of the student

Date: 09.04.2018 **Rohan Sinha**

CONTENTS

LIST OF FIGURES

Chapter 1

Introduction

1.1 INTRODUCTION

Facial recognition is a classification of biometric programming that maps a person's facial highlights numerically and stores the information as a faceprint. The product utilizes profound learning calculations to look at a live catch or advanced picture to the put away faceprint keeping in mind the end goal to confirm a person's character. Amazing cameras in cell phones have made facial acknowledgment a feasible choice for validation and additionally distinguishing proof. Apple's iPhone X, for instance, incorporates Face ID innovation that gives clients a chance to open their telephones with a faceprint mapped by the telephone's camera. The telephone's product, which is outlined with 3-D demonstrating to oppose being mock by photographs or veils, catches and thinks about more than 30,000 factors. As of this written work, Face ID can be utilized to validate buys with Apple Pay and in the iTunes Store, App Store and iBooks Store. Apple scrambles and stores faceprint information in the cloud, yet validation happens specifically on the gadget.

The accompanying four-organize process outlines the way biometric frameworks work:

- **Capture** - a physical or behavioral example is caught by the framework amid enrolment.

- **Extraction** - novel information is extricated from the example and a layout is made

- **Comparison** - the format is then contrasted and another example

- **Matching -** the framework at that point chooses if the highlights separated from the new example are coordinating or not

At the point when the client faces the camera, remaining around two feet from it. The framework will find the client's face and perform matches against the asserted character or the facial database. It is conceivable that the client may need to move and reattempt the confirmation in view of his facial position. The framework normally goes to a choice in under 5 seconds

1.2 EIGENFACES

Eigenfaces is the name given to an arrangement of eigenvectors when they are utilized as a part of the PC vision issue of human face acknowledgment. The approach of utilizing eigenfaces for acknowledgment was produced by Sirovich and Kirby (1987) and utilized by Matthew Turk and Alex Pentland in confront characterization. The eigenvectors are gotten from the covariance framework of the likelihood conveyance over the high-dimensional vector space of face pictures. The eigenfaces themselves shape a premise set of all pictures used to build the covariance network. This produces measurement lessening by permitting the littler arrangement of premise pictures to speak to the first preparing pictures. Order can be accomplished by contrasting how faces are spoken to by the premise set.

Before creating eigenfaces, confront pictures are standardized to arrange the eyes and mouths, at that point all resampled at a similar pixel determination. Eigenfaces are then extricated out of the picture information by methods for primary part examination (PCA) in the accompanying way:

1. Given M face images with the size of $h \times w$, each image is transformed into a vector of size $D(=hw)$ and placed into the set

$$\{\Gamma_1, \Gamma_2, \cdots, \Gamma_M\}$$

The face images should be appropriately scaled and aligned, and the backgrounds (and possibly non-face areas such as hair and neck) should be constant or removed.

2. Each face differs from the average by the vector $\Phi_i = \Gamma_i - \Psi$, where the average face is defined by $\Psi = (1/M) * \sum \Gamma_i$.

3. The covariance matrix $\mathbf{C} \in \mathrm{R}^{D\times D}$ is defined as

$$\mathbf{C} = (1/M) * \sum \Phi_i \Phi^{\top}_i = \mathbf{A}\mathbf{A}^{\top},$$

where $\mathbf{A} = \{\Phi_1, \Phi_2, \cdots, \Phi_M\} \in \mathrm{R}^{D\times M}$.

4. Determining the eigenvectors of $\mathbf{C}$ is an intractable task for typical image sizes when $D \gg M$. However, to efficiently compute the eigenvectors of $\mathbf{C}$, one may first compute the eigenvectors of the much-smaller $M\times M$ matrix $\mathbf{A}^{\top}\mathbf{A}$. The eigenvector and eigenvalue matrices of $\mathbf{A}^{\top}\mathbf{A}$ are defined as

$$\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \cdots \mathbf{v}_r\}$$

and $\Lambda = diag\{\lambda_1, \lambda_2, \cdots \lambda_r\}$, $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_r > 0$, where r is the rank of $\mathbf{A}$. Note that eigenvectors corresponding to eigenvalues of zero have been discarded.

5. The eigenvalue and eigenvector matrices of $\mathbf{C}$ are Λ and $\mathbf{U} = \mathbf{A}\mathbf{V}\Lambda^{-1/2}$, where $\mathbf{U} = \{\mathbf{u}_i\}$ is the collection of eigenfaces.

Eigenface gives a simple and shabby approach to acknowledge confront acknowledgment in that:

- Its preparing process is totally programmed and simple to code.

- Eigenface enough diminishes measurable multifaceted nature in confront picture portrayal.

- Once eigenfaces of a database are ascertained, confront acknowledgment can be accomplished progressively.

- Eigenface can deal with extensive databases.

Be that as it may, the lacks of the eigenface strategy are likewise self-evident:

- Very touchy to lighting, scale and interpretation; requires a very controlled condition.

- Eigenface experiences issues catching articulation changes.

- The most noteworthy eigenfaces are essentially about enlightenment encoding and don't give valuable data with respect to the genuine face.

To adapt to brightening diversion by and by, the eigenface technique more often than not disposes of the initial three eigenfaces from the dataset. Since enlightenment is generally the reason behind the biggest varieties in confront pictures, the initial three eigenfaces will mostly catch the data of 3-dimensional lighting changes, which has little commitment to confront acknowledgment. By disposing of those three eigenfaces, there will be a fair measure of lift in exactness of face acknowledgment, yet different strategies, for example, Fisherface and Linear space still have the preferred standpoint.

1.3 OVERVIEW

The project is a Smartphone Controlled Arduino Robot and Facial Recognition System that can be used for many purposes like, Military Recon. The robot will be controlled with three different ways - Buttons, Accelerometer and Voice. All these ways will be applied using Android Applications. The robot will also have a camera which will be used to take images in any situation (like hostage situation). These images will be used for facial recognition which will help the military/police in recognizing the criminals and also help them in understanding the threat. There are many algorithms for facial recognition in the market today like Artificial Neural Networks, Convolution Neural Network and also hybrid neural networks (Combination of 2 techniques). I will be using one of these techniques for my problem.

1.4 OBJECTIVE

This endeavour is aimed at innovating and developing a security device for Military which can be used in hostile situations

Chapter 2

Literature Survey

For Facial Recognition numerous procedures have been proposed in the current days. In these systems they have utilized diverse sorts of strategies and at times a mix of methods. In Image based Face Detection and Recognition, by: Faizan Ahmed, Aaima Najam and Zeeshan Ahmed, the creators have assessed different face identification and acknowledgment strategies, gave finish answer for picture based face location and acknowledgment with expanded precision, higher reaction rate as the starting advance for video observation. Five datasets been utilized for above trials. **1)** *Face gathering with plain green foundation; no head scale and light variety yet having minor changes in head turn, tilt, and incline, position of face and significant change in looks*. **2)** *Face accumulation with red blind foundation, variety is caused by shadows as subject pushes ahead, having minor changes in head turn, tilt and inclination; expansive head scale variety; some appearance variety, interpretation in position of face and picture lighting variety as subject advances, critical lighting changes happen on faces minute because of the manufactured lighting game plan.* **3)** *Face accumulation with complex foundation; huge head scale variety; minor varieties in head turn, tilt, inclination and demeanor; some interpretation in confront position and critical light variety in light of protest minute in manufactured light.* **4)** *Face accumulation with plain foundation; little head scale variety; impressive variety in head turn, tilt, inclination and real variety in demeanor; minor interpretation in confront position and light variety.* **5)** *Face accumulation with consistent foundation having minor head scale variety and light variety; immense variety thus, tilt, inclination, appearance and face position.*

In Face Recognition: A Literature Survey, by: W. Zhao, the creator gives an exceptional basic overview of still-and video-based face acknowledgment look.

Chapter 3

Design

3.1. SYSTEM ARCHITECTURE

The Car Robot has different hardware components like: Arduino, Motor Driver, Bluetooth Sensor and a Camera. The camera will be used to detect faces which are already trained in the computer system.

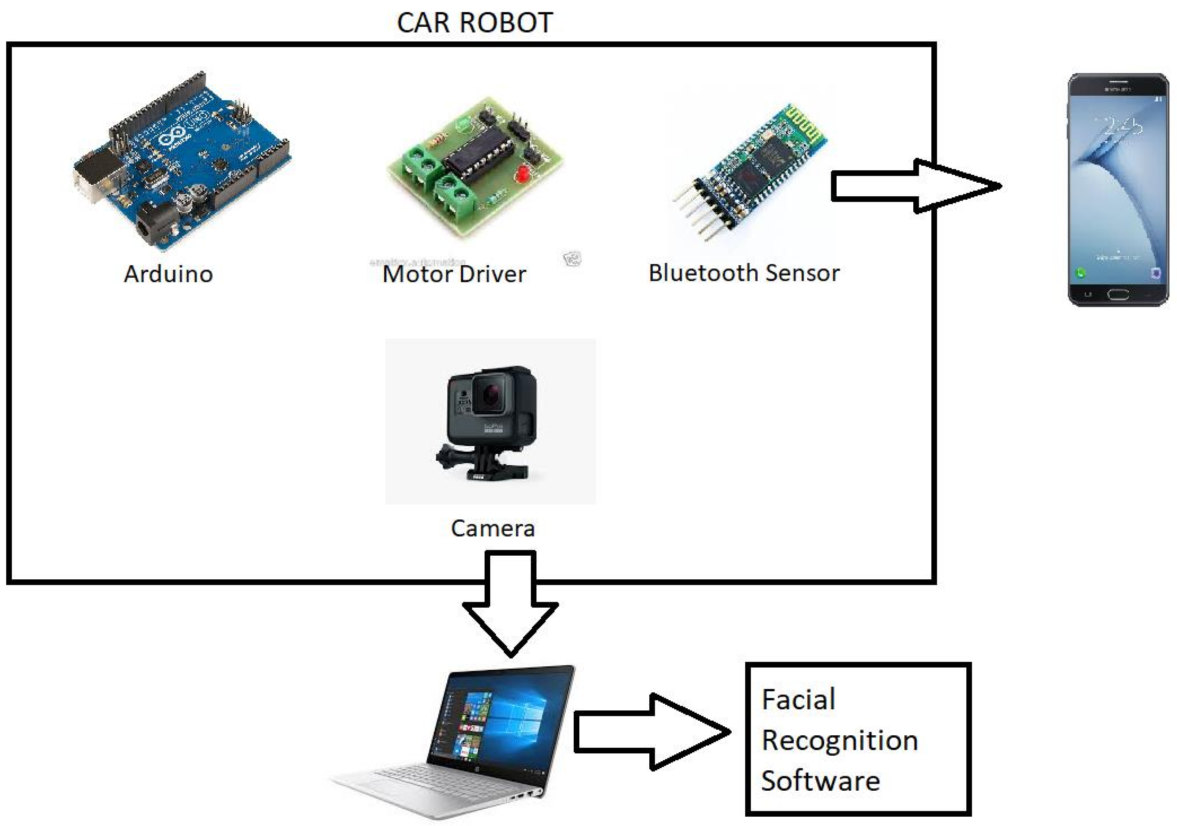


Fig 3.1: System Architecture

3.2. WORKING

The Arduino will be programmed in order to move the DC motors connected to the wheels of the car. Motor Driver will be used as the Arduino cannot provide enough power to run a DC motor. The Bluetooth Sensor will be used to control the car with the Android application in the mobile. The application will be sending signals to the Bluetooth sensor which will in turn send a signal to the Arduino which will move the wheels accordingly.

For example, if we press the “Forward” button in the application, the mobile will send the value ‘1’ to the Bluetooth Sensor which will forward it to the Arduino. The Arduino is coded in such a way that whenever it receives the value ‘1’, it will move forward.

The camera will be used to capture live video as the robot moves. Simultaneously, the software running on the laptop will be detecting faces present in the video (assuming we have trained the model with these faces). The programming language used is C#. In order to detect faces Haar Cascade Classifier is used.

Classifier is trained with a few hundred sample views of a particular object (i.e., a face, eyes or other object), called positive examples, that are scaled to the same size (say, 10x10), and negative examples - arbitrary images of the same size for training. The word “cascade” in the classifier name means that the resultant classifier consists of several simpler classifiers (stages) that are applied subsequently to a region of interest until at some stage the candidate is rejected or all the stages are passed. After a classifier is trained, it can be applied to a region of interest in an input image. An input image is given to check classifier outputs. If output is “1” then region is likely to show the object (i.e., face/eye) it means image matches with already trained images such as face, and “0” otherwise.

To search for the object in the whole image one can move the search window across the image and check every location using the classifier. The classifier is designed so that it can be easily “resized” in order to be able to find the objects of interest at different sizes, which is more efficient than resizing the image itself. So, to find an object of an unknown size in the image the scan procedure should be done several times at different scales.

3.3. CONNECTION

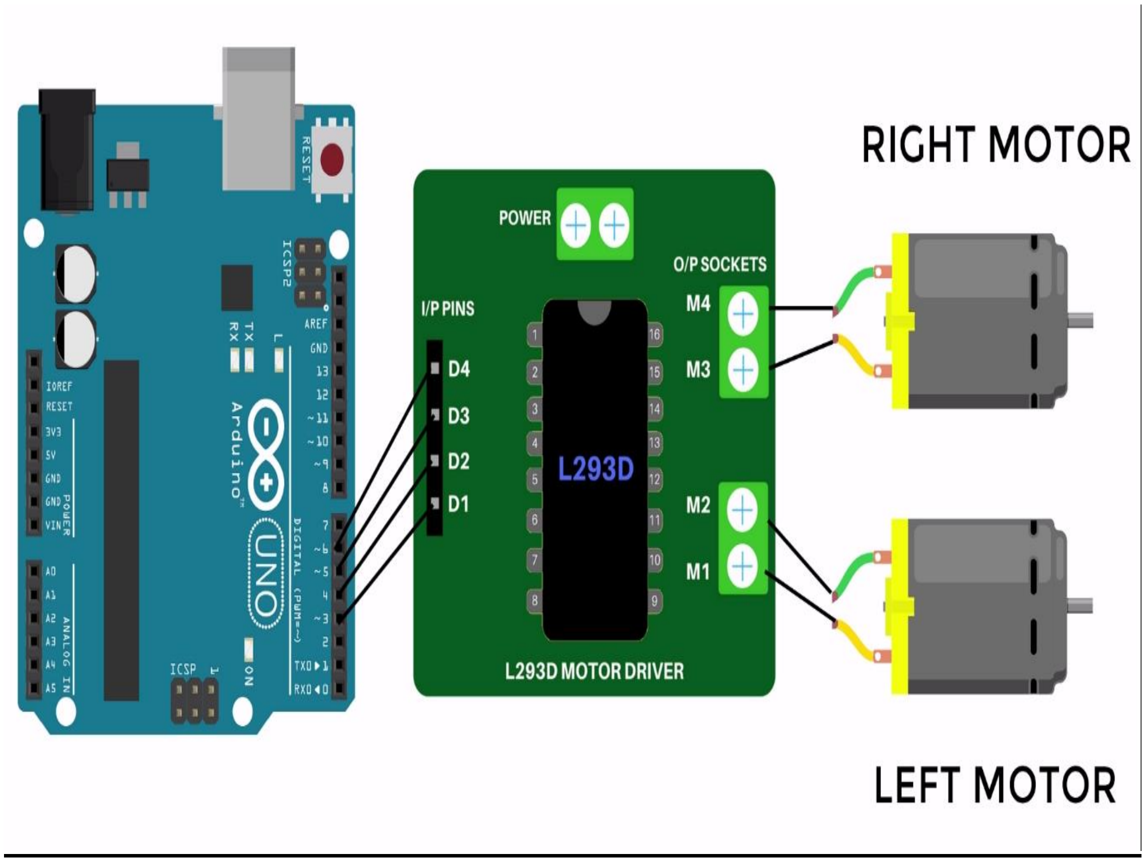


Fig 3.2: Arduino and Motor Driver Connection

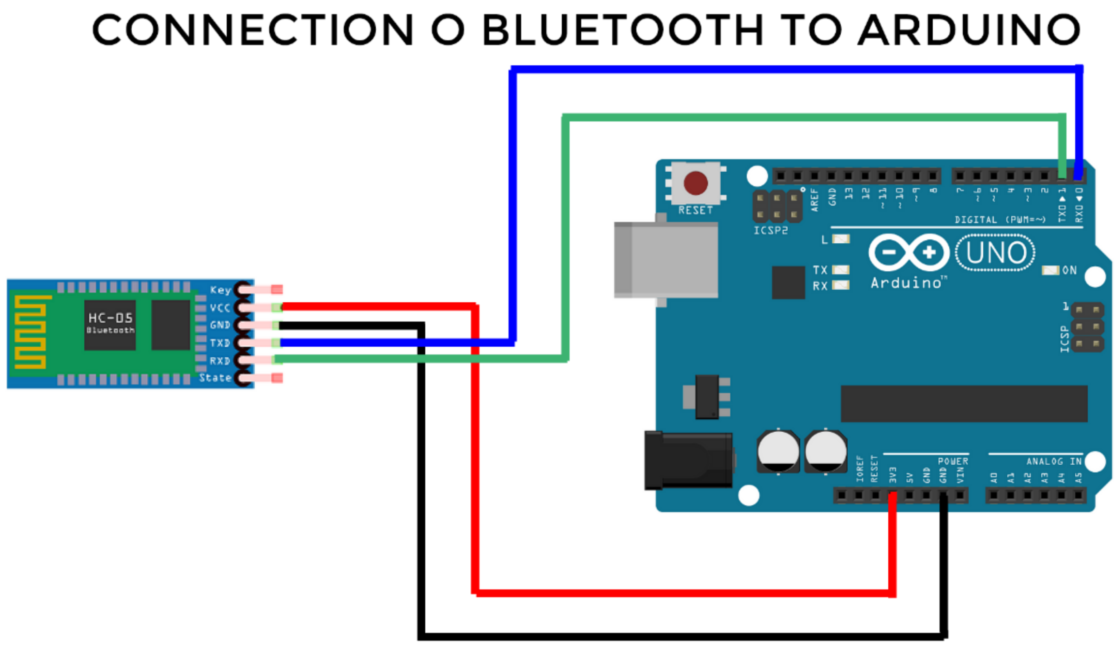


Fig 3.3: Arduino and Bluetooth Module Connection

3.4. HARDWARE COMPONENTS

- <u>Microcontroller</u> (Arduino):

  A microcontroller is a low cost computer that deals with specific tasks. A microcontroller is a small computer on a single integrated circuit which contains a core processor, Memory and Input/output peripherals. Microcontrollers are specially designed for embedded applications. Their main advantage is that they are small in size and less in cost. They can automatically control the other devices connected to it i.e., it can control the Buzzer, Transmitter, Receiver, Temperature sensors and automatic brake controllers. Arduino is an open source computer hardware and software company, project, and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical and digital world. The project's products are distributed as open-source hardware and software, which are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), permitting the manufacture of Arduino boards and software distribution by anyone.

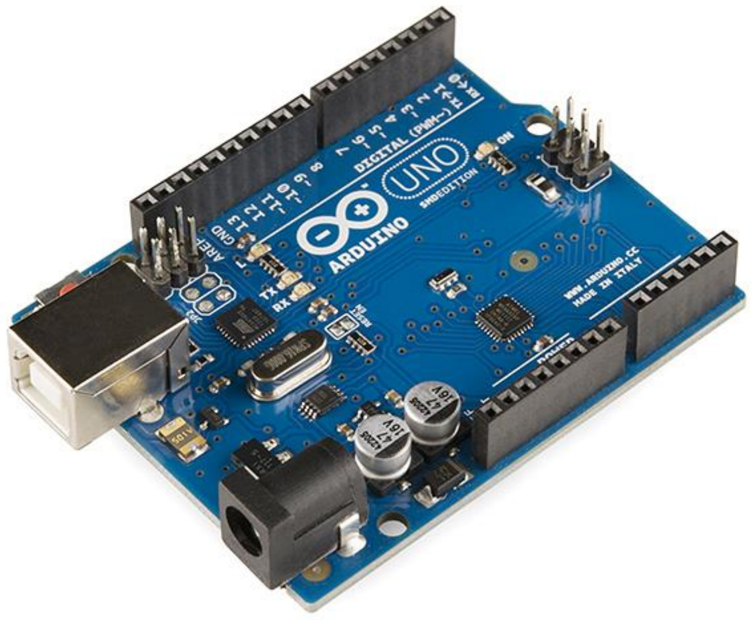


Fig 3.4: Arduino

- <u>L293D Motor Driver</u>:

  L293D contains two inbuilt H-bridge driver circuits. In its common mode of operation, two DC motors can be driven simultaneously, both in forward and reverse direction. The motor operations of two motors can be controlled by input logic at pins 2 & 7 and 10 & 15. Input logic 00 or 11 will stop the corresponding motor. Logic 01 and 10 will rotate it in clockwise and anticlockwise directions, respectively. The Arduino pins provide very low output current compared to what anything larger than a micromotor might take. Thus motor driver is used to amplify the current.

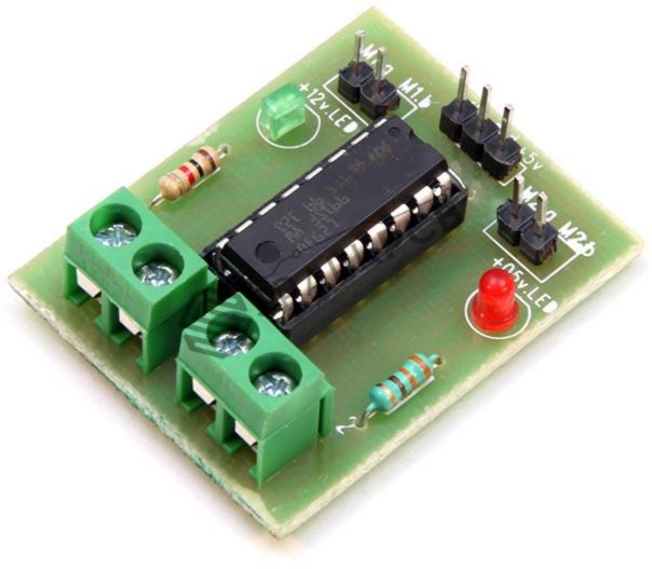


Fig 3.5: Motor Driver

- HC-05 Bluetooth Sensor:

  HC-05 module is a simple to utilize Bluetooth SPP (Serial Port Protocol) module, intended for straightforward remote serial association setup. The HC-05 Bluetooth Module can be utilized as a part of a Master or Slave design, making it an incredible answer for remote correspondence. This serial port Bluetooth module is completely qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with finish 2.4GHz radio handset and baseband. It utilizes CSR Bluecore 04-External single chip Bluetooth framework with CMOS innovation and with AFH (Adaptive Frequency Hopping Feature). It is utilized to interface the Arduino with the Android Application.

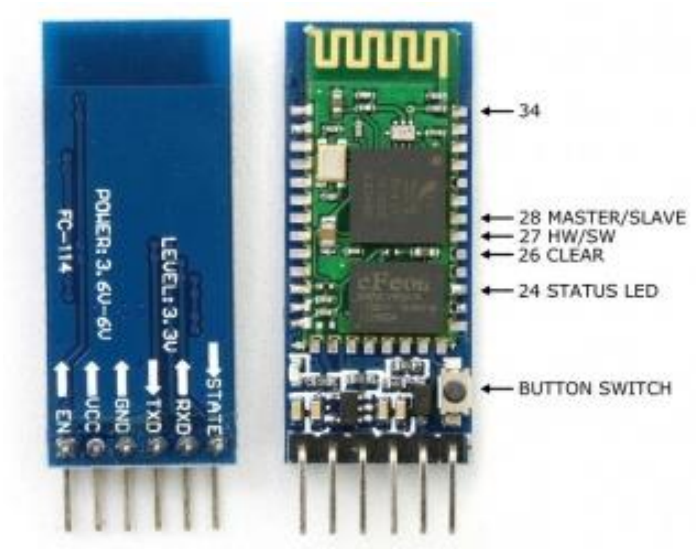


Fig 3.6: Bluetooth Module

- <u>Car Robot Chassis:</u>

  This is a basic car model that comes with 2 DC Motors that can be connected to the wheels and a base

Fig 3.7:Car robot chassis

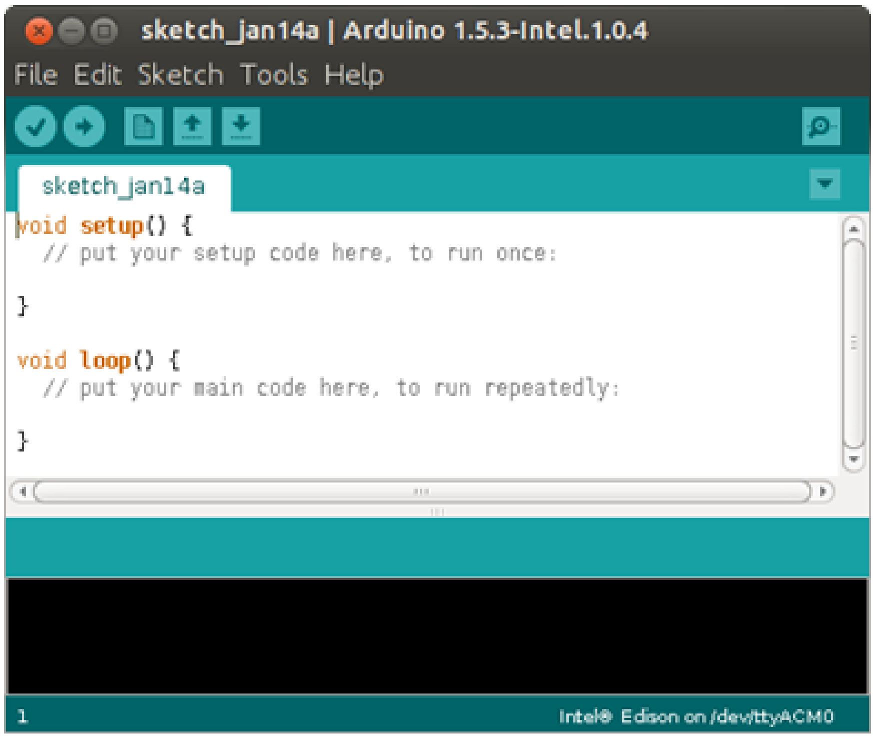
3.5. SOFTWARE COMPONENTS

- Arduino:

  Arduino software is used to program in Arduino which can be uploaded in the Arduino Micro Controller Board through a USB.

A minimal Arduino C/C++ program consist of only two functions:

1) ***setup()***: This function is called once when a sketch starts after power-up or reset. It is used to initialize variables, input and output pin modes, and other libraries needed in the sketch.
2) ***loop()***: After *setup()* has been called, function *loop()* is executed repeatedly in the main program. It controls the board until the board is powered off or is reset.


sketch_jan14a | Arduino 1.5.3-Intel.1.0.4
File Edit Sketch Tools Help
sketch_jan14a
1
Intel® Edison on /dev/ttyACM0

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Fig 3.8: Arduino Interface

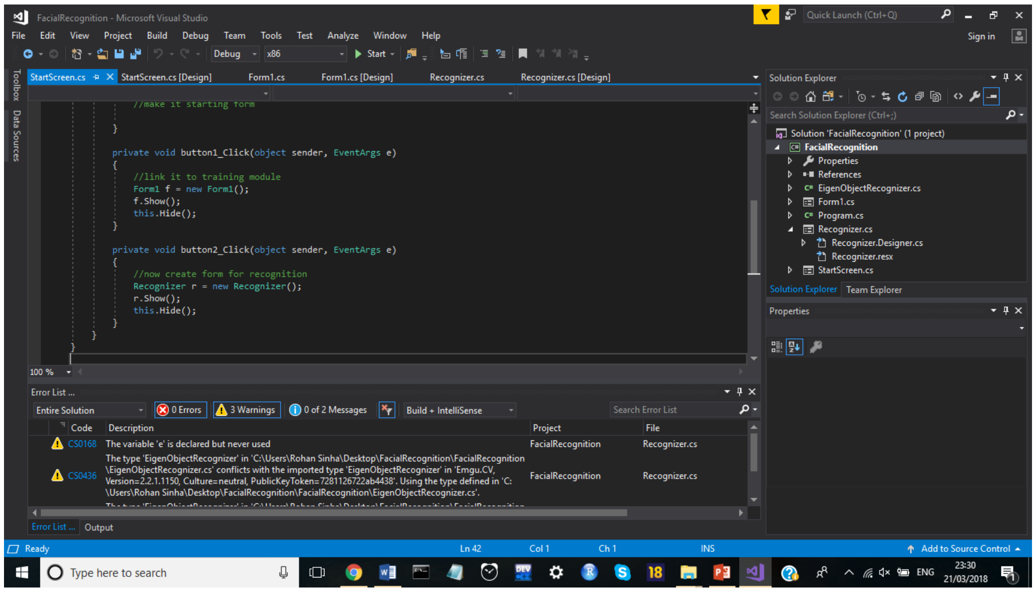
- MIT App Inventor 2:

  Application Inventor for Android is an open-source web application initially gave by Google, and now kept up by the Massachusetts Institute of Technology (MIT). It is utilized to create Apps which will be utilized to control the robot. t enables newcomers to PC programming to make programming applications for the Android working framework (OS). It utilizes a graphical interface, fundamentally the same as Scratch and the StarLogo TNG UI, which enables clients to relocate visual articles to make an application that can keep running on Android gadgets. In making App Inventor, Google drew upon noteworthy earlier research in instructive figuring, and also work done inside Google on online advancement conditions. Application Inventor and the undertakings on which it is based are educated by constructionist learning speculations, which accentuates that programming can be a vehicle for drawing in intense thoughts through dynamic learning. In that capacity, it is a piece of a progressing development in PCs and training that started with crafted by Seymour Papert and the MIT Logo Group in the 1960s and has likewise showed itself with Mitchel Resnick's work on Lego Mindstorms and StarLogo.

Fig 3.9: MIT APP INVENTOR LOGO

- <u>Microsoft Visual Studio:</u>

Microsoft Visual Studio is a coordinated advancement condition (IDE) from Microsoft. It is utilized to create PC programs, and additionally sites, web applications, web administrations and portable applications. Visual Studio utilizes Microsoft programming improvement stages, for example, Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can deliver both local code and oversaw code. The programming dialect used to create Facial Recognition Software is C#.


FacialRecognition - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug x86 Start
StartScreen.cs StartScreen.cs [Design] Form1.cs Form1.cs [Design] Recognizer.cs Recognizer.cs [Design]
Solution Explorer
Solution 'FacialRecognition' (1 project)
FacialRecognition
Properties
References
EigenObjectRecognizer.cs
Form1.cs
Program.cs
Recognizer.cs
Recognizer.Designer.cs
Recognizer.resx
StartScreen.cs
Properties
Ready Ln 42 Col 1 Ch 1 INS
Type here to search

```
                //make it starting form

            }

            private void button1_Click(object sender, EventArgs e)
            {
                //link it to training module
                Form1 f = new Form1();
                f.Show();
                this.Hide();
            }

            private void button2_Click(object sender, EventArgs e)
            {
                //now create form for recognition
                Recognizer r = new Recognizer();
                r.Show();
                this.Hide();
            }
        }
    }
```

Error List ... — Entire Solution — 0 Errors — 3 Warnings — 0 of 2 Messages — Build + IntelliSense

Code	Description	Project	File
CS0168	The variable 'e' is declared but never used	FacialRecognition	Recognizer.cs
CS0436	The type 'EigenObjectRecognizer' in 'C:\Users\Rohan Sinha\Desktop\FacialRecognition\FacialRecognition\EigenObjectRecognizer.cs' conflicts with the imported type 'EigenObjectRecognizer' in 'Emgu.CV, Version=2.2.1.1150, Culture=neutral, PublicKeyToken=7281126722ab4438'. Using the type defined in 'C:\Users\Rohan Sinha\Desktop\FacialRecognition\FacialRecognition\EigenObjectRecognizer.cs'.	FacialRecognition	Recognizer.cs

Fig 3.10: Microsoft Visual Studio

- <u>OpenCV:</u>

OpenCV (Open Source Computer Vision Library) is an open source PC vision and machine learning programming library. OpenCV was worked to give a typical framework to PC vision applications and to quicken the utilization of machine recognition in the business items. Being a BSD-authorized item, OpenCV makes it simple for organizations to use and alter the code.

The library has in excess of 2500 enhanced calculations, which incorporates an exhaustive arrangement of both exemplary and cutting edge PC vision and machine learning calculations. These calculations can be utilized to distinguish and perceive faces, distinguish objects, arrange human activities in recordings, track camera developments, track moving items, separate 3D models of articles, create 3D point mists from stereo cameras, fasten pictures together to deliver a high determination picture of a whole scene, find comparative pictures from a picture database, expel red eyes from pictures taken utilizing streak, take after eye developments, perceive view and build up markers to overlay it with increased reality, and so forth. OpenCV has in excess of 47 thousand individuals of client group and evaluated number of downloads surpassing 14 million. The library is utilized widely in organizations, look into gatherings and by administrative bodies.

Alongside entrenched organizations like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that utilize the library, there are numerous new businesses, for example, Applied Minds, VideoSurf, and Zeitera, that make broad utilization of OpenCV. OpenCV's conveyed utilizes traverse the range from sewing streetview pictures together, recognizing interruptions in reconnaissance video in Israel, observing mine hardware in China, helping robots explore and get objects at Willow Garage, discovery of swimming pool suffocating mishaps in Europe, running intelligent workmanship in Spain and New York, checking runways for flotsam and jetsam in Turkey, investigating marks on items in production lines far and wide on to quick face recognition in Japan.

It has C++, Python, Java and MATLAB interfaces and backings Windows, Linux, Android and Mac OS. OpenCV inclines for the most part towards continuous vision applications and exploits MMX and SSE directions when accessible. A full-highlighted CUDA and OpenCL interfaces are as a rule effectively grew at the present time.

There are more than 500 calculations and around 10 fold the number of capacities that make or bolster those calculations. OpenCV is composed locally in C++ and has a templated interface that works consistently with STL compartments.

3.6. MODEL

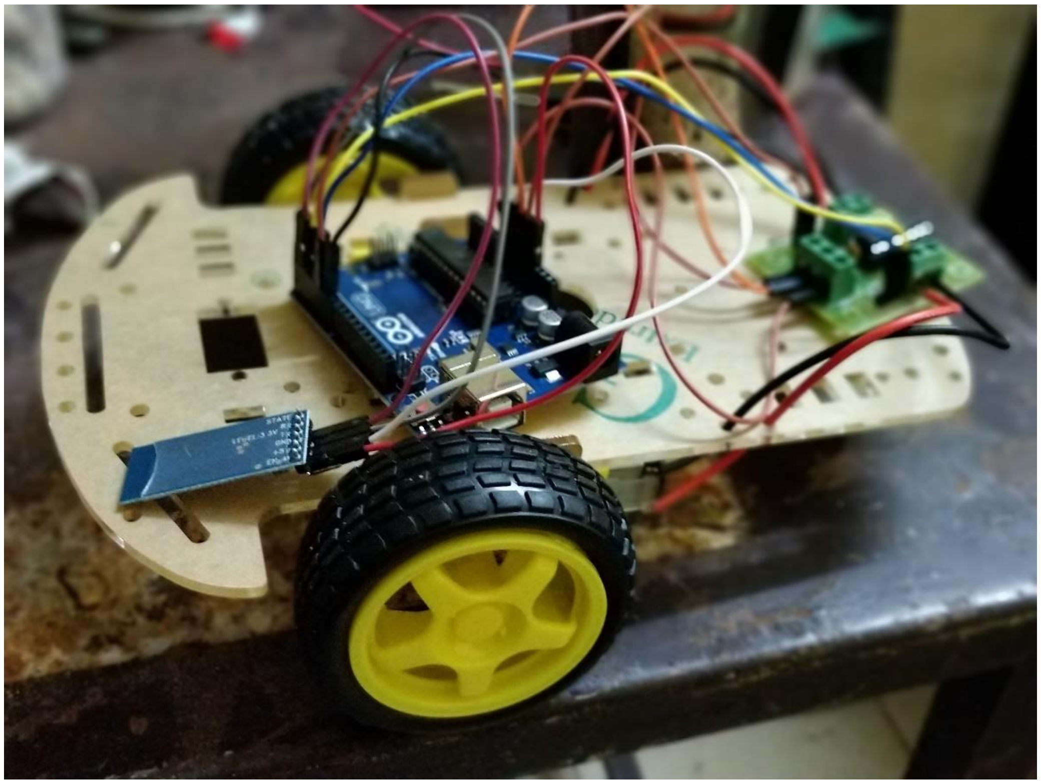


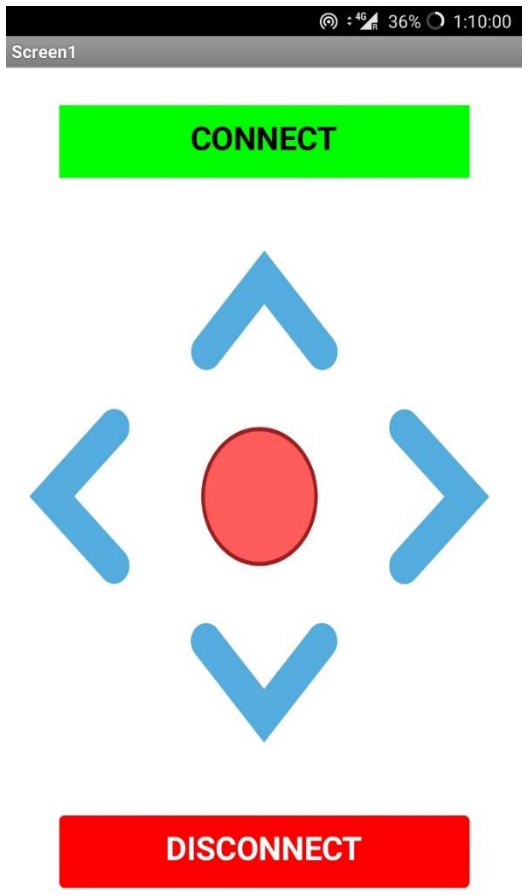
Fig 3.11: Car Model

Chapter 4

Interface

This chapter is about the interfaces of the Android Applications and The Facial Recognition Software.

4.1. ANDROID APPLICATIONS



.

Fig 4.1: SmartBot App Interface

The "CONNECT" button will connect the HC-05 Bluetooth module to the Smartphone and the arrow buttons will move the car in the respective directions. The Red button will stop the car. The "DISCONNECT" button will disconnect the connection.

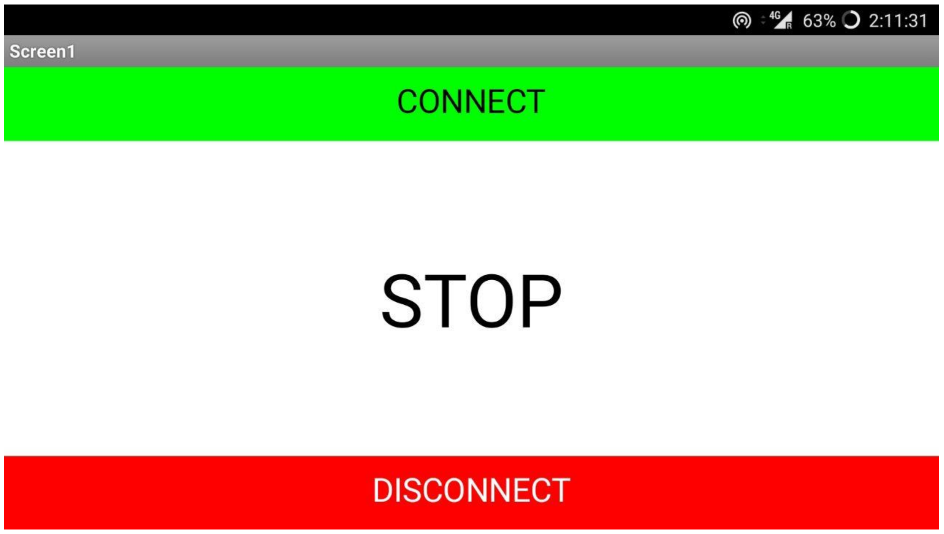


Fig 4.2: AcceleroBot App Interface

In this application (Fig.4.2) the car robot will move as the smartphone is tilted. This application uses the Smartphone's Accelerometer Sensor. Whenever we tilt the smartphone there is a change in X, Y and Z axis of the Smartphone. Depending on the change, the car will move.

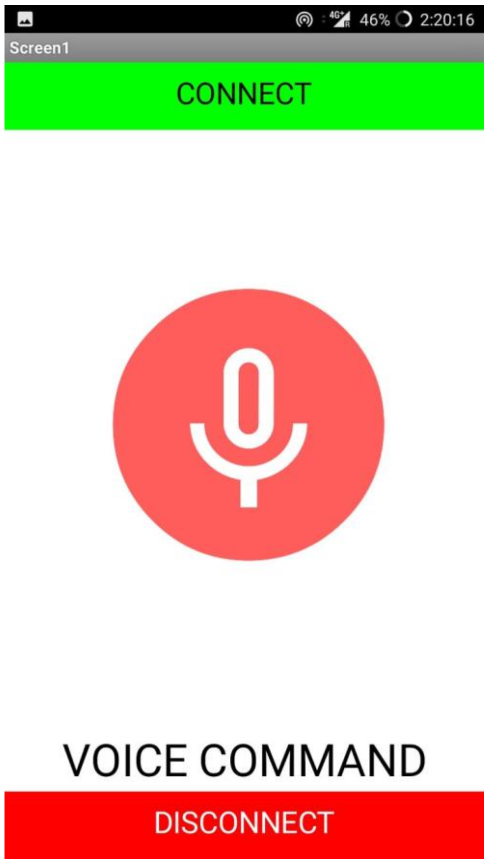


Fig 4.3: VoiceBot App Interface

In this application (Fig. 4.3), human voice commands will control the car. On clicking the microphone button, speech recognizer will be enabled and the word said by the user will replace the label “VOICE COMMAND”. The car is programmed to move whenever the application recognizes the words “FORWARD”, “BACKWARD”, “LEFT” and “RIGHT”. The car will stop moving if it hears the word “STOP”.

4.2. FACIAL RECOGNITION SOFTWARE

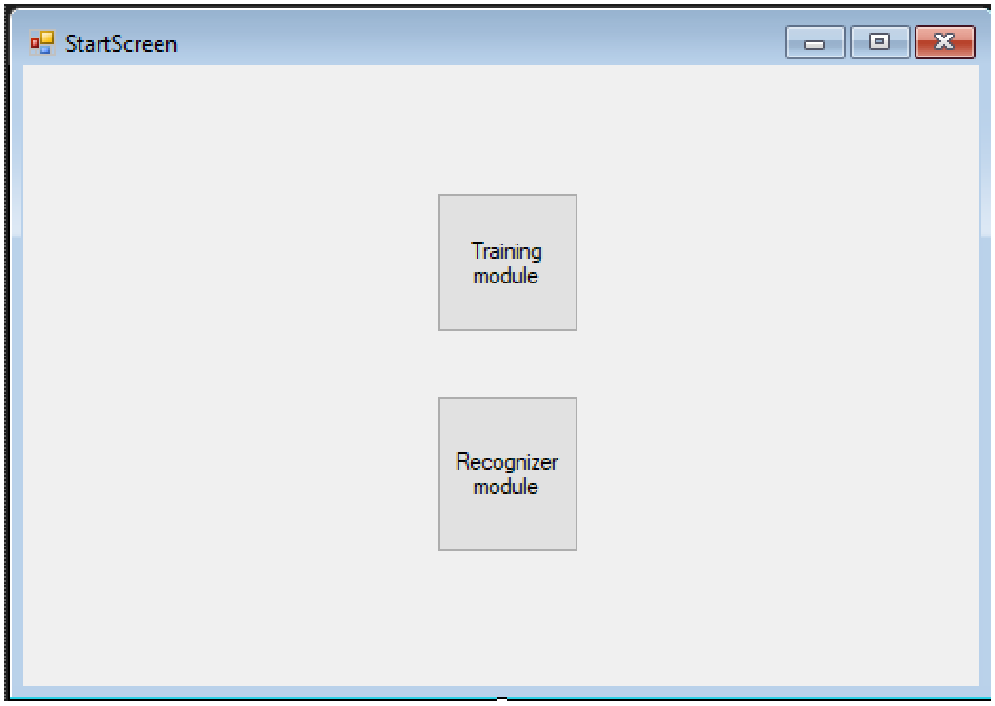


Fig 4.4: StartScreen Interface

Fig. 4.4 shows the Start Screen of the Facial Recognition Application. If we want to train a new face, we will click on “Training Module” which will open the interface shown in Fig. 4.5

If want to detect faces, we can click on “Recognizer Module” which will open the interface shown in Fig. 4.6

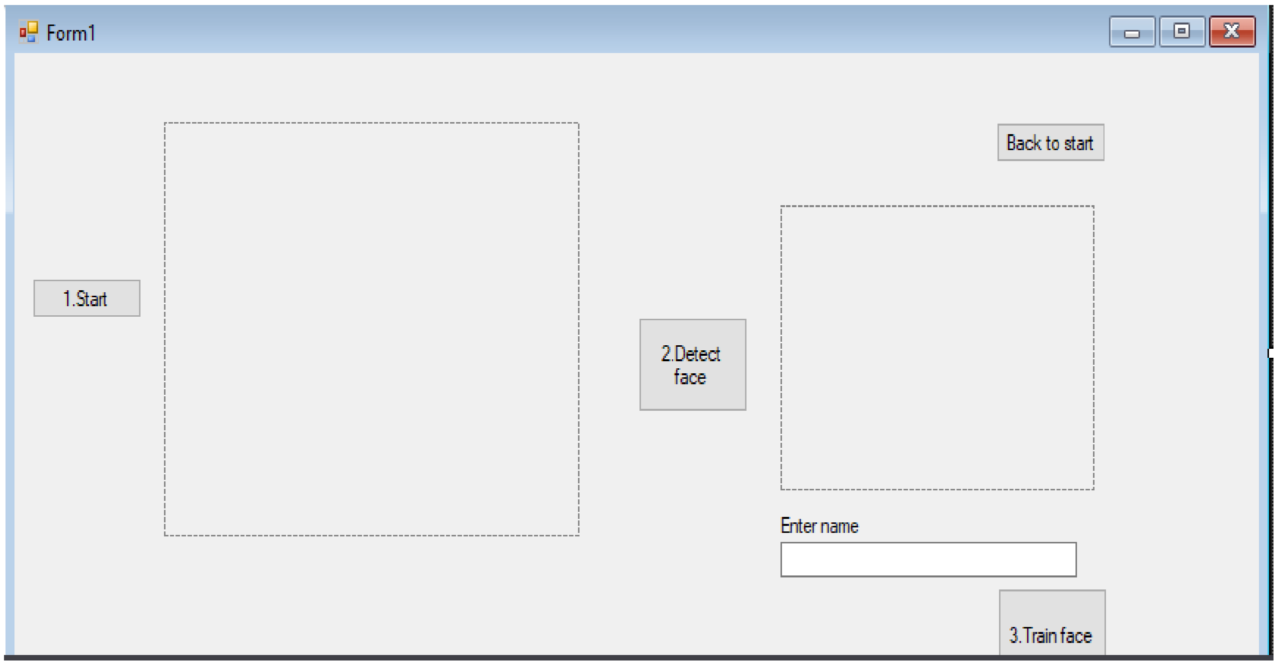


Fig 4.5: Training Interface

Fig. 4.5 shows the training interface. On clicking the "Start" button, the camera will turn on and will display the live feed in the box shown. Since we are using Haar Cascade File, the camera will automatically detect a face and will make a red box around it. When we click on "Detect Face", the last face detected by the camera is saved and displayed in the second box(smaller one). On clicking "Train Face", the name of the person along with his/her face is stored in the database.

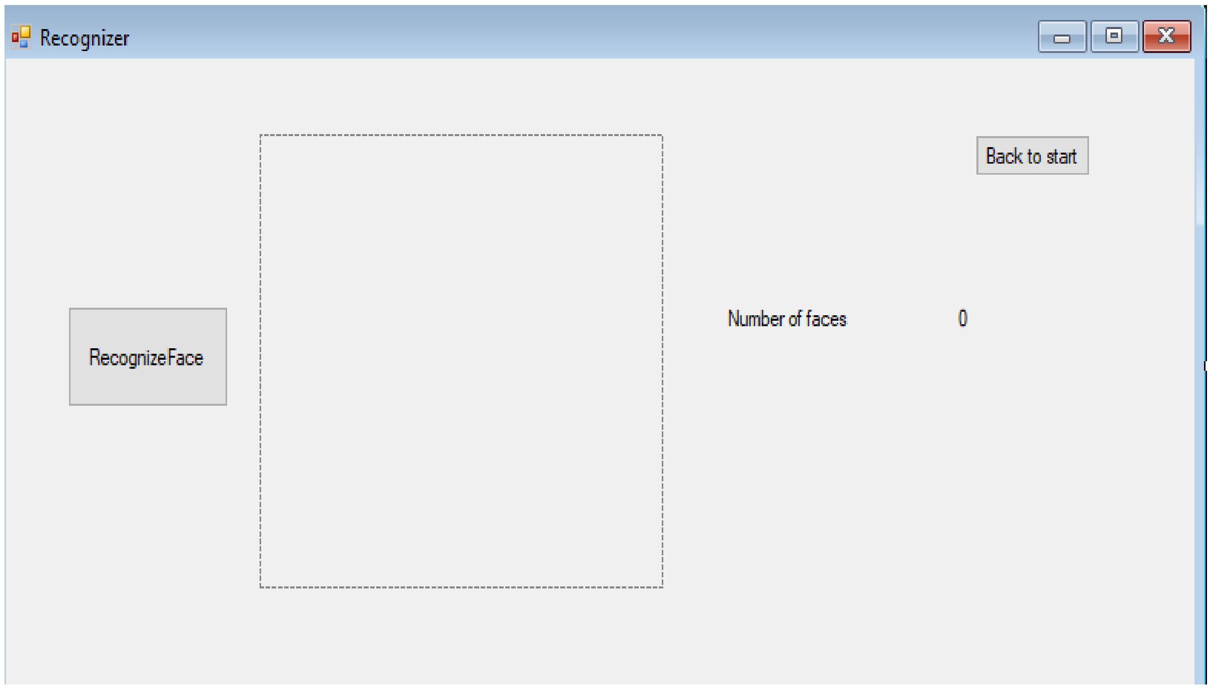


Fig 4.6: Detecting Interface

Fig. 4.6 shows the recognizer module interface. When we click on "Recognize Face", the camera will turn on and the live feed will be displayed. In the live feed, there will be a red box around people along with their names (assuming their faces are stored in the database).

Chapter 5

Implementation

5.1. CODE

5.1.1 SmartBot

```
void setup() {
 pinMode(3,OUTPUT);
 pinMode(4,OUTPUT);
 pinMode(5,OUTPUT);
 pinMode(6,OUTPUT);
 Serial.begin(9600);
}

void loop()
{
 if(Serial.available())
 {
 int z = Serial.read();

 if(z==1)
 {
 digitalWrite(3,HIGH);
 digitalWrite(4,LOW);
 digitalWrite(5,HIGH);
 digitalWrite(6,LOW);
 }
```

```
if(z==2)
{
digitalWrite(3,LOW);
digitalWrite(4,HIGH);
digitalWrite(5,LOW);
digitalWrite(6,HIGH);
}

 if(z==4)
{

digitalWrite(3,HIGH);
digitalWrite(4,HIGH);
digitalWrite(5,HIGH);
digitalWrite(6,LOW);
}

if(z==3)
{

digitalWrite(3,HIGH);
digitalWrite(4,LOW);
digitalWrite(5,HIGH);
digitalWrite(6,HIGH);
}

if(z==5)
{
digitalWrite(3,HIGH);
digitalWrite(4,HIGH);
digitalWrite(5,HIGH);
```

```
 digitalWrite(6,HIGH);
 }
 }

}
```

5.1.2 AcceleroBot

```
void setup() {
 pinMode(3,OUTPUT);
 pinMode(4,OUTPUT);
 pinMode(5,OUTPUT);
 pinMode(6,OUTPUT);
 Serial.begin(9600);
}

void loop()
{
 if(Serial.available())
 {
 int z = Serial.read();

 if(z==1)
 {
 digitalWrite(3,HIGH);
 digitalWrite(4,LOW);
 digitalWrite(5,HIGH);
 digitalWrite(6,LOW);
 }
```

```
if(z==2)
{
digitalWrite(3,LOW);
digitalWrite(4,HIGH);
digitalWrite(5,LOW);
digitalWrite(6,HIGH);
}

 if(z==4)
{

digitalWrite(3,HIGH);
digitalWrite(4,HIGH);
digitalWrite(5,HIGH);
digitalWrite(6,LOW);
}

if(z==3)
{

digitalWrite(3,HIGH);
digitalWrite(4,LOW);
digitalWrite(5,HIGH);
digitalWrite(6,HIGH);
}

if(z==5)
{
digitalWrite(3,HIGH);
digitalWrite(4,HIGH);
```

```
 digitalWrite(5,HIGH);
 digitalWrite(6,HIGH);
 }
 }

}
```

5.1.3 VoiceBot

```
void setup() {
 pinMode(3,OUTPUT);
 pinMode(4,OUTPUT);
 pinMode(5,OUTPUT);
 pinMode(6,OUTPUT);
 Serial.begin(9600);
}

void loop()
{
 if(Serial.available())
 {
 int z = Serial.read();

 if(z==1)
 {
 digitalWrite(3,HIGH);
 digitalWrite(4,LOW);
 digitalWrite(5,HIGH);
digitalWrite(6,LOW);
}

if(z==2)
{
digitalWrite(3,LOW);
digitalWrite(4,HIGH);
digitalWrite(5,LOW);
digitalWrite(6,HIGH);
}

 if(z==4)
{

digitalWrite(3,HIGH);
digitalWrite(4,HIGH);
digitalWrite(5,HIGH);
digitalWrite(6,LOW);
}

if(z==3)
{

digitalWrite(3,HIGH);
digitalWrite(4,LOW);
digitalWrite(5,HIGH);
digitalWrite(6,HIGH);
}

if(z==5)
{
 digitalWrite(3,HIGH);
 digitalWrite(4,HIGH);
 digitalWrite(5,HIGH);
 digitalWrite(6,HIGH);
 }
 }
}
```

5.1.4 StartScreen.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace FacialRecognition
{
    public partial class StartScreen : Form
    {
        public StartScreen()
        {
            InitializeComponent();
        }

        private void StartScreen_Load(object sender, EventArgs e)
        {



        }


        private void button1_Click(object sender, EventArgs e)
        {


            Form1 f = new Form1();
            f.Show();
            this.Hide();
        }


        private void button2_Click(object sender, EventArgs e)
        {
            Recognizer r = new Recognizer();
            r.Show();
            this.Hide();
        }
    }
}
```

5.1.5 Form1.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using Emgu.CV;
using Emgu.CV.Structure;
using Emgu.CV.CvEnum;
using System.IO;
namespace FacialRecognition
{
    public partial class Form1 : Form
    {
        Capture grabber;
        Image<Bgr, byte> currentFrame;
        Image<Gray, byte> gray,result,TrainedFace = null;

        HaarCascade face;

        List<Image<Gray, byte>> trainingImages = new List<Image<Gray,
byte>>();
        List<string> labels = new List<string>();
        int NumLabels,ContTrain=0;
        int t = 0;
        public Form1()
    {

        face = new HaarCascade("haarcascade_frontalface_default.xml");

        InitializeComponent();
        try
        {

            string Labelsinfo = File.ReadAllText(Application.StartupPath +
"/TrainedFaces/TrainedLabels.txt");
            string[] Labels = Labelsinfo.Split('%');
            NumLabels = Convert.ToInt16(Labels[0]);
            ContTrain = NumLabels;

            string LoadFaces;

            for (int tf = 1; tf < NumLabels + 1; tf++)
            {
                LoadFaces = "face" + tf + ".bmp";
                trainingImages.Add(new Image<Gray,
byte>(Application.StartupPath + "/TrainedFaces/" + LoadFaces));
                labels.Add(Labels[tf]);
            }

        }
        catch (Exception e)
        {
            MessageBox.Show("Train item if its first time");
        }
    }

    private void Form1_Load(object sender, EventArgs e)
    {

    }

    private void button1_Click(object sender, EventArgs e)
    {
        grabber = new Capture();

        grabber.QueryFrame();

        Application.Idle += new EventHandler(FrameGrabber);

        button2.Visible = true;

    }
    void FrameGrabber(object sender, EventArgs e)
    {

        currentFrame = grabber.QueryFrame().Resize(320, 240,
Emgu.CV.CvEnum.INTER.CV_INTER_CUBIC);

        gray = currentFrame.Convert<Gray, Byte>();

        MCvAvgComp[][] facesDetected = gray.DetectHaarCascade(
face,
1.2,
10,
Emgu.CV.CvEnum.HAAR_DETECTION_TYPE.DO_CANNY_PRUNING,
new Size(20, 20));

        foreach (MCvAvgComp f in facesDetected[0])
        {

            t = t + 1;

            result = currentFrame.Copy(f.rect).Convert<Gray, byte>().Resize(100,
100, Emgu.CV.CvEnum.INTER.CV_INTER_CUBIC);



            currentFrame.Draw(f.rect, new Bgr(Color.Red), 2);



        }

        imageBox1.Image = currentFrame;
     }

    private void imageBox1_Click(object sender, EventArgs e)
    {

    }

    private void button2_Click(object sender, EventArgs e)
    {
        label1.Visible = true;
        textBox1.Visible = true;
        button3.Visible = true;
        TrainedFace = result.Resize(100, 100,
Emgu.CV.CvEnum.INTER.CV_INTER_CUBIC);

imageBox2.Image = TrainedFace;



    }




  private void button3_Click(object sender, EventArgs e)
    {
        ContTrain=ContTrain+1;

        trainingImages.Add(TrainedFace);
        labels.Add(textBox1.Text);

        File.WriteAllText(Application.StartupPath +
"/TrainedFaces/TrainedLabels.txt", trainingImages.ToArray().Length.ToString()
+ "%");//add library to read/write to input file

        for (int i = 1; i < trainingImages.ToArray().Length + 1; i++)
        {
            trainingImages.ToArray()[i - 1].Save(Application.StartupPath +
"/TrainedFaces/face" + i + ".bmp");//sav faces to folder with name face(i)i is no.
of face and .bmp extension of detected face image
            File.AppendAllText(Application.StartupPath +
"/TrainedFaces/TrainedLabels.txt", labels.ToArray()[i - 1] + "%");//save names
to text file
        }
        MessageBox.Show("Image trained and save to database");
    }

    private void button4_Click(object sender, EventArgs e)
    {
        StartScreen s = new StartScreen();
        s.Show();
        this.Hide();
    }



}
}
```

5.1.6 Recognizer.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using Emgu.CV;
using Emgu.CV.Structure;
using Emgu.CV.CvEnum;
using System.IO;
namespace FacialRecognition
```

```
{

   public partial class Recognizer : Form
   {
       Image<Bgr, Byte> currentFrame;
       Capture grabber;
       HaarCascade face;
       Image<Gray, byte> result, TrainedFace = null;
       Image<Gray, byte> gray = null;
       List<Image<Gray, byte>> trainingImages = new List<Image<Gray,
byte>>();
       List<string> labels = new List<string>();
       MCvFont font = new MCvFont(FONT.CV_FONT_HERSHEY_TRIPLEX,
0.5d, 0.5d);

       List<string> NamePersons = new List<string>();
       string name = null;
       int t, ContTrain, NumLabels;
       public Recognizer()
       {
           InitializeComponent();
             face = new HaarCascade("haarcascade_frontalface_default.xml");
             try
             {

                 string Labelsinfo = File.ReadAllText(Application.StartupPath +
"/TrainedFaces/TrainedLabels.txt");
                 string[] Labels = Labelsinfo.Split('%');
                 NumLabels = Convert.ToInt16(Labels[0]);
                 ContTrain = NumLabels;
                 string LoadFaces;
```

```
                for (int tf = 1; tf < NumLabels + 1; tf++)
                {
                    LoadFaces = "face" + tf + ".bmp";
                    trainingImages.Add(new Image<Gray,
byte>(Application.StartupPath + "/TrainedFaces/" + LoadFaces));
                    labels.Add(Labels[tf]);
                }

            }
            catch (Exception e)
            {
                MessageBox.Show("no image trained");
            }
        }

        private void Recognizer_Load(object sender, EventArgs e)
        {

        }
        private void button1_Click(object sender, EventArgs e)
        {

            grabber = new Capture();
            grabber.QueryFrame();

            Application.Idle += new EventHandler(FrameGrabber);

        }
        void FrameGrabber(object sender, EventArgs e)
        {
```

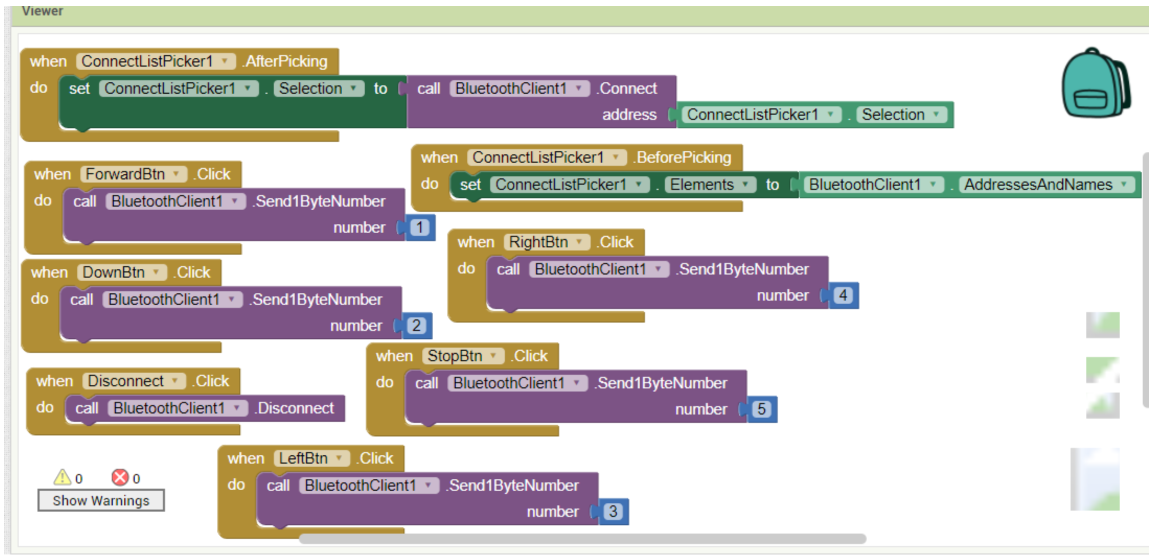
```
        NamePersons.Add("");

        label2.Text = "0";


        currentFrame = grabber.QueryFrame().Resize(320, 240,
Emgu.CV.CvEnum.INTER.CV_INTER_CUBIC);


        gray = currentFrame.Convert<Gray, Byte>();


        MCvAvgComp[][] facesDetected = gray.DetectHaarCascade(
       face,
       1.3,
       10,
       Emgu.CV.CvEnum.HAAR_DETECTION_TYPE.DO_CANNY_PRUNING,
       new Size(20, 20));


        foreach (MCvAvgComp f in facesDetected[0])
        {
           t = t + 1;
           result = currentFrame.Copy(f.rect).Convert<Gray, byte>().Resize(100,
100, Emgu.CV.CvEnum.INTER.CV_INTER_CUBIC);
 currentFrame.Draw(f.rect, new Bgr(Color.Red), 2);
           //initialize result,t and gray if (trainingImages.ToArray().Length != 0)
           {
```

```
                    MCvTermCriteria termCrit = new MCvTermCriteria(ContTrain,
0.001);

                    EigenObjectRecognizer recognizer = new EigenObjectRecognizer(
                        trainingImages.ToArray(),
                        labels.ToArray(),
                        1000,
                        ref termCrit);

                    name = recognizer.Recognize(result);

                    currentFrame.Draw(name, ref font, new Point(f.rect.X - 2, f.rect.Y -
2), new Bgr(Color.LightGreen));

                }
                NamePersons[t - 1] = name;
                NamePersons.Add("");
                label2.Text = facesDetected[0].Length.ToString();
            }
            imageBox1.Image = currentFrame;


        }

        private void button4_Click(object sender, EventArgs e)
        {
            StartScreen s = new StartScreen();
            s.Show();
            this.Hide();
        }
```

```
    }
}
```

5.2. BLOCK DIAGRAM

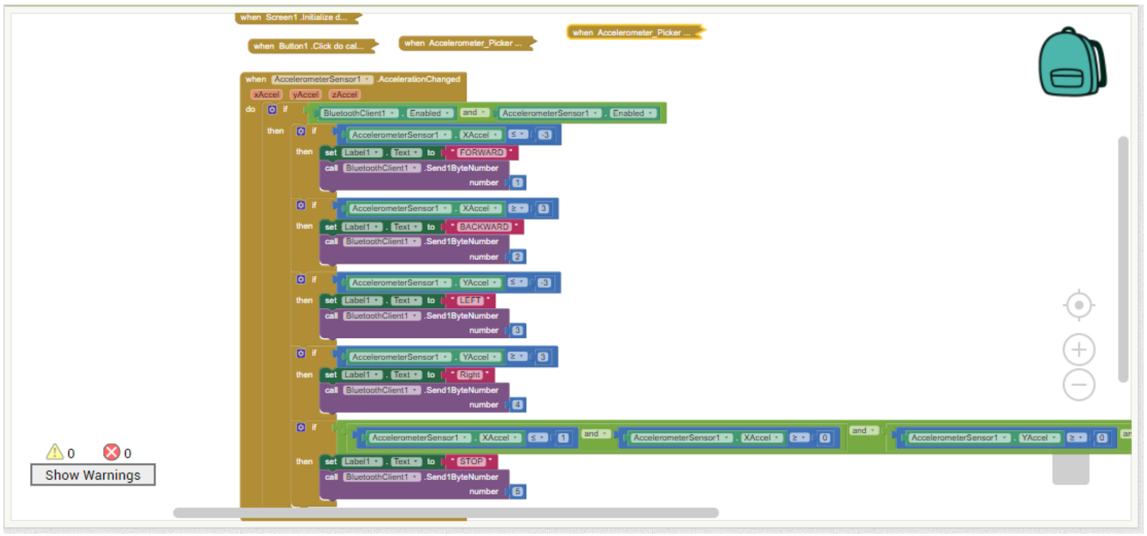
5.2.1 SMARTBOT



```
Viewer

when ConnectListPicker1 .AfterPicking
do  set ConnectListPicker1 . Selection to  call BluetoothClient1 .Connect
                                                    address  ConnectListPicker1 . Selection

when ConnectListPicker1 .BeforePicking
do  set ConnectListPicker1 . Elements to  BluetoothClient1 . AddressesAndNames

when ForwardBtn .Click
do  call BluetoothClient1 .Send1ByteNumber
                             number  1

when RightBtn .Click
do  call BluetoothClient1 .Send1ByteNumber
                             number  4

when DownBtn .Click
do  call BluetoothClient1 .Send1ByteNumber
                             number  2

when StopBtn .Click
do  call BluetoothClient1 .Send1ByteNumber
                             number  5

when Disconnect .Click
do  call BluetoothClient1 .Disconnect

when LeftBtn .Click
do  call BluetoothClient1 .Send1ByteNumber
                             number  3
```

Fig 5.1: SmartBot Block Diagram

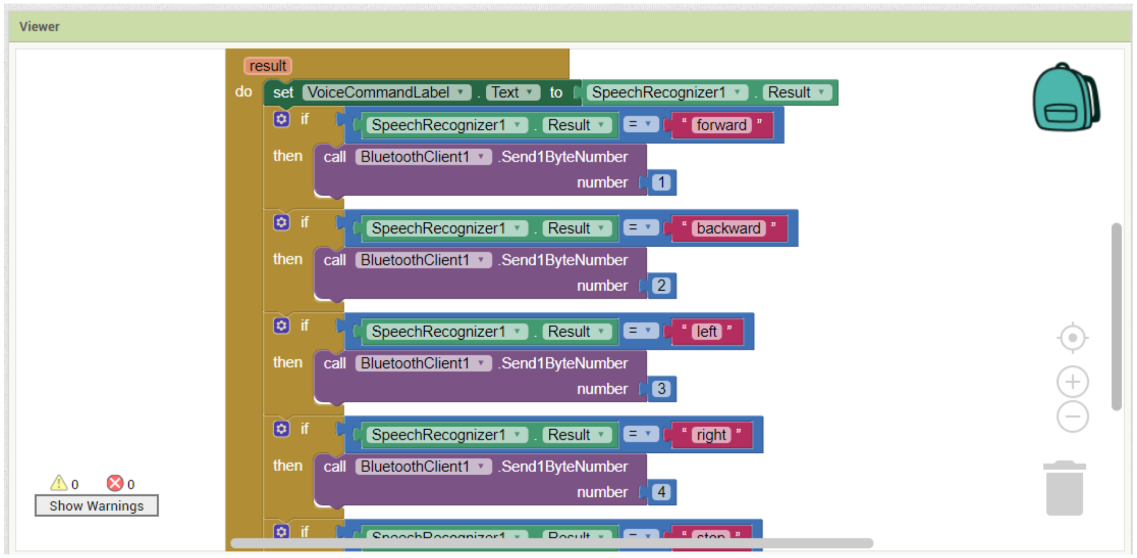
5.2.2 ACCELEROBOT



```
when Screen1 .Initialize d...
when Button1 .Click do cal...
when Accelerometer_Picker ...
when Accelerometer_Picker ...

when AccelerometerSensor1 .AccelerationChanged
  xAccel  yAccel  zAccel
do  if  BluetoothClient1 . Enabled  and  AccelerometerSensor1 . Enabled
    then  if  AccelerometerSensor1 . XAccel  ≤  -3
          then  set Label1 . Text to " FORWARD "
                call BluetoothClient1 .Send1ByteNumber
                                          number 1
          if  AccelerometerSensor1 . XAccel  ≥  3
          then  set Label1 . Text to " BACKWARD "
                call BluetoothClient1 .Send1ByteNumber
                                          number 2
          if  AccelerometerSensor1 . YAccel  ≤  -3
          then  set Label1 . Text to " LEFT "
                call BluetoothClient1 .Send1ByteNumber
                                          number 3
          if  AccelerometerSensor1 . YAccel  ≥  3
          then  set Label1 . Text to " Right "
                call BluetoothClient1 .Send1ByteNumber
                                          number 4
          if  AccelerometerSensor1 . XAccel  ≤  1  and  AccelerometerSensor1 . XAccel  ≥  0  and  AccelerometerSensor1 . YAccel  ≥  0  an
          then  set Label1 . Text to " STOP "
                call BluetoothClient1 .Send1ByteNumber
                                          number 5
```

Fig 5.2: AcceleroBot Block Diagram

5.2.3 VOICEBOT



Viewer

```
  result
do  set VoiceCommandLabel . Text to  SpeechRecognizer1 . Result
    if  SpeechRecognizer1 . Result  =  " forward "
    then  call BluetoothClient1 .Send1ByteNumber
                                    number 1
    if  SpeechRecognizer1 . Result  =  " backward "
    then  call BluetoothClient1 .Send1ByteNumber
                                    number 2
    if  SpeechRecognizer1 . Result  =  " left "
    then  call BluetoothClient1 .Send1ByteNumber
                                    number 3
    if  SpeechRecognizer1 . Result  =  " right "
    then  call BluetoothClient1 .Send1ByteNumber
                                    number 4
    if  SpeechRecognizer1 . Result  =  " stop "
```

Fig 5.3: VoiceBot Block Diagram

Chapter 6

Result

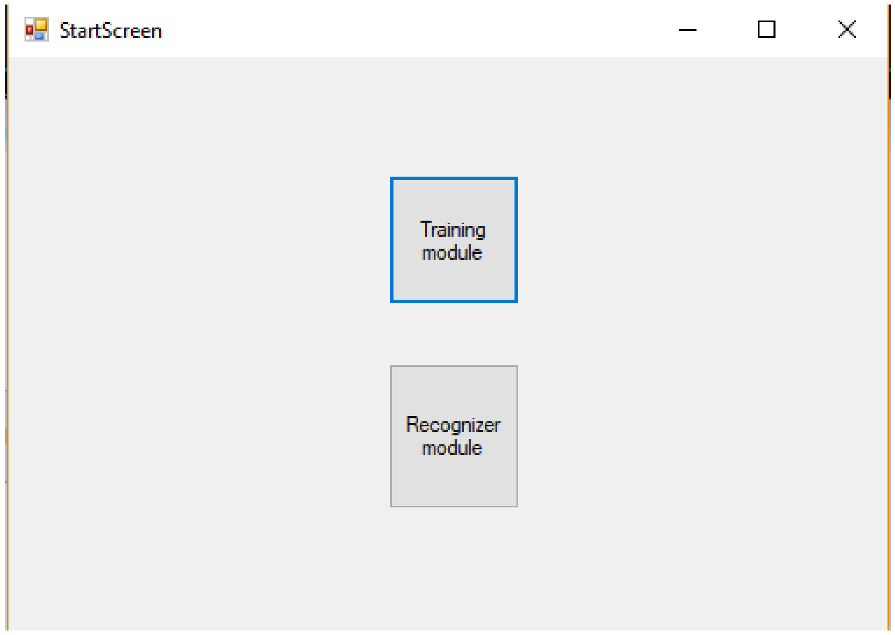


Fig 6.1: Facial Recognition Interface

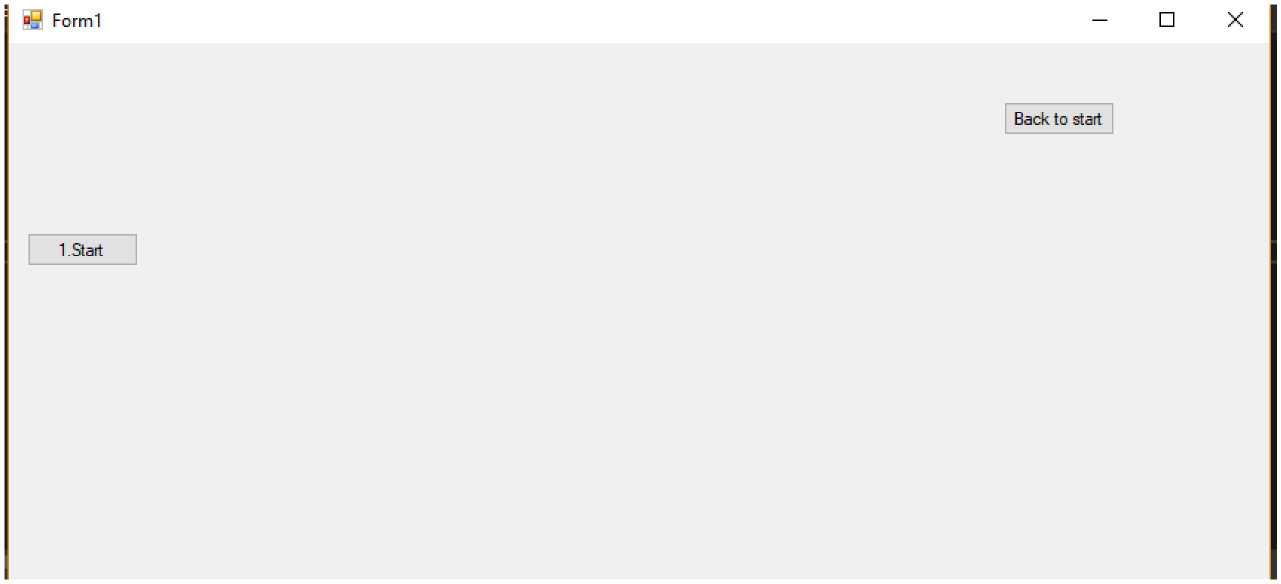


Fig 6.2: Facial Recognition Interface (2)

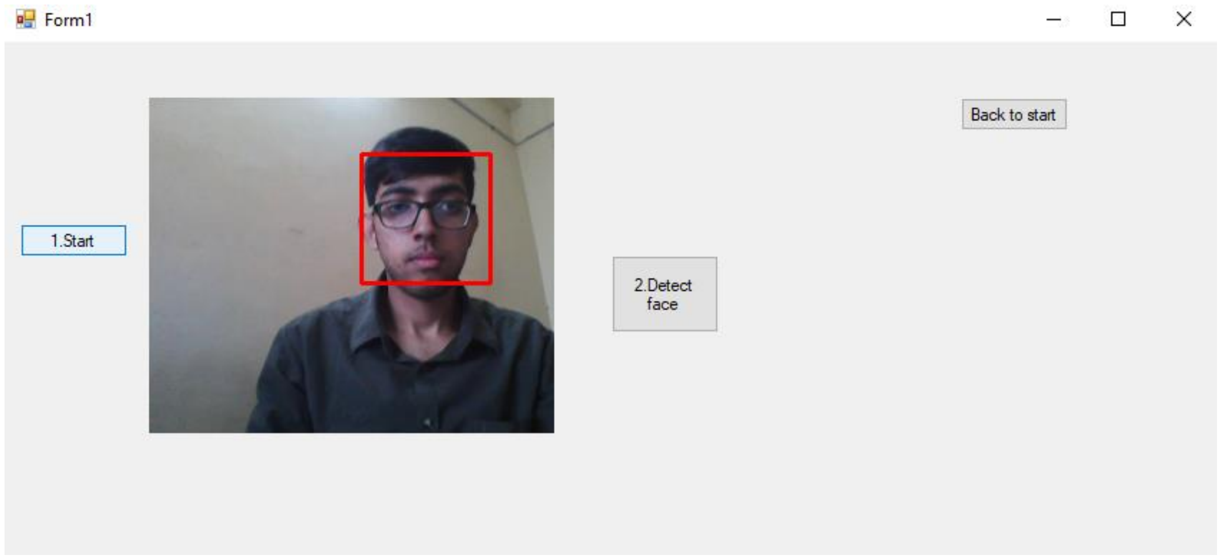


Fig 6.3: Facial Recognition Interface (3)

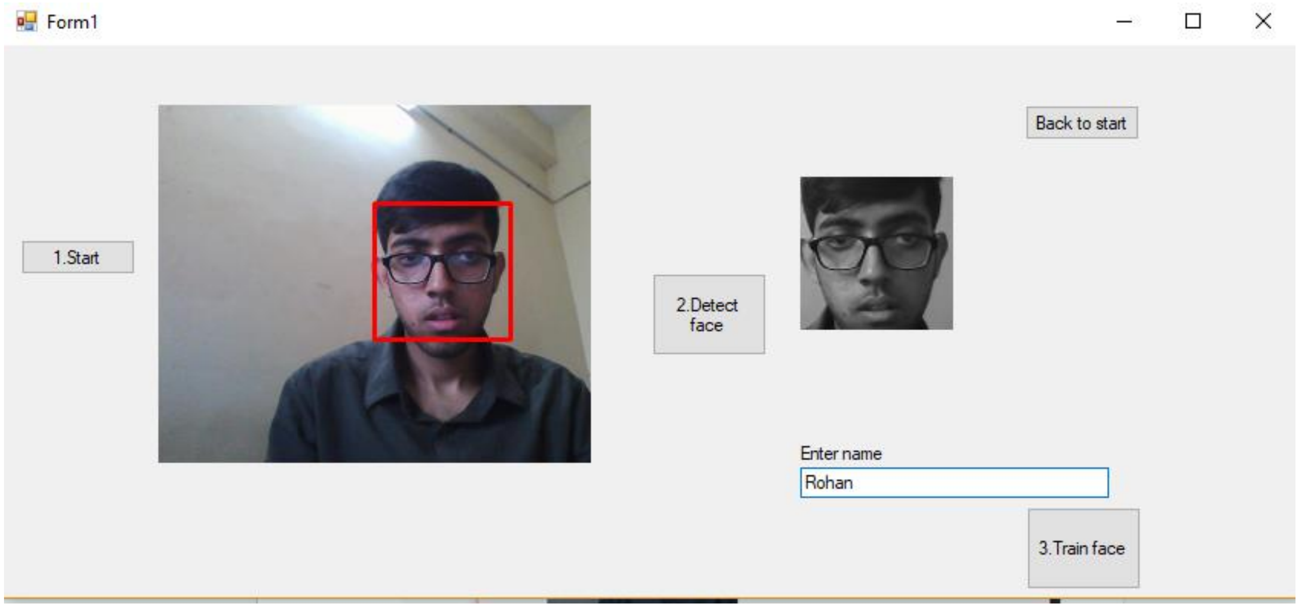


Fig 6.4: Facial Recognition Interface (4)

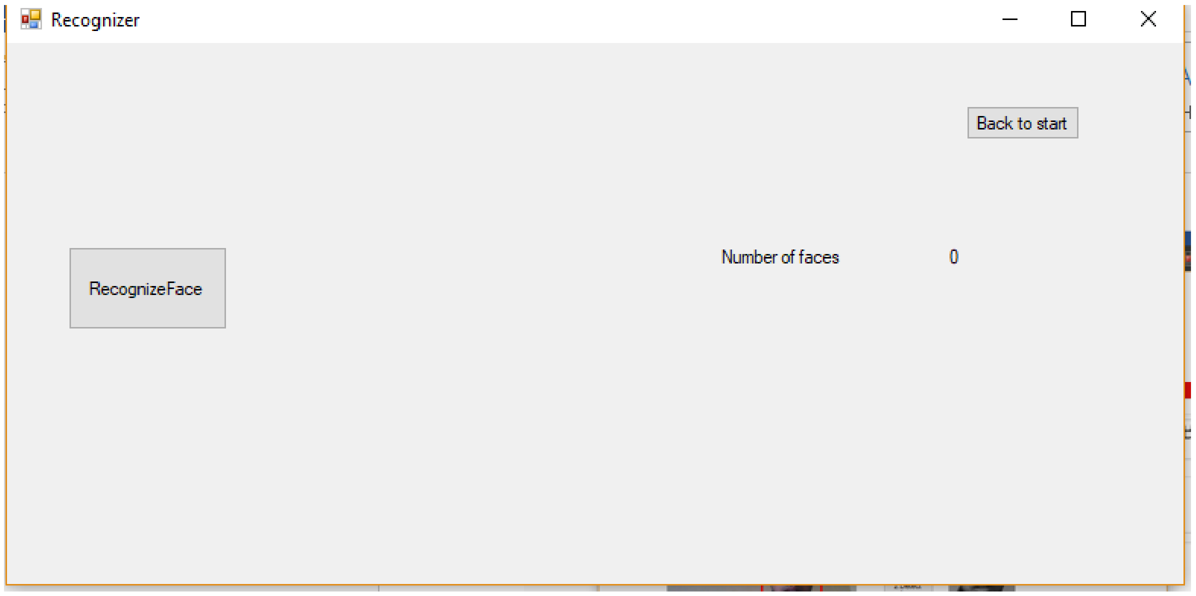


Fig 6.5: Facial Recognition Interface (5)

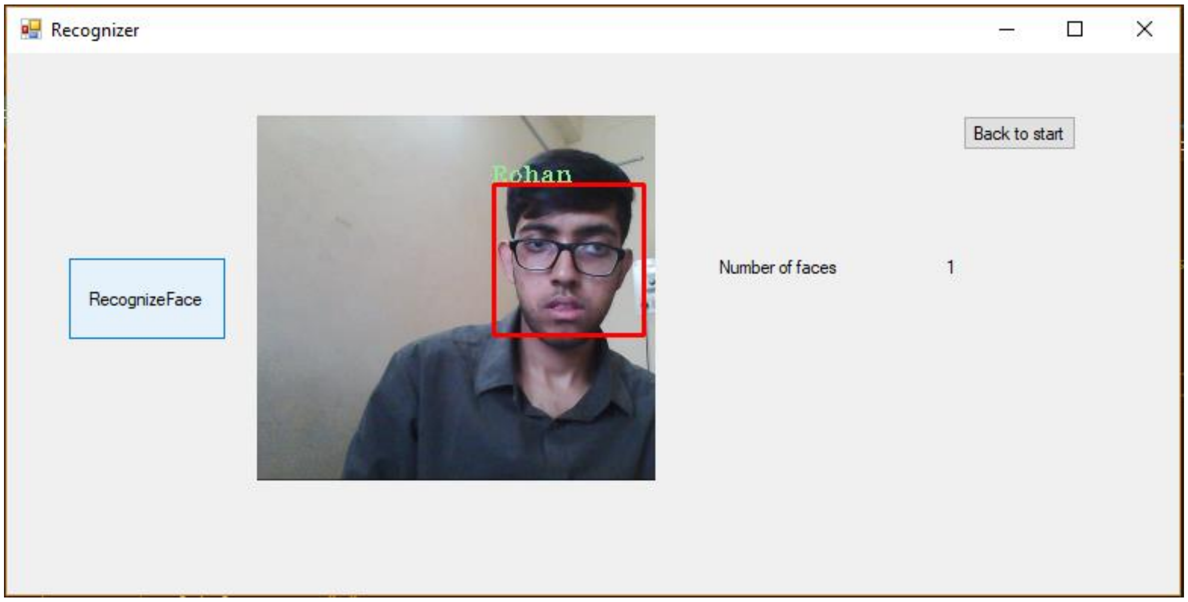


Fig 6.6: Sample situation in this technology (6)

Chapter 7

Conclusion & Future Work

7.1 CONCLUSION

This project is just a prototype for a Military Reckon vehicle. There are many situations when this vehicle will be used. One common example is “The Hostage Situation”. The FBI, Interpol etc. have facial records of known terrorists and criminals in their database. They can use these to train the model. The robot’s small size can be advantageous when entering hostile areas through small entry points. The camera mounted on the robot is used to detect faces of the criminals in the hostage situation. Not only that, but the camera can be used to detect types of guns the criminals are carrying, number of people and to scan the area. This information can help the police, military etc. strategize their next step. Innovations such as fingerprint sensors and iris scanner are now being widely used. However, experts predict that facial recognition will soon rule the biometric security market, offering another level of data security.

7.2. FUTURE WORK

There is an extraordinary amount of features we can add to this. Since this is a prototype, only the basic components have been used. Arduino can be replaced by much more efficient microcontroller board such as Raspberry Pi. The Bluetooth module can communicate with the Smartphone only for short distances. For practical purposes a stronger connection will be required which can work for long distances. Bluetooth Module can be replaced by Antennas and Amplifiers.
Army is developing drones that can recognize a human’s face from a distance. Progeny's system, if it works the way the company and the Army envision it, needs

just 50 pixels between the target's eyes in a 2-D image to build the 3-D model. From that model stored in Progeny's database, the system could identify the target from an even lower resolution image or video. The closer the drone is to the subject, the better all of this works. But progeny also layers in a second kind of recognition that can work at more than 750 feet. This "soft biometric" system basically takes in a bunch of non-facial but otherwise outwardly relevant data--skin color, height and build, age, gender--to build a larger kind of model for its vision algorithms to work with. If a body is moving through the crowd, Progeny claims that a drone circling high overhead can keep track of him or her simply using this larger, whole-body identification system.

REFERENCES

- Animetrics, 2008-06-04,*"Face Recognition Applications"*.
- Consumer Reports, 2016-04-05, *"Facial Recognition: Who's Tracking You in Public?"*
- *"Airport Facial Recognition Passenger Flow Management". hrsid.com.*
- Bonsor, K., 2008-06-02, *"How Facial Recognition Systems Work"*
- Smith, Kelly, 2008-06-04, *"Face Recognition"*
- R. Brunelli and T. Poggio, IEEE Trans. on PAMI, 1993, *"Face Recognition: Features versus Templates"*
- R. Brunelli, 2009, *Template Matching Techniques in Computer Vision: Theory and Practice*
- Hardesty, Larry, August 19, 2010. *"The MIT roots of Google's new software"*. .
- Google, August 10, 2010, *"On the Shoulders of Giants!"*
- Wolber, David; Abelson, Hal; Spertus, Ellen; Looney, Liz, May 2011, *"App Inventor for Android: Create Your Own Android Apps*
- Clay Dillow, 2011, "Army Developing Drones That Can Recognize Your Face From a Distance".
- *Pulli, Kari; Baksheev, Anatoly; Kornyakov, Kirill; Eruhimov, Victor (1 April 2012). "Realtime Computer Vision with OpenCV". Queue. pp. 40:40–40:56. doi:10.1145/2181796.2206309.*
- Intel acquires Itseez: https://opencv.org/intel-acquires-itseez.html
- https://github.com/opencv/opencv/wiki/Deep-Learning-in-OpenCV
- *Adrian Kaehler; Gary Bradski (14 December 2016). Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library. O'Reilly Media. pp. 26ff. ISBN 978-1-4919-3800-3.*
- *Bradski, Gary; Kaehler, Adrian (2008). Learning OpenCV: Computer vision with the OpenCV library. O'Reilly Media, Inc. p. 6.*