Embedded Systems

Facial Recognition Ignition System

Fa.R.I.S.

Group 34

Daniel Leach

dleach2018@fau.edu

Jonathan Northrup

jnorthrup2018@fau.edu

Dr. Bassem Alhalabi

FAU COLLEGE OF ENGINEERING & COMPUTER SCIENCE Florida Atlantic University

Summer 2020

Project Summary:

Currently, a key is the sole way to start a vehicle. With facial recognition, we can alleviate the key all together by using a person's unique facial features, and at the same time prevent unrecognized users from operating the vehicle. The purpose of this project is to create a device that can not only start a vehicle using facial recognition but to also serve as a launching pad for other safety features in the future.

Contents

1 Introduction

1.1 Problem Description

We believe car keys are outdated and that vehicles (especially EVs) should be brought into the future with newer technology. Already, vehicles can be started by push-button as long as the vehicle's key fob is inside of the vehicle. To us, this seemed to be the next logical step and would be a stepping stone to other features using facial recognition and even eye tracking like detecting if the driver's eyes are off the road. This type of technology could also pave the way for security in self driving vehicles. Should there be a fleet that can be delivered to a person's doorstep much like current taxis, the owner of the car would be able to input a photo of the vehicle renter and the automobile would not be able to be taken until the renter is actually in the vehicle.

1.2 Significance of the Problem

The significance of this problem is that when new technology is introduced it is our duty to explore this new technology. Without exploration we would not know the extent these new technologies can be implemented. That being said, with the Fa.R.I.S. we explore how to use facial recognition in a vehicular environment. To us, this is just the first step toward the future of face technology in vehicles. While it may at first seem like a luxury idea, things like power steering and air conditioning in vehicles were once considered luxury and now they are standard in almost every vehicle.

Other problems that are solved by using the Fa.R.I.S. are user experience. Overall user experience is what most consumers look for in a product. Fa.R.I.S gives the user the ability to not have to worry about losing their keys or key fob. This system is better than the existing system where keys and key fobs can be a hassle to carry and where locksmiths, tow trucks,and dealerships take advantage of the user for loss keys.

1.3 Overview of the Solution

The solution to keyless vehicle ignition that we are proposing uses a Raspberry Pi, camera, and facial recognition software, along with existing switches in the vehicle to create an automatic vehicle ignition system.

1.4 Goals and Objectives

1. Design and implement an automatic vehicle ignition system based on facial recognition.
2. Design and develop facial recognition software using the opencv library.
3. Design and develop circuit for lights, switches, and lcd to be interfaced with raspberry pi.

2 System Design

2.1 Project Requirements

1. System should be ready within 15 seconds from initial activation of driver door switch
2. System shall be usable in all lighting conditions
3. System shall start by user input after face is recognized
4. System shall not start vehicle until driver door switch
5. Software shall not allow unauthorized users from operating vehicle.
6. System shall start vehicle when all requirements are met.

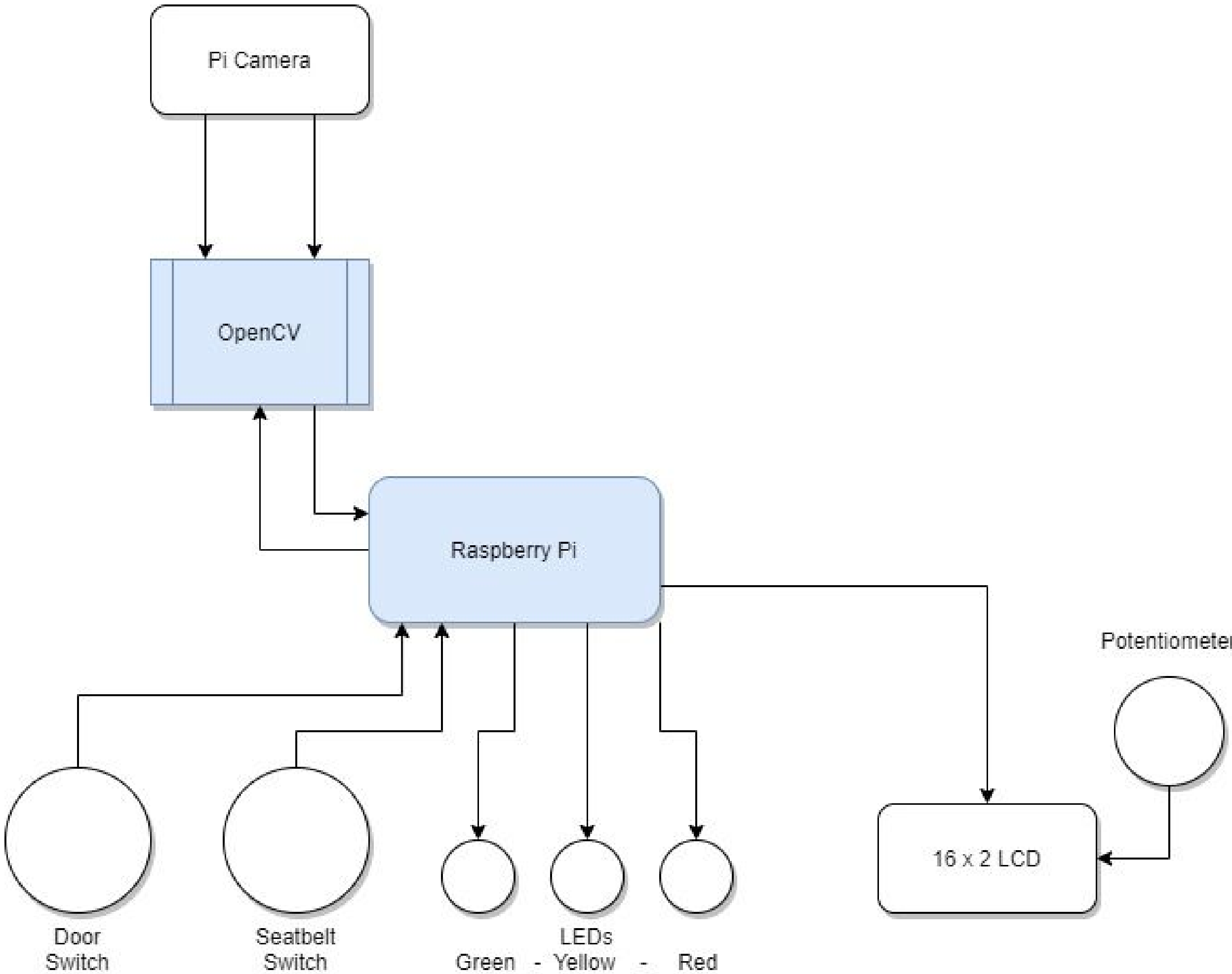
2.2 Project Requirements Not Met

1. System shall have setting for new user input and System shall start automatically with no user input, these requirements have not been met because we needed to manually operate the system for testing
2. System shall be usable in all lighting conditions - the raspberry pi camera doesn’t perform well in low light conditions

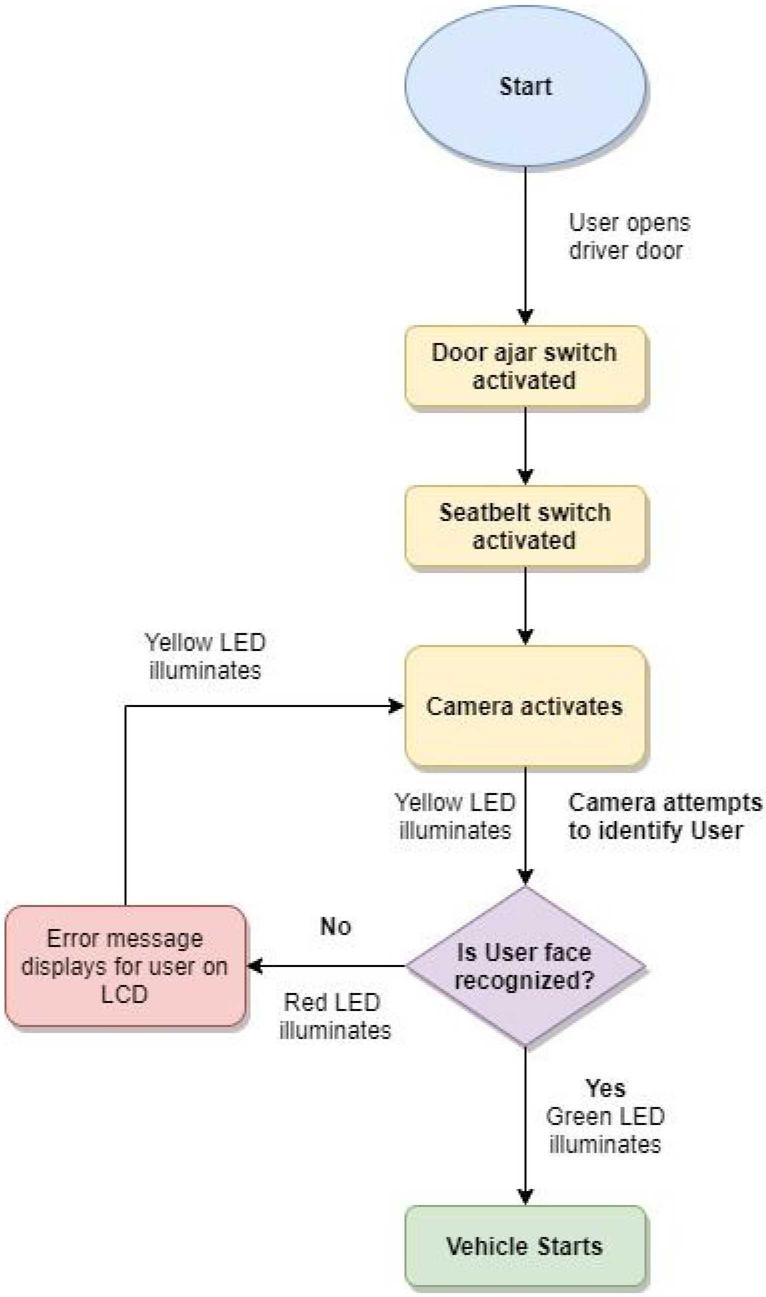
2.3 Project Requirements Added

1. System shall not start vehicle until driver door switch and seatbelt switches are activated - this requirement was added as a safety precaution so the vehicle does not start until door is closed and seatbelt is fastened

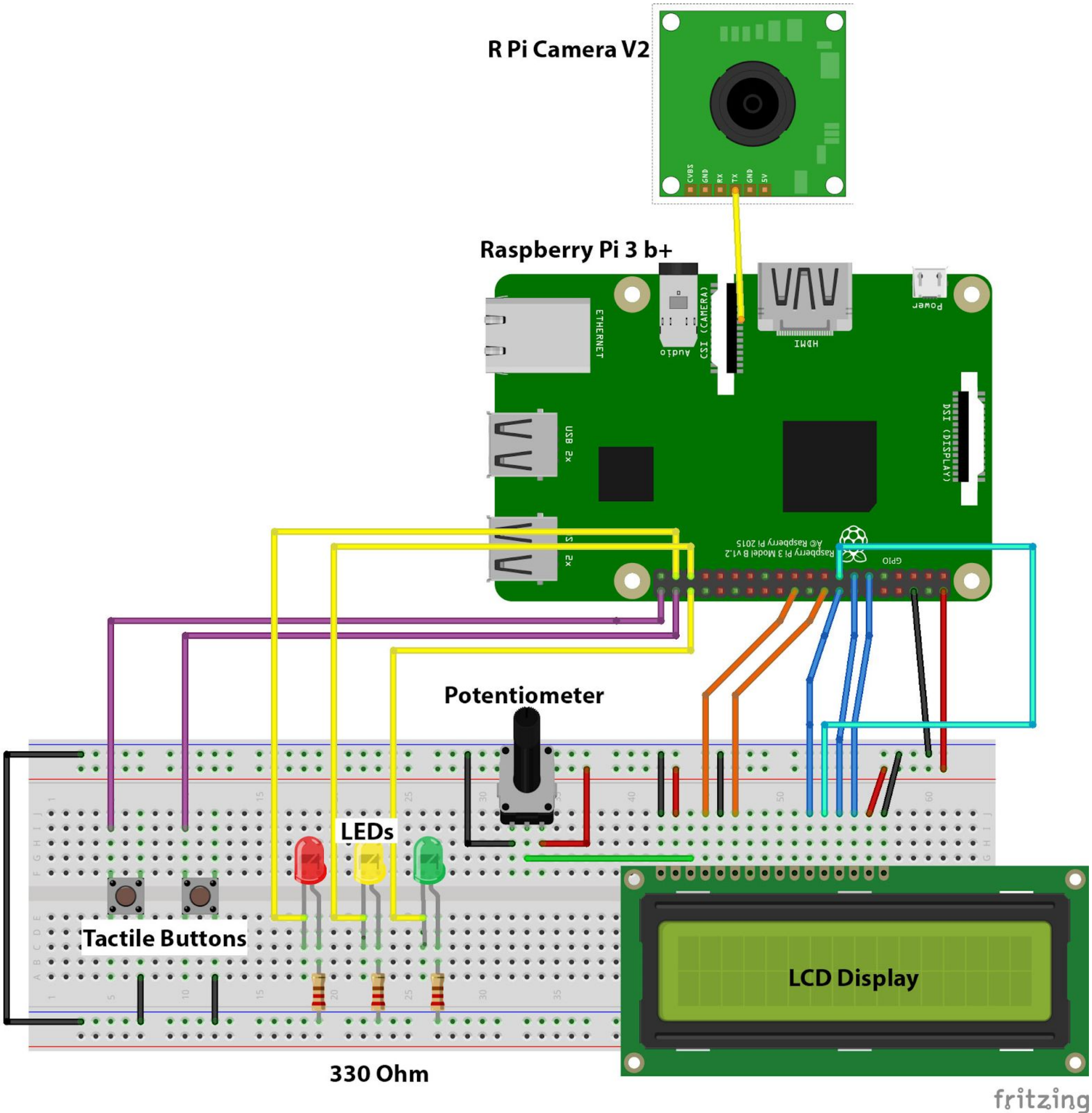
2.4 System Diagram



2.5 State Diagram



2.6 System Schematic



3 System Implementation

3.1 Hardware

The hardware involved in developing the Facial Recognition Ignition System include: Raspberry Pi 3b+, Raspberry Pi V2 Camera Module, 16 x 2 LCD Module, two tactile button switches, Three colored light-emitting diodes (LED), one potentiometer, and three 330 Ohm resistors.

Raspberry Pi 3b+

Description:

The Raspberry Pi 3b+ (RPi) is a single board microcomputer that was released on March 14th 2018. It is the predecessor of the Raspberry Pi 3 b which was released two years earlier. The small single board computer has a 1.4 GHz 64-bit ARMv8 quad core processor, Ethernet, Wi-Fi, Bluetooth connectivity, and four 2.0 USB ports. The Raspberry Pi can use most ARM GNU/ linux distributions and Windows IOT operating systems. The native programming languages supported on the Raspberry PI is C++ and Python. Other programming languages can be used on the system but require additional support.

Testing and calibration (Daniel Leach):

When it comes to testing and calibrating the Raspberry Pi microcomputer Daniel leach was responsible for making sure that this was the right embedded component we could use for this project. The latest Raspbian OS distribution was installed for this project. When it comes to calibrating the Raspberry Pi certain things needed to be enabled such as Wi-Fi for getting the latest software releases, SSH for headless communications, increase the file system size for allowing large amounts of data to be stored, increase CPU speeds to allow for faster processing, and enable Camera to allow the use of the Raspberry v2 Camera module. For the testing of the Raspberry Pi the use of the GPIO pins are essential for the entire project. To ensure that each GPIO pin is working properly a python test script was written to test each input/output pin.The initial test was done using a Voltmeter to ensure the proper output voltage on each pin.

Raspberry Pi V2 Camera Module

Description:

The Raspberry Pi V2 Camera Module was released in 2016. It has a Sony IMX219 8-megapixel sensor. The V2 camera module supports 1080p30, 720p60 and VGA90 video modes and still photographs. The V2 camera is attached via a 15cm ribbon cable to the Raspberry Pi CSI port. Because this module was developed by and for the Raspberry Pi by the Raspberry Pi Foundation this module is an ideal camera for our system.

Testing and Calibration (Jonathan Northrup and Danie Leach):

Testing and Calibrating the V2 camera module for the Facial Recognition Ignition System was performed by Jonathan Northrup and Daniel Leach. Since the V2 was made to be used with the RPi Calibrating the camera module was easy. This was essentially plug and play. The testing of the camera was to ensure that the module worked with our system. We wrote a test script that would take a picture and save it in a designated directory. This was done to see if the camera was taking high quality photos that we can store in a dataset. The next test was video streaming so we can ensure that the camera would work for our system. We use live video streams in the process of facial recognition. This job was performed by two people to ensure that both people were getting the same results. This was only done because of social distancing measures that are in place during the 2020 Coronavirus Pandemic.

Integration (Jonathan Northrup and Daniel Leach)

Integrating the V2 camera module was necessary because it is one of the main components that is used in our system.

16 x 2 LCD Module

Description:

The 16 x 2 LCD (Liquid Crystal Display) Module is a basic display used in many systems and by many engineers to help display information for their system that is being developed. The 16 x 2 LCD can display 16 characters on 2 rows. The module allows for 8-bit and 4-bit data communication. Using 8-bit mode is faster because it allows to write all data in one transfer while in 4-bit mode the data needs to be split and sent in 2 write operations.In our project the LCD is in 4-bit mode. LCD modules can come in many different versions. Some versions include

I2C integration which is another communication method for microcontrollers and embedded systems.

Testing and Calibration (Daniel Leach):
The Testing and Calibration of the 16x2 LCD Module was performed by both team members. The module was first tested on an Arduino microcontroller because the Liquid Crystal Display library is natively supported in the Arduino IDE. Testing this module on a separate embedded system allowed for easy testing. To test the module on the Raspberry Pi additional libraries needed to be installed. In the initial testing phase of this module one of the team members accidentally fried the 16x2 LCD module.This was a minor setback because the team member had a backup that was used to continue the project.

Integration (Daniel Leach):
Integrating the LCD into the system was essential to allow outside communication to the user. Ideally in an automobile this type of communication would be displayed on the dashboard display that are common in many newer models.

Tactile Button Switches

Description:
This project uses two standard 12mm tactile button switches. These are used for user input in our Facial Recognition Ignition System. They are respectively used for system start and ignition start. The reason we have 2 buttons instead of one is to indicate a person opening and closing a door in an automobile. These buttons are strictly for simulation purposes. In a real world situation the system start would be triggered by the door indicator and ignition start would be tied to the automobiles ignition system.

Testing and Calibration (Jonathan Northrup):
To test the tactile buttons a small circuit was built on a solderless breadboard that included the buttons, LEDs, and the resistors. Each button was connected to a colored led and a 330 Ohm resistor. The circuit was powered by 3.3v. When the buttons are pressed the LEDs light up and turn off when the buttons are depressed.

Integration (Jonathan Northrup):

A small python test script was made to check the functionality of the buttons in the system. In testing the buttons we were also able to test the input functionality of the raspberry pi GPIO pins

Colored LEDs (RED, YELLOW, GREEN)

Description:

The Facial Recognition Ignition System uses three LEDs indicators to show state changes. LEDs are a semiconductor light source that emits light when current flows through it.

Testing and Calibration (Jonathan Northrup and Daniel Leach):

To test the LEDs they were applied to the test circuit that was used in the previous section.

Integration (Jonathan Northrup and Daniel Leach):

Integrating the LED into the system was necessary to show the user if their face was recognized or not, or if the system is ready for facial detection.

Potentiometer

Description 10 Ohm potentiometer :

A potentiometer is an adjustable voltage divider used for measuring voltage. It allows for the user to adjust voltage to a desired amount.

Testing and Calibration (Jonathan Northrup and Daniel Leach):

The testing of the potentiometer was done in sync with the 16x2 LCD. This was necessary to control the contrast of the LCD screen.

330 Ohm Resistors

Description:

The 330 Ohm resistors are ballast resistors. They are used to limit the current through LEDs and prevent them from burning out.

Testing and Calibration (Jonathan Northrup and Daniel Leach):
The testing of the resistors was done in sync with with the LEDs and was applied to the test circuit that was created in for testing buttons and LEDs

Integration (Jonathan Northrup and Daniel Leach):
The resistors are integrated into the system to ensure that the LEDs work properly.

3.2 Software

When developing the Facial Recognition Ignition System on the Raspberry Pi the software involved to complete the project included one main software and numerous libraries to make it work seamlessly.

The main software used in this project was OpenCV. OpenCV is an open source computer vision software that is popular among many computer scientists that are studying the field of computer vision and AI. The software was developed by the Intel Research Initiative, the sole purpose of developing OpenCV was to advance CPU-intensive technology. OpenCV officially launched at the IEEE conference on Computer Vision and Pattern Recognition in 2000.The software is currently maintained by the non-profit foundation OpenCV.org. OpenCV was used in the Facial Recognition Ignition System for gathering images for a training dataset and was also used as the main source for face detection and recognition.

Other sources for the system include the Adafruit_Python_CharLCD library and the RPi.GPIO LIbrary. The Adafruit_Python_CharLCD library was used to help control the 16x2 LCD with the Raspberry Pi microcomputer. The Adafruit library was helpful in the development of the system to allow easy communication between the system and user via the 16x2 LCD screen. The Adafruit LCD library was developed and maintained by Adafruit Industries until november 21, 2018 and has since been deprecated. It has now been replaced by a new library called Adafruit_CircuitPython_CharLCD. The new library was not used in this system because it contains many unnecessary components that would not be used in this system hence by using the older library we can decrease storage space on the system needed for the library.

The RPi.GPIO library is a library developed by Ben Croston and was the official GPIO library for the Raspberry Pi until it was replaced by the GPIOZero library that is now installed by default on the Raspberry PI OS. The RPi.GPIO library was used in this project to allow easy control of the GPIO pins in the RPI. The reason for the use of the older library on this system was to allow the developers to quickly be able to work on the system without having to learn the new library which would take time away from the essential work that was being done on the system.

The use of OpenCV software, Adafruit_Python_CharLCD, and the RPi.GPIO libraries allowed for the systems complete software to be developed. Using python to write several scripts for image collection, facial detection, facial training, and facial recognition.

The system that was developed involved all scripts as well as a control script to complete the entire system. The Image collection script uses OpenCV to allow the use of the camera to take images of the person to be recognized. The script also needed access to the OS file system so all images taken would be stored in a dataset. The image collection script includes preprocessing the images to allow for faster training. The images stored are labeled, resized and converted to gray scale then sent to the dataset. The image collection script takes about 200 images of the user but can be changed to include more. Once the images are taken, labeled, and preprocessed they are sent to a file that acts as the dataset for our system

The facial detection script was more of a test script but was incorporated into the system as the base for collecting the required face images in the image collection script and in the facial recognition script. The facial training script was used in training a pretrained model via transfer learning to allow the new model to recognize the user to be identified. The facial recognition script was used as the base for the main control script. This script allows the system to use the newly pretrained model to identify the user. The control script is the main script controlling all switch inputs, messages, and acts as the driving logic behind the scenes to keep everything working together.

3.3 User Interface

The user interface (UI) is the main interface between the user and the system. The Facial Recognition Ignition System works by allowing the user to start an automobile by having their face identified by the system.

The prototype works by waiting for the program start message, This message is to show the user that the program has started and it is ok to press the open button. Once the button is pressed the camera is initialized, a message appears on the display that says “System Start” and the process of identifying the user starts. The three indicator lights allow the user to quickly know if the system has or hasn’t identified the user. If the user is identified the green LED indicator light is turned on and the message “User Identified” is displayed on the screen, if they are not identified the red LED indicator light is turned on and the message “Unknown User” is displayed on the screen. Once the system has identified the correct user then that user is allowed to press the start ignition button. If the user presses the ignition button prior to being identified then the system ignores that request. Only once the user has been identified that the ignition button will allow the engine to be started.

In a real world installation of the system the automobile would identify if a user has entered the vehicle via its door indicator sensor. It would turn on the camera and to identify if a person is actually in the driver's seat. After the seatbelt warning switch is deactivated, the system is ready once facial recognition is achieved. If the system has detected a person then the process of Identifying the user would take place. The automobile would display the camera image on the dashboard screen if possible, for automobiles that do not have this capability the indicator lights would then assist the user as well as displaying messages in the radios OLED/LCD screen which are common in older vehicles. Once the person is identified as a valid user and switch conditions are met, the vehicle will then start

In addition to allowing the user to be able to start the vehicle, other permission could be tied into the system such as settings and radio privileges. The Facial Recognition Ignition System could also be used to instantly change to the preferred user setting upon identification.

3.4 Data Communication

In the Facial Recognition Ignition System, the data that the system uses is stored on the system. The data is collected by a dataset collection script that is to be run in configuring the system. The data that is collected are images that are used in the training process that will later be used to identify users.

The reason why the data is stored locally on the system and not in an off site database is because when working with vehicles they usually do not have internet access. This allows for the user to be able to set up the identification system without having to go through the process on their phone. Having the dataset on the system allows for faster training as well. The decision to keep the data on the system was justified by the fact that many cars do not have internet access and the user would need a phone to start the process of configuring new users. Using a phone to do the initial configuration is an option that we could consider later.

In newer automobiles that have access to the internet another solution would be to use Google's Firebase to store the data and train the model that would be used on the system. This would free up limited resources in the car. The only downside would be if the automobile is in a bad location that does not receive a decent internet signal the system could fail.

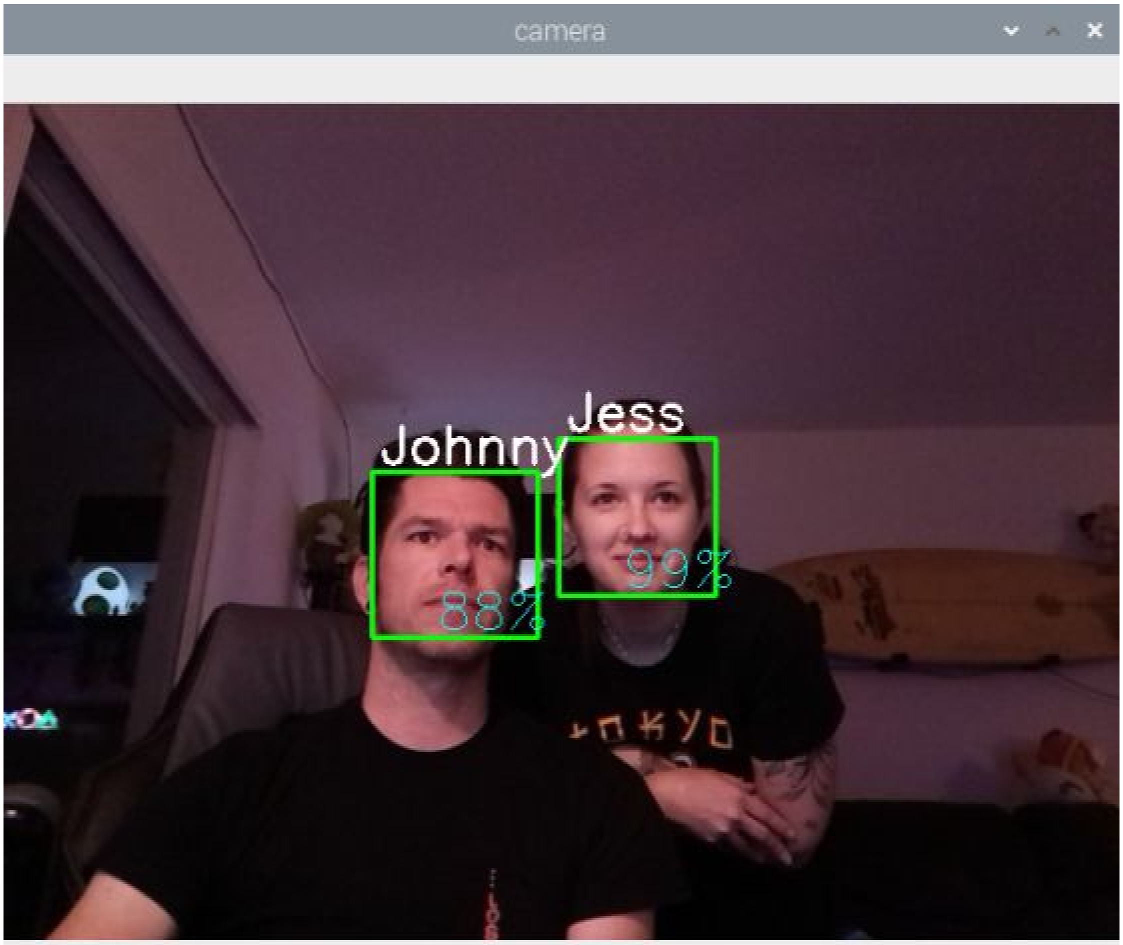
4 Testing and Performance Evaluation

4.1 System Testing

When testing the Facial Recognition Ignition System each component used was tested individually and then integrated into a testing circuit. The circuit was used to ensure that each component would work within the system. The facial recognition component of the system was first tested on the raspberry pi to ensure that camera and OpenCV software would work on the Raspberry Pi microcomputer. The system was tested multiple times to verify that no issues were present in the system. The Facial recognition component was further developed on a PC to allow for faster training and testing.

4.2 Performance Assessment

During the testing process we found that the base facial classification model that comes with the OpenCV software was good enough to detect user faces. This was ok for component testing in the system. A training script was built for transfer learning and was later improved on the PC. To improve the accuracy of the model the dataset was increased with multiple images in different lighting and positions to allow a more accurate identification of the user. The performance accuracy was reasonable for what we are trying to achieve.


camera
Jess
Johnny
99%
88%

5 Budget

Component	Quantity	Price
Raspberry Pi 3 b+	1	$35.00
Raspberry Pi Camera V2	1	25.00
16x2 LCD Display	1	5.99
Solderless Plug-in Breadboard	1	7.90
Tactile Button Switch (6mm)	2	0.26
Green LED	1	0.40
Red LED	1	0.40
Yellow LED	1	0.40
330 Ohm Resistors	3	0.18
10K Ohm Potentiometer	1	0.69
Total		$ 76.22

6 Project Management and Workload Distribution

The team is made up of two computer science majors, Daniel Leach and Jonathan Northrup. The workload was split in half given that there are only two members. We decided that we should have the workload be split into several parts and that each member would be responsible for completing certain tasks within those parts.. The workload was split into research, implementation, testing, and documentation. Within each section we worked together to achieve our goal.

Research	Researched Training Models	Daniel Leach Jonathan Northrup	12.5% 12.5%
Implementation	Project Build	Daniel Leach	25%
Testing	Components	Jonathan Northrup	25%
Documentation	Update, Slides,	Daniel Leach	12.5%

	Project Report	Jonathan Northrup	12.5%

**** Due to Covid-19 restrictions this project was researched together online, built at one location and handed off for testing at a separate location. All documentation was done together online during all phases of the project. ****

7 References

DiCola, T. (n.d.). Character LCD with Raspberry Pi or BeagleBone Black. Retrieved August 08, 2020, from https://learn.adafruit.com/character-lcd-with-raspberry-pi-or-beaglebone-black/usage

OpenCV. (2020, August 04). Retrieved August 08, 2020, from https://opencv.org/

RPi.GPIO. (n.d.). Retrieved August 08, 2020, from https://pypi.org/project/RPi.GPIO/