A Project Report on

"Face Mask and Temperature Detection using IoT"

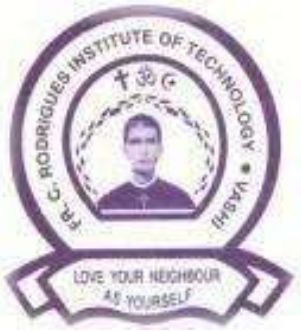
Submitted in partial fulfillment of the requirement for Degree in Bachelor of Engineering (Information Technology)

By

Sanika Deshmukh (501872)
Hritesh Sonawane (501864)
Vinay Joy (501751)

Guided by:

Prof. Sharlene Rebeiro



Department of Information Technology

Fr. Conceicao Rodrigues Institute of Technology

Sector 9A, Vashi, Navi Mumbai–400703

University of Mumbai
2020-2021

CERTIFICATE

This is to certify that the project entitled

Face mask and temperature detection using IoT

Submitted By

Sanika Deshmukh (501872)
Hritesh Sonawane (501864)
Vinay Joy (501871)

In partial fulfillment of degree of **B.E.** in **Information Technology** for term work of the project is approved.

External Examiner **Internal Examiner**

_____ _____

Internal Guide

_____

Head of the Department **Principal**

_____ _____

Date: - **College Seal**

DECLARATION

We declare that this written submission represents our ideas in our own words and where others ideas or words have been included; we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

.

Sanika Deshmukh
(501872)

Hritesh
Sonawane(501864)

Vinay Joy (501871)

Date:12/05/2021

ABSTRACT

COVID19 is spreading rapidly in the world, even breaking out in the communities and big cities. To confront the severe epidemic situation, a system of surveillance to check whether people are wearing a face mask is of utmost importance. Now that many shops, offices and institutions are re-opening again after the COVID19 lockdown, many businesses are faced with the need to provide the best possible protection for their staff and customers. Face masks and body temperature checks play an important part in the protection effort. Wearing a face mask is the most sought-after prevention measure considering the factors like availability, cost, recyclable and many more. Face mask monitoring often requires additional staff resources. Having a high body temperature is a symptom of COVID-19 too. At the same time, body temperature checks by staff come with certain risks in terms of hygiene, and a system to eliminate this system can help prevent transmission of COVID-19.

Keywords: Face mask, Temperature, Arduino

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

INTRODUCTION

The pandemic which was unforeseen, has brought drastic changes in our lifestyles and ensuring hygiene is one of the good things. However, the rapid spread of the COVID virus has made working in close quarters difficult since social distancing has to be followed. The outbreak has made many of us aware of how important face masks are. However, if we wear the mask in a floppy manner that doesn't cover your mouth and nose properly, the risk of catching the virus increases exponentially. It is mandatory to check temperature before entering any premises to ensure that one is causing no harm by transmitting the virus. Unknowingly, face masks have become an integral part of our lives, and ensuring everyone wears one until the pandemic ends, is a task we must all look forward to. Checking both, the body temperature and the proper use of face masks can be done by the proposed system, which eliminates the risk of virus transmission on priority basis at gatherings, educational or office spaces. We need a mechanism which checks the face mask and temperature of a person rather than a temperature gun guardian.

This project deals with the problem of virus transmission by keeping a strict check on the use of face masks by an optimized face mask detection system coupled with the check of body temperature which ensures that the basic means of preventing the COVID-19 virus are under check. This project enables the organizations to meet their needs of allowing employees to work in a hostile environment where people wear their mask properly and alarms if the body temperature or mask is not worn properly. If either of the criteria is not followed, the system will alarm a buzzer alerting the user to maintain safety guidelines.

Chapter 2

LITERATURE SURVEY AND ANALYSIS

LITERATURE SURVEY

AND ANALYSIS

Due to the COVID-19 pandemic, wearing a mask is mandatory in public spaces, as properly wearing a mask offers a maximum preventive effect against viral transmission of the virus. Body temperature has also become an important consideration in determining whether an individual is healthy. Automatic Face Mask Detection and Temperature Detection system that can check if a person is wearing a face mask and is having a normal body temperature will be allowed at the entry point of any place (Colleges, Restaurants, Residential Building). As the growth in the number of people being affected it is the need of the hour. Many researchers have implemented such systems with various technology and functionalities. The papers regarding the methods are referenced as well as the basic idea of implementation:

Smaranjit Ghose and Suhrid Datta in paper entitled Automatic Facial Mask Detection using Deep Learning, authors have described using a Convolutional network which is used to extract meaningful features from images and differentiate amongst them by using certain learning parameters for face mask detection and IR sensor connected to Raspberry Pi for detection of body temperature and a low-cost camera connected to Raspberry Pi for getting Real time Images/Video for detection of face mask. Authors have also used TensorFlow to create large-scale neural networks with many layers. Dataset used was of 10,000 subjects both with mask and without mask. For Pre-processing they have used feature classifiers like Hue, Saturation, Value format with threshold of 5. For Model Architecture they used 3 players on 100 epochs, ReLU activation function on the last layer. They achieved 100% accuracy with precision of 1.0 for both with mask and without mask.[1].

Isack Farady, Chih-Yang Lin, Amornthep Rojanasarit in paper entitled Mask Classification and Head Temperature Detection Combined with Deep Learning Networks have built two modules first is Face Mask Detection and second is RGB Head Thermal temperature detector. For Face Mask Detection, the RetinaNet has CNN backbone and U-Net (Object Detection) to detect three categories of mask-wearing positions and the

temperature of the head. The main aim of the Face Mask detector module is to produce the exact mask-wearing position and head position of an input image, in order to determine the class prediction of the network. Second is done by implementing an RGB thermal camera to generate input images and capture a person's temperature respectively with a live camera as an input of test images. RGB camera fed with a public medical mask dataset from the Kaggle dataset. The dataset contains 678 images with annotations and 3 class labels y (good, bad, none) that indicate mask-wearing position. They have only considered taking the highest temperature inside the bounding box prediction rather than classifying the class image. The main task of Module Thermal RGB module is to detect the head class after obtaining the maximum temperature value inside the bounding box prediction. Face Mask detection and Thermal RGB modules are combined and we get our output in Image/Video format with accuracy of Face Mask Detection Module with 81.31% and Head Temperature Detection with 99.7% [2].

Citations:

[1] Automatic Facial Mask Detection using Deep Learning 1 st Smaranjit Ghose 2 nd Suhrid Datta

[2] Mask Classification and Head Temperature Detection Combined with Deep Learning Networks 1st Isack Farady 2nd Chih-Yang Lin 3rd Amornthep Rojanasarit 4th Kanatip Prompol 5th Fityanul Akhyar

Comparison Chart:

Parameters	Paper [1]	Paper [2]
Deep Learning Model Used	Mobile Net	Retina Net and RetinaNet50
Threshold	5	0.5
Neural Network	CNN (3x3)	CNN(3x3)
Activation Function	ReLU	Sigmoid
No of Classes	2 (With Mask, Without Mask)	3 (Good, Bad, None)
Accuracy	99%	81.31%
Hardware Used	Raspberry Pi and low-cost camera	High resolution camera
Temperature Detection	IR Sensor	RGB Thermal Camera
Temperature	Head or Hand Temperature	Head Temperature
Subjects in dataset	10000	678
Realtime data	Images	Video

Chapter 3
PROBLEM STATEMENT

PROBLEM STATEMENT

In the new world of coronavirus, multidisciplinary efforts have been organized to slow the spread of the pandemic. The developer community has also been a part of these endeavors and finding an all-round solution to this problem at a large scale. In particular, developments for monitoring social distancing or identifying face masks have made the headlines. To compensate for the lack of clarity of the images in still frames, we can think of actually using several different frames of the video to make a better mask/no mask decision.

Following problems are foreseen to resolve using a face mask and temperature detection system:

- Detect people that pass through a security-like camera.
- Identify face mask usage.
- Check body temperature.
- Alert by ringing the buzzer if values below threshold

Chapter 4

REQUIREMENTS

REQUIREMENTS

Sr. No.	Hardware Requirements	Sensors	Software Requirements
1	Buzzer Alarm	Ultrasonic Sensor	Arduino IDE
2	Camera (OV7670 CAMERA)	IR Temperature Sensor	OpenCV
3	Arduino UNO		TensorFlow
4	LCD 16x2		Keras
5			Object Detection Model (ImageNet)

Table 1: Requirements

4.1 Hardware Requirements:

1. Buzzer (Alarm):

A **buzzer** or **beeper** is an audio signaling device, which may be mechanical, electromechanical, or piezoelectric (*piezo* for short). Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke

Buzzer has the Rated voltage 6v Dc and its operating Voltage is 4-8v DC, its rated current is <30mA sound type is like a continuous Beep.

Fig 4.1 Buzzer Alarm

2. Camera (OV7670 CAMERA):

OV7670 Camera Module (Chinese) is a small size, highly sensitive and low voltage CMOS image Sensor module for capturing and processing the image. It is based on an OV7670 image sensor. Through SCCB bus control, the sensor can output the whole frame, sampling, and various resolutions of 8 bits of data. The product VGA image can reach up to a maximum of 30 frames per second. Users can completely control the image quality, data format and transmission mode. All the process of image processing functions can be through the SCCB programming interface, including gamma curve, white balance, saturation and Chroma. The OV7670 is a low-cost image sensor DSP that can operate at a maximum of 30 fps and 640 x 480 ("VGA") resolutions, equivalent to 0.3 Megapixels. The captured image can be pre-processed by the DSP before sending it out. This preprocessing can be configured via the Serial Camera Control Bus (SCCB).

Fig 4.2 Camera (OV7670 CAMERA)

3. Arduino UNO:

The **Arduino UNO** is the best board to get started with electronics and coding. If this is your first experience tinkering with the platform, the UNO is the most robust board you can start playing with. The UNO is the most used and documented board of the whole Arduino family. **Arduino Uno** is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. You can tinker with your Uno without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

Fig 4.3 Arduino UNO

4. LCD 16x2:

An LCD (Liquid Crystal Display) screen is an electronic display module and has a wide range of applications. A 16x2 LCD display is a very basic module and is very commonly used in various devices and circuits. A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in a 5x7 pixel matrix. The 16 x 2 intelligent alphanumeric dot matrix display is capable of displaying 224 different characters and symbols. This LCD has two registers, namely, Command and Data. Command registers stores various commands given to the display. Data register stores data to be displayed. The process of controlling the display involves putting the data that form the image of what you want to display into the data registers, then putting instructions in the instruction register. In your Arduino project Liquid Crystal Library simplifies this for you so you don't need to know the low-level instructions. Contrast of the display can be adjusted by adjusting the potentiometer to be connected across VEE pins.

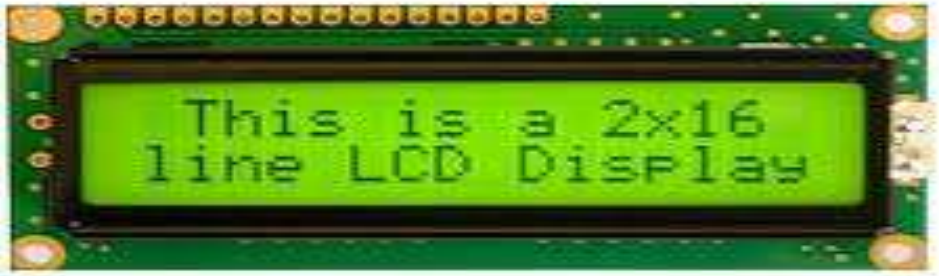


Fig 4.4 LCD 16 X 2

Sensors:

1. Ultrasonic sensor:

An ultrasonic sensor is an instrument that measures the distance to an object using ultrasonic sound waves. An ultrasonic sensor uses a transducer to send and receive ultrasonic pulses that relay back information about an object's proximity.

High-frequency sound waves reflect from boundaries to produce distinct echo patterns. Ultrasonic sensors work by sending out a sound wave at a frequency above the range of human hearing. The transducer of the sensor acts as a microphone to receive and send the ultrasonic sound. Our ultrasonic sensors, like many others, use a single transducer to send a pulse and to receive the echo. The sensor determines the distance to a target by measuring time lapses between the sending and receiving of the ultrasonic pulse.

Fig 4.5 Ultrasonic Sensor

2. Temperature Sensor - IR:

Thermopile sensors are designed to measure temperature from a distance by detecting an object's infrared (IR) energy. The higher the temperature, the more IR energy is emitted. The thermopile sensing element, composed of small thermocouples on a silicon chip, absorbs the

energy and produces an output signal. A reference sensor is designed into the package as a reference for compensation. Thermopile infrared (IR) temperature sensors measure non-contact temperature and are available with various lenses, and filters allowing use in multiple applications, from industrial pyrometers, to climate controls and medical devices. TE Connectivity's (TE) thermopile infrared sensors provide reliable non-contact temperature measurement solutions.

Fig 4.6 IR Temperature Sensor

4.2. Software Requirements:

1. Arduino IDE:

The Arduino integrated development environment is a cross-platform application that is written in the programming language Java. It is used to write and upload programs to the Arduino board. The source code for the IDE is released under the GNU General Public License, version 2. Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards ('shields') or breadboards (For prototyping) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers can be programmed using C and C++ programming languages.

Fig 4.2.1Arduino IDE

2. OpenCV:

OpenCV (Open-Source Computer Vision Library) is an open-source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

3.Tensorflow:

TensorFlow is an open-source library developed by Google primarily for deep learning applications. It also supports traditional machine learning. TensorFlow was originally developed for large numerical computations without keeping deep learning in mind. However, it proved to be very useful for deep learning development as well, and therefore Google open-sourced it. TensorFlow accepts data in the form of multi-dimensional arrays of higher dimensions called tensors. Multi-dimensional arrays are very handy in handling large amounts of data. TensorFlow works on the basis of data flow graphs that have nodes and edges. As the execution mechanism is in the form of graphs, it is much easier to execute TensorFlow code in a distributed manner across a cluster of computers while using GPUs.

4.Keras:

Keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear and actionable feedback upon user error. This makes Keras easy to learn and easy to use. As a Keras user, you are more productive, allowing you to try more ideas than your competition, faster -- which in turn helps you win machine learning competitions. This ease of use does not come at the cost of reduced flexibility: because Keras integrates deeply with low-level TensorFlow functionality, it enables you to develop highly hackable workflows where any piece of functionality can be customized.

6.Object Detection Model (ImageNet):

ImageNet is formally a project aimed at labeling and categorizing images into separate object categories for the purpose of computer vision research. The goal of this image classification challenge is to train a model that can correctly classify an input image category represent object classes that we encounter in our day-to-day lives, such as species of dogs, cats, various household objects, vehicle types, and much more. In Machine Learning and Deep Neural Networks, machines are trained on a vast dataset of various images. Machines are required to learn useful features from these training images. Once learned, they can use these features to classify images and perform many other tasks associated with computer vision. ImageNet gives researchers a common set of images to benchmark their models and algorithms. It’s fair to say that ImageNet has played an important role in the advancement of computer vision.

Chapter 5
SYSTEM DESIGN

SYSTEM DESIGN

Circuit Diagram:

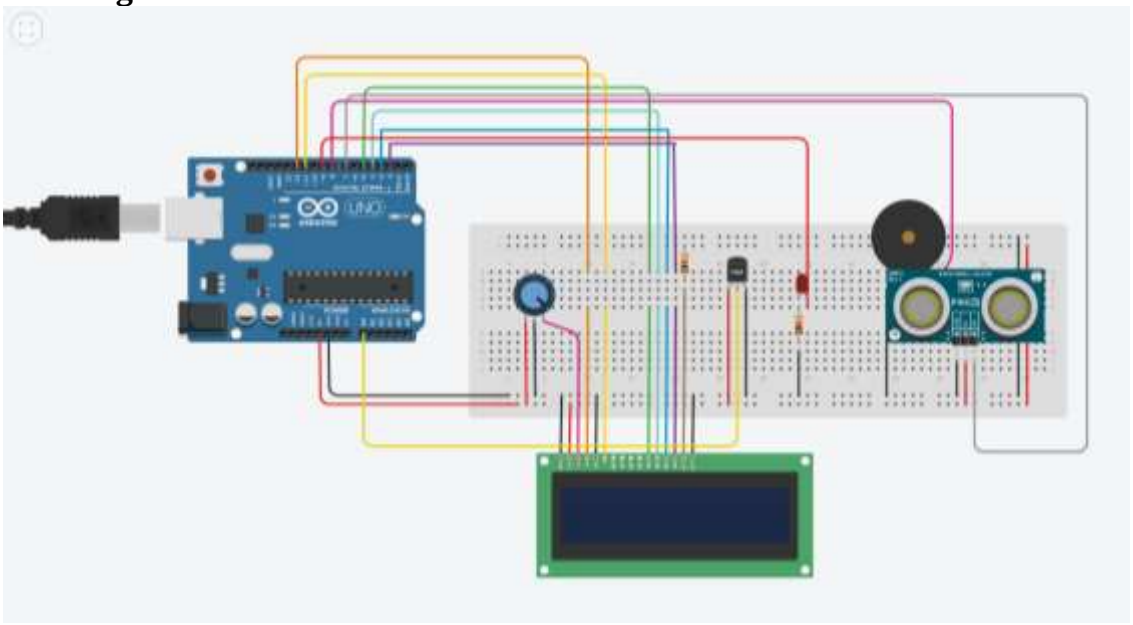


Fig 5.1 Circuit Diagram

Chapter 6
IMPLEMENTATION

IMPLEMENTATION

We have used tinker cad for showing the demonstration of our project:

Below are the connections:

- Arduino uno is connected to the bread board by a 5V Voltage and GND pin that is ground pin of Arduino
- These lines will be considered to connect VCC or GND for other components as well
- We connect 12,11,5,4,3,2 pins to Arduino for LCD to Arduino with a potentiometer on pin V0 and resistor on pin LED of LCD
- A0 is given the sense pin for temperature sensor to the Arduino
- One red LED to display output on pin 9 of Arduino is connected
- pin 7 connected to Arduino to read input given to the ultrasonic sensor
- pin 8 is given to output buzzer

Given condition is if temperature is greater than 99.6 and the distance is less than 15 cm the buzzer will start beeping to notify that person's temperature is greater than 99.6 Fahrenheit and LED will glow

Fig 6.1 Code Arduino

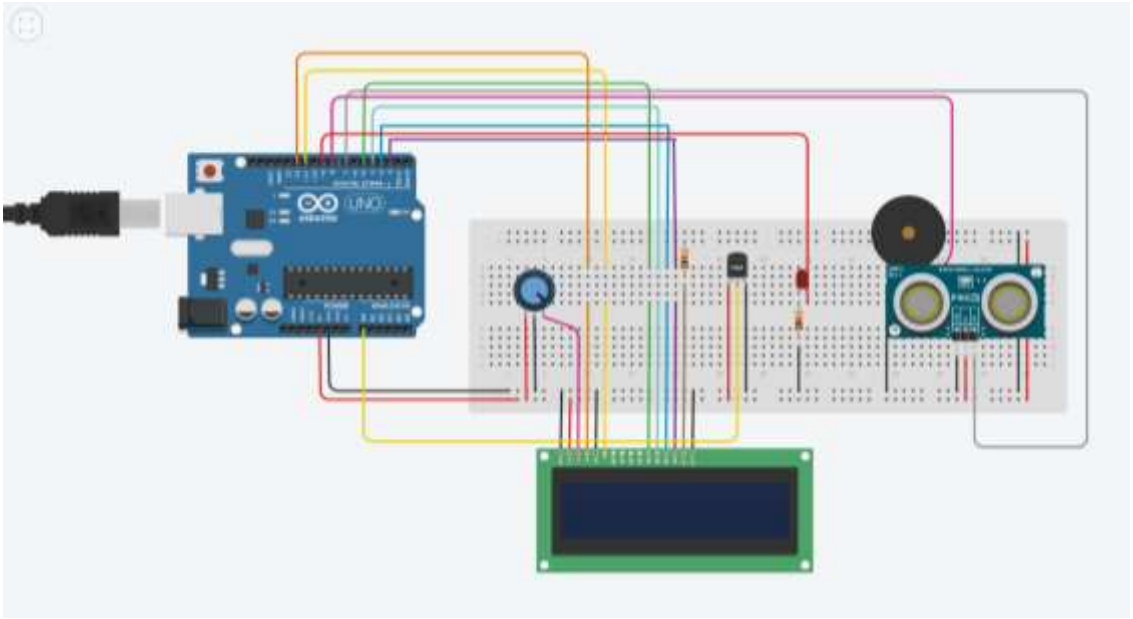


Fig 6.2 System Design

Frames of a real time video is passed to the face mask detection model which detects the face mask with help of image net model and gives a beep sound when the mask is not seen in the frame both image and video can be passed as an input

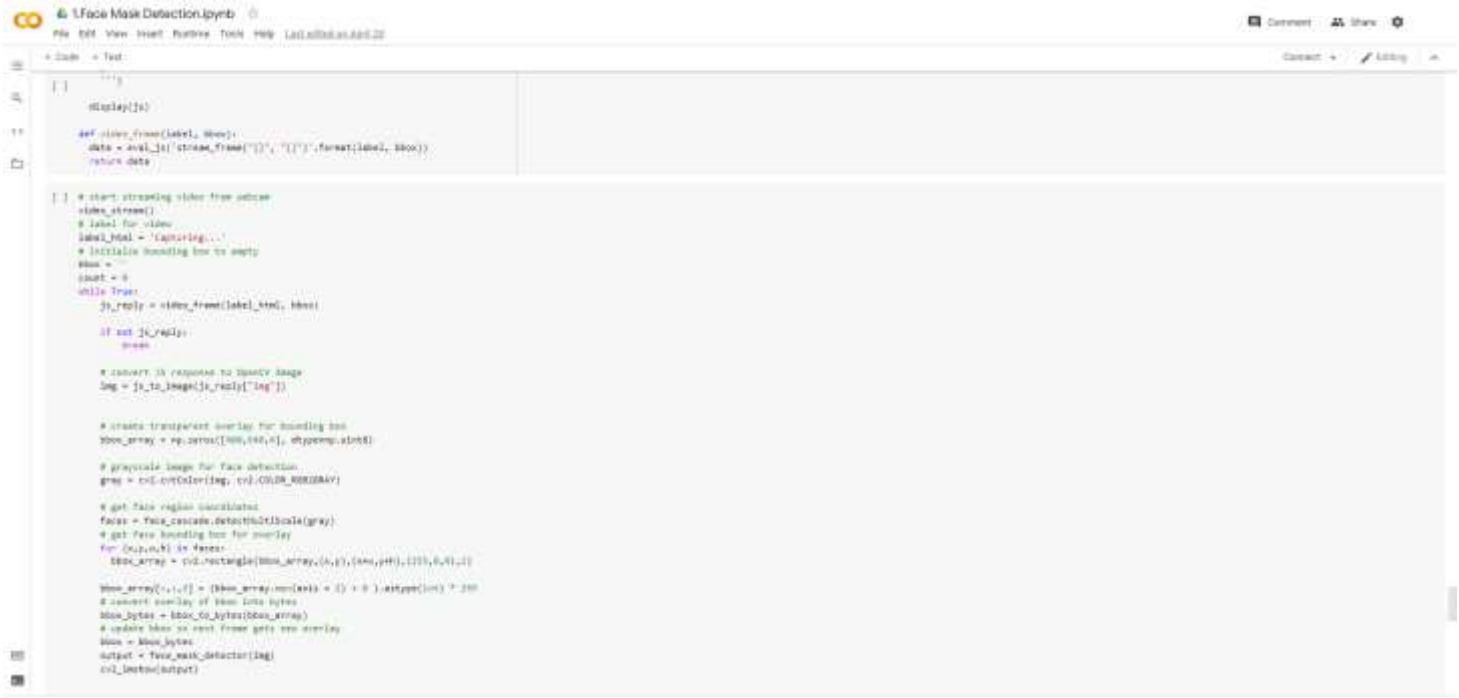


Fig 6.3 Code Face Mask Detection

Code:

```
#include <LiquidCrystal.h>
//Initialize the library with the numbers of the interface pins.
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
//red led
int redLed = 9 ;
const int pingPin = 7;
//This is the Arduino Pin that will read the sensor output.
int sensePin = A0;
//The variable we will use to store the sensor input.
int sensorInput;
//The variable we will use to store temperature in degrees.
double temp;
//stores microseconds
long duration;
//stores converted ms into cm
long cm;
void setup()
{
 //pin mode of red led
pinMode(9, OUTPUT);
//Initialize the LCD's number of columns and rows.
lcd.begin(16, 2);
 //Start the Serial Port at 9600 baud (default).
  Serial.begin(9600);
}

void loop()
{
  //Set the cursor to column 0, line 0
  lcd.setCursor(0, 0);

  // The PING))) is triggered by a HIGH pulse of 2 or more microseconds.
  // Give a short LOW pulse beforehand to ensure a clean HIGH pulse:
  pinMode(pingPin, OUTPUT);
  digitalWrite(pingPin, LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin, HIGH);
  delayMicroseconds(5);
  digitalWrite(pingPin, LOW);
```

```
// The same pin is used to read the signal from the PING))): a HIGH
// pulse whose duration is the time (in microseconds) from the sending
// of the ping to the reception of its echo off of an object.
pinMode(pingPin, INPUT);
duration = pulseIn(pingPin, HIGH);

// convert the time into a distance
cm = microsecondsToCentimeters(duration);
//Read the analog sensor and store it.
sensorInput = analogRead(A0);
float volt = sensorInput * 5;
//Multiply by 5V to get voltage.
volt /= 1024;
Serial.print(volt);
Serial.println(" volts ");

//Subtract the offset.
temp = volt - 0.5;
//Convert to degrees.
temp = temp * 100;
//formula for farheneit
temp = (9.0/5)*(temp) + 32;

Serial.print("Temperature ") ;
Serial.print(temp);
Serial.println("");

// printing on the lcd screen the word "temperature"
lcd.print("Temperature: ");

//Set the cursor to column 0, line 1
lcd.setCursor(0, 1);
// Printing the temperature on the lcd screen
lcd.print(temp);
// printing the name of the scale used for temperature
lcd.print("F");

// if the temperature (var associated: temp) is superior
// to 99.6, then we enter the if loop
if (temp >= 99.6 && cm < 50)
{

  // turning on the red LED, stated as 'high'
```

```
   digitalWrite(redLed, HIGH);
   //INPUT - FREQUENCY - TIME THAT LASTS
   tone(8, 1000, 200);
  }

  // otherwise, if the temperature is below 99.6
  else if (temp < 99.6 && cm > 50)
  {

   // turning on the red LED, stated as 'low'
    digitalWrite(redLed, LOW);

    noTone(8);
  }

  delay(1000);


}
long microsecondsToCentimeters(long microseconds) {
  // The speed of sound is 340 m/s or 29 microseconds per centimeter.
  // The ping travels out and back, so to find the distance of the
  // object we take half of the distance travelled.
  return microseconds / 29 / 2;}
```

Chapter 7

RESULT

RESULT

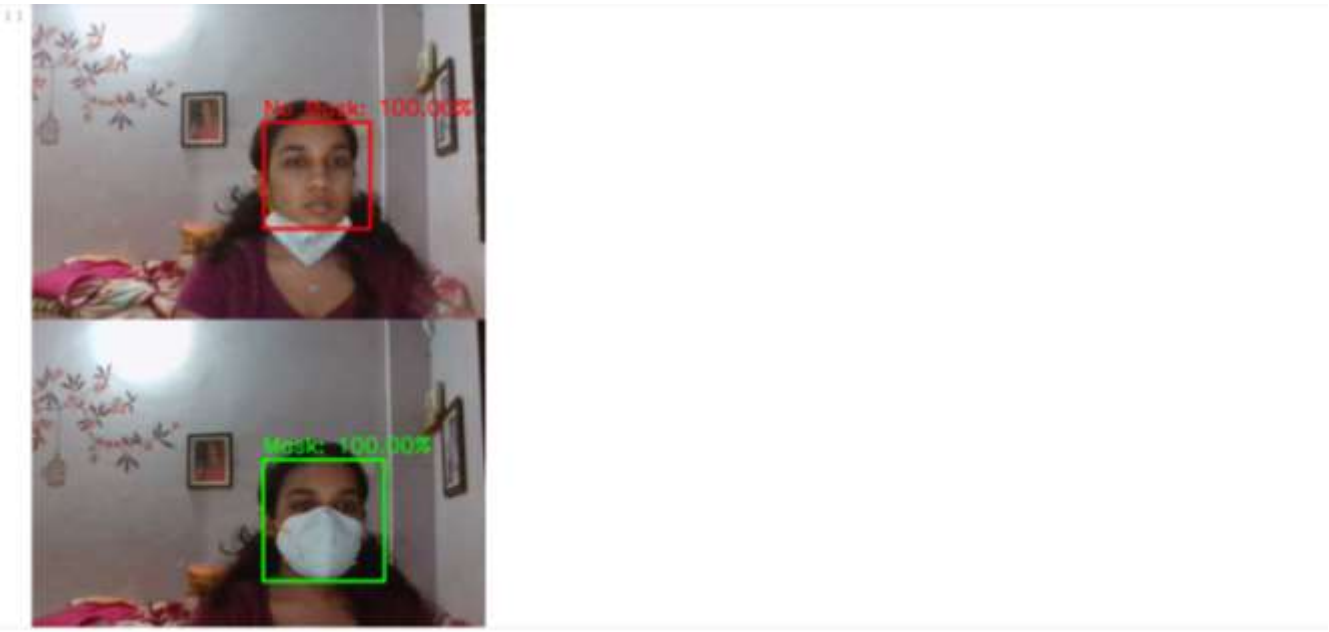


Fig 7.1 Result Face Mask Detector

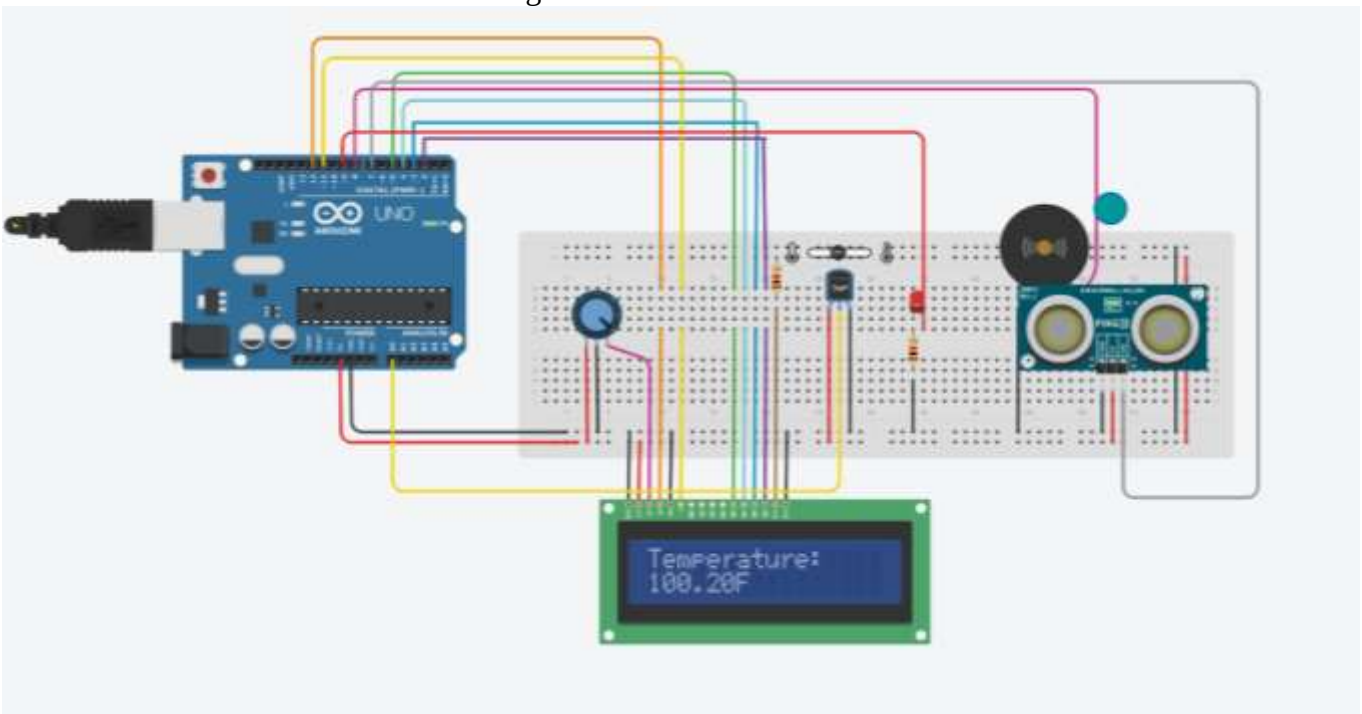


Fig 7.2 Result Temperature Sensor with Buzzer

ACKNOWLEDGEMENT

The success and final outcome of Face Mask and Temperature Detection using IoT required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of our project. All that we have done is only due to such supervision and assistance and we would not forget to thank them

.
We respect and thank Dr. Khot, Principal of Fr. C. Rodrigues Institute of Technology, Prof. Dhanashree Hadsul, H.O.D. of IT department and Prof. Kalpana Wani, Coordinator of IT department for extending their inevitable and valuable support to us.

We owe our deep gratitude to our project guide Prof. Sharlene Rebeiro, who took keen interest on our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system.

We are thankful to and fortunate enough to get constant encouragement support and guidance from all Teaching staffs of Information Technology. Also, we would like to extend our warm thanks to our family members and well-wishers for always supporting and encouraging us.

Yours sincerely,

Sanika Deshmukh

Vinay Joy

Hritesh Sonawane

REFERENCE

[1] https://www.tinkercad.com/things/7KlkHaxx20c-tmp36-temperature-sensor-with-arduino

[2] https://www.tinkercad.com/things/bhkltIIL1Fd-music-using-piezoelectric-buzzer-and-arduino

[3] https://www.tinkercad.com/things/eU5pgW878rd-ultrasonic-sensor

[4] https://ieeexplore.ieee.org/document/8899541

[5] https://medium.com/@i2i-blog/facial-mask-detection-using-deep-learning-and-computer-vision-c0966e14dd94

[6] https://ieeexplore.ieee.org/document/9216386

[7] https://www.scienceopen.com/document?vid=31a5d046-0e15-4d31-b29a-5be869f31144