CHAPTER 1

INTRODUCTION

The situation report 96 of world health organization (WHO) presented that coronavirus disease 2019 (COVID-19) has globally infected over 2.7 million people and caused over 180,000 deaths. In addition, there are several similar large scale serious respiratory diseases, such as severe acute respiratory syndrome (SARS) and the Middle East respiratory syndrome (MERS), which occurred in the past few years . Liu et al. reported that the reproductive number of COVID-19 is higher compared to the SARS. Therefore, more and more people are concerned about their health, and public health is considered as the top priority for governments . Fortunately, Leung et al. showed that the surgical face masks could cut the spread of coronavirus. At the moment, WHO recommends that people should wear face masks if they have respiratory symptoms, or they are taking care of the people with symptoms . Furthermore, many public service providers require customers to use the service only if they wear masks . Therefore, face mask detection has become a crucial computer vision task to help the global society, but research related to face mask detection is limited.

Face mask detection refers to detect whether a person wearing a mask or not and what is the location of the face . The problem is closely related to general object detection to detect the classes of objects and face detection is to detect a particular class of objects, i.e. face . Applications of object and face detection can be found in many areas such as autonomous driving , education , surveillance and so on . Traditional object detectors are usually based on handcrafted feature extractors. Viola Jones detector uses Haar feature with integral image method , while other works adopt different feature extractors, such as histogram of oriented gradients (HOG), scale-invariant feature transform (SIFT) and so on . Recently, deep learning based object detectors demonstrated excellent performance and dominate the development of modern object detectors. Without using prior knowledge for forming feature extractors, deep learning allows neural networks to learn features with an end-to-end manner

There are one-stage and two-stage deep learning based object detectors. One-stage detectors use a single neural network to detect objects, such as single shot detector (SSD) and you only look once (YOLO) . In contrast, two-stage detectors utilize two networks to perform a coarse-to-fine detection, such as region-based convolutional neural network (R-CNN) and faster R-CNN ..

CHAPTER 2

BLOCK DIAGRAM

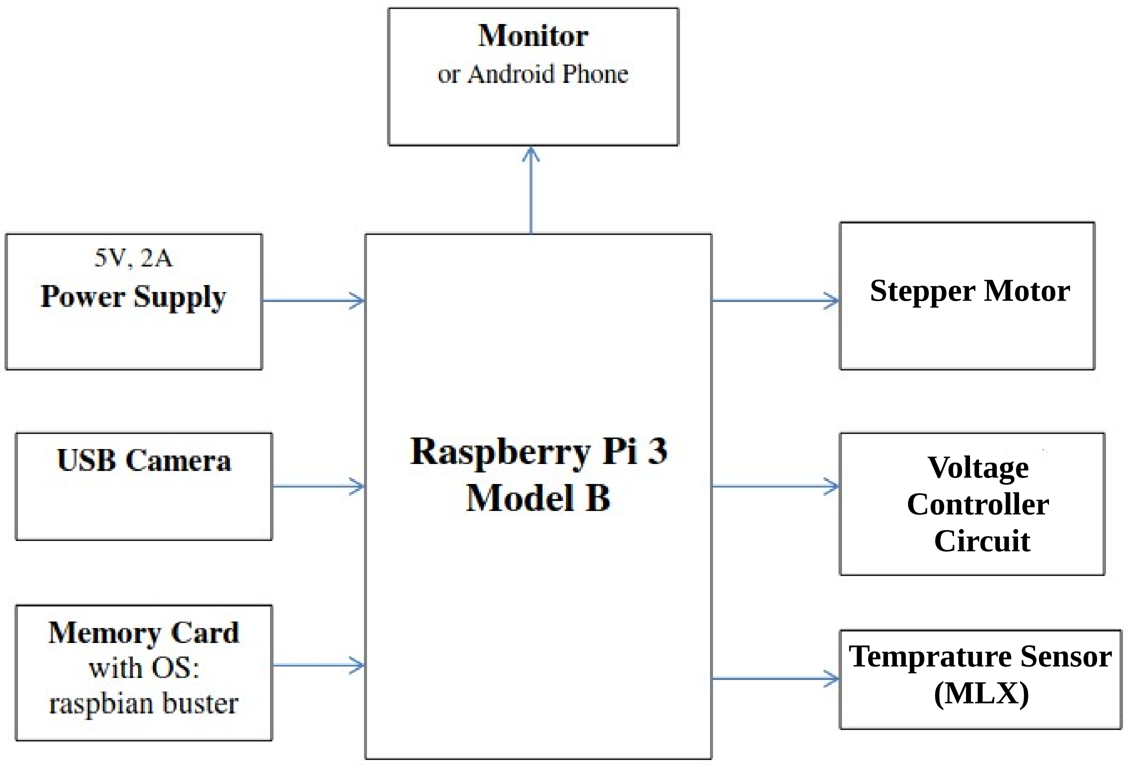


Figure 2.1 Block Diagram

2.1 Working

IOT BASED TEMPRATURE MASK SCAN ENTRY BARRIER first scans for mask in the face and after it detects mask it go for temprature scan if they both ok then the barrior will undo . The is the soul concept of this project and for this to accomplish image procesing and realtime temprature scanning is used . The image processing is done by the help of Raspberry pie and using python 3.x , for this we use only Raspberry pie to monitor and control .In image processing which is the process of transforming an image into a digital form and performing certain operations to get some useful information from it , the OpenCV which is an open source vision and machine learning software library, is also used

2.2 Components used

- Raspberry pie 3 Model B
- USB Webcam
- Stepper Motor
- Voltage Regulator
- Power Supply 12v
- Power Supply 5v

2.3 Software used

- Python IDE
- Real VNC Viewer

CHAPTER 3

HARDWARE DESCRIPTION

3.1 Raspberry Pi

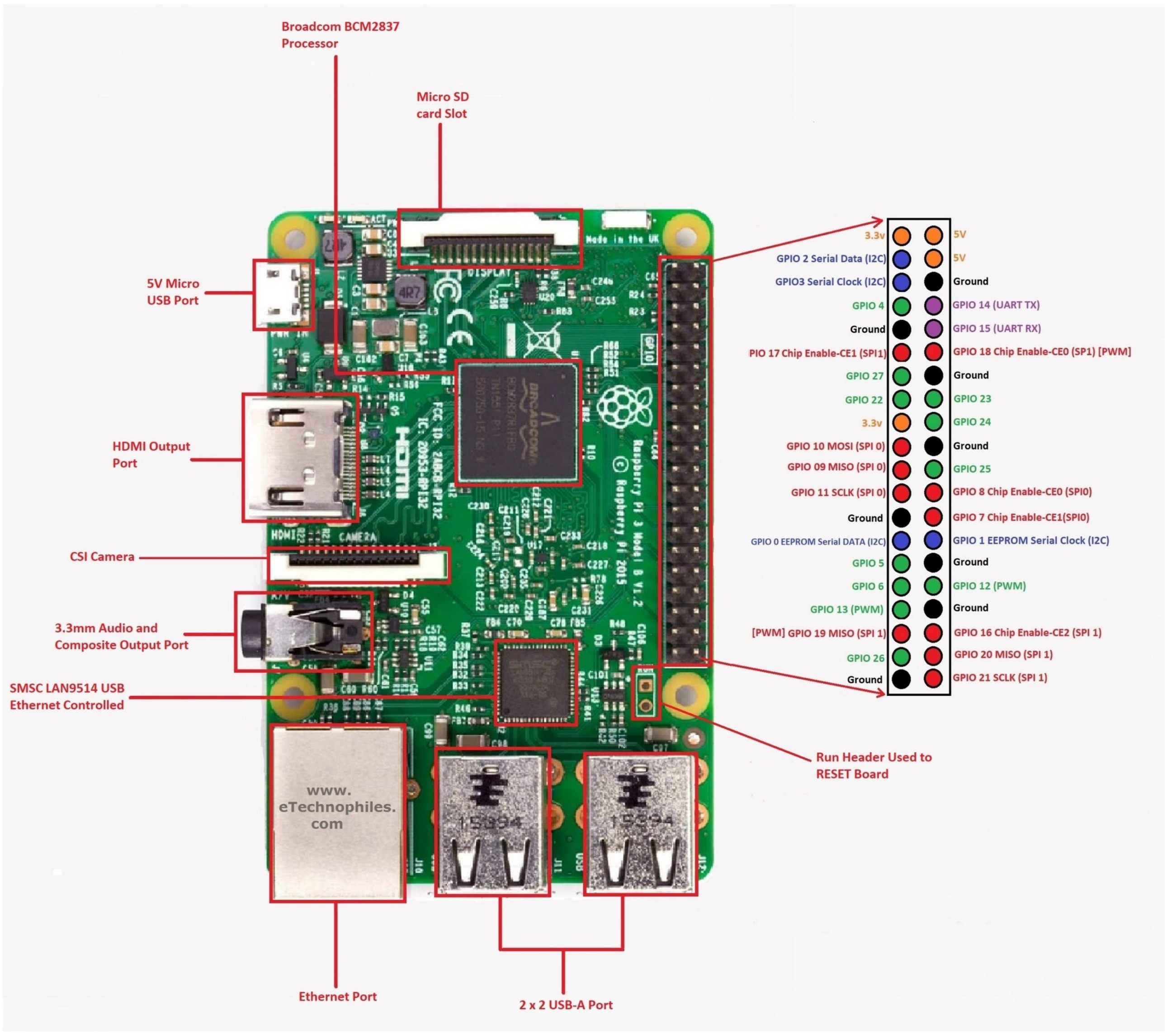


Figure 3.1 Raspberry pi 3 model B board

Raspberry Pi is a small, powerful, cheap, hackable and education-oriented computer board introduced in 2012 . It operates in the same way as a standard PC, requiring a keyboard for command entry, a display unit and a power supply. This credit card-sized computer with many performances and affordable for 25-35$ is perfect platform for interfacing with many devices. The vast majority of the system's components – its central and graphics processing units, audio and communications hardware along with 256 MB (Model A) – 512 MB (Model B) memory chip, are built onto single component. The Raspberry Pi board contains essential (processor, graphics chip, program memory - RAM) and other optional devices (various interfaces and connectors for peripherals). The processor of Raspberry Pi is a 32 bit, 700 MHz System on a Chip, which is built on the ARM11 architecture and can be overclocked for more power [4]. SD

Flash memory serves as a hard drive to Raspberry Pi's processor. The unit is powered via the micro USB connector while internet connectivity may be via an Ethernet/LAN cable or via anUSB dongle (WiFi connectivity) The Raspberry Pi, like any other computer, uses an operating system. The Linux option called Raspbian is a great match for Raspberry Pi because it's free and open source, keeping the price of the platform low, and making it more hackable. There are also a few non-Linux OS options available.

Raspberry Pi 3 model 3 specs

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB RAM
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
- 100 Base Ethernet
- 40-pin extended GPIO
- 4 USB 2 ports
- 4 Pole stereo output and composite video port
- Full size HDMI
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- Micro SD port for loading your operating system and storing data
- Upgraded switched Micro USB power source up to 2.5A

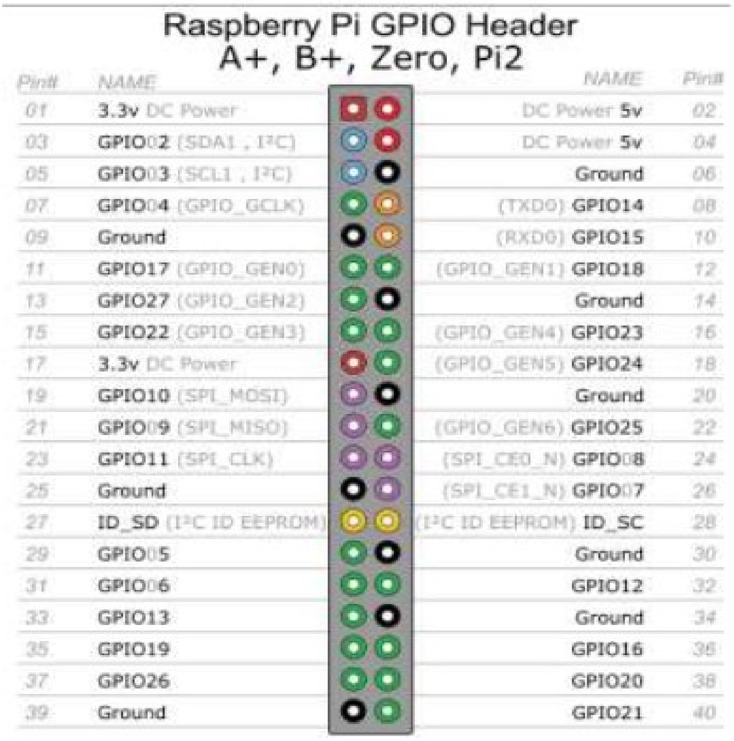

Raspberry Pi GPIO Header
A+, B+, Zero, Pi2
Pin# NAME NAME Pin#
01 3.3v DC Power DC Power 5v 02
03 GPIO02 (SDA1 , I²C) DC Power 5v 04
05 GPIO03 (SCL1 , I²C) Ground 06
07 GPIO04 (GPIO_GCLK) (TXD0) GPIO14 08
09 Ground (RXD0) GPIO15 10
11 GPIO17 (GPIO_GEN0) (GPIO_GEN1) GPIO18 12
13 GPIO27 (GPIO_GEN2) Ground 14
15 GPIO22 (GPIO_GEN3) (GPIO_GEN4) GPIO23 16
17 3.3v DC Power (GPIO_GEN5) GPIO24 18
19 GPIO10 (SPI_MOSI) Ground 20
21 GPIO09 (SPI_MISO) (GPIO_GEN6) GPIO25 22
23 GPIO11 (SPI_CLK) (SPI_CE0_N) GPIO08 24
25 Ground (SPI_CE1_N) GPIO07 26
27 ID_SD (I²C ID EEPROM) (I²C ID EEPROM) ID_SC 28
29 GPIO05 Ground 30
31 GPIO06 GPIO12 32
33 GPIO13 Ground 34
35 GPIO19 GPIO16 36
37 GPIO26 GPIO20 38
39 Ground GPIO21 40

Figure 3.1.1 Raspberry Pi GPIO Headers

The GPIO is the most basic, yet accessible aspect of the Raspberry Pi. GPIO pins are digital which means they can have two states, off or on. They can have a direction to receive or send current (input, output respectively) and we can control the state and direction of the pins using programming languages such as Python, JavaScript, node-RED etc.

The operating voltage of the GPIO pins is 3.3v with a maximum current draw of 16mA. This means that we can safely power one or two LEDs (Light Emitting Diodes) from a single GPIO pin, via a resistor (see resistor color codes). But for anything requiring more current, a DC motor for example, we will need to use external components to ensure that we do not damage the GPIO.

Controlling a GPIO pin with Python is accomplished by first importing a library of pre-written code. The most common library is RPi.GPIO (https://pypi.org/project/RPi.GPIO/) and it has been used to create thousands of projects since the early days of the Raspberry Pi. In more recent times a new library called GPIO Zero (https://pypi.org/project/gpiozero/)has been introduced, offering an easier entry for those new to Python and basic electronics. Both of these libraries come pre-installed with the Raspbian operating system.

GPIO pins have multiple names; the first most obvious reference is their "physical" location on the GPIO. Starting at the top left of the GPIO, and by that we mean the pin nearest to where the micro SD card is inserted, we have physical pin 1 which provides 3v3 power. To the right of that pin is physical pin 2 which provides 5v power. The pin numbers then increase as we move down each column, with pin 1 going to pin 3, 5,7 etc until we reach pin 39. You will quickly see that each pin from 1 to 39 in this column follows an odd number sequence. And for the column starting with pin 2 it will go 4,6,8 etc until it reaches 40. Following an even number sequence. Physical pin numbering is the most basic way to locate a pin, but many of the tutorials written for the Raspberry Pi follow a different numbering sequence.

Broadcom (BCM) pin numbering (aka GPIO pin numbering) seems to be chaotic to the average user. With GPIO17, 22 and 27 following on from each other with little thought to logical numbering. The BCM pin mapping refers to the GPIO pins that have been directly connected to the System on a Chip (SoC) of the Raspberry Pi. In essence we have direct links to the brain of our Pi to connect sensors and components for use in our projects.

You will see the majority of Raspberry Pi tutorials using this reference and that is because it is the officially supported pin numbering scheme from the Raspberry Pi Foundation. So it is best practice to start using and learning the BCM pin numbering scheme as it will become second nature to you over time. Also note that BCM and GPIO pin numbering refer to the same scheme. So for example GPIO17 is the same as BCM17.

Certain GPIO pins also have alternate functions that allow them to interface with different kinds of devices that use the I2C, SPI or UART protocols. For example GPIO3 and GPIO 4 are also SDA and SCL I2C pins used to connect devices using the I2C protocol. To use these pins with these protocols we need to enable the interfaces using the Raspberry Pi Configuration application found in the Raspbian OS, Preferences menu.

3.2 USB Camera

Figure 3.2 USB camera

A webcam is a video camera that feeds or streams an image or video in real time to or through a computer network, such as the Internet. Webcams are typically small cameras that sit on a desk, attach to a user's monitor, or are built into the hardware. Webcams can be used during a video chat session involving two or more people, with conversations that include live audio andvideo. Webcam software enables users to record a video or stream the video on the Internet. As video streaming over the Internet requires much bandwidth, such streams usually use compressed formats. The maximum resolution of a webcam is also lower than most handheld video cameras, as higher resolutions would be reduced during transmission. The lower resolution enables webcams to be relatively inexpensive compared to most video cameras, but the effect is adequate for video chat sessions.

3.3 MLX 90614

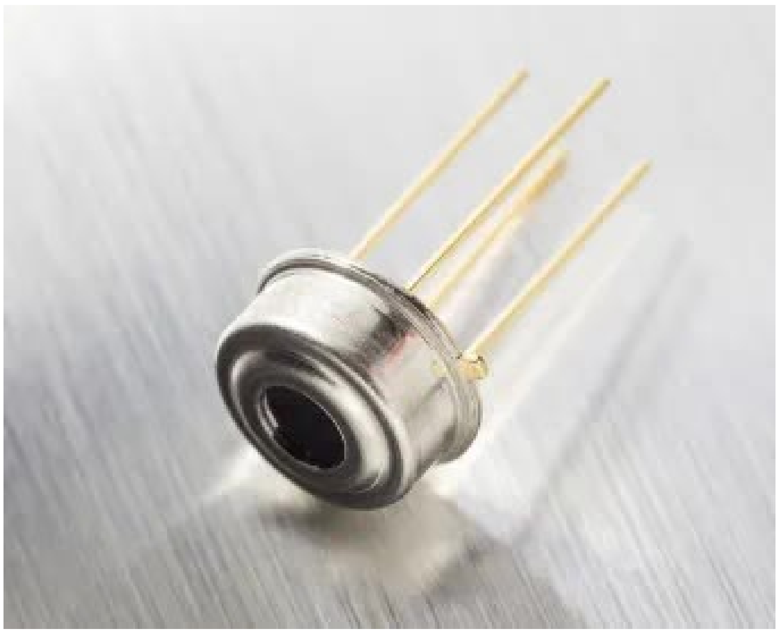


Figure 3.3 MLX 90614

The MLX90614 is an infrared thermometer for non-contact temperature measurements. Both the IR sensitive thermopile detector chip and the signal conditioning ASIC are integrated in the same TO-39 can. Integrated into the MLX90614 are a low noise amplifier, 17-bit ADC and powerful DSP unit thus achieving high accuracy and resolution of the thermometer ,The thermometer comes factory calibrated with a digital SMBus output giving full access to the measured temperature in the complete temperature range(s) with a resolution of 0.02°C. , The user can configure the digital output to be pulse width modulation (PWM). As a standard, the 10-bit PWM is configured to continuously transmit the measured temperature in range of -20 to 120°C, with an output resolution of 0.14°C.

3.4 POWER SUPPLY

Power is the backbone of any electronic system and the power supply is what feeds the system. Choosing the right supply can be the critical difference between a device working at optimum levels and one that may deliver inconsistent results. The power supply unit is the part of the hardware that is used to convert the power provided from the outlet into usable power to many parts inside an electrical device. Every energy supply must drive its load, which is connected to it. Depending on its design, a power supply unit may obtain energy from various types of energy sources, like electrical energy transmission systems, electromechanical systems such as generators and alternators, solar power converters, energy storage devices such as a battery and fuel cells, or other power supply. There are two types of power supplies existed, AC and DC power supply. Based on the electrical device’s electric specifications it may use AC power or DC power.

3.4.1 Power Supply Block Diagram

The Power supply circuit is used in various electrical & electronic devices. The power supply circuits are classified into different types based on the power they utilize for providing for circuits or devices. For instance, the microcontroller based circuits are generally the 5V DC regulated power supply (RPS) circuits, which can be designed with the help of different method for changing the power from 230V AC to 5V DC.The power supply block diagram, and the step by step conversion of 230V AC to 5V

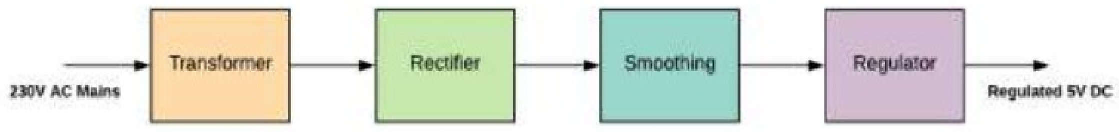


Figure 3.4.1 Power Supply Block Diagram

- A step-down transformer converts the 230V AC into 5v.
- The bridge rectifier is used to change AC to DC

- A capacitor is used to filter the AC ripples and gives to the voltage regulator.
- Finally voltage regulator regulates the voltage to 5V

3.5 5V Power Supply Using 7805

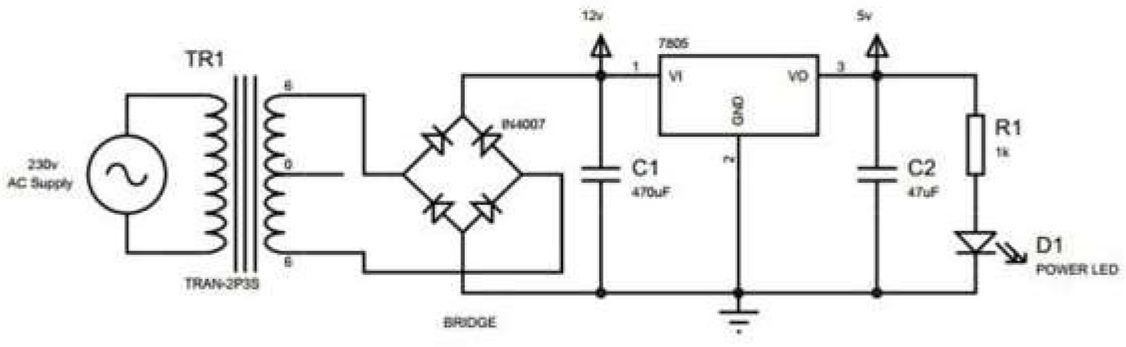


Figure 3.5 Full bridge rectifier Circuit

Components Used

- 6-0-6 Step Down Transformer
- 4 x 1N4007 Diode (Bridge Rectifier)
- 470uF Electrolytic Capacitor
- 7805 Voltage Regulator
- 47uF Electrolytic Capacitor 1k Resistor
- 3mm Red Led (Power Indication)

Step Down Transformer :

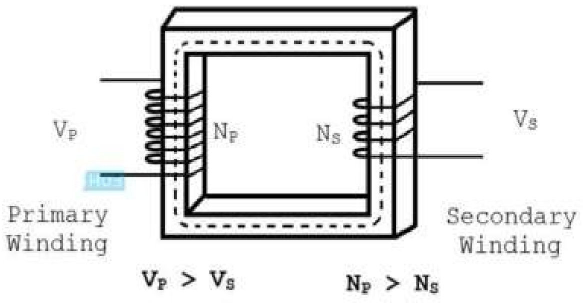


Figure 3.5.1 Step Down Transformer

A Transformer is a static apparatus, with no moving parts, which transforms electrical power from one circuit to another with changes in voltage and current and no change in frequency. There are two types of transformers classified by their function: Step up Transformer and Step down Transformer.

Principle of Working of Transformers :

An electrical transformer works on the principle of Mutual Induction, which states that a uniform change in current in a coil will induce an E.M.F in the other coil which is inductively coupled to the first coil.

In its basic form, a transformer consists of two coils with high mutual inductance that are electrically separated but have common magnetic circuit. The following image shows the basic construction of a Transformer.

Bridge Rectifier :

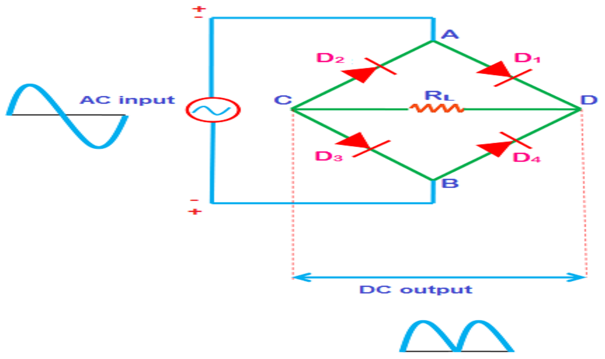


Figure 3.5.2 Bridge Rectifier

Rectifiers are mainly classified into three types: Half-wave rectifier, Center tapped full-wave rectifier and Bridge rectifier. All these three rectifiers have a common aim that is to convert Alternating Current (AC) into Direct Current (DC). Not all these three rectifiers efficiently convert the Alternating Current (AC) into Direct Current (DC), only the center tapped full-wave rectifier and bridge rectifier efficiently convert the Alternating Current (AC) into Direct Current (DC).

7805 Voltage Regulator :

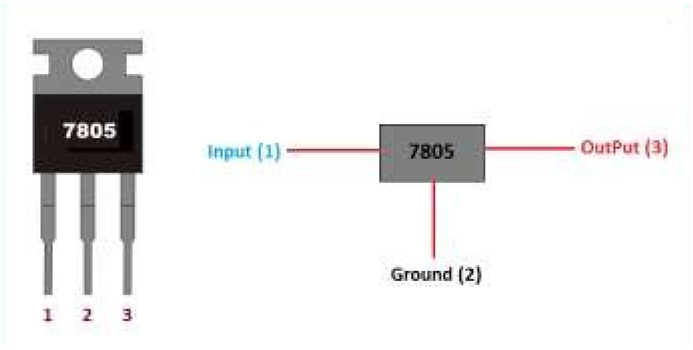


Figure 3.5.3 7805 IC Pinouts

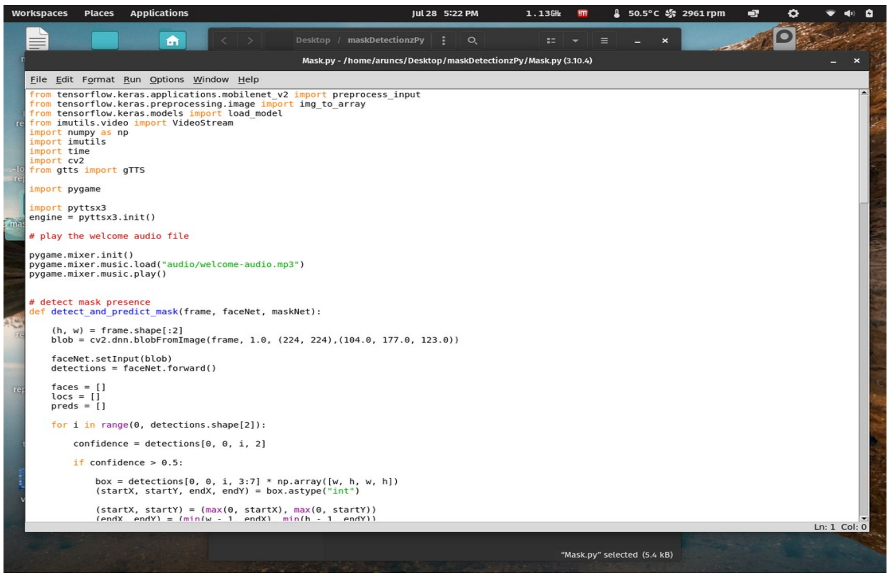
It is a positive voltage regulator used for providing constant output voltage over a wide range of input voltage. Voltage regulation is referred as the measure of voltage change between input and output. The IC 7805 does the same thing. It provides constant output voltage when a range of different voltage is applied at the input terminal. This component comes with three terminals called input, ground, and output. This is called positive voltage regulator because it generates positive voltage with respect to the ground terminal. Transistors and voltage regulator IC like 7805 work in a similar way with the intention of providing voltage regulation. 7805 is an IC used for voltage regulation and comes in TO-220 version. This component belongs to 78xx series where xx defines the output voltage it generates. Voltage fluctuation is a common practice during the execution of many electronic projects. This component overcomes and prevents this voltage fluctuation by providing a constant output voltage at the output terminal.The best part is that it doesn’t require any additional components to set output voltage. It is a compact IC that comes with a built-in protection circuit that avoids the circuits from too much heating, making it suitable for circuits drawing high current. The input voltage range applied to the input terminals of this IC varies from 7 V to 18 V (in some cases 7 to 35 V), resulting in the generation of constant output voltage around 5 V. You can see, there is a huge difference between input voltage and the output voltage that gets regulated. This difference is discharged as heat. The surge of heat generation can damage the device and affect the overall project performance. There are two ways to overcome this heat generation i.e. you can use a heat sink that is widely used for heat dissipation OR you can limit the input current 2 to 3 V above the regulated voltage at the output terminal. For example, you’ll get 5 V at the output terminal, so it is suitable to limit input voltage within 7 or 8 V.

CHAPTER 4

SOFTWARE DESCRIPTION

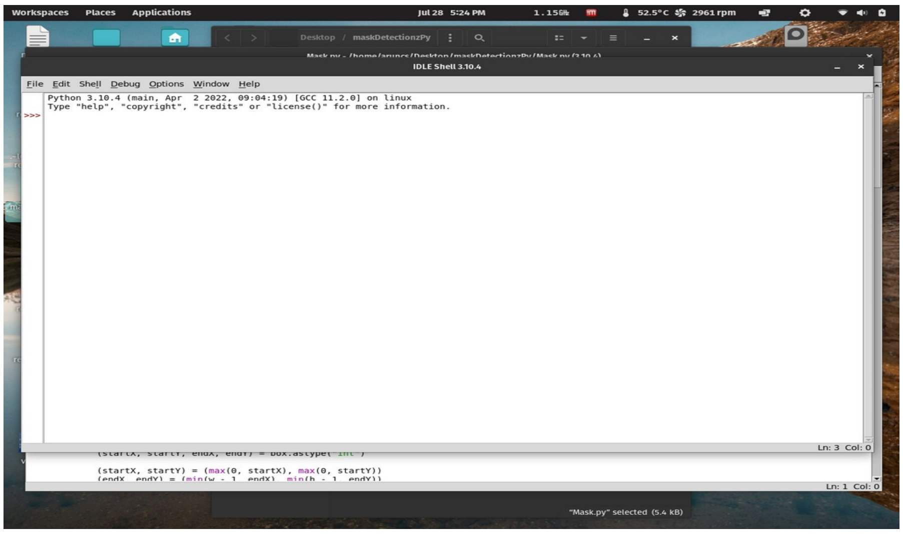
4.1 Python IDLE

An integrated development environment is a software application that provides comprehensive facilities to computer programmers for software development And Python IDE is software application dedicated IDE for python and IDLE is Python’s Integrated Development and Learning Environment. And Python IDLE is used in this project to run the python code


Workspaces
Places
Applications
Jul 28 5:22 PM
Desktop / maskDetectionzPy
Mask.py - /home/aruncs/Desktop/maskDetectionzPy/Mask.py (3.10.4)
File Edit Format Run Options Window Help
Ln: 1 Col: 0
"Mask.py" selected (5.4 kB)

```
from tensorflow.keras.applications.mobilenet_v2 import preprocess_input
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.models import load_model
from imutils.video import VideoStream
import numpy as np
import imutils
import time
import cv2
from gtts import gTTS

import pygame

import pyttsx3
engine = pyttsx3.init()

# play the welcome audio file

pygame.mixer.init()
pygame.mixer.music.load("audio/welcome-audio.mp3")
pygame.mixer.music.play()


# detect mask presence
def detect_and_predict_mask(frame, faceNet, maskNet):

    (h, w) = frame.shape[:2]
    blob = cv2.dnn.blobFromImage(frame, 1.0, (224, 224),(104.0, 177.0, 123.0))

    faceNet.setInput(blob)
    detections = faceNet.forward()

    faces = []
    locs = []
    preds = []

    for i in range(0, detections.shape[2]):

        confidence = detections[0, 0, i, 2]

        if confidence > 0.5:

            box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
            (startX, startY, endX, endY) = box.astype("int")

            (startX, startY) = (max(0, startX), max(0, startY))
            (endX, endY) = (min(w - 1, endX), min(h - 1, endY))
```

Figure 4.1(a) Python IDLE Main Window


Workspaces
Places
Applications
Jul 28 5:24 PM
Desktop / maskDetectionzPy
IDLE Shell 3.10.4
File Edit Shell Debug Options Window Help
Ln: 3 Col: 0
Ln: 1 Col: 0
"Mask.py" selected (5.4 kB)

```
Python 3.10.4 (main, Apr  2 2022, 09:04:19) [GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```

Figure 4.1(b) Python IDLE run window

4.2 VNC viewer

VNC viewer is used for local computers and mobile devices you want to control from A device such as a computer, tablet, or smart phone with VNC Viewer software installed can access and take control of a computer in another location.

It is a graphical desktop sharing system that allows a user to remotely control the desktop of a remote computer (running VNC Server) from your device, and it transmits the keyboard and mouse or touch events to VNC Server, so that once you are connected, you have control over the computer you've accessed. If you're using your mobile phone, for example, you would be able to use the computer you've remotely accessed as though you were sitting right in front of it. And VNC viewer is used to Access and control the Raspberry Pi .

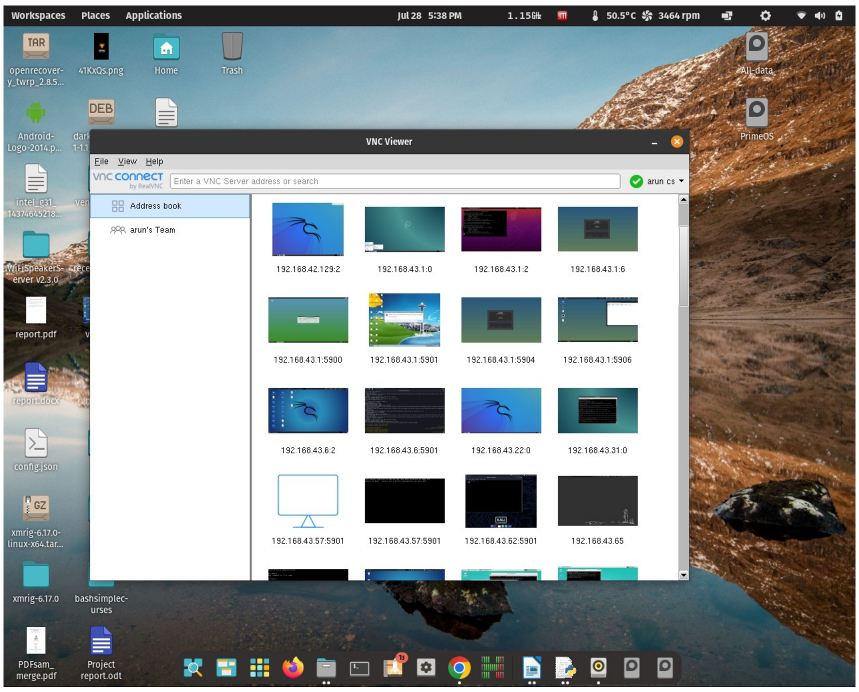


Figure 4.2 Real VNC Viewer Main Window

CHAPTER 5

CIRCUIT DIAGARM

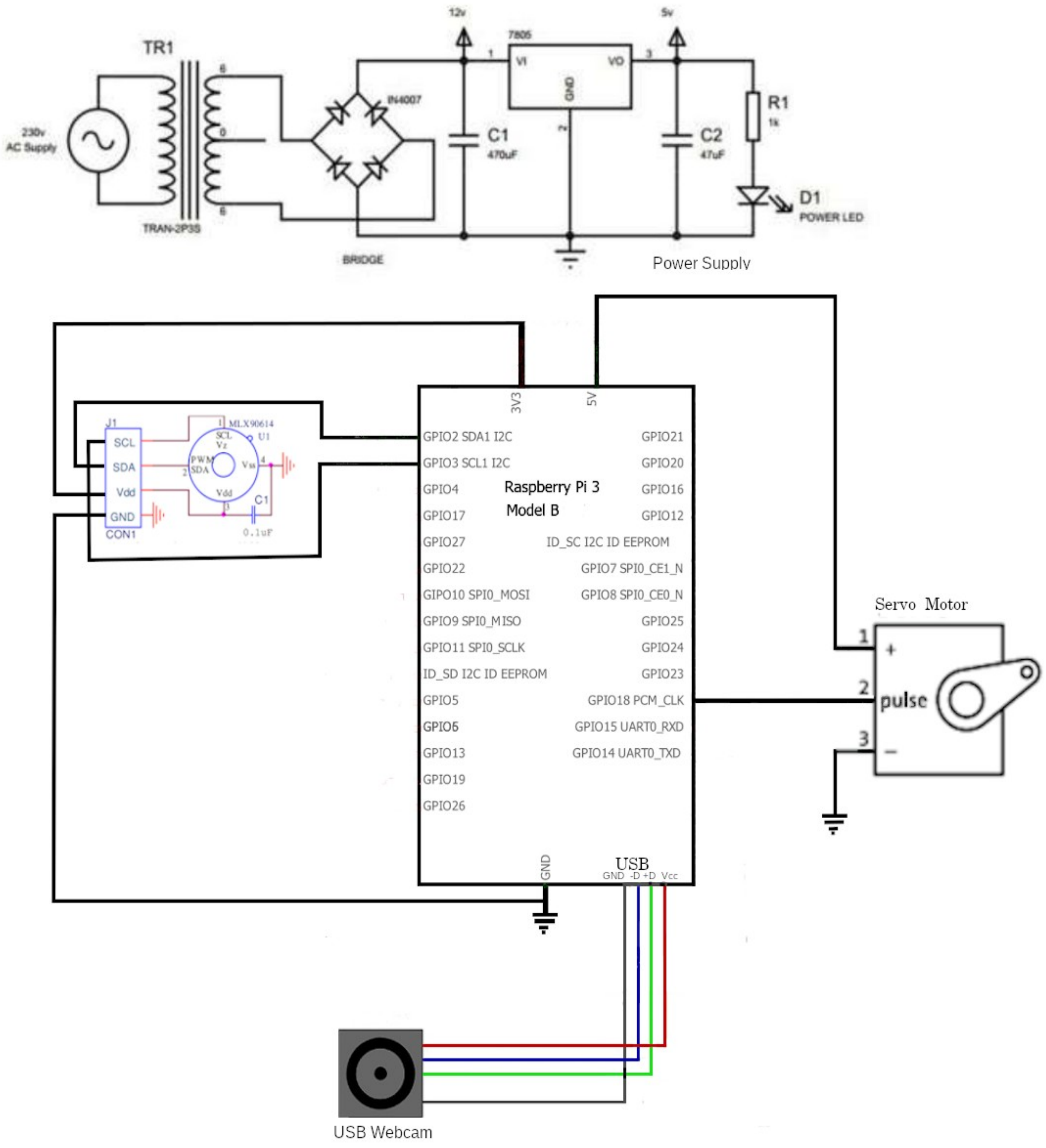


Figure 5.1 Circuit Diagram

CHAPTER 6

IMPLEMENATATION

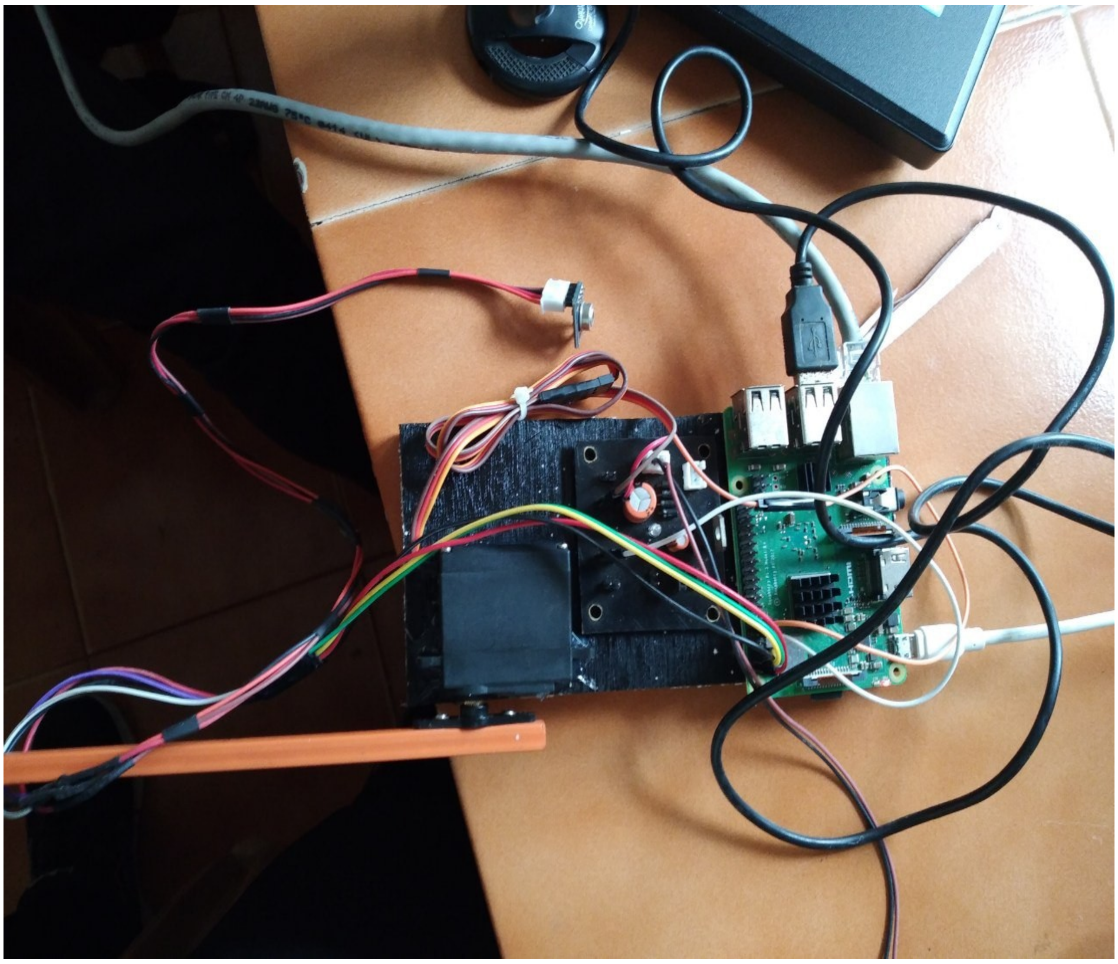


Figure 6.1 Circuit setup

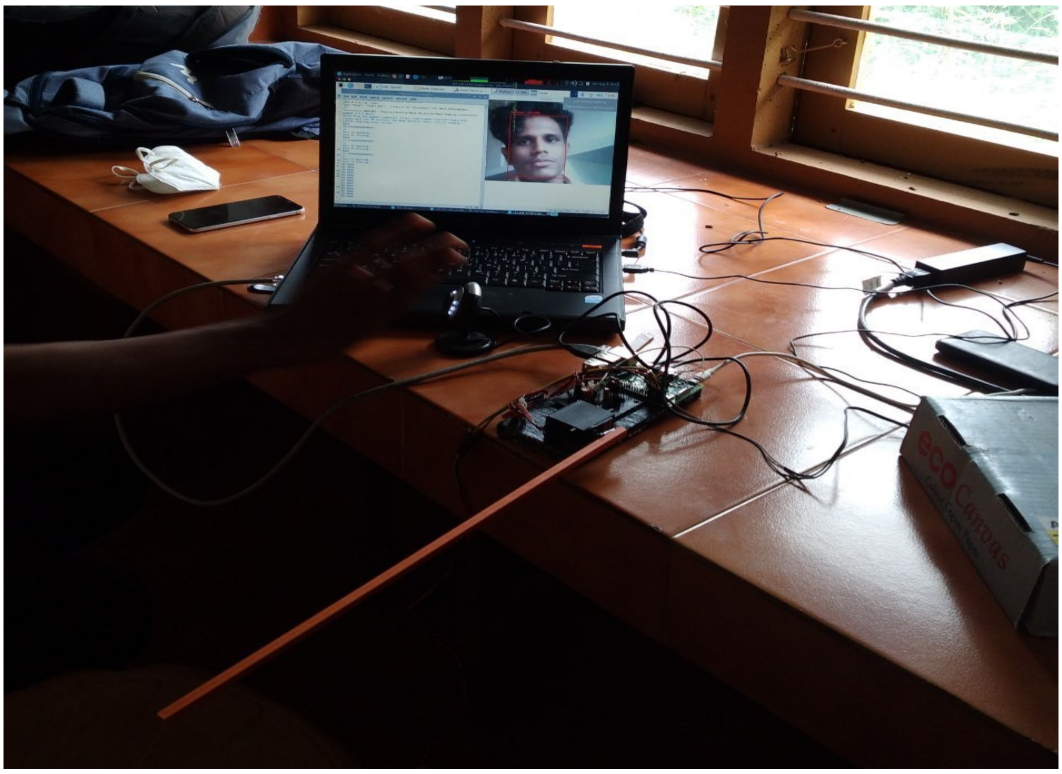


Figure 6.2 Test 1 Without mask

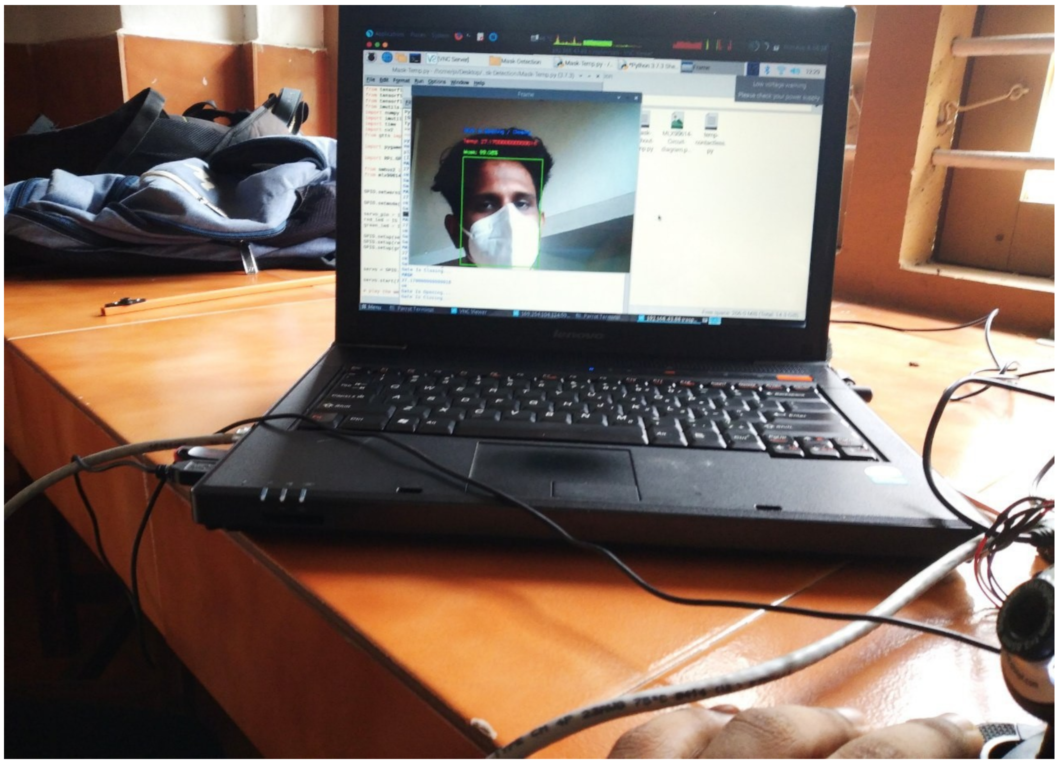
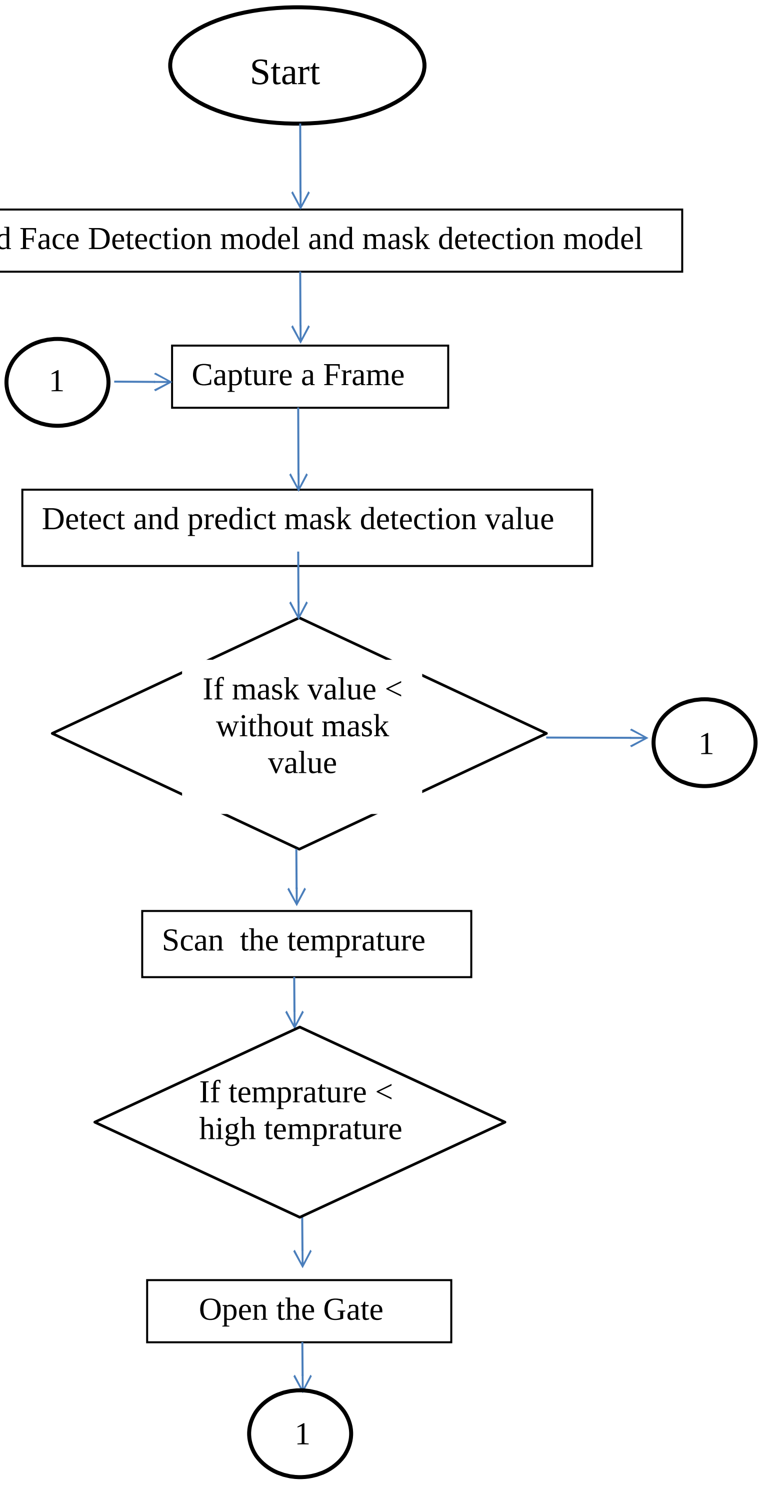


Figure 6.3 Test 2 With Mask

CHAPTER 7

FLOW CHART


Start
Load Face Detection model and mask detection model
1
Capture a Frame
Detect and predict mask detection value
If mask value < without mask value
1
Scan the temprature
If temprature < high temprature
Open the Gate
1

CHAPTER 8

COST ESTIMATION

Sl. No.	**ITEMS**	**QUANTITY**	**COST**
1	Raspberry Pie	1	2800
2	Servo Motor	1	220
3	USB Webcam	1	580
4	MLX 90614	1	1300
5	Power Supply 5v	2	180
6	IC 7805	1	10
	TOTAL COST		5270

CHAPTER 9

ADVANTAGES AND DISADVANTAGES

7.1 Advantages

An automated approach to detect mask and without mask persons There is no manual method Convey the importance of mask in the covid 19 pandemic.

7.2 Disadvantages

The system works with the help of camera, so that good lighting is needed for better result.

CHAPTER 10

CONCLUSION AND FUTURE SCOPE

10.1 Conclusion

The hardware to detect mask and without mask is implemented using raspberry pi and USB camera, The system works perfectly and classifies the persons properly. The mask detection software is developed using python program using open CV ,1000 images of mask images and 1000 images of without mask images are used to train the deep neural network module using caffe models

10.2 Future Scope

The system we develop for temprature and mark detection using raspberry pi can reduce rules violators from entering into public occasions

CHAPTER 11

REFERENCE

1. **https://github.com/opencv/opencv/wiki/TensorFlow-Object-Detection-API**

2. **https://en.m.wikipedia.org/wiki/Face_detection**

3. **https://www.tensorflow.org/**

4. **https://github.com/opencv/opencv/wiki/TensorFlow-Object-Detection-API**

CHAPTER 12

PROGRAM

```
from tensorflow.keras.applications.mobilenet_v2 import preprocess_input

from tensorflow.keras.preprocessing.image import img_to_array

from tensorflow.keras.models import load_model

from imutils.video import VideoStream

import numpy as np

import imutils

import time

import cv2

from gtts import gTTS

import pygame

import pyttsx3

engine = pyttsx3.init()

# play the welcome audio file

pygame.mixer.init()

pygame.mixer.music.load("audio/welcome-audio.mp3")

pygame.mixer.music.play()

# detect mask presence

def detect_and_predict_mask(frame, faceNet, maskNet):

    (h, w) = frame.shape[:2]

    blob = cv2.dnn.blobFromImage(frame, 1.0, (224, 224),(104.0, 177.0, 123.0))

    faceNet.setInput(blob)

    detections = faceNet.forward()
    faces = []
    locs = []
    preds = []
    for i in range(0, detections.shape[2]):
        confidence = detections[0, 0, i, 2]
        if confidence > 0.5:
            box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
            (startX, startY, endX, endY) = box.astype("int")
            (startX, startY) = (max(0, startX), max(0, startY))
            (endX, endY) = (min(w - 1, endX), min(h - 1, endY))
            face = frame[startY:endY, startX:endX]
            face = cv2.cvtColor(face, cv2.COLOR_BGR2RGB)
            face = cv2.resize(face, (224, 224))
            face = img_to_array(face)
            face = preprocess_input(face)
            faces.append(face)
            locs.append((startX, startY, endX, endY))
    if len(faces) > 0:
        faces = np.array(faces, dtype="float32")
        preds = maskNet.predict(faces, batch_size=32)
    return (locs, preds)
print("please wait for 20 seconds, the mask detector model file is loading ...")


prototxtPath = "face_detector/deploy.prototxt"
```

```
weightsPath = "face_detector/res10_300x300_ssd_iter_140000.caffemodel"
faceNet = cv2.dnn.readNet(prototxtPath, weightsPath)
maskNet = load_model("mask_detector.model")
print("[INFO] starting video stream...")
vs = VideoStream(src=0).start()
#vs = cv2.VideoCapture(0)
####################################################################
for i in range(0,10):
    frame = vs.read()
    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1) & 0xFF
    time.sleep(0.1)
    ########################################################################
while True:
    frame = vs.read()
    frame_copy = frame.copy()
    #frame = imutils.resize(frame, width=400)
    (locs, preds) = detect_and_predict_mask(frame, faceNet, maskNet)
    for (box, pred) in zip(locs, preds):
        (startX, startY, endX, endY) = box
        (mask, withoutMask) = pred
        print(pred)
        if(mask < withoutMask):
            label = "No Mask"
            color = (0, 0, 255)
            label = "{}: {:.2f}%".format(label, max(mask, withoutMask) * 100)
                cv2.putText(frame_copy, label, (startX, startY - 10), cv2.FONT_HERSHEY_SIMPLEX,
0.45, color, 2)
            cv2.rectangle(frame_copy, (startX, startY), (endX, endY), color, 2)
            cv2.imshow("Frame", frame_copy)
            key = cv2.waitKey(1) & 0xFF
            engine.say("Entry is restricted for you. Since you are not wearing mask")
            engine.runAndWait()
        else:
            label = "Mask"
            color = (0, 255, 0)
            color1 = (0, 0, 255)
            color2 = (255, 0, 0)
            label = "{}: {:.2f}%".format(label, max(mask, withoutMask) * 100)
                cv2.putText(frame_copy, label, (startX, startY - 10), cv2.FONT_HERSHEY_SIMPLEX,
0.45, color, 2)
            cv2.rectangle(frame_copy, (startX, startY), (endX, endY), color, 2)
            cv2.imshow("Frame", frame_copy)
            key = cv2.waitKey(1) & 0xFF
            engine.say("Welcome, Wash your hands berfore entering the building")
            engine.runAndWait()


           label = "{}: {:.2f}%".format(label, max(mask, withoutMask) * 100)
```

```
            cv2.putText(frame_copy, label, (startX, startY - 10), cv2.FONT_HERSHEY_SIMPLEX,
0.45, color, 2)
            cv2.rectangle(frame_copy, (startX, startY), (endX, endY), color, 2)
            cv2.imshow("Frame", frame_copy)
            key = cv2.waitKey(1) & 0xFF
        label = "Mask" if mask > withoutMask else "No Mask"
        color = (0, 255, 0) if label == "Mask" else (0, 0, 255)
        label = "{}: {:.2f}%".format(label, max(mask, withoutMask) * 100)
         cv2.putText(frame_copy, label, (startX, startY - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.45,
color, 2)
        cv2.rectangle(frame_copy, (startX, startY), (endX, endY), color, 2)
    cv2.imshow("Frame", frame_copy)
    key = cv2.waitKey(1) & 0xFF
    if key == ord("q"):
        break
cv2.destroyAllWindows()
vs.stop()
```