SMART WEARABLE LEARNING DEVICE

FOR VISUALLY DISABLED PERSONS

INTERNAL GUIDE

Mrs. A. VISHNU PRIYA

BY

MUKESH BADIGINENI

OBJECTIVE:-

- The designed Smart wearable device can identify objects in the outside environment and give output as an audio format.
- The designed Smart wearable device provides Voice Command Assistance
- This help visually disabled person aid them in the primary learning task of identifying objects without the supervision of the third party.

BASE PAPER:-

- Smart Hat: Design and Implementation of a Wearable Learning Device for Kids using AI and IoTs Techniques by Hsiung Chang, Chih-Yung Chang, Bhargavi Dande.

COMPARISON

Previous Project	Present Project
• Captures Only Image.	• Captures image, voice command and voice assistance
• Complex Hardware(Arduino, raspberryPI,wifi,Bluetooth module).	• Simple Hardware (RaspberryPI)
• More Power Consumption.	• Less Power Consumption.
• Issue with Arduino and raspberry communication.	• No, need of communication.
• Contains single functionality button.	• Consists 3 different buttons for 3 different functionalities.
• Hardware system is costly.	• Hardware system is economic.

Requirements:-

❖Hardware:-

I. Raspberry pi 3
II. Speaker
III. Microphone
IV. PI camera
V. Breadboard with Switches

❖Software:-

I. Python 3
II. Tensor Flow
III. Anaconda Navigator

PHASE'S:-

1. IOT PHASE

a) Interfacing Peripherals.

b) Capturing Image.

c) Speech Recording.

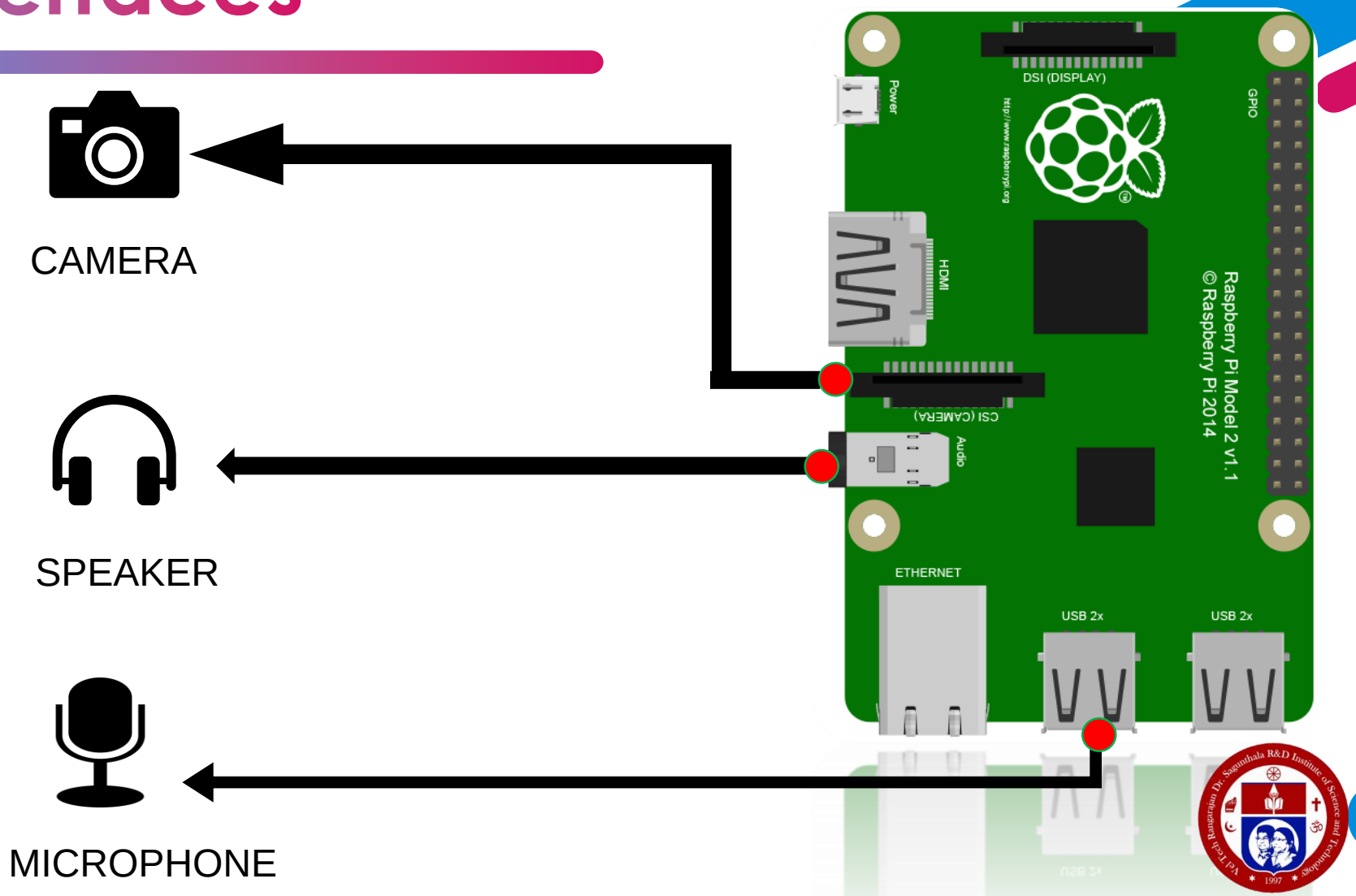
d) Audio Output.

2. AI PHASE

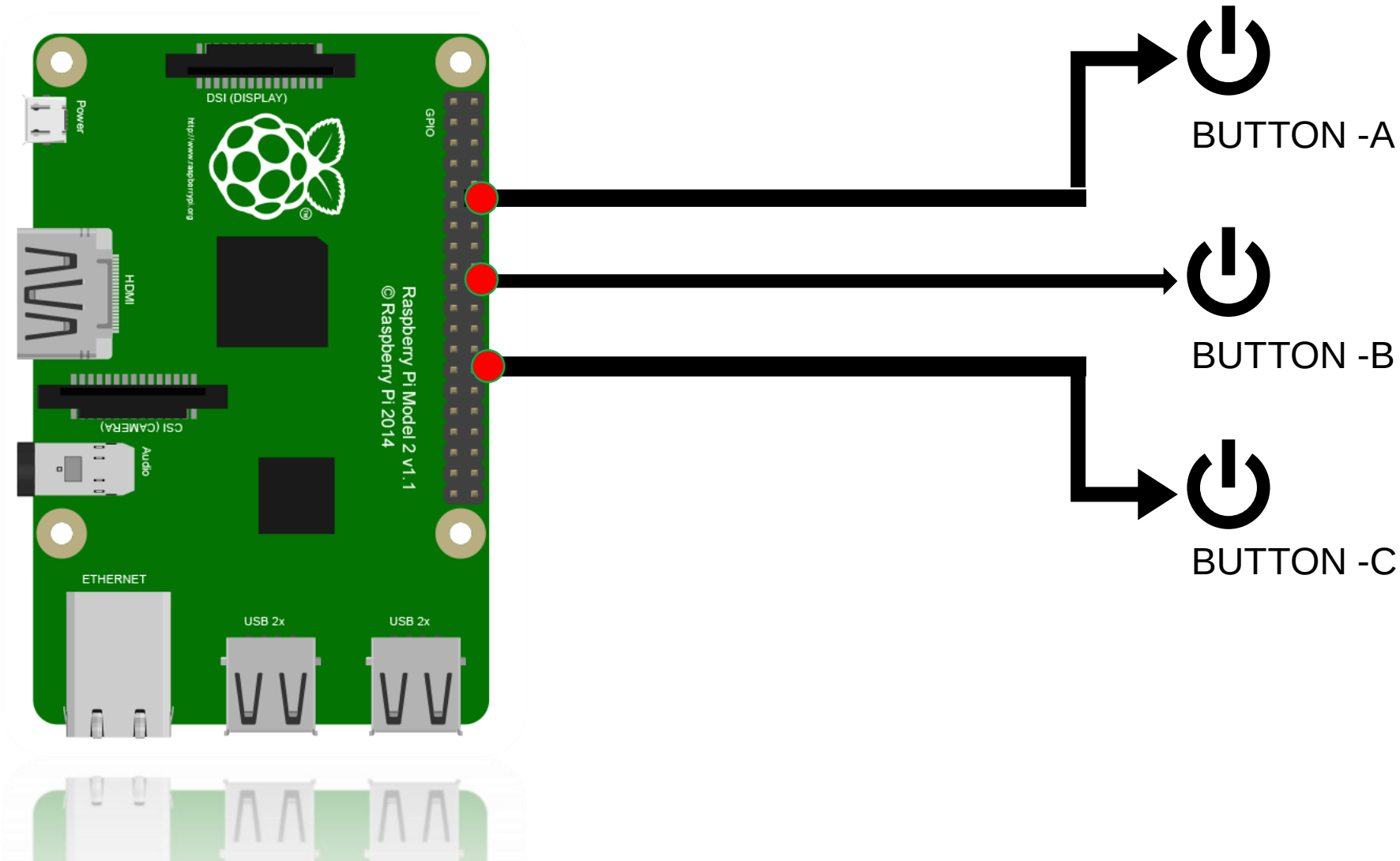
a) Image Recognition.

b) Command Assistance.

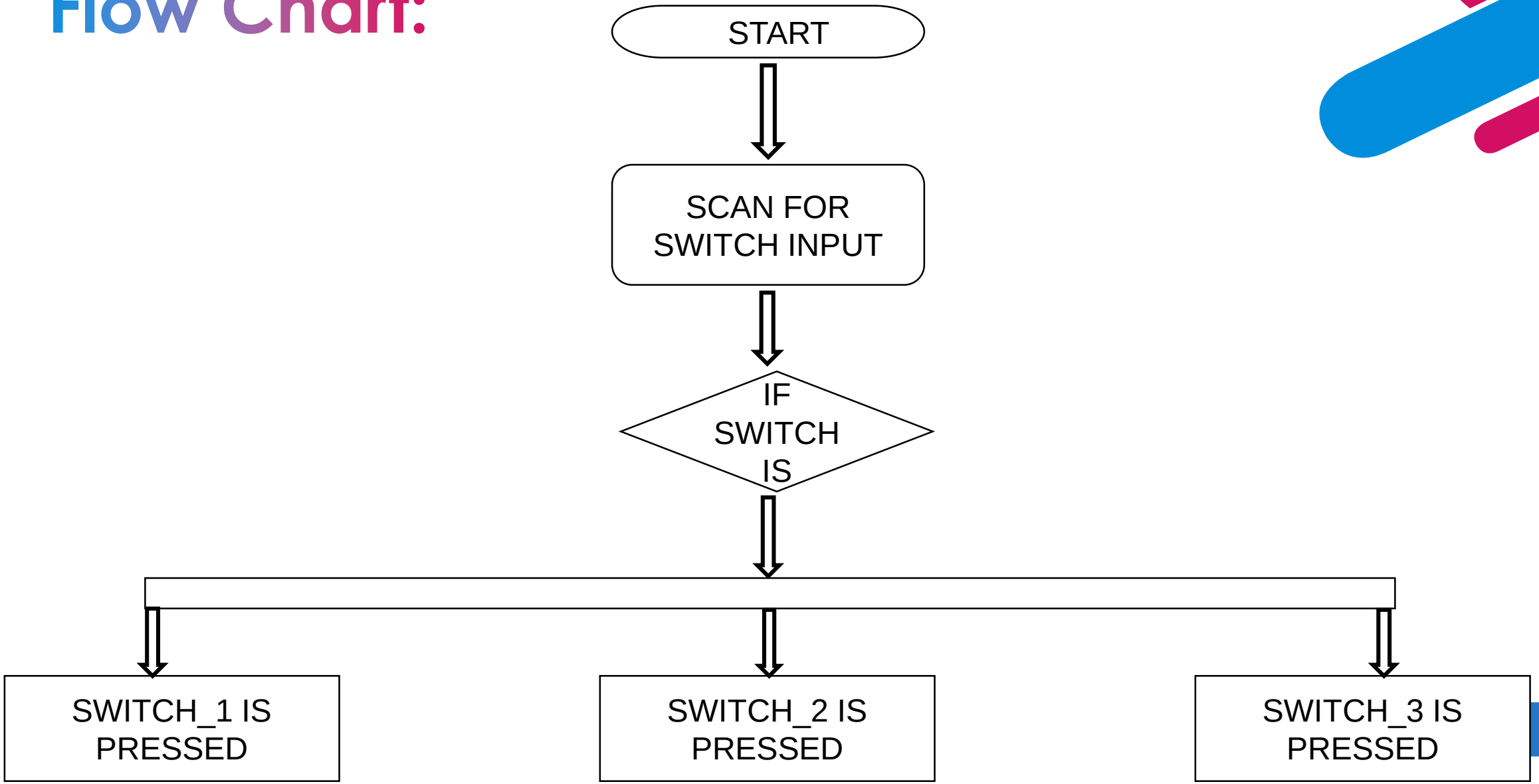
Out Put interfaces



Input Interfaces



Flow Chart:


START
SCAN FOR SWITCH INPUT
IF SWITCH IS
SWITCH_1 IS PRESSED
SWITCH_2 IS PRESSED
SWITCH_3 IS PRESSED

CAMERA == TRUE

DELAY (500 ms)

CAMERA == FALSE

MICROPHONE == TRUE

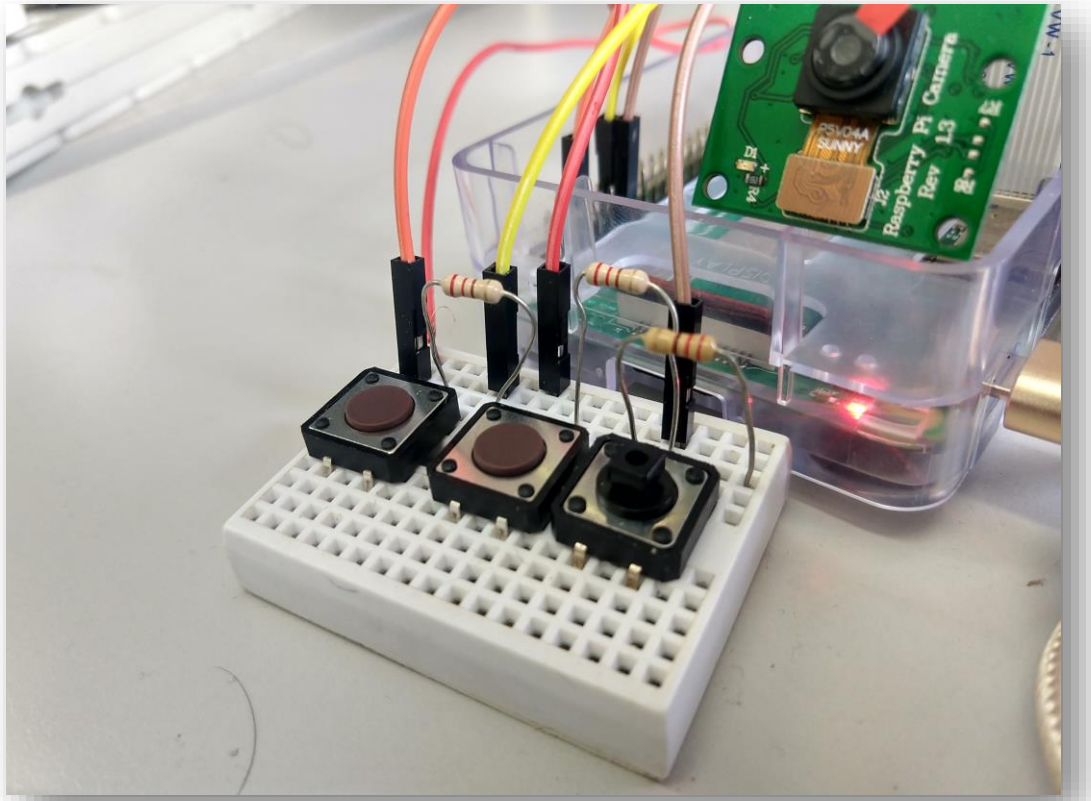
RECORDS 8 SECONDS

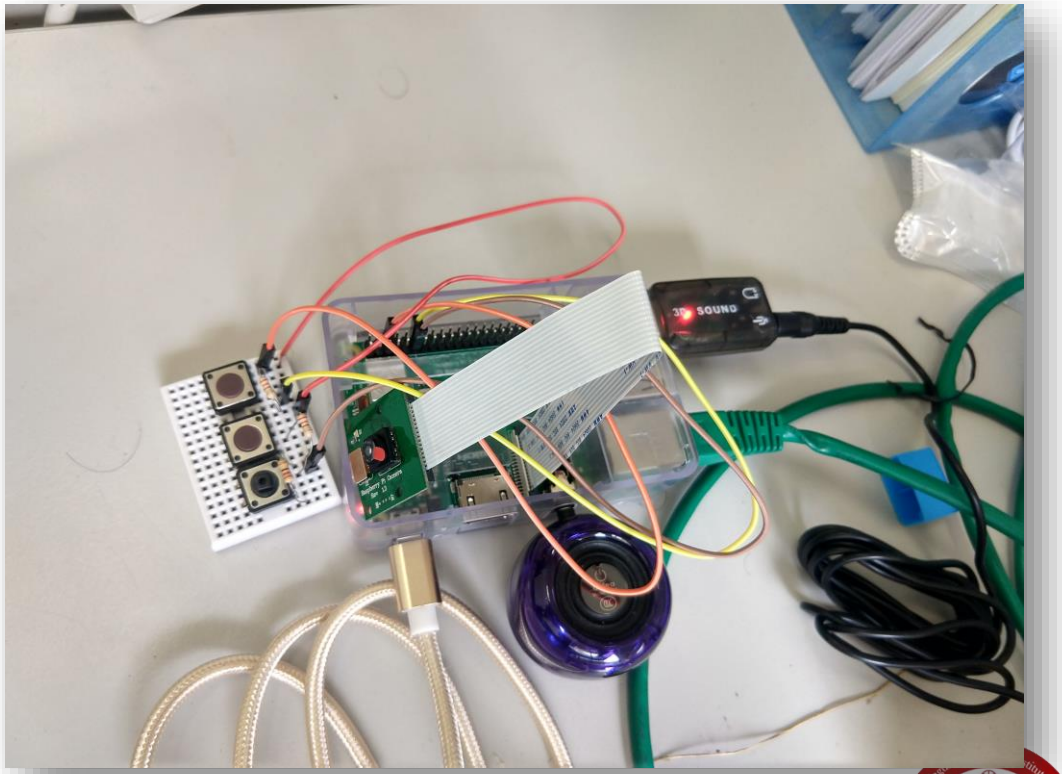
MICROPHONE == FALSE

GOOGLE API== TRUE

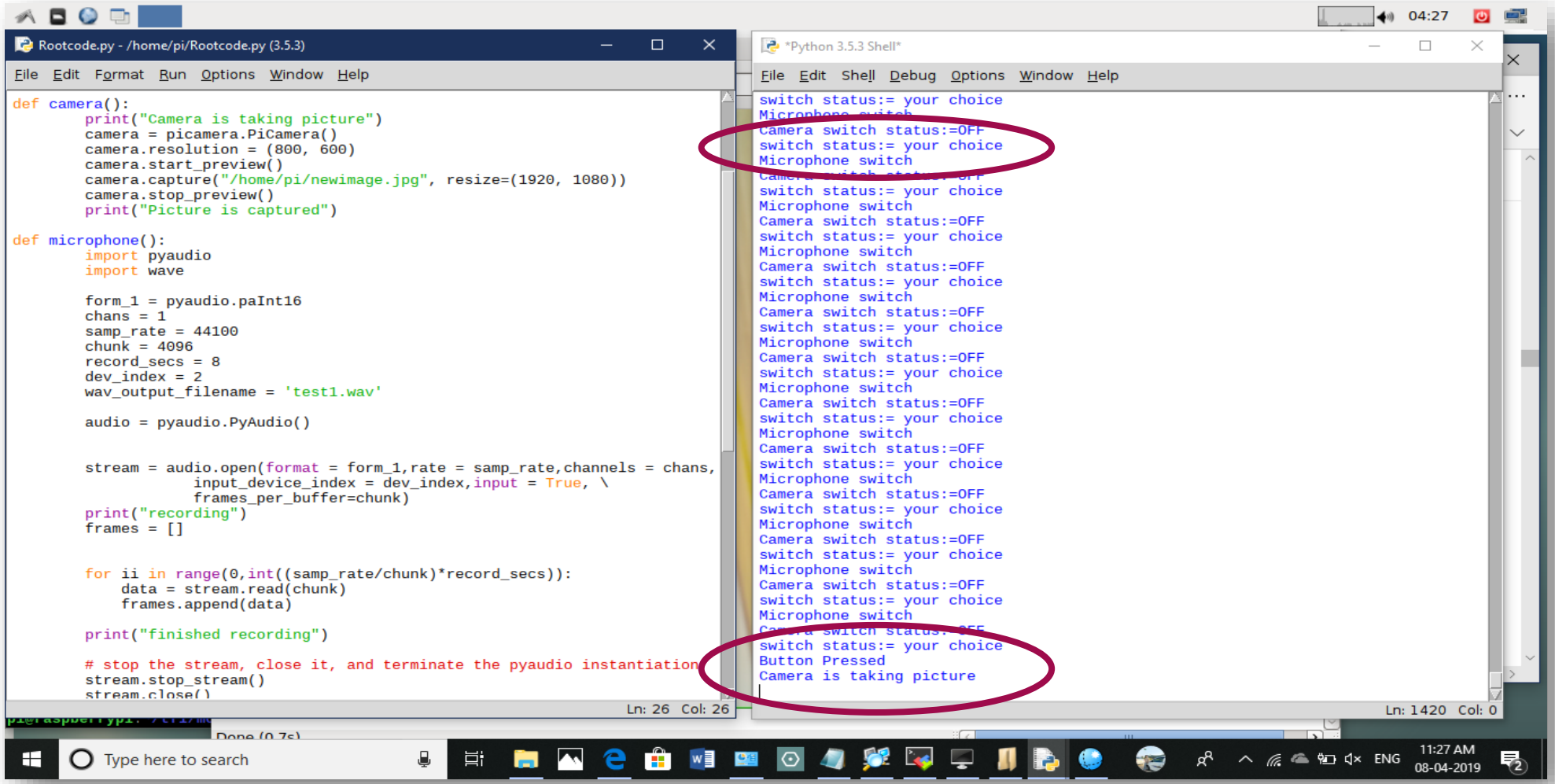
SPEAKER == ENABLE

Hardware Setup





WORKING MODEL



WORKING MODEL

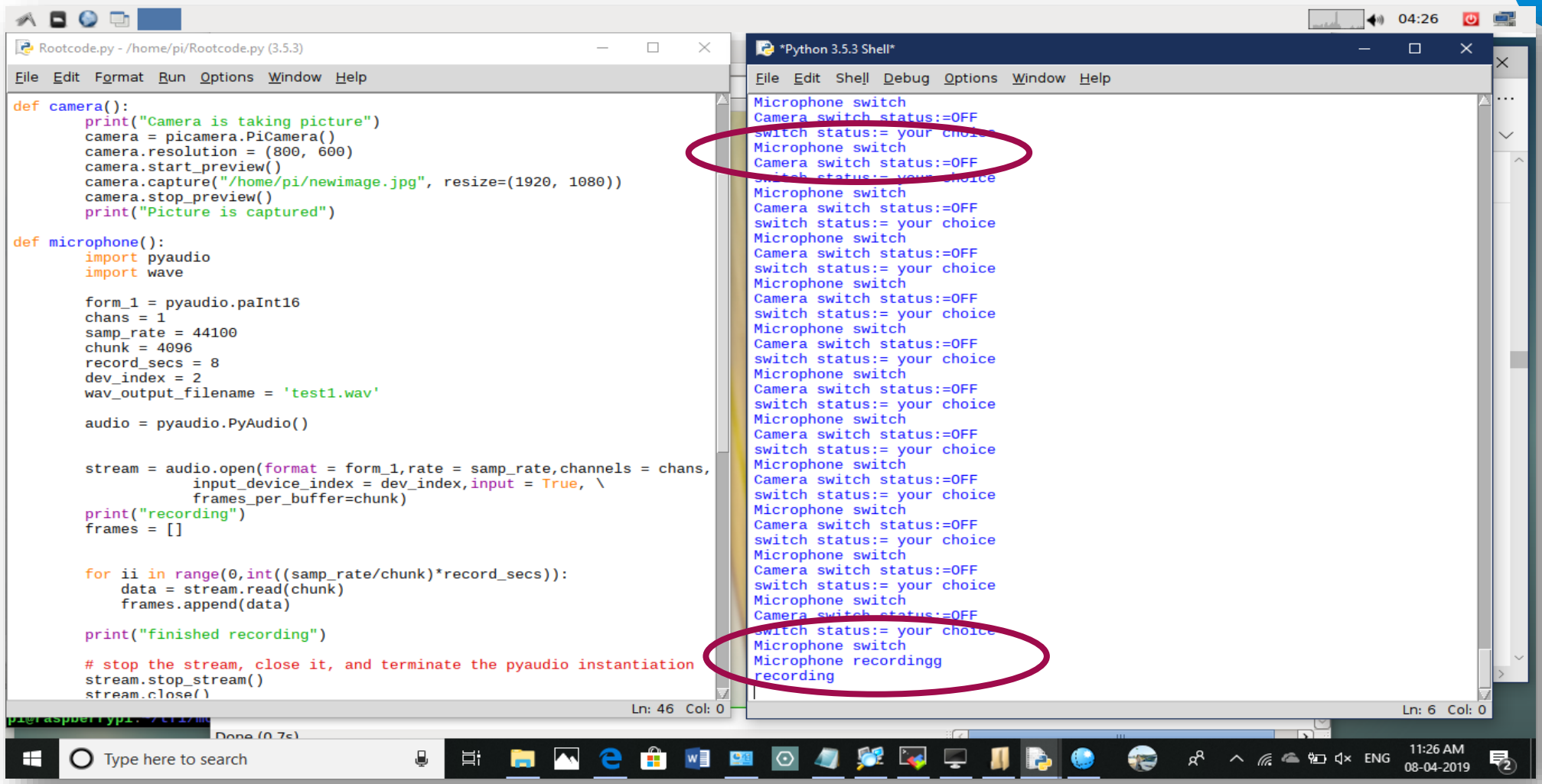
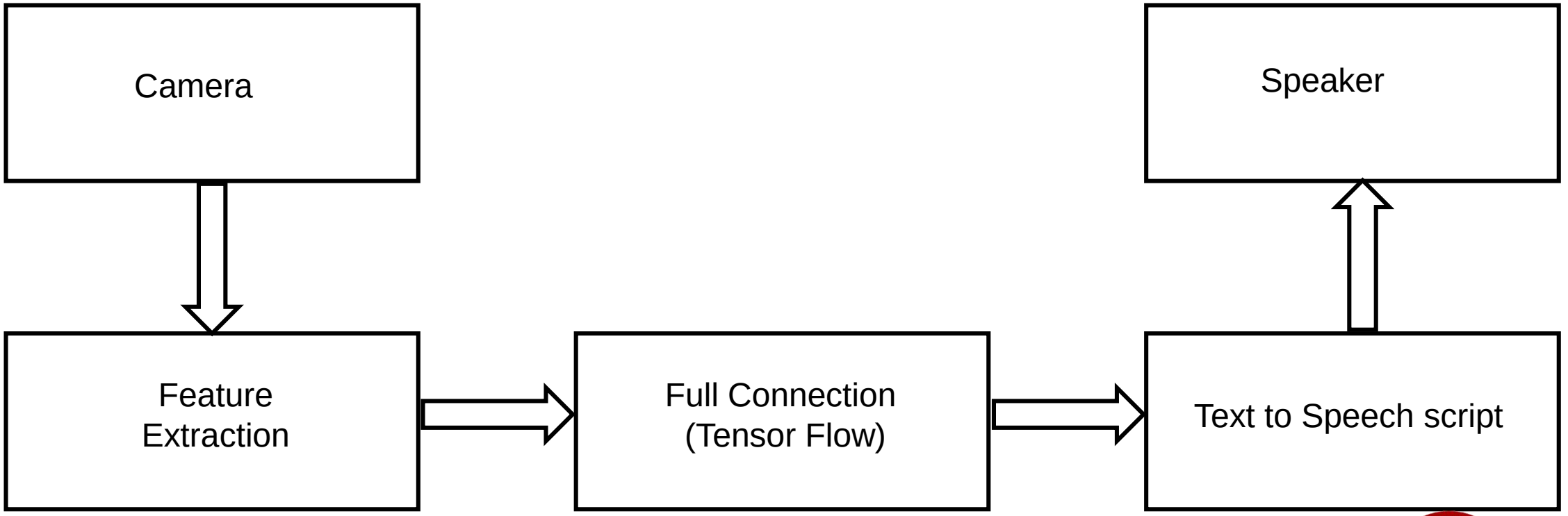


Image Recognition



Building The CNN

Building the CNN consists of three parts –

- Convolution
- Polling
- Flattening

Convolution:-

- Consider a 5 x 5 image whose pixel values are only 0 and 1.

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

- consider tensor flow filter 3 x 3 matrix as shown below:

1	0	1
0	1	0
1	0	1

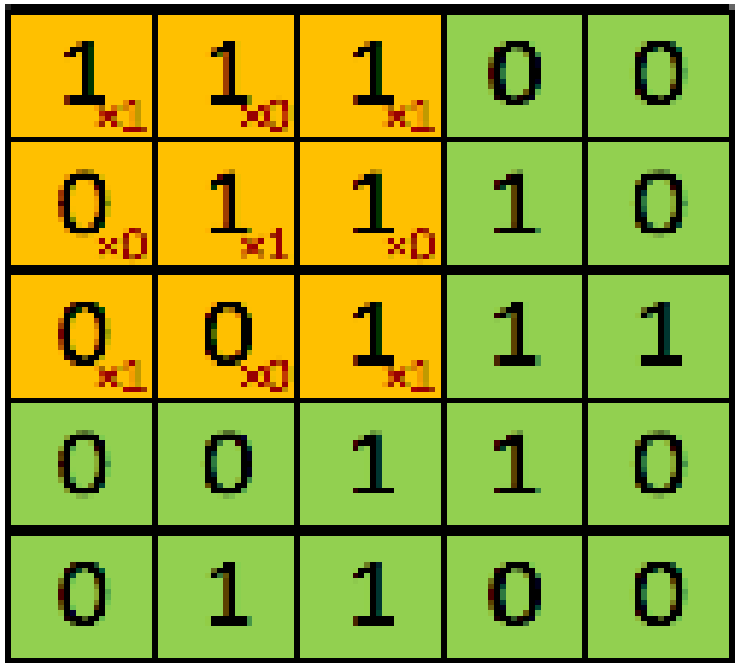
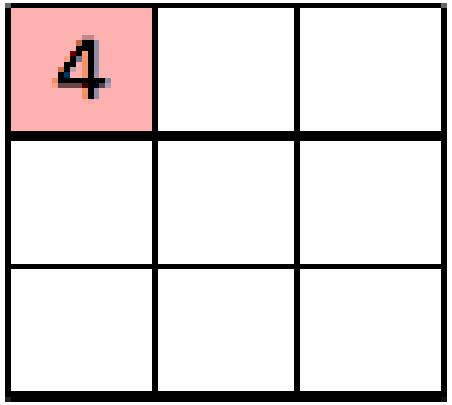


Image with Filter



Convolved feature map

Activation Function:-

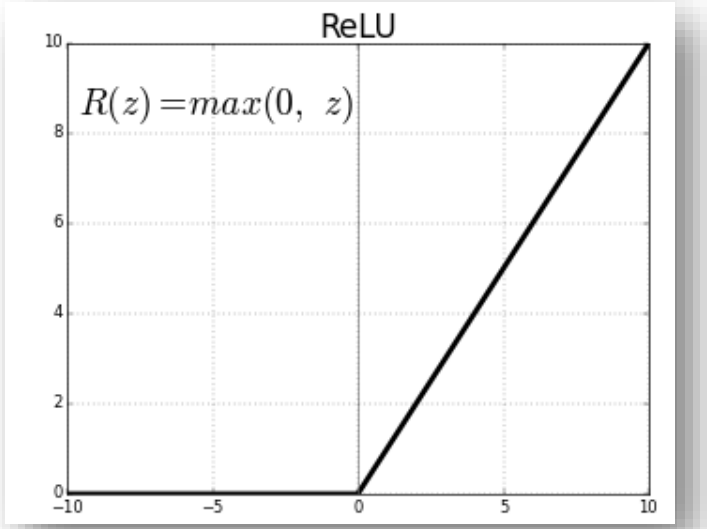
- Activation function of a node defines the output of that node, or "neuron," given an input or set of inputs
- ReLU (Rectified Linear Unit) activation Function has been used in the CNN.

- If input > 0:

    return input

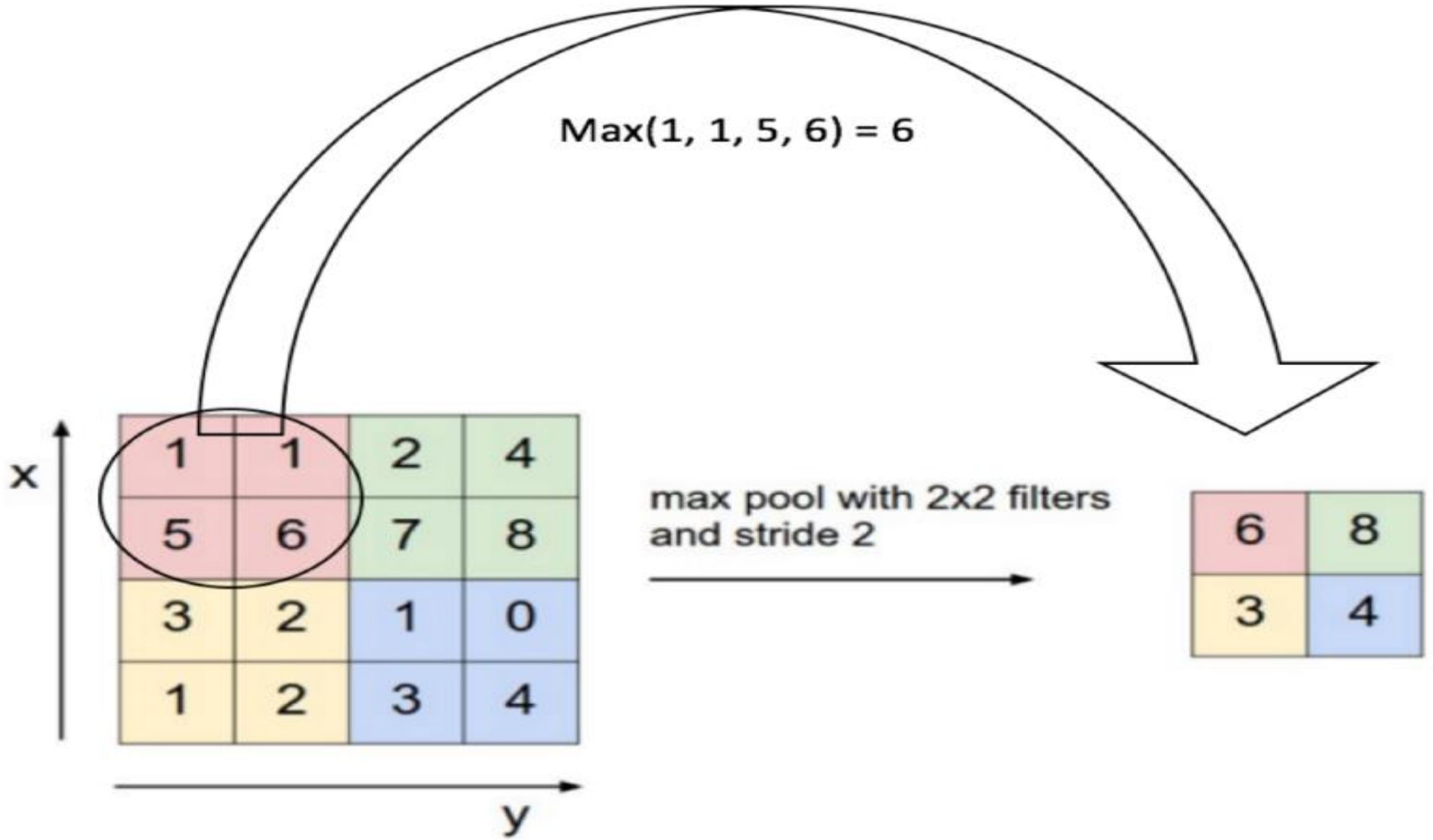
  else:

    return 0



Pooling:-

- Pooling reduces the dimensionality of each feature map but retains the most important information.
- But in this project we have been used Max Pooling Technique.
- Tensor flow graph having better results with max pooling technique.

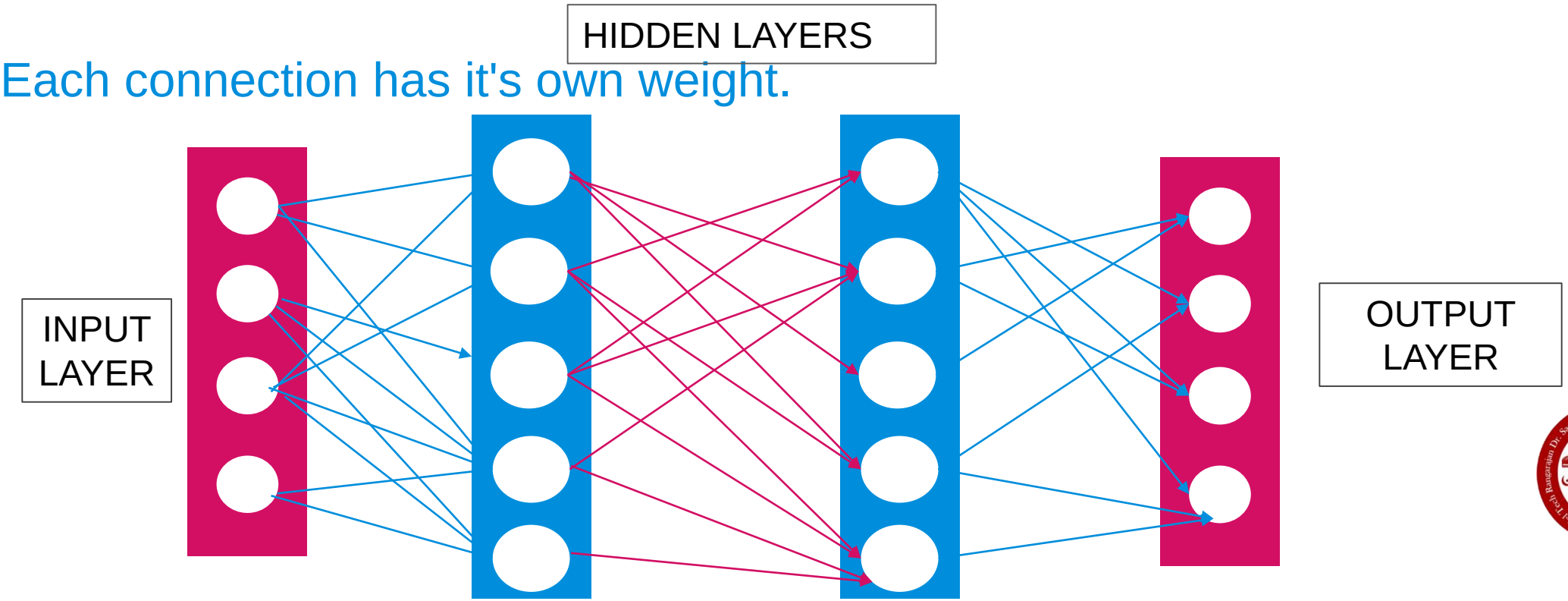

Max(1, 1, 5, 6) = 6
x
y
1 1 2 4
5 6 7 8
3 2 1 0
1 2 3 4
max pool with 2x2 filters
and stride 2
6 8
3 4
Rectified Feature Map

Flattening:-

- Here the matrix is converted into a linear array
- so that to input it into the nodes of our neural network.

Fully Connected:-

- Fully connected layers are those where each nodes is connected to every previous and every next node.
- Each connection has it's own weight.



Softmax Function:-

- The weight calculation is carried out by softmax function.
- It normalizes it into a probability distribution consisting of K probabilities

$$S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_i}}$$

CNN Algorithm:-

```
model = Sequential()

model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(100, 100, 3)))

model.add(Conv2D(32, (3, 3), activation='relu'))

model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(64, (3, 3), activation='relu'))

 model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(64, (3, 3), activation='relu'))

 model.add(Conv2D(128, (3, 3), activation='relu'))

model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Dropout(0.25))
```

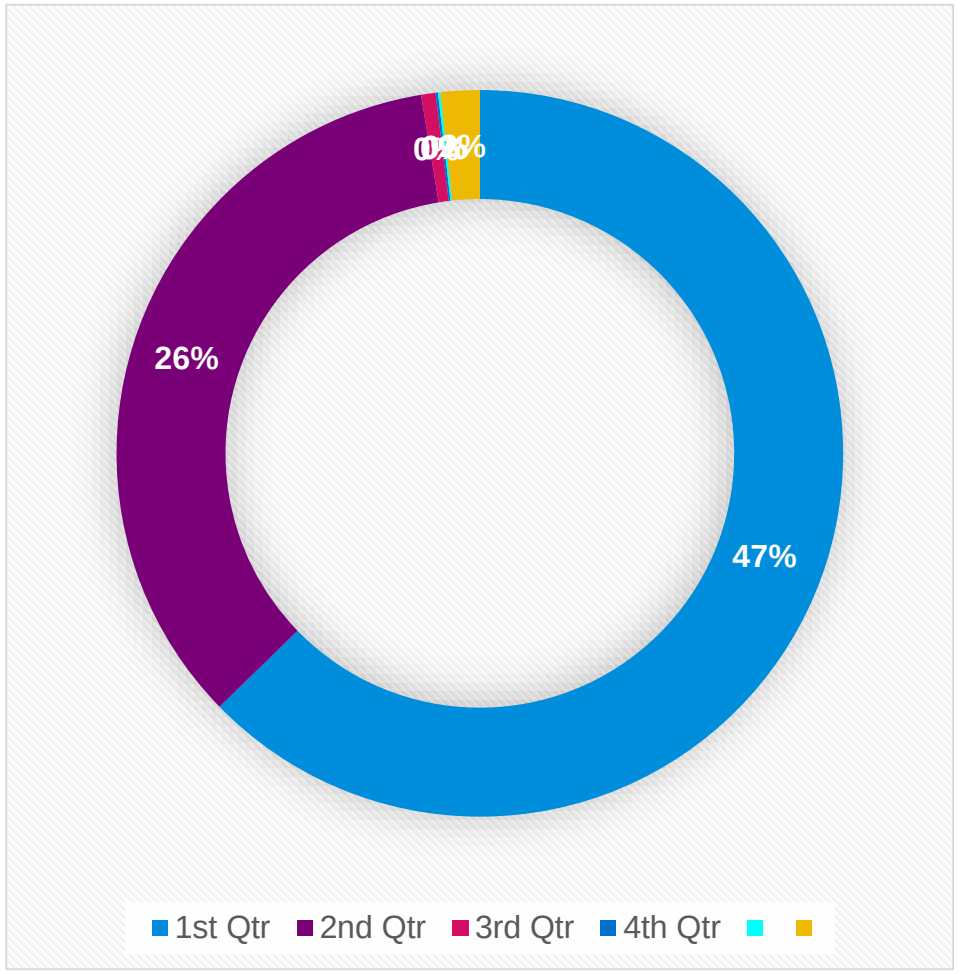
IMAGE RECOGNITION

```
        print( Picture is captured )

def microphone():
        import pyaudio
        import wave

        form_1 = pyaudio.paInt16
        chans = 1
        samp_rate = 44100
        chunk = 4096
        record_secs = 8
        dev_index = 2
        wav_output_filename = 'test1.wav'

        audio = pyaudio.PyAudio()


        stream = audio.open(format = form_1,rate = samp_rate,channels = chans, \
                    input_device_index = dev_index,input = True, \
                    frames_per_buffer=chunk)
        print("recording")
        frames = []


        for ii in range(0,int((samp_rate/chunk)*record_secs)):
            data = stream.read(chunk)
            frames.append(data)

        print("finished recording")

        # stop the stream, close it, and terminate the pyaudio instantiation
        stream.stop_stream()
        stream.close()
        audio.terminate()

        # save the audio frames as .wav file
        wavefile = wave.open(wav_output_filename,'wb')
        wavefile.setnchannels(chans)
        wavefile.setsampwidth(audio.get_sample_size(form_1))
        wavefile.setframerate(samp_rate)
```

```
switch status:= your choice
Microphone switch
Camera switch status:=OFF
switch status:= your choice
Microphone switch
Camera switch status:=OFF
switch status:= your choice
Microphone switch
Camera switch status:=OFF
switch status:= your choice
Microphone switch
Camera switch status:=OFF
switch status:= your choice
Microphone switch
Camera switch status:=OFF
switch status:= your choice
Microphone switch
Camera switch status:=OFF
switch status:= your choice
Microphone switch
Camera switch status:=OFF
switch status:= your choice
Microphone switch
Camera switch status:=OFF
switch status:= your choice
Button Pressed
Camera is taking picture
Picture is captured
WARNING:tensorflow:From /home/pi/Rootcode.py:196: FastGFile.__init__ (from tenso
rflow.python.platform.gfile) is deprecated and will be removed in a future versi
on.
Instructions for updating:
Use tf.gfile.GFile.
water jug
water jug (score = 0.47978)
water bottle (score = 0.26398)
pill bottle (score = 0.04873)
soap dispenser (score = 0.00872)
nipple (score = 0.00871)
>>>
```

PREDICTIONS

```
Camera switch status:=OFF
switch status:= your choice
Microphone switch
Camera switch status:=OFF
switch status:= your choice
Microphone switch
Camera switch status:=OFF
switch status:= your choice
Button Pressed
Camera is taking picture
Picture is captured
WARNING:tensorflow:From /home/pi/Rootcode.py:19
rflow.python.platform.gfile) is deprecated and
on.
Instructions for updating:
Use tf.gfile.GFile.
water jug
water jug (score = 0.47978)
water bottle (score = 0.26398)
pill bottle (score = 0.04873)
soap dispenser (score = 0.00872)
nipple (score = 0.00871)
>>>
```



Prediction

47% Water Mug

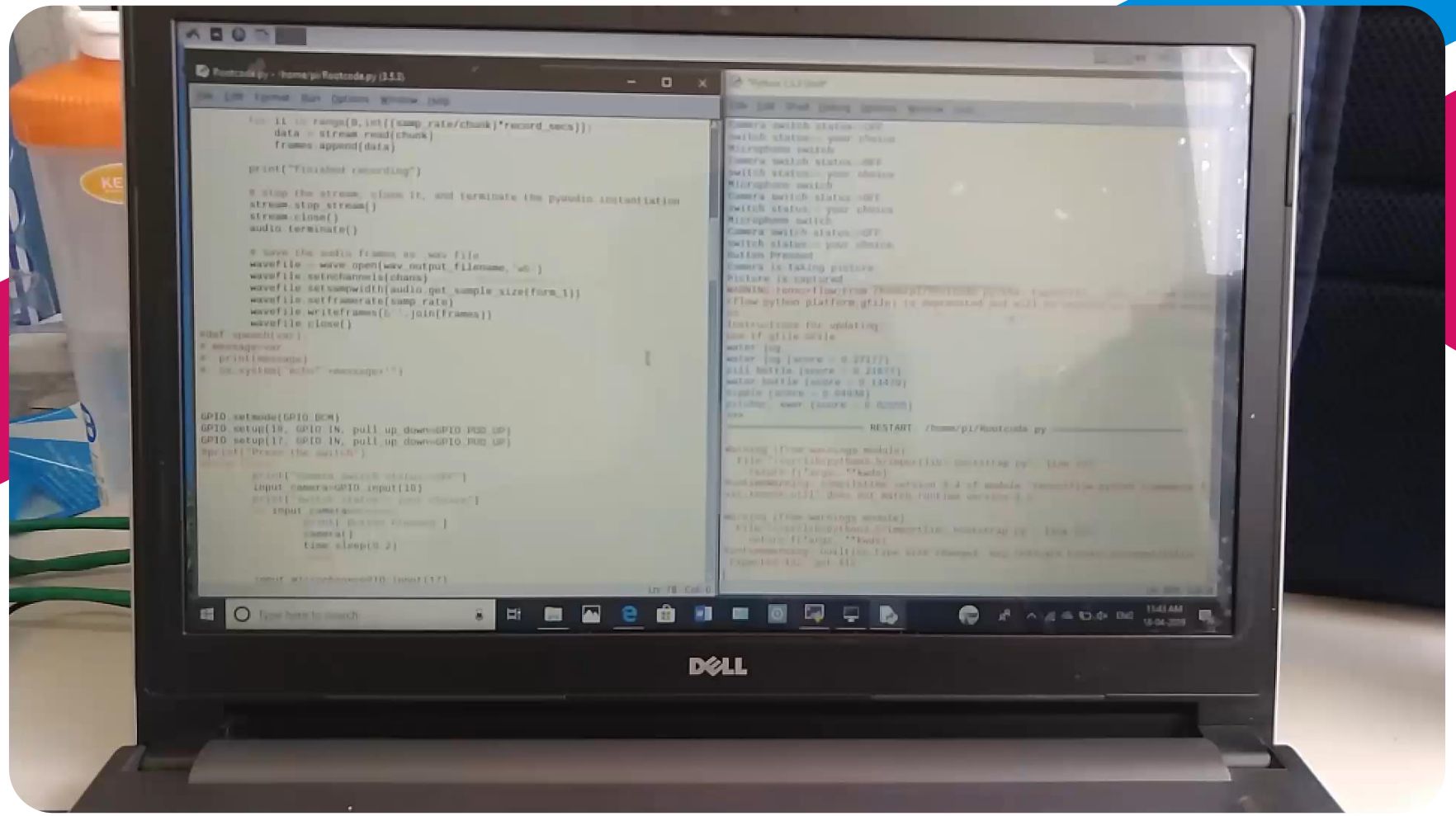
26% Water Bottle

0.48% Pill bottle

0.08% Soap Dispenser

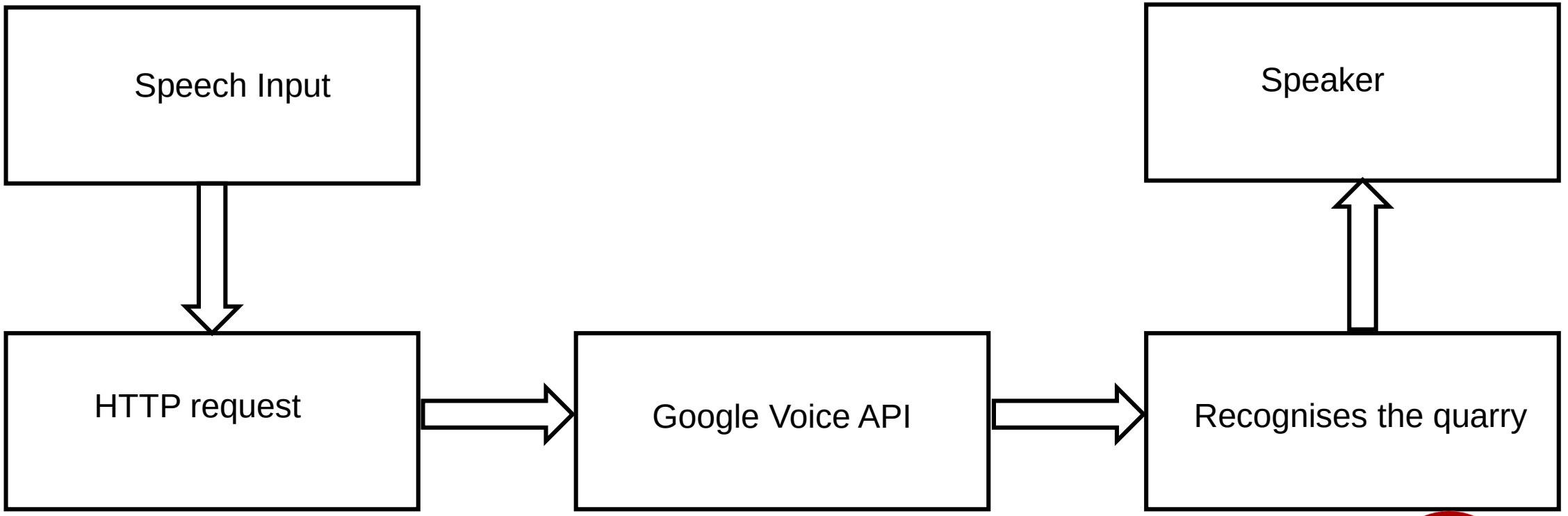
0.08% Cap

1.3% others



Demonstration

Voice Assistance

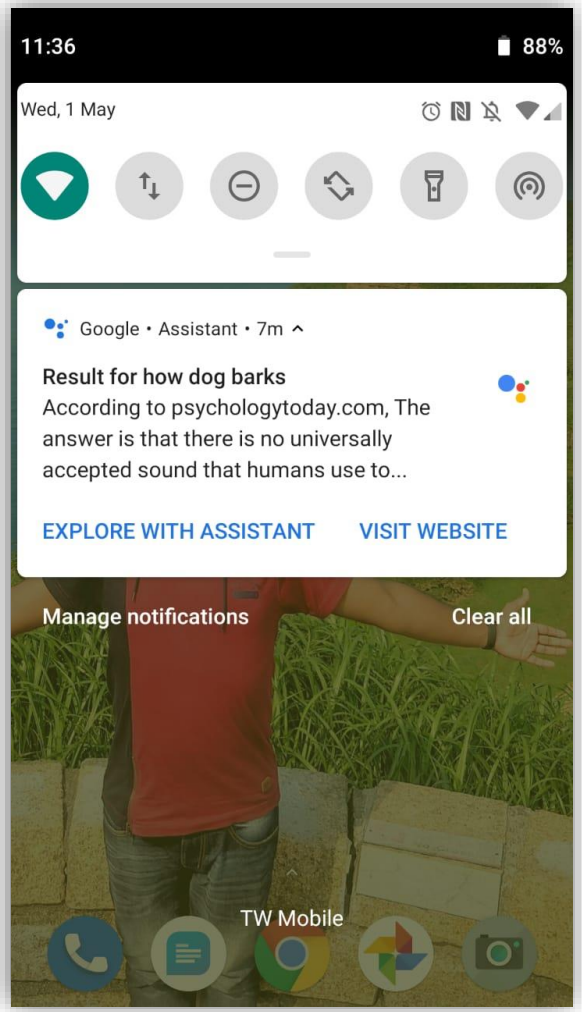


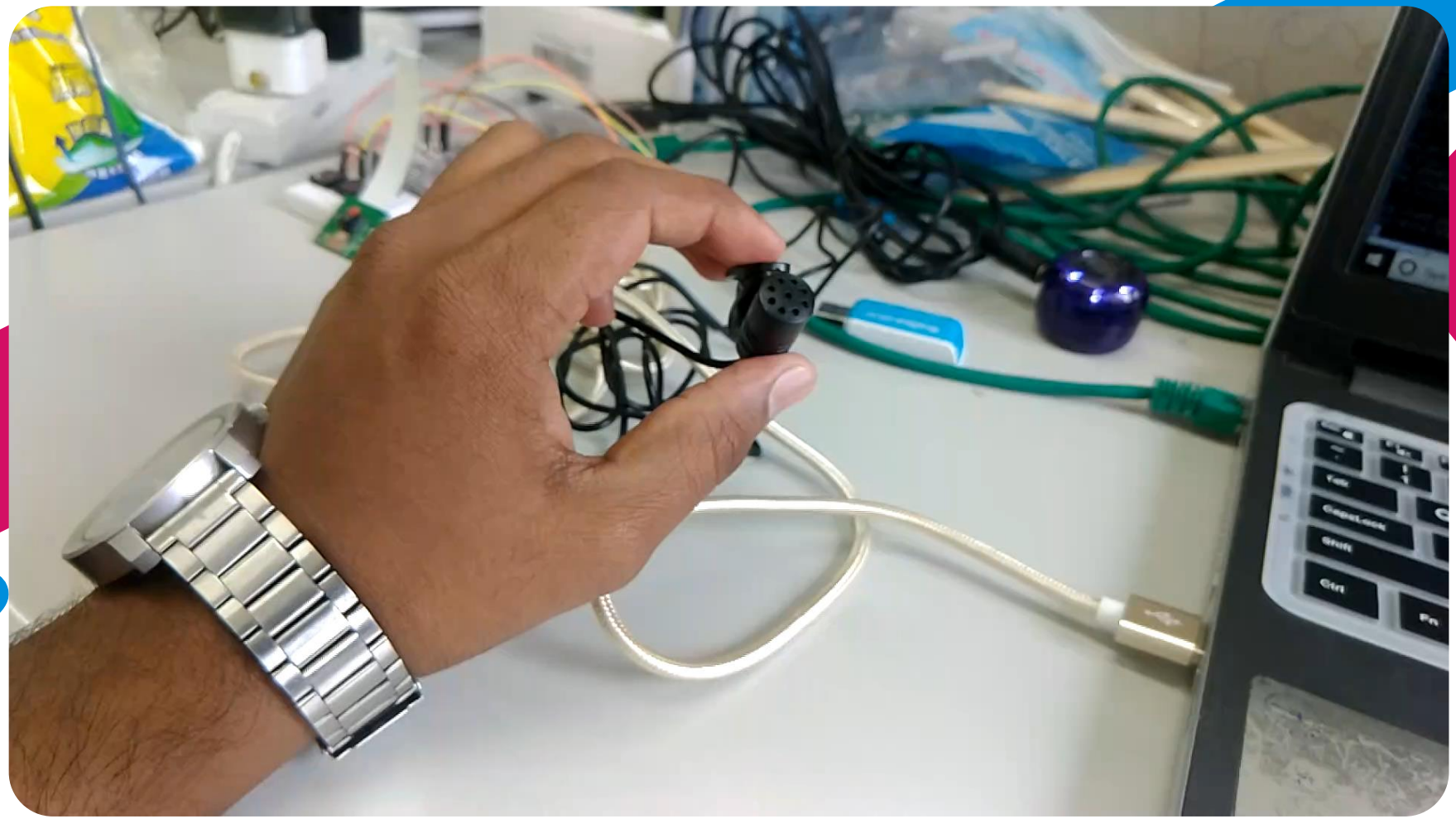
Notifications

Voice Assistant

Synced with Gmail

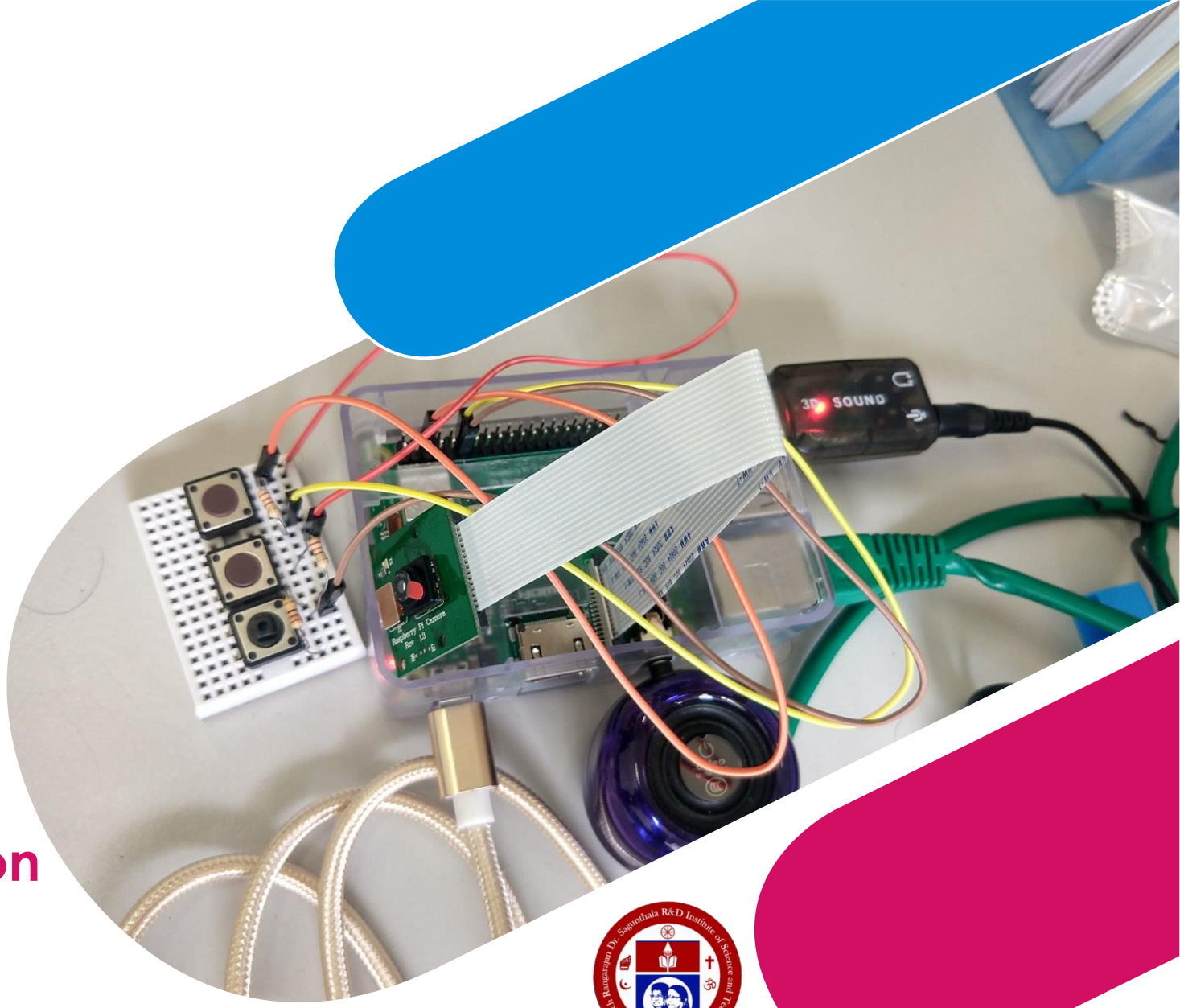
Notification to Android Device





Voice Assistant Demonstration

THANK YOU!



MUKESH BADIGINENI

Dept:
Electronics & Communication

VTU:
Vtu-6569 / 15UEEC0013