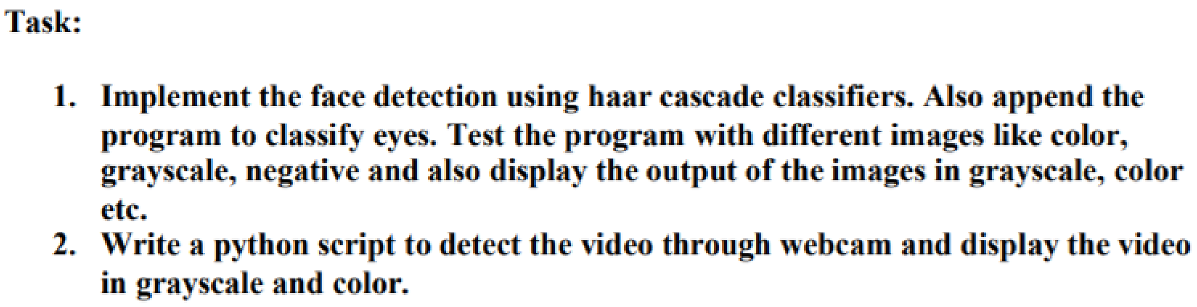
Name: **Swarup Tripathy** Assignment Number: **2**
Course: **ECE3502: IoT Domain Analyst** Date: February 9, 2022

Reg No: 19BEE0167

Problem 1



Task:

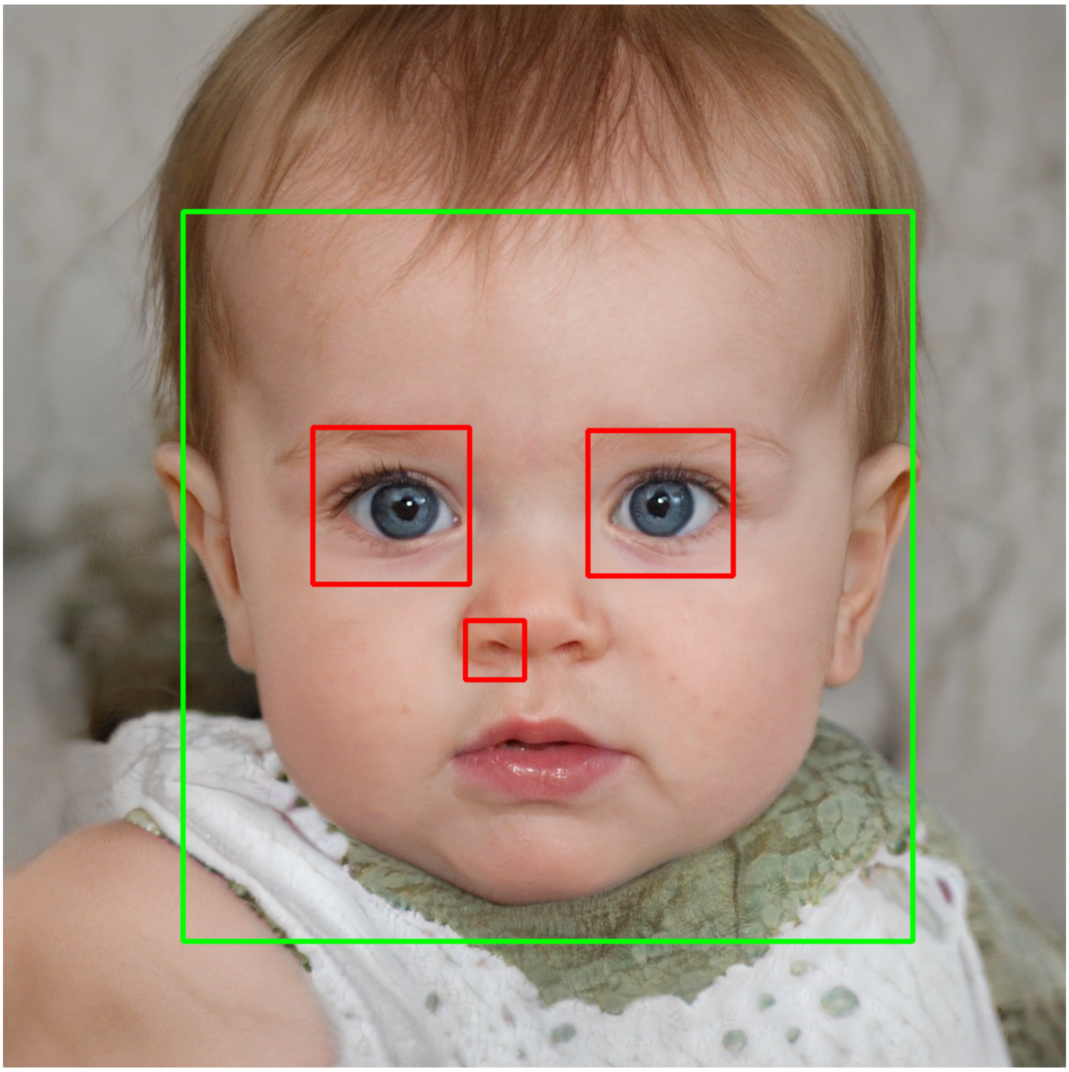
1. **Implement the face detection using haar cascade classifiers. Also append the program to classify eyes. Test the program with different images like color, grayscale, negative and also display the output of the images in grayscale, color etc.**
2. **Write a python script to detect the video through webcam and display the video in grayscale and color.**

1 Python Code

```
# Necessary imports
import cv2
import numpy as np
from google.colab.patches import cv2_imshow

# Loading the image
img = cv2.imread("/content/baby.jfif")
# cv2_imshow(img)

face_cascade = cv2.CascadeClassifier("/content/haarcascade_frontalface_default.xml")
eye_cascade = cv2.CascadeClassifier("/content/haarcascade_eye.xml")

gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# Creating an object faces
faces= face_cascade.detectMultiScale (gray, 1.1, 10)
# Drawing rectangle around the face
for(x , y,  w,  h) in faces:
  cv2.rectangle(img, (x,y) ,(x+w, y+h), (0,255,0), 3)
# cv2_imshow(img)

# Creating two objects of interest
roi_gray=gray[y:(y+h), x:(x+w)]
roi_color=img[y:(y+h), x:(x+w)]
```

```
eyes = eye_cascade.detectMultiScale(roi_gray, 1.1, 10)
for (x_eye, y_eye, w_eye, h_eye) in eyes:
  cv2.rectangle(roi_color,(x_eye, y_eye),(x_eye+w_eye, y_eye+h_eye), (0, 0, 255), 3)
cv2_imshow(img)
```

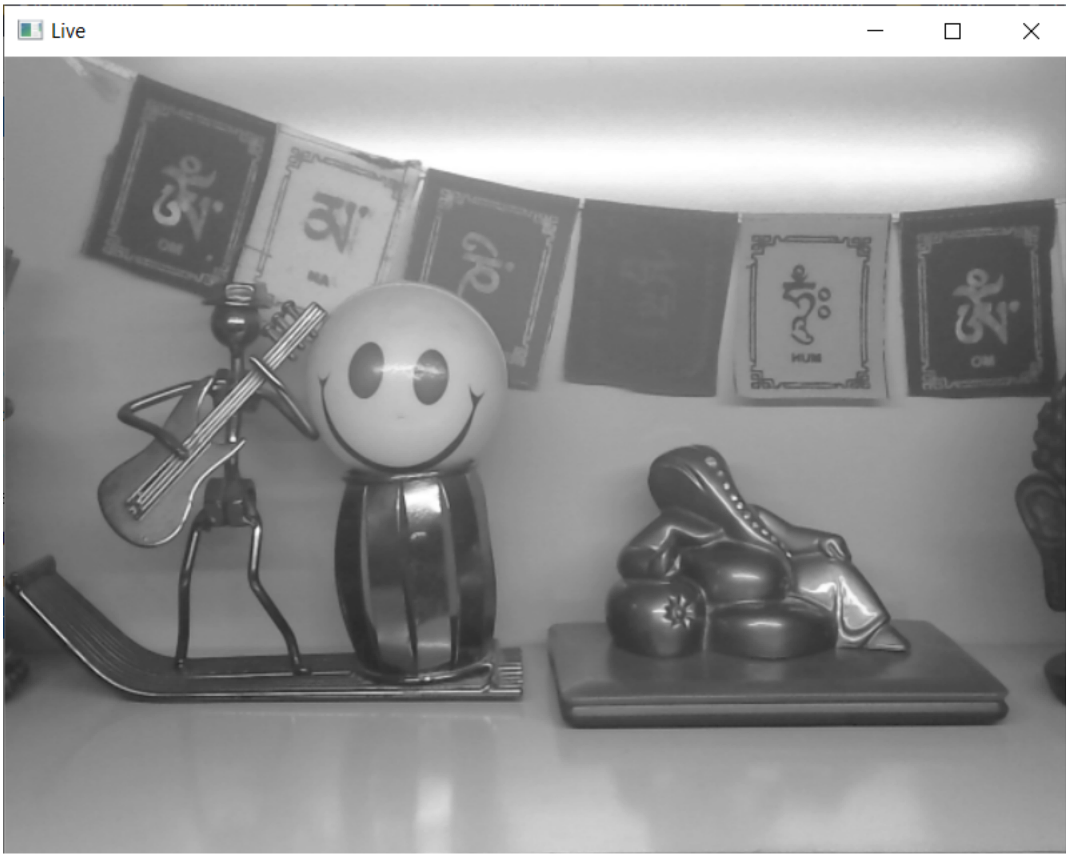
1.1 Code output



2 Python Code for problem 2

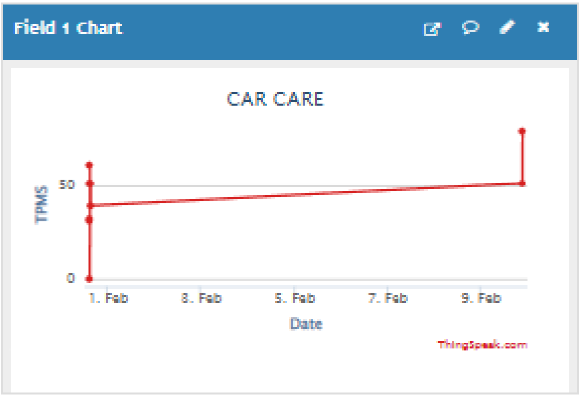
```
import cv2

# reading the video
source = cv2.VideoCapture(0)

# running the loop
while True:

    # extracting the frames
    ret, img = source.read()
    # converting to gray-scale
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    # displaying the video
    cv2.imshow("Live", gray)

    # exiting the loop
    key = cv2.waitKey(1)
    if key == ord("q"):
        break

# closing the window
cv2.destroyAllWindows()
source.release()
```

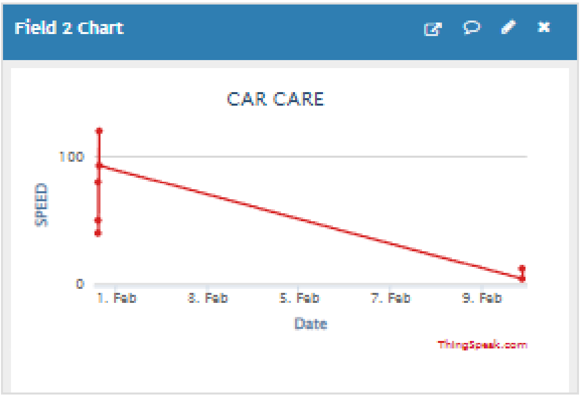
2.1 Code output

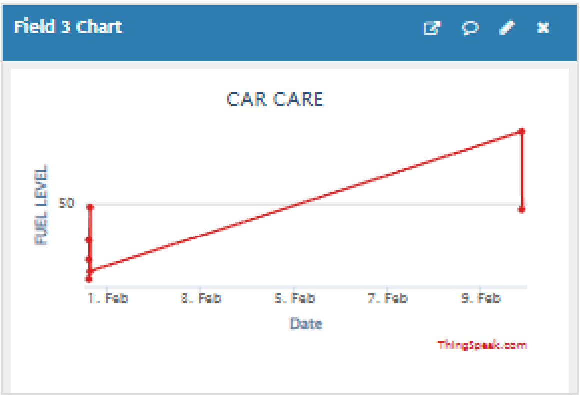


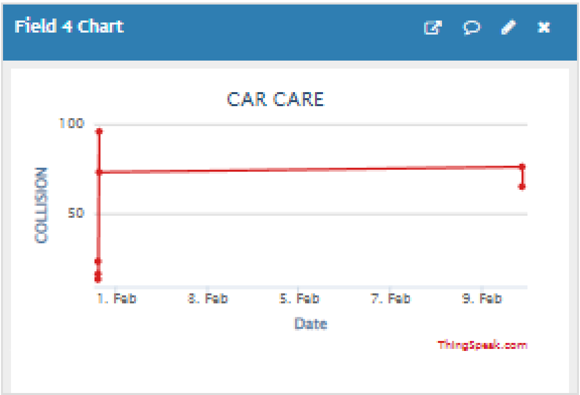
Problem 2

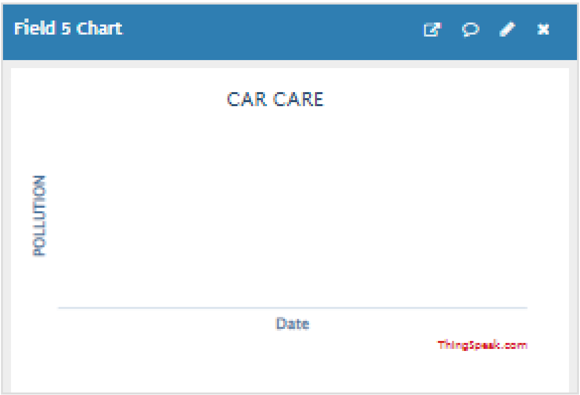
1. A. Create a Channel in thingspeak
B. Write data to a field using HTTP push.
C. Read data from a Field directly
2. Create a channel in ThingSpeak to have 5 fields. (field1 – Tyre Pressure, field2 – Speed, field3-fuel, field4 – collision, field5-pollution)
3. Read the thingspeak and find the maximum tyrepressure, maximum speed in python
4. Read the file "pollution.csv" using Python and push the temperature (Column Label 'Temp') data to field5.











3 Python Code

```
import sys
import urllib
import urllib.request

# Posting a random value to the field1
import random
a= random.randint(0,100);
b= random.randint(0,100);
c= random.randint(0,100);
d= random.randint(0,100);

data=urllib.request.urlopen("https://api.thingspeak.com/update?api_key=HVCGD
OI5QMNGDXR5&field1="+str(a)+"&field2="+str(b)+"&field3="+str(c)+"&field4="+str(d))

for i in range(1,5):
  rd=urllib.request.urlopen('https://api.thingspeak.com/channels/1643821/fields/%s.json?
  api_key=URI80Q0KQ1WN7SOA&results=2' \
                            % (i))
response = rd.read()
data = json.loads(response)
print(data)
print(data['channel']['field1'])
```

3.1 Output

```
    {'channel': {'id': 1643821, 'name': 'CAR CARE', 'description': 'Measures the
    safety of car', 'latitude': '0.0', 'longitude': '0.0', 'field1': 'TPMS', 'field2':
    'SPEED', 'field3': 'FUEL LEVEL', 'field4': 'LDR', 'created_at':
    '2022-01-31T08:55:49Z', 'updated_at': '2022-01-31T08:55:49Z', 'last_entry_id': 7},
    'feeds': [{'created_at':'2022-01-31T09:31:49Z', 'entry_id': 6, 'field4': '96'},
    {'created_at': '2022-01-31T09:32:47Z', 'entry_id': 7, 'field4': '73'}]}
TPMS
```

4 Python Code

```
    import json
import time

READ_API_KEY='HVCGDOI5QMNGDXR5'
CHANNEL_ID= '1643821'

t=0
while True:
    for i in range(1,5):
      TS = urllib.request.urlopen("http://api.thingspeak.com/channels/%s/feeds/%s.json?
      api_key=%s" \
                                  % (CHANNEL_ID,i,READ_API_KEY))
      response = TS.read()
      data=json.loads(response)
    a = data['created_at']
    b = data['field1']
    c = data['field2']
    d = data['field3']
    e = data['field4']
    print(a + "    " + b + "    " + c + "    " + d+"   "+e)
    t=t+1

    TS.close()
```

4.1 OUTPUT

```
    2022-01-31T09:06:49Z    61    50    11   13
```

5 Python Code for sending the data from pollution.csv file to Thingspeak

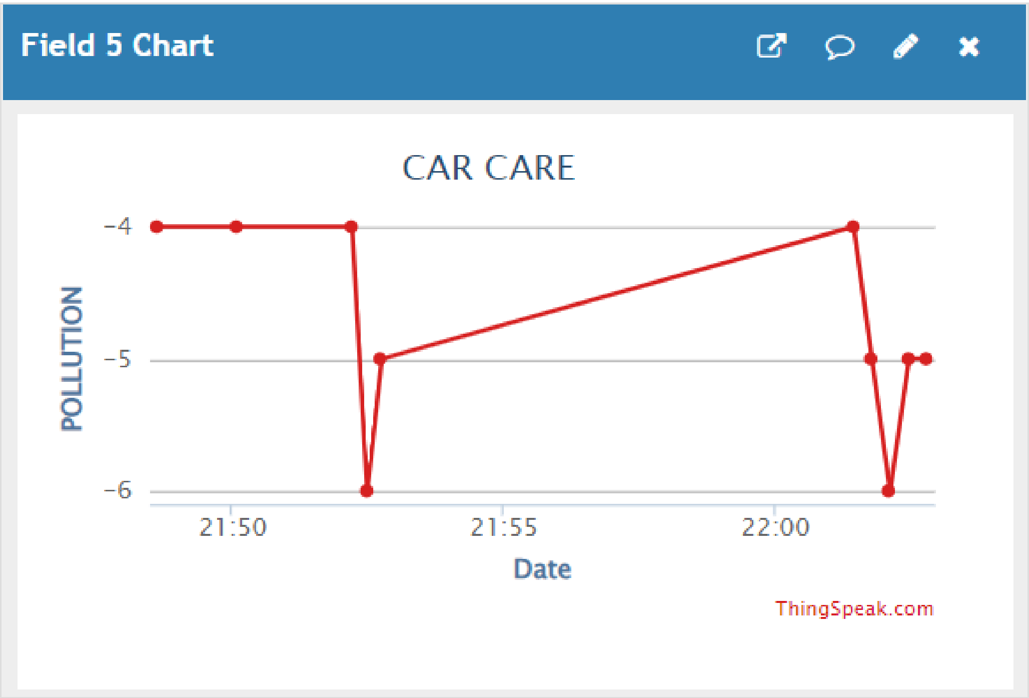
Here we use the library 'time' where we give a delay of 4 seconds after which the data will be sent to server. Herein, we are sending the first 10 data to the server.

```
import pandas as pd
df = pd.read_csv('/content/pollution.csv')
df10=newdf.head(10) #temperature values
print(df10)
arr = df10.to_numpy()
print(arr)
arr[1]
print(len(arr))

import time
for i in range(len(arr)):
  data=urllib.request.urlopen("https://api.thingspeak.com/update?api_key=HVCGDOI5QMNGD
  XR5&field5="+str(arr[i]))
  time.sleep(4)
```

5.1 OUTPUT

```
0   -4
1   -5
2   -5
3   -5
4   -6
5   -6
6   -5
7   -6
8   -5
9   -5
```

```
[-4 -5 -5 -5 -6 -6 -5 -6 -5 -5]
10
```


Field 5 Chart
CAR CARE
-4
-5
-6
POLLUTION
21:50
21:55
22:00
Date
ThingSpeak.com