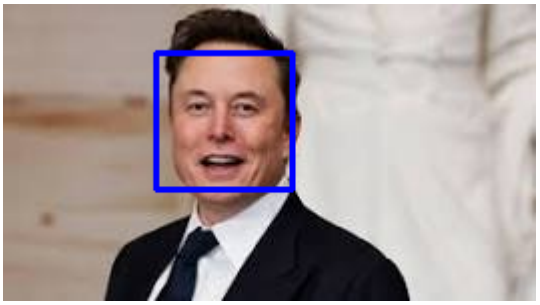
Experiment 6: Load and Implement the Face Detection Method in OpenCV
Date: 25/3/25

Aim:
To implement face detection using OpenCV and the Haar Cascade classifier for detecting faces in an image.

Code:

```
# Step 1: Install OpenCV
!pip install opencv-python-headless

# Step 2: Import libraries
import cv2
from google.colab.patches import cv2_imshow
import numpy as np
from IPython.display import Image
from PIL import Image as PILImage
import io
from google.colab import files

# Step 3: Upload an image
uploaded = files.upload()
for fn in uploaded.keys():
    image_path = fn

# Step 4: Load the image
img = cv2.imread(image_path)
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# Step 5: Load Haar cascade for face detection
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')

# Step 6: Detect faces
faces = face_cascade.detectMultiScale(gray, 1.1, 4)

# Step 7: Draw rectangles around faces
for (x, y, w, h) in faces:
    cv2.rectangle(img, (x, y), (x+w, y+h), (255, 0, 0), 2)

# Step 8: Display the result
cv2_imshow(img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Output:



Result:
The face detection model using OpenCV's Haar Cascade classifier successfully detected faces in the uploaded image and displayed them with rectangles around the faces, demonstrating the effective application of this method.