6: FACE DETECTION USING OPENCV

AIM:

To load and implement face detection using Haar cascades in OpenCV.

PROCEDURE:

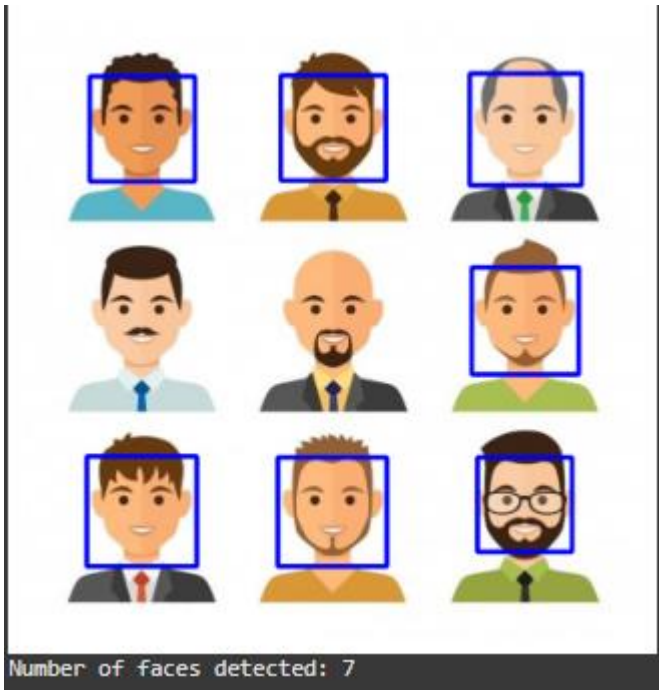
1. Load Haar cascade for face detection.
2. Open webcam feed.
3. Convert each frame to grayscale.
4. Detect faces in the frame.
5. Draw rectangles around detected faces.

CODE:

```
import cv2
import matplotlib.pyplot as plt
from google.colab import files
import numpy as np


# Load the pre-trained Haar cascade classifier for face detection
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +
'haarcascade_frontalface_default.xml')


# Upload an image
uploaded = files.upload()
image_path = list(uploaded.keys())[0]
```

```
image = cv2.imread(image_path)

# Convert image to grayscale (Haar cascade works better on grayscale images)
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# Detect faces with further optimized parameters to ensure one face detection
faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1,
minNeighbors=10, minSize=(80, 80))

# Print detected face coordinates
print("Detected Faces:", faces)

# Draw rectangles around detected faces
for (x, y, w, h) in faces:
    cv2.rectangle(image, (x, y), (x + w, y + h), (255, 0, 0), 3)

# Convert BGR image to RGB for displaying with matplotlib
image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# Display the image
plt.imshow(image_rgb)
plt.axis('off')
plt.show()

print(f"Number of faces detected: {len(faces)}")
```

OUTPUT:



RESULT:

Real-time face detection was successfully implemented using OpenCV