Face and Eye Detection by using Haar Cascade Classifiers

Face Detection by using HaarCascade Classifier

In [ ]:
```
import numpy as np
import cv2
# We point OpenCV's CascadeClassifier function to where our
# classifier(XML file format) is stored
face_classifier = cv2.CascadeClassifier('Haarcascades/haarcascade_frontalface_dafault.xml')
# load our image then convert it to grayscale
image = cv2.imread(r'C:/Users/HP/Pictures/My Pics/shivani01.jpg')
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
# Our classifier returns the ROI of the detected face as a tuple
# It stores the top left coordinate and the bottom right coordiantes
# Load the cascade
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')

# Use the cascade to detect faces
faces = face_cascade.detectMultiScale(gray, scaleFactor=1.3, minNeighbors=5)

#faces = face_classifier.detectMultiScale(gray, 1.3, 5)
# When no faces detected, face_classifier returns and empty tuple
#if not faces:
   # print("No faces found")
if faces is None or len(faces) == 0:
    print("No faces found")
else:
    print(f"Number of faces found: {len(faces)}")
    # Add code to draw rectangles around the faces

    # Add code to draw rectangles around the faces

# We iterate through our faces array and draw a rectangle
# over each face in faces
for (x,y,w, h) in faces:
    cv2.rectangle(image, (x,y), (x+w,y+h),(127,0, 255),2)
    cv2.imshow('Face Detection', image)
    cv2.waitKey(0)
cv2.destroyAllWindows()
```

Convert image into gray color

In [ ]:
```
import cv2

# Load our image
image = cv2.imread(r'C:/Users/HP/Pictures/My Pics/shivani01.jpg')
#image = cv2.imread(image_path)

# Check if the image is loaded successfully
if image is not None:
    # Convert the image to grayscale
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    # Continue with the rest of your code...
    # (e.g., face detection using the classifier)

    # Display the grayscale image
    cv2.imshow('Grayscale Image', gray)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
else:
    print(f"Error: Unable to load the image at path {image_path}.")
```

Lets make a live face and eye detection(detect eye and face on webcame live view) keeping the face inview at all times

In [ ]:
```python
import cv2

# Load the cascade classifiers
face_classifier = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
eye_classifier = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_eye.xml')

def face_and_eye_detector(img, size=0.5):
    # Convert image to grayscale
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    # Detect faces
    faces = face_classifier.detectMultiScale(gray, scaleFactor=1.3, minNeighbors=5)

    for (x, y, w, h) in faces:
        cv2.rectangle(img, (x, y), (x+w, y+h), (255, 0, 0), 2)
        roi_gray = gray[y:y+h, x:x+w]
        roi_color = img[y:y+h, x:x+w]

        # Detect eyes within the region of interest (ROI)
        eyes = eye_classifier.detectMultiScale(roi_gray, scaleFactor=1.3, minNeighbors=5)

        for (ex, ey, ew, eh) in eyes:
            cv2.rectangle(roi_color, (ex, ey), (ex+ew, ey+eh), (0, 255, 0), 2)

    return img

# Your main loop
cap = cv2.VideoCapture(0)

while True:
    ret, frame = cap.read()
    cv2.imshow('Face and Eye Extractor', face_and_eye_detector(frame))

    if cv2.waitKey(1) == 13:  # 13 is the Enter Key
        break

cap.release()
cv2.destroyAllWindows()
```

Face Detection

In [ ]:
```python
import cv2

# Load the cascade classifier
face_classifier = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')

# Read the image
img = cv2.imread('C:/Users/HP/Pictures/My Pics/shivani01.jpg')

# Convert the image to grayscale
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# Detect faces
faces = face_classifier.detectMultiScale(gray, scaleFactor=1.3, minNeighbors=5)

# When no faces detected, face_classifier returns an empty array
if len(faces) == 0:
    print("No faces found")
else:
    # Process the detected faces (e.g., draw rectangles)
    for (x, y, w, h) in faces:
        cv2.rectangle(img, (x, y), (x+w, y+h), (255, 0, 0), 2)

    # Display the image with rectangles around detected faces
    cv2.imshow('Detected Faces', img)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
```

Face and Eye Detection on Loaded Image

In [*]:
```
import cv2

# Load the cascade classifiers
face_classifier = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
eye_classifier = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_eye.xml')

# Read the image
img = cv2.imread(r'C:/Users/HP/Pictures/My Pics/shivani01.jpg')


# Convert the image to grayscale
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# Detect faces
faces = face_classifier.detectMultiScale(gray, scaleFactor=1.3, minNeighbors=5)

# When no faces detected, face_classifier returns an empty array
if len(faces) == 0:
    print("No faces found")
else:
    # Process the detected faces
    for (x, y, w, h) in faces:
        cv2.rectangle(img, (x, y), (x+w, y+h), (255, 0, 0), 2)

        # Region of Interest (ROI) for eyes within the detected face
        roi_gray = gray[y:y+h, x:x+w]
        roi_color = img[y:y+h, x:x+w]

        # Detect eyes within the ROI
        eyes = eye_classifier.detectMultiScale(roi_gray)

        # Process the detected eyes
        for (ex, ey, ew, eh) in eyes:
            cv2.rectangle(roi_color, (ex, ey), (ex+ew, ey+eh), (0, 255, 0), 2)

    # Display the image with rectangles around detected faces and eyes
    cv2.imshow('Detected Faces and Eyes', img)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
```

In [ ]: