In [ ]:

```

```

In [ ]:

```

```

OPENCV

Reading,Writing and Displaying images

In [42]:

```python
# Press CTRL + ENTER to run this line
# You should see an * between the [ ] on the left
# OpenCV takes a couple seconds to import the first time

import cv2
```

In [43]:

```python
# Now let's import numpy
# We use as np, so that everything we call on numpy, we can type np instead
# It's short and looks neater

import numpy as np
```

In [1]:

```python
# We don't need to do this again, but it's a good habit
import cv2

# Load an image using 'imread' specifying the path to image
input = cv2.imread('E:\Data Science & AI\Dataset files\Tensorflow_env\Happy & Sa

# Our file 'input.jpg' is now loaded and stored in python
# as a varaible we named 'image'

# To display our image variable, we use 'imshow'
# The first parameter will be title shown on image window
# The second parameter is the image varialbe
cv2.imshow('Test Boy Image', input)

# 'waitKey' allows us to input information when a image window is open
# By leaving it blank it just waits for anykey to be pressed before
# continuing. By placing numbers (except 0), we can specify a delay for
# how long you keep the window open (time is in milliseconds here)
cv2.waitKey()

# This closes all open windows
# Failure to place this will cause your program to hang
cv2.destroyAllWindows()
```

```
<>:5: SyntaxWarning: invalid escape sequence '\D'
<>:5: SyntaxWarning: invalid escape sequence '\D'
C:\Users\roy62\AppData\Local\Temp\ipykernel_16196\3732839095.py:5: SyntaxWarning:
invalid escape sequence '\D'
  input = cv2.imread('E:\Data Science & AI\Dataset files\Tensorflow_env\Happy & S
ad\Testing\images (2).jpg')
```

In [2]:

```python
import cv2

input = cv2.imread('E:\Data Science & AI\Dataset files\Tensorflow_env\Happy & Sa

cv2.imshow('Test Boy image', input)
cv2.waitKey()
cv2.destroyAllWindows()
```

```
<>:3: SyntaxWarning: invalid escape sequence '\D'
<>:3: SyntaxWarning: invalid escape sequence '\D'
C:\Users\roy62\AppData\Local\Temp\ipykernel_16196\2843710759.py:3: SyntaxWarning:
invalid escape sequence '\D'
  input = cv2.imread('E:\Data Science & AI\Dataset files\Tensorflow_env\Happy & S
ad\Testing\images (2).jpg')
```

In [46]:
```
# Let's take a closer look at how images are stored
        # Import numpy

import numpy as np
```

In [47]:
```
print(input.shape)
```

```
(194, 259, 3)
```

In [48]:
```
# Shape gives the dimensions of the image array

    ##The 2D dimensions are 830 pixels in high bv 1245 pixels wide.
    ## The '3L' means that there are 3 other components (RGB) that make up this

# Let's print each dimension of the image

print('Height of Image:', int(input.shape[0]), 'pixels')
print('Width of Image: ', int(input.shape[1]), 'pixels')
```

```
Height of Image: 194 pixels
Width of Image:  259 pixels
```

In [49]:
```
# Simply use 'imwrite' specificing the file name and the image to be saved
cv2.imwrite('output.jpg', input)
cv2.imwrite('output.png', input)
```

Out[49]:
```
True
```

In [ ]:
```
```

Face & Eye Detection

In [50]:
```
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('E:\Data Science & AI\Dataset files\Haar

# Load our image then convert it to grayscale
image = cv2.imread('E:\Data Science & AI\Dataset files\Tensorflow_env\Happy & Sa
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
faces = face_classifier.detectMultiScale(gray, 1.3, 5)

# When no faces detected, face_classifier returns and empty tuple
```

```
if faces is ():
    print("No faces found")

# 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()
```

```
<>:6: SyntaxWarning: invalid escape sequence '\D'
<>:9: SyntaxWarning: invalid escape sequence '\D'
<>:17: SyntaxWarning: "is" with 'tuple' literal. Did you mean "=="?
<>:6: SyntaxWarning: invalid escape sequence '\D'
<>:9: SyntaxWarning: invalid escape sequence '\D'
<>:17: SyntaxWarning: "is" with 'tuple' literal. Did you mean "=="?
C:\Users\roy62\AppData\Local\Temp\ipykernel_24792\1329246641.py:6: SyntaxWarning:
invalid escape sequence '\D'
  face_classifier = cv2.CascadeClassifier('E:\Data Science & AI\Dataset files\Haa
rcascades\haarcascade_frontalface_default.xml')
C:\Users\roy62\AppData\Local\Temp\ipykernel_24792\1329246641.py:9: SyntaxWarning:
invalid escape sequence '\D'
  image = cv2.imread('E:\Data Science & AI\Dataset files\Tensorflow_env\Happy & S
ad\Testing\images (3).jpg')
C:\Users\roy62\AppData\Local\Temp\ipykernel_24792\1329246641.py:17: SyntaxWarnin
g: "is" with 'tuple' literal. Did you mean "=="?
  if faces is ():
```

In [51]:

```
# Let's combine face and eye detection

import numpy as np
import cv2

face_classifier = cv2.CascadeClassifier("E:\Data Science & AI\Dataset files\Haar
eye_classifier = cv2.CascadeClassifier("E:\Data Science & AI\Dataset files\Haarc

img = cv2.imread('E:\Data Science & AI\Dataset files\Tensorflow_env\Happy & Sad\
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

faces = face_classifier.detectMultiScale(gray, 1.3, 5)

# When no faces detected, face_classifier returns and empty tuple
if faces is ():
    print("No Face Found")

for (x,y,w,h) in faces:
    cv2.rectangle(img,(x,y),(x+w,y+h),(127,0,255),2)
    cv2.imshow('img',img)
    cv2.waitKey(0)
    roi_gray = gray[y:y+h, x:x+w]
    roi_color = img[y:y+h, x:x+w]
    eyes = eye_classifier.detectMultiScale(roi_gray)
    for (ex,ey,ew,eh) in eyes:
        cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(255,255,0),2)
        cv2.imshow('img',img)
        cv2.waitKey(0)

cv2.destroyAllWindows()
```

```
<>:6: SyntaxWarning: invalid escape sequence '\D'
<>:7: SyntaxWarning: invalid escape sequence '\D'
<>:9: SyntaxWarning: invalid escape sequence '\D'
<>:15: SyntaxWarning: "is" with 'tuple' literal. Did you mean "=="?
<>:6: SyntaxWarning: invalid escape sequence '\D'
<>:7: SyntaxWarning: invalid escape sequence '\D'
<>:9: SyntaxWarning: invalid escape sequence '\D'
<>:15: SyntaxWarning: "is" with 'tuple' literal. Did you mean "=="?
C:\Users\roy62\AppData\Local\Temp\ipykernel_24792\3611651891.py:6: SyntaxWarning: 
invalid escape sequence '\D'
  face_classifier = cv2.CascadeClassifier("E:\Data Science & AI\Dataset files\Haa
rcascades\haarcascade_frontalface_default.xml")
C:\Users\roy62\AppData\Local\Temp\ipykernel_24792\3611651891.py:7: SyntaxWarning: 
invalid escape sequence '\D'
  eye_classifier = cv2.CascadeClassifier("E:\Data Science & AI\Dataset files\Haar
cascades\haarcascade_eye.xml")
C:\Users\roy62\AppData\Local\Temp\ipykernel_24792\3611651891.py:9: SyntaxWarning: 
invalid escape sequence '\D'
  img = cv2.imread('E:\Data Science & AI\Dataset files\Tensorflow_env\Happy & Sad
\Testing\images (3).jpg')
C:\Users\roy62\AppData\Local\Temp\ipykernel_24792\3611651891.py:15: SyntaxWarnin
g: "is" with 'tuple' literal. Did you mean "=="?
  if faces is ():
```

In [ ]:

```python
# Let's make a live face & eye detection, keeping the face inview at all times

import cv2
import numpy as np

face_classifier = cv2.CascadeClassifier("E:\Data Science & AI\Dataset files\Haar
eye_classifier = cv2.CascadeClassifier("E:\Data Science & AI\Dataset files\Haarc

def face_detector(img, size=0.5):
    # Convert image to grayscale
    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    faces = face_classifier.detectMultiScale(gray, 1.3, 5)
    if faces is ():
        return img

    for (x,y,w,h) in faces:
        x = x - 50
        w = w + 50
        y = y - 50
        h = h + 50
        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]
        eyes = eye_classifier.detectMultiScale(roi_gray)

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

    roi_color = cv2.flip(roi_color,1)
    return roi_color

cap = cv2.VideoCapture(0)

while True:

    ret, frame = cap.read()
    cv2.imshow('Our Face Extractor', face_detector(frame))
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

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

Face and Eye Detection from videos

In [ ]:
```
# Face Recognition

# Importing the libraries
import cv2

# Loading the cascades
face_cascade = cv2.CascadeClassifier("E:\Data Science & AI\Dataset files\Haarcas
eye_cascade = cv2.CascadeClassifier("E:\Data Science & AI\Dataset files\Haarcasc

# Defining a function that will do the detections
def detect(gray, frame):
    faces = face_cascade.detectMultiScale(gray, 1.3, 5)
    for (x, y, w, h) in faces:
        cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 0, 0), 2)
        roi_gray = gray[y:y+h, x:x+w]
        roi_color = frame[y:y+h, x:x+w]
        eyes = eye_cascade.detectMultiScale(roi_gray, 1.1, 3)
        for (ex, ey, ew, eh) in eyes:
            cv2.rectangle(roi_color, (ex, ey), (ex+ew, ey+eh), (0, 255, 0), 2)
    return frame

# Doing some Face Recognition with the webcam
video_capture = cv2.VideoCapture(0)
while True:
    _, frame = video_capture.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    canvas = detect(gray, frame)
    cv2.imshow('Video', canvas)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

video_capture.release()
cv2.destroyAllWindows()
```

In [ ]:

Car & Pedestrian Detection

In [ ]:
```
import cv2
import numpy as np

# Create our body classifier
body_classifier = cv2.CascadeClassifier("E:\Data Science & AI\Dataset files\Haar

# Initiate video capture for video file
cap = cv2.VideoCapture("E:\Data Science & AI\Dataset files\Cars Moving On Road S
# Loop once video is successfully loaded
while cap.isOpened():

    # Read first frame
    ret, frame = cap.read()
    #frame = cv2.resize(frame, None,fx=0.5, fy=0.5, interpolation = cv2.INTER_LI

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    # Pass frame to our body classifier
    bodies = body_classifier.detectMultiScale(gray, 1.2, 3)

    # Extract bounding boxes for any bodies identified
    for (x,y,w,h) in bodies:
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 255), 2)
        cv2.imshow('Pedestrians', frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

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

In [ ]:

Full Body Detection

In [ ]:

```
import cv2
import numpy as np

# Create our body classifier
body_classifier = cv2.CascadeClassifier("E:\Data Science & AI\Dataset files\Haar

# Initiate video capture for video file
cap = cv2.VideoCapture("E:\Data Science & AI\Dataset files\People walking.mp4")

# Loop once video is successfully loaded
while cap.isOpened():

    # Read first frame
    ret, frame = cap.read()
    #frame = cv2.resize(frame, None,fx=0.5, fy=0.5, interpolation = cv2.INTER_LI

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    # Pass frame to our body classifier
    bodies = body_classifier.detectMultiScale(gray, 1.2, 3)

    # Extract bounding boxes for any bodies identified
    for (x,y,w,h) in bodies:
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 255), 2)
        cv2.imshow('Pedestrians', frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

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

Capture and mouse draw rectangle from webcam and sketch process it on a live feed

In [ ]:
```
import cv2
from matplotlib import pyplot as plt
import numpy as np
```

In [ ]:
```
def sketch_transform(image):
    image_grayscale = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    image_grayscale_blurred = cv2.GaussianBlur(image_grayscale, (7,7), 0)
    image_canny = cv2.Canny(image_grayscale_blurred, 10, 80)
    _, mask = image_canny_inverted = cv2.threshold(image_canny, 30, 255, cv2.THR
    return mask
```

In [ ]:
```
cam_capture = cv2.VideoCapture(0)
cv2.destroyAllWindows()

while True:
    _, im0 = cam_capture.read()
    showCrosshair = False
    fromCenter = False
    r = cv2.selectROI("Image", im0, fromCenter, showCrosshair)
    break

while True:
    _, image_frame = cam_capture.read()

    rect_img = image_frame[int(r[1]):int(r[1]+r[3]), int(r[0]):int(r[0]+r[2])]

    sketcher_rect = rect_img
    sketcher_rect = sketch_transform(sketcher_rect)

    #Conversion for 3 channels to put back on original image (streaming)
    sketcher_rect_rgb = cv2.cvtColor(sketcher_rect, cv2.COLOR_GRAY2RGB)

    #Replacing the sketched image on Region of Interest
    image_frame[int(r[1]):int(r[1]+r[3]), int(r[0]):int(r[0]+r[2])] = sketcher_r

    cv2.imshow("Sketcher ROI", image_frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cam_capture.release()
cv2.destroyAllWindows()
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: