Dt: 12/5/22
10:00AM

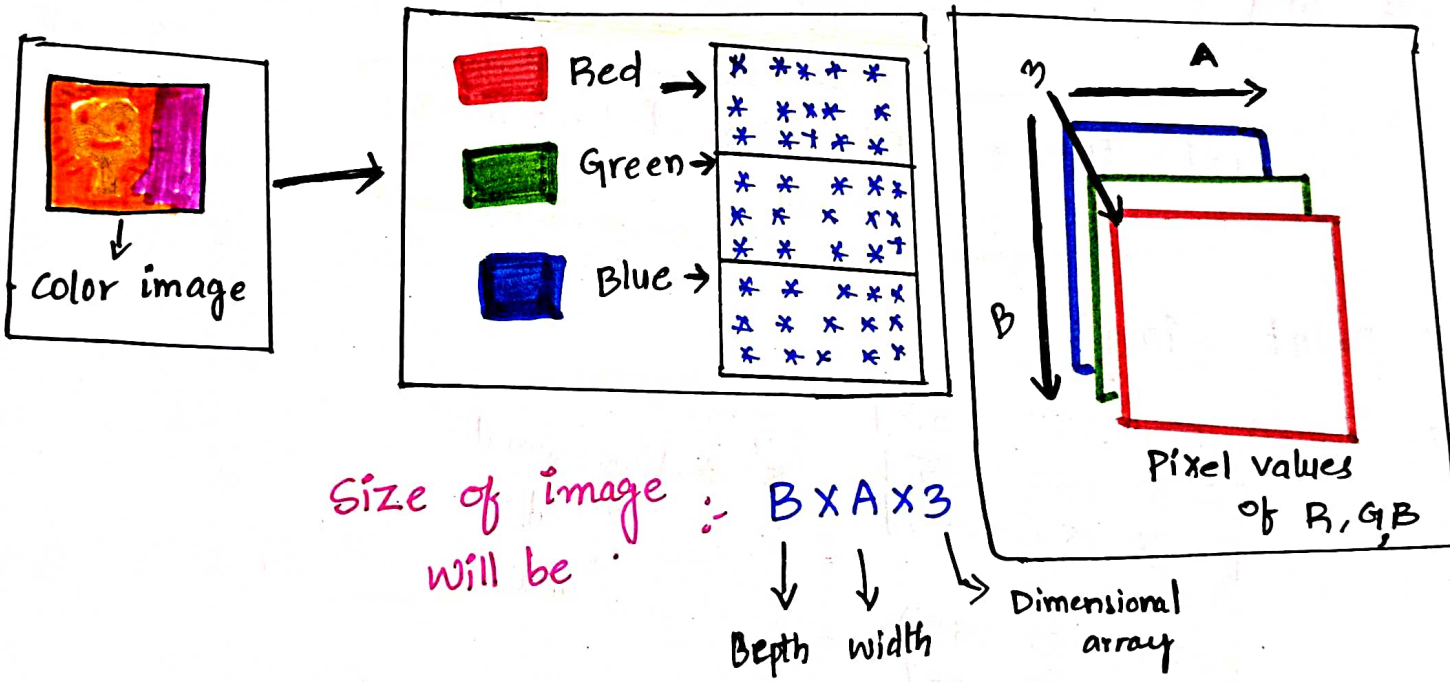
Computer Vision

Open CV

Q. What is Computer Vision?

Object detection, it identifying various object, it used for image recognisation.

Q How Computer reads an image ?



Size of image will be :- B X A X 3

Depth, width, Dimensional array

Q. What is open cv ?

* Open cv is a liberary of python
* it is designed to Solve computer vision problems
* it converts images into "Numpy arrays"
* From that we process the data.

[illegible]

```
import cv2
import numpy as np
```

Load/Read Image

Grayscale (b&w)

```
# Load an image using "imread" specifying the path to image.
# img = cv2.imread("image_examples/modi.jpg", 1)
```

Path (or) location

1 → Color image.

Let's take a closer look at how images are stored.

```
# print (img)
```

Out:

```
[[[255 255 255]
  [255 255 255]
  [255 255 255]
  ....
  [255 255 255]
  [255 255 255]
  [255 255 255]]
 ...
 :
```

→ 3 dimensional array → values.

Shape gives the dimensions of image array

```
# img.shape
```

Out: (1358, 1500, 3) → Tuple

1358 x 1500 pixels

3 → color image.

(1358, 1500) → For B&W (or) Grayscale, 2-Dimensional

```
# img.shape[0]
Out: 1353  → Height

# img.shape[1]
Out: 1500  → width.
```

Display the image

```
# cv2.imshow("PM", img)
```
- "PM" → title of image
- img → image


```
# cv2.waitKey()(0)(2000)
```
- (0) → till you Enter the any key it waits in new page/window
- (2000) → wait for 2000 milliseconds.

```
# cv2.destroyAllWindows()
```
→ closes all windows

Save image

```
# cv2.imwrite("new Pm.jpg", img)
```
→ image is saved to present working directory

```
Out: True
```

Resize image

```
# img = cv2.imread("image_examples/Modi.jpg")
# resized_image = cv2.resize(img, (500, 500))
# cv2.imshow("Modi image", resized_image)
# cv2.waitKey()
# cv2.destroyAllWindows()
```
- "Modi image" → title
- original size = 135
- resize = 500 x 500

img.shape[0] * 0.5 (height)

out: 679.0

img.shape[1] * 0.5 (width)

out: 750.0

```
# img = cv2.imread("image-examples/Modi.JPG")
# resized_image = cv2.resize(img, (679, 750))
                                   50/50
# cv2.imshow("Modi image", resized_image)
                  title
# cv2.waitKey()
# cv2.destroyAllWindows()
```

out

(50 x 50 size)

Face Detection using HAAR Cascade Classifers

Q:- What is HAAR Cascade classifier ?

HAAR cascade classifier will load "XML" files.

Xml : array of values.

Step: 1

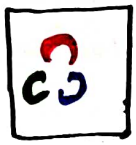
[Face Detection]

image

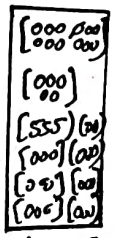
Create a cascade classifier. it will contain the Features of the Face.

↓ Open cv will read the image and feature file.

Step: 2



Open CV →

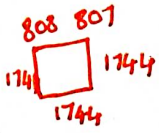


Numpy Array

Search for the row & columns values of the face numpy ndarray (the face rectangle co-ordinates)

↓

Step: 3



Display the image with rectangluar Face box

face_classifier = cv2.CascadeClassifier('haarcascades/haarcascade_frontalface_default.xml')

stores in ↑ — xml file (location)

image = cv2.imread('image_examples/Modi.JPG')

image = cv2.resize(image, (500,500)) → resize

Convert into grayscale (or) b/w image.

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

gray → stored in "gray"

cvtColor → convert color

Colour to Gray (or) Black and white

tuning casacade classifie - detectMultiscale.

faces = face_classifier.detectMultiScale(gray, 1.05, 5)

scaleFactor=1.05

minNeighbors=5

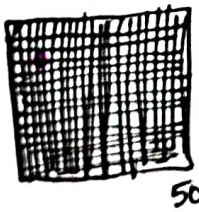
- method to search for the face rectangle co-ordinates
- Decreases the shape value by 5%, until the face is found. Smaller the value, greater the accuracy (zooming)
- Find the Nearest neighbors "5" in the image

print(faces)

Out: [[205, 79, 217, 217]]

x y

↳ Coordinates Found in the image.



500×500 Pixels

In rectangle box "Face is available"

205 + 217 = 422

79 + 217 = 296

when no faces detected, face_classifier returns and empty tuple

```
if face is ():
    Print ("No faces found")
```

we Iterate through our faces array and draws a rectangle

over each faces in faces

```
for (x,y,w,h) in faces :
    cv2.rectangle (image , (x,y) , (x+w,y+h) ,(0,0,255)
                                                ,2)
```

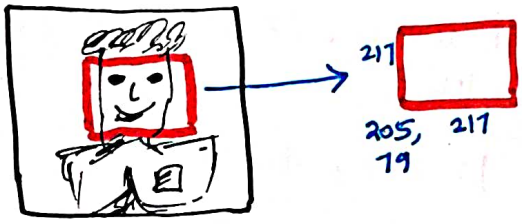
x+width
y+height

(colors) intensity

Thickness of line of Rectangular box

```
# cv2.imshow ("Face Detection", image)
# cv2.WaitKey ()
# cv2.destroyAllWindows ()
```

out



Face & Eyes Detection

```
# face_classifier = cv2.CascadeClassifier ("Haarcascades/
```

For Face :- haarcascade_frontalface_default.xml")

```
# eye_classifier = cv2.CascadeClassifier ("Haarcascades/
```

For eye : haarcascade_eye.xml")

12/05/22
5:30 pm

img = cv2.imread ('image...example/mod.jpg')

resized_image = cv2.resize (img, (fx=..., fy=...))

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

Convert color to Gray

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

Identify faces.

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

if condition is true, than it work if condition.
if condition is false, its not going to Excaute

```
for (x,y,w,h) in faces:
    cv2.rectangle (resized_image, (x,y), (x+w, y+h), (255,0,0), (2))
```

Blue

```
    roi_gray = gray [y:y+h, x:x+w]
    roi_color = resized_image [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), (0,255,0), 2)
```

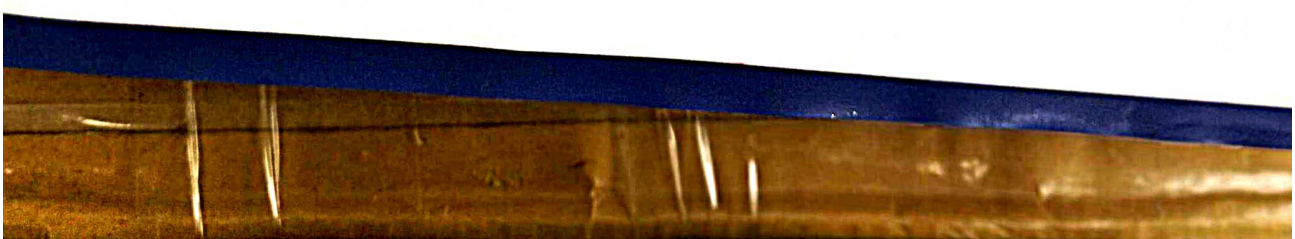
instead Searching total image search only Face image

In Face image detect only eyes

cv2.imshow ("img", resized_image)

cv2.waitKey (0)

cv2.destroyAllWindows ()

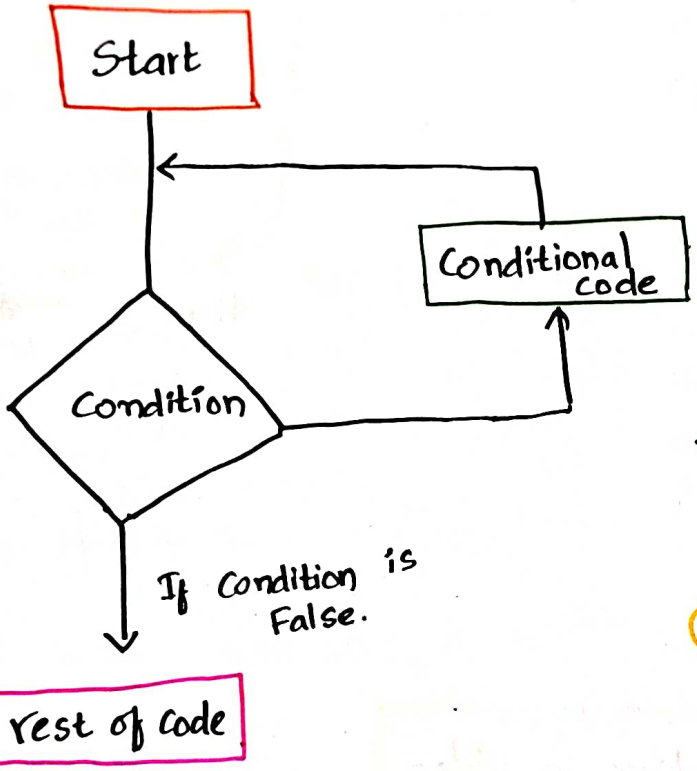


Capture a video

↓

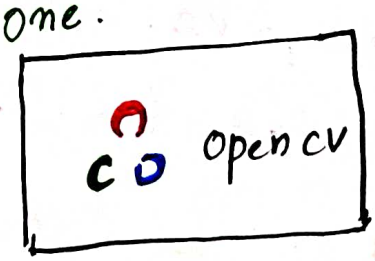
multiple images one-by-one in a loop is called "video".

Loops



We will using loops to build a window, where images will appear really Fast, so that You can see it as a video.

We will be using opencv for reading frames & images one-by-one.



Import cv2

Video = cv2.VideoCapture(0) → Webcam

while True : → Infinite loop (continuously read the images in milliseconds)

check, frame = Video.read()

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

cv2.imshow("Video", frame)
title

If cv2.waitKey(1) == ord("P"):

break

Video.release()

cv2.destroyAllwindows()

Here it is convert it Gray. But, in Video (webcam) it showing color means Here, we wrote Frame, instead of "Gray" If written, it displays. Gray'

Out

Face & Eye Detection in Video

Defining a function that will do the detections.

face_cascade = cv2.CascadeClassifier("Haarcascades/haarcascade_frontalface_default.xml")

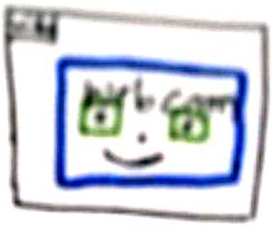
eye_cascade = cv2.CascadeClassifier("Haarcascade/haarcascade_eye.xml")

```
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 webcam.
# video = cv2.VideoCapture(0)

While True:
    check, frame = video.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRA
    canvas = detect(gray, frame)
    cv2.imshow("video", canvas)
    if cv2.waitKey(1) == ord("p")
        break
# video.release()
# cv2.destroyAllWindows()
```

Out:



Pedistrian Detection

```
# create our body classifier
# body_classifier = cv2.CascadeClassifier("Haarcascades\
                    haarcascade_fullbody.xml")

# initiate Video cature for video file.
#cap = cv2.VideoCapture("image_examples/walking.avi")

# loop once Video is Succesfully loaded
While cap.isOpened():
    # read first frame
    check, frame = cap.read()
    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) == ord ("p"):
            break
# cap.release()
# cv2.destroyAllWindows()
```

traffic cars Detections

```
import time
    # create our body classifier
# car_classifier = cv2.CascadeClassifier ("Haarcascades\
                   haarcascade_car.xml")
    # initiate video capture for video file.
# cap = cv2.VideoCapture ("image_examples/car.avi")

# loop once video is succesfully loaded
While cap.isOpened():
        time.sleep(.05)
        # read first frame
        check, frame = cap.read()
```

get boundary boxes for any bodies identified

```
for (x, y, w, h) in cars :
    cv2.rectangle (frame, (x, y), (x+w, y+h),
                   (0, 255, 255), 2)

    cv2.imshow ("cars", frame)

if cv2.waitKey (1) == ord("p"):
    break

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

13/05/22

3:30pm