In [ ]:
```
pip install opencv-python
```

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

In [ ]:
```
img=cv2.imread('stats\image_examples/Modi.jpg',1)
```

In [ ]:
```
print(img)
```

In [ ]:
```
img.shape
```

In [ ]:
```
img=cv2.imread('stats\image_examples/Modi.jpg',1)
cv2.imshow('PM',img)
cv2.waitKey(2000)
cv2.destroyAllWindows()
```

In [ ]:
```
cv2.imwrite("pm_b_w.jpg",img)
```

Resize Image

In [ ]:
```
img=cv2.imread('stats\image_examples/Modi.jpg',1)
resized_image= cv2.resize(img,(500,500))
gray=cv2.cvtColor(resized_image,cv2.COLOR_BGR2GRAY)
cv2.imshow("Modi Image",gray)
cv2.waitKey()
cv2.destroyAllWindows()
```

In [ ]:
```
img.shape[0]*0.5
```

In [ ]:
```
img.shape[1]*0.5
```

Face Detection using HAAR Cascade Classifiers

In [ ]:
```
face_classifier =cv2.CascadeClassifier('stats/Haarcascades/haarcascade_fron
image=cv2.imread('stats\image_examples/Modi.jpg')
image=cv2.resize(img,(500,500))
gray=cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
faces=face_classifier.detectMultiScale(gray,1.05,5)
cv2.imshow("Modi Image",gray)
cv2.waitKey()
cv2.destroyAllWindows()
```

In [ ]:
```
print(faces)
```

In [ ]:
```
face_classifier = cv2.CascadeClassifier('stats/Haarcascades/haarcascade_fro
image=cv2.imread('stats\image_examples/Modi.jpg')
image=cv2.resize(img,(500,500))
gray=cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
faces=face_classifier.detectMultiScale(gray,1.05,5)
if faces is ():
    print("No faces found")
for(x,y,w,h) in faces:
      cv2.rectangle(image,(x,y),(x+w,y+h),(0,255,100),1)
cv2.imshow('Face Detection',image)
cv2.waitKey()
cv2.destroyAllWindows()
```

Face & Eye Detection

In [ ]:
```
face_classifier = cv2.CascadeClassifier('stats/Haarcascades/haarcascade_fro
eye_classifer=cv2.CascadeClassifier('stats/Haarcascades/haarcascade_eye.xml
image=cv2.imread('stats\image_examples/Modi.jpg')
image=cv2.resize(img,(500,500))
gray=cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
faces=face_classifier.detectMultiScale(gray, 1.3,5)
if faces is ():
    print("No Face found")
for (x,y,w,h) in faces:
    cv2.rectangle(resized_image,(x,y),(x+w,y+h),(250,0,0),2)
    roi_gray =gray[y:y+h, x:x+w]
    roi_color=resized_image[y:y+h, x:x+w]
    eyes=eye_classifer.detectMultiScale(roi_gray)
    for(ex,ey,ew,eh) in eyes:
        cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)
cv2.imshow("img",resized_image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Capture a Video

In [ ]:
```
video =cv2.VideoCapture(0)
while True:
    check, frame= video.read()
    gray=cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
    cv2.imshow('Video',gray)
    if cv2.waitKey(1)==ord('q'):
        break
video.release()
cv2.destroyAllWindows()
```

In [ ]:
```
face_cascade=cv2.CascadeClassifier('stats/Haarcascades/haarcascade_frontalf
eye_cascade=cv2.CascadeClassifier('stats/Haarcascades/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),(250,0,0),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
video =cv2.VideoCapture(0)
while True:
    check, frame= video.read()
    gray=cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
    canvas=detect(gray,frame)
    cv2.imshow('Video',canvas)
    if cv2.waitKey(1)==ord('q'):
        break
video.release()
cv2.destroyAllWindows()
```

Pedistrian Detection

In [ ]:
```
body_classifier=cv2.CascadeClassifier('stats/Haarcascades/haarcascade_fullb
cap=cv2.VideoCapture('stats/image_examples/walking.avi')
while cap.isOpened():
    check,frame=cap.read()
    gray=cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
    bodies=body_classifier.detectMultiScale(gray,1.2,3)
    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('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

In [ ]:
```
import time
import cv2
car_classifier=cv2.CascadeClassifier('stats\Haarcascades\haarcascade_car.xm
cap=cv2.VideoCapture('stats/image_examples/cars.avi')
while cap.isOpened():
    check,frame=cap.read()
    gray=cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
    cars=car_classifier.detectMultiScale(gray, 1.8,2)
    for (x,y,w,h) in cars:
        cv2.rectangle(frame,(x,y),(x+w,y+h),(0,355,355),2)
        cv2.imshow('Cars',frame)
        if cv2.waitKey(1)==ord('q'):
            break
cap.release()
cv2.destroyAllWindows()
```

In [ ]:

In [ ]: