import numpy as np

import cv2

# reading image

image = cv2.imread('mealso.jpg', 1)

# making image smaller to fit

img = cv2.resize(image, (0,0), fx=.2, fy=.2)

# creating gray scale image

gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

# defining haarcascades to use

path = "haarcascade\_frontalface\_default.xml"

path\_to\_eyes = "haarcascade\_eye.xml"

face\_cascade = cv2.CascadeClassifier(path)

# finding faces

faces = face\_cascade.detectMultiScale(gray, scaleFactor=1.05, minNeighbors=5, minSize=(40,40))

for (x, y, w, h) in faces:

# drawing rectangle around face

cv2.rectangle(img, (x,y), (x+w, y+h), (0,0,255), 2)

# labeling image

cv2.putText(img, 'This is me', (x,y), cv2.FONT\_HERSHEY\_SIMPLEX, 0.6, (0,0,0), 2)

eye\_cascade = cv2.CascadeClassifier(path\_to\_eyes)

# defining where to look for eyes

gray\_roi = gray[y:y+h, x:x+w]

# finding eyes

eyes = eye\_cascade.detectMultiScale(gray\_roi)

color = img[y:y+h, x:x+w]

# transforming circles to the center

for (ex, ey, ew, eh) in eyes:

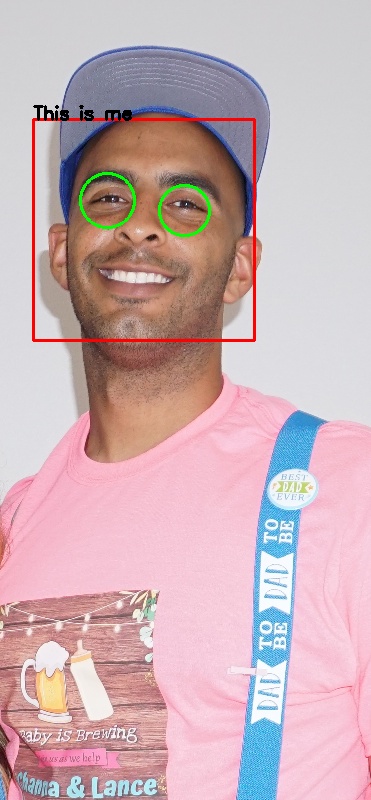
# need the radius of the circle

radius = eh//2

eye\_x = int(ex+.5\*ew)

eye\_y = int(ey+.5\*ey)

cv2.circle(color, (eye\_x,eye\_y), radius, (0,255,0), 2)

# saving and showing image

cv2.imwrite('ThisIsMe.jpg', img)

cv2.imshow('image', img)

cv2.waitKey(0)

cv2.destroyAllWindows()