import cv2

import sys

cascPath = sys.argv[1]

faceCascade = cv2.CascadeClassifier(cascPath)

video\_capture = cv2.VideoCapture(0)

while True:

# Capture frame-by-frame

ret, frame = video\_capture.read()

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

faces = faceCascade.detectMultiScale(

gray,

scaleFactor=1.1,

minNeighbors=5,

minSize=(30, 30),

flags=cv2.cv.CV\_HAAR\_SCALE\_IMAGE

)

# Draw a rectangle around the faces

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

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

# Display the resulting frame

cv2.imshow('Video', frame)

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

break

# When everything is done, release the capture

video\_capture.release()

cv2.destroyAllWindows()

import cv2

import sys

cascPath = sys.argv[1]

faceCascade = cv2.CascadeClassifier(cascPath)

video\_capture = cv2.VideoCapture(0)

while True:

# Capture frame-by-frame

ret, frame = video\_capture.read()

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

faces = faceCascade.detectMultiScale(

gray,

scaleFactor=1.1,

minNeighbors=5,

minSize=(30, 30),

flags=cv2.cv.CV\_HAAR\_SCALE\_IMAGE

)

# Draw a rectangle around the faces

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

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

# Display the resulting frame

cv2.imshow('Video', frame)

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

break

# When everything is done, release the capture

video\_capture.release()

cv2.destroyAllWindows()