

```
#!pip install dlib
```

```
import numpy as np
import cv2
from google.colab.patches import cv2_imshow
import dlib
import os
from imutils import face_utils
from google.colab import drive
```

```
drive.mount('/content/drive')
```

↳ Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_

```
path = '/content/drive/MyDrive/test_opencv'
# Hiển thị tất cả thư mục hoặc file
os.listdir(path)
```

↳ ['face2.jpg',
'face.jpg',
'res10_300x300_ssd_iter_140000_fp16.caffemodel',
'deploy.prototxt']

```
# Tải ảnh đầu vào
img = cv2.imread(path + '/face.jpg')
```

```
# Hiển thị ảnh gốc
cv2_imshow(img)
```

↳



```
# Tải mô hình đã được huấn luyện từ trước
net = cv2.dnn.readNetFromCaffe('/content/drive/MyDrive/test_opencv/models/deploy.prototxt.txt', '/content/drive/MyDrive,
```

```
# Chuẩn bị dữ liệu đầu vào
```

```
# img: Đây là ảnh đầu vào mà bạn muốn nhận dạng khuôn mặt. Trong trường hợp của bạn, bạn đã đọc ảnh từ tệp "face.png".
```

```

# 1.0: Đây là tỷ lệ co giãn cho ảnh. Trong trường hợp này, ảnh sẽ không bị co giãn hoặc mở rộng, và giữ nguyên kích thước
# (300, 300): Đây là kích thước mà mô hình yêu cầu cho ảnh đầu vào. Mô hình mà bạn đang sử dụng mong muốn ảnh có kích thước
# (104, 177, 123): Đây là giá trị trung bình màu sắc được trừ đi từ mỗi pixel của ảnh. Điều này thường được sử dụng để
# swapRB=False: Đây là một cờ để xác định xem có cần hoán đổi các kênh màu đỏ và xanh (Red-Blue) trong ảnh hay không. Tr
blob = cv2.dnn.blobFromImage(img, 1.0, (300, 300), (104, 177, 123), swapRB=False)

# Đặt dữ liệu đầu vào cho mạng
net.setInput(blob)

# Chạy mạng để phát hiện khuôn mặt
faces = net.forward()

# Lấy kích thước của ảnh đầu vào
h = img.shape[0]
w = img.shape[1]

# In thông tin
print(faces.shape)
print(faces[0, 0, 0, ])

# Duyệt từng khuôn mặt đã được phát hiện
for i in range(0, faces.shape[2]):
    confidence = faces[0,0,i,2]
    # Kiểm tra nếu mặt có độ tin cậy là 0.5
    if confidence>0.5:
        # Trích xuất tọa độ
        print(faces[0, 0, i, 3:7])
        startx = int(faces[0, 0, i, 3]*w)
        starty = int(faces[0, 0, i, 4]*h)
        endx = int(faces[0, 0, i, 5]*w)
        endy = int(faces[0, 0, i, 6]*h)
        print(startx, starty, endx, endy)

# Vẽ hình chữ nhật xung quanh khuôn mặt đã phát hiện
cv2.rectangle(img, (startx, starty), (endx, endy), (0, 255, 0),)

# Hiển thị độ tin cậy
text = 'Face: {:.2f}%'.format(confidence*100);
cv2.putText(img, text, (startx, starty-10), cv2.FONT_HERSHEY_PLAIN, 1, (255,255,255),)

# Hiển thị ảnh gốc
cv2.imshow(img)

```

```
↵ (1, 1, 200, 7)
  [0.          1.          0.9999244  0.25330234  0.38538244  0.4949335
    0.79577124]
  [0.25330234  0.38538244  0.4949335  0.79577124]
  162 184 316 381
```

