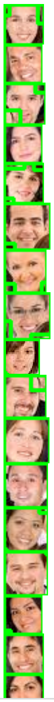
```
pip install face-recognition
```

```
Requirement already satisfied: face-recognition in /usr/local/lib/python3.10/dist-packages (1.3.0)
Requirement already satisfied: face-recognition-models>=0.3.0 in /usr/local/lib/python3.10/dist-packages (from face-recognition) (0.3.0)
Requirement already satisfied: Click>=6.0 in /usr/local/lib/python3.10/dist-packages (from face-recognition) (8.1.3)
Requirement already satisfied: dlib>=19.7 in /usr/local/lib/python3.10/dist-packages (from face-recognition) (19.24.2)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from face-recognition) (1.22.4)
Requirement already satisfied: Pillow in /usr/local/lib/python3.10/dist-packages (from face-recognition) (8.4.0)
```

```
import cv2
import numpy as np
from google.colab.patches import cv2_imshow

# Load some pre-trained data on face frontal from opencv (haar cascade algorithm)
trained_face_data = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')

# Choose an image to detect faces in
img = cv2.imread('/content/drive/MyDrive/Group.jpeg')

# Must convert to greyscale
grayscaled_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# Detect Faces
face_coordinates = trained_face_data.detectMultiScale(grayscaled_img)

img_crop = []

# Draw rectangles around the faces
for (x, y, w, h) in face_coordinates:
    cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)
    img_crop.append(img[y:y + h, x:x + w])

for counter, cropped in enumerate(img_crop):
    cv2_imshow(cropped)
    cv2.imwrite("pose_result_{}.png".format(counter), cropped)
    cv2.waitKey(0)
```



✓ 2s completed at 4:26 PM