

```

import cv2
import os
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.models import load_model
from tensorflow.keras.applications.mobilenet_v2 import preprocess_input
import numpy as np
from google.colab.patches import cv2_imshow

faceCascade = cv2.CascadeClassifier("haarcascade_frontalface_alt2.xml")
model = load_model("mask_recog.h5")

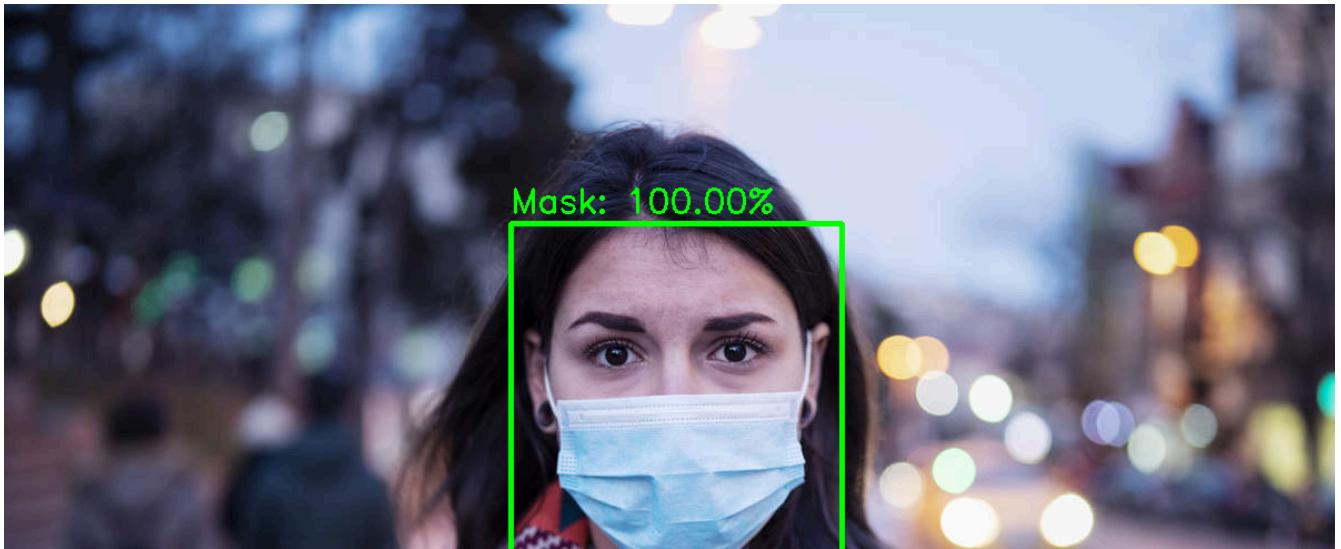
def face_mask_detector(frame):
    # frame = cv2.imread(fileName)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(gray,
                                         scaleFactor=1.1,
                                         minNeighbors=5,
                                         minSize=(60, 60),
                                         flags=cv2.CASCADE_SCALE_IMAGE)

    faces_list=[]
    preds=[]
    for (x, y, w, h) in faces:
        face_frame = frame[y:y+h,x:x+w]
        face_frame = cv2.cvtColor(face_frame, cv2.COLOR_BGR2RGB)
        face_frame = cv2.resize(face_frame, (224, 224))
        face_frame = img_to_array(face_frame)
        face_frame = np.expand_dims(face_frame, axis=0)
        face_frame = preprocess_input(face_frame)
        faces_list.append(face_frame)
    if len(faces_list)>0:
        preds = model.predict(faces_list)
    for pred in preds:
        (mask, withoutMask) = pred
    label = "Mask" if mask > withoutMask else "No Mask"
    color = (0, 255, 0) if label == "Mask" else (0, 0, 255)
    label = "{}: {:.2f}%".format(label, max(mask, withoutMask) * 100)
    cv2.putText(frame, label, (x, y- 10),
               cv2.FONT_HERSHEY_SIMPLEX, 1, color, 2)

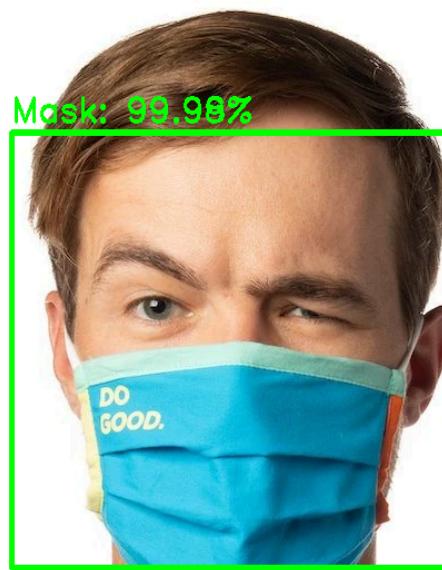
    cv2.rectangle(frame, (x, y), (x + w, y + h),color , 3)
# cv2_imshow(frame)
    return frame

input_image = cv2.imread("image.jpg")
output = face_mask_detector(input_image)
cv2_imshow(output)

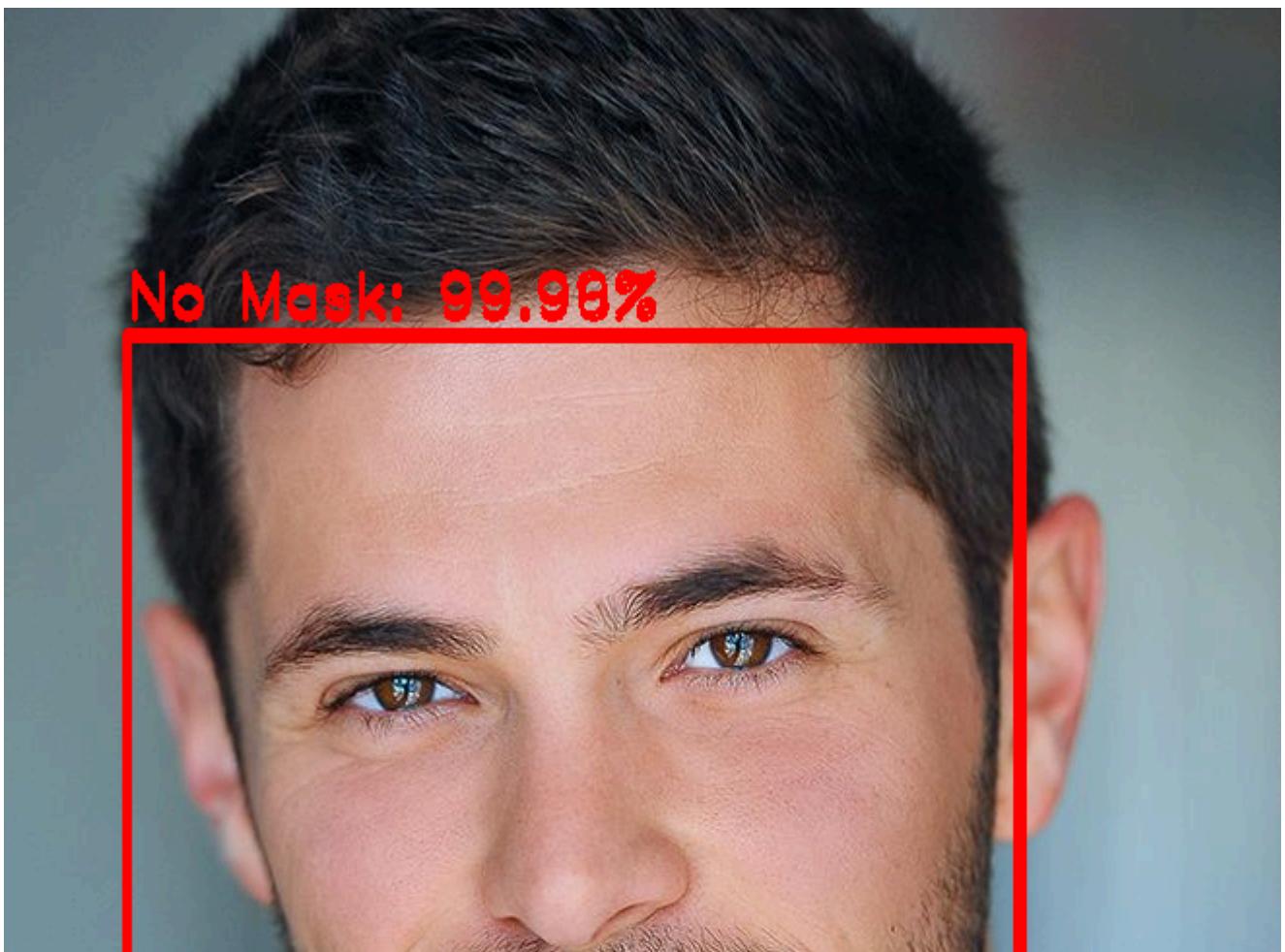
```



```
input_image = cv2.imread("image2.png")
output = face_mask_detector(input_image)
cv2_imshow(output)
```



```
input_image = cv2.imread("image3.png")
output = face_mask_detector(input_image)
cv2_imshow(output)
```



```
input_image = cv2.imread("image4.jpg")
output = face_mask_detector(input_image)
cv2.imshow(output)
```



```
cap = cv2.VideoCapture('video.mp4')
ret, frame = cap.read()
frame_height, frame_width, _ = frame.shape
out = cv2.VideoWriter('output.avi', cv2.VideoWriter_fourcc('M','J','P','G'), 10, (frame_wid
print("Processing Video...")
```

```
while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        out.release()
        break
    output = face_mask_detector(frame)
    out.write(output)
out.release()
print("Done processing video")
```

Processing Video...
Done processing video