```
import cv2
import numpy as np
from tensorflow.keras.models import load_model

---------------------------------------------------------------------
-----
ModuleNotFoundError                       Traceback (most recent call
last)
~\AppData\Local\Temp/ipykernel_10880/4024556568.py in <module>
      1 import cv2
      2 import numpy as np
----> 3 from tensorflow.keras.models import load_model

ModuleNotFoundError: No module named 'tensorflow'

# Load the trained model
path_to_model
='/home/Documents/Dissertation/driver-drive/Models/Predefine_Architect
ure_VGG16_Model.h5'
model = load_model(path_to_model)

def preprocess(frame):
    # Resize the frame to 180x180 pixels
    resized_frame = cv2.resize(frame, (180, 180))
    # Convert the frame to grayscale
    gray_frame = cv2.cvtColor(resized_frame, cv2.COLOR_BGR2GRAY)
    # Add a third color channel to the frame
    processed_frame = cv2.merge([gray_frame, gray_frame, gray_frame])
    # Expand the dimensions of the frame to match the input shape of
the model
    processed_frame = np.expand_dims(processed_frame, axis=0)
    return processed_frame

# Create a VideoCapture object to capture frames from the camera
cam= cv2.VideoCapture(0)

while True:
    # Capture a frame from the camera
    ret, frame = cam.read()

    # Check if a frame was successfully captured
    if not ret:
        print("Failed to grab frame")
        break
    # Our operations on the frame come here
    # gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    # # Display the captured frame
    cv2.imshow('Test', frame)

    k= cv2.waitKey(1)
```

```
    # Wait for a key press to exit
    if k%256 == 27:
        print('escape hit, closing the app')
        break

    # Preprocess the frame
    processed_frame = preprocess(frame)

    # Pass the preprocessed frame through the model to get the
drowsiness detection result
    result = model.predict(processed_frame)

    # Analyze the output of the model and take appropriate actions
    if result > 0.5:
        # Driver is drowsy
        # Display an alert message on the screen
        cv2.putText(frame, "Drowsy Driver Detected!", (10, 50),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)
        # Play an alarm sound to alert the driver
        playsound.playsound('alarm.wav')

    # Display the output frame with the detection result
    cv2.imshow('Output Frame', frame)

    # Exit the program if 'q' is pressed
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break


# Release the capture and destroy all windows
cam.release()
# cv2.destroyAllWindows()
```