

Performance Analysis of the Drowsiness Detection Module, EMPROVE

May 26, 2023

0.1 EMPROVE - An Employee Productivity System using Machine Learning

Project Members

- Alen George
- Joel Raju
- Roshan Roy
- San Baby Francis

```
[1]: #@title #### Install the necessary packages
!pip install mediapipe opencv-python numpy scikit-learn
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting mediapipe
  Downloading mediapipe-0.10.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (33.9 MB)
33.9/33.9 MB
43.2 MB/s eta 0:00:00
Requirement already satisfied: opencv-python in /usr/local/lib/python3.10/dist-packages (4.7.0.72)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (1.22.4)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (1.2.2)
Requirement already satisfied: absl-py in /usr/local/lib/python3.10/dist-packages (from mediapipe) (1.4.0)
Requirement already satisfied: attrs>=19.1.0 in /usr/local/lib/python3.10/dist-packages (from mediapipe) (23.1.0)
Requirement already satisfied: flatbuffers>=2.0 in /usr/local/lib/python3.10/dist-packages (from mediapipe) (23.3.3)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (from mediapipe) (3.7.1)
Requirement already satisfied: opencv-contrib-python in /usr/local/lib/python3.10/dist-packages (from mediapipe) (4.7.0.72)
Requirement already satisfied: protobuf<4,>=3.11 in /usr/local/lib/python3.10/dist-packages (from mediapipe) (3.20.3)
Collecting sounddevice>=0.4.4 (from mediapipe)
```

```
  Downloading sounddevice-0.4.6-py3-none-any.whl (31 kB)
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.10/dist-
packages (from scikit-learn) (1.10.1)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-
packages (from scikit-learn) (1.2.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/usr/local/lib/python3.10/dist-packages (from scikit-learn) (3.1.0)
Requirement already satisfied: CFFI>=1.0 in /usr/local/lib/python3.10/dist-
packages (from sounddevice>=0.4.4->mediapipe) (1.15.1)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->mediapipe) (1.0.7)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-
packages (from matplotlib->mediapipe) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->mediapipe) (4.39.3)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->mediapipe) (1.4.4)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->mediapipe) (23.1)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-
packages (from matplotlib->mediapipe) (8.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->mediapipe) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->mediapipe) (2.8.2)
Requirement already satisfied: pycparser in /usr/local/lib/python3.10/dist-
packages (from CFFI>=1.0->sounddevice>=0.4.4->mediapipe) (2.21)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-
packages (from python-dateutil>=2.7->matplotlib->mediapipe) (1.16.0)
Installing collected packages: sounddevice, mediapipe
Successfully installed mediapipe-0.10.0 sounddevice-0.4.6
```

[2]: *#@title ##### Import the necessary packages*

```
import mediapipe as mp
import cv2
import numpy as np
from sklearn.metrics import mean_absolute_error
from google.colab.patches import cv2_imshow
```

[3]: *#@title ##### Load and initialize the MediaPipe Face Mesh model*

```
# Load the MediaPipe Face Mesh model
mp_face_mesh = mp.solutions.face_mesh

# Initialize the Face Mesh model
face_mesh = mp_face_mesh.FaceMesh(static_image_mode=False, max_num_faces=1)
```

```
[4]: #@title ##### Function to calculate the Euclidean distance between two keypoints
```

```
def calculate_distance(x1, y1, x2, y2):
    return np.sqrt((x1 - x2) ** 2 + (y1 - y2) ** 2)
```

```
[5]: #@title ##### Function to calculate the Eye Aspect Ratio and Mouth Aspect Ratio
```

```
def eyeAspectRatio(p1, p2, p3, p4, p5, p6):
    A = calculate_distance(p2[0], p2[1], p6[0], p6[1])
    B = calculate_distance(p3[0], p3[1], p5[0], p5[1])
    C = calculate_distance(p1[0], p1[1], p4[0], p4[1])
    return ((A + B) / (2.0) * C)

def mouthAspectRatio(p1, p2, p3, p4, p5, p6):
    A = calculate_distance(p2[0], p2[1], p6[0], p6[1])
    B = calculate_distance(p3[0], p3[1], p5[0], p5[1])
    C = calculate_distance(p1[0], p1[1], p4[0], p4[1])
    return ((A + B) / (2.0) * C)
```

```
[6]: #@title ##### Load an image for testing
```

```
image_path = "/content/drowsy.jpg"
image = cv2.imread(image_path)
image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
cv2_imshow(image)
```



```
[7]: #@title ##### Process the image with the Face Mesh model
```

```
results = face_mesh.process(image_rgb)
```

```
[8]: #@title ##### Extract the facial keypoints and draw the face mesh on the image
```

```
if results.multi_face_landmarks:  
    face_landmarks = results.multi_face_landmarks[0]  
    keypoints = []  
    for landmark in face_landmarks.landmark:  
        keypoints.append((landmark.x, landmark.y))  
    # Draw the face mesh on the image  
    mp.solutions.drawing_utils.draw_landmarks(  
        image,  
        face_landmarks,  
        mp_face_mesh.FACEMESH_TESSELATION,  
        landmark_drawing_spec=mp.solutions.drawing_utils.DrawingSpec(  
            color=(0, 255, 0), thickness=2, circle_radius=2),  
        connection_drawing_spec=mp.solutions.drawing_utils.DrawingSpec(  
            color=(0, 255, 0), thickness=2)  
)
```

```
else:  
    print("No face detected.")
```

[9]: *#@title ##### Display the image with keypoints*

```
cv2_imshow(image)
```



[31]: *#@title ##### Provide the ground truth keypoints of the image*

```
[32]: #@title ##### Calculate the mean squared error between the detected keypoints  
and ground truth keypoints
```

```
mse = mean_absolute_error(ground_truth_keypoints, keypoints)  
# Print the mean squared error  
print("Mean Absolute Error:", mse)
```

Mean Absolute Error: 2.4626068376068373

```
[33]: #@title ##### Calculate EAR and MAR from facial keypoints
```

```
MAR = mouthAspectRatio(keypoints[61], keypoints[37], keypoints[267],  
keypoints[291], keypoints[314], keypoints[84])  
L_EAR = eyeAspectRatio(keypoints[33], keypoints[160], keypoints[158],  
keypoints[133], keypoints[153], keypoints[144])  
R_EAR = eyeAspectRatio(keypoints[362], keypoints[385], keypoints[387],  
keypoints[263], keypoints[373], keypoints[380])
```

```
[34]: #@title ##### Determine whether the person is drowsy
```

```
if (MAR > 6000 or (L_EAR < 250 and R_EAR < 250)):  
    print("Drowsy")  
else:  
    print("Not Drowsy")
```

Drowsy