

Descrição do projeto

O projeto consiste em um programa que coleta métricas do usuário, como:

- Identidade
- Emoções
- Nível de atenção (olhou para o lado, para baixo ou está com os olhos fechados)

As estatísticas coletadas são salvas em um arquivo no formato json.

Relevância

Os casos de uso do projeto foram pensados para quando se deseja medir as reações e o nível de atenção do usuário quando este está vendo a tela do computador. Situações de exemplo seriam:

- Pesquisas de marketing
 - Por exemplo, para coletar dados de análise das reações do usuário quando assistem a uma propaganda ou a novos lançamentos de um produto
- Palestras e aulas
 - Por exemplo, para identificar usuários que estão muito desatentos e poder tomar medidas para ajudá-los

Implementação

Os requisitos do projeto foram implementados da seguinte forma:

- Algoritmo de percepção 3D (vídeo):
 - Face Mesh da biblioteca MediaPipe
- Algoritmo de reconhecimento de padrões
 - Emotions Detector da biblioteca FER
 - Face Recognition da OpenCV
- Algoritmo de processamento de imagens
 - Equalização do histograma do canal V do HSV para melhorar o contraste e auxiliar os algoritmos a encontrar os contornos e feições da face da pessoa.

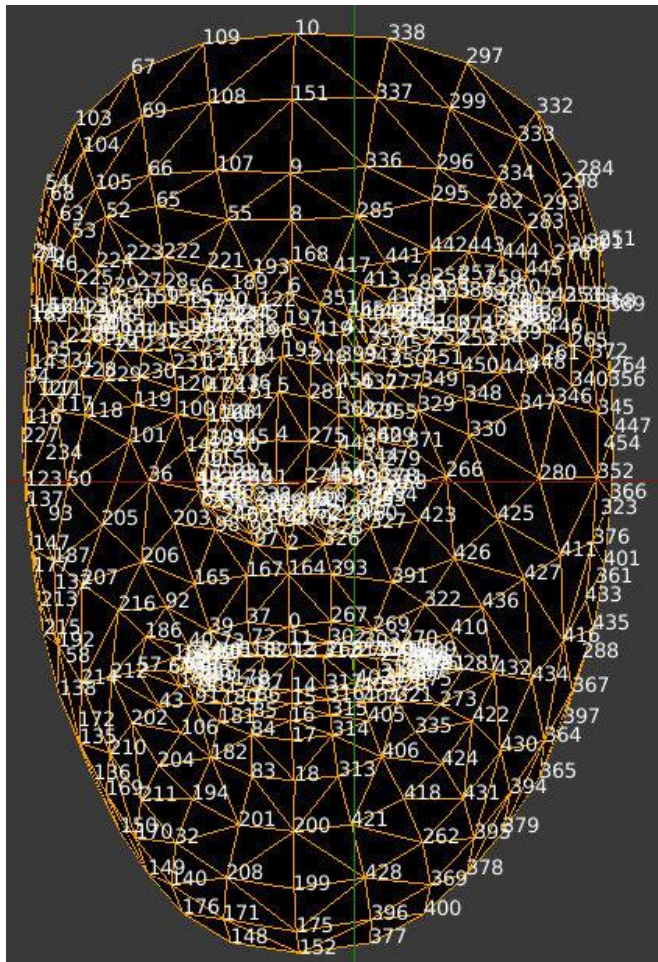
Face Mesh

O algoritmo de Face Mesh da MediaPipe (referência:

https://google.github.io/mediapipe/solutions/face_mesh.html) consiste em um mapeamento da face da pessoa, em que 468 pontos de referência 3D são estimados.

Ele emprega um modelo de Machine Learning para inferir a geometria da superfície 3D, exigindo apenas uma entrada de câmera para o cálculo. A partir destes pontos, consegue criar uma malha de face triangular.

A imagem abaixo mostra um exemplo destes pontos mapeados em uma face.



De posse do índice da posição dos pontos, é possível estimar a posição da face na tela e, por meio da distância entre pontos, determinar se os olhos de uma pessoa estão abertos ou fechados.

Emotions Detector

Para a detecção das emoções pela face do usuário, foi utilizada biblioteca FER (referência: <https://pypi.org/project/fer/>).

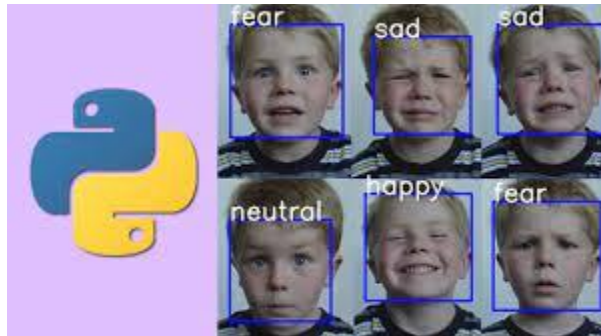
Para detectar as faces. O modelo usa o Classificador Haar Cascade da OpenCV. O modelo de detecção de emoções foi montado utilizando Redes Neurais Convolucionais (CNN).

Existem certas classes de emoções, em que determinados padrões são identificados, sendo estas listadas abaixo:

- Felicidade
- Tristeza
- Medo
- Raiva

- Surpresa
- Nojo
- Neutro

O detector retorna um dicionário com duas chaves. A primeira possui como valores as coordenadas do quadrado com a face detectada. A segunda consiste em outro dicionário com cada uma das emoções como chaves e os valores de 1 a 0 indicando a probabilidade da respectiva emoção. Existe ainda a opção de buscar apenas a emoção dominante e seu score (usada no projeto)



Face Recognition

Para reconhecer a face da pessoa em uma imagem ou vídeo é importante primeiro detectar onde ela se encontra. Isso pode ser feito por meio de um algoritmo da OpenCV chamado de Haar Cascade Classifier (referência: https://docs.opencv.org/3.3.0/d7/d8b/tutorial_py_face_detection.html)

Em seguida, para o reconhecimento facial é fornecer imagens de comparação. Para isso, o usuário pode fazer um cadastro, em que 20 fotos são tiradas e salvas em uma pasta com o nome da pessoa (passado como argumento no terminal).

Depois disso, é preciso treinar o modelo. O módulo de reconhecimento utilizado é o LBPHFaceRecognizer, da OpenCV (referência:

https://docs.opencv.org/3.4/df/d25/classcv_1_1face_1_1LBPHFaceRecognizer.html). Ele possui uma função chamada train, que recebe as imagens e os ids dos usuários para o treinamento do modelo. Como resultado, um arquivo chamado de trainer.yml é salvo.

Por fim, o modelo implementado irá fazer uma previsão da identidade da pessoa, com base na imagem da câmera e do modelo implementado. A porcentagem de certeza da previsão também é dada e se for muito baixa, a face da pessoa é classificada como desconhecida.