**VIETNAM NATIONAL UNIVERSITY HCMC**

**UNIVERSITY OF INFORMATION TECHNOLOGY**

A blue logo with text

Description automatically generated

**PROJECT REPORT**

**TOPIC:**

**Phát hiện tài xế ngủ gật trên xe và đưa ra âm thanh cảnh báo**

**Instructors: Ths Đỗ Văn Tiến**

**Team members: Nguyễn Vũ Hoàng Long 21520058**

**Phạm Quốc Anh Khoa 21520296**

**Class: Xử lý ảnh và ứng dụng**

***Ho Chi Minh City, November 10, 2023***

[**I. Introduction 3**](#_heading=h.8m2fg1doo7bw)

[**III. Dataset 4**](#_heading=h.2s8eyo1)

[1. Mô Tả Tổng Quan về Dataset 4](#_heading=h.xs3kbp4n56h7)

[2. Quy trình 4](#_heading=h.4yvgehwbr4gv)

[3. Tiền Xử Lý Dữ Liệu. 4](#_heading=h.yf10idanl6j3)

[**IV. Method 4**](#_heading=h.17dp8vu)

[1. Experiment 4](#_heading=h.i0q4xtvy551q)

[a. YOLOv8 5](#_heading=h.29qgdk23jzka)

[**V. References 7**](#_heading=h.g6e1i0a5bk3n)

**Thông tin đồ án**

Các thông tin bao gồm slide thuyết trình, file báo cáo và source code đều được cập nhật ở link github bên dưới

https://github.com/binhere/Drowsiness-Detection

**Bảng phân công công việc**

|  |  |  |
| --- | --- | --- |
| Công việc | Phạm Quốc Anh Khoa  21520296 | Nguyễn Vũ Hoàng Long  21520058 |
| Tìm dataset | x | x |
| Tìm hiểu dataset, tiển xử lý |  | x |
| Tìm hiểu YOLOv8 | x | x |
| Train model YOLOv8 |  | x |
| Xây dựng app bằng Tkinter |  | x |
| Phát triển thêm cho app với âm thanh cảnh báo | x |  |
| Phát hiện và khắc phục nhược điểm YOLOv8 | x |  |
| Tìm hiểu Mediapile | x | x |
| Kết hợp 2 model | x |  |
| Làm slide báo cáo | x | x |
| Làm báo cáo word | x | x |
| Đảm bảo deadline | x | x |
| Họp nhóm đầy đủ tối thiểu 2 lần | x | x |

# Introduction

Theo báo cáo của Tổng cục thống kê, trong năm 2019, trên địa bàn cả nước xảy ra 17.626 vụ tai nạn giao thông. Theo thống kê khác của Ủy ban An toàn giao thông quốc gia, mỗi năm có hơn 6000 người chết từ những vụ tai nạn giao thông liên quan tới tài xế ngủ gật. Nhận thấy rằng nguyên nhân do thiếu ngủ chiếm tới 30% tổng các vụ giao thông trong một năm. Với mong muốn hạn chế những ca tai nạn giao thông do tài xế buồn ngủ gây ra, nhóm đã huấn luyện một mô hình máy học nhằm phát hiện những trường hợp người lái xe có biểu hiện buồn ngủ và rồi đưa ra âm thanh cảnh báo.

# Dataset

## Mô Tả Tổng Quan về Dataset

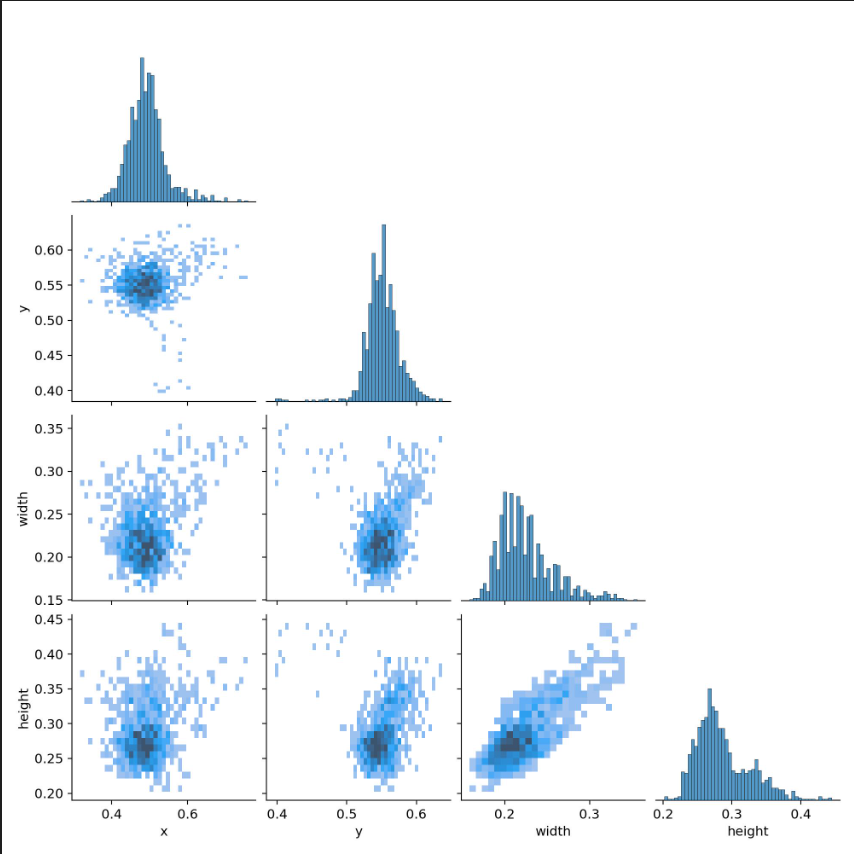
Đồ án thực hiện trên tệp dataset Drowsiness Detection Computer Vision Project (<https://universe.roboflow.com/augmented-startups/drowsiness-detection-cntmz>)

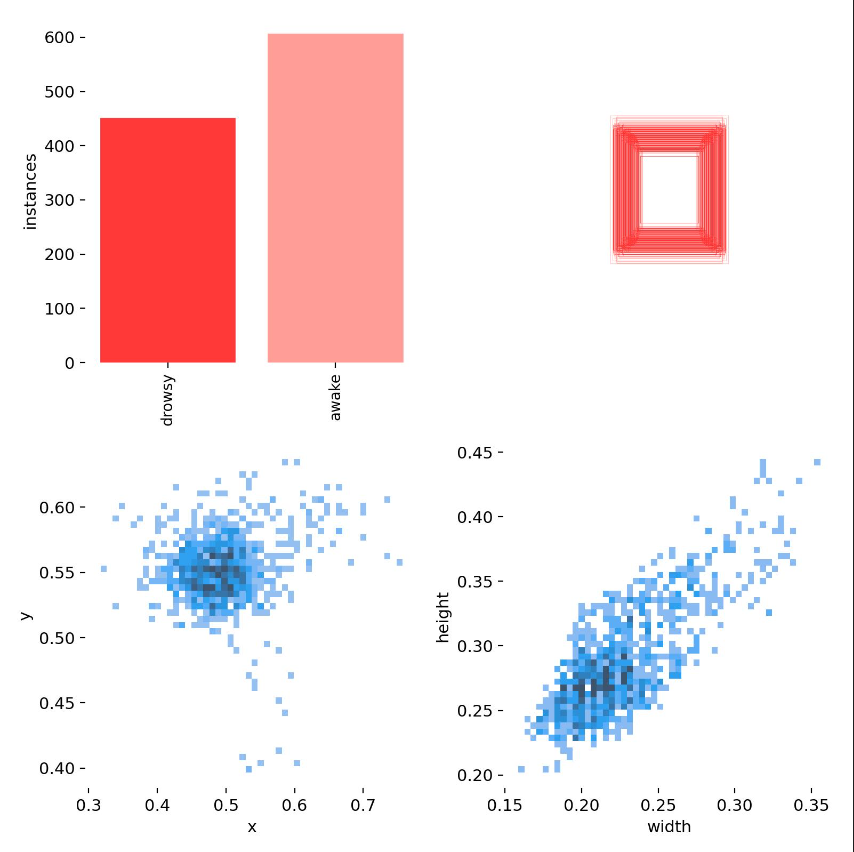
Tập dữ liệu thành ba bộ:

* Train: 1.056 samples
* Valid: 103 samples
* Test: 71 samples

## Tiền Xử Lý Dữ Liệu.

Đối với tập dữ liệu của Drowsiness Detection Computer Vision Project các bức ảnh đã được xử lý thông tin huấn luyện gồm nhãn và tọa độ bounding box.





Tuy nhiên, các dữ liệu này được lưu ở dạng file csv cho nên cần chuyển đổi sang định dạng mà mô hình YOLO có thể hiểu được.

# Duyệt qua từng dòng trong dataframe và tạo nội dung cho file YOLO cho mỗi ảnh

for index, row in annotations\_df.iterrows():

width = row['width']

height = row['height']

# Chuyển đổi class\_id thành số nguyên (1 cho 'awake' và 0 cho 'drowsy')

class\_id = 1 if row['class'] == 'awake' else 0

xmin, ymin, xmax, ymax = row['xmin'], row['ymin'], row['xmax'], row['ymax']

# Chuyển đổi tọa độ bbox thành tọa độ trung tâm và chiều rộng, chiều cao

center\_x = (xmin + xmax) / (2 \* width)

center\_y = (ymin + ymax) / (2 \* height)

bbox\_width = (xmax - xmin) / width

bbox\_height = (ymax - ymin) / height

# Chuyển đổi tọa độ bbox thành tọa độ trung tâm và chiều rộng, chiều cao

center\_x = (xmin + xmax) / (2 \* width)

center\_y = (ymin + ymax) / (2 \* height)

bbox\_width = (xmax - xmin) / width

bbox\_height = (ymax - ymin) / height

# Tạo nội dung cho file YOLO, bao gồm class\_id và các thông số bbox

yolo\_line = f"{class\_id} {center\_x} {center\_y} {bbox\_width} {bbox\_height}"

# Lưu nội dung vào file YOLO

yolo\_file\_path = os.path.join(output\_folder, f"{row['filename'].replace('.jpg', '.txt')}")

with open(yolo\_file\_path, 'w') as yolo\_file:

yolo\_file.write(yolo\_line + '\n')

# Method

## Experiment

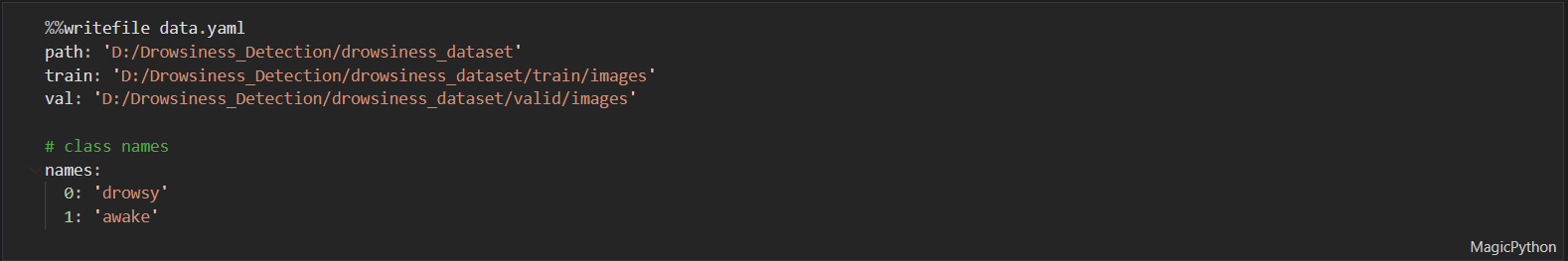
Nhóm tiếp cận bài toán bằng phương pháp Object Detection của YOLOv8. Bên cạnh đó, nhóm cũng thử một phương pháp khác tên là facial landmark detection bằng thư viện Mediapile của Google.

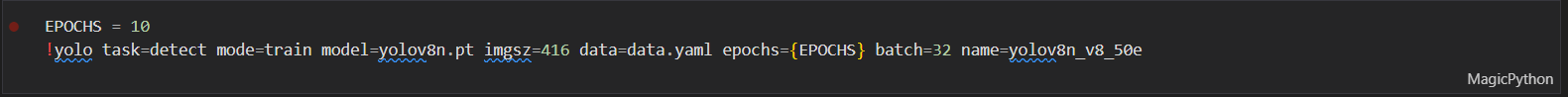
Mục đích của nhóm khi sử dụng 2 phương pháp song song nhau không chỉ để so sánh về hiệu suất, tốc độ, độ chính xác mà còn phối hợp để khắc phục những thiếu sót của mô hình.

### YOLOv8

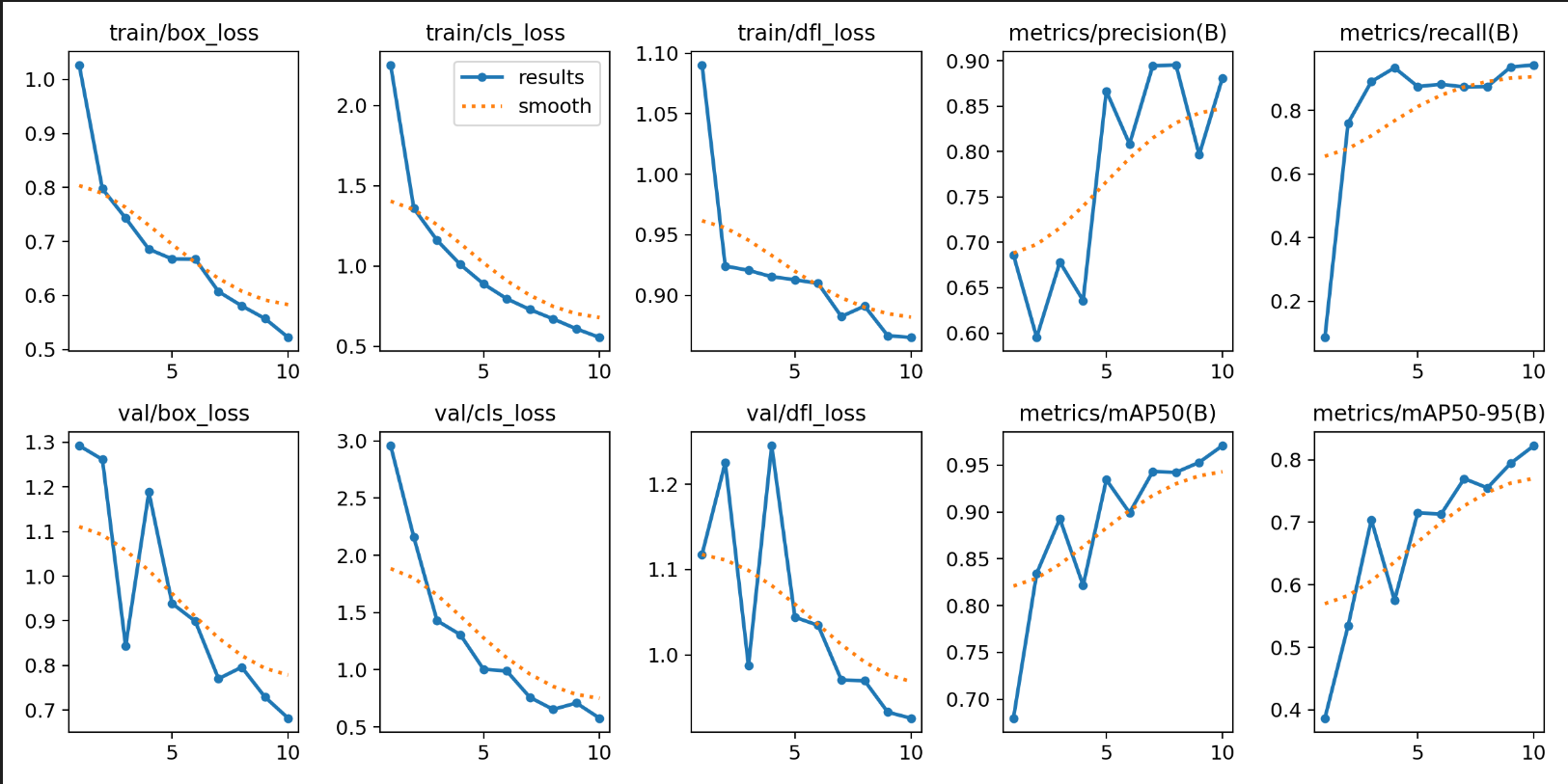
Nhận diện đối tượng là một lĩnh vực quan trọng trong thị giác máy tính, nơi mà chúng ta tập trung vào việc xác định vị trí và loại của các đối tượng trong ảnh. YOLOv8 là giải pháp công nghệ tiên tiến, đã đưa ra những đóng góp quan trọng vào lĩnh vực này. Điểm mạnh của YOLOv8 nằm ở việc sử dụng các mô hình học sâu và thuật toán hiện đại để nhận diện và phân loại các đối tượng với tốc độ cao. Nhờ vào khả năng này, YOLOv8 không chỉ giúp cải thiện trải nghiệm người dùng trong lĩnh vực nhận diện đối tượng, mà còn mang lại những tiện ích quan trọng trong các ứng dụng như nhận diện biển số xe, theo dõi đối tượng, và nhiều ứng dụng khác trong thế giới số ngày nay. Nhóm đã triển khai YOLOv8 - thư viện nhận diện đối tượng của Ultralytics, để xác định vị trí và loại của các đối tượng trong ảnh mà không cần quá trình huấn luyện lại mô hình. Nhóm đặc biệt tập trung vào việc xác định đối tượng ở các góc độ khác nhau.

Bằng việc thay đổi nhãn với 1: “awake” và 0: “drowsy” ở layer cuối trong mô hình Yolov8n đã có sẵn, nhóm đã thu được một mô hình Yolov8 tùy chỉnh phù hợp cho việc huấn luyện nhận biết người buồn ngủ trên xe.



Chọn Epochs =10, batch\_size = 32 rồi thực hiện huấn luyện mô hình   


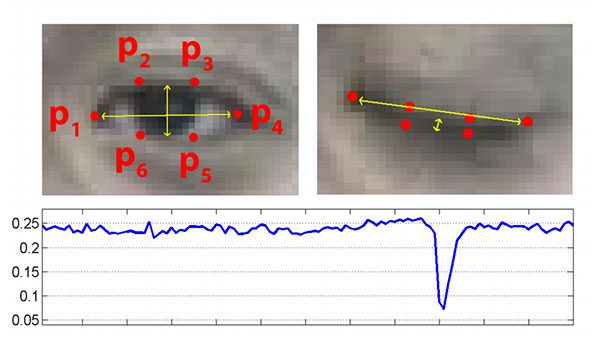
Sau khi huấn luyện, kết quả thu được



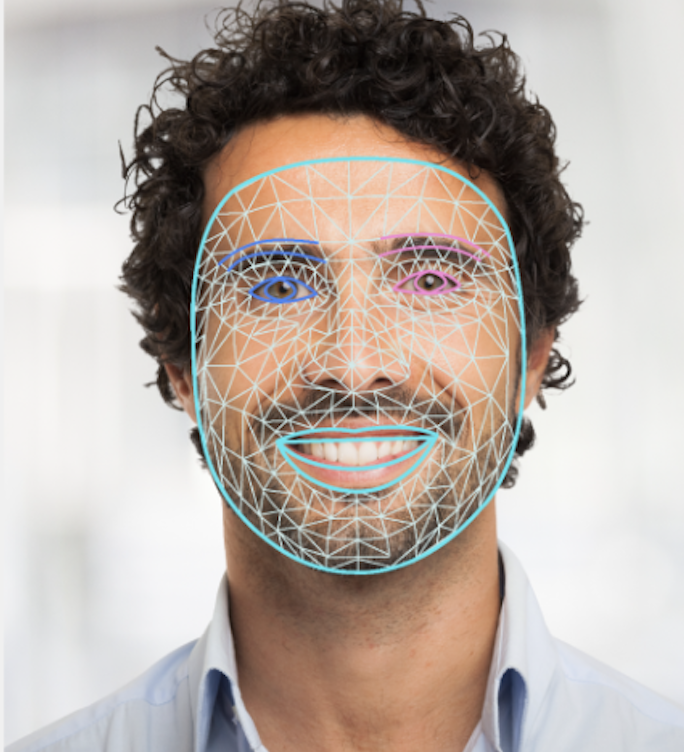
1. **Facial landmark detection - Mediapile**

Facial landmark detection là một lĩnh vực quan trọng trong xử lý ảnh, nơi mà chúng ta tập trung vào việc xác định vị trí chính xác của các điểm đặc trưng trên khuôn mặt. Mediapile là giải pháp công nghệ tiên tiến, đã đưa ra những đóng góp quan trọng vào lĩnh vực này. Điểm mạnh của Mediapile nằm ở việc sử dụng các mô hình học máy và thuật toán hiện đại để nhận diện và định vị các điểm landmark với độ chính xác cao. Nhờ vào khả năng này, Mediapile không chỉ giúp cải thiện trải nghiệm người dùng trong lĩnh vực nhận diện khuôn mặt, mà còn mang lại những tiện ích quan trọng trong các ứng dụng như nhận diện biểu cảm khuôn mặt, theo dõi chuyển động, và nhiều ứng dụng khác trong thế giới số ngày nay.

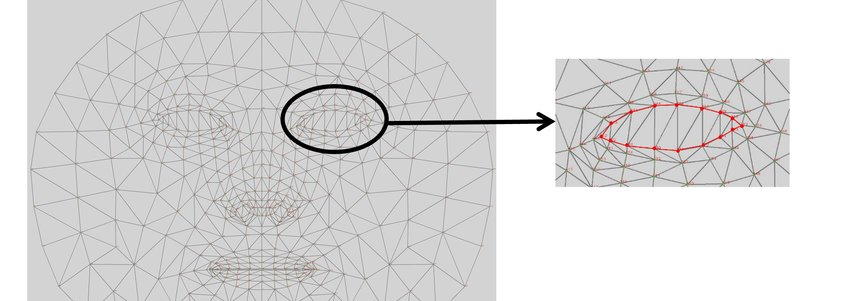
Nhóm đã triển khai Mediapile - thư viện nhận diện landmark của Google, để xác định vị trí đặc trưng trên khuôn mặt mà không cần quá trình huấn luyện lại mô hình. Nhóm đặc biệt tập trung vào việc xác định landmark ở mắt trái và phải. Sau đó, nhóm đã sử dụng thông tin này để tính toán tỉ lệ EAR (Eye Aspect Ratio) - một chỉ số quan trọng để đánh giá mức độ mở của mắt. Việc này giúp nhóm phát hiện tự động mức độ mở của mắt của tài xế và tự động phát âm thanh cảnh báo khi phát hiện rủi ro về sự mệt mỏi hoặc mất tập trung trong quá trình lái xe. Điều này không chỉ nâng cao an toàn giao thông mà còn thể hiện tính ứng dụng cao của Mediapile trong các lĩnh vực giảm sát tài xế ([Driver Monitoring Systems - DMS)](https://www.nxp.com/pages/driver-monitoring-systems-dms-and-occupant-monitoring-systems-:DRIVER-MONITORING-SYSTEMS#:~:text=The%20driving%20monitoring%20system%20includes,the%20control%20of%20the%20vehicle.)

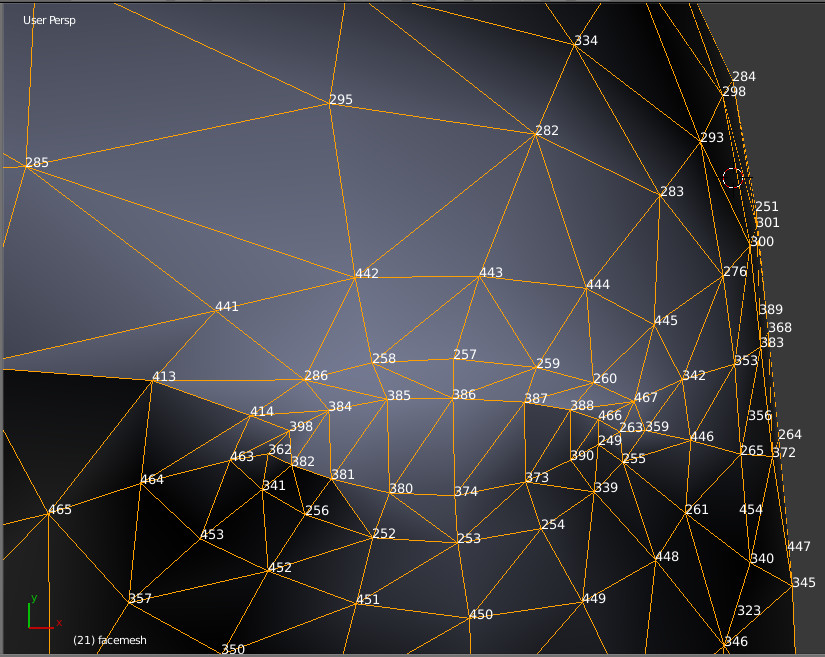


Cụ thể, sau khi áp dụng mediapile, nhóm thu được kết quả như sau

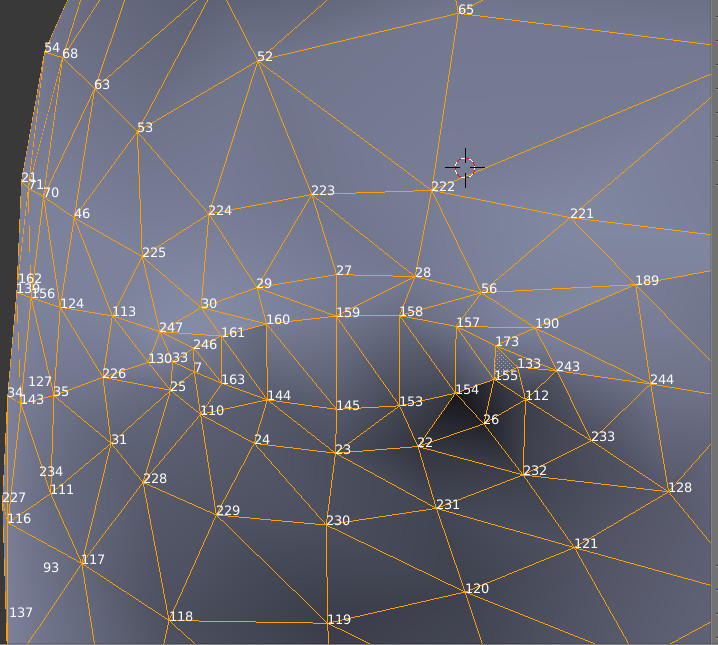


Như đã để cập, nhóm chỉ chú trọng vào những landmark thuộc mắt trái và phải, do đó nhóm cần lấy được tọa độ các điểm landmark đó



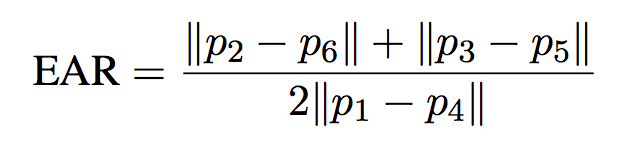


Mắt phải



Mắt trái

Sau khi đã chọn ra được những cặp tọa độ (x, y) cần thiết, nhóm tiến hành tính chỉ số EAR cho mỗi mắt theo công thức sau



Lấy tổng khoảng cách 2 cặp điểm p2, p6 và p3, p5 chia cho 2 lần khoảng cách của 2 điểm p1 và p4. Sau khi đã tính được chỉ số EAR cho mỗi mắt, ta lấy chỉ sổ trung bình cộng cho cả hai mắt. Tiếp tục, để định được mắt có độ mở như thế nào được coi là thức hoặc ngủ, ta phải tiến hành thực nghiệm. Nhóm tiến hành thực nghiệm như sau:



Sau quá trình thực nghiệm, nhóm quyết định chọn ngưỡng bằng 27 để phân biệt. Nếu EAR chỏ cả 2 mắt lớn hơn 27, ta coi mắt đang mở. Ngược lại, nếu EAR nhỏ hơn hoặc bằng 27, ta coi mắt có biệu hiện buồn ngủ (bao gồm cả mắt lim dim và mắt đã hoàn toàn khép lại)

**Nhóm thực hiện so sánh YOLOv8 và Mediapile như sau**

**YOLOv8**

Ưu:

* Nhanh, triển khai trên máy có cấu hình yếu.
* Độ chính xác tương đối cao

Nhược:

* Chi phí cho dataset cao
* Khó mở rộng trường hợp

**Mediapile**

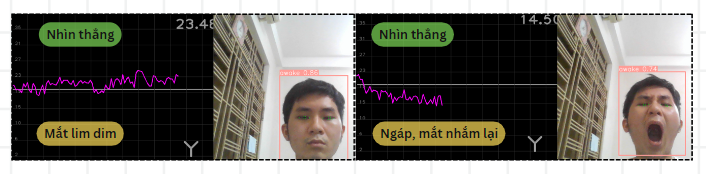
Ưu:

* Nhanh, triển khai trên máy có cấu hình yếu.
* Độ chính xác tương đối cao
* Dễ dàng mở rộng trường hợp mới
* Không cần chuẩn bị custom dataset cũng như không cần train lại

Nhược:

* Chỉ dựa vào các điểm landmark trên mặt làm cơ sở, cần phải tự đề xuất giải pháp cho các trường hợp khác nhau

Tóm lại, YOLOv8 nhanh nhưng chưa chính xác trong một vài trường hợp. Mediapile và hướng tiếp cận sử dụng ngưỡng EAR = 27 sẽ khắc phục được sai sót của YOLOv8



**Định hướng phát triển:**

Từ phương pháp nền là facial landmarks detection, nhóm có thể

* Kết hợp head pose estimation: detect hướng của gương mặt tài xế, từ đó phát hiện tài xế bị phân tâm, mất tập trung khi lái xe
* Kết hợp eyes’ direction detection: detect hướng nhìn của tài xế, từ đó phát hiện tài xế không tập trung nhìn phía về phía trước khi lái xe
* Kết hợp detect độ mở của mắt, tần xuất chớp mắt, hoặc lim dim, hoặc ngáp để đưa ra cảnh báo tài xế đang mệt mỏi cần nghỉ ngơi

# References

[**https://www.v7labs.com/blog/yolo-object-detection**](https://www.v7labs.com/blog/yolo-object-detection)

**https://universe.roboflow.com/augmented-startups/drowsiness-detection-cntmz**

**https://developers.google.com/mediapipe/solutions/vision/face\_landmarker**

**https://github.com/ManuelTS/augmentedFaceMeshIndices**

**https://docs.ultralytics.com/**