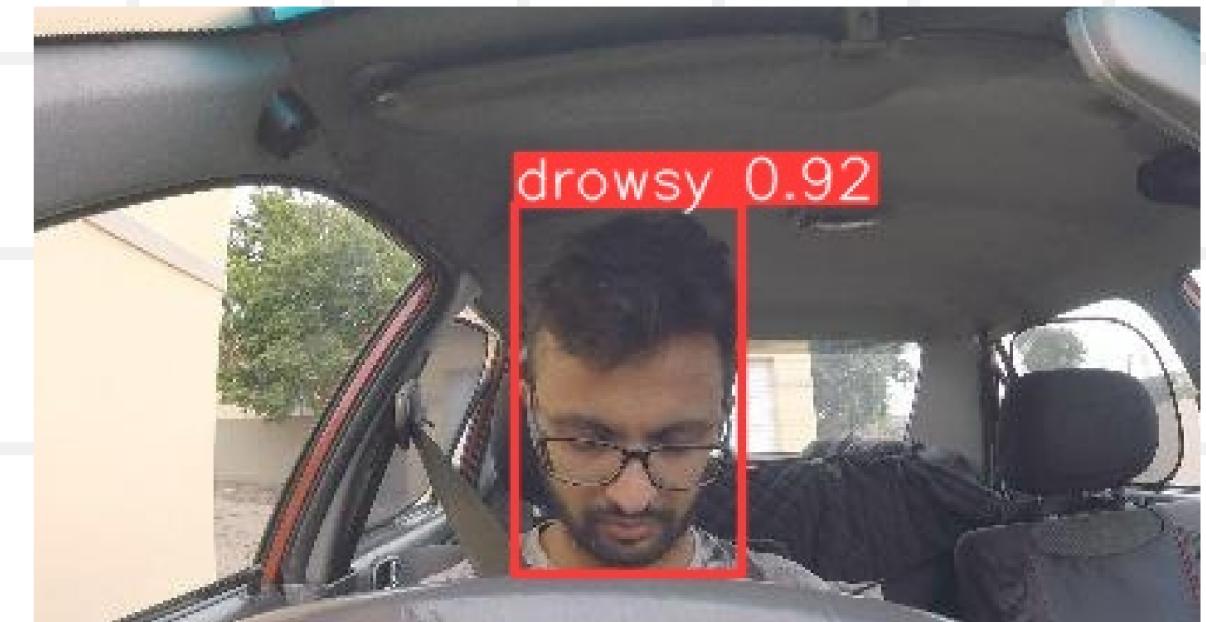


DETECT DROWSINESS IN THE CAR AND ISSUE A WARNING

Presentation by:
Nguyễn Vũ Hoàng Long- 21520058
Phạm Quốc Anh Khoa- 21520296



GIỚI THIỆU

- Bài toán
- Định hướng
- Phương pháp
- Thực hiện
- Kết Quả
- Kết luận
- Demo

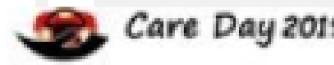


BÀI TOÁN



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

BÀI TOÁN



Care Day 2019

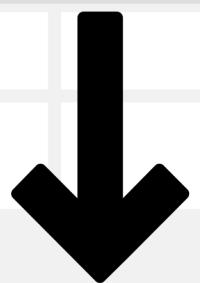
LÁI MUÔN NƠI KHÔNG VƠI NĂNG LƯỢNG

- 1 = Ít ngủ hơn **2** tiếng/1 ngày tương đương uống **3** ly bia
- 2 Ngủ gật khi lái xe gây ra **6.400** người chết hàng năm
- 3 **30%** lái xe thừa nhận có tình trạng ngủ gật sau tay lái
- 4 Nguy cơ tai nạn tăng gấp **4** lần với những tài xế lái xe ngày hôm trước đó chỉ ngủ 4-5 tiếng
- 5 **109** tỉ đô la (không bao gồm thiệt hại tài sản) mất đi hàng năm cho các vụ tai nạn liên quan đến ngủ gật khi lái xe
- 6 **21%** tổng nguyên nhân tai nạn là do ngủ gật khi lái xe

Đau Nhức &

Theo thống kê 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. Nguyên nhân do thiếu ngủ chiếm tới 30% tổng các vụ giao thông trong một năm.

ĐỊNH HƯỚNG



TỔNG QUAN

Đưa ra nhắc nhở người lái xe
khi họ có biểu hiện buồn ngủ.



HƯỚNG TIẾP CẬN

Tạo ứng dụng tích hợp vào camera
hành trình sử dụng YOLO và
Mediapile để nhận diện dấu hiệu
của sự buồn ngủ và phát âm thanh
cảnh báo

PHƯƠNG PHÁP

YOLOV8

Object Detection: detect gương mặt, tư thế
đầu của tài xế
Output: awake / drowsy + bbox

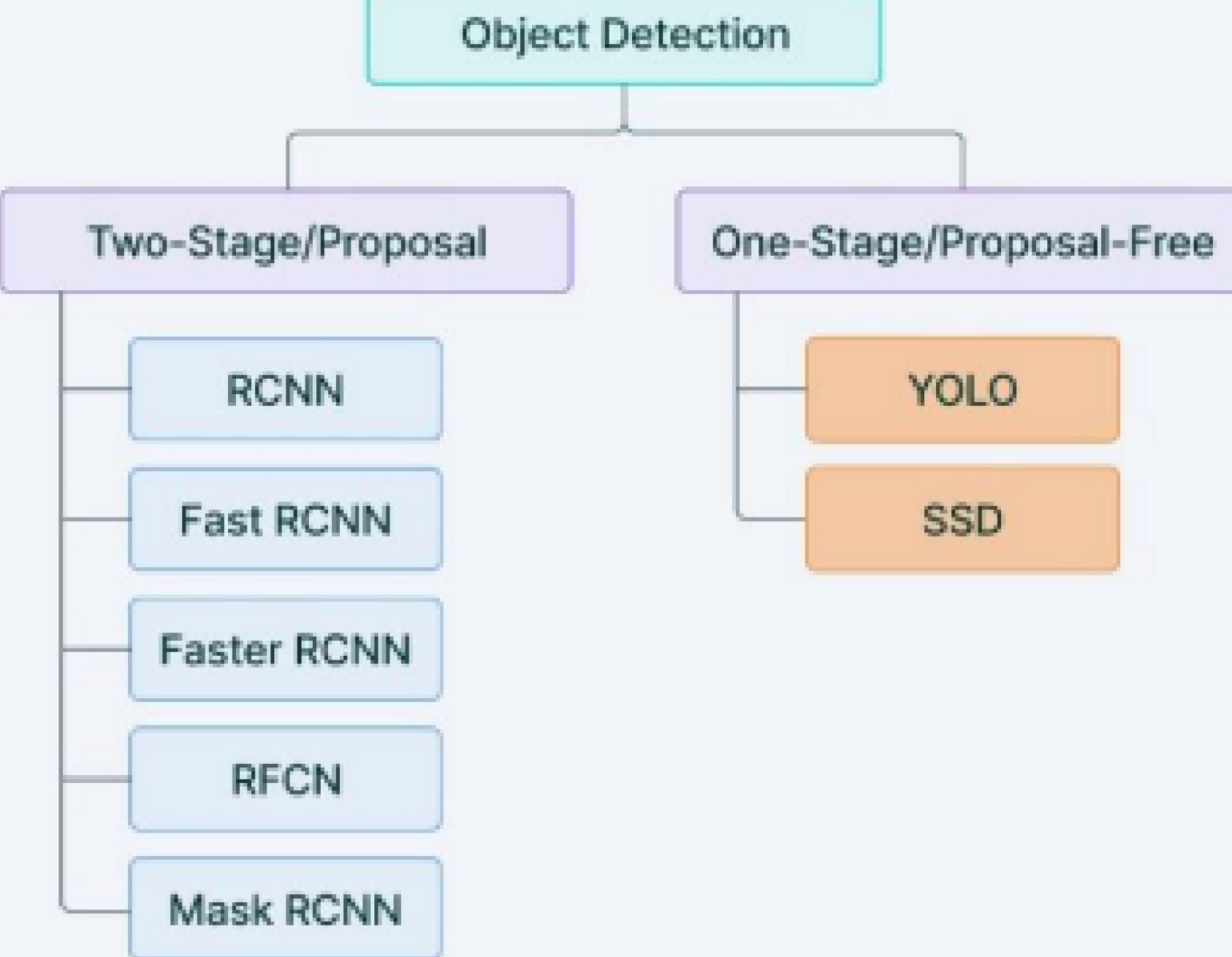
MEDIAPILE

Nhận diện 468 điểm trên gương mặt: trích
xuất ra vị trí (x, y) của các điểm trên mắt trái
và phải sau đó tính chỉ số EAR

YOLOV8

- Khá nhanh, có thể xử lý tới 155 khung hình/s, nhanh hơn nhiều so với các thuật toán phát hiện đối tượng khác
- Có khả năng hiểu được biểu diễn tổng quát của đối tượng. Nó cho phép mô hình phát hiện đối tượng nhìn toàn bộ hình ảnh khi kiểm tra
- Tính tiện lợi vì có thể triển khai trên nhiều phần cứng và kiến trúc khác nhau
- Cho phép tạo ra một mô hình phát hiện đối tượng tùy biến bằng cách huấn luyện nó trên các hình ảnh thu thập được

YOLOV8



THỰC HIỆN

YOLO

GIAI ĐOẠN 1

Xử lý dữ liệu

GIAI ĐOẠN 2

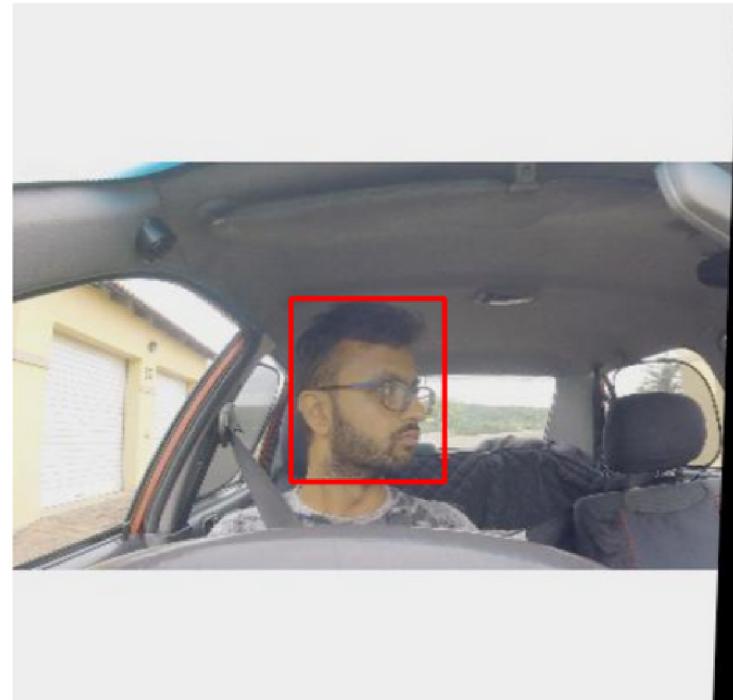
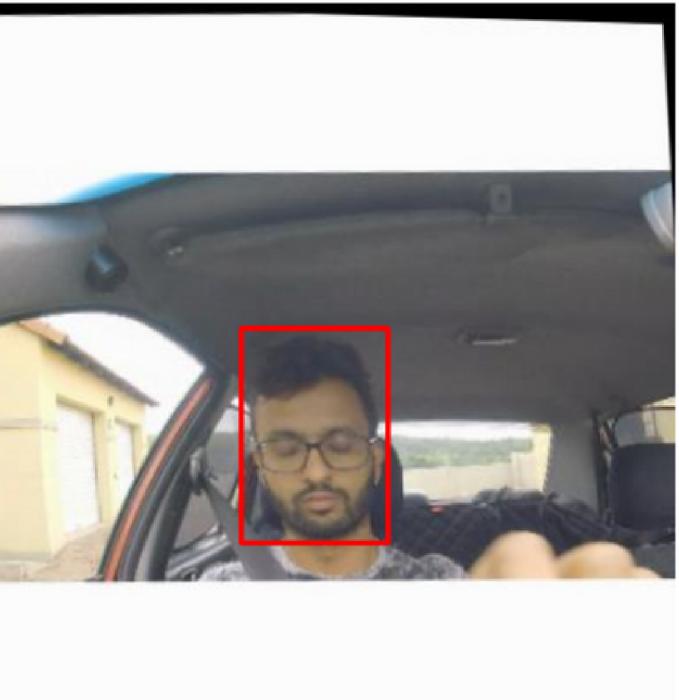
Huấn luyện mô
hình

GIAI ĐOẠN 3

Thực hiện test
trên ảnh và
trong thực tế

GIAI ĐOẠN 1

```
.filename, width, height, class, xmin, ymin, xmax, ymax      You, 6 days ago • first commit  
OPR0492_MP4-268.jpg.rf.e5c05791c11d4e46c2924dd9eb1d7182.jpg, 416, 416, drowsy, 178, 167, 252, 279  
OPR0492_MP4-508.jpg.rf.e174a38d1a728a2fef4a9702225babf7.jpg, 416, 416, awake, 161, 174, 257, 281  
OPR0492_MP4-401.jpg.rf.e1aef595027ff1e4c8ad6304150bfb72.jpg, 416, 416, drowsy, 169, 182, 259, 315  
OPR0492_MP4-426.jpg.rf.e26d52af3b4e6649129166a88e6d4876.jpg, 416, 416, awake, 161, 180, 244, 281  
OPR0492_MP4-484.jpg.rf.e73dd519981c915630891779c1d218f6.jpg, 416, 416, awake, 143, 162, 236, 270  
OPR0492_MP4-189.jpg.rf.e5743bd5ddfc9537205ab37be9621f3b.jpg, 416, 416, drowsy, 166, 160, 249, 272  
OPR0492_MP4-297.jpg.rf.e2ebde1b46c1e0979032d9434f23b5e8.jpg, 416, 416, drowsy, 170, 165, 266, 305  
OPR0492_MP4-509.jpg.rf.e5936b5cfcf72f06f1f9ca18599a205.jpg, 416, 416, awake, 177, 180, 261, 277  
OPR0492_MP4-106.jpg.rf.e259f54ead010e2e56c6bba9498d1417.jpg, 416, 416, awake, 179, 155, 292, 298  
OPR0492_MP4-91.jpg.rf.e4bb5fabe95b2be3db29368488df1d7d.jpg, 416, 416, awake, 143, 179, 222, 273  
OPR0492_MP4-245.jpg.rf.e6475794b57539642d1854e8ef688d90.jpg, 416, 416, drowsy, 155, 152, 273, 314  
OPR0492_MP4-11.jpg.rf.e3f608af5de89dfd982b441c11e57629.jpg, 416, 416, awake, 176, 93, 309, 274  
OPR0492_MP4-31.jpg.rf.e4dfffd1851af477e5c10c471f8010fac.jpg, 416, 416, awake, 148, 152, 242, 277  
OPR0492_MP4-348.jpg.rf.e30ae7263a38555b12f6121a3ffbbd58.jpg, 416, 416, awake, 138, 174, 222, 295  
OPR0492_MP4-95.jpg.rf.e7ed7fdbdc558d141314f2fa9a340dd.jpg, 416, 416, awake, 172, 165, 260, 280  
OPR0492_MP4-366.jpg.rf.e836152cbcfc2355853ef7a45811476c1.jpg, 416, 416, drowsy, 138, 172, 236, 279  
OPR0492_MP4-404.jpg.rf.e68b78dd9c17f9d341fa6ae3f2ece0f4.jpg, 416, 416, drowsy, 145, 148, 236, 289  
OPR0492_MP4-210.jpg.rf.e50af1676377264e8908966542c1d1aa.jpg, 416, 416, awake, 167, 163, 250, 279  
OPR0492_MP4-83.jpg.rf.e86b028e750e689ab3e9d476f8e7a48f.jpg, 416, 416, awake, 124, 159, 221, 292
```



GIAI ĐOẠN 1

```
# 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')
```

GIAI ĐOẠN 2

Thay đổi nhãn trong layer cuối trong mô hình yolov8 có sẵn

Nhãn gồm: 1: 'Awake' và 0: 'Drowsy'

Epochs = 10

Batch_size = 32

```
%%writefile data.yaml
path: 'D:/Drowsiness_Detection/drowsiness_dataset'
train: 'D:/Drowsiness_Detection/drowsiness_dataset/train/images'
val: 'D:/Drowsiness_Detection/drowsiness_dataset/valid/images'

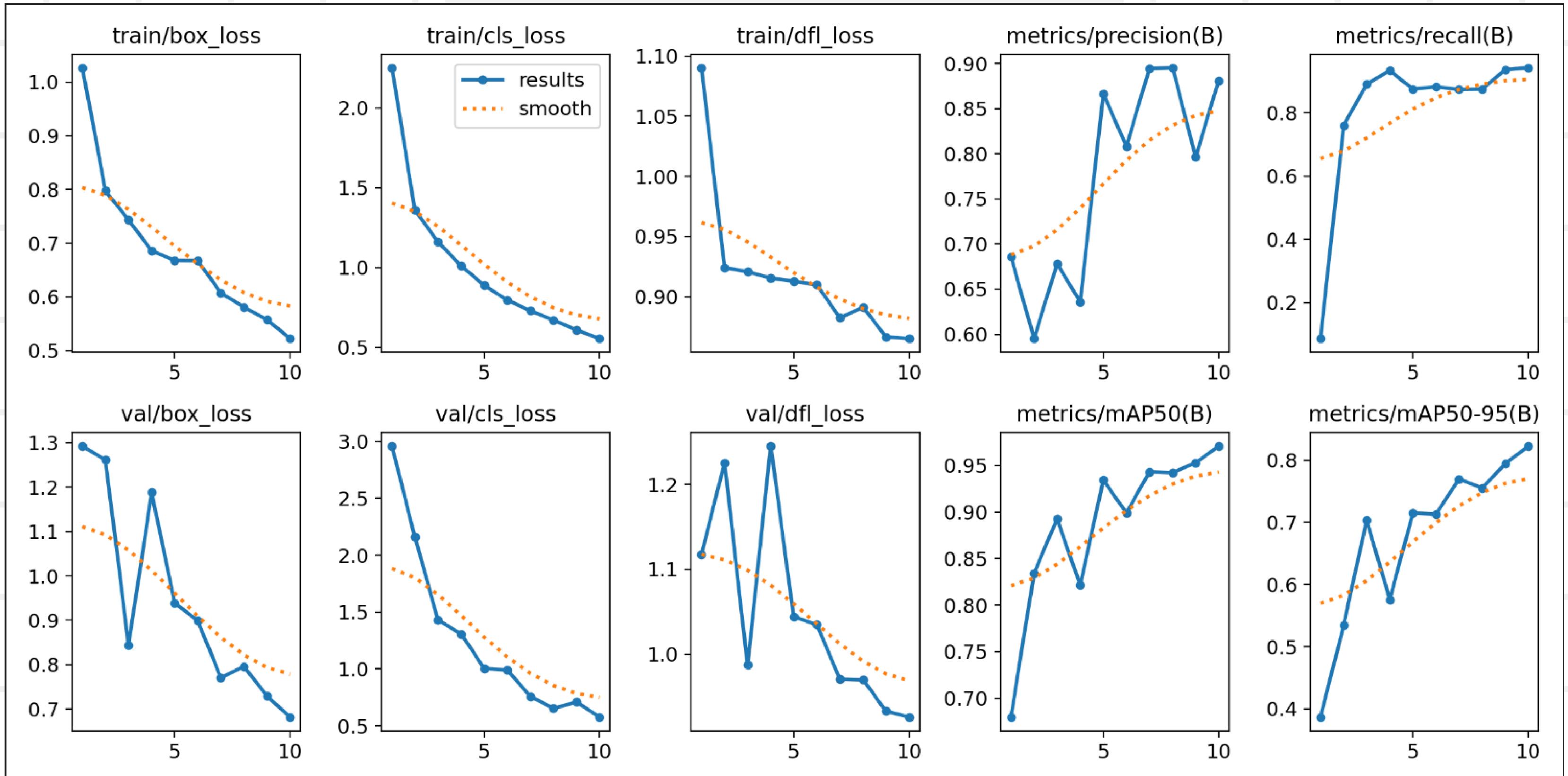
# class names
names:
  0: 'drowsy'
  1: 'awake'
```

MagicPython

```
EPOCHS = 10
!yolo task=detect mode=train model=yolov8n.pt imgsz=416 data=data.yaml epochs={EPOCHS} batch=32 name=yolov8n_v8_50e
```

MagicPython

GIAI ĐOẠN 2



GIAI ĐOẠN 3

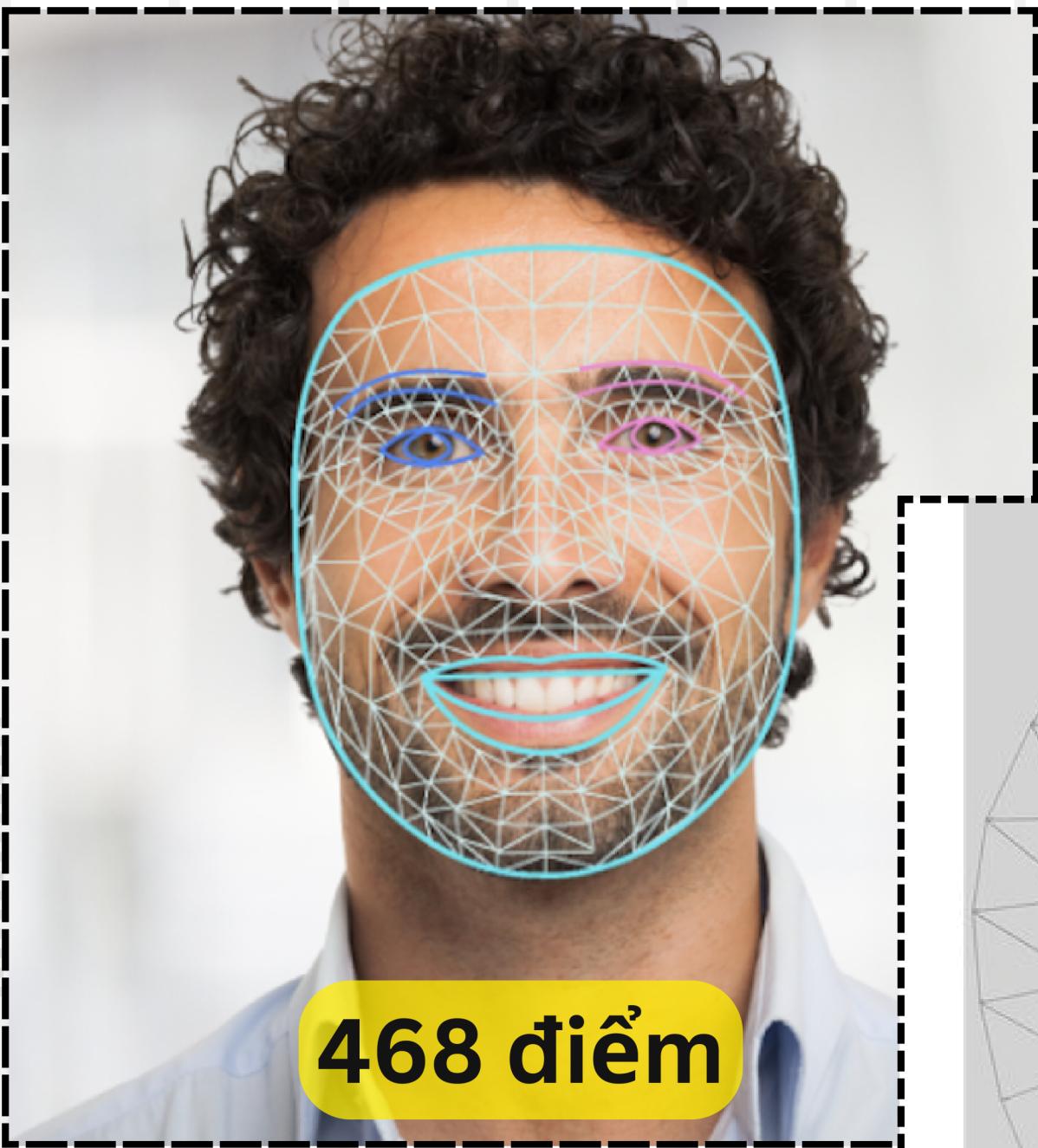
Thực hiện test trên tập Validation



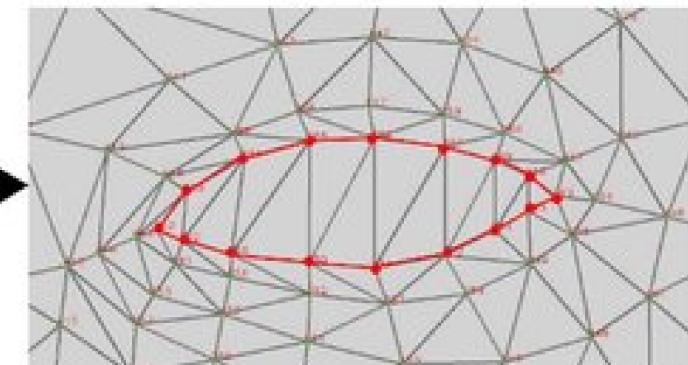
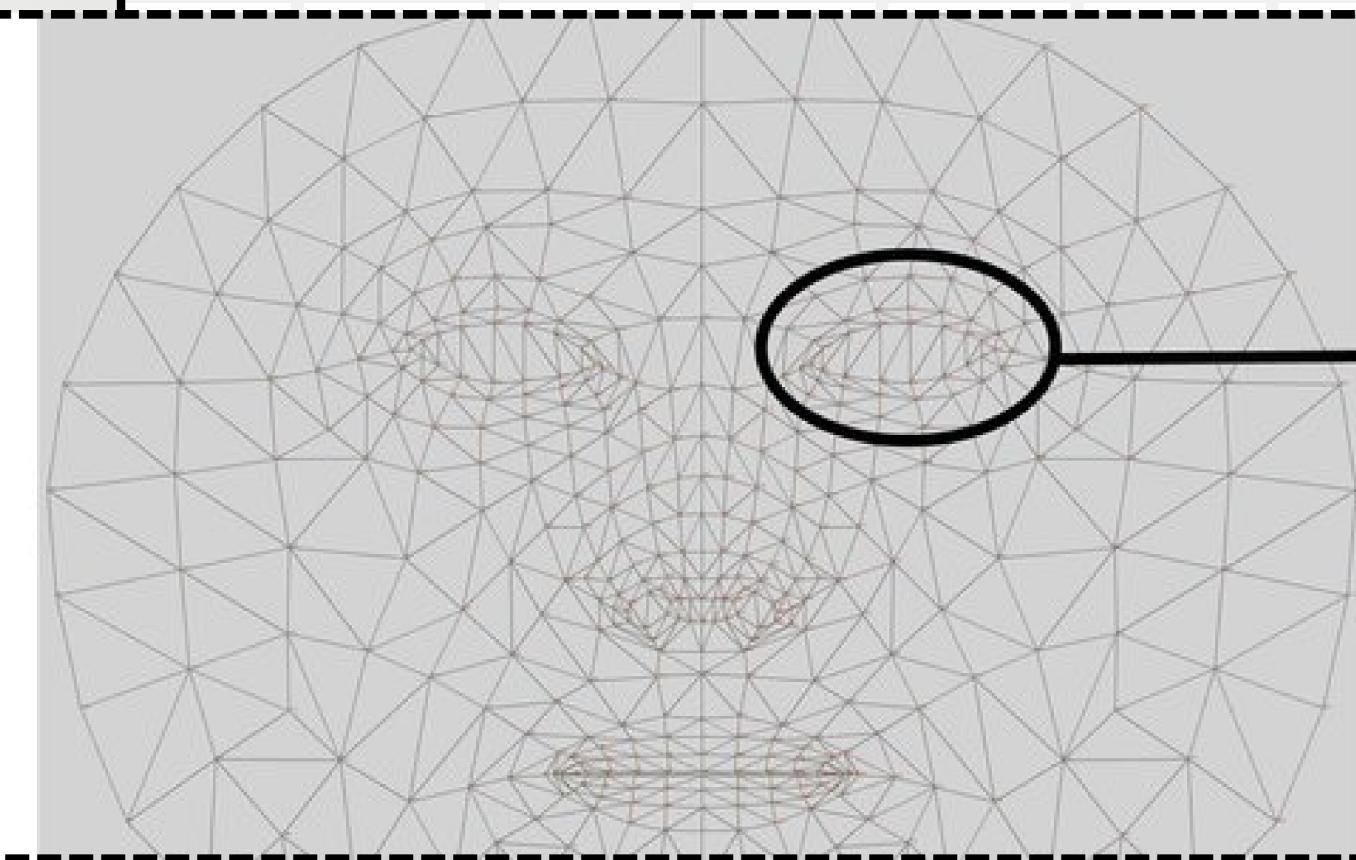
Nhận xét

**Mô hình YOLOv8 custom tốt với dữ liệu tự thu thập.
Tuy nhiên, do hạn chế về kích thước dữ liệu, dẫn đến
mô hình còn nhiều hạn chế**

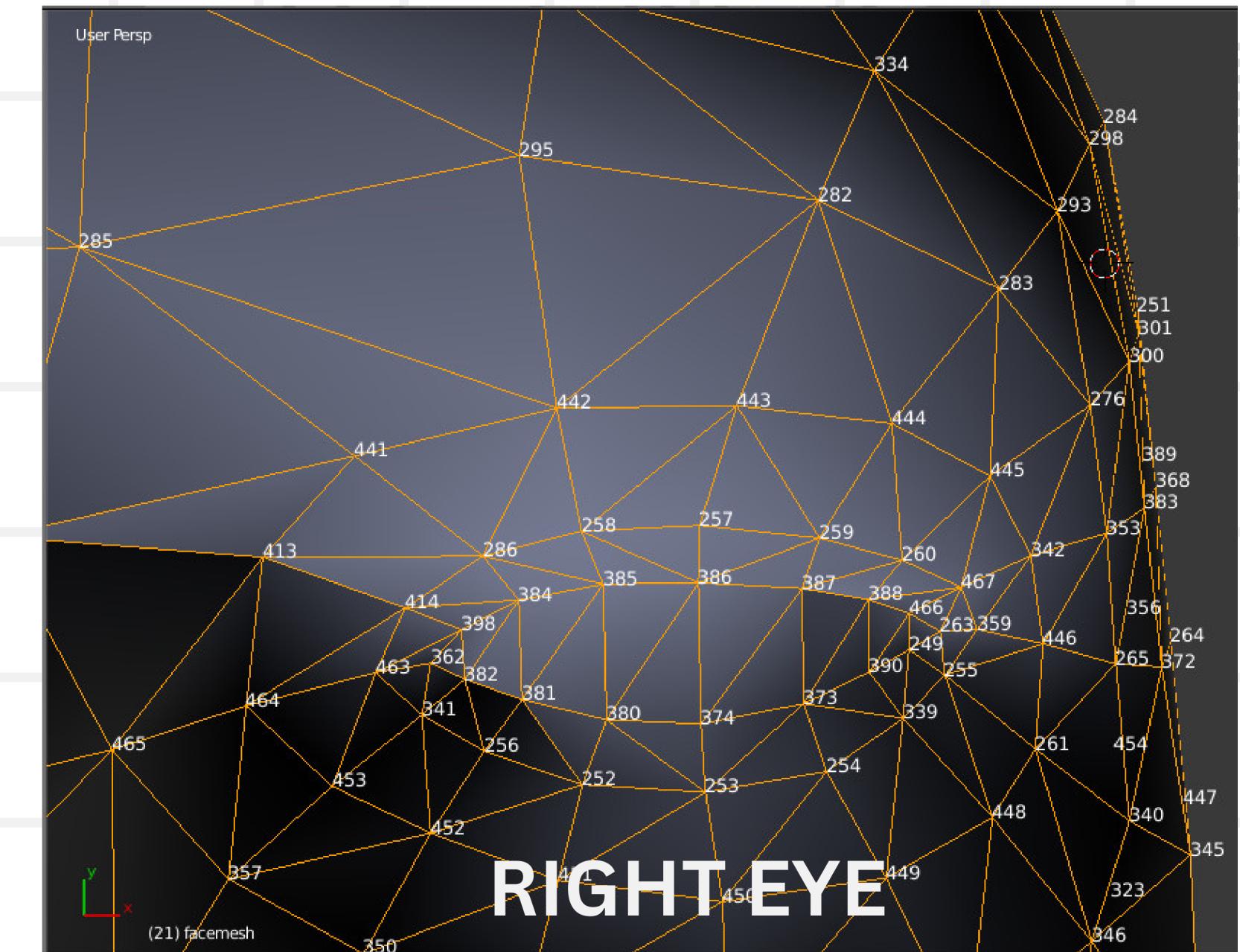
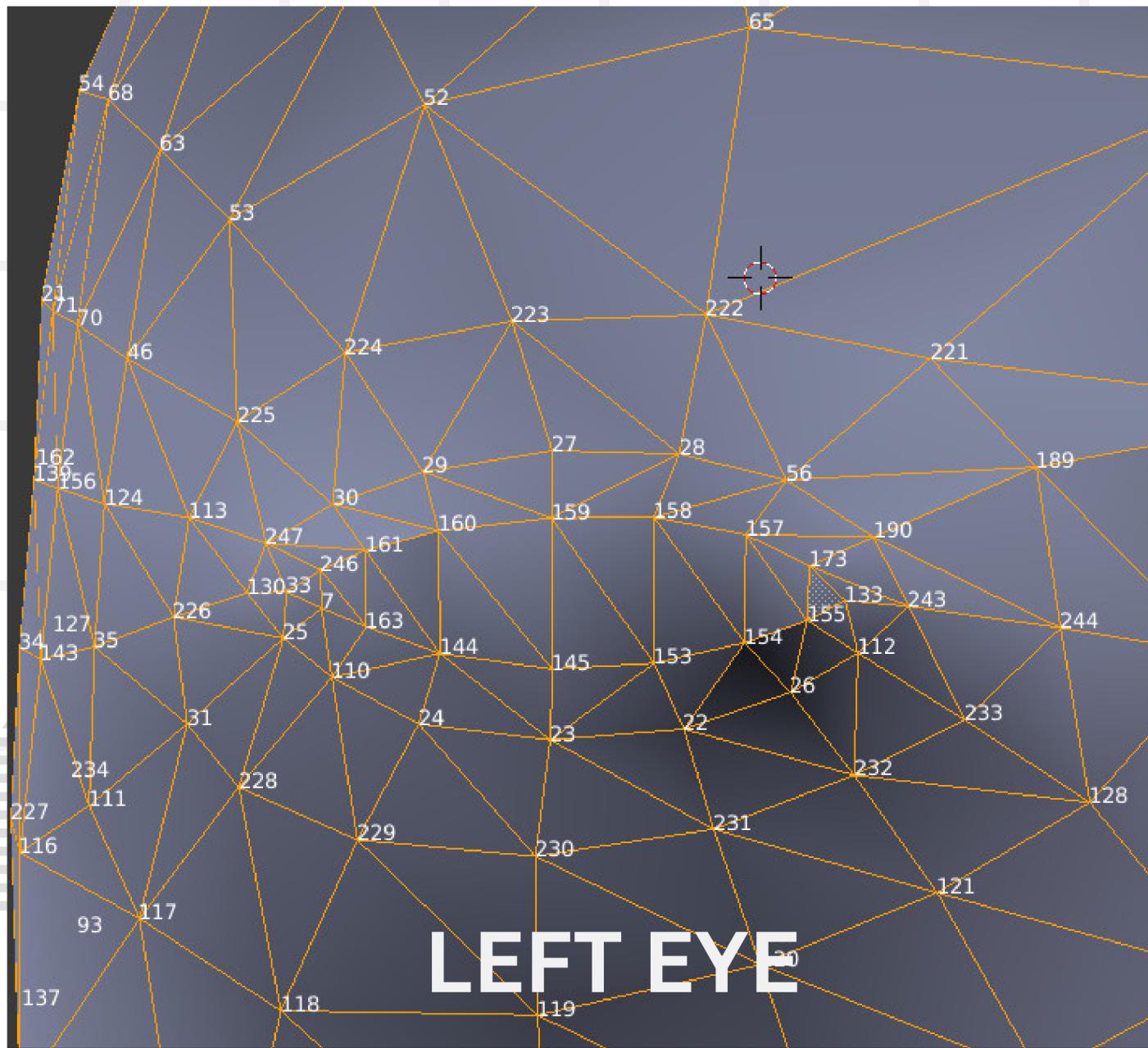
MEDIAPILE



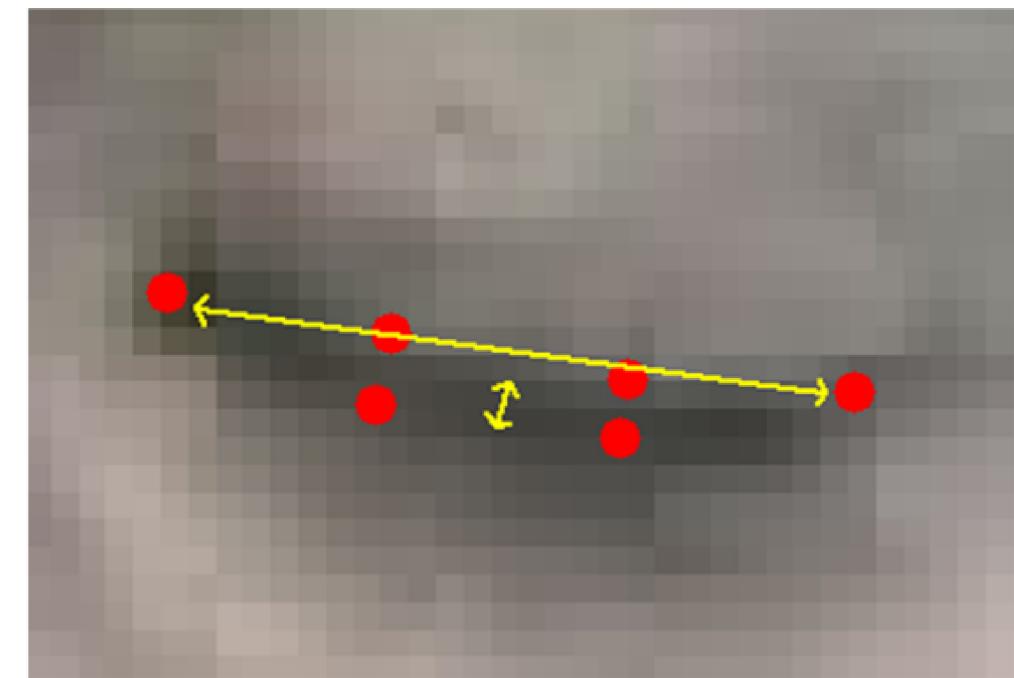
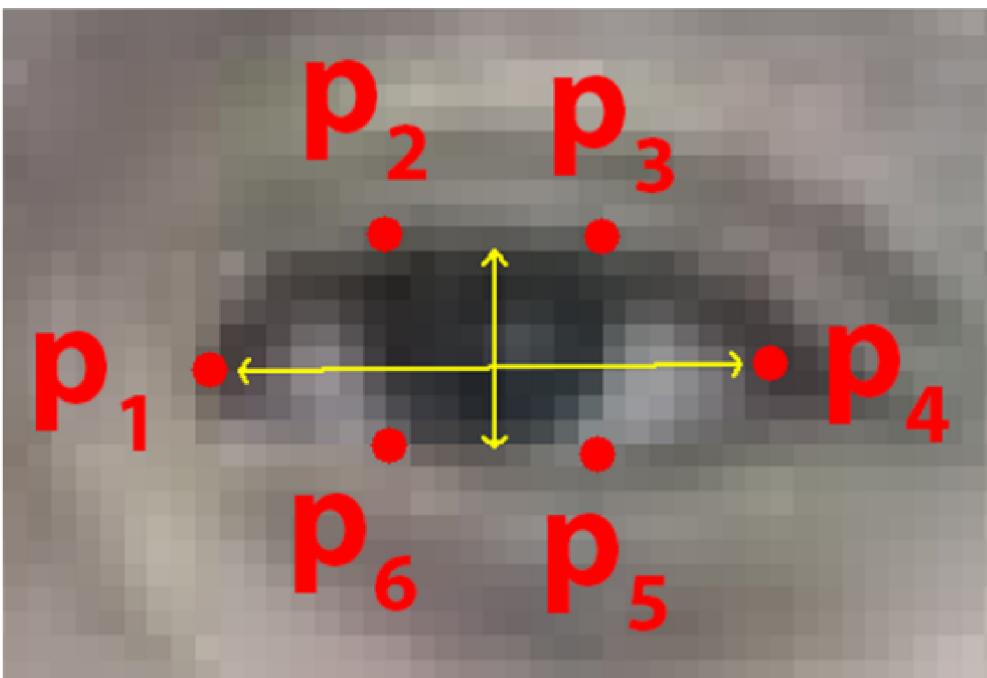
Facial landmark?



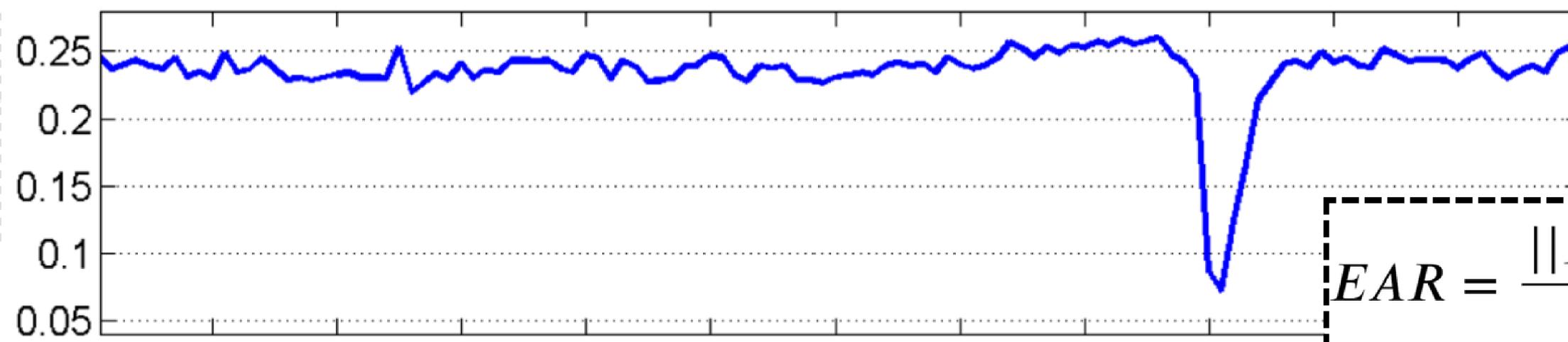
MEDIAPILE



MEDIAPILE

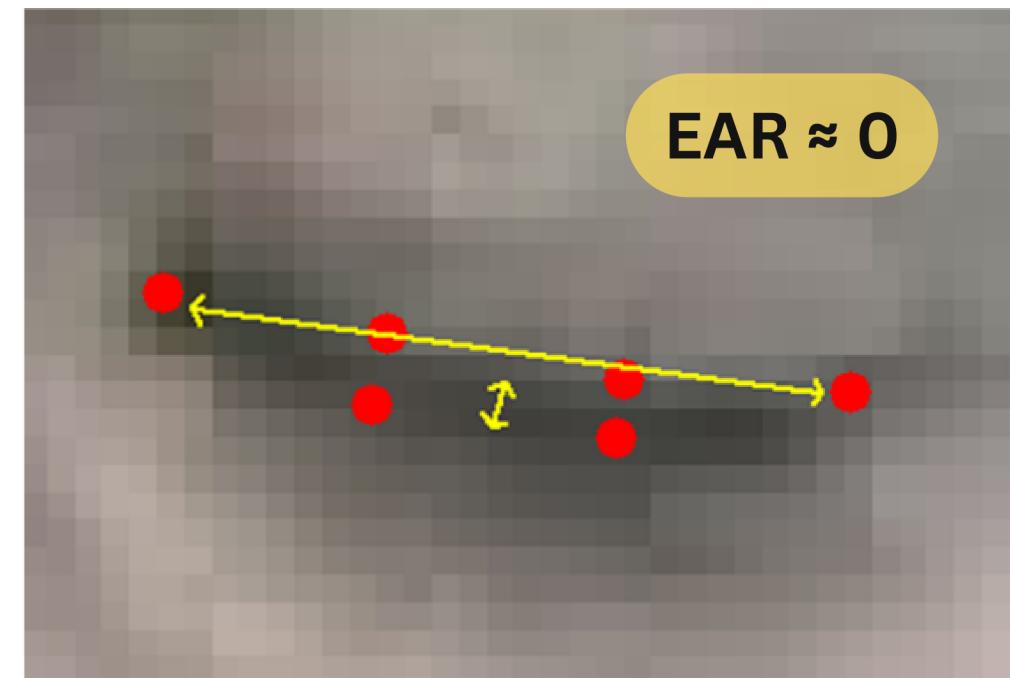
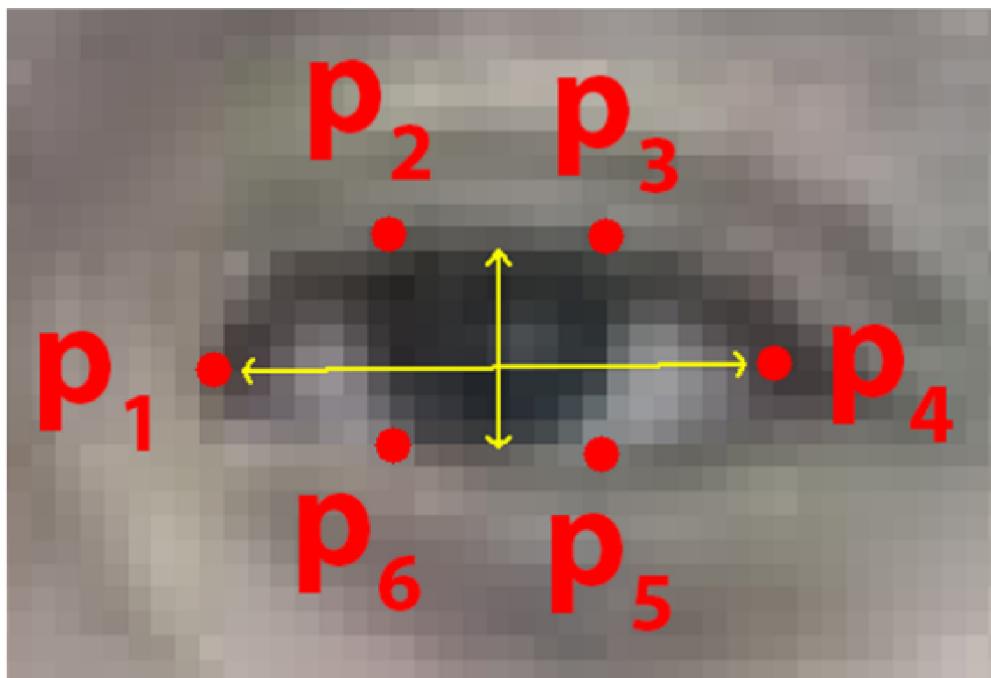


EAR?
Eye Aspect Ratio

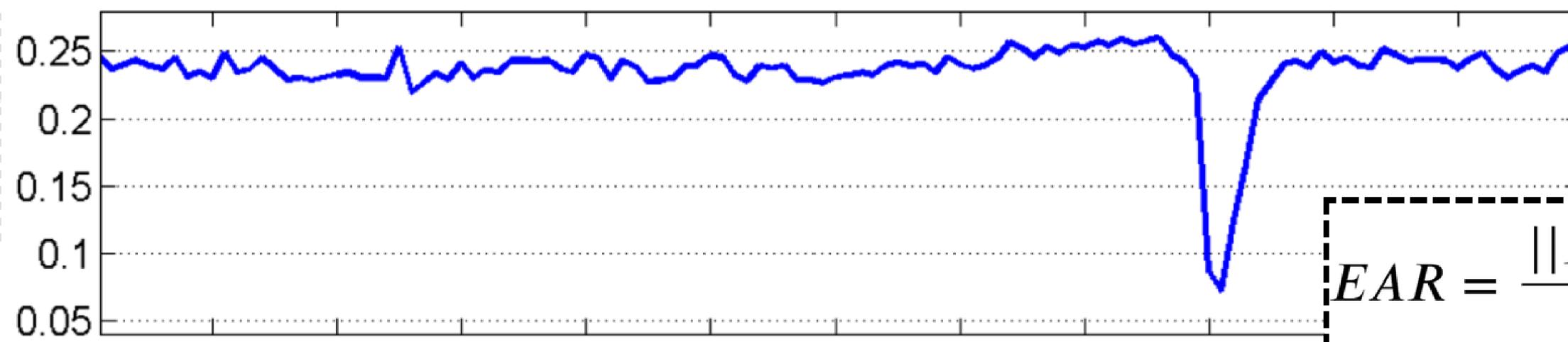


$$EAR = \frac{||P_2 - P_6|| + ||P_3 - P_5||}{2||P_1 - P_4||}$$

MEDIAPILE

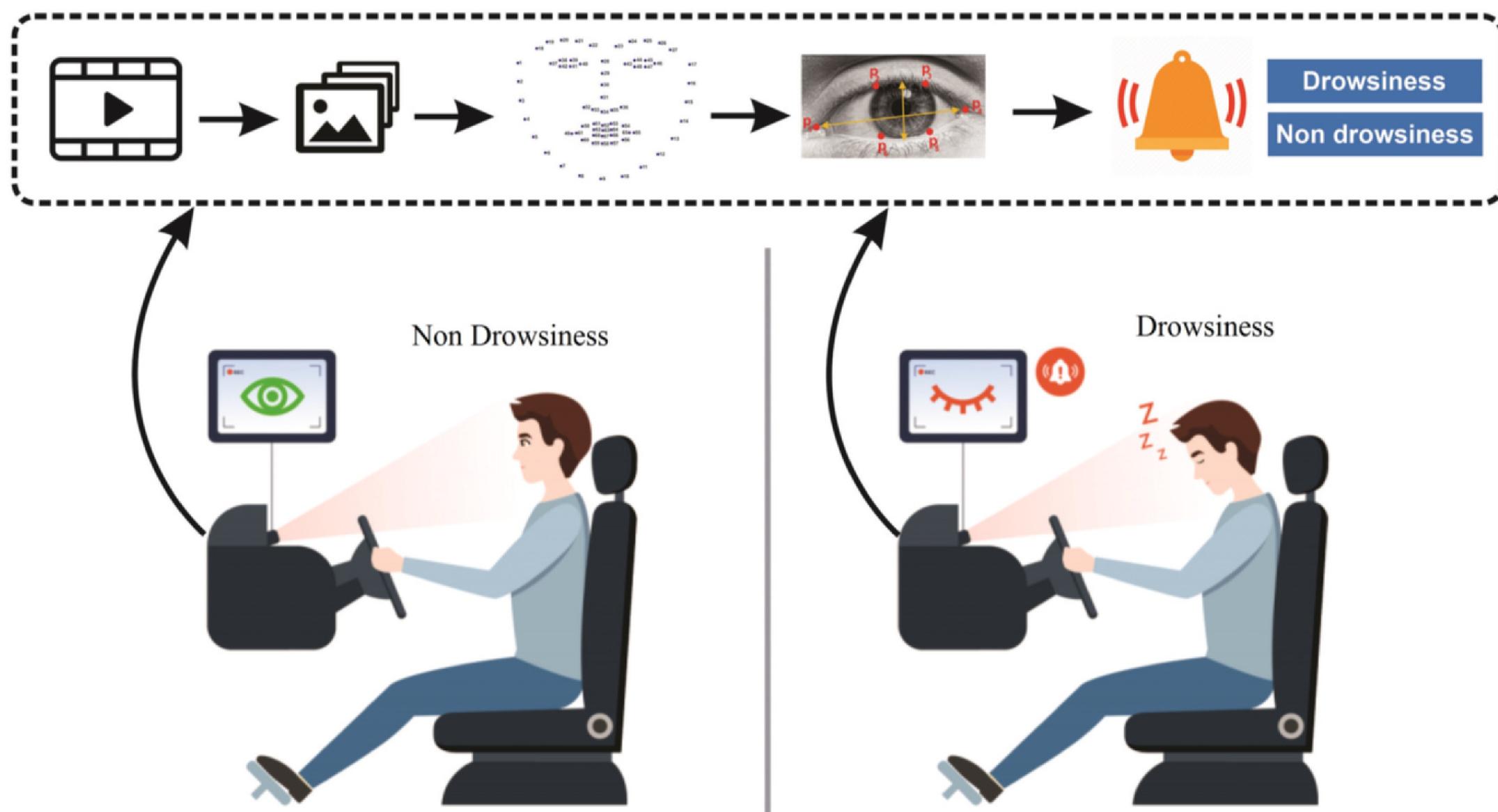


EAR?
Eye Aspect Ratio

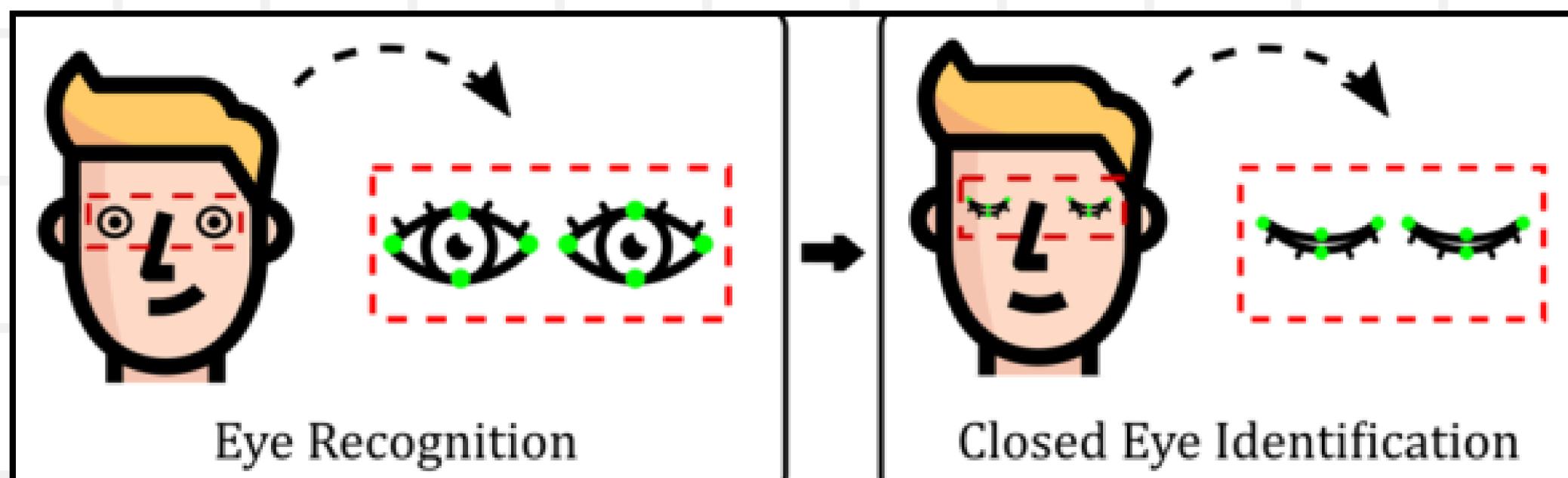


$$EAR = \frac{||P_2 - P_6|| + ||P_3 - P_5||}{2||P_1 - P_4||}$$

MEDIAPILE



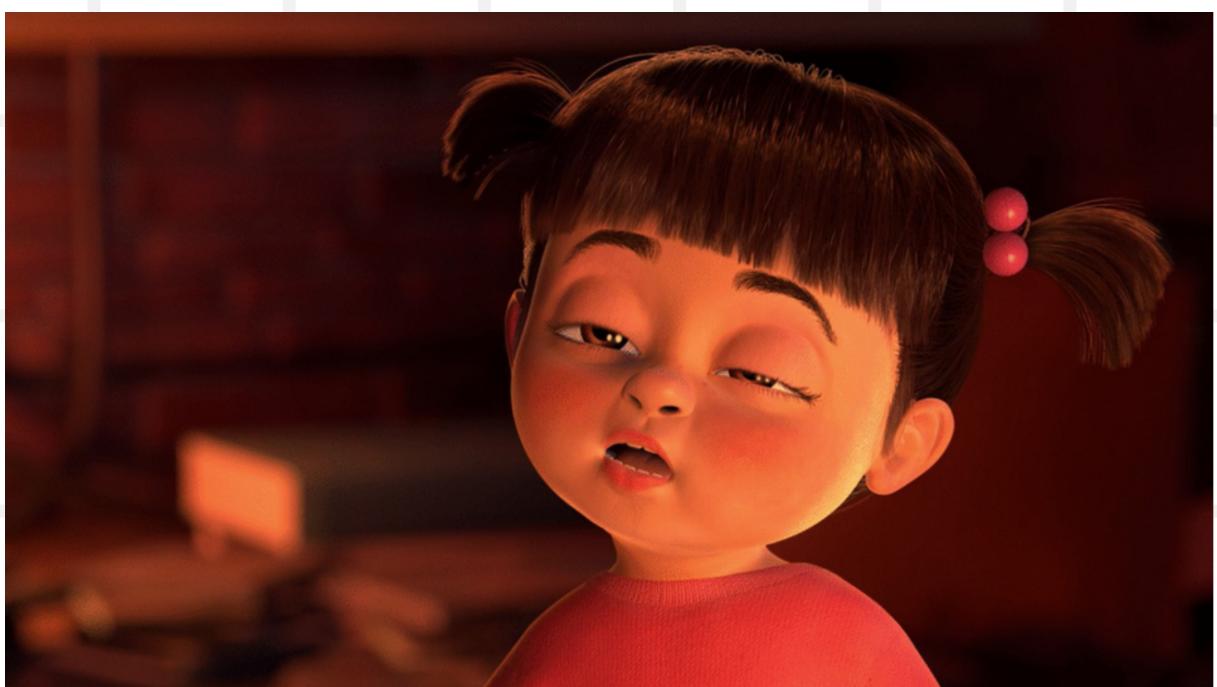
THRESHOLD



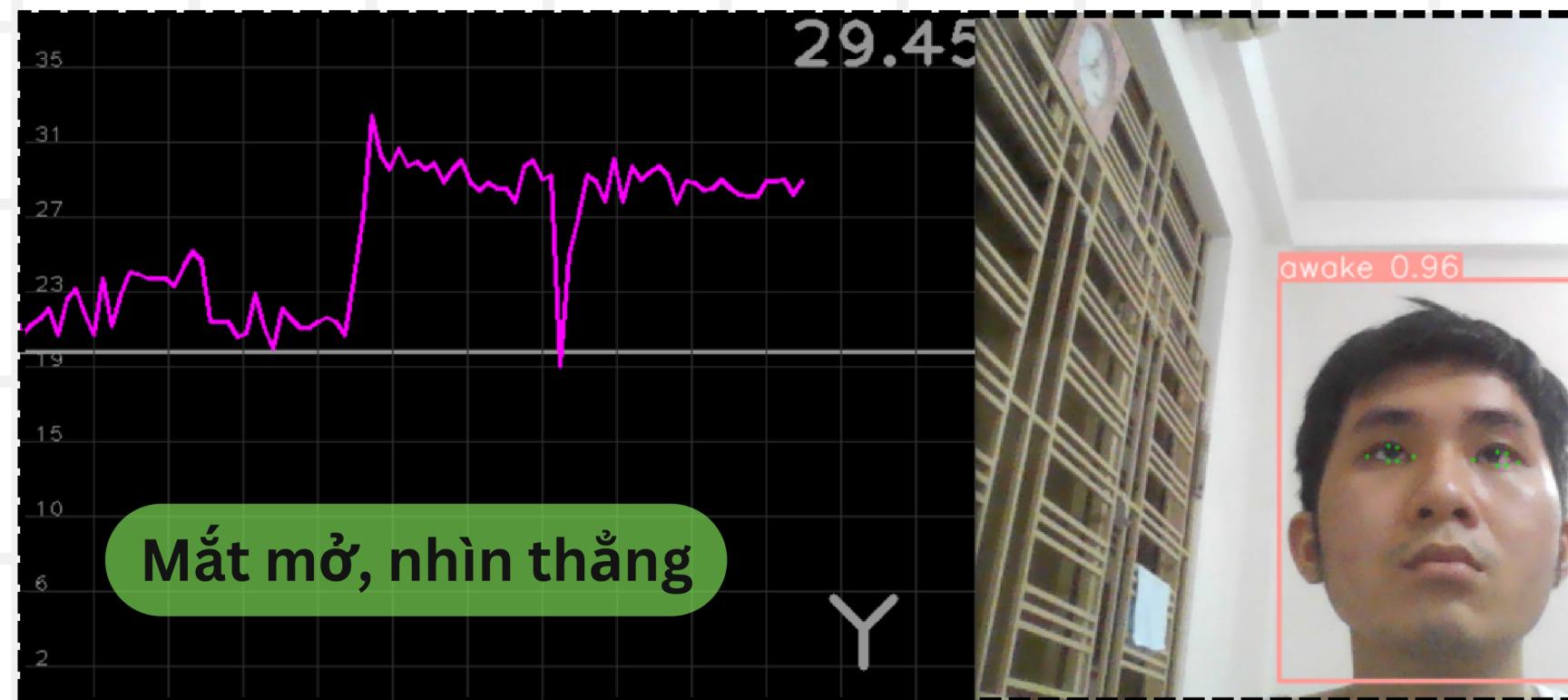
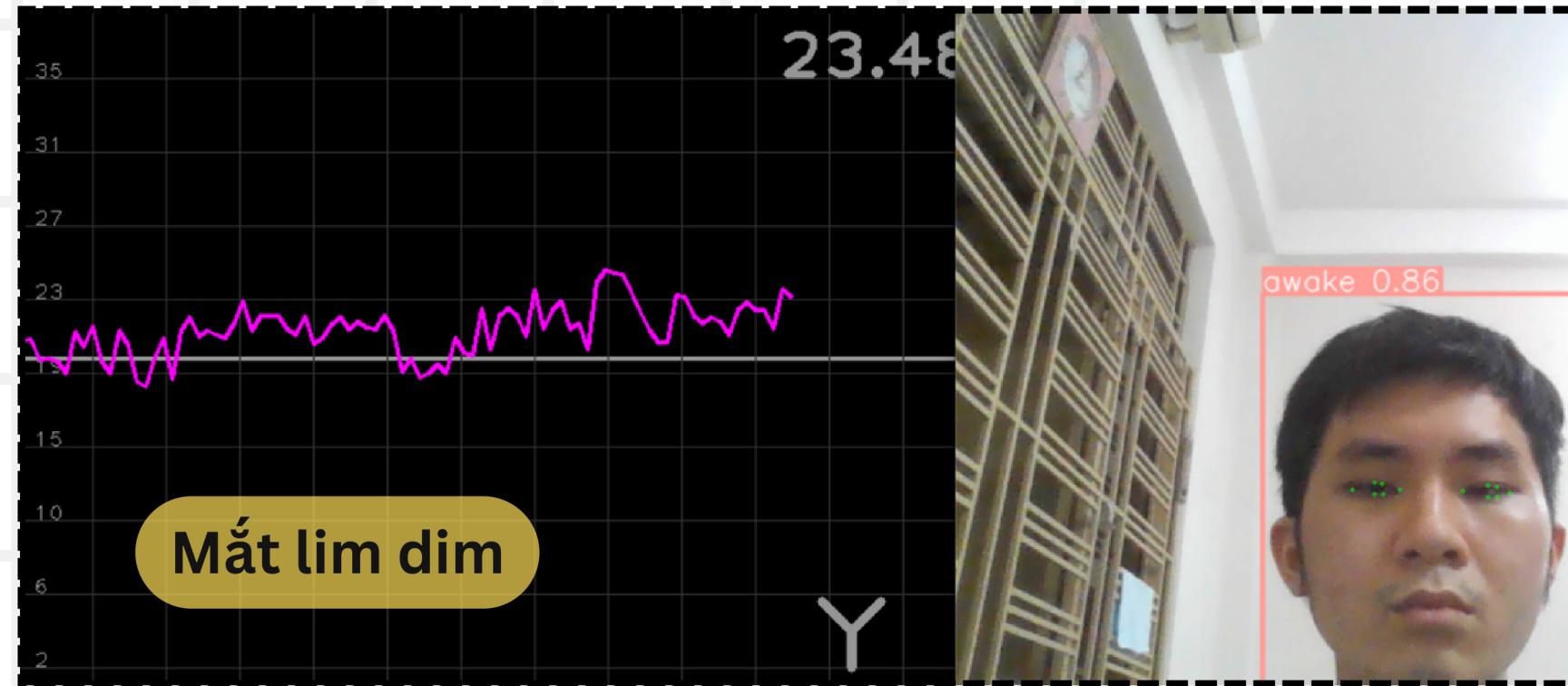
mắt lim dim ?

mắt nhắm ≈ 0

? < mắt mở < ?



THRESHOLD - THỰC NGHIỆM



THRESHOLD - THỰC NGHIỆM

Chọn: EAR threshold= 27

YOLO vs Mediapipe

YOLOv8

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

Chi phí cho dataset cao
Khó mở rộng trường hợp

Mediapile

YOLO vs Mediapile

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

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

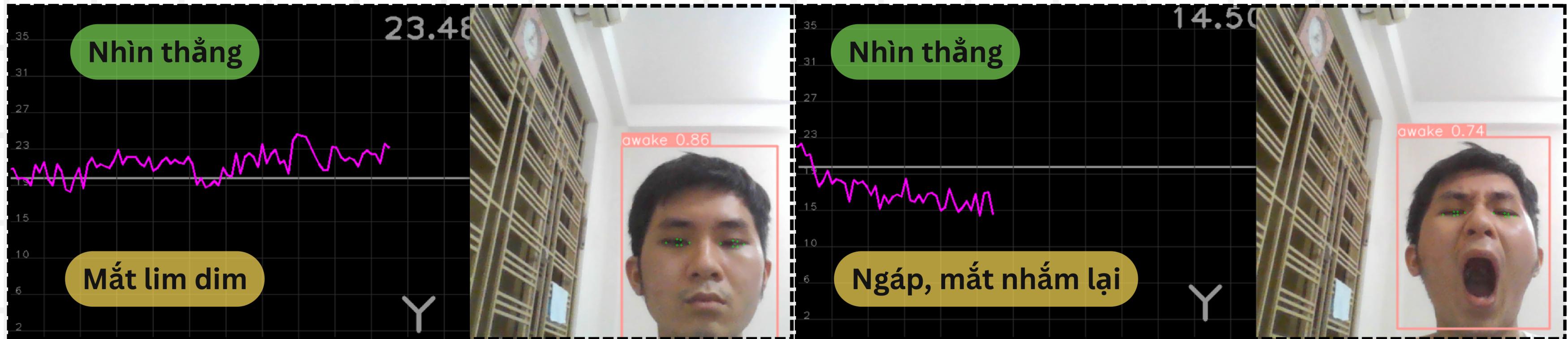
YOLO vs Mediapipe

YOLOv8 nhanh nhưng chưa chính xác trong một vài trường hợp



YOLO vs Mediapipe

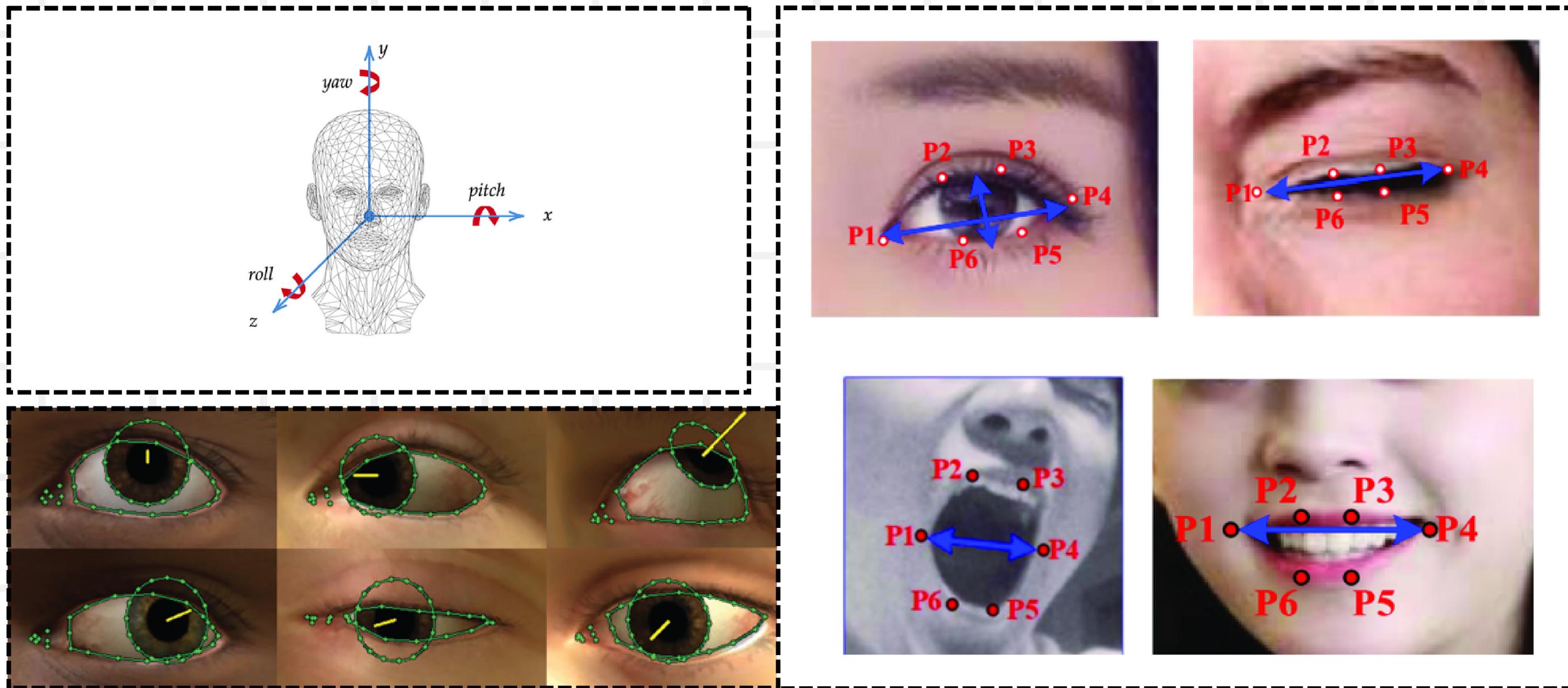
Mediapile và hướng tiếp cận sử dụng ngưỡng EAR = 27 sẽ khắc phục được sai sót của YOLOv8



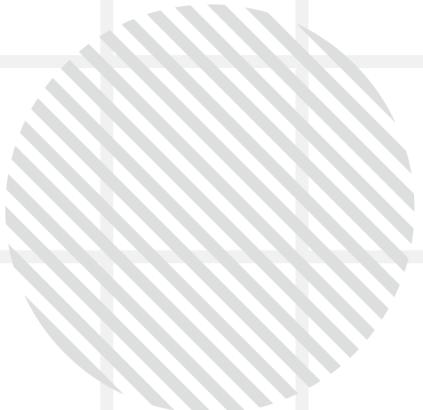
Khó khăn

Kết hợp 2 phương pháp để hỗ trợ lẫn nhau
Ghi nhận biểu hiện gương mặt khi phải xem xét nhiều yếu tố

Tương Lai



DEMO



TÀI LIỆU THAM KHẢO



<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/>

Bảng phân công công việc

Công việc	Phạm Quốc Anh Khoa	Nguyễn Vũ Hoàng Long
Tìm dataset	x	x
Tìm hiểu dataset, tiền 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

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>

THANK YOU

