PHÁT HIỆN TÀI XẾ BUỒN NGỦ HOẶC MẤT TẬP TRUNG KHI LÁI XE

Group 15

Nguyễn Khắc Luật - 21099741 Hoàng Ngọc Tân - 21074741 Nguyễn Thị Ty Ty – 21096511



Nội dung trình bày



01

Giới thiệu

02

Dữ liệu sử dụng

03

06

Quy trình thực hiện



04

Ứng dụng thực tế và triển vọng 05

Thách thức

Kết luận

1.1. Đặt vấn đề



- Hàng năm, hơn 12.000 vụ tai nạn do tài xế gây ra, 30% trong số đó liên quan đến buồn ngủ hoặc mất tập trung, đặc biệt là hiện tượng "giấc ngủ trắng" trên các tuyến đường dài, cao tốc, tăng nguy cơ tai nạn nghiêm trọng.
- Tai nạn do buồn ngủ gây nhiều ca tử vong. Hệ thống an toàn hiện tại chỉ can thiệp muộn, không phát hiện sớm mệt mỏi. Giải pháp là ứng dụng Computer Vision và Deep Learning để giám sát và cảnh báo kịp thời tình trạng tài xế.

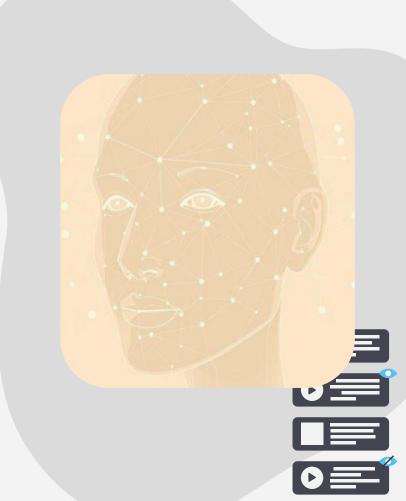


1.1. Mục tiêu nghiên cứu



- Xây dựng một hệ thống AI sử dụng Computer Vision để phát hiện và phân loại trạng thái tài xế (ngáp, buồn ngủ, tỉnh táo).
- Phát triển cơ chế cảnh báo thời gian thực nhằm giúp tài xế duy trì sự tỉnh táo, giảm nguy cơ tai nạn giao thông.









- Sử dụng ảnh của mắt và miệng ở các trạng thái khác nhau để xác định tình trạng của tài xế.
- Buồn ngủ được xác định khi mắt nhắm, miệng ngáp hoặc khi cả hai điều đó xảy ra
- Bộ dữ liệu có 2589 ảnh mắt nhắm, 3845 ảnh mắt mở, 2591 ảnh miệng ko ngáp và 2528 ảnh miệng đang ngáp.
- Tất cả ảnh được resize về 224x224 đảm bảo độ chi tiết và phù hợp với các mô hình.









Dữ liệu ảnh nhắm mắt







Dữ liệu ảnh mở mắt







Dữ liệu ảnh không ngáp

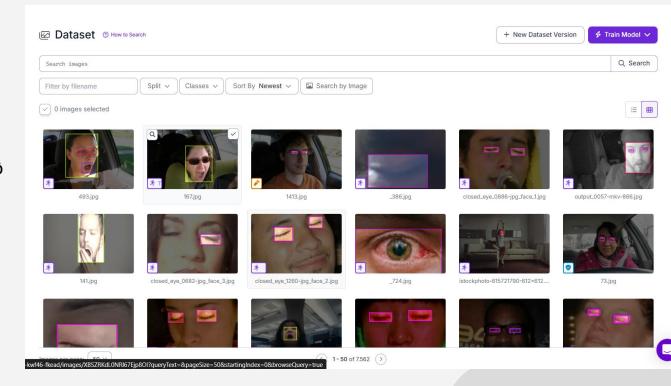




Dữ liệu ảnh ngáp



Dữ liệu dành cho YOLO: Được label nhãn thủ công trên Roboflow và theo cấu trúc của mô hình YOLO













3.1. Sử dụng CNN

- Ånh được resize về 224 x 224
- Tập dữ liệu được chia thành 2 tập train và val với tỉ lệ là 80:20, sử dụng random_split của Pytorch để chia dataset thành 2 tập con là train_dataset và val_dataset. Random_split để chia ngẫu nhiên nhưng không trộn lẫn các phần tử.

```
Tỷ lệ chia dữ liệu: Train 80% / Val 20%

close_eyes: 2589 ảnh ➤ Train: 2071, Val: 518

no_yawn: 2591 ảnh ➤ Train: 2072, Val: 519

open_eyes: 3845 ảnh ➤ Train: 3076, Val: 769

yawn: 2528 ảnh ➤ Train: 2022, Val: 506
```



Số lượng ảnh mỗi tập sau khi chia



3.1. Sử dụng CNN

- Dữ liệu được huấn luyện trên nền tảng Kaggle, sử dụng GPU P100. huấn luyện với 30 epochs.
- Thời gian thực hiện train 30 epochs là 15 phút 35 giây.
- Độ chính xác cao nhất đạt được trên tập val là 0.9416
- Model được lưu lại với tên cnn_model_02.pth (cnn_model_01.pth có 10 epochs lên ko đảm bảo độ chính xác) để sử dụng cho các bước tiếp theo.





3.1. Sử dụng CNN

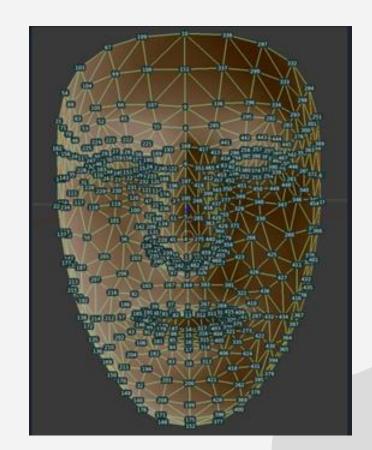
- Khuôn mặt và các bộ phận của khuôn mặt sẽ được tách ra riêng để xác định trạng thái, đối với mắt có open_eyes và close_eyes, miệng có yawn và no_yawn.
- Sử dụng thư viện mediapipe, một framework mã nguồn mở từ Google, được thiết kế để xử lý các tác vụ nhận diện và theo dõi chuyển động trong thời gian thực, chủ yếu trong thị giác máy tính (computer vision).
- Khuôn mặt, miệng và hai mắt được tách ra dựa vào 468 điểm đặc trưng trên khuô mặt.





3.1. Sử dụng CNN

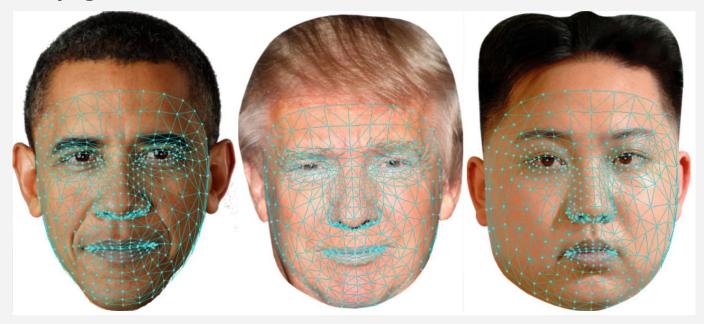
 468 điểm đặc trưng trên khuôn mặt







3.1. Sử dụng CNN





468 điểm đặc trưng trên khuôn mặt được áp dụng cho các khuôn mặt khác nhau

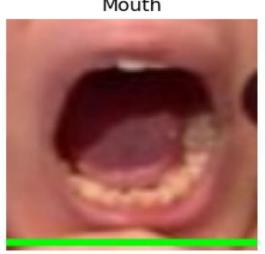


3.1. Sử dụng CNN

Face with Bounding Box



Mouth



Right eye



Left_eye





Khuôn mặt được tách thành các bộ phận



3.1. Sử dụng CNN

 Tình trạng buồn ngủ hay không sẽ được dự đoán dựa trên các label đã dự đoán từ trạng thái mắt và miệng.



Dự đoán chi tiết:

Left Eye: close_eyes (99.8%) Right Eye: open_eyes (66.2%)

Mouth: yawn (94.9%)

Tổng kết: DROWSINESS (97.4%)





3.1. Sử dụng CNN

 Đối với video sẽ chia ra 20 khung hình mỗi giây, dự đoán từng khung hình và sắp xếp làm thành 1 video.





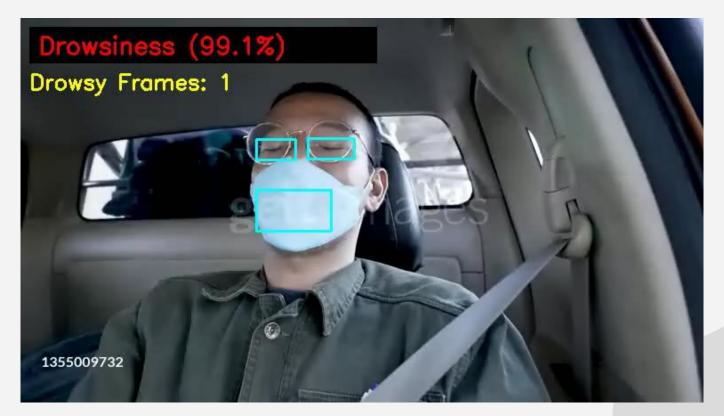
3.1. Sử dụng CNN







3.1. Sử dụng CNN







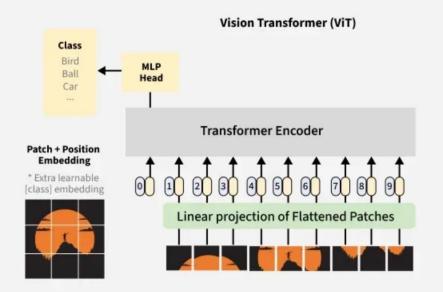
3.2. Sử dụng ViT (Vision Transformer)

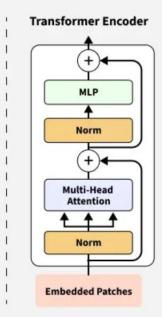
- Quy trình xử lý dữ liệu tương tự như CNN, nhưng sử dụng mô hình vit_base_patch16_224.
- Được train trên nền tảng kaggle với GPU P100, thời gian train là 1 giờ 5 phút với 20 epochs.
- Độ chính xác cao nhất trên tập val là 0.9559
- Mô hình đạt độ chính xác cao nhất được lưu lại với tên vit_model_01.pth
 để xử dụng trong các bước xử lý trong tương lai.





3.2. Sử dụng ViT (Vision Transformer)



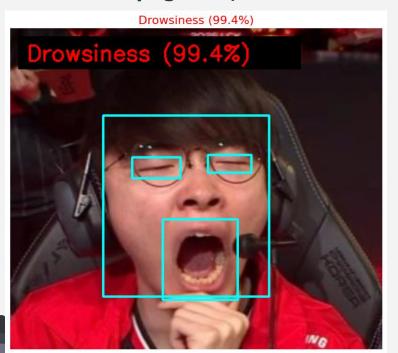




Kiến trúc mô hình ViT



3.2. Sử dụng ViT (Vision Transformer)







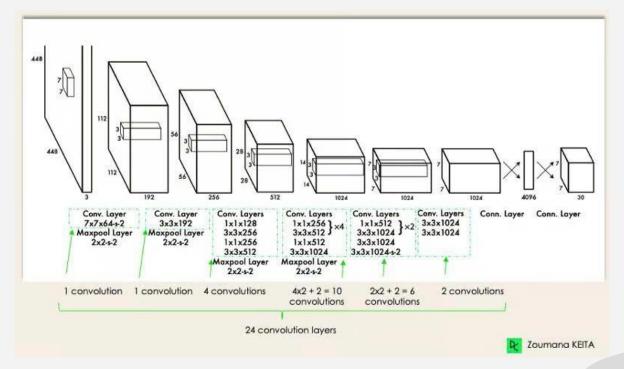
3.2. Sử dụng ViT (Vision Transformer)







3.3. Sử dụng YOLO:

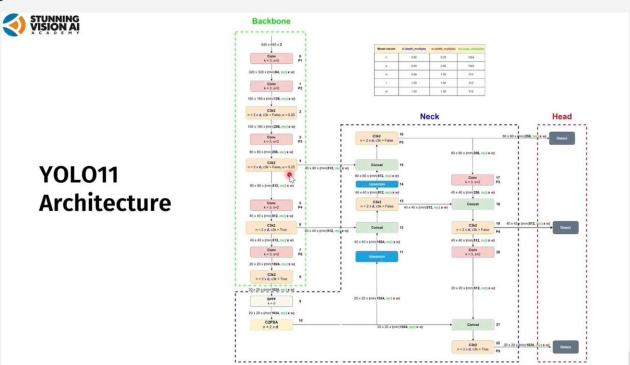




Kiến trúc YOLO



3.3. Sử dụng YOLO:





Kiến trúc YOLO11



3.3. Sử dụng YOLO:

Từ 7562 ảnh thiết lập tập valid = 2115 ảnh và test = 1218 ảnh Roboflow chúng ta có thể augment tập train lên 3 lần và nhận được 12676 ảnh tập train Và tỷ lệ train/valid/test là 79:13:8

Dataset Split	TRAIN SET 79%	VALID SET 13%	TEST SET	8%
	12676 Images	2115 Images	1218 Images	



3.3. Sử dụng YOLO:



Resize ảnh về kích thước 640x640
Chạy với GPU P100 trên Kaggle với 30
epochs
Phiên bản sử dụng là yolo11n
Dùng dataset đã label từ Roboflow thông
qua API

Giai đoạn train sử dụng CLI

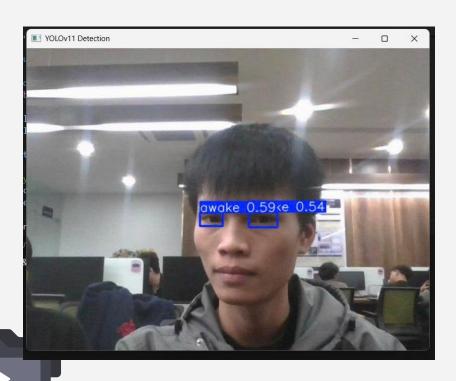
```
train: /kaggle/working/datasets/final2-2/train/images
val: /kaggle/working/datasets/final2-2/valid/images
test: /kaggle/working/datasets/final2-2/test/images
nc: 14
names: ['awake', 'Drowsy', 'Drowsy', 'Drowsy', 'Drowsy', 'Drowsy', 'Drowsy', 'face', 'awake', 'awake', 'Drowsy', 'awake', 'Drowsy', 'Drowsy', 'Drowsy', 'Drowsy', 'awake', 'Drowsy', 'Drowsy', 'Drowsy', 'Drowsy', 'Awake', 'Drowsy', 'Drowsy', 'Drowsy', 'Drowsy', 'Drowsy', 'Tace', 'awake', 'awake', 'Drowsy', 'Drowsy',
```

File data.yaml





3.3. Sử dụng YOLO:







4. Ứng dụng thực tế và triển vọng



Ứng dụng thực tế

- Ô tô cá nhân: Giám sát tài xế trên hành trình dài.
- Xe thương mại: Đảm bảo an toàn cho xe tải, xe khách, taxi.
- Xe tự hành: Hỗ trợ chuyển đổi chế độ tự động.
- Công ty vận tải: Giảm rủi ro tai nạn, tối ưu quản lý đội xe.

Triển vọng phát triển

- Kết hợp đa cảm biến: Tích hợp nhịp tim, EEG, nhiệt độ cơ thể để tăng độ chính xác.
- Cá nhân hóa: Điều chỉnh cảnh báo theo thói quen tài xế.
- Ứng dụng IoT: Kết nối đám mây, phân tích hành vi, gửi cảnh báo thời gian thực.



5. Thách thức

Điều kiện môi trường

Vấn đề: Ánh sáng yếu, góc quay không tối ưu, tài xế đeo kính hoặc khẩu trang làm giảm độ chính xác mô hình.

Giải pháp đề xuất: Tăng cường huấn luyện với dữ liệu đa dạng (ánh sáng, góc quay, phụ kiện), áp dụng kỹ thuật học sâu cải tiến để xử lý điều kiện bất lợi.

Hiệu suất thời gian thực

Vấn đề: Xử lý video liên tục yêu cầu tốc độ suy luận nhanh và phần cứng mạnh. Giải pháp đề xuất: Sử dụng mô hình nhẹ, tối ưu hóa thuật toán hoặc triển khai trên phần cứng chuyên dụng (GPU/TPU) để đảm bảo hiệu suất.

5. Thách thức

Độ chính xác trong phân loại trạng thái buồn ngủ



Vấn đề: Phân biệt các trạng thái gần giống nhau (buồn ngủ, mệt mỏi, mất tập trung) là thách thức lớn.

Giải pháp đề xuất: Kết hợp nhiều đặc trưng (mắt, tư thế đầu, nhịp tim) và sử dụng mô hình học sâu đa phương thức để cải thiện độ chính xác.

Quyền riêng tư và bảo mật

Vấn đề: Giám sát liên tục trạng thái tài xế gây lo ngại về quyền riêng tư và rò rỉ dữ liệu cá nhân.

Giải pháp đề xuất: Áp dụng mã hóa dữ liệu, lưu trữ ẩn danh và thiết kế hệ thống minh bạch với sự đồng ý của tài xế.

5. Thách thức

Drowsiness (97.3%) Drowsiness (97.3%)



6. Kết luận



Đề tài "Phát hiện tài xế buồn ngủ hoặc mất tập trung" cung cấp giải pháp hiệu quả, sử dụng Computer Vision và Deep Learning để phát hiện sớm dấu hiệu nguy hiểm, cảnh báo kịp thời, giảm tai nạn giao thông. Tương lai, hệ thống có thể tích hợp vào phương tiện hiện đại, nâng cao an toàn và chất lượng cuộc sống.



Thanks

