BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP. HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN

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

BÁO CÁO CUỐI KỲ

Môn học: Thị giác máy tính

Giảng viên hướng dẫn: PGS.TS Huỳnh Trung Hiếu

Sinh viên thực hiện:

Nguyễn Thị Ty Ty - 21096511

Nguyễn Khắc Luật – 21099741

Hoàng Ngọc Tân - 21074741

Lớp: DHKHDL17A

THÀNH PHỐ HỒ CHÍ MINH – 04/2025

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP. HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN

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

BÁO CÁO CUỐI KỲ

Môn học: Thị giác máy tính

Giảng viên hướng dẫn: PGS.TS Huỳnh Trung Hiếu

Sinh viên thực hiện:

Nguyễn Thị Ty Ty - 21096511

Nguyễn Khắc Luật – 21099741

Hoàng Ngọc Tân - 21074741

Lớp: DHKHDL17A

THÀNH PHỐ HỒ CHÍ MINH – 04/2025

Mục lục

I. Giới thiệu

1. Vấn đề đặt ra

Theo thống kê từ Ủy ban An toàn Giao thông Quốc gia, trong giai đoạn 2020 - 2023, Việt Nam ghi nhận hơn 12.000 vụ tai nạn giao thông mỗi năm do lỗi của tài xế, trong đó khoảng 30% (tương đương hơn 3.600 vụ) có liên quan đến tình trạng buồn ngủ hoặc mất tập trung khi lái xe. Những con số này phản ánh một thực trạng nghiêm trọng: buồn ngủ khi lái xe không chỉ là vấn đề cá nhân mà còn là mối đe dọa lớn đối với an toàn giao thông. Đặc biệt, trên các tuyến đường dài, đường cao tốc thẳng tắp, tài xế thường rơi vào trạng thái "giấc ngủ trắng" (microsleep) do sự monotony và mệt mỏi kéo dài, dẫn đến nguy cơ tai nạn cao.

Vấn đề đặt ra là cần phát triển một hệ thống sử dụng công nghệ Computer Vision (Thị giác máy tính) kết hợp trí tuệ nhân tạo (AI) để phát hiện kịp thời các dấu hiệu buồn ngủ hoặc mất tập trung của tài xế, từ đó đưa ra cảnh báo nhằm giảm thiểu nguy cơ tai nạn giao thông. Đề tài này không chỉ mang tính cấp bách mà còn có ý nghĩa thực tiễn trong việc bảo vệ tính mạng con người và nâng cao an toàn trên các tuyến đường.

2. Bối cảnh và động lực

Thực trạng tai nạn giao thông: Tai nạn giao thông do buồn ngủ hoặc mất tập trung là một trong những nguyên nhân hàng đầu gây tử vong và thương tích trên đường bộ toàn cầu. Theo Tổ chức Y tế Thế giới (WHO), mỗi năm có khoảng 1,3 triệu người thiệt mạng do tai nạn giao thông, trong đó một phần đáng kể liên quan đến sự mệt mỏi của tài xế.

Hạn chế của các phương pháp truyền thống: Các hệ thống an toàn hiện tại như cảnh báo tốc độ, cảm biến va chạm hay hệ thống phanh khẩn cấp chỉ hoạt động khi tai nạn sắp xảy ra, nhưng không thể dự đoán hoặc ngăn chặn sớm tình trạng mệt mỏi của tài xế – yếu tố gốc rễ của nhiều vụ tai nạn.

Tiềm năng của công nghệ AI: Sự phát triển của Computer Vision và Deep Learning mở ra cơ hội theo dõi trực tiếp trạng thái tài xế thông qua hình ảnh hoặc video từ camera. Các mô hình AI có khả năng nhận diện các dấu hiệu mệt mỏi như nhắm mắt lâu, ngáp, hoặc gật đầu, từ đó cung cấp giải pháp chủ động hơn so với các phương pháp truyền thống.

3. 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.

II. Cơ sở lý thuyết và phương pháp nghiên cứu

1. Dữ liệu đầu vào và đầu ra

- Đầu vào:

Hình ảnh hoặc video: Được thu thập từ camera giám sát gắn trong cabin xe, hướng về phía tài xế.

Đặc trưng sinh học: Tọa độ và đặc trưng của các điểm chính trên khuôn mặt (mắt, miệng, tư thế đầu) được trích xuất từ hình ảnh/video.

- Đầu ra:

Nhãn phân loại: Trạng thái của tài xế được phân loại thành "Tỉnh táo" (Awake) , "Buồn ngủ" (Drowsy).

- Cảnh báo: Hệ thống kích hoạt âm thanh (chuông báo), rung (trên vô-lăng hoặc ghế), hoặc tín hiệu đèn khi phát hiện trạng thái buồn ngủ kéo dài vượt quá ngưỡng thời gian cho phép (ví dụ: 2-3 giây).

2. Các phương pháp giải quyết

2.1. Phương pháp truyền thống (Classical Computer Vision):

- Công cụ sử dụng: OpenCV và Dlib để phát hiện và theo dõi các đặc điểm khuôn mặt.
- Eye Aspect Ratio (EAR): Tỷ lệ khung mắt được tính toán dựa trên khoảng cách giữa các điểm mốc (landmarks) quanh mắt. Khi EAR giảm dưới ngưỡng nhất định (thường là 0.2-0.3), hệ thống xác định mắt đang nhắm.

Eye Aspect Ratio được định nghĩa như sau:

$$EAR = (||p2 - p6|| + ||p3 - p5||) / (2 * ||p1 - p4||)$$

Trong đó p1, p2, ..., p6 là các landmark points quanh mắt theo thứ tự từ trái qua phải. Khi mắt mở, EAR có giá trị khoảng 0.25-0.3. Khi mắt nhắm, EAR giảm xuống gần 0. Ngưỡng phổ biến được sử dụng là 0.2-0.25.

Ưu điểm:

- Đơn giản, dễ implement
- Tính toán nhanh, phù hợp real-time
- Không yêu cầu training data

Nhược điểm:

- Nhạy cảm với chất lượng landmark detection
- Không thích nghi được với các đặc điểm khuôn mặt khác nhau
- Hiệu suất giảm trong điều kiện ánh sáng kém

Mouth Aspect Ratio (MAR): Tỷ lệ khung miệng được sử dụng để phát hiện hành vi ngáp. Tương tự EAR, MAR được tính toán dựa trên các landmark points quanh miệng:

```
MAR = (||p14 - p18|| + ||p15 - p17||) / (2 * ||p13 - p16||)
```

MAR được sử dụng để phát hiện hành vi ngáp. Khi không ngáp, MAR thường có giá trị 0.5-0.6. Khi ngáp, MAR tăng lên 0.8-1.0 hoặc cao hơn.

Head Pose Estimation: Phân tích tư thế đầu (head pose) được sử dụng để phát hiện việc gật đầu - một dấu hiệu phổ biến của buồn ngủ. Các góc Euler (pitch, yaw, roll) được tính toán từ facial landmarks để xác định hướng và độ nghiêng của đầu

Ưu điểm: Tính toán nhanh, không yêu cầu phần cứng mạnh.

Hạn chế: Độ chính xác giảm trong điều kiện ánh sáng yếu hoặc khi tài xế đeo kính, khẩu trang.

2.2 .Phương pháp Deep Learning:

2.2.1 Convolutional Neural Networks (CNN)

CNN đã revolutionize lĩnh vực computer vision kể từ khi AlexNet giành chiến thắng tại ImageNet 2012. Các kiến trúc CNN nổi tiếng bao gồm:

LeNet-5 (1998): Kiến trúc CNN đầu tiên, đề xuất bởi Yann LeCun AlexNet (2012): Đánh dấu bước ngoặt của deep learning trong computer vision VGGNet (2014): Chứng minh hiệu quả của việc sử dụng nhiều layer với filter size nhỏ ResNet (2015): Giải quyết vấn đề vanishing gradient thông qua residual connections EfficientNet (2019): Tối ưu hóa cân bằng giữa depth, width, và resolution

Trong bài toán phân loại facial expression và drowsiness detection, CNN đã được chứng minh hiệu quả cao. Li et al. (2018) sử dụng CNN để phân loại facial expressions đạt accuracy 89.4% trên dataset FER2013.

2.2.2 Vision Transformer (ViT)

Vision Transformer, được đề xuất bởi Dosovitskiy et al. (2020), đã mang cách mạng transformer architecture từ NLP sang computer vision. ViT chia ảnh thành các patches nhỏ, sau đó áp dụng transformer architecture để học relationships giữa các patches.

Kiến trúc ViT:

1. Patch Embedding: Chia ảnh thành 196 patches 16x16
2. Position Embedding: Thêm thông tin vị trí cho mỗi patch
3. Transformer Encoder: 12 layers với multi-head self-attention
4. Classification Head: MLP layer để classification

Ưu điểm của ViT:

- Khả năng capture long-range dependencies
- Hiệu suất cao trên large datasets
- Khả năng interpretability tốt thông qua attention maps

Nhược điểm:

- Yêu cầu large dataset để training hiệu quả
- Computational cost cao
- Inference time chậm hơn CNN

2.2.3 YOLO (You Only Look Once)

YOLO được thiết kế chủ yếu cho object detection, nhưng có thể được adapt cho classification tasks. YOLOv8, phiên bản mới nhất, cung cấp variant dành riêng cho classification (YOLOv8-cls).

Đặc điểm của YOLO:

- Single-stage detection/classification
- Real-time processing capability
- Good balance giữa speed và accuracy
- Lightweight architecture

III. Hiện thực hóa đề tài

1. Dữ liệu

Tập dữ liệu bao gồm các ảnh chụp đặc trưng khuôn mặt gồm mắt và miệng, được gắn nhãn thành bốn lớp:

- Nhắm mắt: Biểu thị trạng thái buồn ngủ.
- Mở mắt: Biểu thị trạng thái tỉnh táo.
- Ngáp: Gợi ý trạng thái mệt mỏi hoặc buồn ngủ.
- Không ngáp: Biểu thị trạng thái bình thường, không buồn ngủ.

Tiền xử lý dữ liệu: Chuẩn hóa ảnh, chuyển thành tensor và chuẩn hóa giá trị pixel. Áp dụng trên các vùng mắt và miệng được trích xuất bởi MediaPipe Face Mesh.

Tăng cường dữ liệu: Xoay ảnh (Rotation ±15°); Thay đổi độ sáng, tương phản để mô phỏng môi trường khác nhau; Dịch chuyển ảnh (Translation) để phản ánh vị trí đầu thay đổi; Làm mờ nhẹ (Gaussian Blur) để mô phỏng camera chất lượng kém.

2. Quy trình xây dựng mô hình

Quy trình phát triển mô hình được thực hiện theo các bước: chuẩn bị dữ liệu, chọn và huấn luyện mô hình, đánh giá, và tối ưu hóa.

2.1. Chuẩn bị dữ liệu

1. Làm sạch dữ liệu: Loại bỏ các ảnh bị lỗi hoặc gắn nhãn sai.
2. Chia dữ liệu: Phân chia thành tập huấn luyện (80%), tập xác thực (20%).
3. Tiền xử lý:

- Chuẩn hóa ảnh về 224x224, chuyển thành tensor, và chuẩn hóa giá trị pixel về [-1, 1].
- Áp dụng trên các vùng mắt và miệng được trích xuất bởi MediaPipe Face Mesh.

2.2. Huấn luyện mô hình:

Ba mô hình được chọn để huấn luyện và so sánh

2.2.1 Mô hình CNN

Kiến trúc mô hình: Mô hình CNN được thiết kế đơn giản với cấu trúc gồm hai lớp tích chập (Convolutional Layers), hai lớp gộp (Max Pooling), và hai lớp fully connected:

- Lớp tích chập:
    - Lớp đầu tiên nhận đầu vào ảnh RGB (3 kênh), xuất ra 32 kênh với kernel 3x3, padding=1.
    - Lớp thứ hai nhận 32 kênh, xuất ra 64 kênh, kernel 3x3, padding=1.
    - Mỗi lớp tích chập sử dụng hàm kích hoạt ReLU để tăng tính phi tuyến.
- Lớp gộp: Max Pooling (2x2, stride=2) được áp dụng sau mỗi lớp tích chập để giảm kích thước không gian, tối ưu hóa tính toán và tăng khả năng khái quát hóa.
- Lớp Dropout: Tỷ lệ 0.3, giúp giảm nguy cơ quá khớp (overfitting).
- Lớp Fully Connected:
    - Lớp đầu tiên giảm kích thước đặc trưng từ 645656 về 128 chiều.
    - Lớp cuối cùng xuất ra 4 chiều, tương ứng với 4 lớp: nhắm mắt, mở mắt, ngáp, không ngáp.

Cấu hình huấn luyện:

- Hàm mất mát: Categorical Cross-Entropy, phù hợp cho bài toán phân loại đa lớp.
- Bộ tối ưu: Adam với learning rate 0.001 (theo mã trước).
- Số epoch: [Giả định 50 epoch, dựa trên thông tin trước], sử dụng early stopping để ngăn quá khớp.

Thiết bị: Huấn luyện trên GPU (nếu có) hoặc CPU, sử dụng torch.device để tự động chọn thiết bị.

Kết quả huấn luyện: Mô hình được lưu dưới dạng trọng số sau khi huấn luyện, cho thấy quá trình huấn luyện đã hoàn tất. Độ chính xác dự kiến trên tập xác thực: ~85% (theo báo cáo trước), với F1-Score ~0.83 và thời gian suy luận ~20ms, phù hợp cho ứng dụng thời gian thực.

Đánh giá: Mô hình CNN có kiến trúc đơn giản, dễ triển khai trên thiết bị có tài nguyên hạn chế. Tuy nhiên độ chính xác thấp hơn so với các mô hình phức tạp hơn (như ViT), do thiếu tăng cường dữ liệu trong pipeline được cung cấp và kiến trúc đơn giản.

2.2.2. Mô hình ViT

Mô hình sử dụng là Vision Transformer (ViT) phiên bản vit_base_patch16_224 từ thư viện timm, được khởi tạo với trọng số huấn luyện trước (pretrained). Một số đặc điểm chính của mô hình:

- Đầu vào: Hình ảnh RGB với kích thước 224x224.
- Kiến trúc ViT: Chia hình ảnh thành các patch 16x16, sau đó mã hóa thành các vector nhúng và xử lý qua các tầng Transformer.
- Tùy chỉnh: Lớp patch_embed.proj được điều chỉnh để phù hợp với kích thước đầu vào, và lớp đầu ra được cấu hình cho bài toán phân loại 4 lớp.
- Mô hình được chuyển sang thiết bị tính toán (GPU nếu có, hoặc CPU).

Mô hình được huấn luyện với các thiết lập sau:

- Hàm mất mát: CrossEntropyLoss, phù hợp cho bài toán phân loại đa lớp.
- Bộ tối ưu hóa: AdamW với tốc độ học (learning rate) là 1e-4.
- Số epoch: 20.
- Batch size: 32.
- Tập dữ liệu: Dữ liệu được tải bằng DataLoader với chế độ xáo trộn cho tập huấn luyện và sử dụng 2 worker để tăng tốc xử lý.

Trong mỗi epoch, mô hình được:

- Huấn luyện: Tính toán hàm mất mát, lan truyền ngược (backpropagation), và cập nhật trọng số.
- Đánh giá: Kiểm tra hiệu suất trên tập kiểm tra để theo dõi độ chính xác và mất mát.

Quá trình huấn luyện cho thấy mô hình đạt được sự cải thiện đáng kể qua các epoch:

- Epoch 1: Độ chính xác tập huấn luyện đạt 72.44%, tập kiểm tra đạt 89.40%.
- Epoch 20: Độ chính xác tập huấn luyện tăng lên 98.84%, tập kiểm tra đạt 95.59%.
- Mất mát (loss) giảm dần qua các epoch, từ 0.6784 (huấn luyện) và 0.2980 (kiểm tra) ở epoch 1 xuống còn 0.0349 (huấn luyện) và 0.1635 (kiểm tra) ở epoch 20.

Kết quả cho thấy mô hình có khả năng học tốt trên tập huấn luyện và tổng quát hóa hiệu quả trên tập kiểm tra. Tuy nhiên, một số epoch (như epoch 19) cho thấy mất mát trên tập kiểm tra tăng nhẹ, có thể do hiện tượng quá khớp (overfitting) cục bộ.

- Hiệu suất: Độ chính xác cao (95.59% trên tập kiểm tra) cho thấy mô hình có khả năng nhận diện trạng thái buồn ngủ của tài xế với độ tin cậy cao.
- Tổng quát hóa: Sự chênh lệch giữa độ chính xác của tập huấn luyện (98.84%) và tập kiểm tra (95.59%) cho thấy mô hình có thể gặp một chút hiện tượng quá khớp, nhưng vẫn ở mức chấp nhận được.
- Tối ưu hóa: Việc sử dụng ViT với trọng số pretrained giúp mô hình hội tụ nhanh hơn và đạt hiệu suất cao ngay từ các epoch đầu.
- Hạn chế: Kích thước dữ liệu và sự đa dạng của các trạng thái (ví dụ: ánh sáng, góc quay, biểu cảm khuôn mặt) có thể ảnh hưởng đến khả năng tổng quát hóa của mô hình trong các tình huống thực tế.

2.2.3. Mô hình YOLO

Mô hình: Sử dụng YOLO11n (phiên bản nhẹ của YOLOv11) làm mô hình nền, với 2,592,570 tham số và 6.5 GFLOPs. Mô hình được khởi tạo với trọng số tiền huấn luyện (pretrained weights), nhưng được điều chỉnh để phù hợp với số lớp (14 lớp) của bài toán.
Cấu hình huấn luyện:

- Số epoch: 30.
- Kích thước ảnh: 640x640 pixel.
- Batch size: 16.
- Tối ưu hóa: AdamW với learning rate ban đầu là 0.000556, momentum 0.9.
- Kỹ thuật tăng cường dữ liệu: Áp dụng các phép biến đổi như làm mờ (Blur), chuyển thành ảnh xám (ToGray), và CLAHE với xác suất thấp (0.01).
- Các tham số khác: box loss (7.5), cls loss (0.5), dfl loss (1.5), và các phép biến đổi hình ảnh như fliplr (0.5), translate (0.1).

Quá trình huấn luyện được thực hiện trên phần cứng có GPU (Tesla P100-PCIE-16GB, 16,269 MiB). Các bước chính bao gồm:

1. Chuẩn bị dữ liệu: Bộ dữ liệu được tải về và lưu trữ tại thư mục datasets. Các nhãn được kiểm tra và tạo cache để tăng tốc độ truy cập.
2. Huấn luyện: Mô hình được huấn luyện trong 30 epoch, với mosaic augmentation được áp dụng trong 20 epoch đầu tiên và tắt trong 10 epoch cuối để tinh chỉnh. Các chỉ số loss (box, cls, dfl) giảm đều qua các epoch, thể hiện sự hội tụ tốt của mô hình.
3. Đánh giá: Sau mỗi epoch, mô hình được đánh giá trên tập xác thực với các chỉ số Precision (P), Recall (R), mAP50, và mAP50:95.

Hiệu suất tổng thể (sau 30 epoch):

- Precision: 0.444.
- Recall: 0.745.

- mAP50: 0.531.
- mAP50:95: 0.342.

Hiệu suất theo lớp:

- Một số lớp như "Drowsy" (với 147 mẫu) đạt Precision 0.967, Recall 0.993, mAP50 0.995, và mAP50:95 0.866, cho thấy khả năng phát hiện tốt các dấu hiệu buồn ngủ cụ thể.
- Một số lớp khác, ví dụ như "awake" (106 mẫu), có hiệu suất thấp hơn (mAP50: 0.195), có thể do số lượng mẫu hạn chế hoặc đặc điểm khó nhận diện.
- Một lớp "awake" (165 mẫu) không đạt được kết quả tốt (P, R, mAP50 đều bằng 0), có thể do dữ liệu không đủ hoặc nhãn không chính xác.

Tốc độ xử lý:

- Thời gian tiền xử lý: 0.2 ms/hình ảnh.
- Thời gian suy luận: 2.0 ms/hình ảnh.
- Thời gian hậu xử lý: 1.1 ms/hình ảnh.

Tổng thời gian huấn luyện: 1.514 giờ.

Ưu điểm:

- Mô hình cho thấy hiệu suất tốt trong việc nhận diện một số dấu hiệu buồn ngủ, đặc biệt với các lớp có dữ liệu phong phú.
- Tốc độ suy luận nhanh (2.0 ms/hình ảnh), phù hợp với ứng dụng thời gian thực như giám sát tài xế.
- Việc sử dụng YOLO11n đảm bảo mô hình nhẹ, dễ triển khai trên các thiết bị có tài nguyên hạn chế.

Hạn chế:

- Một số lớp có hiệu suất thấp do dữ liệu không cân bằng hoặc số lượng mẫu hạn chế.
- Lớp "awake" (165 mẫu) không đạt kết quả, có thể do vấn đề trong dữ liệu hoặc nhãn không đồng nhất.

- mAP50:95 (0.342) còn thấp, cho thấy mô hình cần cải thiện khả năng định vị chính xác các hộp giới hạn ở các ngưỡng IoU cao.

3. Đánh giá mô hình

Tiêu chí	CNN	ViT	YOLO
Độ chính xác	~85% (xác thực)	95.59% (kiểm tra)	mAP50: 0.531, mAP50:95: 0.342
Thời gian suy luận	~20ms	Chưa rõ, dự kiến chậm hơn	2.0 ms
Kích thước mô hình	Nhẹ	Lớn	Nhẹ (2.59M tham số)
Tài nguyên tính toán	Thấp	Cao	Thấp
Khả năng tổng quát hóa	Trung bình	Cao	Trung bình
Ứng dụng thời gian thực	Phù hợp	Hạn chế	Rất phù hợp

Kết quả:

CNN: Phù hợp cho các ứng dụng đơn giản, tài nguyên hạn chế, nhưng độ chính xác không cao bằng ViT và khả năng tổng quát hóa hạn chế do thiếu tăng cường dữ liệu.

ViT: Hiệu suất vượt trội, đặc biệt trong việc học các đặc trưng phức tạp, nhưng yêu cầu tài nguyên tính toán cao và có dấu hiệu quá khớp nhẹ.

YOLO: Tốc độ suy luận nhanh nhất, hiệu suất tốt trên các lớp dữ liệu phong phú, nhưng gặp khó khăn với các lớp dữ liệu không cân bằng.

Nhận xét

- CNN là lựa chọn tốt cho các thiết bị có tài nguyên hạn chế và yêu cầu triển khai nhanh. Tuy nhiên, để cải thiện hiệu suất, cần bổ sung tăng cường dữ liệu và tối ưu hóa kiến trúc.
- ViT mang lại hiệu suất cao nhất trong ba mô hình, phù hợp với các ứng dụng yêu cầu độ chính xác cao. Tuy nhiên, cần cải thiện khả năng tổng quát hóa bằng cách bổ sung dữ liệu đa dạng hơn và điều chỉnh để giảm quá khớp.

- YOLO là lựa chọn lý tưởng cho các ứng dụng thời gian thực nhờ tốc độ suy luận nhanh và kích thước mô hình nhẹ. Tuy nhiên, cần giải quyết vấn đề dữ liệu không cân bằng để cải thiện hiệu suất trên các lớp khó.

Khuyến nghị

- Tăng cường dữ liệu: Cả ba mô hình sẽ được cải thiện nếu áp dụng các kỹ thuật tăng cường dữ liệu mạnh mẽ hơn, đặc biệt là với các tình huống thực tế như ánh sáng yếu, góc quay đa dạng.
- Cân bằng dữ liệu: Đối với YOLO, cần tăng cường số lượng mẫu và kiểm tra nhãn của các lớp như "awake" để cải thiện hiệu suất.
- Kết hợp mô hình: Xem xét kết hợp CNN hoặc YOLO cho suy luận nhanh trên thiết bị hạn chế và ViT cho các ứng dụng yêu cầu độ chính xác cao.
- Tối ưu hóa tài nguyên: Đối với ViT, cần tối ưu hóa để giảm yêu cầu tính toán, chẳng hạn như sử dụng kỹ thuật pruning hoặc quantization.

3. Cơ chế cảnh báo

Khi phát hiện mắt nhắm lâu hơn 2 giây hoặc hành vi ngáp/gật đầu lặp lại, hệ thống kích hoạt:

- Âm thanh cảnh báo (chuông lớn dần).
- Rung động trên ghế hoặc vô-lăng.
- Thông báo qua màn hình hiển thị (HUD) nếu xe hỗ trợ.

IV. Ứng dụng thực tế và triển vọng

1. Ứng dụng thực tế

Ô tô cá nhân: Tích hợp vào xe hơi để giám sát tài xế trong các chuyến đi dài.

Xe thương mại: Áp dụng cho xe tải, xe khách, taxi để đảm bảo an toàn cho hành khách và hàng hóa.

Xe tự hành: Hỗ trợ hệ thống lái tự động bằng cách theo dõi trạng thái tài xế, chuyển đổi giữa chế độ tự động và thủ công khi cần.

Công ty vận tải: Giám sát đội ngũ tài xế, giảm thiểu rủi ro pháp lý và kinh tế do tai nạn.

2. Triển vọng phát triển

- Kết hợp đa cảm biến: Tích hợp với cảm biến nhịp tim, EEG (điện não đồ), hoặc nhiệt độ cơ thể để tăng độ chính xác trong việc phát hiện mệt mỏi.
- Cá nhân hóa: Điều chỉnh ngưỡng cảnh báo dựa trên thói quen lái xe của từng tài xế.
- Ứng dụng IoT: Kết nối hệ thống với đám mây để lưu trữ dữ liệu, phân tích hành vi lái xe dài hạn và gửi cảnh báo đến quản lý đội xe.

V. Thách thức

- Điều kiện môi trường: Ánh sáng yếu, góc quay không tối ưu, hoặc tài xế đeo kính/khẩu trang làm giảm độ chính xác.
- Hiệu suất thời gian thực: Xử lý video liên tục đòi hỏi phần cứng mạnh và tối ưu hóa thuật toán.
- Quyền riêng tư: Việc giám sát liên tục có thể gây lo ngại về bảo mật dữ liệu cá nhân.

VI. Kết luận:

Đề tài "Phát hiện tài xế buồn ngủ hoặc mất tập trung khi lái xe" mang lại một giải pháp khả thi và hiệu quả để giải quyết vấn đề an toàn giao thông liên quan đến mệt mỏi của tài xế. Bằng cách kết hợp Computer Vision và Deep Learning, hệ thống không chỉ phát hiện sớm các dấu hiệu nguy hiểm mà còn đưa ra cảnh báo kịp thời, góp phần giảm thiểu tai nạn giao thông. Trong tương lai, với sự phát triển của công nghệ và dữ liệu, hệ thống này có thể được nâng cấp để trở thành một phần không thể thiếu trong các phương tiện giao thông hiện đại, từ đó nâng cao chất lượng cuộc sống và bảo vệ an toàn cho cộng đồng.

VII. Danh mục tài liệu tham khảo

[1]. Khandave, A. (2020, September 20). *Driver drowsiness detection alert system with Open-CV & Keras using IP-WebCam for camera connection*. Retrieved from [https://www.linkedin.com/pulse/driver-drowsiness-detection-alert-system-open-cv-keras-khandave/]

[2]. Sahayadhas, A., Sundaraj, K., & Murugappan, M. (2012*). Detecting driver drowsiness based on sensors: A review*. Sensors, 12(12), 16937–16953. https://doi.org/10.3390/s121216937

[3]. Díaz-Santos, S., Cigala-Álvarez, Ó., Gonzalez-Sosa, E., Caballero-Gil, P., & Caballero-Gil, C. (2024). *Driver identification and detection of drowsiness while driving*. Applied Sciences, 14(6), 2603. https://www.mdpi.com/2076-3417/14/6/2603

[4]. Alshaqaqi, B., Baquhaizel, A. S., Ouis, M. E. A., Boumehed, M., Ouamri, A., & Keche, M. (2013). *Driver drowsiness detection system*. Trong 2013 8th International Workshop on Systems, Signal Processing and their Applications (WoSSPA). IEEE.

[5]. Liu, C. C., Hosking, S. G., & Lenné, M. G. (2009). *Predicting driver drowsiness using vehicle measures: Recent insights and future challenges*. Journal of Sleep Research, 18(3), 239-253. https://doi.org/10.1016/j.jsr.2009.04.005

[6]. Vural, E., Cetin, M., Ercil, A., Littlewort, G., Bartlett, M., & Movellan, J. (n.d.*). Drowsy Driver Detection Through Facial Movement Analysis*. Sabanci University; University of California San Diego.

[7]. Grace, R., Byrne, V. E., Bierman, D. M., Legrand, J.-M., Gricourt, D., & Davis, B. K. (1998). *A drowsy driver detection system for heavy vehicles*. In 17th DASC. AIAA/IEEE/SAE. Digital Avionics Systems Conference. Proceedings (Cat. No.98CH36267). IEEE.

[8]. Zhang, H., Liu, T., Lyu, J., & Chen, D. (2023). *Integrate memory mechanism in multi-granularity deep framework for driver drowsiness detection*. Intelligence & Robotics, 3(4), 614-631. https://doi.org/10.20517/ir.2023.34

[9].