Driver behavior monitoring system

Drowsiness detection / Fatigue detection

Computer vision / Motion detection

Driver vigilance monitoring

Human activity recognition (HAR)

Sleep deprivation effect on driving

AI in transportation safety

Driver monitoring system (DMS)

Smart vehicle safety systems

**🛠️ Tech Stack (Công nghệ sử dụng)**

| **Công nghệ** | **Mô tả** |
| --- | --- |
| **Python 3.x** | Ngôn ngữ lập trình chính |
| **OpenCV** | Xử lý hình ảnh và video |
| **Dlib** | Phát hiện khuôn mặt và nhận diện điểm đặc trưng trên khuôn mặt |
| **NumPy** | Xử lý mảng và tính toán số học |
| **SciPy** | Tính toán khoảng cách Euclid |
| **argparse** | Phân tích tham số dòng lệnh |
| **winsound** | Phát âm thanh cảnh báo (chỉ trên Windows) |
| **threading** | Xử lý bất đồng bộ cho cảnh báo âm thanh |
| **collections** | Sử dụng deque để lưu lịch sử PERCLOS hiệu quả |

**Kỹ thuật tiêu biểu**

* **Phát hiện buồn ngủ theo thời gian thực** dựa trên các chỉ số sinh trắc học khuôn mặt như EAR (Eye Aspect Ratio), MAR (Mouth Aspect Ratio), PERCLOS (Percentage of Eye Closure), đếm ngáp.
* **Cảnh báo bằng âm thanh** và/hoặc nháy màn hình khi phát hiện người dùng buồn ngủ hoặc ngáp (runtime alarm).
* **Xử lý khung hình hiệu quả**: resize khung hình để tăng tốc độ xử lý.
* **Theo dõi và hiển thị thông tin**: thời gian thực, thời lượng phiên, giá trị EAR, MAR, PERCLOS, số lần ngáp, trạng thái cảnh báo.
* **Điều khiển runtime** qua phím nóng: tắt/mở cảnh báo, chỉnh ngưỡng EAR, chuyển chế độ toàn màn hình, reset bộ đếm.

**Gợi ý mở rộng tương lai**

* Cảnh báo âm thanh đa nền tảng (không chỉ Windows).
* Ước lượng tư thế đầu (head pose estimation).
* Phát hiện trong điều kiện ánh sáng yếu, người đeo kính, khuôn mặt bị che khuất.
* Giao diện đồ họa (GUI) bằng Tkinter, PyQt, Kivy hoặc web framework.
* Phân tích tần suất chớp mắt/ngáp để đánh giá mức độ buồn ngủ.

**1. Phát hiện buồn ngủ qua mắt (Eye Aspect Ratio - EAR)**

* **EAR (Eye Aspect Ratio)**: Thuật toán tính tỉ lệ chiều cao/chều ngang mắt dựa trên 6 điểm landmark quanh mắt, để phát hiện mắt nhắm lâu (dấu hiệu buồn ngủ).
* **Công thức:** EAR = (|P2-P6| + |P3-P5|) / (2 \* |P1-P4|)
* Nếu EAR < ngưỡng (thường 0.23) liên tục qua số khung hình nhất định, sẽ cảnh báo buồn ngủ.

**2. Phát hiện ngáp qua miệng (Mouth Aspect Ratio - MAR)**

* **MAR (Mouth Aspect Ratio)**: Tính tỉ lệ chiều cao/chều ngang miệng bằng landmark quanh miệng, MAR tăng cao khi ngáp.
* Nếu MAR > ngưỡng (thường 0.6), hệ thống ghi nhận một lần ngáp.

**3. Phát hiện & theo dõi khuôn mặt**

* Sử dụng **Dlib** kết hợp mô hình "shape\_predictor\_68\_face\_landmarks.dat" để phát hiện khuôn mặt và 68 điểm landmark (mắt, miệng, mũi...).

**4. Giám sát PERCLOS (Percentage of Eye Closure)**

* **PERCLOS**: Tính tỉ lệ phần trăm thời gian mắt nhắm trong một cửa sổ thời gian (ví dụ 60 giây), giúp đánh giá mức độ buồn ngủ dài hạn.

**5. Cảnh báo (Alerting)**

* Hệ thống cảnh báo bằng:
  + **Visual**: Hiển thị chữ "DROWSINESS ALERT!" hoặc "YAWN DETECTED!" trên màn hình, nháy màu.
  + **Audio**: Phát âm thanh (chỉ trên Windows) khi có sự kiện buồn ngủ/ngáp, cảnh báo lặp lại khi trạng thái tiếp diễn.

**6. Các thuật toán phụ trợ**

* **Tính khoảng cách Euclid** (qua SciPy) để đo các đoạn nối giữa các landmark.
* **Cửa sổ trượt (Sliding window)** dùng collections.deque để lưu trạng thái mắt trong tính toán PERCLOS.

**Cách hoạt động của hệ thống phát hiện buồn ngủ:**

1. **Thu thập dữ liệu**

# Lấy hình ảnh từ camera

gray\_frame = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

detected\_faces = self.face\_detector(gray\_frame)

1. **Phát hiện khuôn mặt & điểm mốc**

# Phát hiện khuôn mặt

face = detected\_faces[0]

xf,yf,wf,hf = face.left(),face.top(),face.width(),face.height()

# Phát hiện 68 điểm mốc trên khuôn mặt

lndmks = self.landmark\_predictor(gray\_frame,face)

1. **Tính toán các chỉ số**

# Tính EAR (Eye Aspect Ratio - Tỷ lệ khía cạnh mắt)

avg\_e = (eye\_aspect\_ratio(le) + eye\_aspect\_ratio(re))/2.0

# Tính MAR (Mouth Aspect Ratio - Tỷ lệ miệng)

mar\_v = mouth\_ratio(mo)

# Tính PERCLOS (% thời gian nhắm mắt)

perclos\_value = (closed\_frames/total\_frames)\*100

1. **Logic phát hiện buồn ngủ**

if avg\_e < ear\_threshold:  # Nếu mắt nhắm

    eye\_blink\_counter += 1

    if eye\_blink\_counter >= ear\_consec\_frames:

        drowsiness\_alert\_visual\_active = True

**Các thành phần AI chính:**

1. **Face Detector**

* Sử dụng HOG + Linear SVM hoặc CNN
* Xác định vị trí khuôn mặt trong khung hình

1. **Landmark Predictor**

* Sử dụng mô hình học sâu
* Xác định 68 điểm mốc trên khuôn mặt
* Dùng để tính toán EAR, MAR

1. **Ngưỡng thích nghi**

* EAR threshold: Ngưỡng nhận biết mắt nhắm
* MAR threshold: Ngưỡng nhận biết ngáp
* PERCLOS: Đánh giá mức độ buồn ngủ tổng thể

**Luồng xử lý:**

Camera → Face Detection → Landmark Detection → Tính toán chỉ số → Phát hiện buồn ngủ → Cảnh báo

Chương 4: Ứng Dụng Trí Tuệ Nhân Tạo vào Hệ Thống Phát Hiện Buồn Ngủ

4.1. Tích Hợp AI và Vai Trò Trong Nhận Diện Hành Vi

Hệ thống phát hiện buồn ngủ sử dụng AI để nhận diện các hành vi như ngáp, mệt mỏi, nhắm mắt từ hình ảnh camera thời gian thực.

4.1.1. Tự động phát hiện và theo dõi khuôn mặt

Nhận diện chính xác khuôn mặt trong nhiều điều kiện ánh sáng, góc nhìn.

4.1.2. Nhận diện đặc trưng hình thái học khuôn mặt

Xác định các điểm mốc quan trọng phục vụ phân tích trạng thái mắt, miệng.

4.2. Mô Hình Học Sâu Phát Hiện Trạng Thái Mệt Mỏi

Hệ thống gồm 3 thành phần AI chính, sử dụng công nghệ học sâu để nhận diện dấu hiệu buồn ngủ.

4.2.1. Face Detector (Bộ Phát Hiện Khuôn Mặt)

Công nghệ: HOG + SVM hoặc CNN.

Vai trò: Xác định vị trí khuôn mặt trong mỗi khung hình.

4.2.2. Landmark Predictor (Bộ Dự Đoán Điểm Mốc)

Công nghệ: Dlib với 68 điểm mốc.

Vai trò: Xác định chính xác mắt, miệng, mũi... để phân tích trạng thái.

4.2.3. Tính Toán Các Chỉ Số Đặc Trưng

EAR (Eye Aspect Ratio): Phát hiện mắt nhắm lâu → nghi ngờ buồn ngủ.

MAR (Mouth Aspect Ratio): Nhận diện ngáp qua độ mở miệng.

PERCLOS: Tỷ lệ thời gian mắt nhắm, đánh giá mức độ buồn ngủ.

4.3. Xây Dựng Mức Cảnh Báo Nguy Hiểm Theo Hành Vi

Hệ thống phân tích hành vi để đưa ra cảnh báo theo mức độ nguy hiểm.

4.3.1. Logic phát hiện buồn ngủ

EAR thấp hoặc PERCLOS cao → nghi ngờ buồn ngủ.

MAR cao đột ngột → phát hiện ngáp.

4.3.2. Phân loại mức độ cảnh báo

Cảnh báo ban đầu: Khi mới xuất hiện dấu hiệu.

Cảnh báo nguy hiểm cao: Khi hành vi kéo dài hoặc lặp lại nhiều lần.

4.4. Hệ Thống Cảnh Báo: Âm Thanh, Hình Ảnh, Kết Nối IoT

Cảnh báo được kích hoạt đa dạng và có thể kết nối mạng.

4.4.1. Cảnh báo đa phương tiện

Bằng âm thanh (buzzer, loa), hình ảnh (đèn, màn hình cảnh báo).

4.4.2. Kết nối IoT

Gửi cảnh báo đến trung tâm giám sát hoặc thiết bị khác qua mạng.

Tóm Tắt Vai Trò và Điểm Mạnh của AI

Tự động hóa toàn diện việc phát hiện mệt mỏi từ hình ảnh.

Nâng cao độ chính xác nhờ mô hình học sâu.

Cá nhân hóa qua điều chỉnh ngưỡng phân tích.

Dễ tích hợp vào hệ thống IoT và các thiết bị cảnh báo.

➡️ AI đóng vai trò cốt lõi, giúp hệ thống trở nên thông minh, chủ động, linh hoạt, góp phần giảm thiểu tai nạn giao thông do mệt mỏi hay ngủ gật.