

4. Descripción detallada del invento

Sistema de Detección de Somnolencia del Conductor en Tiempo Real mediante Procesamiento Multi-Modal de Características Faciales y Gestuales

Concepto Inventivo Central

La presente invención consiste en un sistema computacional integrado para la detección automática de signos de somnolencia en conductores mediante el análisis en tiempo real de múltiples biomarcadores visuales extraídos de video streaming. El concepto inventivo central reside en la integración sincronizada de cinco módulos de procesamiento independientes que analizan simultáneamente: (1) parpadeos y microsueños a través de la medición de distancias entre párpados superiores e inferiores, (2) bostezos mediante el cálculo de la relación entre distancia labial y distancia del mentón, (3) fricción ocular detectando la proximidad de dedos a las regiones oculares, (4) inclinación de cabeza (pitch) mediante análisis de posicionamiento relativo de puntos faciales clave, y (5) la generación automática de reportes estadísticos con ventanas temporales configurables. La innovación principal radica en el procesamiento paralelo de estas características utilizando técnicas de visión por computadora basadas en MediaPipe, combinado con algoritmos de detección temporal que identifican patrones de fatiga mediante umbrales de duración específicos para cada biomarcador, permitiendo una evaluación integral y no invasiva del estado de alerta del conductor.

Descripción Funcional de los Componentes

4.1. Módulo de Extracción de Puntos Clave (PointsExtractor)

Este módulo constituye la primera etapa del sistema y se encarga de la detección y localización de puntos anatómicos faciales y de las manos mediante el procesamiento de frames de video en tiempo real. Utiliza dos subprocesadores especializados:

4.1.1. FaceMeshProcessor: Implementa la detección facial mediante MediaPipe Face Mesh, extrayendo 468 puntos faciales tridimensionales que incluyen contornos oculares (puntos superiores e inferiores de párpados derecho e izquierdo), puntos labiales (comisuras y puntos centrales superiores e inferiores), puntos nasales (punta de nariz y región frontal), y puntos de mejillas (derecha e izquierda). La función particular de este componente es generar coordenadas normalizadas (x, y, z) de estos puntos que servirán como entrada para los módulos de procesamiento posterior.

4.1.2. HandsProcessor: Detecta y rastrea hasta dos manos simultáneamente utilizando MediaPipe Hands, extrayendo 21 puntos por mano que incluyen puntos de la muñeca y de cada una de las cinco falanges (pulgar, índice, medio, anular y meñique). Este procesador calcula las distancias euclidianas entre cada punto de los dedos y los puntos oculares detectados por el FaceMeshProcessor, generando matrices de proximidad que permiten identificar cuando una mano se aproxima a menos de 40 píxeles de distancia de una región ocular.

4.1.3. Integración de Puntos: El componente PointsExtractor fusiona los puntos faciales y manuales en una estructura de datos unificada, verificando la validez de la detección facial como condición necesaria para el procesamiento posterior. Si la detección facial falla, el sistema retorna un estado de control que interrumpe el flujo de procesamiento.

4.2. Módulo de Procesamiento de Puntos (PointsProcessing)

Este módulo transforma las coordenadas crudas de los puntos detectados en métricas geométricas calculadas mediante distancias euclidianas, preparando los datos para el análisis de características de somnolencia.

4.2.1. EyesProcessor: Calcula cuatro distancias clave para cada ojo: (a) distancia del párpado superior derecho, (b) distancia del párpado superior izquierdo, (c)

distancia del párpado inferior derecho, y (d) distancia del párpado inferior izquierdo. Estas distancias se obtienen midiendo la separación entre puntos específicos de los contornos palpebrales superiores e inferiores. La función de estas métricas es determinar el estado de apertura/cierre ocular mediante comparación relativa: cuando la distancia del párpado superior es menor que la del inferior, los ojos están cerrados.

4.2.2. MouthProcessor: Procesa puntos labiales para calcular dos distancias críticas: (a) distancia labial (separación vertical entre puntos superior e inferior de los labios) y (b) distancia del mentón (distancia desde un punto labial inferior hasta un punto del mentón). La relación entre estas distancias permite determinar el estado de apertura bucal: cuando la distancia labial excede la distancia del mentón, la boca está abierta, indicando potencial bostezo.

4.2.3. HeadProcessor: Analiza la orientación de la cabeza mediante el cálculo de: (a) distancia nariz-boca, (b) distancia nariz-frente, y (c) posicionamiento relativo de puntos nasales respecto a puntos de mejillas. Comparando estas métricas, el sistema determina si la cabeza está inclinada hacia abajo (pitch negativo), hacia arriba (pitch positivo), o hacia los lados (roll). Específicamente, cuando el punto nasal se posiciona entre los puntos de mejillas y la distancia nariz-boca es menor que la distancia nariz-frente, se detecta inclinación hacia abajo.

4.2.4. FirstHandProcessor y SecondHandProcessor: Estos procesadores calculan distancias entre cada uno de los cinco dedos de cada mano detectada y los puntos oculares (derecho e izquierdo por separado). Generan estructuras de datos que contienen las distancias mínimas entre cada falange y cada región ocular, permitiendo la detección de fricción ocular cuando alguna de estas distancias cae por debajo del umbral de 40 píxeles.

4.3. Módulo de Procesamiento de Características de Somnolencia (FeaturesDrowsinessProcessing)

Este módulo constituye el núcleo analítico del sistema, implementando algoritmos temporales que convierten las métricas geométricas en eventos de somnolencia mediante la aplicación de umbrales de duración y técnicas de detección de patrones.

4.3.1. FlickerEstimator (Detección de Parpadeos y Microsueños): Este procesador monitorea continuamente el estado de apertura/cierre ocular mediante

la comparación de las cuatro distancias palpebrales calculadas por el EyesProcessor. Implementa dos algoritmos de detección:

- **Detección de Parpadeos:** Identifica transiciones rápidas de estado ocular (cierre seguido de apertura) mediante una máquina de estados binaria que detecta cuando la condición de ojos cerrados (distancias superiores $<$ distancias inferiores en ambos ojos) cambia a ojos abiertos. Cada transición completa incrementa un contador de parpadeos.
- **Detección de Microsueños:** Utiliza un algoritmo de temporización que activa un cronómetro cuando se detecta cierre ocular y lo detiene cuando los ojos se reabren. Si la duración del cierre es igual o superior a 2 segundos, se registra como un evento de microsueño, incluyendo su duración exacta. Los reportes de parpadeos se generan cada 60 segundos, mientras que los microsueños se reportan inmediatamente cuando se detectan.

4.3.2. YawnEstimator (Detección de Bostezos): Procesa las métricas del MouthProcessor para detectar eventos de bostezo mediante un algoritmo de dos etapas:

- **Detección de Apertura Bucal:** Compara continuamente la distancia labial con la distancia del mentón. Cuando la distancia labial excede la distancia del mentón, se marca el inicio de una apertura bucal.
- **Detección de Bostezo:** Utiliza una máquina de estados temporales que inicia un cronómetro cuando se detecta apertura bucal y lo detiene cuando la boca se cierra. Si la duración de la apertura es superior a 4 segundos, se registra como bostezo. Los bostezos se contabilizan y sus duraciones se almacenan en una lista. Los reportes se generan cada 180 segundos (3 minutos), incluyendo el conteo total y las duraciones de cada evento.

4.3.3. EyeRubEstimator (Detección de Fricción Ocular): Analiza las distancias calculadas por los procesadores de manos para detectar cuando los dedos entran en contacto con las regiones oculares. El algoritmo implementa detección independiente para cada ojo y cada mano:

- **Detección de Proximidad:** Evalúa continuamente si alguna de las distancias entre falanges y puntos oculares es menor a 40 píxeles para cada ojo (derecho e izquierdo) y cada mano (primera y segunda).
- **Detección de Fricción:** Utiliza temporizadores independientes para cada combinación mano-ojo. Cuando se detecta proximidad, se inicia un cronómetro específico. Si la proximidad se mantiene por más de 1 segundo, se registra como un evento de fricción ocular, incluyendo la duración y el lado afectado

(derecho o izquierdo). Los reportes se generan cada 300 segundos (5 minutos), consolidando los eventos de ambas manos y ambos ojos.

4.3.4. PitchEstimator (Detección de Inclinación de Cabeza): Procesa las métricas del HeadProcessor para detectar inclinaciones sostenidas de la cabeza hacia abajo, que pueden indicar somnolencia:

- **Determinación de Posición de Cabeza:** Evalúa la posición relativa del punto nasal respecto a los puntos de mejillas y compara las distancias nariz-boca y nariz-frente. Determina tres estados posibles: cabeza inclinada hacia abajo a la derecha, cabeza inclinada hacia abajo a la izquierda, o cabeza erguida.
- **Detección de Pitch Sostenido:** Cuando se detecta inclinación hacia abajo, se activa un cronómetro. Si la inclinación se mantiene por 3 segundos o más, se registra como un evento de pitch, incluyendo la duración y la dirección de la inclinación. Los eventos de pitch se reportan inmediatamente cuando se detectan.

4.4. Módulo de Visualización (ReportVisualizer)

Este componente genera representaciones visuales en tiempo real sobre los frames de video procesados, superponiendo anotaciones gráficas que incluyen: (a) dibujo de mallas faciales y de manos detectadas, (b) etiquetas de texto indicando el estado actual de cada biomarcador (abierto/cerrado para ojos y boca, posición de cabeza), (c) contadores de eventos (número de parpadeos, bostezos, fricciones oculares, microsueños, inclinaciones), y (d) alertas visuales cuando se detectan eventos críticos de somnolencia (microsueños, bostezos frecuentes, inclinaciones sostenidas).

4.5. Módulo de Generación de Reportes (DrowsinessReports)

Este módulo gestiona la persistencia y exportación de datos de somnolencia mediante dos mecanismos:

4.5.1. Almacenamiento en CSV: Los eventos detectados se registran en archivos CSV con formato estructurado que incluye timestamps, tipos de evento, duraciones, y conteos acumulados. Los reportes se organizan temporalmente según ventanas de análisis configuradas para cada biomarcador.

4.5.2. Generación de Reportes JSON: Convierte los datos procesados en formato JSON para transmisión en tiempo real a través de la interfaz de red, permitiendo la integración con sistemas externos de monitoreo o alerta.

4.6. Arquitectura de Comunicación (API WebSocket)

El sistema implementa una arquitectura cliente-servidor basada en WebSockets para permitir el procesamiento remoto de video streaming:

4.6.1. Servidor FastAPI: Implementa un endpoint WebSocket (`/ws`) que recibe frames de video codificados en Base64, los decodifica, procesa mediante el DrowsinessDetectionSystem, y retorna resultados que incluyen: (a) el frame original procesado, (b) un sketch con anotaciones visuales, y (c) un reporte JSON con todas las métricas de somnolencia detectadas.

4.6.2. Cliente de Video: Captura frames de una fuente de video (cámara web o archivo), los codifica en JPEG y Base64, los transmite al servidor vía WebSocket, y recibe los resultados procesados para visualización local.

4.7. Interfaz Gráfica de Usuario (GUI)

Desarrollada con Flet, la interfaz proporciona tres páginas principales: (a) página de inicio para inicialización del sistema, (b) página de selección de interfaz para configurar la fuente de video, y (c) página de detección de somnolencia que muestra el video procesado en tiempo real junto con paneles de información de métricas y alertas.

Relaciones entre Componentes

El flujo de procesamiento sigue una arquitectura de pipeline secuencial donde cada módulo depende de la salida del módulo anterior:

- 1. Extracción → Procesamiento de Puntos:** El PointsExtractor proporciona coordenadas de puntos clave al PointsProcessing, que las transforma en distancias geométricas.
- 2. Procesamiento de Puntos → Características de Somnolencia:** Las distancias calculadas alimentan los estimadores de características (FlickerEstimator, YawnEstimator, EyeRubEstimator, PitchEstimator), que aplican algoritmos temporales para detectar eventos.
- 3. Características → Visualización y Reportes:** Los eventos detectados se envían simultáneamente al ReportVisualizer (para anotación visual) y al DrowsinessReports (para almacenamiento y exportación).
- 4. Integración con Red:** El DrowsinessDetectionSystem orquesta todos los módulos y se integra con la API WebSocket para procesamiento remoto, mientras que la GUI proporciona la interfaz de usuario final.

Parámetros de Operación

Los umbrales y parámetros críticos del sistema son:

- **Umbral de proximidad para fricción ocular:** 40 píxeles
- **Duración mínima de microsueño:** 2 segundos
- **Duración mínima de bostezo:** 4 segundos
- **Duración mínima de fricción ocular:** 1 segundo
- **Duración mínima de inclinación de cabeza (pitch):** 3 segundos
- **Ventana de reporte de parpadeos:** 60 segundos
- **Ventana de reporte de bostezos:** 180 segundos
- **Ventana de reporte de fricción ocular:** 300 segundos
- **Resolución de video:** Configurable (por defecto 1920x1080)
- **Frecuencia de procesamiento:** Tiempo real (limitada por la tasa de frames de la fuente de video)

Ventajas Técnicas

El sistema ofrece las siguientes ventajas técnicas respecto a soluciones previas:

1. **Procesamiento Multi-Modal:** Integra cinco biomarcadores independientes que proporcionan una evaluación más robusta y precisa que sistemas basados en un solo indicador.
2. **Detección en Tiempo Real:** Utiliza algoritmos optimizados y arquitectura WebSocket para procesamiento con latencia mínima, permitiendo alertas inmediatas.
3. **No Invasividad:** No requiere sensores físicos adicionales, utilizando únicamente cámaras estándar y procesamiento por software.
4. **Modularidad y Extensibilidad:** Arquitectura modular permite agregar nuevos biomarcadores o modificar umbrales sin afectar el resto del sistema.
5. **Persistencia de Datos:** Generación automática de reportes estructurados para análisis histórico y cumplimiento normativo.