

Detección de Fatiga y Sueño en Conductores



Erick Huallpa Vargas

Universidad Privada Domingo Savio

Facultad de Ingeniería de Sistemas

Curso: Inteligencia Artificial I

Potosí, 30 de mayo de 2025

Documentación del Proyecto: Detección de Fatiga y Sueño en Conductores

Introducción

El presente documento describe el desarrollo de un sistema de monitoreo basado en visión artificial y redes neuronales que permite detectar signos de fatiga o somnolencia en conductores en tiempo real. Utilizando técnicas de procesamiento de imágenes y aprendizaje automático, el sistema identifica patrones de parpadeo, movimiento facial y distracción del conductor, activando una alerta sonora cuando se detecta un posible estado de sueño.

Problema a Solucionar

La fatiga del conductor es una causa significativa de accidentes de tránsito. Conductores somnolientos pueden perder momentáneamente el control del vehículo, poniendo en riesgo su vida y la de otros. El objetivo es detectar señales tempranas de somnolencia o falta de atención (como ojos cerrados por varios segundos o ausencia de movimiento facial) y generar una alerta a tiempo.

Forma de Resolver

La solución consiste en:

1. **Capturar el rostro del conductor** mediante la cámara y detectar puntos clave faciales usando **MediaPipe**.
2. Calcular el **EAR (Eye Aspect Ratio)** de ambos ojos para determinar si están cerrados.
3. Estimar el **movimiento facial** mediante la variación de la posición de la nariz.
4. Utilizar un **modelo de red neuronal entrenado** para clasificar el estado del conductor como "despierto" o "durmiendo".
5. En caso de detección de somnolencia, **activar una alarma sonora** para alertar al conductor.
6. De forma paralela, se analiza un segundo video para detectar objetos cercanos mediante **YOLOv5**, capturando evidencia visual si hay riesgo.

Requerimientos Técnicos y de Ambiente

- Sistema Operativo: Windows/Linux
- Python: 3.8 o superior
- Cámara web o video simulado (MP4)
- Almacenamiento para modelo, capturas y dataset

- GPU (opcional, recomendado para YOLOv5)

Librerías Utilizadas

Las principales librerías utilizadas en el proyecto son:

- `opencv-python`: procesamiento de video
- `mediapipe`: detección de puntos faciales
- `numpy`: cálculo de distancias y EAR
- `tensorflow.keras`: creación y entrenamiento de la red neuronal
- `joblib`: serialización del escalador de datos
- `pygame`: reproducción de audio para la alarma
- `torch`: inferencia con modelo YOLOv5
- `pandas`: manipulación del dataset
- `sklearn`: escalado de datos y partición de entrenamiento/prueba

Algoritmos

a) EAR (Eye Aspect Ratio):

Se calcula con la fórmula:

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

Donde los puntos corresponden a coordenadas de los ojos detectados con MediaPipe.

b) Movimiento Facial:

Se estima como la distancia Euclidiana entre la posición actual de la nariz y su posición en el frame anterior.

c) Clasificación del Estado:

Se utiliza una red neuronal simple con dos capas ocultas que recibe como entrada:

- EAR de ojo izquierdo
- EAR de ojo derecho
- EAR promedio
- Movimiento facial

La salida es una predicción categórica (0 = despierto, 1 = durmiendo).

Entrenamiento

Se generó un dataset simulado con 1000 muestras (500 por clase), usando funciones aleatorias controladas para simular:

- Ojos abiertos con movimiento (estado despierto)
- Ojos cerrados con bajo movimiento (estado durmiendo)

El modelo fue entrenado con:

- 2 capas ocultas de 16 neuronas (ReLU)
- Función de pérdida: `sparse_categorical_crossentropy`
- Optimizador: Adam
- 30 épocas
- División de datos: 80% entrenamiento, 20% prueba

Validación de la Red Neuronal

Durante el entrenamiento se observó:

- Precisión de validación $> 95\%$
- Bajo overfitting debido al tamaño pequeño y controlado del dataset
- Comprobación empírica en video con resultados coherentes con el

comportamiento del conductor

Se utilizó `StandardScaler` para normalizar los valores de entrada, manteniendo consistencia entre entrenamiento e inferencia en tiempo real.

Conclusiones

El sistema desarrollado demuestra que es posible detectar signos de somnolencia en conductores utilizando visión artificial y redes neuronales simples. Su implementación en tiempo real con alarmas y monitoreo de objetos ofrece un enfoque integral para mejorar la seguridad vial.

Entre sus principales ventajas se encuentran:

- Bajo requerimiento computacional
- Modularidad y extensibilidad (puede integrarse a vehículos reales)
- Capacidad de entrenamiento personalizado con datasets reales en

el futuro

Como mejoras futuras se sugiere:

- Recolección de datos reales para mejorar precisión
- Inclusión de más señales fisiológicas (como pestañeo, bostezo, inclinación de cabeza)
- Integración con sistemas de control vehicular (frenado automático, reducción de velocidad)

Repositorio

El código fuente completo del proyecto está disponible en el siguiente repositorio:

https://github.com/ErickHuallpa/proyecto_deteccion.git