

Manual de Usuario

Sistema de Detección de Somnolencia del Conductor

Versión 1.0

Fecha: 2024

Índice

- [1. Introducción](#)
 - [2. Requisitos del Sistema](#)
 - [3. Instalación](#)
 - [4. Inicio del Sistema](#)
 - [5. Guía de Uso](#)
 - [6. Interpretación de Resultados](#)
 - [7. Solución de Problemas](#)
 - [8. Preguntas Frecuentes](#)
-

1. Introducción

El Sistema de Detección de Somnolencia del Conductor es una aplicación de software diseñada para monitorear en tiempo real el estado de alerta de un conductor mediante el análisis de video en tiempo real. El sistema detecta automáticamente signos de somnolencia como parpadeos, microsueños, bostezos, fricción ocular e inclinación de cabeza.

1.1 Características Principales

- **Detección en Tiempo Real:** Análisis continuo de video mediante cámaras web estándar
- **Múltiples Biomarcadores:** Detección simultánea de 5 tipos de indicadores de somnolencia
- **Interfaz Gráfica Intuitiva:** Interfaz de usuario desarrollada con Flet
- **Reportes Automáticos:** Generación de reportes en formato CSV y JSON
- **Visualización en Tiempo Real:** Muestra el video original y procesado simultáneamente

1.2 Biomarcadores Detectados

1. **Parpadeos (Flickers):** Detección de parpadeos rápidos de los ojos
2. **Microsueños:** Detección de cierres oculares prolongados (≥ 2 segundos)
3. **Bostezos:** Detección de aperturas bucales prolongadas (> 4 segundos)
4. **Fricción Ocular:** Detección de contacto de manos con las regiones oculares
5. **Inclinación de Cabeza (Pitch):** Detección de inclinaciones sostenidas de la cabeza (≥ 3 segundos)

2. Requisitos del Sistema

2.1 Requisitos de Hardware

- **Procesador:** Intel Core i5 o equivalente (AMD Ryzen 5 o superior recomendado)
- **Memoria RAM:** Mínimo 4 GB (8 GB recomendado)
- **Cámara Web:** Cámara USB estándar (resolución mínima 640x480)
- **Espacio en Disco:** 500 MB libres para la instalación
- **Sistema Operativo:**
 - Windows 10/11 (64-bit)
 - Linux (Ubuntu 20.04 o superior, Debian 11 o superior)
 - macOS 10.15 o superior

2.2 Requisitos de Software

- **Python:** Versión 3.10 o superior
- **Navegador Web:** Chrome, Firefox, Edge o Safari (última versión)
- **Conexión a Internet:** Requerida solo para la instalación inicial

2.3 Requisitos Adicionales

- **Iluminación:** Ambiente bien iluminado para mejor detección facial
 - **Posicionamiento:** Usuario debe estar frente a la cámara a una distancia de 50-100 cm
 - **Permisos de Cámara:** El sistema requiere permisos para acceder a la cámara web
-

3. Instalación

3.1 Instalación desde Código Fuente

Paso 1: Clonar o Descargar el Repositorio

```
git clone https://github.com/tu_usuario/driver_fatigue_detection.git  
cd driver_fatigue_detection
```

Paso 2: Crear Entorno Virtual

Windows:

```
python -m venv venv  
venv\Scripts\activate
```

Linux/macOS:

```
python3 -m venv venv  
source venv/bin/activate
```

Paso 3: Instalar Dependencias del Servidor

```
pip install -r requirements_server.txt
```

Paso 4: Instalar Dependencias del Cliente

```
pip install -r requirements_client.txt
```

3.2 Instalación mediante Docker

Paso 1: Construir la Imagen Docker

```
docker build -t drowsiness-server .
```

Paso 2: Ejecutar el Contenedor

```
docker run -d -p 8000:8000 --name drowsiness-server drowsiness-server
```

3.3 Instalación desde Ejecutable

Si tiene acceso a un ejecutable pre-compilado:

1. Descargue el archivo ejecutable
 2. Ejecute el instalador
 3. Siga las instrucciones del asistente de instalación
 4. La aplicación se instalará en la ubicación predeterminada
-

4. Inicio del Sistema

4.1 Iniciar el Servidor

El sistema requiere que el servidor esté ejecutándose antes de iniciar la interfaz gráfica.

Método 1: Ejecución Manual

Abra una terminal y ejecute:

```
uvicorn app:app --host 0.0.0.0 --port 8000
```

El servidor estará disponible en <http://localhost:8000>

Método 2: Ejecución con Docker

Si utiliza Docker, el servidor ya está ejecutándose en el contenedor.

Método 3: Ejecución mediante Script

En sistemas Windows, puede utilizar el script `start_server.bat`. En Linux/macOS, utilice `start_server.sh`.

4.2 Iniciar la Interfaz Gráfica

Una vez que el servidor esté ejecutándose, abra una nueva terminal y ejecute:

```
flet run main.py
```

O si utiliza el ejecutable:

```
./drowsiness_detection_app
```

La interfaz gráfica se abrirá automáticamente en una ventana.

4.3 Verificar la Conexión

Antes de comenzar, verifique que:

- El servidor está ejecutándose (puerto 8000)
- La cámara web está conectada y funcionando
- Los permisos de cámara están habilitados

5. Guía de Uso

5.1 Pantalla de Inicio

Al iniciar la aplicación, se mostrará la pantalla de inicio con:

- **Título de la Aplicación:** "Educare IA"
- **Botón "Welcome":** Haga clic para continuar a la siguiente pantalla

Acción: Haga clic en el botón "Welcome" para proceder.

5.2 Pantalla de Selección de Interfaz

En esta pantalla se muestra:

- **Imagen de Somnolencia:** Representación visual del módulo de detección
- **Botón "Somnolencia":** Para acceder al módulo de detección de somnolencia

Acción: Haga clic en el botón "Somnolencia" para iniciar la detección.

5.3 Pantalla de Detección de Somnolencia

Esta es la pantalla principal del sistema. Muestra:

Panel Izquierdo:

- **Vista Original:** Muestra el video en tiempo real de la cámara
- **Botón "Start":** Inicia la detección de somnolencia

Panel Derecho:

- **Vista Procesada:** Muestra el video con anotaciones y detecciones

- **Botón "Stop":** Detiene la detección

5.4 Proceso de Detección

Paso 1: Iniciar la Detección

1. Asegúrese de estar frente a la cámara
2. Mantenga una distancia de 50-100 cm de la cámara
3. Asegúrese de que el rostro esté bien iluminado
4. Haga clic en el botón "**Start**"

Paso 2: Monitoreo en Tiempo Real

Una vez iniciada la detección:

- El sistema comenzará a procesar los frames de video
- En la vista procesada se mostrarán:
 - Mallas faciales detectadas
 - Puntos clave de las manos (si están visibles)
 - Anotaciones de texto con el estado actual
 - Alertas visuales cuando se detecten eventos

Paso 3: Detener la Detección

Para detener la detección:

1. Haga clic en el botón "**Stop**"
2. El procesamiento se detendrá inmediatamente
3. Las vistas de video volverán a mostrar la imagen de lugar

5.5 Interpretación de las Visualizaciones

Indicadores Visuales:

1. **Malla Facial:** Líneas verdes que conectan los puntos faciales detectados
2. **Puntos de Manos:** Puntos amarillos que indican las posiciones de los dedos

3. Estado de Ojos:

4. Verde: Ojos abiertos
5. Rojo: Ojos cerrados

6. Estado de Boca:

7. Verde: Boca cerrada
8. Rojo: Boca abierta

9. **Posición de Cabeza:** Indicador de inclinación (normal, abajo, izquierda, derecha)

Contadores en Pantalla:

- **Flicker Count:** Número de parpadeos detectados (se actualiza cada 60 segundos)
- **Micro Sleep Count:** Número de microsueños detectados (se reporta inmediatamente)
- **Yawn Count:** Número de bostezos detectados (se actualiza cada 180 segundos)
- **Eye Rub Count:** Número de fricciones oculares detectadas (se actualiza cada 300 segundos)
- **Pitch Count:** Número de inclinaciones de cabeza detectadas (se reporta inmediatamente)

6. Interpretación de Resultados

6.1 Reportes en Tiempo Real

El sistema genera reportes JSON en tiempo real que incluyen:

```
{  
  "timestamp": "2024-01-15 14:30:25",  
  "eye_rub_first_hand": {  
    "report": false,  
    "count": 0,  
    "durations": []  
  },  
  "flicker": {  
    "count": 0,  
    "durations": []  
  }  
}
```

```
        "report": true,
        "count": 45
    },
    "micro_sleep": {
        "report": false,
        "count": 0,
        "durations": []
    },
    "pitch": {
        "report": true,
        "count": 2,
        "durations": ["1 pitch: 4.0 seconds", "2 pitch: 3.5 seconds"]
    },
    "yawn": {
        "report": false,
        "count": 0,
        "durations": []
    }
}
```

6.2 Reportes CSV

Los reportes CSV se almacenan en:

```
drowsiness_processor/reports/august/drowsiness_report.csv
```

Cada fila contiene: - **Timestamp**: Fecha y hora del evento - **Eye Rub Reports**: Reportes de fricción ocular (primera y segunda mano) - **Flicker Report**: Reporte de parpadeos - **Micro Sleep Report**: Reporte de microsueños - **Pitch Report**: Reporte de inclinaciones de cabeza - **Yawn Report**: Reporte de bostezos

6.3 Umbrales de Alerta

El sistema utiliza los siguientes umbrales:

Biomarcador	Umbral	Significado
Parpadeos	Cada 60 segundos	Conteo normal: 15-20 por minuto
Microsueños	≥ 2 segundos	Alerta inmediata
Bostezos	>4 segundos	Alerta si >3 en 3 minutos
Fricción Ocular	>1 segundo	Alerta si frecuente
Inclinación Cabeza	≥ 3 segundos	Alerta inmediata

6.4 Niveles de Alerta

- **Normal:** No se detectan eventos anormales
 - **Atención:** Se detectan algunos eventos menores
 - **Alerta:** Se detectan múltiples eventos o eventos críticos (microsueños, inclinaciones sostenidas)
 - **Crítico:** Se detectan múltiples eventos críticos simultáneamente
-

7. Solución de Problemas

7.1 Problemas Comunes

Problema: La cámara no se detecta

Solución: 1. Verifique que la cámara esté conectada 2. Verifique que ningún otro programa esté usando la cámara 3. Revise los permisos de cámara en la configuración del sistema 4. Reinicie la aplicación

Problema: El servidor no inicia

Solución: 1. Verifique que el puerto 8000 no esté en uso: ```bash # Windows netstat -ano | findstr :8000

```
# Linux/macOS ls of -i :8000 `` 2. Cambie el puerto en app.py` si es necesario 3. Verifique que todas las dependencias estén instaladas
```

Problema: No se detecta el rostro

Solución: 1. Asegúrese de estar frente a la cámara 2. Mejore la iluminación del ambiente 3. Asegúrese de que el rostro esté completamente visible 4. Elimine obstáculos entre la cámara y el rostro

Problema: La detección es lenta

Solución: 1. Cierre otras aplicaciones que consuman recursos 2. Reduzca la resolución de la cámara 3. Verifique que el hardware cumpla con los requisitos mínimos 4. Actualice los controladores de la cámara

Problema: Los reportes no se generan

Solución: 1. Verifique que la carpeta `drowsiness_processor/reports/august/` exista 2. Verifique los permisos de escritura en la carpeta 3. Revise los logs del servidor para errores

7.2 Mensajes de Error

Error: "WebSocket connection failed"

Causa: El servidor no está ejecutándose o no es accesible.

Solución: Inicie el servidor antes de usar la interfaz gráfica.

Error: "Camera not found"

Causa: La cámara no está conectada o no es accesible.

Solución: Verifique la conexión de la cámara y los permisos.

Error: "Module not found"

Causa: Faltan dependencias instaladas.

Solución: Ejecute `pip install -r requirements_client.txt` y `pip install -r requirements_server.txt`.

8. Preguntas Frecuentes

P: ¿Puedo usar el sistema sin conexión a Internet?

R: Sí, una vez instalado, el sistema funciona completamente offline. Solo se requiere Internet para la instalación inicial.

P: ¿Funciona con cualquier cámara web?

R: Sí, el sistema funciona con cualquier cámara USB estándar compatible con OpenCV.

P: ¿Puedo usar el sistema con videos pre-grabados?

R: Actualmente, el sistema está diseñado para trabajar con video en tiempo real. Para videos pre-grabados, se requiere modificación del código.

P: ¿Los datos se almacenan de forma segura?

R: Los reportes CSV se almacenan localmente en su computadora. No se transmiten datos a servidores externos.

P: ¿Puedo personalizar los umbrales de detección?

R: Los umbrales están definidos en el código fuente. Para modificarlos, consulte el Manual de Aplicación del Código Fuente.

P: ¿El sistema funciona en dispositivos móviles?

R: Actualmente, el sistema está diseñado para computadoras de escritorio y laptops. No hay soporte oficial para dispositivos móviles.

P: ¿Cuál es la precisión del sistema?

R: La precisión depende de varios factores como iluminación, calidad de la cámara y condiciones del ambiente. En condiciones óptimas, la precisión es superior al 90%.

P: ¿Puedo usar múltiples cámaras?

R: El sistema está configurado para usar una cámara a la vez. Para usar múltiples cámaras, se requiere modificación del código.

9. Soporte Técnico

Para soporte técnico, reporte de errores o sugerencias:

- **Email:** midelgadol@ucvvirtual.edu.pe
 - **GitHub:** https://github.com/letalandroid/driver_fatigue_detection
 - **Documentación:** Consulte el Manual de Aplicación del Código Fuente
-

10. Información de Licencia

Este software se proporciona "tal cual" sin garantías de ningún tipo. Consulte el archivo LICENSE para más información.

Anexos

diagrama

A. Glosario de Términos

- **Biomarcador:** Indicador biológico medible de un estado fisiológico
- **Frame:** Imagen individual de un video
- **MediaPipe:** Framework de Google para percepción multimodal
- **WebSocket:** Protocolo de comunicación bidireccional en tiempo real
- **CSV:** Formato de archivo de valores separados por comas
- **JSON:** Formato de intercambio de datos ligero

B. Referencias

- Documentación de MediaPipe: <https://mediapipe.dev>
 - Documentación de OpenCV: <https://opencv.org>
 - Documentación de Flet: <https://flet.dev>
-

Fin del Manual de Usuario