



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E
INDUSTRIAL**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y
COMUNICACIONES**

TEMA:

**SISTEMA DE CONTROL DEL ESTADO DE SOMNOLENCIA EN
CONDUCTORES DE VEHÍCULOS**

Trabajo de Graduación Modalidad: Proyecto de Investigación, presentado previo la obtención del título de Ingeniero en Electrónica y Comunicaciones

SUBLÍNEA DE INVESTIGACIÓN: Procesamiento Digital de Señales e Imágenes

AUTOR: Wilfrido Leonel López Romero

TUTOR: Ing. Giovanni Brito, Mg.

Ambato - Ecuador
Enero, 2016

APROBACIÓN DEL TUTOR

En mi calidad de Tutor del Trabajo de Investigación sobre el Tema: “**Sistema de control del estado de somnolencia en conductores de vehículos**”, del señor, López Romero Wilfrido Leonel, estudiante de la Carrera de Ingeniería en Electrónica y Comunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos suficientes para que continúe con los trámites y consiguiente aprobación de conformidad con el Art. 16 del Capítulo II, del Reglamento de Graduación para Obtener el Título Terminal de Tercer Nivel de la Universidad Técnica de Ambato.

Ambato, Enero de 2016

Ing. Giovanni Brito, Mg.
TUTOR

AUTORÍA

El presente trabajo de investigación titulado: **“Sistema de control del estado de somnolencia en conductores de vehículos”**. Es absolutamente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, Enero de 2016

Wilfrido Leonel López Romero
C.I: 1804679015

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación, con fines de difusión pública. además autorizo su reproducción dentro de las regulaciones de la Universidad.

Ambato, Enero de 2016

Wilfrido Leonel López Romero
C.I: 1804679015

APROBACIÓN COMISIÓN CALIFICADORES

La Comisión Calificadora del presente trabajo conformada por los señores docentes Ing. Patricio Córdova, Mg. e Ing. Clara Sánchez, Mg., revisó y aprobó el Informe Final del trabajo de graduación titulado *Sistema de control del estado de somnolencia en conductores de vehículos*, presentado por el señor Wilfrido Leonel López Romero de acuerdo al Art. 17 del Reglamento de Graduación para obtener el título Terminal de tercer nivel de la Universidad Técnica de Ambato.

Ing. Vicente Morales L., Mg.
PRESIDENTE DEL TRIBUNAL

Ing. Patricio Córdova, Mg.
DOCENTE CALIFICADOR

Ing. Clara Sánchez, Mg.
DOCENTE CALIFICADOR

DEDICATORIA

El trabajo que lo he realizado está dedicado a mis padres Carlos López y Narcisa Romero por su apoyo que me han brindado durante mi vida estudiantil

Leonel López

AGRADECIMIENTO

Nombrar a todos quienes me ayudaron sin olvidarme de nombrar a alguien a logar este importante paso sería una lista muy grande, por lo que lo resumo de la siguiente manera: gracias Dios, gracias Familia, gracias Ingenieros, gracias amigos y a todas las personas adecuadas en mi camino. . .

Leonel López

ÍNDICE

APROBACIÓN DEL TUTOR	ii
AUTORÍA	iii
DERECHOS DE AUTOR	iv
APROBACIÓN COMISIÓN CALIFICADORA	v
Dedicatoria	vi
Agradecimiento	vii
Resumen	xv
Abstract	xvi
Glosario de términos	xvii
Introducción	xix
CAPÍTULO I EL PROBLEMA	1
1.1 Tema	1
1.2 Planteamiento del problema	1
1.3 Delimitación	3
1.4 Justificación	3
1.5 Objetivos	4
1.5.1 General	4
1.5.2 Específicos	4
CAPÍTULO II MARCO TEÓRICO	5
2.1 Antecedentes Investigativos	5
2.2 Fundamentación teórica	6
2.2.1 El vehículo	6
2.2.2 El sueño	6

2.2.3	La somnolencia excesiva	7
2.2.4	La fase paradójica o sueño paradójico	7
2.2.5	Cambios fisiológicos durante el sueño	8
2.2.5.1	Sistema nervioso central	8
2.2.5.2	Sistema nervioso periférico	8
2.2.5.3	Sistema nervioso autónomo	9
2.2.6	Oximetría de pulso	11
2.2.7	Visión artificial	13
2.2.7.1	Dificultades de la visión artificial	14
2.2.7.2	Adquisición de Imágenes	17
2.2.7.3	Pre-procesamiento de imágenes	19
2.2.7.4	Segmentación	21
2.2.7.5	La parametrización o selección de rasgos	22
2.2.7.6	El reconocimiento y la interpretación.	22
2.2.7.7	Seguimiento de Objetos	31
2.3	Propuesta de Solución	33

CAPÍTULO III METODOLOGÍA 34

3.1	Modalidad de la investigación	34
3.2	Recolección de información	34
3.3	Procesamiento de la información	35
3.4	Desarrollo del proyecto	35

CAPÍTULO IV DESARROLLO DE LA PROPUESTA 36

4.1	Introducción	36
4.2	Análisis de requerimientos	37
4.2.1	Hardware	37
4.2.2	Software	42
4.3	Distribuciones de Linux para Cubietruck	42
4.4	Descripción general del sistema	43
4.5	Adquisición de la señal para la oximetría de pulso	44
4.6	Desarrollo del software para la detección de somnolencia	45
4.6.1	Acceso a la cámara web y visualización de vídeo	45
4.6.2	Procesos para la detección de rostro y ojos	46
4.6.3	Seguimiento del rostro	51
4.6.4	Clasificación de ojos abiertos o cerrados	53
4.6.5	Orientación del rostro	57
4.6.6	Dirección del rostro	57

4.7	Instalación del sistema de control del estado de somnolencia en una camioneta Nissan doble cabina.	58
4.8	Pruebas de funcionamiento	61
4.8.0.1	Falsa alarma	65
4.8.1	Resultados del detector de somnolencia	67
4.9	Costo del proyecto	70
CAPÍTULO V CONCLUSIONES Y RECOMENDACIONES		71
5.1	Conclusiones	71
5.2	Recomendaciones	72
Referencias Bibliográficas		73
ANEXOS		78

ÍNDICE DE FIGURAS

2.1	Absorción de luz en la hemoglobina [23]	12
2.2	Luz absorbida por el dedo [23]	12
2.3	Señal típica de un oxímetro de pulso [23]	13
2.4	Ambigüedad en la definición de conceptos. [25]	14
2.5	Cambio de iluminación. [25]	14
2.6	Cambio de escala. [25]	15
2.7	Deformación. [25]	15
2.8	Oclusión. [25]	16
2.9	Movimiento. [25]	16
2.10	Perdida de información. (a) Observador de una imagen en 3D. (b) Imagen tomada por una cámara 2D. [25]	16
2.11	Técnicas de iluminación. (a) Iluminación Frontal. (b) Iluminación trasera. [28]	19
2.12	Tonos en una imagen	20
2.13	Histograma de una imagen	20
2.14	Esquema general de un clasificador de aprendizaje supervisado. [26] .	24
2.15	Clasificador SVM [33]	24
2.16	Conjunto de datos linealmente separables. [33]	25
2.17	Truco del kernel para el caso no linealmente separable. [33]	26
2.18	Conjunto de datos de aprendizaje o entrenamiento [34]	27
2.19	Clasificación del nuevo objeto en la clase más frecuente [34]	28
2.20	Características tipo Haar. (a) las de bordes, (b) las de líneas, (c) las de forma de X. [35]	29
2.21	Características seleccionadas por la cascada de detectores de Viola & Jones. [35]	29
2.22	Imagen integral. [37]	30
2.23	Diagrama de cascada de clasificadores. [35]	31
2.24	Deslizamiento de la imagen plantilla sobre la imagen de entrada. [38]	32
2.25	Resultado de deslizar la plantilla sobre la imagen de entrada. [38] . .	32
4.1	Convertor 12Vcc a 5Vcc @ 3A	40

4.2	Circuito integrado CMOS TSL235R [41]	41
4.3	Circuito integrado CMOS TSL257 [42]	42
4.4	Diagrama de bloques del funcionamiento del sistema de control del estado de somnolencia	44
4.5	Circuito esquemático del oxímetro de pulso	45
4.6	Conversión de color a escala de gris. (a) imagen RGB. (b) imagen en escala de grises.	48
4.7	Ecuilización del histograma. (a) imagen de entrada en escala de grises. (b) imagen ecualizada.	49
4.8	Detección de rostro y ojos durante la conducción	51
4.9	Región de búsqueda del rostro	51
4.10	Plantilla del rostro para su búsqueda	52
4.11	Matriz de resultado que contiene la coincidencia del rostro	53
4.12	Base de datos ojos abiertos: (a) Ojos abiertos durante el día. (b) Ojos abiertos durante la noche	54
4.13	Base de datos ojos cerrados: (a) Ojos cerrados durante el día. (b) Ojos cerrados durante la noche	54
4.14	Muestras de aprendizaje de ojos abiertos y cerrados	55
4.15	Orientación del Rostro	57
4.16	Dirección del rostro	58
4.17	Sistema de control del estado de somnolencia	59
4.18	Ubicación de la cámara web sobre el tablero del vehículo	59
4.19	Ubicación de la pantalla en el tablero del vehículo	60
4.20	Ubicación del zumbador en el apoya cabezas	60
4.21	Ubicación de la carcasa que contiene la Cubietruck, conversor de voltaje y el Arduino dentro del vehículo	61
4.22	Colocación del sensor de oximetría de pulso en el dedo índice de la mano izquierda	61
4.23	Detección de rostro y ojos del conductor durante el día	62
4.24	Detección de rostro y ojos del conductor durante la noche	62
4.25	Conductor con los ojos cerrados.	63
4.26	Conductor distraído durante el día	63
4.27	Conductor distraído durante la noche	64
4.28	Conductor con somnolencia durante día	64
4.29	Conductor con somnolencia durante la noche	65
4.30	Falsa alarma por presencia de movimiento rápido	66
4.31	Cambios bruscos de iluminación	66

4.32 Falsos positivos	67
4.33 Historial de la monitorización del ritmo cardiaco	67
4.34 Historial de la monitorización de la saturación de oxígeno	68

ÍNDICE DE TABLAS

2.1	Frecuencia cardíaca en reposo [22]	11
4.1	Comparación de cámaras web	37
4.2	Comparación de ordenadores de una placa (SBC)	38
4.3	Resultados del sistema de control del estado de somnolencia durante el día	68
4.4	Resultados de clasificación de ojos cerrados y la detección de distracción durante el día	69
4.5	Resultados del sistema de control del estado de somnolencia durante la noche	69
4.6	Resultados de clasificación de ojos cerrados y la detección de distracción durante la noche	69
4.7	Costo de los materiales para el proyecto	70

RESUMEN

A nivel mundial la somnolencia y la distracción durante la conducción han sido descritas como causantes de accidentes de tránsito dejando altísimo siniestralidad en término de pasajeros muertos, heridos y pérdidas materiales. Las condiciones que hacen sospechar que un accidente de tránsito fue provocado por la somnolencia son: el automóvil sale inadvertidamente de la vía o invade carril contrario y por último el conductor no deja huellas de haber intentado una maniobra evasiva para evitar el accidente.

En este trabajo se ha desarrollado un prototipo de un sistema de control del estado de somnolencia en conductores de vehículos, el sistema utiliza la visión por artificial o visión por computador y la oximetría de pulso.

El sistema se desarrolló utilizando C++ con OpenCV para la detección de ojos abiertos o cerrados, distracción y orientación, esto mediante técnicas de visión artificial, y se utiliza la oximetría de pulso que está constituido por un fotodiodo que detecta la cantidad de luz que incide un led rojo (660nm) y un led infrarrojo (940nm) al atravesar el tejido del dedo de la mano la cual permite calcular la oxigenación de la sangre y el ritmo cardíaco.

Palabras claves: Somnolencia, distracción, visión artificial, oximetría.

ABSTRACT

At level world the somnolence and distraction during driving have been reported to cause traffic accidents soaring accident leaving passengers in terms of deaths, injuries and property damage. The conditions that make suspect that a traffic accident was caused by somnolence are the car out of the way or inadvertently invades opposite lane and finally the driver leaves no traces of having attempted an evasive maneuver to avoid the accident.

This paper has developed a prototype of a system for monitoring the state of somnolence in drivers, the system uses artificial vision or computer vision and pulse oximetry.

The system was developed using C ++ with OpenCV to detect of open or closed eyes, distraction and guidance, this by techniques computer vision, and pulse oximetry which is constituted by a photodiode that detects the amount of incident light of the red led (660 nm) and an infrared led (940 nm) to pass through tissue finger which calculates the blood oxygenation and heart rate.

Keywords: Somnolence, distraction, artificial vision, oximetry.

Glosario de términos

- **IFN:** Instituto Ferrero de Neurología y Sueño
- **NCSDR/NHTSA:** Centro Nacional de Investigación de Trastornos del Sueño / Administración Nacional de Seguridad Vial
- **NSF:** National Sleep Foundation
- **MOR:** Movimientos Oculares Rápidos
- **NMOR:** No Movimientos Oculares Rápidos
- **SpO2:** Saturación de oxígeno
- **Hb:** Hemoglobina
- **HbO2:** Desoxihemoglobina
- **DC:** Corriente Continua
- **AC:** Corriente Alterna
- **Ey:** Ecuación fundamental de la Luminancia
- **RGB:** Red, Green, Blue.
- **AT&T:** American Telephone and Telegraph
- **SVM:** Support Vector Machine (máquinas de soporte vectorial o máquinas de vectores de soporte)
- \mathfrak{R} : Puntos de entrada de características para un clasificador
- **K-nn:** K-Nearest Neighbors (K - vecinos más cercanos)
- **VGA:** Video Graphics Array o Adaptador Gráfico de Video
- **XVGA:** eXtended Video Graphics Array
- **CMOS:** Complementary metal oxide semiconductor (semiconductor complementario de óxido metálico)
- **SBC:** Single Board Computer
- **SoC:** system on chip
- **CPU:** Central Processing Unit - Unidad de Procesamiento Central
- **RAM:** Random Access Memory (memoria de acceso aleatorio)
- **ARM:** Advanced RISC Machine - Máquina avanzada RISC

- **DDR3:** Double Data Rate type three
- **SD:** Secure Digital (SD) es un formato de tarjeta de memoria para dispositivos portátiles.
- **SATA:** Serial Advanced Technology Attachment
- **HDMI:** High-Definition Multimedia Interface
- **USB:** Universal Serial Bus
- **UART:** Universal Asynchronous Receiver-Transmitter
- **GPIO:** General Purpose Input/Output
- **LCD:** Liquid Crystal Display
- **LDR:** Light Dependent Resistor
- **IDE:** Integrated Development Environment
- **BSD:** Berkeley Software Distribution
- **XML:** eXtensible Markup Language (lenguaje de marcas extensible)
- **JSON:** JavaScript Object Notation

INTRODUCCIÓN

Esta investigación se realiza debido a que la somnolencia y la distracción son unas de las causas que ocasiona atropellamiento, pérdida de pista y choque. Como consecuencia a estos accidentes, se producen heridos leves, heridos graves y fallecimientos. Por lo que se ha desarrollado un sistema de control del estado de somnolencia, permitiendo alertar al conductor en caso de estar somnoliento.

La distribución de este trabajo se conforma por cinco capítulos los cuales se describen brevemente a continuación:

En el Capítulo I, se detalla las razones por las cuales es importante el desarrollo de esta investigación partiendo de un análisis macro hasta detallar específicamente la necesidad que se debe cubrir sus límites y objetivos.

En el Capítulo II, se reúne la información teórica necesaria y las experiencias descritas en trabajos similares para el adecuado desarrollo del dispositivo propuesto, este capítulo contiene los conceptos teóricos sobre el sueño y cambios fisiológicos, la oximetría de pulso y la visión artificial, entre otros.

En el Capítulo III, se describe el tipo de investigación que se llevó a cabo y los procesos que se siguieron para desarrollar este trabajo.

En el Capítulo IV, se describe el proceso que se llevó para la selección de equipos, se explica cómo se desarrolla la fase de captura de video, el pre-procesamiento, la detección del rostro y ojos, la clasificación de ojos abiertos o cerrados y la oximetría de pulso para determinar la saturación de oxígeno en la sangre y el ritmo cardíaco.

En el Capítulo V, se mencionan las conclusiones y recomendaciones que se derivan del desarrollo de este trabajo. Finalmente, se presentan la lista de referencias consultadas y los anexos.

CAPÍTULO I

EL PROBLEMA

1.1. Tema

“SISTEMA DE CONTROL DEL ESTADO DE SOMNOLENCIA EN CONDUCTORES DE VEHÍCULOS”

1.2. Planteamiento del problema

La información publicada por Jorge Rey de Castro indica: que a nivel Mundial el cansancio y la somnolencia durante la conducción han sido descritas como causa de accidentes de tránsito en la ciudad y las carreteras ya que disminuye progresivamente su concentración y pierde capacidad de respuesta ante condiciones específicas que exigen reacciones inmediatas al momento de conducir; pestañear y dormir durante la conducción expresan un nivel extremo de fatiga y de sueño. Habitualmente, los accidentes producidos en estas circunstancias tienen altísima siniestralidad en términos de pasajeros muertos, heridos y pérdidas materiales [1].

El Instituto Ferrero de Neurología y Sueño (IFN) dice que una de las causas más ignoradas y que más muertes y lesiones producen, son los accidentes originados a causa del sueño que presenta el conductor. La repetida frase “el conductor, por razones que se desconocen, perdió el control del vehículo. . .”, suele esconder que la persona se durmió, o debido a la somnolencia generó un accidente evitable [2].

Según el Centro Nacional de Investigación de Trastornos del Sueño / Administración Nacional de Seguridad Vial (NCSDR/NHTSA) de los Estados Unidos, las

condiciones específicas que hacen sospechar si un accidente de tránsito en carretera fue provocado por somnolencia durante la conducción son: privación aguda del sueño, hábitos del conductor durante su trabajo como manejar durante la noche y/o en horario vespertino o jornadas prolongadas sin el descanso requerido, uso de medicación como sedantes. Según el informe de NCSDR/ NHTSA y la Nacional Sleep Foundation (NSF) las características que rodean al escenario son: el accidente ocurre generalmente entre las 00:00-7:00 horas y en horario vespertino entre las 13:00-15:00 horas, el automóvil sale inadvertidamente de la ruta o invade carril contrario y por último el conductor no deja huellas de haber intentado una maniobra evasiva para evitar la colisión [3].

La Organización Mundial de la Salud (OMS), informa que Ecuador es el segundo país en Sudamérica con el mayor índice de muertes por accidentes de tránsito. Esta cifra la corroboran estadísticas del INEC que identifican estos siniestros como la primera causa de fallecimiento en los hombres y como la quinta a nivel general [4].

La Agencia Nacional de Tránsito (ANT) registra que cada 20 minutos ocurre un accidente de tránsito en el país y cada cuatro horas alguien muere por esta causa. Sin embargo según los datos del Sistema Público para Pago de Accidentes de Tránsito (SPPAT) cada diez minutos alguien se accidenta y cada dos horas hay fallecidos por percances viales [4].

El director ejecutivo de la Comisión Interinstitucional de Educación, Seguridad y Prevención vial (Covial) explica que en Ecuador cada 18 días hay accidentes denominados como siniestros de magnitud, es decir, aquellos en que hay más de 15 víctimas, ya sean heridos o muertos [4].

Estadísticas del Servicio de Investigación de Accidentes de Tránsito (SIAT) de la Policía Nacional del Ecuador determinan las causas para que el accidente se produzca son: Desatiende momentáneamente la atención en la conducción (20%), se producen cuando los conductores no descansan apropiadamente, excediéndose en el tiempo de conducción; sobreviniendo el sueño, cansancio y fatiga. Los accidentes según los tipos de vehículo tenemos: En cuanto a los Buses (23%), de los cuales 19.71% de ellos eran de transporte interprovincial y 3.29% de transporte escolar. De la misma manera los vehículos Particulares (Automóvil, Camioneta, Jeep) representan el (37%), de los accidentes investigados. Los Tráileres y Camiones representan el (40%) restante. Cuando se accidentan los vehículos de transporte de pasajeros ocasionan más víctimas fatales y no fatales por su capacidad de transportar pasajeros a diferencia de los vehículos particulares [5].

Vehículos de gama alta de Mercedes Benz y Volkswagen como el Clase S y el Passat

CC respectivamente tienen detectores de somnolencia pero estos vehículos tienen un costo elevado mientras que los vehículos gama baja no constan de estos detectores de somnolencia siendo los vehículos de gama baja los más usados por las personas para transportarse.[6].

1.3. Delimitación

- **Área Académica:** Comunicaciones
- **Línea de Investigación:** Tecnologías de comunicación
- **Sublínea de Investigación:** Procesamiento digital de señales e imágenes
- **Delimitación Espacial:** La presente investigación se la realizará para los conductores de vehículos de la ciudad de Ambato
- **Delimitación Temporal:** El proyecto se desarrolló del 8 de mayo de 2014 al 5 de noviembre de 2015 después de su aprobación por parte del Honorable Consejo Directivo de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

1.4. Justificación

Las personas necesitan transportarse de un lugar a otro, y en el transcurso del viaje se corre el riesgo de sufrir un accidente de tránsito debido a que el conductor pierde la capacidad de concentración y reacción a causa de la somnolencia que produce muchos daños como lesiones y muertes tanto a conductores, peatones y pasajeros en el caso de empresas de transporte además de altos costes económicos, generando serios problemas a nivel social, por este motivo se desarrollará el sistema de control del estado de somnolencia en conductores de vehículos en el cual se analizará cambios faciales, frecuencia cardíaca y saturación de oxígeno.

Con la ejecución de este proyecto se trata de evitar los accidentes de tránsito provocado por la somnolencia, cansancio y desatención durante la conducción que trae como consecuencia pérdidas humanas y materiales, al solucionar este problema permitirá trabajar con mayor confianza y tranquilidad, ya que este dispositivo brinda al conductor una herramienta de uso constante que no interfiere con su trabajo.

Los beneficiarios directos de este proyecto son los conductores de todo tipo de vehículos ya que tendrán un sistema adicional para salvar sus vidas y como beneficiarios indirectos son los pasajeros y peatones.

1.5. Objetivos

1.5.1. General

Implementar un sistema de control del estado de somnolencia en conductores de vehículos.

1.5.2. Específicos

- Analizar los síntomas y causas de somnolencia en conductores de vehículos.
- Identificar los parámetros técnicos y algoritmos que permitan procesar señales del estado de somnolencia.
- Diseñar un prototipo con dispositivos electrónicos que permitan indicar al conductor que está con síntomas de somnolencia.
- Realizar pruebas de funcionamiento del prototipo para el control del estado de somnolencia en conductores de vehículos.

CAPÍTULO II

MARCO TEÓRICO

2.1. Antecedentes Investigativos

En el trabajo de investigación de Ávila Jiménez y sus colaboradores, de la Universidad de San Buenaventura, Bogotá - Colombia desarrollaron un sistema que se basa en el tiempo de duración de parpadeo del ojo humano. Utilizan sensores infrarrojos ubicados en unas gafas, de manera que al cerrar los ojos las pestañas interrumpen la señal entre los infrarrojos, de esta forma se detecta si los ojos están cerrados o abiertos, mientras la señal esta interrumpida se estima el tiempo que permanecen cerrados los ojos y se toma la decisión, pero tiene un inconveniente en que las facciones de la cara de cada persona no son iguales, se tiene que presentar un prototipo específico de gafas para cada cliente. Pero existe un mayor problema en el caso de las mujeres, que usan rímel en las pestañas y se las viran lo cual produce interferencia de forma continua marcando estado de sueño todo el tiempo [7].

En el proyecto de Flores Calero, de la Universidad Carlos III de Madrid, plantea un sistema de asistencia a la conducción (SAAC) diseñado para detectar automáticamente la somnolencia del conductor, y paralelamente su distracción. El sistema se centra en localizar el rostro y los ojos para analizarlos a través del tiempo y generar un índice de somnolencia y uno de distracción, utilizado técnicas de Visión por Computador e Inteligencia Artificial, sobre imágenes capturadas con un sistema de percepción con iluminación infrarroja. Este sistema ha sido probado con varios conductores sobre un vehículo en condiciones reales de conducción nocturna [8].

En el proyecto de Córdova Ochoa, de la Escuela Politécnica del Ejercito extensión Latacunga, plantea un sistema para la prevención de accidentes mediante el frenado

del vehículo al momento que el conductor pierde el control de sí mismo por causa del sueño, se activa una alarma sonora cuando se detecta el cabeceo. Esta alarma también activa un módulo de control por medio de una señal eléctrica. El módulo al recibir la señal, automáticamente procesa mediante un programa que regula los voltajes y corrientes del sistema para al final activar un mecanismo de freno. Este mecanismo activa un freno que reduce la velocidad y se desactiva mediante una señal de reinicio dada por el conductor consciente de su estado o por otro que se encuentre en mejores condiciones [9].

En la publicación de J. L. García, E. Rogado, R. Barea y demás colaboradores, tiene como objetivo determinar el nivel de fatiga del conductor y establecer si se encuentra en condiciones de conducir mediante el análisis de variables biológicas. Este trabajo se centrará en el estudio del ECG durante la conducción y en concreto en la variabilidad del ritmo cardiaco -HRV- que refleja las interacciones de las dos ramas del sistema nervioso central (simpática y parasimpática). Se ha desarrollado un hardware para la adquisición y el procesamiento de estas variables así como un algoritmo para la detección de latidos y el cálculo del HRV teniendo en cuenta la presión con la que se agarra el volante así como la temperatura en la cabina y el exterior. [10]

2.2. Fundamentación teórica

2.2.1. El vehículo

Es un medio de transporte de cualquier tipo, forma o época que se utiliza para transportar cargas o personas, con o sin motor, de un lugar a otro.

2.2.2. El sueño

El sueño es un estado fisiológico de autorregulación y reposo uniforme de un organismo.

El sueño se caracteriza por los bajos niveles de actividad fisiológica (presión sanguínea, respiración) y por una respuesta menor ante estímulos externos. [11]

Cuando el sueño es interrumpido o acortado, puede que no se pase suficiente tiempo en determinadas etapas del sueño. Lo bien que hayas descansado y lo bien que funciones al día siguiente dependerá del tiempo total que se haya dormido y de la calidad que se haya tenido.

Se necesita dormir para pensar claramente, reaccionar rápido y asentar nuestra memoria. De hecho los procesos del cerebro que ayudan a aprender y recordar son especialmente activos mientras dormimos.

Escatimar sueño tiene un precio. Restar tan solo una hora de sueño puede hacer que sea difícil concentrarse al día siguiente y enlentecer tu tiempo de respuesta. Los estudios también indican que la falta de sueño hace que tomemos malas decisiones y asumamos riesgos innecesarios. Esto puede dar lugar a bajo rendimiento en el trabajo o la escuela y mayor riesgo de sufrir accidentes de tránsito. [12]

El sueño también afecta al estado de ánimo. Si no dormimos suficiente estaremos más irritables y eso afectará a nuestro comportamiento y relaciones. [12]

2.2.3. La somnolencia excesiva

La Somnolencia excesiva (SE) se define como la incapacidad de llegar a un estado completo de vigilia y alerta (sentirse despierto), para llevar a cabo las tareas de la vida diaria. El sueño puede aparecer sin querer o en momentos inadecuados, dando como consecuencia una imposibilidad por parte de las personas para llevar a cabo su vida normal.

Alguno de los problemas más comunes dentro de este tipo de padecimiento es [13]:

- Fatiga, falta de energía, etc.
- Poca motivación.
- Pérdida del sentido del bienestar.
- Dificultad para concentrarse.
- Baja autoestima.
- Frustración.
- Enfado a causa de no sentirse comprendidos.

2.2.4. La fase paradójica o sueño paradójico

El sueño paradójico es la fase del sueño en la los ojos se mueven rápidamente; por eso también se llama sueño MOR (Movimientos Oculares Rápidos). Es el momento en que el individuo está más relajado, aunque es relativamente fácil despertarlo, de ahí su nombre de paradójico. [14]

Sus características son: respiración rápida y superficial, discreta aceleración del ritmo cardíaco, fenómeno de ensoñación, el relajamiento muscular es mayor durante el

sueño paradójico y la razón de consumo de oxígeno del cuerpo es más alta, como si indicara que este es un sueño ligero y la temperatura corporal no está bien regulada y se aproxima a la temperatura ambiente; la regulación de la temperatura se produce por mecanismos independientes de los que dirigen el ritmo circadiano de sueño, pero existe cierta relación entre ambos en el hipotálamo. Durante el sueño nocturno se produce la mayor disminución de la temperatura corporal. Los mecanismos de termorregulación son más inestables durante el sueño NMOR que durante la vigilia, por lo que la temperatura desciende. Durante el sueño MOR la temperatura corporal se va ajustando a la temperatura ambiente. Las temperaturas ambientales extremas provocan interrupción del sueño, con frecuentes despertares, reduciéndose fundamentalmente la cantidad de sueño MOR. [14]

2.2.5. Cambios fisiológicos durante el sueño

2.2.5.1. Sistema nervioso central

El sistema nervioso central es una estructura biológica que se encuentra en el ser humano. El sistema nervioso central está constituido por el encéfalo y la médula espinal. Se trata de un sistema muy complejo, ya que se encarga de percibir estímulos procedentes del mundo exterior así como transmitir impulsos a nervios y a músculos instintivamente. [15]

En el sistema nervioso central la actividad encefálica va disminuyendo progresivamente en el paso de la vigilia al sueño NMOR (No Movimientos Oculares Rápidos). El consumo de glucosa en la corteza cerebral durante el sueño NMOR es menor y más homogéneo que en otras regiones. En el sueño MOR por el contrario la actividad aumenta, siendo esto particularmente evidente en las regiones relacionadas con la vía visual.[15]

2.2.5.2. Sistema nervioso periférico

El Sistema nervioso periférico durante el sueño produce una disminución de la actividad motora y un aumento de la intensidad necesaria de estímulos externos. En el sueño NMOR la actividad disminuye pero se conserva un cierto grado de activación que permite los movimientos corporales y se manifiesta por el registro de tono muscular. En el sueño MOR predomina la atonía de la musculatura esquelética, exceptuando la ocular, por una inhibición de las motoneuronas, aunque de manera intermitente se observan contracciones de músculos faciales que suelen coincidir con las salvas de movimientos oculares rápidos.[16]

El Sistema Nervioso Periférico se compone por todos los nervios que parten del Sistema Nervioso Central y se van ramificando para llegar a todas las partes del cuerpo. [16]

Existen principalmente dos tipos de nervios: Los craneales y los espinales.

- Nervios craneales

Los nervios craneales, se conectan directamente con el cerebro, son doce pares y pertenecen a los ojos, oídos, nariz, paladar y lengua.

- Nervios espinales

Los nervios espinales forman además otro sistema, el Nervioso Esquelético encargado de controlar todos los movimientos musculares voluntarios, es decir los que hacemos cuando queremos, como caminar, correr, escribir o masticar.

2.2.5.3. Sistema nervioso autónomo

Puede tener cambios como carecer de control voluntario y durante el sueño se provocan importantes modificaciones tanto en su porción simpática como parasimpática. Las funciones vegetativas son diferentes en sueño MOR y NMOR, fundamentalmente la respiración, la actividad cardiovascular y la regulación de la temperatura corporal. [17]

Respiración

Durante la vigilia la respiración tiene un doble control: voluntario (a través de la corteza premotora) e involuntario (a través de neuronas localizadas en la región inferior del tronco). Durante el sueño se reduce la ventilación ya que desaparece el control voluntario y aumenta el umbral sensorial. Además cambia la sensibilidad de los quimiorreceptores a las concentraciones de CO₂ y O₂. El descenso de la ventilación alveolar da lugar a hipoxemia e hipercapnia relativas, que serán las que regulen la frecuencia y amplitud de la respiración. En la transición vigilia-sueño la respiración suele ser irregular, estabilizándose a medida que el sueño NMOR se hace más profundo. El estímulo fundamental que controla la respiración en esta fase es la concentración arterial de CO₂. Durante el sueño MOR la frecuencia respiratoria aumenta y la respiración vuelve a ser irregular pudiendo depender de una cierta activación cortical. La atonía además aumenta la resistencia de la vía aérea superior. [17]

Actividad cardiovascular

Denominadas asimismo actividades cardiorrespiratorias o aeróbicas. Se trata de actividades prolongadas que requieren que el cuerpo aumente la frecuencia cardíaca y el ritmo respiratorio con el fin de suministrar oxígeno a los músculos que están trabajando.[18]

La actividad cardiovascular durante el sueño NMOR la tensión arterial disminuye entre 2 y 5 mm/Hg, fundamentalmente en las fases III y IV, haciéndose muy estable. Por el contrario durante el sueño MOR sufre importantes variaciones alcanzándose en esta fase los valores máximos y mínimos de la noche. Por su parte la frecuencia cardíaca disminuye en el paso de la vigilia al sueño NMOR y durante el sueño MOR se hace irregular pudiendo llegar a valores máximos por encima del 30% del basal en vigilia. Durante el sueño NMOR se produce una vasodilatación activa que aumenta el aporte sanguíneo a las vísceras. En el sueño MOR tónico se mantiene la vasodilatación excepto en la musculatura esquelética donde se produce vasoconstricción [17][19].

Ritmo cardíaco durante el sueño

Cuando el ser humano duerme el ritmo cardíaco generalmente es menor que cuando se está despierto, se debe a que el cuerpo se relaja profundamente por ende las necesidades metabólicas son menores.

El descenso que experimenta el ritmo cardíaco durante el sueño no es igual para todas las personas, tampoco es igual durante todas las distintas fases de sueño, pero la mayoría de expertos están de acuerdo en que el ritmo cardíaco debería descender en torno al 8% respecto a la frecuencia cardíaca basal (ritmo cardíaco en vigilia y estado de reposo). [20]

El ritmo cardíaco comienza a disminuir tan pronto como comenzamos a quedarnos dormidos, en la transición vigilia-sueño (fase 1 del sueño) se producen cambios en el sistema nervioso autónomo que llevan a un predominio del sistema parasimpático que explicaría, entre otros, el descenso del ritmo cardíaco. [20]

Después de quedarnos dormidos, la frecuencia de los latidos del corazón sigue disminuyendo durante las fases 2 (sueño ligero), fase 3 (transición al sueño profundo) y en la fase 4 (sueño delta). Durante el sueño delta el ritmo cardíaco alcanza la frecuencia mínima durante todo el período de sueño. Tras el sueño delta viene la fase 5, más conocida como sueño MOR, durante la cuál el ritmo cardíaco puede

variar notablemente de una persona a otra e incluso en la misma persona en días diferentes.[21]

Ritmo cardíaco en reposo

Se obtiene inmediatamente después de despertarse o bien, relajándose 5 minutos antes de la medición. En la Tabla 2.1, se muestra la frecuencia cardíaca en reposo.

Tabla 2.1: Frecuencia cardíaca en reposo [22]

Frecuencia cardíaca en reposo	
Edad	Latidos por minuto
1 a 10 años	70 a 120
mas de 10 años	60 a 100
Atletas bien entrenados	40 a 60

2.2.6. Oximetría de pulso

La oximetría de pulso es la medición no invasiva de la saturación de oxígeno (SpO_2). La saturación de oxígeno se define como la medición de la cantidad de oxígeno disuelto en la sangre, basado en la detección de la hemoglobina (Hb) y Desoxihemoglobina (HbO_2). Dos luces con diferentes longitudes de onda se utilizan para medir la diferencia real en los espectros de absorción de Hb y HbO_2 . El torrente sanguíneo se ve afectada por la concentración de Hb y HbO_2 , y sus coeficientes de absorción se mide usando dos longitudes de onda 660 nm (espectros de luz roja) y 940 nm (espectros de luz infrarroja). La Hemoglobina desoxigenada y oxigenada absorben diferentes longitudes de onda como se ve en la figura 2.1. La Hemoglobina desoxigenada (Hb) tiene una absorción mayor a 660 nm y la hemoglobina oxigenada (HbO_2) tiene una mayor absorción en 940 nm. [22, 23]

Un fotodetector percibe la luz emitida por de los Leds, el resultado es una señal como el de la figura 2.2. Esta señal representa la luz que ha sido absorbida por el dedo y se divide en un componente de DC y un componente de AC. El componente de DC representa la absorción de luz del tejido, la sangre venosa, arterial y la sangre no pulsátil. El componente de AC representa la sangre arterial pulsátil.

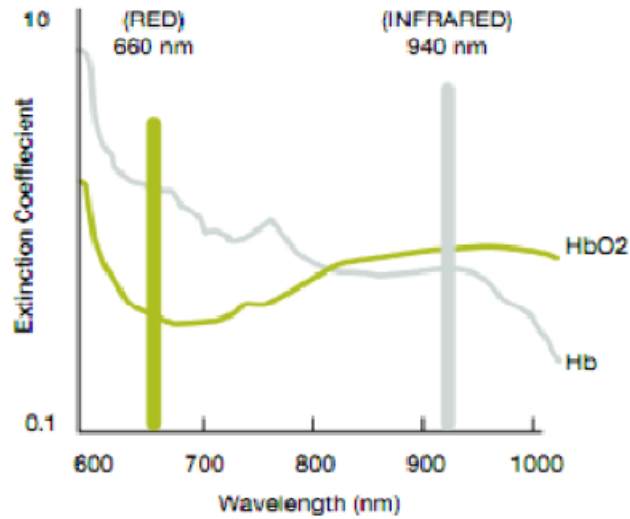


Fig. 2.1: Absorción de luz en la hemoglobina [23]

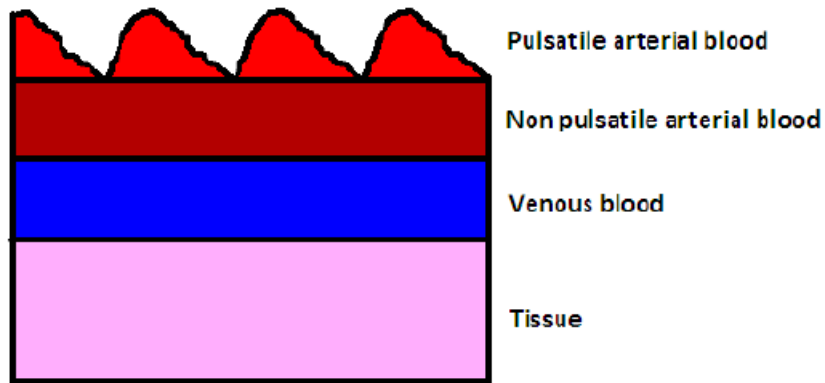


Fig. 2.2: Luz absorbida por el dedo [23]

Para obtener el valor SpO2 el oxímetro de pulso analiza la absorción de la luz (componentes de AC / DC) en las dos longitudes de onda y calcula la relación de absorción usando la siguiente ecuación.

$$R = \frac{\frac{AC_{660}}{DC_{660}}}{\frac{AC_{940}}{DC_{940}}} \quad (2.1)$$

Donde R es un coeficiente normalizado de la luz transmitida.

Valores normales de saturación están entre 87 y 97%.

Otra forma para el cálculo de SpO2 es tomando la componente AC de señal, la relación se determinada mediante el uso de la siguiente ecuación:

$$R = \frac{\log_{10}(I_{ac})\lambda_1}{\log_{10}(I_{ac})\lambda_2} \quad (2.2)$$

I_{ac} = La intensidad de luz en 1 (660 nm) o 2 (940 nm), donde sólo el nivel de CA está presente.

Una señal típica de oximetría de pulso se representa en la figura 2.3. La señal representa la absorción de la sangre arterial pulsátil. Los latidos por minuto se pueden calcular utilizando esta señal. [23].

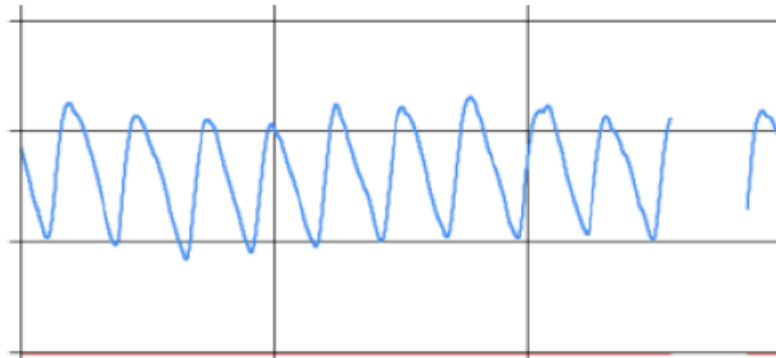


Fig. 2.3: Señal típica de un oxímetro de pulso [23]

2.2.7. Visión artificial

La visión artificial, también conocida como visión por computador (del inglés computer vision) o visión técnica, es un subcampo de la inteligencia artificial. El propósito de la visión artificial es programar un computador para que entienda una escena o las características de una imagen [24].

Objetivos de la visión artificial:

- La detección, segmentación, localización y reconocimiento de ciertos objetos en imágenes (caras humanas).
- La evaluación de los resultados (segmentación, registro).
- Registro de diferentes imágenes de una misma escena u objeto, es decir, hacer concordar un mismo objeto en diversas imágenes.
- Seguimiento de un objeto en una secuencia de imágenes.
- Mapeo de una escena para generar un modelo tridimensional de la escena; este modelo podría ser usado por un robot para navegar por la escena.

- Estimación de las posturas tridimensionales de humanos.
- Búsqueda de imágenes digitales por su contenido. Estos objetivos se consiguen por medio de reconocimiento de patrones, aprendizaje estadístico, geometría de proyección, procesamiento de imágenes, teoría de grafos y otros campos. La visión artificial cognitiva está muy relacionada con la psicología cognitiva y la computación biológica.

2.2.7.1. Dificultades de la visión artificial

Ambigüedad en la definición de conceptos

Se tiene variabilidad dentro de una misma clase como se muestra en la siguiente figura que la clase es la silla pero se tiene diferentes modelos y colores.



Fig. 2.4: Ambigüedad en la definición de conceptos. [25]

Cambios de iluminación

Los cambios de iluminación pueden crear sombras o reflejos, y producir pérdidas importantes de información.

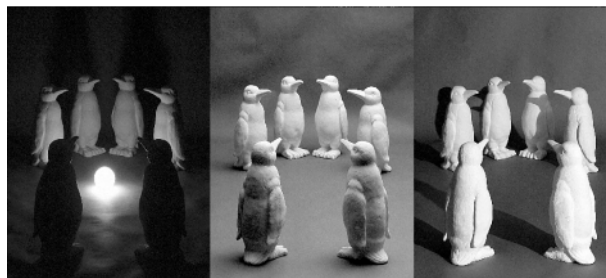


Fig. 2.5: Cambio de iluminación. [25]

Cambios de escala

Los cambios de escala hacen que se pierda el seguimiento de objetos lo que hace que se inicie de nuevo la búsqueda del objeto para posteriormente su seguimiento provocando un mayor uso de recursos de hardware.



Fig. 2.6: Cambio de escala. [25]

Deformación

La deformación dan confusiones al sistema lo que el seguimiento de objetos se pierde.



Fig. 2.7: Deformación. [25]

Oclusión

Una oclusión es la percepción visual de un objeto estando atrás o en frente de otro objeto, otorgando información sobre el orden de las capas de la textura. Cuando un objeto es ocluido, el sistema visual solo tiene información sobre las partes del objeto que se pueden ver, entonces el resto del procedimiento tiene que ser más profundo.



Fig. 2.8: Oclusión. [25]

Movimiento

Con movimientos rápidos se pierde información del objeto lo que el seguimiento o segmentación de ciertas características son falsas

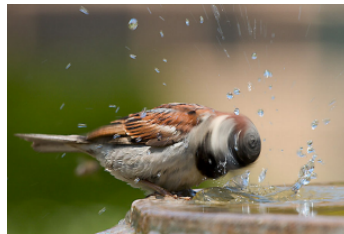


Fig. 2.9: Movimiento. [25]

Pérdida de información

Las escenas que se perciben suelen ser tridimensionales (3D) Los dispositivos de captura (cámaras fotográficas o de vídeo, pantallas de rayos X, etc.) generan imágenes dimensionales (2D). [25]

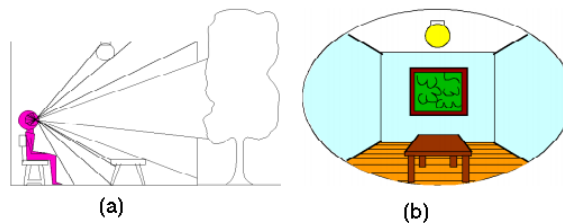


Fig. 2.10: Pérdida de información. (a) Observador de una imagen en 3D. (b) Imagen tomada por una cámara 2D. [25]

2.2.7.2. Adquisición de Imágenes

En este paso se trata de conseguir que la imagen sea lo mas adecuada para continuar con el resto de pasos. Una correcta adquisición de la imagen constituye un paso muy importante para que el proceso reconocimiento sea exitoso. Dentro de esta etapa influye la resolución de la cámara, la iluminación y posicionamiento de los elementos.

La cámara de video

La cámara es el dispositivo encargado de transformar las señales luminosas que aparecen en la escena, en señales analógicas. La cámara se divide en dos partes, el sensor, que captura las propiedades del objeto en forma de señales luminosas y lo transforma en señales analógicas, y la óptica que se encarga de proyectar los elementos adecuados de la escena ajustando una distancia focal adecuada. [26]

Tipos de cámaras de video

- *Webcam*

La Webcam es una pequeña cámara digital conectada a una computadora la cual puede capturar imágenes y transmitir las a través de Internet, ya sea a una página web o a otra u otras computadoras de forma privada. Las cámaras web necesitan una computadora para transmitir las imágenes. [27]

- *Cámaras IP*

Las cámaras IP son sistemas completos que se conectan directamente a Internet y muestran la imagen del lugar donde está colocada. Con una cámara IP puede utilizar su móvil para ver su casa desde cualquier parte del mundo, sin necesidad de otros equipos.

- *Cámaras analógicas*

La cámara de video analógico contiene información de los píxeles con todas las señales de sincronismo en un mismo cable, lo que hace que la conexión sea simple y barata. Si la señal de video es de barrido variable, el cable es algo más complejo y debe llevar algunos hilos pero sigue siendo simple. El cable además puede tener algunas decenas de metros sin perder potencia la señal. Todas las cámaras estándar son analógicas, existen algunas cámaras de alta velocidad y de alta definición. [28]

- *Cámaras Digitales*

Las cámaras digitales transmiten la información directamente en digital, lo que supone una mejor calidad frente a ruidos que puedan aparecer en los elementos de transmisión (cables, conectores, etc.) además pueden implementar bits de detección/corrección de errores que permitan verificar y corregir la información recibida, como desventaja el cable necesario es de un espesor considerable (formado por gran cantidad de hilos). [28]

La iluminación

La iluminación de la escena es importante en el desarrollo de un sistema de visión artificial. Un entorno debidamente controlado es imprescindible para obtener unas condiciones de captura de imágenes óptimas, antes de intentar corregir un problema de iluminación por medio de algoritmos. Algunas técnicas de iluminación se exponen a continuación:

Iluminación Frontal: Es la más usada, y consiste en iluminar directamente sobre el objeto, permite distinguir los detalles de los objetos como su forma, color, detalles internos, etc., permitiendo una mejor segmentación. Por otro lado, Presenta problemas para obtener un buen contraste entre la pieza y el fondo, debido a la aparición de brillos y sombras que alteran las propiedades de las piezas a estudio. [28]

Iluminación trasera o retroiluminación: Donde se ilumina una pantalla de forma que lo que se busca es el contorno del objeto a modo de sombra. El objeto se encuentra entre la cámara y la lámpara. Esta técnica proporciona imágenes con un alto contraste entre la pieza y el fondo, facilitando la segmentación mediante una binarización aunque se pierden los detalles de la escena. Las principales aplicaciones donde se comporta bien esta técnica de iluminación son para medir el grado de porosidad de ciertas sustancias y en inspección dimensional para calcular el tamaño de una pieza. [28]

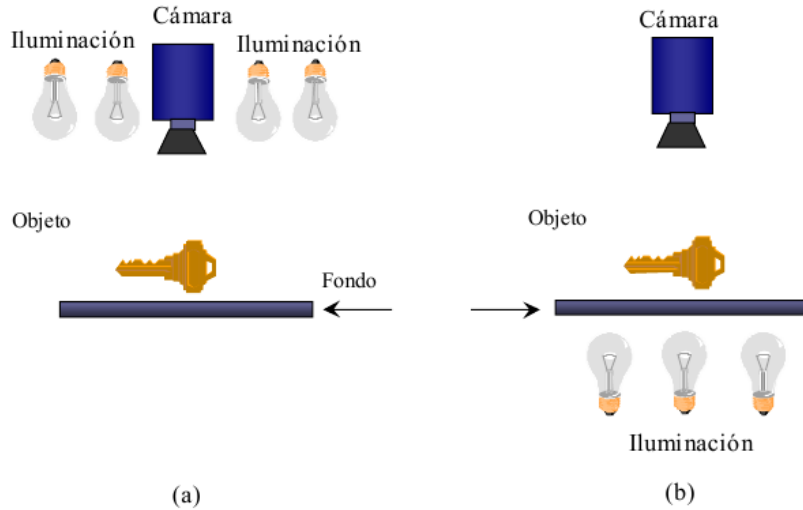


Fig. 2.11: Técnicas de iluminación. (a) Iluminación Frontal. (b) Iluminación trasera. [28]

2.2.7.3. Pre-procesamiento de imágenes

Consiste en la transformación de la imagen original en otra imagen en la cual hayan sido eliminados los efectos de degradación (ruido), que afectan a la fidelidad y definición de la imagen.

La parte de pre-procesamiento se realiza con el objetivo mejorar o destacar algún elemento de las imágenes, de manera que sea posible realizar las etapas posteriores de segmentación y clasificación.

Conversión de una imagen RGB a escala de grises

La conversión de una imagen en color a escala de grises, es el equivalente a la luminancia de la imagen. Como sabemos el ojo percibe distintas intensidades de luz en función del color que se observe, esto es debido a la respuesta del ojo al espectro visible, por esa razón el cálculo de la escala de grises o luminancia de la imagen debe realizarse como una media ponderada de las distintas componentes de color de cada pixel.

La ecuación de la luminancia (E_y) es la expresión matemática de ese fenómeno, y los factores de ponderación de cada componente de color nos indican la sensibilidad del ojo humano a las frecuencias del espectro cercanas al rojo, verde y azul. [29]

$$E_y = 0.3 * R + 0.5 * G + 0.11 * B \quad (2.3)$$

Al aplicar la ecuación 2.1 a cada pixel de la imagen de color, resultará una nueva matriz de un byte por pixel que daría la información de luminancia. [29]

Ecualización de histograma

Esta técnica es la más utilizada para realizar las primeras aproximaciones a la resolución del problema del realce. El histograma es una representación gráfica de la distribución de los distintos tonos de una imagen.

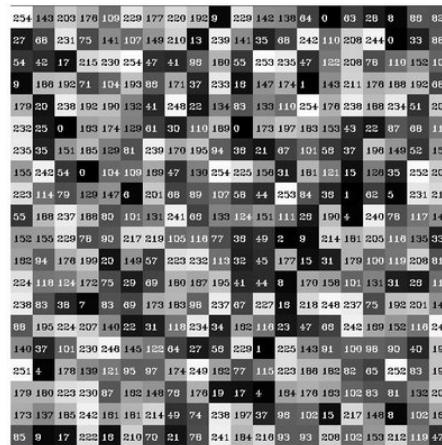


Fig. 2.12: Tonos en una imagen
Elaborado por: Investigador



Fig. 2.13: Histograma de una imagen
Elaborado por: Investigador

La ecualización del histograma trata de repartir los píxeles, de forma que la luminancia de éstos esté más distribuida, consiguiendo, de esta forma, aumentar el contraste y distinguir mejor los distintos objetos que forman la imagen. Como defecto fundamental, la ecualización del histograma tiende a aumentar el ruido. Para exponer el ecualizado del histograma se modifica ligeramente la representación del histograma para asimilarla a la de una función de densidad de probabilidad. Para ello se precisa normalizar el histograma, de manera que los niveles de intensidad

pasen a ser una variable aleatoria r que varíe entre 0 y 1, y el área del histograma normalizado sea igual a la unidad. Para conseguir esto, se normaliza el número de intensidades a valores entre 0 y 1 para normalizar se divide cada elemento del histograma por el número de píxeles de la imagen (para que su suma sea 1). [26][28]

El histograma normalizado se tiene que $P_R(r)$, la probabilidad de que un píxel tenga la intensidad r , se expresa como:

$$P_R(r) = \frac{n_r}{n} \quad (2.4)$$

Donde n es el número total de píxeles en la imagen, y n_r el número de píxeles con nivel de intensidad r

La ecualización del histograma tiene como objetivo obtener un histograma uniforme, es decir, que la probabilidad de cualquier nivel de gris en la imagen sea la misma

$$S_k = T(r_k) = \sum_{j=0}^k P_R(r_j) = \sum_{j=0}^k \frac{n_j}{n} \quad (2.5)$$

La función de transformación cumple la siguiente condición

$$0 \leq T(r) \leq 1 \text{ para } 0 \leq r \leq 1 [26][28]$$

2.2.7.4. Segmentación

Segmentar una imagen digital consiste en dividirla en regiones homogéneas con respecto a una o más características. Es decir, consiste en diferenciar los diferentes objetos y donde se encuentran del fondo, que puede ser tan sencillo como complejo con el fin de facilitar un posterior análisis o reconocimiento automático. Localizar la cara de una persona dentro de la imagen de una fotografía o encontrar los límites de una palabra dentro de una imagen de un texto, constituyen ejemplos de problemas de segmentación. A pesar de que existen diferentes enfoques para realizar la segmentación en la práctica se demuestra que la segmentación no tiene reglas estrictas a seguir, y dependiendo del problema en cuestión, puede ser necesario idear técnicas a medida. [30]

El resultado de la segmentación, produce otra imagen en la que cada pixel tiene asociada una etiqueta distintiva del objeto al que pertenece.

Los diferentes objetos que aparecen en una imagen pueden ser localizados atendiendo a aspectos como: sus contornos o su textura.

La textura

La textura de un objeto dentro de una imagen es el conjunto de formas que se aprecia sobre su superficie y que lo dota de cierto grado de regularidad. Una definición clásica de textura es la siguiente: “uno o más patrones locales que se repiten de manera periódica”. [30]

El contorno

El contorno de un objeto en una imagen digital corresponde al mínimo conjunto de píxeles que separa ese objeto del fondo o background de la imagen. Normalmente estos contornos se corresponden con los puntos donde se producen discontinuidades en los valores de píxeles adyacentes (cambios en el matiz o el brillo) o con los puntos donde cambia un patrón que se repite (cambios de textura).[30]

2.2.7.5. La parametrización o selección de rasgos

Después de la segmentación se continua con la extracción de características de los objetos que van a permitir su clasificación. La selección de características es una etapa en la que se intenta identificar aquellas características que realmente aportan información cualitativa o relevante que son útiles para el proceso que se esté desarrollando. Las características de un objeto pueden ser internas o externas.

Las características externas permiten obtener información de la forma y el tamaño del objeto. Con el contorno se pueden hallar características morfológicas como: área proyectada, perímetro, longitud de los ejes, volumen, diámetro, etc. [31]

Las características internas permite obtener información del color y textura del objeto estas características son extraídas de las propiedades de los píxeles dentro del contorno.[31]

2.2.7.6. El reconocimiento y la interpretación.

El reconocimiento consiste en clasificar las diferentes características de los objetos de la imagen. Los objetos detectados que presenten características semejantes se agrupan en una misma clase.[32]

Se utiliza algoritmos que permiten clasificar los elementos que aparecen dentro de una escena para poder interpretarla. Los algoritmos de clasificación tienen la misión de distinguir entre objetos diferentes de un conjunto predefinido llamado universo de trabajo. Normalmente, el universo de trabajo se considera dividido en una colección K de clases $(\alpha_1, \alpha_2, \dots, \alpha_K)$, perteneciendo los diferentes objetos a algunas de estas

clases. Los métodos que permiten determinar, de manera automática, en qué clase se encuentra un objeto de un universo de trabajo se conocen como **clasificadores**. Para construir un clasificador se tiene en cuenta las características discriminantes y las muestras de aprendizaje. [32]

Características discriminantes

Para realizar el reconocimiento automático de los objetos se realiza una transformación que convierte un objeto del universo de trabajo en un vector X cuyas N componentes se llaman características discriminantes o rasgos. [26]

Estas características deben permitir discriminar a qué clases puede pertenecer cualquier objeto del universo de trabajo.

$$X = (x_1, x_2, \dots, x_N) \text{ con } N \in \mathbb{N} \text{ y } x_i \in \mathbb{R} \forall i = 1 \dots N$$

El valor del vector de características para un objeto concreto se conoce como patrón. Es decir, un patrón es una instancia particular de un vector de características determinado. [26]

Una vez determinadas las características discriminantes para un problema concreto, la clasificación de un objeto comienza por la obtención de su patrón. El siguiente paso consiste en determinar la proximidad o grado de pertenencia de este patrón a cada una de las clases existentes. A este efecto se definen las funciones discriminantes o funciones de decisión como aquellas funciones que asignan a un patrón un grado de semejanza respecto a cada una de las diferentes clases. [26]

La muestra de aprendizaje

Para el cálculo de las funciones discriminantes se tiene la existencia de un conjunto de patrones similares a los que se desea reconocer, que se denomina conjunto de aprendizaje o conjunto de entrenamiento. Los patrones de este conjunto se utilizan a modo de modelos para crear la función discriminante que clasificará correctamente los patrones del universo de trabajo. Por eso, el conjunto de aprendizaje debe estar constituido por un subconjunto representativo del universo de trabajo. [26]

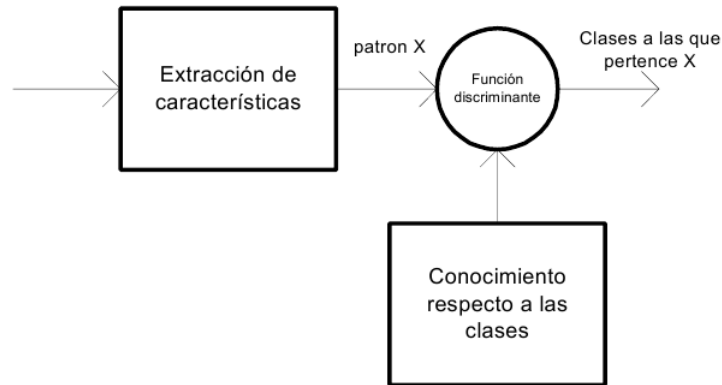


Fig. 2.14: Esquema general de un clasificador de aprendizaje supervisado. [26]

Algoritmo de clasificación Support Vector Machine (SVM)

Las máquinas de soporte vectorial o máquinas de vectores de soporte son un conjunto de algoritmos desarrollados recientemente por Vladimir Vapnik y su equipo en los laboratorios AT&T. Pertenecen a la familia de los clasificadores de aprendizaje supervisado y a su vez es un clasificador lineal puesto que inducen separadores lineales o hiperplanos en espacios de características de muy alta dimensionalidad (introducidos por funciones núcleo o kernel) con un sesgo inductivo muy particular (maximización del margen). Aprendizaje supervisado se refiere a que la máquina aprende el algoritmo a través de la utilización de datos etiquetados. Necesitamos entrenar el algoritmo con una cantidad de los datos que tienen la etiqueta; cada conjunto de datos tiene que tener una clase. [33]

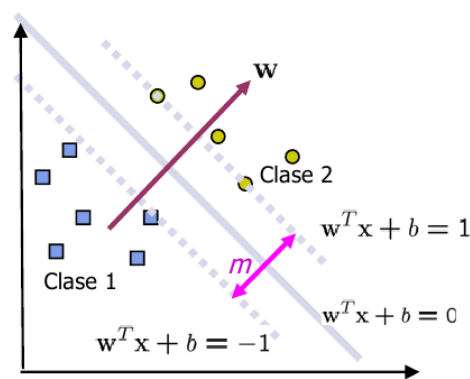


Fig. 2.15: Clasificador SVM [33]

Una SVM primero mapea los puntos de entrada \mathcal{R} a un espacio de características de una dimensión mayor (si los puntos de entrada están en \mathcal{R}^2 entonces son mapeados

por la SVM a \mathfrak{R}^3) y crea uno o más hiperplanos que los separe y maximice el margen m entre cada clase de los datos, la frontera de decisión debe estar tan lejos de los datos de ambas clases como sea como se muestra en la figura 2.15. [33]

La SVM encuentra el hiperplano óptimo utilizando el producto punto con funciones en el espacio de características que son llamadas kernels. La solución del hiperplano óptimo puede ser escrita como la combinación de unos pocos puntos de entrada que son llamados vectores de soporte. [33]

Casos de clasificación de SVM

Linealmente separable

Dado un conjunto S de puntos etiquetados para entrenamiento como se aprecia en al Figura 2.16

$$(y_1, x_1), \dots, (y_i, x_i) \tag{2.6}$$

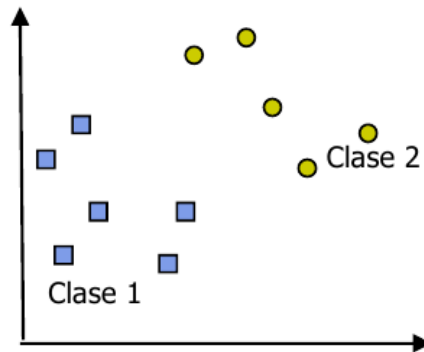


Fig. 2.16: Conjunto de datos linealmente separables. [33]

Cada punto de entrenamiento $x_i \in \mathfrak{R}^N$ pertenece alguna de dos clases y se ha dado una etiqueta $y_i \in \{-1, 1\}$ para $i = 1, \dots, l$. En la mayoría de los casos, la búsqueda de un hiperplano adecuado en un espacio de entrada es demasiado restrictivo para ser de uso práctico. Una solución a esta situación es mapear el espacio de entrada en un espacio de características de una dimensión mayor y buscar el hiperplano óptimo. Sea $z = \varphi(x)$ la notación del correspondiente vector en el espacio de características con un mapeo φ de \mathfrak{R}^N a un espacio de características Z . [33] Se desea encontrar el hiperplano

$$w \cdot z + b = 0 \tag{2.7}$$

Tal que se pueda separar el punto x_i de acuerdo a la siguiente función

$$f(x_i) = \text{sign}(w \cdot z_i + b) = \begin{cases} 1 & y_i = 1 \\ -1 & y_i = -1 \end{cases} \quad (2.8)$$

Donde $w \in Z$ y $b \in \mathfrak{R}$. Más precisamente, el conjunto S se dice que es linealmente separable si existe (w, b) tal que las inecuaciones

$$\begin{cases} (w \cdot z_i + b) \geq 1, & y_i = 1 \\ (w \cdot z_i + b) \leq -1, & y_i = -1 \end{cases} \quad i = 1, \dots, l \quad (2.9)$$

Sean válidas para todos los elementos del conjunto S . Para el caso linealmente separable de S , se puede encontrar un único hiperplano óptimo, para la cual, el margen entre las proyecciones de los puntos de entrenamiento de dos diferentes clases es máximo. [33]

Truco del Kernel para el caso no linealmente separable

Se necesita de una función $K(x_i, x_j)$ llamada kernel que calcule el producto punto de los puntos de entrada en el espacio de características Z (Ver figura 2.17), esto es

$$z_i \cdot z_j = \varphi(x_i) \cdot \varphi(x_j) = K(x_i, x_j) \quad (2.10)$$

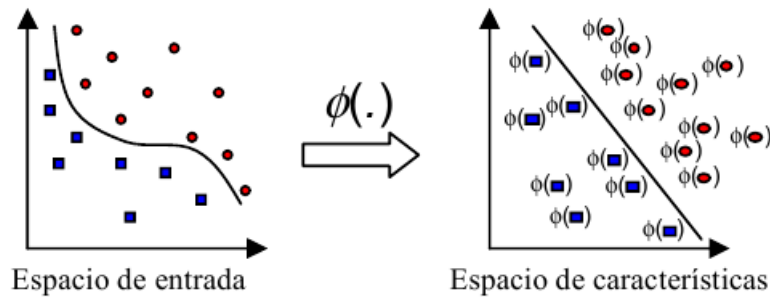


Fig. 2.17: Truco del kernel para el caso no linealmente separable. [33]

Las Funciones que satisfacen el teorema de Mercer pueden ser usadas como productos punto y por ende pueden ser usadas como kernels. Se usa el kernel polinomial de grado d (Ecuación 2.11) para construir un clasificador SVM.

$$K(x_i, x_j) = (1 + x_i \cdot x_j)^d \quad (2.11)$$

Entonces el hiperplano no lineal de separación puede ser encontrado como la solución de

$$\text{Max } W(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad (2.12)$$

$$\sum_{i=1}^l y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, \dots, l$$

y la función de decisión es

$$f(x) = \text{sign}(w \cdot z + b) = \text{sign} \left(\sum_{i=1}^l \alpha_i y_i K(x_i, y_i) + b \right) \quad (2.13)$$

Algoritmo de clasificación K-Nearest Neighbors (K-nn)

El algoritmo de clasificación de los K-Nearest Neighbors (K - vecinos más cercanos) es un clasificador supervisado, se utiliza en aquellos casos en los que la región de clasificación no es lineal ni sigue una curva cónica como se aprecia en la figura 2.18.

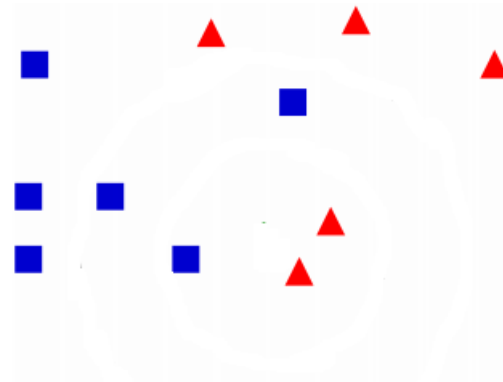


Fig. 2.18: Conjunto de datos de aprendizaje o entrenamiento [34]

Un patrón se clasifica como perteneciente a la clase a la que pertenezca la mayoría de los k vecinos más cercanos. Una medida Distancia euclidiana se utiliza para calcular qué tan cerca este cada patrón del conjunto de entrenamiento. La principal ventaja de este clasificador reside en su potencia. Su principal desventaja se tiene en la dificultad para la elección de k, para la determinación del número de vecinos por clase y su disposición, tareas que en general son eminentemente empíricas. Además, el tiempo de cálculo puede ser significativamente mayor si el número de vecinos llega a ser muy elevado. [26]

La idea que fundamenta este algoritmo es que el nuevo objeto (círculo verde) se clasificará en la clase más frecuente (triángulos rojos o cuadrados azules) de sus K vecinos más cercanos como se aprecia en la figura 2.19. Si $k=3$, entonces la predicción del círculo verde será un triángulo rojo. Si $k=5$, entonces la predicción del círculo verde será un cuadrado azul. Se calculan las distancias de todos los casos ya clasificados al nuevo objeto con la ecuación 2.14.

$$D(X_1, X_2) = \sqrt{\sum_{i=1}^n (X_{1i} - X_{2i})^2} \quad (2.14)$$

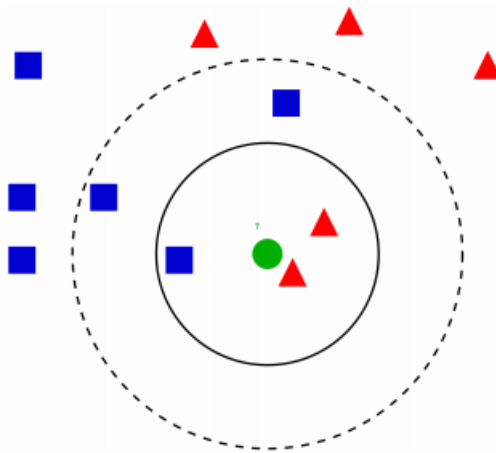


Fig. 2.19: Clasificación del nuevo objeto en la clase más frecuente [34]

Algoritmo Viola & Jones

Paúl Viola y Michael J. Jones desarrollaron un algoritmo que tiene un coste computacional muy bajo, y consta de dos partes principales: clasificador en cascada, que garantiza una discriminación rápida y un entrenador de clasificadores.

Los primeros clasificadores son muy sencillos y permiten rechazar una gran cantidad de objetos que no se está buscando mientras que aceptan un porcentaje muy alto del objeto deseado. La cascada permite desechar gran parte de las regiones de la imagen y sólo concentrarse en las zonas en las que es más probable que haya un objeto que se desea encontrar por lo que la detección es bastante rápida. [35]

Características tipo Haar o Haar-like features

Estas características tipo Haar se definen sobre regiones rectangulares de una imagen en escala de grises. Una característica está formado por un número finito de

rectángulos y su valor escalar consistirá la sumar de los píxeles de cada rectángulo sumados aplicando un cierto factor de peso.

$$característica = \sum_{i=1}^N w_i \cdot suma_rectangul(r_i) \quad (2.15)$$

Donde $\{r_1, \dots, r_N\}$ son los rectángulos que forman la característica y w_i el peso de cada uno. [35][36]

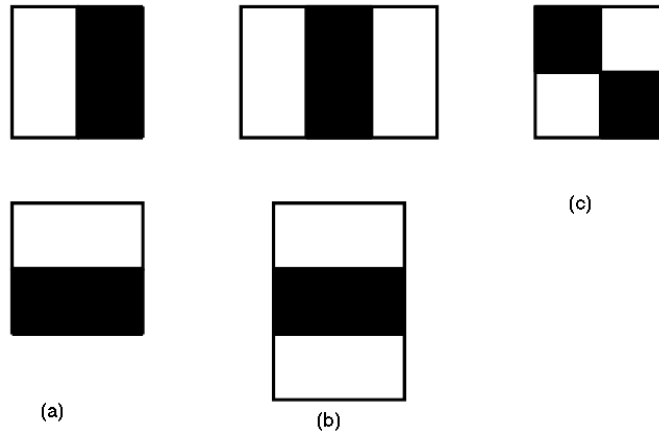


Fig. 2.20: Características tipo Haar. (a) las de bordes, (b) las de líneas, (c) las de forma de X. [35]

El valor de una característica se obtiene sumando todos los píxeles del rectángulo blanco y restándose todos los píxeles del rectángulo negro. Por ejemplo, la característica central de 3 rectángulos trataría de representar que en general la región de los ojos es más oscura que las regiones de alrededor. Como se ve en la figura 2.21.

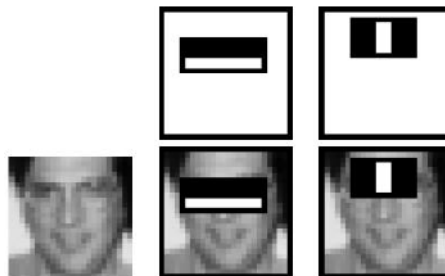


Fig. 2.21: Características seleccionadas por la cascada de detectores de Viola & Jones. [35]

Imagen Integral

La imagen integral como se muestra en la figura 2.22 es el resultado de sumar todos los píxeles desde la esquina superior izquierda (referencia de coordenadas) hasta posición (x, y) .

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (2.16)$$

Donde $ii(x, y)$ es la imagen integral e $i(x', y')$ es la imagen original. [37][36]



Fig. 2.22: Imagen integral. [37]

Clasificador

Una vez extraídas las características de la imagen lo siguiente es clasificar. Para ello se emplea un entrenador para cada característica. Este entrenador es capaz de calcular el umbral óptimo para la clasificación, a través de las muestras positivas y negativas.

$$h_i(x) = \begin{cases} 1 & \text{si } p_i f_i(x) < p_i \theta_i \\ 0 & \text{otro valor} \end{cases} \quad (2.17)$$

Donde:

$h_i(x)$ es el clasificador en función de x , que es la imagen de 24x24 píxeles.

p_i la paridad o la dirección de la inecuación

θ_i el umbral

$f_i(x)$ la característica que estamos evaluando.

Cascada de clasificadores

Los clasificadores se encadenan y en cada etapa se rechaza o se acepta una imagen para continuar en la siguiente etapa véase la figura 2.23. Si atraviesa todas las etapas existe una alta probabilidad de que en esa zona se encuentre el objeto requerido.

Cada etapa de la cascada de clasificadores se entrena haciendo que la tasa de aciertos aumenta lo suficiente y la de falsos positivos desciende drásticamente. [35]

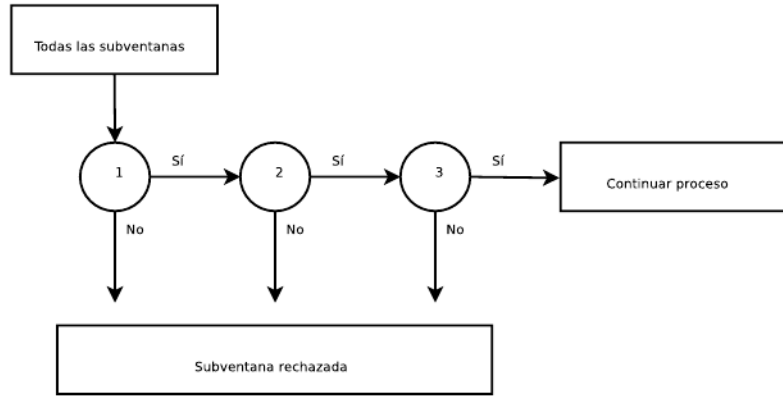


Fig. 2.23: Diagrama de cascada de clasificadores. [35]

2.2.7.7. Seguimiento de Objetos

Cuando tratamos con una fuente de video, en ocasiones tenemos un objeto u objetos en particular que queremos seguir a través del campo de visión artificial una cuestión que tiene dos componentes principales: identificación y modelado.

La identificación, consiste en buscar el objeto en el que estamos interesados en un frame dentro de una secuencia. Técnicas como los momentos o los histogramas de color nos ayudaran a identificar el objeto que estamos buscando. Seguir cosas que todavía no hemos identificado es un problema. El seguimiento de objetos no identificados es importante cuando queremos determinar algo con su movimiento o cuando el movimiento del propio objeto es lo que le hace interesante. Las técnicas de seguimiento de objetos no identificados normalmente involucra el seguimiento de puntos clave muy significantes.

El modelado, nos ayuda con técnicas para conseguir una medida ruidosa de la posición actual del objeto. Algunas de las técnicas matemáticas más potentes han sido desarrolladas estimando la trayectoria de un objeto medido de una manera ruidosa. Estos métodos son aplicables a modelos bidimensionales y tridimensionales de los objetos y sus posiciones. [30]

Template Matching

Template Matching (coincidencia de plantilla) es una técnica de visión artificial de alto nivel que se identifican las partes en una imagen (I) que coincidan con una plantilla predefinida (T). Algoritmos avanzados de Template Matching permiten encontrar las apariciones de la plantilla independientemente de su orientación y el brillo local.

Las técnicas de template matching son flexibles y relativamente sencillos de utilizar, lo que los hace uno de los métodos más populares para la localización de objetos. Su aplicación está limitada por la potencia de cálculo disponible, como la identificación de las plantillas grandes y complejas puede llevar mucho tiempo.

Template Matching es un método en la cual desliza la imagen de la plantilla sobre la imagen de entrada (como en convolución 2D) y compara la plantilla y el parche de imagen de entrada bajo la imagen de la plantilla como se aprecia en la figura 2.24. [38]



Fig. 2.24: Deslizamiento de la imagen plantilla sobre la imagen de entrada. [38]

Para cada localización de T sobre I , se almacena la métrica en la matriz de resultados (R). Cada ubicación (x, y) en R contiene la métrica de partido. La figura 2.25 es el resultado R de deslizar el parche con una métrica TM_CCORR_NORMED . Los lugares más brillantes indican las mayores coincidencias. Como puede ver, la ubicación marcada por el círculo rojo es probablemente el que tiene el valor más alto, por lo que la ubicación (el rectángulo formado por ese punto como una esquina y anchura y altura igual a la imagen parche) se considera el partido. [38]

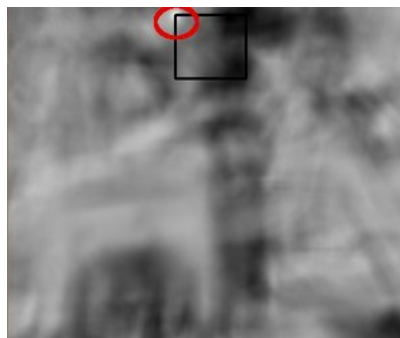


Fig. 2.25: Resultado de deslizar la plantilla sobre la imagen de entrada. [38]

2.3. Propuesta de Solución

Las personas necesitan trasladarse de un lugar a otro, y en el transcurso del viaje debido al cansancio, aburrimiento, uso de medicamentos el conductor pierde su capacidad de vigilia entrando al estado de somnolencia por lo que se implementa un sistema de control del estado de somnolencia en conductores de vehículos para brindar seguridad a las personas que están viajando y disminuir el índice de accidentes.

CAPÍTULO III

METODOLOGÍA

3.1. Modalidad de la investigación

La presente investigación se basó en una metodología de proyecto de investigación aplicada, debido a que se buscó la resolución de un problema y al desarrollo de ideas para conseguir la disminución de accidentes de tránsito a causa de la somnolencia por medio de la puesta en práctica de conocimientos teóricos para llegar a un diseño esquemático y finalmente a la implementación de un sistema electrónico.

El estudio se realizó basándose en la combinación de las modalidades de investigación bibliográfica – documental y de campo.

La investigación bibliográfica – documental se realizó mediante libros, revistas, artículos y tesis para poder guiarse y profundizar en el tema

La investigación de campo se llevó a cabo al monitorear al conductor en un vehículo, en donde se determinó el problema de estudio, se recopiló información suficiente y necesaria para realizar este proyecto.

3.2. Recolección de información

Se recopiló información de fuentes bibliográficas, artículos, tesis y páginas de internet que contienen información referente al tema de investigación, así como la guía del tutor y varios profesores de la FISEI - UTA.

3.3. Procesamiento de la información

Para el procesamiento y análisis de la información se llevan a cabo los siguientes pasos:

- Revisión de la información obtenida, que ayudó a plantear estrategias para la solución del problema a través de la propuesta.
- Interpretación de resultados con el apoyo de fundamentos teóricos para cada aspecto del desarrollo del proyecto.

3.4. Desarrollo del proyecto

Para el desarrollo del presente proyecto se llevó a cabo la siguiente estructura de pasos:

- Investigar en: libros, artículos y revistas, que se refieran a la somnolencia.
- Análisis y selección de los diferentes tipos de equipos para el desarrollo del proyecto de investigación.
- Seleccionar el software más apropiado para la adquisición de señales y procesamiento de imágenes.
- Determinar los algoritmos que permitan el desarrollo del sistema de control del estado de somnolencia.
- Analizar la frecuencia cardíaca, saturación de oxígeno, cambios del rostro (ojos abiertos o cerrados) y movimiento de cabeza, para detectar somnolencia en los conductores de vehículos.
- Implementar el diseño final en la placa de circuito impreso para el sistema de alerta en caso de somnolencia.
- Instalar el sistema en un vehículo.
- Realizar pruebas de funcionamiento del sistema electrónico para la detección de somnolencia.

CAPÍTULO IV

DESARROLLO DE LA PROPUESTA

4.1. Introducción

La somnolencia se define como el transcurso de estado en vigilia al estado de sueño, se produce por excesivas de horas de trabajo, número inadecuado de horas de sueño, conducción nocturna, horarios irregulares de trabajo-descanso, entro otros factores mas.

Según numerosos estudios llevados a cabo, el sueño se encuentra entre las cinco primeras causas de los accidentes con víctimas. La necesidad de dormir es una de las necesidades primarias de las más importantes que tiene el ser humano. Si no se duerme nada o no se duerme lo suficiente, el organismo reacciona con toda una serie de desajustes, muchos de los cuales son altamente peligrosos para el conductor. [39]

Algunos síntomas se pueden vislumbrase con antelación, como pueden ser cambios corporales, que se traducen en continuos movimientos y cambios de postura, así como parpadeo constante, sensación de pesadez de cabeza, restregarse la cara, estirarse, etc... Se puede observar torpeza, cierta dificultad para mantener y concentrar la atención, menor número de correcciones de dirección, aumento en el tiempo de reacción de las respuestas. [39]

4.2. Análisis de requerimientos

4.2.1. Hardware

El presente proyecto está formado por dos partes una que es analizar cambios faciales (ojos abiertos o cerrados), orientación y giro del rostro para la cual se emplea un sistema de visión artificial que está constituido por las siguientes elementos cámara, iluminación, ordenador y actuadores; y para la siguiente parte que es la de analizar la oxigenación de la sangre o nivel de saturación de oxígeno y el ritmo cardíaco.

Cámara

Existe variedad de cámaras web con diferentes especificaciones y precio, en la tabla 4.1 se hace la comparación de algunas cámaras web que se puede comprar dentro de la ciudad.

Tabla 4.1: Comparación de cámaras web

Cámaras Web	Omega 300k	Genius FaceCam 321	Logitech C170
Resolución de video	600 x 200@ 15fps 640 x 480@ 30fps	VGA@ 30fps	XVGA (1024 x 768) @ 30fps
Sensor	CMOS (300k píxeles)	CMOS (VGA pixel)	CMOS de 1280 x 720 (HD)
Compatible	Microsoft Windows 2000, XP, Vista, 7, Linux	Windows7/Vista/ XP SP2, Mac/ Linux	Windows 8, 7, Vista, XP, 2000, Linux y MAC
Conector	USB	USB	USB
Otros	Exposición automática/ balance de blanco/ control de color	Enfoque manual	Exposición automática/ balance de blanco/ control de color
Precio	\$ 16,00	\$ 22,00	\$ 36,00

Elaborado por: Investigador

Después de haber comparado las diferentes características que tiene cada cámara web, se eligió la cámara Omega 300k por exposición automática de balance de blancos y control de color y su precio es bajo.

Ordenador

El ordenador esta ubicado en la cabina del auto y debido al espacio disponible en este lugar se necesita de ordenadores de una placa (SBC) el cual será encargado de procesar los algoritmos del sistema de control de estado de somnolencia. En la tabla 4.2 se hace la comparación de SBC.

Tabla 4.2: Comparación de ordenadores de una placa (SBC)

Características	Raspberry Pi mod B	Cubieboard 1 [2]*	Cubietruck
SoC	Broadcom BCM2835	Allwinner A10 [A20]	Allwinner A20
CPU	Arm11-v6 700Mhz	Cortex-A8 1Ghz 256Kb L2 cache [Dual core A7 512Kb L2 cache]	ARM Cortex-A7 Dual Core 512Kb L2 cache
RAM	215 Mb	1 Gb DDR3 480 Mhz	2 Gb DDR3 480 Mhz
SD	SD slot	microSD slot hasta 64 Gb	microSD slot hasta 64 Gb
Nand Flash	-	4 Gb	8 Gb
SATA	-	hasta 2 Tb	hasta 2 Tb
Video output	HDMI 1.3/1.4 1080p, Video compuesto RCA	HDMI 1.3/1.4 1080p	HDMI 1.3/1.4 1080p, VGA
USB	2 x USB 2.0	2 x USB 2.0 + 1 mini USB 2.0 OTG	2 x USB 2.0 + 1 mini USB 2.0 OTG
Red ethernet WIFI Bluetooth	RJ-45 Eth 10/100	RJ-45 Eth 10/100	RJ-45 Eth 10/100/1000 WIFI 802.11bng Bluetooth
P. serie UART	-	1	1
GPIO	26 pins	96 pins	54 pins
Toma de Corriente	5V 1A	5V 2A	5V 2.5A

[]* Diferencia que existe entre la cubieboard 1 y la cubieboard 2

Elaborado por: Investigador

Una vez analizado las características técnicas se escogió la Cubietruck por su capacidad de procesamiento y almacenamiento.

Iluminación

Se necesita iluminar el ambiente durante la noche para el cual se utiliza un anillo de leds infrarrojos que proporcionan un haz de luz en la longitud de onda de 850 nm, debido a que no son perceptibles para el ojo humano evitando encandelillar al conductor debido a que se emplea el método de iluminación frontal.

Actuadores

Como actuador para alertar al conductor que se está quedando dormido se utiliza un zumbador y para visualizar el proceso del sistema de detección de somnolencia al conductor se utiliza una pantalla LCD de 7'' touch que viene como parte del kit de la Cubietruck que está ubicada en el tablero del vehículo.

Convertor 12Vcc a 5Vcc @ 3A

La batería del vehículo suministra un voltaje de 12Vcc y la Cubietruck necesita de un voltaje de 5Vcc a 2.5A por lo que se necesita de un convertor de voltaje de 12Vcc a 5Vcc para su funcionamiento.

En la figura 4.1 se muestra un regulador de voltaje de alta corriente basado en el LM317 pero con un bypass transistorizado 2N3055 que le permitirá a este regulador manejar corrientes muy por encima del nivel de operación del LM317. Como sistema de protección se colocó un fusible de 3A y se realizó un circuito comparador de voltaje, en caso de que el voltaje de salida sea superior a los 5V se activa un relé desconectando la salida de alimentación de la Cubietruck.

Sensores de luz para la construcción de un oxímetro de pulso

Fotorresistencia o LDR

Una fotorresistencia es un componente electrónico cuya resistencia disminuye con el aumento de intensidad de luz incidente.

El valor de la fotorresistencia (en Ohmios) no varía de forma instantánea cuando se pasa de luz a oscuridad o viceversa. [40]

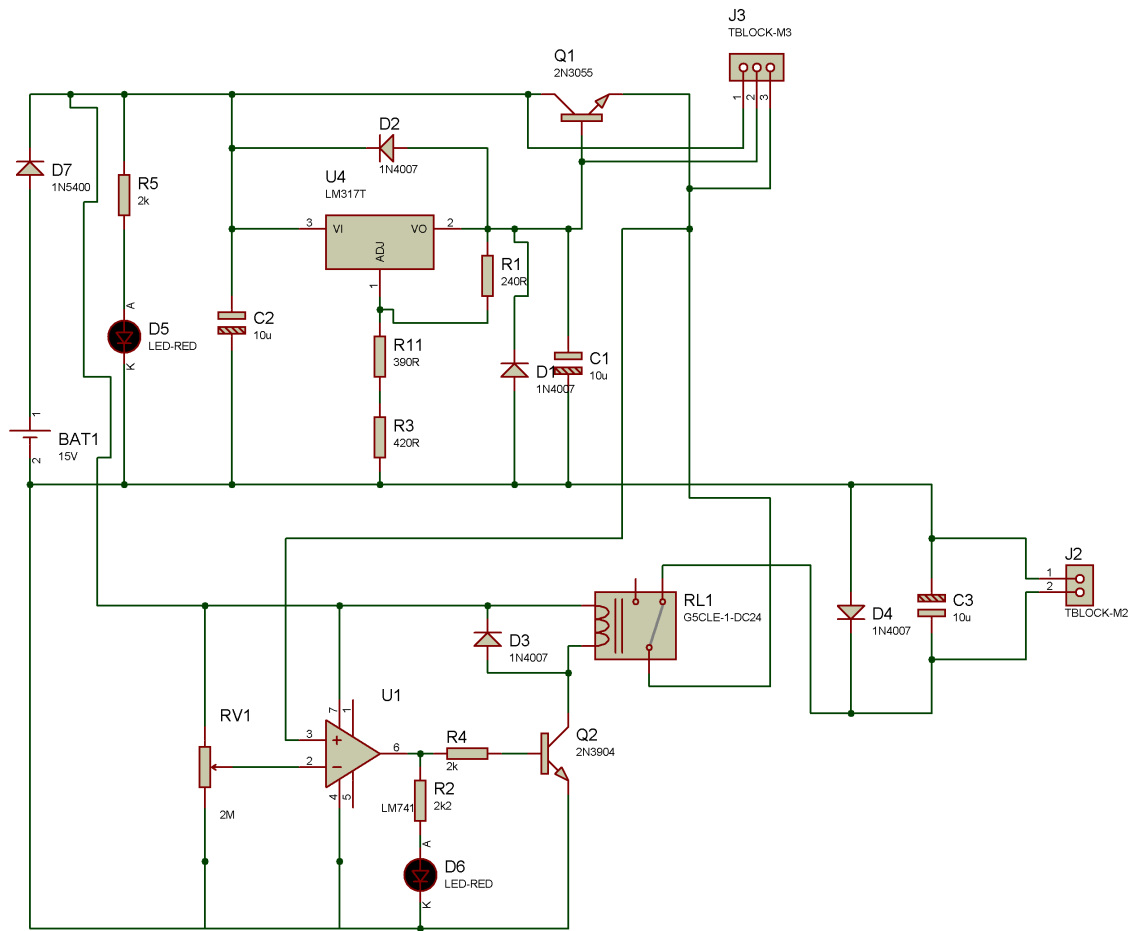


Fig. 4.1: Conversor 12Vcc a 5Vcc @ 3A
 Elaborado por: Investigador

Fotodiodo

El fotodiodo es un dispositivo que conduce una cantidad de corriente eléctrica proporcional a la cantidad de luz que lo incide.

A diferencia del LDR o fotorresistencia, el fotodiodo responde a los cambios de oscuridad a iluminación y viceversa con mucha más velocidad, y puede utilizarse en circuitos con tiempo de respuesta más pequeño. Esta cualidad también la tienen los fototransistores. [40]

Fototransistor

La luz incide sobre la región de base, generando portadores en ella. Esta carga de base lleva el transistor al estado de conducción. El fototransistor es más sensible que el fotodiodo por el efecto de ganancia propio del transistor. [40]

Sensores de luz integrados CMOS

TSL235R

El TSL235R es un convertidor de luz a frecuencia, combina un fotodiodo de silicio y un convertidor de corriente a frecuencia en un solo circuito integrado CMOS monolítica (Figura 4.2). La salida es una onda cuadrada (ciclo de trabajo del 50 %) con una frecuencia directamente proporcional a la intensidad de luz (irradiancia) en el fotodiodo. La salida digital permite interfaz directa a un microcontrolador u otra circuitería lógica. El dispositivo ha sido compensado en temperatura para el rango de luz ultravioleta a visible de 320 nm a 700 nm y responde sobre el rango de luz de 320 nm a 1050 nm. El TSL235R se caracteriza por el funcionamiento en todo el rango de temperatura de $-25\text{ }^{\circ}\text{C}$ a $70\text{ }^{\circ}\text{C}$. [41] Para más información, en el anexo C se tiene el Datasheet del dispositivo.

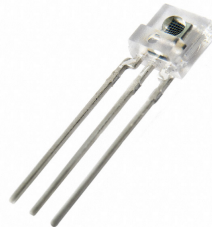


Fig. 4.2: Circuito integrado CMOS TSL235R [41]

TSL257

El TSL257 es un convertidor óptico de bajo nivel de ruido y de alta sensibilidad a la luz a voltaje que combina un fotodiodo y un amplificador de transimpedancia en un solo circuito integrado CMOS monolítica. La tensión de salida es directamente proporcional a la intensidad de luz (irradiancia) en el fotodiodo. El TSL257 tiene una ganancia de transimpedancia de $320\text{ M}\Omega$. El dispositivo ha mejorado la estabilidad de tensión de offset y de bajo consumo de energía. [42]



Fig. 4.3: Circuito integrado CMOS TSL257 [42]

4.2.2. Software

Se utilizó el entorno de desarrollo integrado (IDE) Eclipse, para el lenguaje de programación C++ al que se le asoció librerías de OpenCV. OpenCV (Open Source Computer Vision) es liberado bajo una licencia BSD y por lo tanto es gratis, tanto para uso académico y comercial. OpenCV fue diseñado para la eficiencia computacional y con un fuerte enfoque en aplicaciones en tiempo real. Escrito y optimizado en C/C++, adoptado en todo el mundo, OpenCV tiene más de 47 mil personas de la comunidad de usuarios y el número estimado de descargas superiores a 9 millones. Rangos de uso del arte interactivo, a la inspección de minas, mapas de costura en la web o a través de la robótica avanzada.

4.3. Distribuciones de Linux para Cubietruck

Las distribuciones de Linux para la Cubietruck son varias como Lubuntu, Android, Fedora pero el más popular y el que mejor desempeño y estabilidad tiene en estas placas es Ubuntu es un sistema operativo basado en GNU/Linux y que se distribuye como software libre, el cual incluye su propio entorno de escritorio denominado Unity. Está orientado al usuario novel y promedio, con un fuerte enfoque en la facilidad de uso y en mejorar la experiencia de usuario. Está compuesto de múltiple software normalmente distribuido bajo una licencia libre o de código abierto.

4.4. Descripción general del sistema

El Sistema de control del estado de somnolencia tiene como finalidad alertar al conductor cuando se esté quedando dormido. El sistema está compuesto por: la etapa de monitoreo, para la detección de síntomas de somnolencia en el conductor, para ello se utiliza una cámara Web a la que se ha modificado para una visión nocturna para adquirir el video en tiempo real y para la adquisición de señal de Oximetría mediante el paso de luz roja e infrarroja en la base arteriolar del dedo índice. Para la etapa de procesamiento se utiliza una Cubietruck, se trata de una Single Board Computer (SBC) con un procesador ARM que puede funcionar con distintas distribuciones de Linux, en ella se programó todos los algoritmos necesarios para detectar si el conductor tiene síntomas de somnolencia. Por último se tiene la visualización del sistema de monitoreo mediante una pantalla de 7'' y una alerta sonora para advertir al conductor que se está quedando dormido.

Diagrama de bloques de funcionamiento del sistema de control del estado de somnolencia

El sistema consta dos partes, la primera parte trata de la visión artificial, se inicia con la captura de video en tiempo real por medio de una cámara web, continua con un pre-procesamiento para la detección del rostro si el rostro no es detectado se regresa a una nueva búsqueda, una vez detectado se realiza el seguimiento del mismo cuando se pierde la búsqueda se inicia nuevamente la detección del rostro. Una vez que se sigue el movimiento del rostro se puede detectar los ojos para posteriormente clasificarlos y determinar la orientación y el giro del rostro. La segunda parte corresponde a la oximetría de pulso, un sensor compuesto por un led rojo, un led infrarrojo y un fotodiodo, permiten la lectura del pulso y frecuencia cardíaca que son procesados para estimar los cambios que tienen la frecuencia cardíaca y la oxigenación de la sangre cuando se pasa de vigilia a sueño.

En la figura 4.4 se observa el diagrama de bloques del sistema para la detección de somnolencia y distracción.

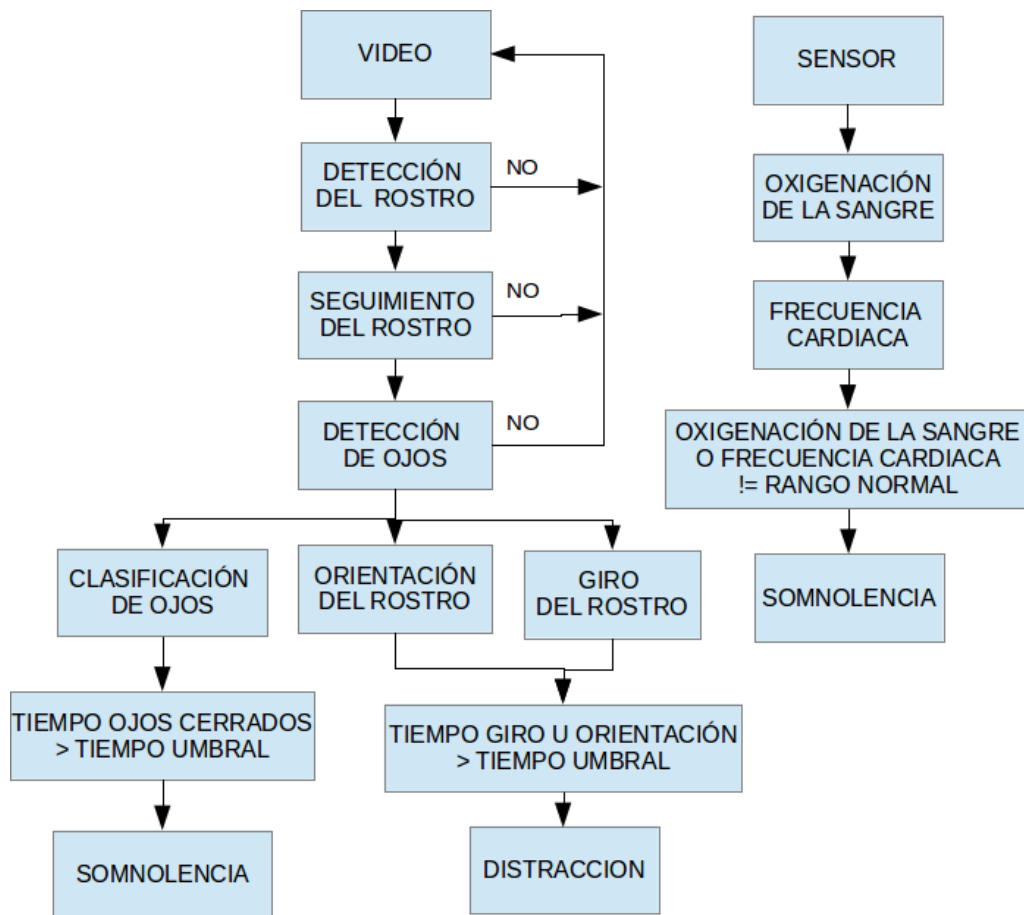


Fig. 4.4: Diagrama de bloques del funcionamiento del sistema de control del estado de somnolencia

Elaborado por: Investigador

4.5. Adquisición de la señal para la oximetría de pulso

El oxímetro de pulso necesita dos longitudes de onda diferentes para realizar mediciones. Estas longitudes de onda se generan utilizando dos diodos emisores de luz, un led rojo (660 nm), y un led infrarrojo (940 nm). Las muestras no pueden tomarse al mismo tiempo, porque sólo hay un fotodetector para dos señales, por lo tanto señales deben ser multiplexadas. Para seleccionar la longitud de onda de la señal a muestrear (led rojo o led infrarrojo) se realiza mediante las salidas GPIO (10 y 11) del Arduino que conmuta un puente H, en primera instancia se polariza al led rojo ánodo - cátodo y al led infrarrojo cátodo - ánodo durante un tiempo, para luego intercambiar la polarización al led rojo cátodo - ánodo y el led infrarrojo ánodo - cátodo así sucesivamente. El puente H también permite suministrar la energía suficiente al led para trabajar.

El fotodetector que se utilizó es un convertidor de luz a frecuencia (TSL235),

combina un fotodiodo de silicio y un convertidor de corriente a frecuencia en un solo encapsulado, su salida va conectada directamente a un pin de entrada del Arduino.

En la figura 4.5 se muestra el circuito esquemático del oxímetro de pulso

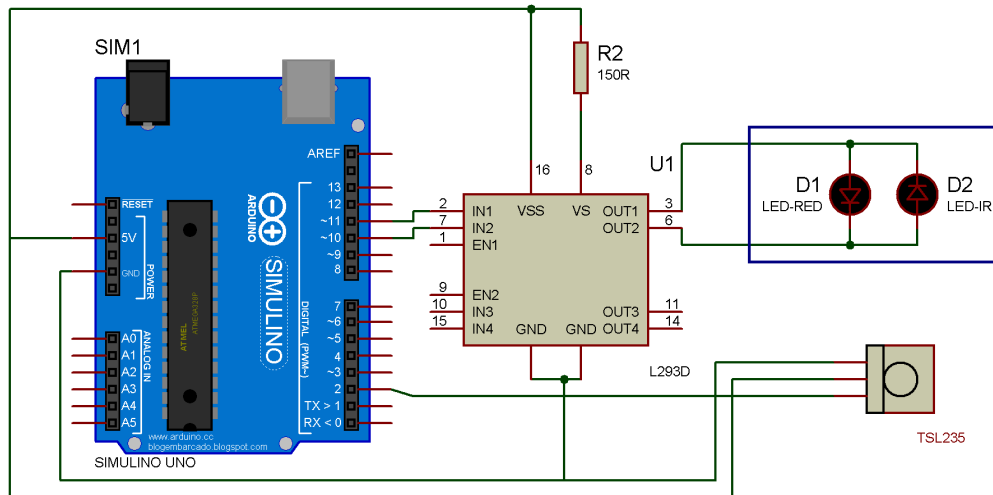


Fig. 4.5: Circuito esquemático del oxímetro de pulso
Elaborado por: Investigador

4.6. Desarrollo del software para la detección de somnolencia

A continuación se describen los pasos que se llevo a cabo para desarrollar el software para la detección de somnolencia en el anexo A se encuentra el diagrama de flujo y en el anexo B el código completo del sistema.

4.6.1. Acceso a la cámara web y visualización de vídeo

A menudo, tenemos que capturar video con la cámara. OpenCV proporciona una interfaz muy simple de esto. Vamos a capturar un vídeo de la cámara y mostrarlo.

Para capturar un vídeo, es necesario crear un objeto `cv::VideoCapture` (método de OpenCV para acceder al dispositivo de cámara). Su argumento puede ser el índice de dispositivo o el nombre de un archivo de vídeo. Índice de dispositivo es sólo el número para especificar qué cámara necesita iniciar la captura de video en caso de tener varias cámaras conectadas, índice cero (0) como el número de identificación de la cámara por defecto.

Si desea mostrar una ventana de interfaz gráfica de usuario en la pantalla usando OpenCV, se llama `cv::imshow()` para cada imagen que se desee visualizar,

pero también hay que llamar `cv::waitKey()` una vez por trama, de lo contrario las ventanas no se actualizarán en absoluto. Llamando `cv::waitKey(0)` espera indefinidamente hasta que el usuario pulsa una tecla en la ventana, pero un número positivo como `cv::waitKey (50)` o más van a esperar por lo menos algunos milisegundos.

Nota: Las funciones de OpenCV inician con `cv::` para evitar anteponer esta identificación en cada función se escribe la siguiente línea de código `using namespace cv` después del llamado de las librerías de igual manera para las funciones que pertenecen a la librería `std` se escribe la siguiente línea de código `using namespace std;`

A continuación el código con el cual se captura y se visualiza el vídeo de la Webcam.

```
1 int main()
2 {
3 Mat frame;
4 VideoCapture camara(0);
5 // bucle infinito para la captura y visualización de video
6 // hasta que se presione una tecla para finalizar
7     for(;;) {
8         camara >> frame;
9         // Mostrar video de la webcam
10        imshow("video", frame);
11        // presionar una tecla para terminar el proceso
12        if(waitKey(50) >= 0) break;
13    }
14 }
```

4.6.2. Procesos para la detección de rostro y ojos

En el año 2001 Viola y Jones inventó el clasificador cascada Haar-based para la detección de objetos, y en el 2002 fue mejorado por Lienhart y Matt. El resultado es un detector o clasificador de objetos bastante rápido (puede detectar rostros y ojos en tiempo real en un ordenador de escritorio típico con una cámara web VGA) además es fiable (detecta aproximadamente el 95 por ciento de los objetos correctamente). Este detector de objetos revolucionó el campo de reconocimiento de rostros (así como el de la robótica y visión por computador en general), ya que finalmente permitió en tiempo real la detección de rostros, Lienhart escribió el código del detector de

objetos que viene gratis con OpenCV.

Los detectores Haar-based pueden ser entrenados de forma automática para encontrar caras de un gran conjunto de imágenes, con la información almacenada en archivos XML. Estos clasificadores en cascada están capacitados típicamente usando al menos 1.000 imágenes únicas de la cara y 10.000 imágenes no cara (por ejemplo, fotos de árboles, coches, y texto), el proceso de formación del clasificador puede llevar mucho tiempo. OpenCV viene con algunos detectores Haar predeterminados para su uso, se puede detectar las caras frontales, perfil, los ojos o la nariz cargando diferentes archivos XML del clasificador cascada.

Carga los detectores de Haar para la detección de la cara y de los ojos

Para llevar a cabo objeto o detección de rostros, primero se debe cargar el archivo XML utilizando la clase `CascadeClassifier` predeterminado de OpenCV de la siguiente manera:

```
1 CascadeClassifier face_cascade;  
2 CascadeClassifier clasificador_ojo_izq, clasificador_ojo_dere  
   ;  
3  
4 face_cascade.load("haarcascade_frontalface_alt2.xml");  
5 clasificador_ojo_izq.load("haarcascade_lefteye_2splits.xml");  
6 clasificador_ojo_dere.load("haarcascade_righteye_2splits.xml "  
   );
```

La detección del rostro utilizando el clasificador Haar

Ya cargado el clasificador se puede utilizar para detectar los rostros en cada nuevo frame capturado por la cámara, pero antes se debe hacer algo de procesamiento inicial en la imagen capturada para la detección de rostros, para ello se realiza los siguientes pasos:

- Conversión de color a escala de gris.
- Ecuilización del histograma.

Conversión de color a escala de grises

La detección de rostros sólo funciona en imágenes en escala de grises. Así que se tiene que convertir el frame de color a escala de grises.

Para convertir fácilmente una imagen de color RGB a escala de grises se utiliza la función `cvtColor()`, como se muestra en el siguiente código:

```
1 Mat frame_gris; // Matriz en la que se almacena la conversión
   en escala de gris
2 cvtColor(frame, frame_gris, CV_BGR2GRAY);
```

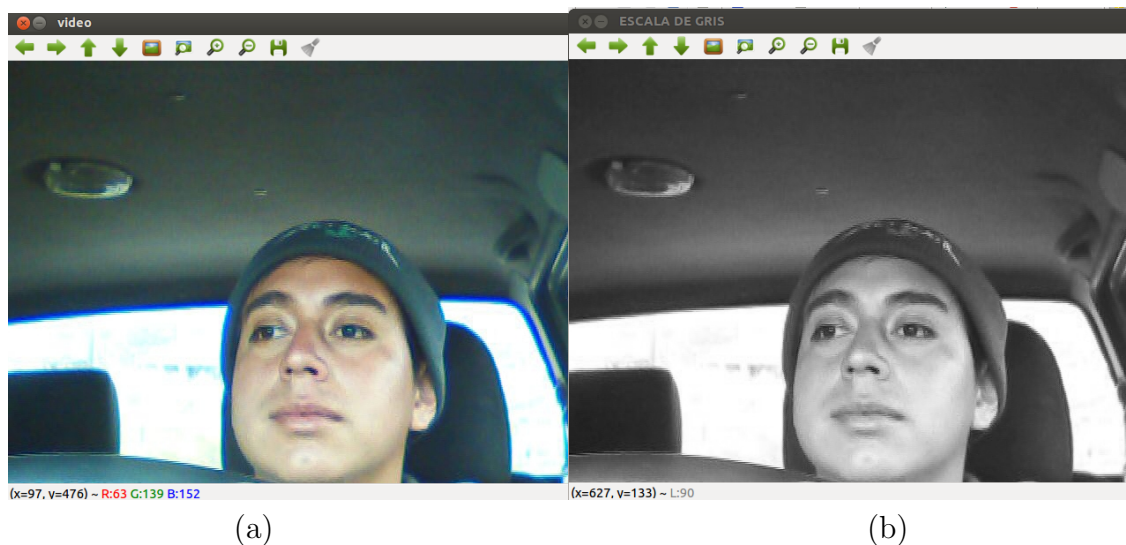


Fig. 4.6: Conversión de color a escala de grises. (a) imagen RGB. (b) imagen en escala de grises.

Elaborado por: Investigador

Ecuación del Histograma

La detección de la cara no es tan fiable en condiciones de poca luz. Así que se realiza la ecualización del histograma para mejorar el contraste y el brillo.

La ecualización del histograma se realiza utilizando la función `equalizeHist(inputImg, equalizedImg)`. A veces, esto hará que la imagen tenga un aspecto extraño, pero en general se debe mejorar el brillo y el contraste y ayudar a la detección de rostros. En la figura 4.7 se puede ver la ecualización del histograma.

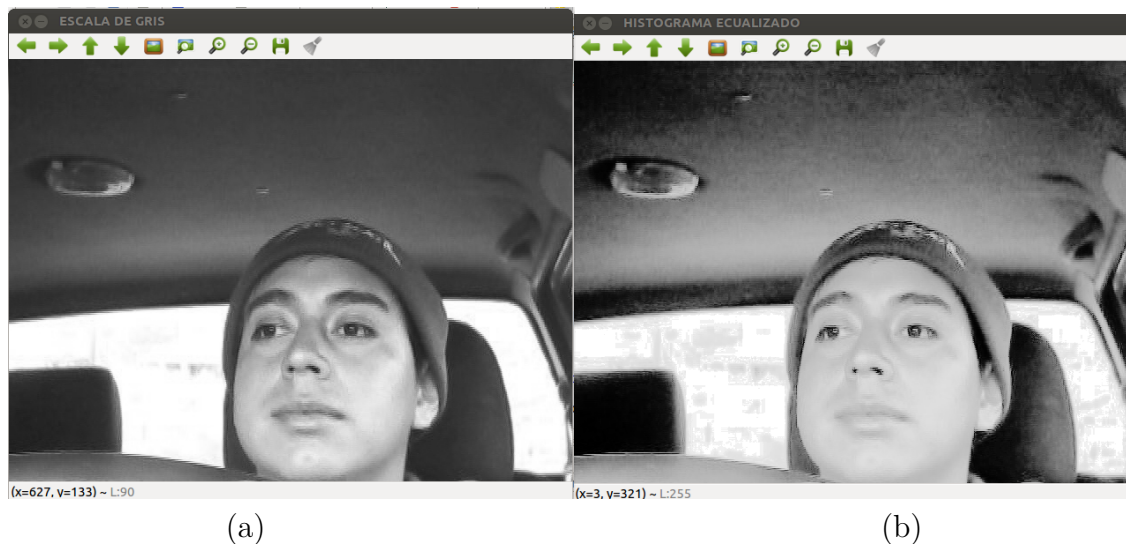


Fig. 4.7: Ecuación del histograma. (a) imagen de entrada en escala de grises. (b) imagen ecualizada.

Elaborado por: Investigador

La detección de la cara y ojos

Convertido la imagen a escala de grises y ecualizado el histograma, queda por detectar las caras utilizando la función `cascadeClassifier::detectMultiScale()`. La salida de la función `detectMultiScale()` será un `std::vector` del tipo de objeto `cv::Rect`. Por ejemplo, si detecta dos caras entonces almacenará un conjunto de dos rectángulos en la salida. La función `detectMultiScale()` tiene la siguiente sintaxis:

```

1 std::vector<Rect> objeto_detectado;
2 Detector.detectMultiScale(img_entrada, objeto_detectado,
   searchScaleFactor, minNeighbors, flags, minFeatureSize);

```

Los parámetros que se establecen a esta función son:

- **img_entrada**: La imagen donde se desea encontrar el objeto, en nuestro caso encontrar una cara
- **objeto_detectado**: Este parámetro devuelve un vector de rectángulos con la ubicación de la cara detectada.
- **minFeatureSize**: Este parámetro determina el tamaño mínimo de la cara que se puede encontrar en la imagen, típicamente 20 x 20 o 30 x 30 píxeles, pero esto depende del caso de uso y tamaño de la imagen. Si está realizando la detección de rostros en una Webcam o un teléfono inteligente donde la cara

estará siempre muy cerca de la cámara, se puede ampliar esto a 80 x 80 para tener detecciones mucho más rápido.

- **searchScaleFactor**: El parámetro determina el número de tamaños diferentes de caras a buscar; normalmente sería 1,1 para una buena detección, o 1.2 para la detección más rápida pero que no encuentra la cara con tanta frecuencia.
- **minNeighbors**: Este parámetro determina que tan seguro que debe ser el detector de que se ha detectado una cara, por lo general un valor de 3, pero se puede configurar a un número mayor si se quiere caras más fiables, incluso si no se detectan muchas caras.
- **flags**: Este parámetro permite especificar si mirar el objeto de mayor tamaño (**CASCADE_FIND_BIGGEST_OBJECT**). Otros parámetros que se pueden agregar para hacer la detección de un uno por ciento o dos por ciento más rápido, como **CASCADE_DO_ROUGH_SEARCH** o **CASCADE_SCALE_IMAGE**.

El siguiente código permite detectar rostros y ojos.

```
1 vector<Rect> rostro;
2 face_cascade.detectMultiScale(video_gris, rostro, 1.1, 2, 0|
   CV_HAAR_SCALE_IMAGE, Size(150,150));
3
4 vector<Rect> ojo_izq, ojo_dere;
5 clasificador_ojo_izq.detectMultiScale(rostro, ojo_izq, 1.1,
   2, 0|CV_HAAR_SCALE_IMAGE, Size(40,40));
6 clasificador_ojo_dere.detectMultiScale(rostro, ojo_dere, 1.1,
   2, 0|CV_HAAR_SCALE_IMAGE, Size(40,40));
```

Para dibujar un rectángulo sobre el rostro y los ojos detectados tenemos a continuación el código, el resultado se muestra en la figura 4.8.

```
1 rectangle(frame, region_rostro, CV_RGB(0,255,0),4);
2 Rect ojo_dere(region_rostro.x+region_ojo_dere.x,region_rostro
   .y+region_ojo_dere.y+region_rostro.height*0.1,
   region_ojo_dere.width, region_ojo_dere.height);
3 rectangle(frame, ojo_dere, CV_RGB(255,30,0),2);
```

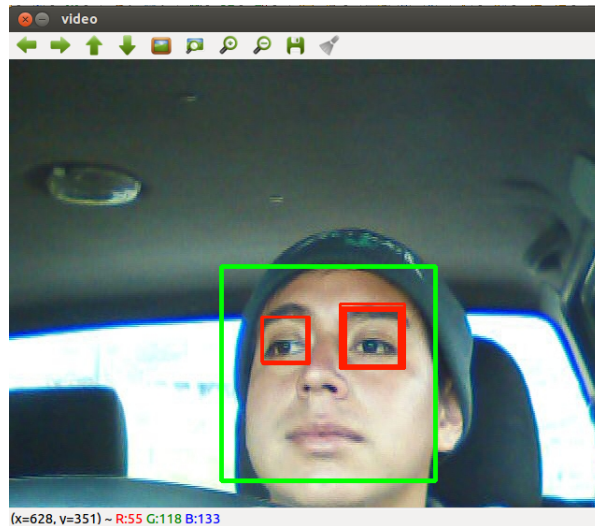


Fig. 4.8: Detección de rostro y ojos durante la conducción
Elaborado por: Investigador

4.6.3. Seguimiento del rostro

Para el seguimiento del rostro se utiliza la Template Matching es un método para buscar y encontrar la ubicación de una imagen de plantilla en una imagen más grande. OpenCV viene con una función `cv2.matchTemplate()` para este propósito. La función retorna una imagen en escala de grises, donde cada píxel indica cuánto hace la vecindad de ese partido de píxeles con la plantilla.

La imagen de entrada donde se buscara el rostro como se muestra en la figura 4.9 es de un tamaño $(W \times H)$ y la imagen de plantilla (rostro) como indica la figura 4.10 es de tamaño $(w \times h)$, la imagen de resultado tendrá un tamaño de $(W-w+1, H-h+1)$ como muestra la figura 4.11.

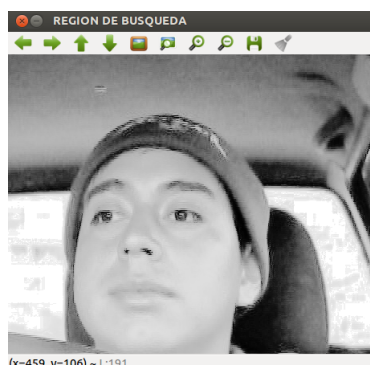


Fig. 4.9: Región de búsqueda del rostro
Elaborado por: Investigador

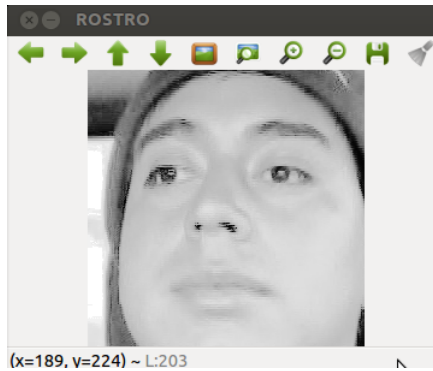


Fig. 4.10: Plantilla del rostro para su búsqueda
Elaborado por: Investigador

El siguiente código realiza lo antes mencionado.

```

1 Mat frame_gris, rostro_encontrado;
2 Rect region_rostro;
3
4 Size dimension_rostro(region_rostro.width, region_rostro.
   height);
5 Rect region_seguimiento_rostro(region_rostro +
   dimension_rostro - Point(dimension_rostro.width/4,
   dimension_rostro.height/2));
6
7 region_seguimiento_rostro &= Rect(0, 0, frame_gris.cols,
   frame_gris.rows);
8 // se crea la matriz de resultados que almacenará los
   coincidencia para cada ubicación plantilla.
9 Mat resultado(region_seguimiento_rostro.width -
   rostro_encontrado.rows + 1, region_seguimiento_rostro.
   height - rostro_encontrado.cols + 1, CV_32FC1);
10 // Realiza la operación de comparación de plantillas:
11 matchTemplate(frame_gris(region_seguimiento_rostro),
   rostro_encontrado, resultado, CV_TM_SQDIFF_NORMED);

```

Una vez que se tiene el resultado (figura 4.11), se utiliza la función `cv2.minMaxLoc()` para encontrar dónde está el valor máximo/mínimo. Estos valores son tomados como la esquina superior izquierda del rectángulo y `(w, h)` como la anchura y la altura del rectángulo. Ese rectángulo es la región de la plantilla. El siguiente código permite encontrar el valor mínimo y máximo en la imagen de resultado.



Fig. 4.11: Matriz de resultado que contiene la coincidencia del rostro
Elaborado por: Investigador

```

1 // Localizar los valores máximo y mínimo en la matriz de
  resultado utilizando minMaxLoc.
2 double minval, maxval;
3 Point minloc, maxloc;
4 minMaxLoc(resultado, &minval, &maxval, &minloc, &maxloc);
5 // Para el método CV_SQDIFF_NORMED el mejor de los partidos
  son los valores más bajos.
6 if (minval <= 0.2) {
7     region_rostro.x = region_seguimiento_rostro.x + minloc.x;
8     region_rostro.y = region_seguimiento_rostro.y + minloc.y;
9 }
10 else
11     region_rostro.x = region_rostro.y = region_rostro.width =
    region_rostro.height = 0;

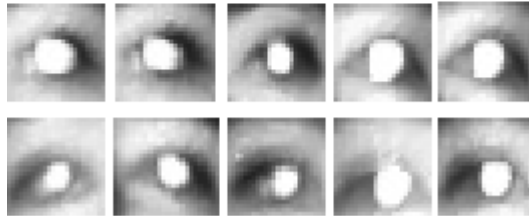
```

4.6.4. Clasificación de ojos abiertos o cerrados

La primera tarea antes que cualquier clasificación es la formación del clasificador; este trabajo se lleva a cabo antes de iniciar la aplicación principal. Esto no es un trabajo fácil, ya que requiere una cantidad suficiente de datos para entrenar el sistema, un conjunto de datos más grande no siempre implica los mejores resultados. Para la formación del clasificador, se utilizó la base de datos `dataset_B_Eye_Images` a la que se le agregó más imágenes de ojos abiertos y ojos cerrados tomados durante el proceso de segmentación, en la figura 4.12 y en la figura 4.13 se muestran los ejemplos de la base de datos creado.



(a)



(b)

Fig. 4.12: Base de datos ojos abiertos: (a) Ojos abiertos durante el día. (b) Ojos abiertos durante la noche

Elaborado por: Investigador



(a)



(b)

Fig. 4.13: Base de datos ojos cerrados: (a) Ojos cerrados durante el día. (b) Ojos cerrados durante la noche

Elaborado por: Investigador

Una vez preparada la base de datos con 1350 imágenes de ojos cerrados y 2180 imágenes de ojos abiertos de 24 x 24 píxeles para el ojo izquierdo en diferentes condiciones de iluminación, de igual manera para el ojo derecho, se almacena todos los datos de formación de cada imagen que se procesa, en un archivo llamado `ojo_izquierdo.xml` y un archivo `ojo_derecho.xml` para su uso directamente con

la función `CvKNearest` en el código principal. En el Anexo C se tienen el código que crea dichos archivos con los Training Data o muestras de aprendizaje de la base de datos con sus respectivas etiquetas, 1 para los ojos cerrados y -1 para los ojos abiertos.

En la figura 4.14 se puede apreciar la distribución de las muestras de aprendizaje que se a creado para el clasificador, los puntos rojos tienen la etiqueta 1 para los ojos cerrados y los puntos verdes tienen la etiqueta de -1 que corresponde a los ojos abiertos.

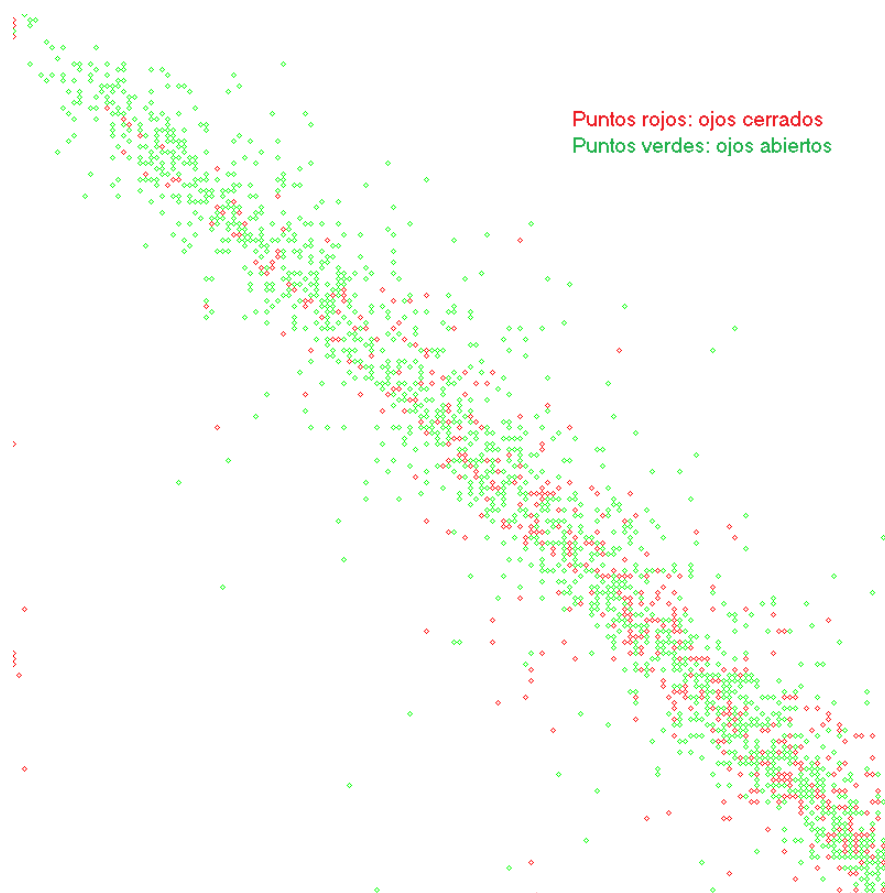


Fig. 4.14: Muestras de aprendizaje de ojos abiertos y cerrados
Elaborado por: Investigador

Para gestionar los archivos `ojo_izquierdo.xml` y `ojo_derecho.xml` se utiliza la clase `FileStorage`, esta clase permite almacenar y leer variables y estructuras de OpenCV. Con esta función, puede leer las clases de la matriz y de formación de capacitación de datos y guardarlo. A continuación se muestra el código que permite leer el archivo que contienen los Training Data con sus respectivas etiquetas para posteriormente utilizarlos en el entrenamiento del clasificador.

```

1 // leer archivo train_ojo_izquierdo.xml
2   FileStorage fs;
3   fs.open("ojo_izquierdo.xml", FileStorage::READ);
4   Mat TrainingData_ojo_izq;
5   Mat Classes_ojo_izq;
6   fs["TrainingData"] >> TrainingData_ojo_izq;
7   fs["classes"] >> Classes_ojo_izq;
8 // leer archivo train_ojo_izquierdo.xml
9   fs.open("ojo_derecho.xml", FileStorage::READ);
10  Mat TrainingData_ojo_dere;
11  Mat Classes_ojo_dere;
12  fs["TrainingData"] >> TrainingData_ojo_dere;
13  fs["classes"] >> Classes_ojo_dere;

```

Algoritmo K-Nearest Neighbors

Después del pre-proceso y la segmentación realizada para detectar los ojos, ahora toca decidir si cada segmento son (o no son) ojos abiertos o cerrados . Para ello, se ha empleado el algoritmo K-Nearest Neighbors (K-nn). El algoritmo almacena en caché todas las muestras (ver figura 4.14) de entrenamiento y predice la respuesta para una nueva muestra mediante el análisis de un número determinado (K) de los vecinos más cercanos a la muestra utilizando la votación, el cálculo de la suma ponderada, y así sucesivamente. El método se refiere a veces como "aprender con el ejemplo o aprendizaje supervisado", ya que para la predicción busca el vector de características con una respuesta conocida que es más cercano al vector dado.

Aprendizaje supervisado, es un algoritmo de aprendizaje automático que aprende a través de la utilización de los datos de la etiqueta. Tenemos que entrenar el algoritmo con una cantidad de datos que tiene la etiqueta 1 para la clase ojos cerrados y -1 para la clase ojos abiertos. Para el entrenamiento del clasificador se realiza de la siguiente manera:

```

1 k=1; // k es el número de vecinos mas cercanos
2 CvKNearest knn_ojo_izq( TrainingData_ojo_izq, Classes_ojo_izq
   , cv::Mat(), false, k);
3 CvKNearest knn_ojo_dere( TrainingData_ojo_dere,
   Classes_ojo_dere, cv::Mat(), false, k);

```

El clasificador está listo para usarlo, se utiliza la función `find_nearest` para que devuelva el identificador de la clase: 1 si el ojo esta cerrado o -1 si el ojo esta abierto.

```
clasificacion_ojo_izq = knn_ojo_izq.find_nearest(
    recorte_ojo_izq, 1);
```

4.6.5. Orientación del rostro

Para estimar la orientación del rostro se ha tomado en cuenta la posición de los ojos para la cual se ha utilizado la siguiente ecuación

$$\theta = \tan^{-1} \left(\frac{y_2 - y_1}{x_2 - x_1} \right) \quad (6.1)$$

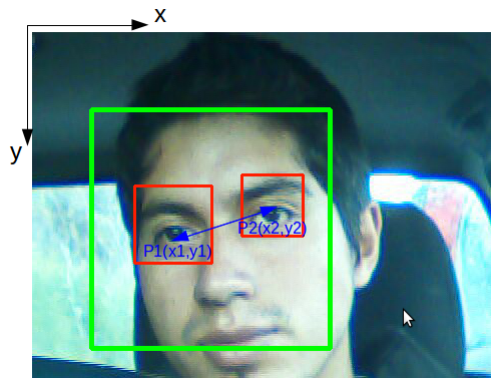


Fig. 4.15: Orientación del Rostro
Elaborado por: Investigador

De forma experimental se ha determinado los límites para la ecuación 6.1 que estiman la orientación del rostro

$$\begin{cases} izquierda & \theta > 15 \\ centro & |\theta| \leq 15 \\ derecha & \theta < -15 \end{cases} \quad (6.2)$$

4.6.6. Dirección del rostro

Al obtener la dirección del rostro se determina si el conductor está o no está distraído, es importante analizar si el conductor esta distraído, ya que es una de las causas

para generar muchos accidentes y ha contribuido de manera negativa en el aumento de las estadísticas de accidentes de tráfico.

El índice de distracción se calcula con la distancia que existe entre los ojos para el cual se utiliza la ecuación 4.3, una distancia muy corta entre los ojos indica que existe distracción.

$$d = \sqrt{(x1 - x2)^2 + (y1 - y2)^2} \quad (6.3)$$

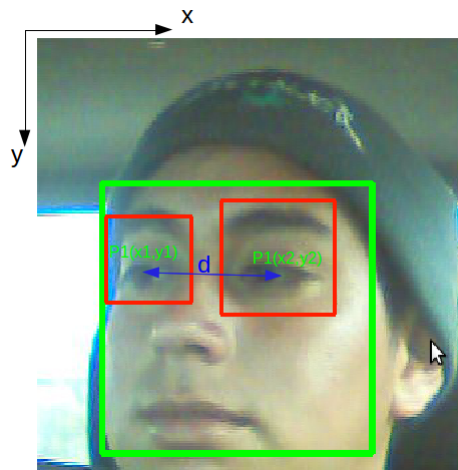


Fig. 4.16: Dirección del rostro
Elaborado por: Investigador

4.7. Instalación del sistema de control del estado de somnolencia en una camioneta Nissan doble cabina.

Esquema de conexión del sistema de control del estado de somnolencia en el vehículo

En la figura 4.17 se muestra el esquema de conexión del sistema dentro del vehículo

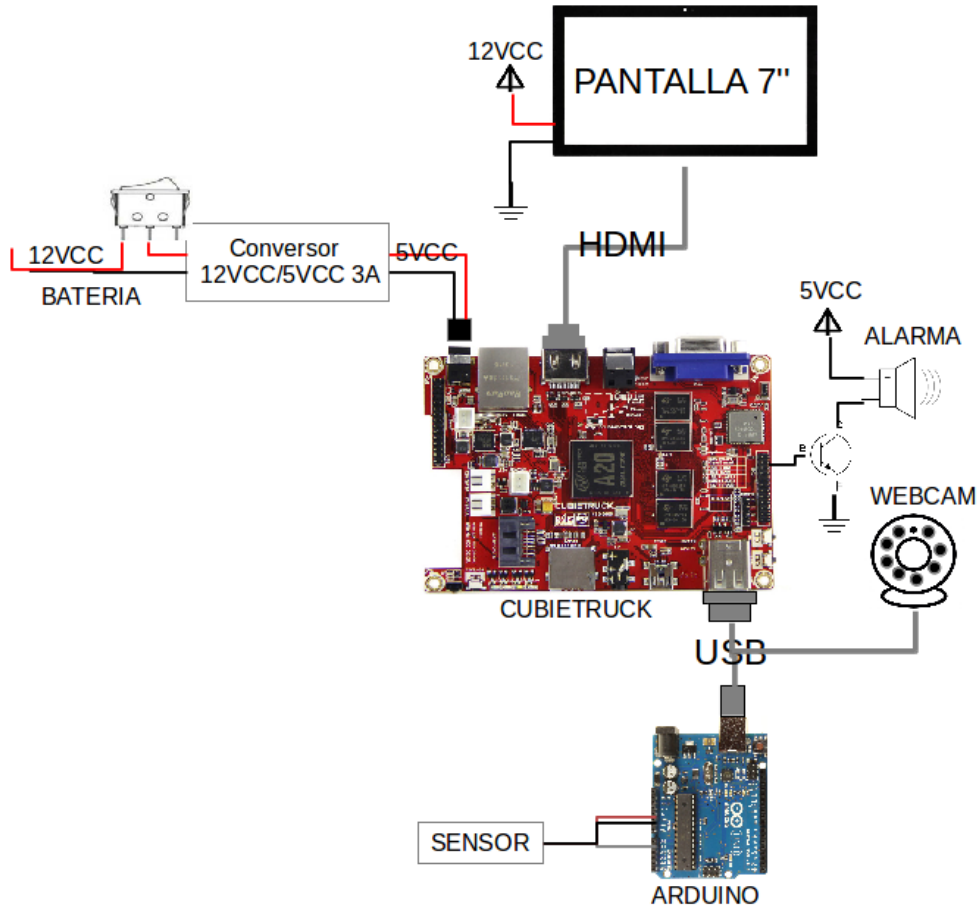


Fig. 4.17: Sistema de control del estado de somnolencia
Elaborado por: Investigador

La cámara web

La cámara se debe colocar sobre el tablero del vehículo tratando de en lo más posible enfocar el rostro frontalmente. En la figura 4.18 se ve como se a colocado la cámara web para nuestro prototipo.



Fig. 4.18: Ubicación de la cámara web sobre el tablero del vehículo
Elaborado por: Investigador

La pantalla

La ubicación de la pantalla de 7" va a depender del usuario del vehículo. Este caso se a colocado como se muestra en la figura 4.19.



Fig. 4.19: Ubicación de la pantalla en él tablero del vehículo
Elaborado por: Investigador

El zumbador

El zumbador se coloca en la parte inferior del apoya cabezas como se observa en la figura 4.20.



Fig. 4.20: Ubicación del zumbador en el apoya cabezas
Elaborado por: Investigador

La Cubietruck y el Arduino

La Cubietruck con el conversor de voltaje de 12Vcc a 5Vcc y el Arduino se han colocado dentro de una carcasa para protegerla de golpes y del usuario, la carcasa será colocada según el espacio que disponible el vehículo. En la figura 4.21 se muestra la colocación de nuestra carcasa.



Fig. 4.21: Ubicación de la carcasa que contiene la Cubietruck, convertidor de voltaje y el Arduino dentro del vehículo

Elaborado por: Investigador

Sensor de oximetría de pulso

El sensor de oxímetro de pulso se encuentra a la parte izquierda del volante para su colocación en el dedo de la mano izquierda como se puede ver en la figura 4.22.



Fig. 4.22: Colocación del sensor de oximetría de pulso en el dedo índice de la mano izquierda

Elaborado por: Investigador

4.8. Pruebas de funcionamiento

A continuación se detallan las pruebas realizadas en la carretera entre Ambato y Mocha.

Detección de rostro y ojos

En la figura 4.23 se muestra la detección del rostro y de los ojos durante el día y en la figura 4.24 la detección del rostro y ojos durante la noche.

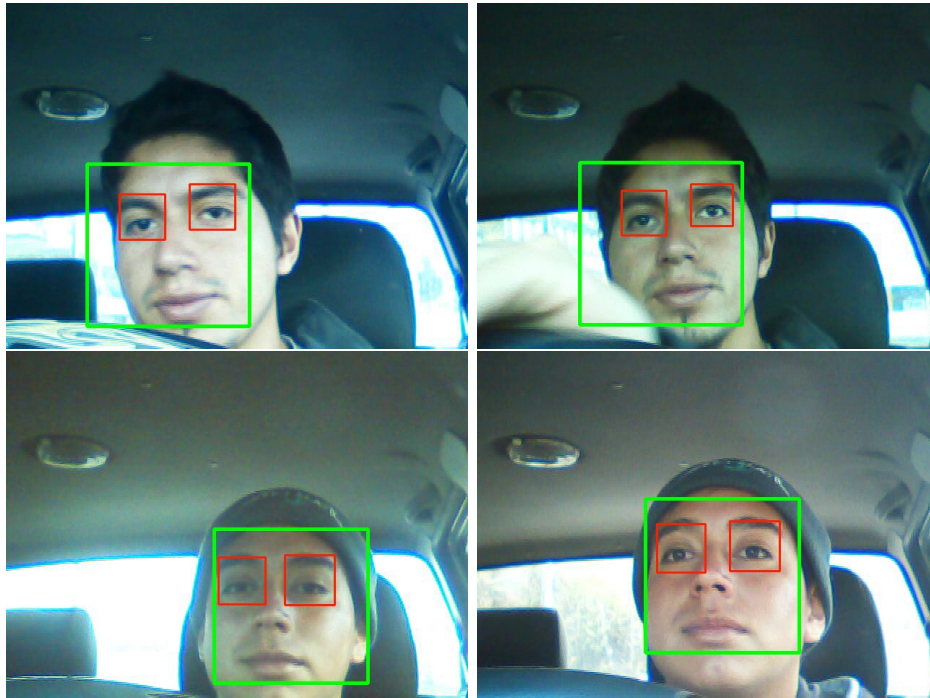


Fig. 4.23: Detección de rostro y ojos del conductor durante el día
Elaborado por: Investigador



Fig. 4.24: Detección de rostro y ojos del conductor durante la noche
Elaborado por: Investigador

Detección de ojos cerrados

En la figura 4.25 se muestra la detección de los ojos cerrados pero el tiempo que permanecen cerrados no sobrepasa el tiempo establecido para dar la alerta de somnolencia.

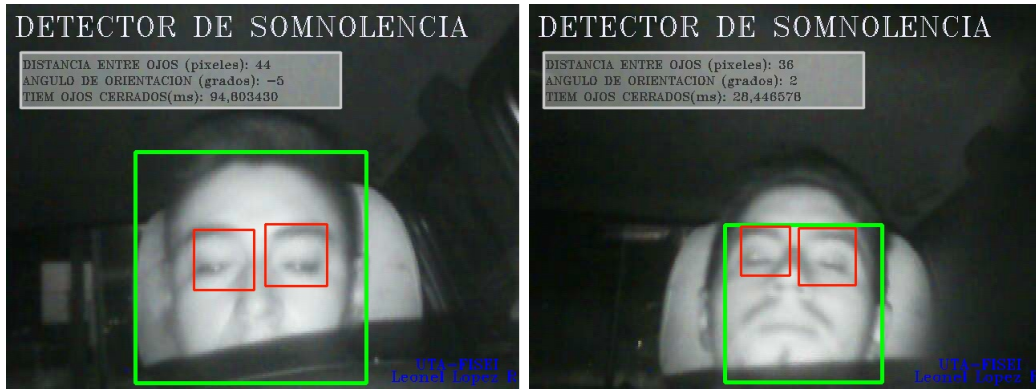


Fig. 4.25: Conductor con los ojos cerrados.
Elaborado por: Investigador

Detección de distracción

La distracción se da cuando, la distancia entre los ojos es menor a 20 pixeles durante un tiempo mayor al establecido para distracción. En las figuras 4.26 y 4.27 se puede ver la detección de distracción.



Fig. 4.26: Conductor distraído durante el día
Elaborado por: Investigador

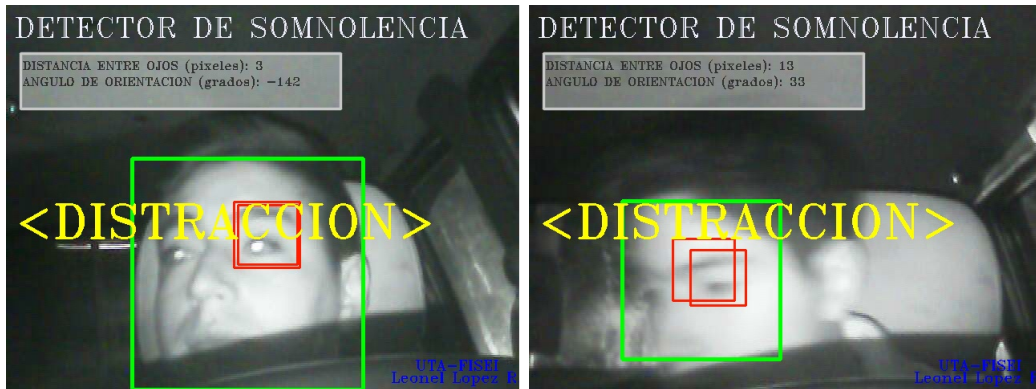


Fig. 4.27: Conductor distraído durante la noche
Fuente: El Autor

Detección de somnolencia

En las figuras 4.28 y 4.29 se ve que se a dado la alerta de somnolencia ya que el tiempo de ojos cerrados es igual o mayor al tiempo de somnolencia

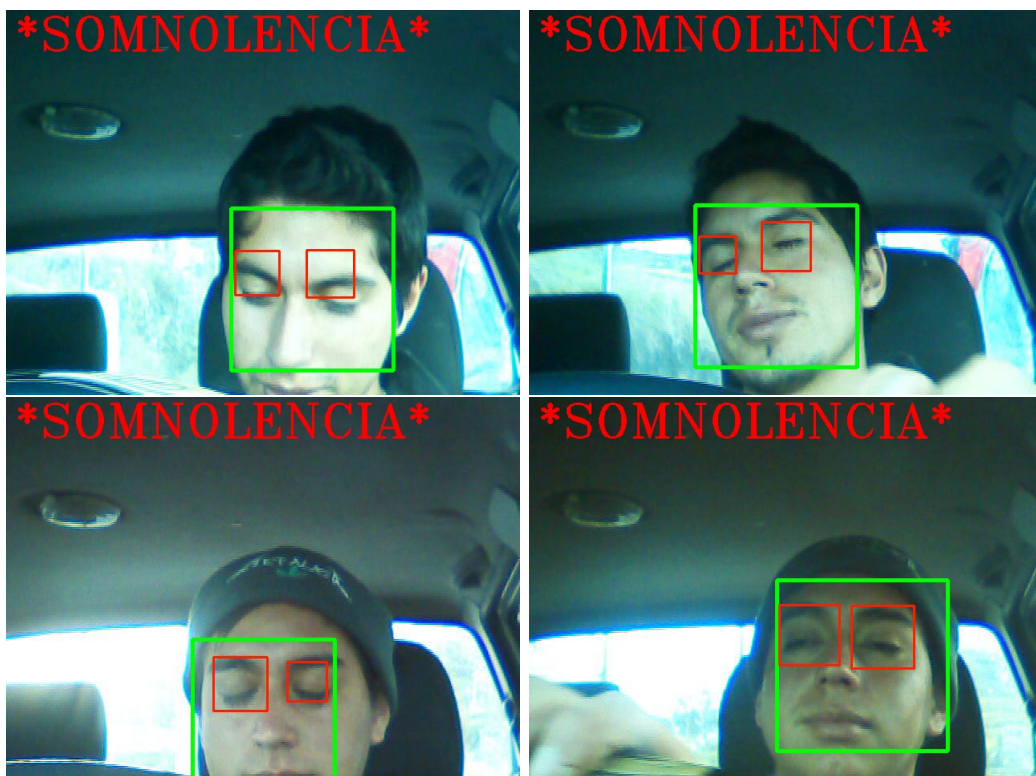


Fig. 4.28: Conductor con somnolencia durante día
Elaborado por: Investigador



Fig. 4.29: Conductor con somnolencia durante la noche
Elaborado por: Investigador

4.8.0.1. Falsa alarma

El sistema da una falsa alarma debido a las dificultades de la visión artificial que se nombraron en la sección 2.2.7.1 en este caso se presenta el movimiento rápido como se muestra la figura 4.30 que impide que se detecte los ojos correctamente dando a una clasificación incorrecta y por ende una falsa alarma, de igual manera sucede con los cambios bruscos de iluminación como se aprecia en la figura 4.31.

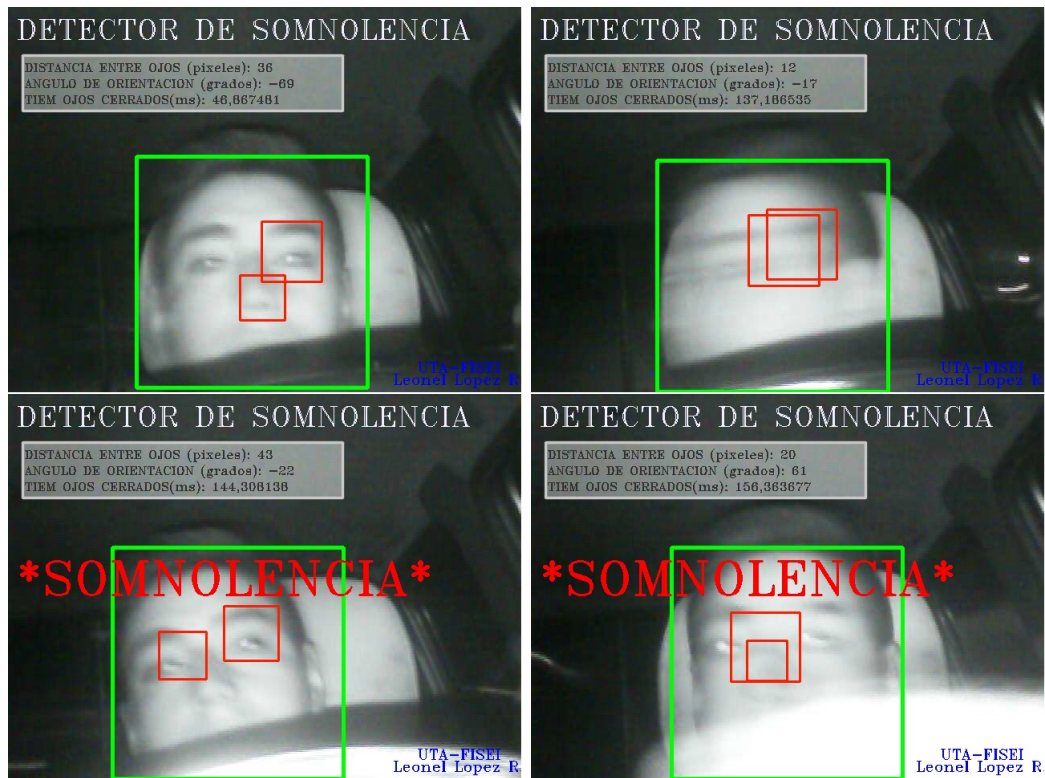


Fig. 4.30: Falsa alarma por presencia de movimiento rápido
Elaborado por: Investigador



Fig. 4.31: Cambios bruscos de iluminación
Elaborado por: Investigador

Cuando se tiene falsos positivos en la detección de los ojos se da una falsa alarma como se aprecia en la figura 4.32.

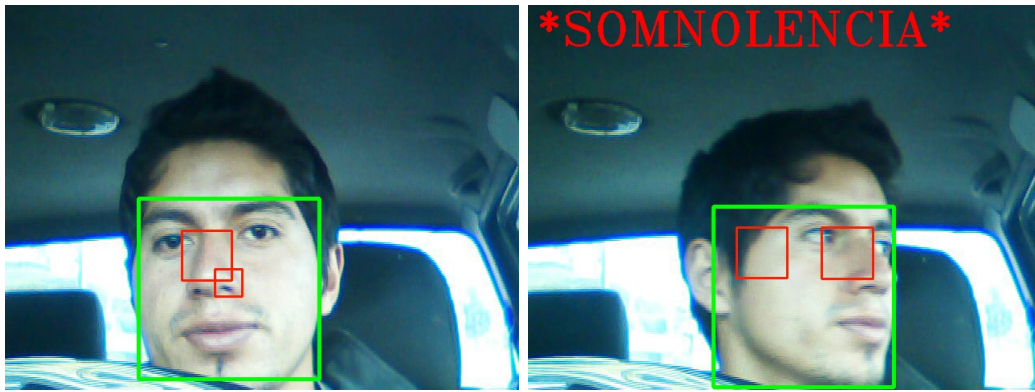


Fig. 4.32: Falsos positivos
Elaborado por: Investigador

4.8.1. Resultados del detector de somnolencia

Resultados de la oximetría de pulso

En la figura 4.33 se muestra el resultado de monitorear el ritmo cardiaco. Como se puede apreciar entre los minutos 11 se tiene una disminución del ritmo cardiaco a 67ppm por causas de somnolencia de igual manera se ve afectada la saturación de oxígeno como se muestra en la figura 4.34.

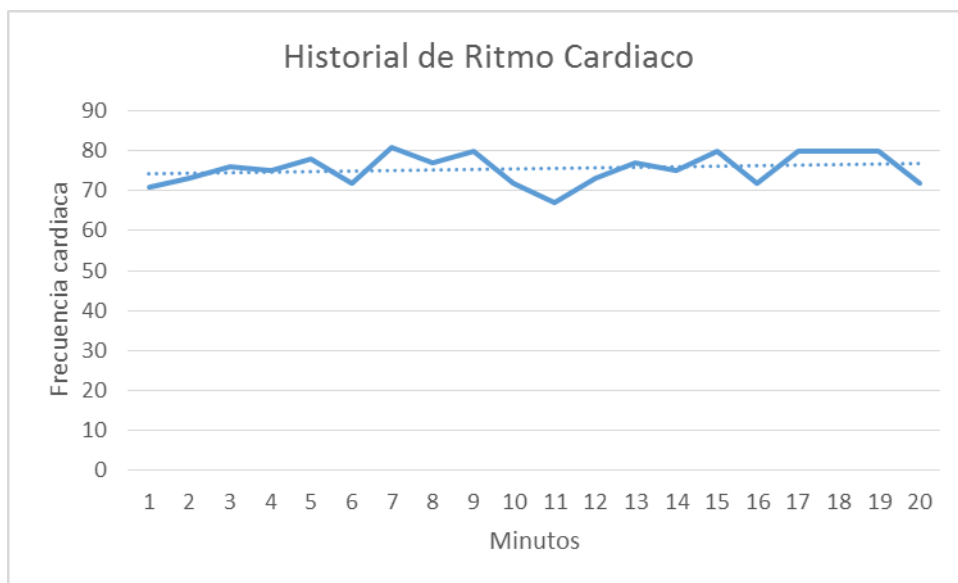


Fig. 4.33: Historial de la monitorización del ritmo cardiaco
Elaborado por: Investigador

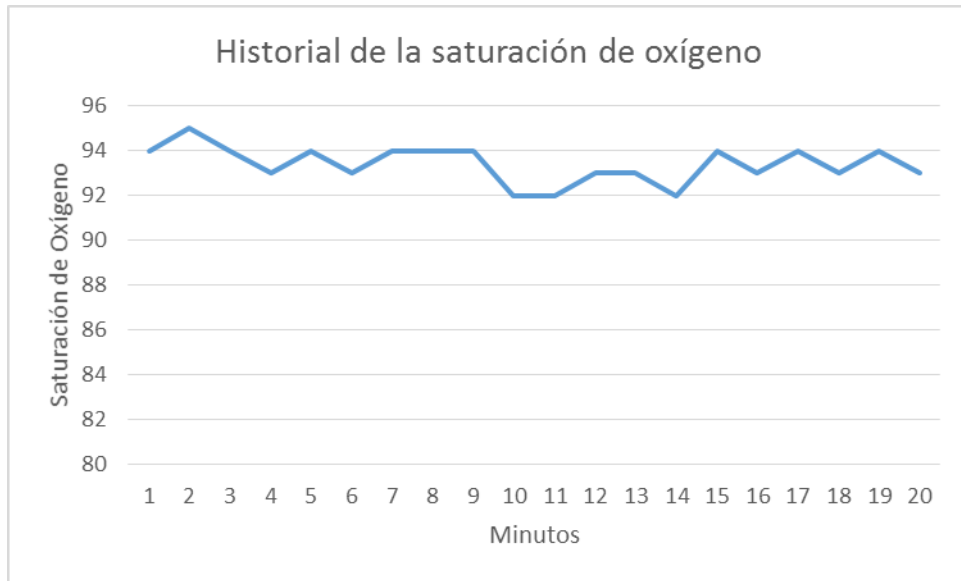


Fig. 4.34: Historial de la monitorización de la saturación de oxígeno
Elaborado por: Investigador

Resultados de la detección de rostro y ojos durante el día

En la siguiente tabla 4.3 se detalla tres pruebas (P1, P2, P3) realizadas durante el día con sus respectivos número de frames en las que se ha determinado el porcentaje de aciertos totales que ha tenido el seguimiento del rostro con 98.74%, la detección del ojo izquierdo un 98.08% y del ojo derecho un 98.03%.

Tabla 4.3: Resultados del sistema de control del estado de somnolencia durante el día

Pruebas	Nº frames	Seguimiento Rostro		Detección del ojo izquierdo		Detección del ojo derecho	
		Fallos	%Aciertos	Fallos	%Aciertos	Fallos	%Aciertos
P1	5186	72	98.61	101	98.05	104	97.99
P2	3657	39	98.93	68	98.14	70	98.08
TOTAL	8843	111	98.74	169	98.08	174	98.03

Elaborado por: Investigador

En la tabla 4.4 se tiene el número total de ojos detectados por el sistema de somnolencia, de los cuales se ha obtenido el porcentaje de aciertos durante la clasificación de los ojos cerrados un 98.28% y se ha detectado distracción en un 98.91%.

Tabla 4.4: Resultados de clasificación de ojos cerrados y la detección de distracción durante el día

Pruebas	Nº Ojos detectados	Ojos Cerrados		Distracción	
		Fallos	%Aciertos	Fallos	% Aciertos
P1	5083	91	98.20	58	98.85
P2	3588	58	98.38	36	98.99
TOTAL	8671	149	98.28	94	98.91

Elaborado por: Investigador

Resultados de la detección de rostro y ojos durante la noche

En la siguiente tabla 4.5 se detalla las pruebas (P1, P2, P3) realizadas durante la noche con sus respectivos número de frames en las que se ha determinado el porcentaje de aciertos totales que ha tenido el seguimiento del rostro con 98.75 %, la detección del ojo izquierdo un 96.76 % y del ojo derecho un 96.78 %.

Tabla 4.5: Resultados del sistema de control del estado de somnolencia durante la noche

Pruebas	Nº frames	Seguimiento Rostro		Detección del ojo izquierdo		Detección del ojo derecho	
		Fallos	%Aciertos	Fallos	%Aciertos	Fallos	%Aciertos
P1	1749	21	98.79	61	96.51	64	96.34
P2	952	8	99.05	35	94.53	33	96.53
P3	3800	51	98.65	114	97.34	112	97.05
TOTAL	6501	81	98.75	210	96.76	209	96.78

Elaborado por: Investigador

En la tabla 4.6 se tiene el número total de ojos detectados por el sistema de somnolencia, de los cuales se ha obtenido el porcentaje de aciertos durante la clasificación de los ojos cerrados un 98.41 % y se ha detectado distracción en un 99.03 %.

Tabla 4.6: Resultados de clasificación de ojos cerrados y la detección de distracción durante la noche

Pruebas	Nº Ojos detectados	Ojos Cerrados		Distracción	
		Fallos	%Aciertos	Fallos	% Aciertos
P1	1687	25	98.51	17	98.99
P2	918	18	98.03	9	99.01
P3	3687	67	98.18	35	99.05
TOTAL	6292	110	98.41	61	99.03

Elaborado por: Investigador

4.9. Costo del proyecto

El costo de los materiales del sistema de control del estado de somnolencia en conductores de vehículos se detalla en la tabla 4.7 el cual a sido financiado por el Autor.

Tabla 4.7: Costo de los materiales para el proyecto

Descripción	Cantidad	Valor U.	Valor Total
Cubietruck Kit con Cable Sata 2.0 y mini USB a OTG	1	\$230	\$230
Pantalla Tontec 7 Inches LCD Touch Screen TFT Monitor AT070TN90	1	\$160	\$160
Web Cam Omega 300k 1600x1200	1	\$15	\$15
Anillo de leds infrarrojo	1	\$10	\$10
Arduino Uno	1	\$27	\$27
Oxímetro	1	\$54	\$54
Cable HDMI 1m	1	\$6	\$6
Convertor 12VCC/5VCC 3A	1	\$16	\$16
1mx1m de acrílico	1	\$10	\$10
Cable número 20	8m	\$0.30	\$2.40
Transistor 3904	1	\$0.20	\$0.20
Conector de Alimentación	2	\$0.40	\$0.40
Zumbador	1	\$0.30	\$0.30
Fusibles	3	\$0.50	\$1.50
Tornillos	25	\$0.05	\$1.25
		Total	\$534.05

Elaborado por: Investigador

Debido a que el sistema de control del estado de somnolencia en conductores de vehículos se a utilizado software libre para su desarrollo de igual manera el sistema se encuentra instalado en el sistema operativo Ubuntu el que no tiene ningún costo al momento de utilizarlo, Por lo tanto el sistema al momento de comercializar solo tendrá un costo adicional en la instalación.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- La depresión, la ansiedad, el estrés, el aburrimiento, jornadas prolongadas de trabajo sin descanso, uso de medicación como sedantes, ingerir alcohol causan parpadeo constante, pesadez de la cabeza, y reducen gradualmente la concentración del conductor haciendo que pierda su capacidad de respuesta al conducir por lo tanto el vehículo se sale repentinamente del trayecto o invade carril contrario sin dejar huellas de haber pretendido una maniobra para evitar el accidente ocasionando heridos leves o graves, fallecimientos y pérdidas materiales, aportando al menos en un 20 % de los accidentes de tránsito.
- El ritmo cardiaco y la saturación de oxígeno en la sangre son diferentes en cada persona, por lo que se monitorizó por un lapso de 3min al momento de colocar el sensor en el dedo para obtener un promedio como referencia del conductor en estado de vigilia, también, se generó las muestras de aprendizaje para la clasificación de ojos abiertos y cerrados pero no son lineales ni siguen una curva cónica por lo tanto el algoritmo SVM no es útil en nuestro caso por lo que se utiliza el algoritmo de K-NN, el tiempo de ejecución del programa en la Cubietruck tiene un promedio de 260ms y en un computador de escritorio con un procesador Intel Core Duo @ 2.48GHz el tiempo de ejecución es de 27ms.
- Se tiene una mejor detección de ojos abiertos u ojos cerrados, cuando se agregan imágenes con diferentes escalas y ángulos a las muestras de aprendizaje del conductor que está haciendo uso del sistema o de los que van hacer uso del

sistema de control del estado de somnolencia, pero a medida que se agregan más imágenes a la muestra de aprendizaje se hace demasiado grande y por ende aumenta el tiempo de ejecución.

5.2. Recomendaciones

- Antes de conducir evite ingerir comida en abundancia, la ingesta de licor, y el uso de sustancias psicotrópicas o estupefacientes, mantenga una buena ventilación en el vehículo, evite conducir por horas prolongadas, si siente sueño, cansancio físico o mental tomar descansos, y lo más importante Tomar conciencia y responsabilidad al momento de conducir.
- La tecnología cada día va evolucionando y se siguen mejorando las computadoras de una sola placa (SBC) así se tiene una nueva versión la Cubietruck que es la Cubieboard 4 con un procesador Allwinner A80 Octa-Core @ 2GHz la cual puede ser utilizada para mejorar el tiempo de computo de este proyecto o de otros proyectos que deseen realizar con placas SBC.
- Diseñar un algoritmo que permita crear de forma automática e instantánea una base de datos de rostro, ojos abiertos y ojos cerrados en diferentes ángulos y escalas del conductor al momento de colocarse frente al volante para crear muestras de aprendizaje de solo un individuo, mejorando la confiabilidad y la eficiencia del sistema. Por lo que se recomienda a la comunidad estudiantil, que desee involucrarse en la visión artificial, procesamiento de señales, y aplicaciones en tiempo real, OpenCV es una buena alternativa por tener un alto rendimiento y compatibilidad con Windows, Mac, Linux y Android, y con lenguajes de programación C/C++, Java y Python.

Referencias Bibliográficas

- [1] M. E. R. Jorge Rey de Castro Mujica, Edmundo Rosales Mayor, “Somnolencia y cansancio durante la conducción: accidentes de tránsito en las carreteras del Perú,” Rev Perú Med Exp Salud Publica. 2010.
- [2] I. F. de Neurología y Sueño (IFN), “El sueño y los accidentes de tránsito,” Fundación Argentina de Estudio del Cerebro. 2012.
- [3] N. T. S. A. N. E. P. National Center for Sleep Disorders Research, “Drowsy driving and automobile crashes,” Washington DC: National Highway Safety Association 1998.
- [4] E. Times. (2013, noviembre) Ecuador registra altos índices de accidentes de tránsito. [Online]. Available: <http://www.ecuadortimes.net/es/2013/11/04/ecuador-registra-altos-indices-de-accidentes-de-transito/>
- [5] P. N. del Ecuador SIAT. (2010) Informe sobre accidentabilidad de la vía a loag-santo domingo de los años 2000 al 2009. [Online]. Available: <http://www.pdfio.com/k-7676520.html#>
- [6] T. auto seguro. (2012, Mayo) Nuevo volante detecta fatiga y somnolencia del conductor. [Online]. Available: <http://www.tuautoseguro.cl/conduccion/nuevo-volante-detecta-fatiga-y-somnolencia-del-conductor/>
- [7] M. V. Avila Jiménez and J. A. Castillo Corredor, “Diseño e implementación de un sistema de detección del estado del sueño del conductor de un vehículo,” Ph.D. dissertation, BOGOTÁ D.C.-COLOMBIA/UNIVERSIDAD DE SAN BUENAVENTURA, 2013.
- [8] E. de la Arturo y Armingol José., “Sistema avanzado de asistencia a la conducción para la detección de la somnolencia y la distracción durante la

- noche,” Ph.D. dissertation, Leganés (Madrid), España /Universidad Carlos III de Madrid.
- [9] F. S. Córdova Ochoa and P. A. Benavides Peralta, “Diseño y construcción de un sistema de alarma y frenado automático para un vehículo al detectar conductores somnolientos.” Ph.D. dissertation, LATACUNGA/ESPE/2012, 2012.
- [10] R. B. L. M. B. E. L. M. O. J. L. García, E. Rogado and D. Schleicher, “Sistema detector de fatiga en la conducción,” *Departamento de Electrónica. Universidad de Alcalá*. [Online]. Available: <http://tv.uvigo.es/uploads/material/Video/2664/P09.pdf>
- [11] *Psicoanálisis: El Sueño Como Estado Fisiológico y el Posible Lenguaje de Éstos*, Neurofeedback Std. [Online]. Available: <http://www.neurobiofeedback-neuroterapia.com/item/psicoanalisis-el-sueno-como-estado-fisiologico-y-el-posible-lenguaje-de-estos.html>
- [12] *¿Por qué es importante dormir bien?*, Buena Salud Std. [Online]. Available: <http://www.buenasalud.net/2012/07/04/por-que-es-importante-dormir-bien.html>
- [13] Lundbeck. (2013, Febrero) Somnolencia excesiva. [Online]. Available: <http://www.lundbeck.com/mx/pacientes/trastornos-del-sueo/sindrome-de-sueo-excesivo-diurno-sed/somnolencia-excesiva>
- [14] D. R. I. A. Navarrete, “Cambios fisiológicos en el sueño,” *Revista Ecuatoriana de Neurología*, vol. 22, pp. 1–3, 2013. [Online]. Available: http://www.medicosecuador.com/revecuatneurol/vol22_n1-3_2013/9-Cambios.pdf
- [15] V. H. Nieuwenhuys, Voogd, *El sistema Nervioso Central.*, E. M. Panamericana, Ed. 4^a ed. Tomo I., 2009.
- [16] L. C. Meraz. (2013, Diciembre) Sistema nervioso periférico. Salud. [Online]. Available: <http://www2.esmas.com/salud/668346/sistema-nervioso-periferico-snp-que-conforma/>
- [17] D. Puente., “Cambios fisiológicos durante el sueño,” *Neurowikia*, Tech. Rep., 2013.
- [18] S. S. e. I. Ministerio de Sanidad. Actividad física y salud en la infancia y la adolescencia, definiciones. msssi. [Online]. Available: http://www.msssi.gob.es/ciudadanos/proteccionSalud/adultos/actiFisica/docs/definiciones_Es.pdf
- [19] H. S. Calhoun DA, “Sleep and hypertension,” pp. 434–43, *Chest*. 2010 Aug.

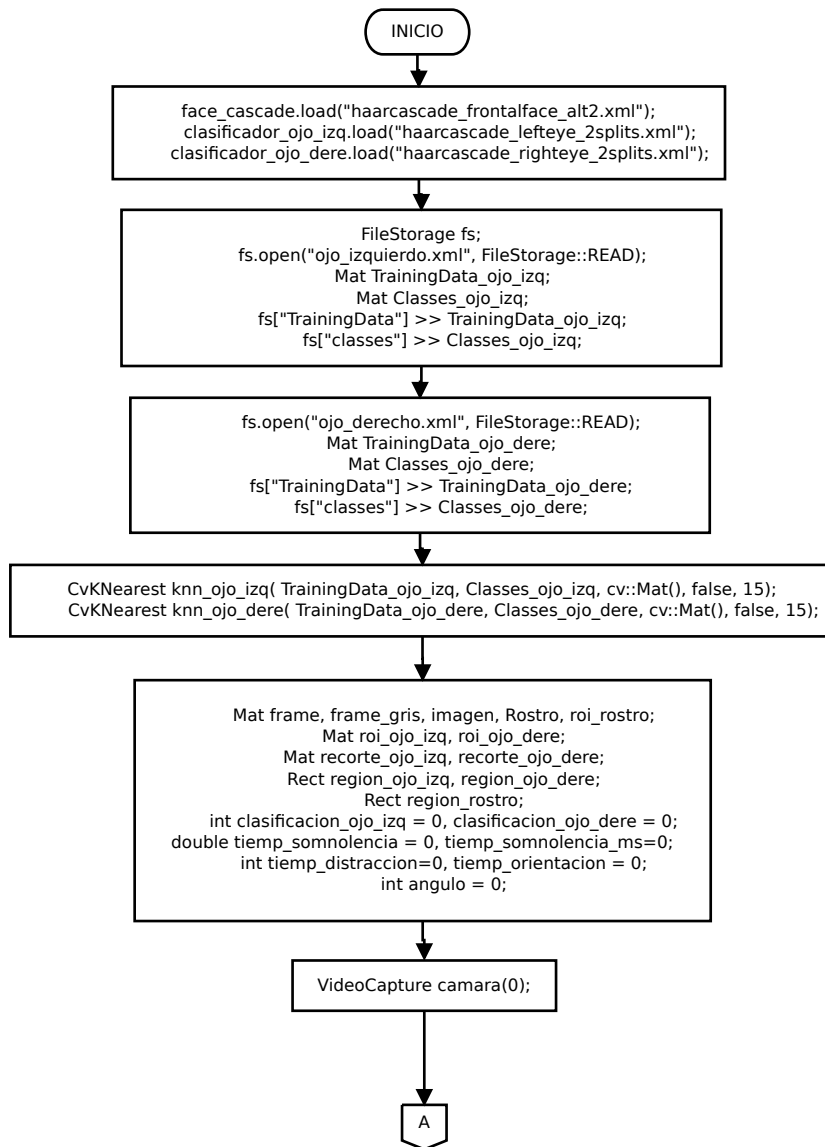
- [20] Y. D. A. B. Zvi Shinar, Solange Akselrod, “Autonomic changes during wake-sleep transition: A heart rate variability based approach,” *Autonomic Neuroscience*, vol. 130, pp. 17–27, December, 2006.
- [21] D. A. M.H. Bonnet, “Heart rate variability: sleep stage, time of night, and arousal influences,” *Electroencephalography and Clinical Neurophysiology*, vol. 102, pp. 390–396, 1997.
- [22] L. F. A. VALENCIA, “Diseño y construcción de un oxímetro de pulso mediante la utilización de una plataforma de automatización momentum,” Ph.D. dissertation, ESCUELA POLITÉCNICA DEL EJÉRCITO, INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL, 2007.
- [23] S. Lopez, *Pulse Oximeter Fundamentals and Design*, F. Semiconductor, Ed. Document Number:AN4327, 2012.
- [24] M. Cazorla, “Robótica y visión artificial,” Grupo de Visión Robótica / Universidad de Alicante, Tech. Rep.
- [25] D. N. L. F. García, “Introducción a la visión artificial,” Departamento de Informática y Análisis Numérico, Escuela Politécnica Superior Universidad de Córdoba, Tech. Rep.
- [26] J. F. V. S. A. B. M. D. Ángel Sánchez Calle José Luis Esteban Sánchez-Marín, *Visión por Computación*, 2003.
- [27] E. C. Alicia S. Clark, *Diccionario Inglés a Español de Computación e Internet.*, Universal-Publishers., Ed. 1581124996, 9781581124996, 2004.
- [28] I. del Grupo de Investigación EDMANS, *Técnicas y Algoritmos Básicos de Visión Artificial*, S. d. P. Universidad de La Rioja, Ed. ISBN 84-689-9345-X, 2006.
- [29] E. A. S. MALPARTIDA, “Sistema de visión artificial para el reconocimiento y manipulación de objetos utilizando un brazo robot,” Master’s thesis, Pontificia Universidad Católica del Perú, 2003.
- [30] F. J. G. Fernández, “Reconocimiento de objetos en una cocina con una webcam,” Master’s thesis, Universidad Carlos III de Madrid, 2009.
- [31] Z. L. S. Niño, “Caracterización y clasificación de café cereza usando visión artificial,” Master’s thesis, UNIVERSIDAD NACIONAL DE COLOMBIA, 2005.
- [32] A. R. García, “Sistemas de percepción y visión por computador,” UNIVERSIDAD DE MURCIA, Tech. Rep., 2015.

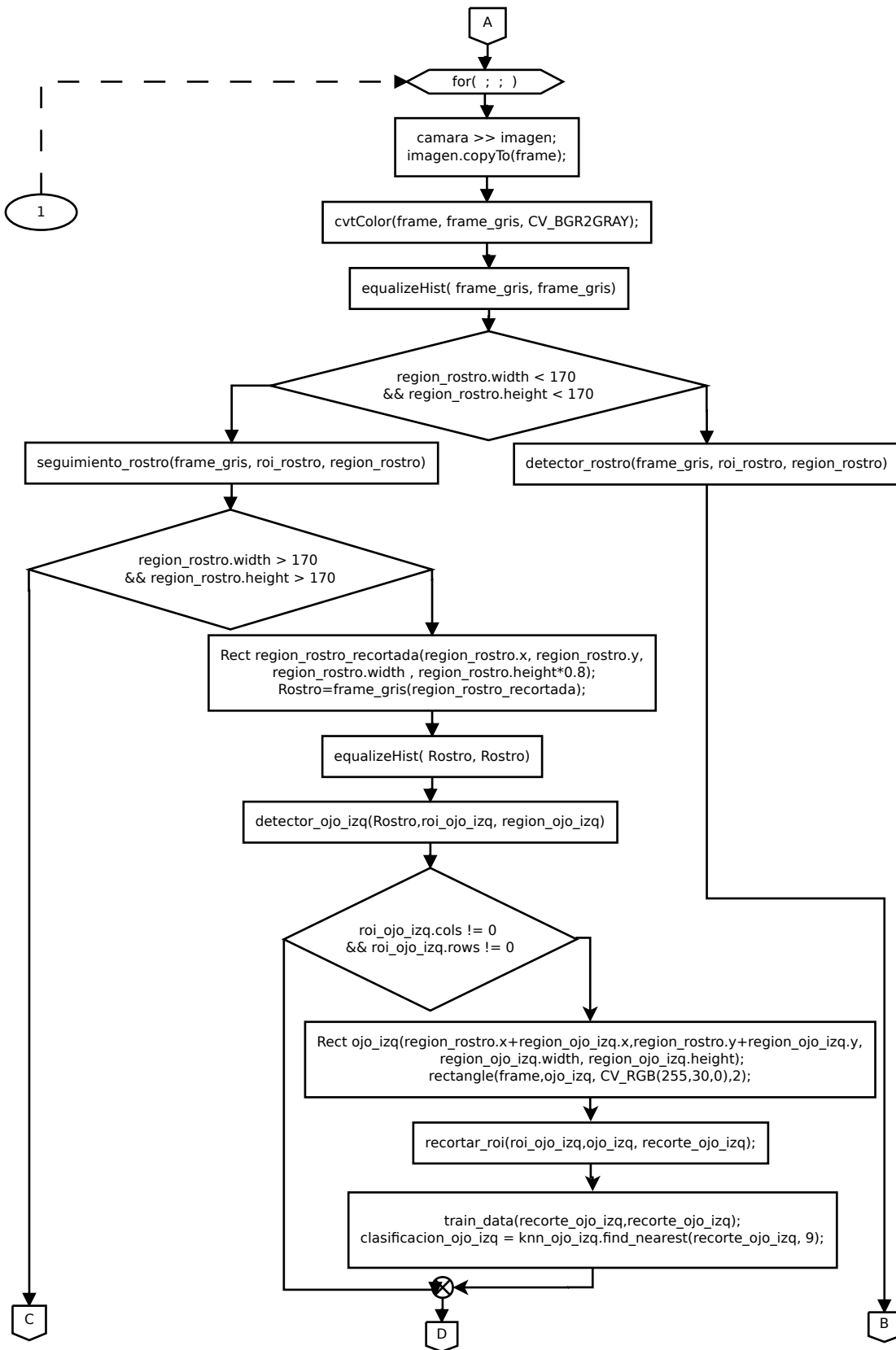
- [33] G. A. BETANCOURT, “Las máquinas de soporte vectorial (svms),” Universidad Tecnológica de Pereira, Tech. Rep., 2005.
- [34] M. C. Montes, “K-nearest neighbors gráficas, estadística y minería de datos con python,” *Centro de Investigaciones Energéticas Medioambientales y Tecnológicas, Madrid, Spain*, 2015.
- [35] J. P. Lerma, “Implementación del algoritmo de detección facial de viola-jones,” 2009.
- [36] M. D. Rodríguez, “Extracción automática de caras en imágenes captadas con móviles android,” Master’s thesis, UPC, 2012.
- [37] M. L. GUEVARA, “Detección de rostros en imágenes digitales usando clasificadores en cascada,” Master’s thesis, Universidad Tecnológica de Pereira, 2008.
- [38] OpenCV. (2015) Opencv 2.4.11.0 documentation » opencv tutorials » imgproc module. image processing. docs.opencv. [Online]. Available: http://docs.opencv.org/doc/tutorials/imgproc/histograms/template_matching/template_matching.html
- [39] *Factores físicos que afectan al conductor*, Comisariado Europeo del Automóvil Std. [Online]. Available: <http://www.seguridad-vial.net/conduccion/seguridad-en-la-circulacion/39-factores-que-afectan-al-conductor>
- [40] D. Bolaños, “Electronica, recopilación sensores opticos,” Tech. Rep. [Online]. Available: <http://www.bolanosdj.com.ar/TEORIA/SENSORESOPVICOS.PDF>
- [41] *TSL235R LIGHT-TO-FREQUENCY CONVERTER, Datasheet*.
- [42] *TSL257 HIGH-SENSITIVITY LIGHT-TO-VOLTAGE CONVERTER, Datasheet*.

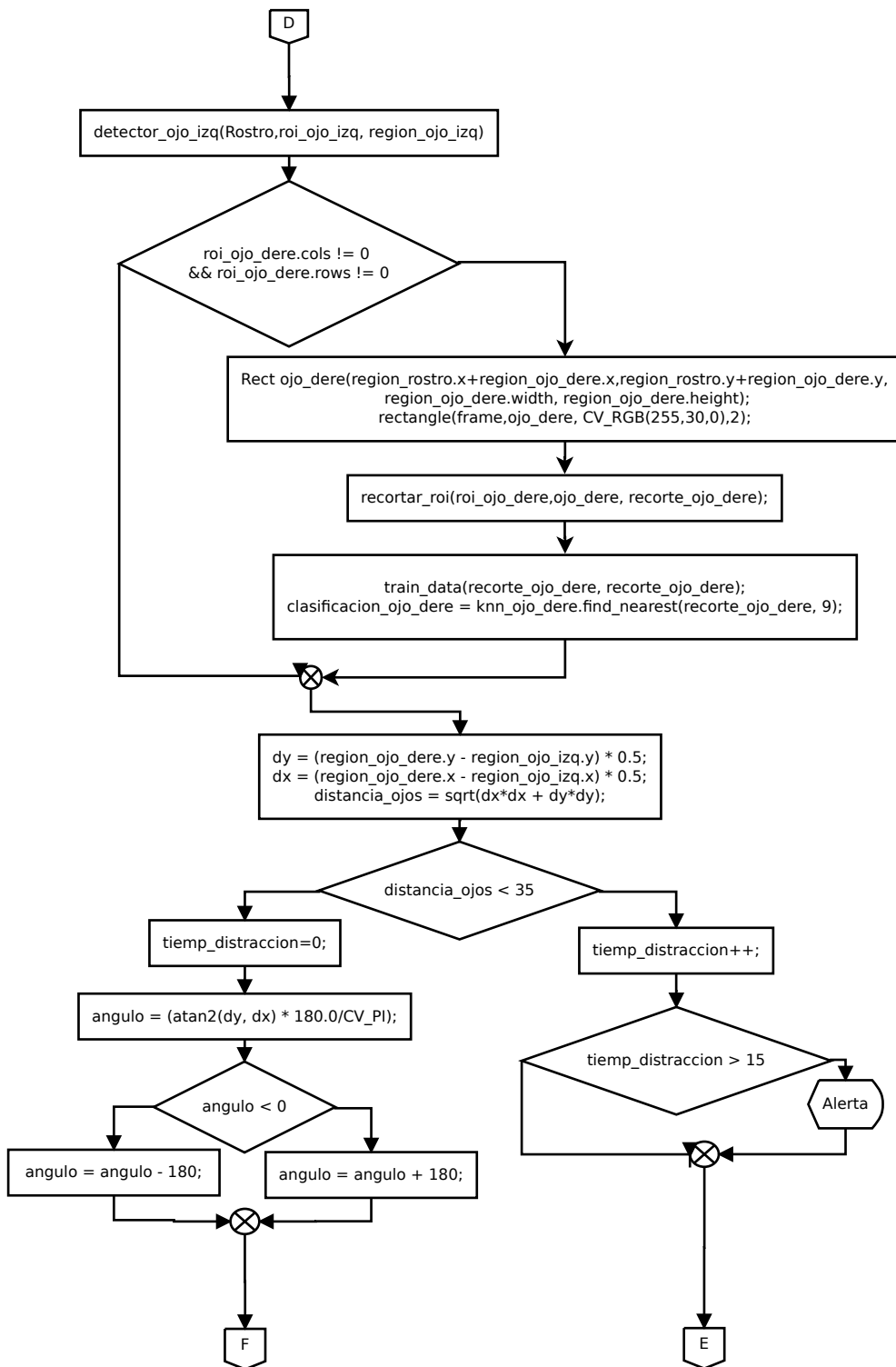
Anexos

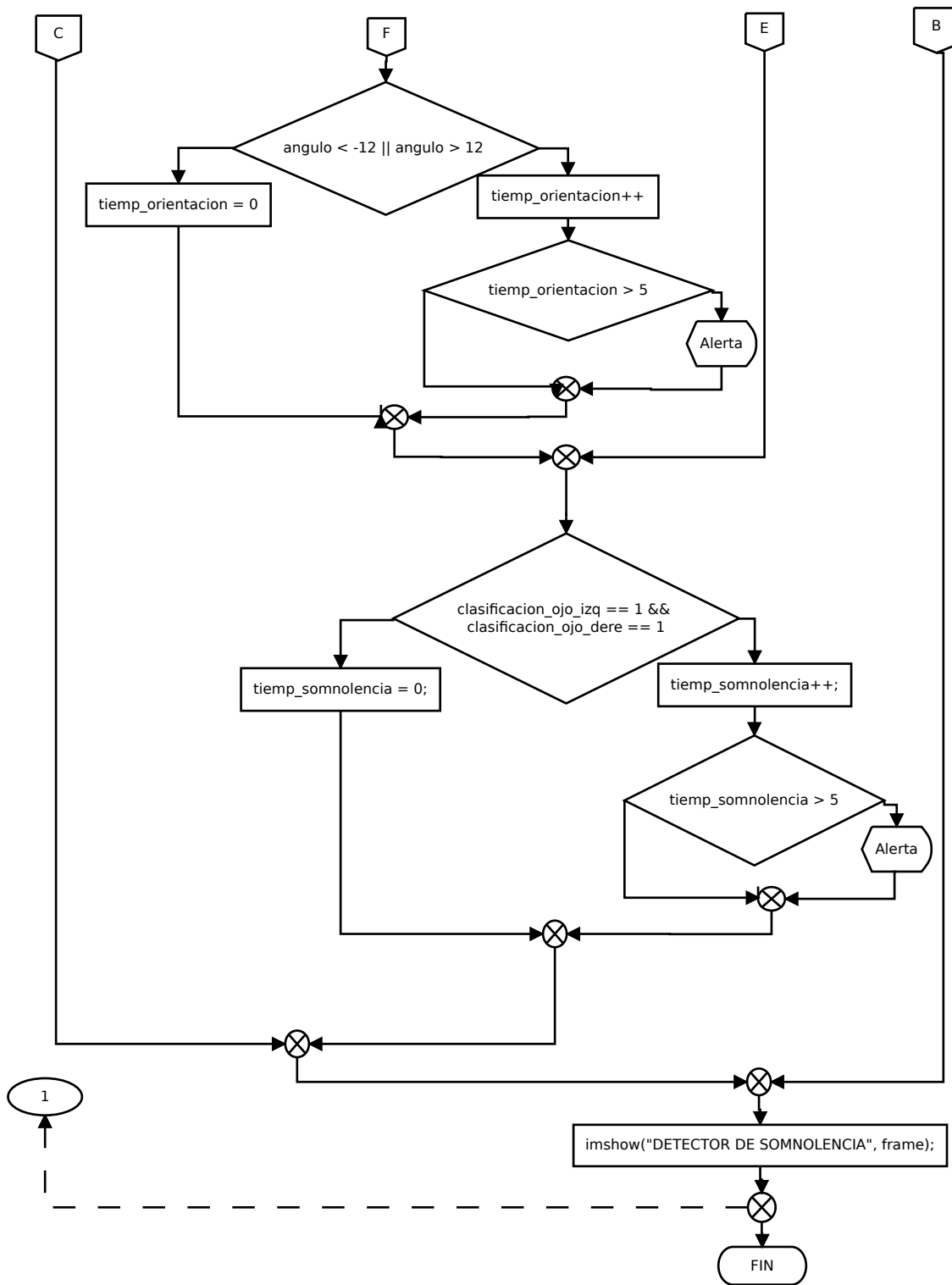
Anexo A

Diagrama de flujo del sistema de control del estado de somnolencia en conductores de vehículos









Anexo B

Código del sistema de control del estado de somnolencia en C++ con OpenCV

```
1 #include <opencv2/imgproc/imgproc.hpp>
2 #include <opencv2/highgui/highgui.hpp>
3 #include <opencv2/objdetect/objdetect.hpp>
4 #include <opencv2/ml/ml.hpp>
5 #include <opencv2/core/core.hpp>
6
7 using namespace std;
8 using namespace cv;
9
10 CascadeClassifier face_cascade;
11 CascadeClassifier clasificador_ojo_izq, clasificador_ojo_dere
    ;
12 /*****
13 *          FUNCION DETECTAR ROSTRO          *
14 *****/
15 int detector_rostro(Mat& video_gris, Mat& rostro_encontrado,
    Rect& roi_rostro)
16 {
17     vector<Rect> rostro, ojo;
18     face_cascade.detectMultiScale(video_gris, rostro, 1.2, 1,
    0|CASCADE_SCALE_IMAGE, Size(90,90));
19
20     for (size_t i = 0; i < rostro.size(); i++) {
21
```

```

22     rostro_encontrado = video_gris(rostro[i]);
23     roi_rostro = rostro[i];
24 }
25 return rostro.size();
26 }
27 /*****
28 *           FUNCION SEGUIMIENTO ROSTRO           *
29 *****/
30 void seguimiento_rostro(cv::Mat& frame_gris, cv::Mat&
31     rostro_encontrado, cv::Rect& region_rostro)
32 {
33     Size dimension_rostro(region_rostro.width, region_rostro.
34         height);
35     Rect region_seguimiento_rostro(region_rostro +
36         dimension_rostro - Point(dimension_rostro.width/4,
37         dimension_rostro.height/2));
38     region_seguimiento_rostro &= Rect(0, 0, frame_gris.cols,
39         frame_gris.rows);
40     // se crea la matriz de resultados que almacenará los
41     coincidencia para cada ubicación plantilla.
42     Mat resultado(region_seguimiento_rostro.width -
43         rostro_encontrado.rows + 1, region_seguimiento_rostro.
44         height - rostro_encontrado.cols + 1, CV_32FC1);
45     // Realiza la operación de comparación de plantillas:
46     matchTemplate(frame_gris(region_seguimiento_rostro),
47         rostro_encontrado, resultado, CV_TM_SQDIFF_NORMED);
48     // Localizar los valores máximo y mínimo en la matriz de
49     resultado utilizando minMaxLoc.
50     double minval, maxval;
51     Point minloc, maxloc;
52     minMaxLoc(resultado, &minval, &maxval, &minloc, &maxloc);
53     // Para el método CV_SQDIFF_NORMED el mejor de los
54     partidos son los valores más bajos.
55     if (minval <= 0.2) {
56         region_rostro.x = region_seguimiento_rostro.x + minloc.
57             x;
58         region_rostro.y = region_seguimiento_rostro.y + minloc.
59             y;
60     } else

```

```

48     region_rostro.x = region_rostro.y = region_rostro.width
        = region_rostro.height = 0;
49 }
50 /*****
51 *     FUNCION DETECTAR OJO IZQUIERDO     *
52 *****/
53 void detector_ojo_izq(Mat& rostro, Mat& ojo_izq_encontrado,
    Rect& roi_ojo_izq)
54 {
55     vector<Rect> ojo_izq;
56     clasificador_ojo_izq.detectMultiScale(rostro, ojo_izq,
        1.2, 2, CASCADE_FIND_BIGGEST_OBJECT, Size(35,35));
57     if(ojo_izq.size()){
58         roi_ojo_izq = ojo_izq[0];
59         ojo_izq_encontrado = rostro(roi_ojo_izq);
60     }
61 }
62 /*****
63 *     FUNCION DETECTAR OJO DERECHO     *
64 *****/
65 void detector_ojo_dere(Mat& rostro, Mat& ojo_dere_encontrado,
    Rect& roi_ojo_dere)
66 {
67     vector<Rect> ojo_dere;
68     clasificador_ojo_dere.detectMultiScale(rostro, ojo_dere,
        1.2, 2, CASCADE_FIND_BIGGEST_OBJECT, Size(35,35));
69     if(ojo_dere.size()) {
70         roi_ojo_dere = ojo_dere[0];
71         ojo_dere_encontrado = rostro(roi_ojo_dere);
72     }
73 }
74
75 void train_data(Mat& recorte_ojo, Mat& train_test) {
76     resize(recorte_ojo, recorte_ojo, Size(24, 24), 1.0, 1.0,
        INTER_CUBIC);
77     recorte_ojo= recorte_ojo.clone().reshape(1, 1);
78     train_test.convertTo(recorte_ojo, CV_32FC1);
79 }
80 void recortar_roi(Mat& roi, Rect& region, Mat& recorte) {

```

```

81     Rect recortar_roi(region.width*0.2, region.height*0.3,
82         region.width-region.width*0.2, region.height-region.
83         height*0.3);
84     recorte=roi(recortar_roi);
85 }
86 void orientacion(double& dx, double& dy, int&
87     tiemp_orientacion, int& angulo){
88     angulo = (atan2(dy, dx) * 180.0/CV_PI);
89     if (angulo < 0)
90         angulo = angulo + 180;
91     else
92         angulo = angulo - 180;
93     if((angulo > -90 && angulo < -12) || (angulo > 12 &&
94         angulo < 90)) {
95         tiemp_orientacion++;
96     }
97     else
98         tiemp_orientacion = 0;
99 }
100 /*****
101 *   PROGRAMA PRINCIPAL DETECTOR DE SOMNOLENCIA   *
102 *****/
103 int main()
104 {
105     // cargar los clasificadores cascada
106     face_cascade.load("haarcascade_frontalface_alt2.xml");
107     clasificador_ojo_izq.load("haarcascade_lefteye_2splits.xml
108         ");
109     clasificador_ojo_dere.load("haarcascade_righteye_2splits.
110         xml");
111     // leer archivo train_ojo_izquierdo.xml
112     FileStorage fs;
113     fs.open("SVM_ojo_izquierdo.xml", FileStorage::READ);
114     Mat TrainingData_ojo_izq;
115     Mat Classes_ojo_izq;
116     fs["TrainingData"] >> TrainingData_ojo_izq;
117     fs["classes"] >> Classes_ojo_izq;
118     // leer archivo train_ojo_izquierdo.xml
119     fs.open("SVM_ojo_derecho.xml", FileStorage::READ);
120     Mat TrainingData_ojo_dere;

```

```

115 Mat Classes_ojo_dere;
116 fs["TrainingData"] >> TrainingData_ojo_dere;
117 fs["classes"] >> Classes_ojo_dere;
118 CvKNearest knn_ojo_izq( TrainingData_ojo_izq,
    Classes_ojo_izq, cv::Mat(), false, 1);
119 CvKNearest knn_ojo_dere( TrainingData_ojo_dere,
    Classes_ojo_dere, cv::Mat(), false, 1);
120 imshow("datos izq",datos_izq);
121 // matriz para las regiones de interes a analizar
122 Mat frame, imagen, roi_rostro;
123 Mat roi_ojo_izq, roi_ojo_dere;
124 Mat recorte_ojo_izq, recorte_ojo_dere;
125 Rect region_ojo_izq, region_ojo_dere;
126 Rect region_rostro;
127 int clasificacion_ojo_izq = 0, clasificacion_ojo_dere = 0;
128 double tiemp_somnolencia = 0; // tiemp_somnolencia_ms=0;
129 int tiemp_distraccion=0, tiemp_orientacion = 0;
130 int angulo = 0;
131 string box_text;
132
133 // abrir puerto de la webcam
134 //VideoCapture camara("/media/LeoneL/UTA/x/opencv/3.avi");
135 VideoCapture camara(0);
136 camara.set(CV_CAP_PROP_FRAME_WIDTH, 320);
137 camara.set(CV_CAP_PROP_FRAME_HEIGHT, 240);
138 //camara.set(CV_CAP_PROP_FPS, 01);
139 namedWindow("DETECTOR DE SOMNOLENCIA", CV_WINDOW_NORMAL);
140 /*
    *****
141 * bucle infinito para la captura de video y su
    procesamiento *
142 *****
    */
143 int i = 0, d = 0;
144 for(;;) {
145 // iniciar tiempo de conteo
146 double tiemp = 0;
147 tiemp=(double)getTickCount();
148 double tiemp_program = 0;

```

```

149     tiemp_program=(double)getTickCount();
150 // conseguir un nuevo frame de la webcam
151     camara >> imagen;
152     //resize(imagen, imagen, Size(280, 300), INTER_AREA);
153     imagen.copyTo(frame);
154 // Convertir a escala de gris
155     Mat frame_gris;
156     cvtColor(frame, frame_gris, CV_BGR2GRAY);
157 // ecualizar histograma de frame en escla de gris
158     equalizeHist( frame_gris, frame_gris);
159     /*****
160         DETECCION DE ROSTRO
161     *****/
162     if (region_rostro.width < 100 && region_rostro.height <
163         100) {
164         detector_rostro(frame_gris, roi_rostro,
165             region_rostro);
166     }
167     /*****
168         SEGUIMIENTO DE ROSTRO
169     *****/
170     else {
171         seguimiento_rostro(frame_gris, roi_rostro,
172             region_rostro);
173     /*
174         *****/
175         SI EL SEGUIMIENTO DEL ROSTRO ES SATISFACTORIO
176         *****/
177         */
178     if(region_rostro.width > 100 && region_rostro.height
179         > 100) {
180         // Ribujar un rectangulo en la region del rostro
181         rectangle(frame, region_rostro, CV_RGB(0,255,0)
182             ,4);
183         // Recortar el rostro para reducir la region de
184             busqueda de los ojos
185         Rect region_rostro_recortada(region_rostro.x,
186             region_rostro.y+region_rostro.height*0.1,
187             region_rostro.width , region_rostro.height

```

```

*0.5);
178 Mat Rostro=frame_gris(region_rostro_recortada);
179 imshow("rostro",Rostro);
180 equalizeHist( Rostro , Rostro);
181 /*****
182     DETECTAR EL OJO IZQUIERDO
183     *****/
184     detector_ojo_izq(Rostro,roi_ojo_izq,
185         region_ojo_izq);
186     // SI LA DETECCION DE OJOS ES SATISFACTORIA
187     if(region_ojo_izq.width != 0 && region_ojo_izq
188         .height != 0) {
189         // dibujar un rectangulo en la region del
190         ojo
191         Rect ojo_izq(region_rostro.x+region_ojo_izq
192             .x, region_rostro.y+region_ojo_izq.y+
193             region_rostro.height*0.1, region_ojo_izq
194             .width, region_ojo_izq.height);
195         rectangle(frame,ojo_izq, CV_RGB(255,30,0)
196             ,2);
197         // recortar la region del ojo para eliminar
198         la zona de la ceja
199         recortar_roi(roi_ojo_izq,ojo_izq,
200             recorte_ojo_izq);
201         train_data(recorte_ojo_izq,recorte_ojo_izq)
202             ;
203         clasificacion_ojo_izq = knn_ojo_izq.
204             find_nearest(recorte_ojo_izq, 1);
205         plot_clasifi(recorte_ojo_izq,
206             clasificacion_ojo_izq,datos_izq);
207     }
208
209 /*****
210     DETECCION DEL OJO DERECHO
211     *****/
212     detector_ojo_dere(Rostro,roi_ojo_dere,
213         region_ojo_dere);
214     // SI LA DETECCION DE OJOS ES SATISFACTORIA
215     if(region_ojo_dere.width != 0 &&
216         region_ojo_dere.height != 0) {

```



```

203 // dibujar un rectangulo en la region del ojo
204 Rect ojo_dere(region_rostro.x+region_ojo_dere.
      x,region_rostro.y+region_ojo_dere.y+
      region_rostro.height*0.1, region_ojo_dere.
      width, region_ojo_dere.height);
205 rectangle(frame,ojo_dere, CV_RGB(255,30,0),2);
206 // recortar la region del ojo para eliminar la
      zona de la ceja
207 recortar_roi(roi_ojo_dere,ojo_dere,
      recorte_ojo_dere);
208 train_data(recorte_ojo_dere, recorte_ojo_dere)
      ;
209 clasificacion_ojo_dere = knn_ojo_dere.
      find_nearest(recorte_ojo_dere, 1);
210 }
211
212 // CALCULAR LA DISTANCIA ENTRE LOS OJOS
213 double dy = (region_ojo_dere.y - region_ojo_izq.y
      ) * 0.5;
214 double dx = (region_ojo_dere.x - region_ojo_izq.x
      ) * 0.5;
215 int distancia_ojos = sqrt(dx*dx + dy*dy);
216 if (distancia_ojos < 20) {
217     tiemp_distraccion++;
218     if(tiemp_distraccion > 15) {
219         // imprimir el resultado en pantalla
220         putText(frame, "<DISTRACCION>", Point
            (10,50), FONT_HERSHEY_TRIPLEX, 1, CV_RGB
            (255, 255, 0), 2.0);
221         tiemp_distraccion=0;
222         tiemp_somnolencia=0;
223
224     }
225 }
226 else {
227     tiemp_distraccion=0;
228     // CALCULAR EL ANGULO ENTRE LOS DOS OJOS
229     orientacion(dx,dy,tiemp_orientacion,angulo);
230     // ALERTA DE SOMNOLENCIA

```

```

231         if (clasificacion_ojo_izq == 1 &&
232             clasificacion_ojo_dere == 1) {
233             tiemp_somnolencia++;
234             if(tiemp_somnolencia > 4) {
235                 putText(frame, "*SOMNOLENCIA*", Point
236                     (10,50), FONT_HERSHEY_TRIPLEX, 1,
237                     CV_RGB(255, 0, 0), 2.0);
238             }
239         }
240     else
241         tiemp_somnolencia = 0;
242     }
243 }
244 // Mostrar video de la webcam
245 imshow("DETECTOR DE SOMNOLENCIA", frame);
246 // presionar una tecla para terminar el proceso
247 if(waitKey(25) >= 0) break;
248 tiemp_program=((double)getTickCount()-tiemp_program)/((
249     double)getTickFrequency());
250 cout<<"tiemp program "<< tiemp_program*1000<<endl;
251 }
252 return 0;
253 }

```

Anexo C

Código para la formación del archivo ojo_izquierdo.xml que contiene los TrainingData de la base de datos

```
1 #include <cv.h>
2 #include <highgui.h>
3 #include <cvaux.h>
4 #include <iostream>
5 #include <vector>
6 using namespace std;
7 using namespace cv;
8
9 int main ( int argc, char** argv )
10 {
11     int num_ojos_cerrados = 1350;
12     int num_ojos_abiertos= 2180;
13     Mat classes;
14     Mat trainingData;
15     Mat trainingImages;
16     vector<int> trainingLabels;
17
18     for(int i=0; i< num_ojos_cerrados; i++)
19     {
20         stringstream directorio_imagenes(stringstream::in |
21             stringstream::out);
22         directorio_imagenes << "/media/LeoneL/UTA/opencv/
23             dataset_B_Eye_Images/cerrados_izq/" << i << ".jpg";
24         Mat img=imread(directorio_imagenes.str(), 0);
```

```

23         img= img.reshape(1, 1);
24         trainingImages.push_back(img);
25         trainingLabels.push_back(1);
26     }
27
28     for(int i=0; i< num_ojos_abiertos; i++)
29     {
30         stringstream directorio_imagenes(stringstream::in
31         | stringstream::out);
32         directorio_imagenes << "/media/LeoneL/UTA/opencv/
33         dataset_B_Eye_Images/abiertos_izq/" << i << ".jpg"
34         ;
35         Mat img=imread(directorio_imagenes.str(),0);
36         img= img.reshape(1, 1);
37         trainingImages.push_back(img);
38         trainingLabels.push_back(-1);
39     }
40     Mat(trainingImages).copyTo(trainingData);
41     trainingData = trainingData.reshape(1,trainingData.rows);
42     trainingData.convertTo(trainingData, CV_32FC1);
43     Mat(trainingLabels).copyTo(classes);
44
45     FileStorage fs("SVM_ojo_izquierdo.xml", FileStorage::
46     WRITE);
47     fs << "TrainingData" << trainingData;
48     fs << "classes" << classes;
49     fs.release();
50     cout<<"Proceso terminado";
51     return 0;
52 }

```


Anexo D

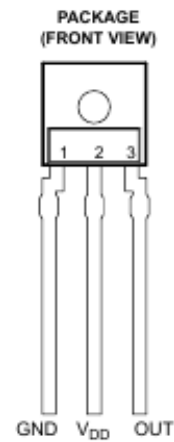
DATASHEET TSL235R CONVERTIDOR DE LUZ A FRECUENCIA



TSL235R LIGHT-TO-FREQUENCY CONVERTER

TAOS038 – APRIL 2002

- High-Resolution Conversion of Light Intensity to Frequency With No External Components
- Communicates Directly With a Microcontroller
- Compact Three-Leaded Clear-Plastic Package
- Single-Supply Operation Down to 2.7 V
- Nonlinearity Error Typically 0.2% at 100 kHz
- Stable 150 ppm/°C Temperature Coefficient
- Single-Supply Operation



Description

The TSL235R light-to-frequency converter combines a silicon photodiode and a current-to-frequency converter on a single monolithic CMOS integrated circuit. Output is a square wave (50% duty cycle) with frequency directly proportional to light intensity (irradiance) on the photodiode. The digital output allows direct interface to a microcontroller or other logic circuitry. The device has been temperature compensated for the ultraviolet-to-visible light range of 320 nm to 700 nm and responds over the light range of 320 nm to 1050 nm. The TSL235R is characterized for operation over the temperature range of -25°C to 70°C and is supplied in a 3-lead clear plastic side-looker package with an integral lens.

Functional Block Diagram



TSL235R LIGHT-TO-FREQUENCY CONVERTER

TAOS038 – APRIL 2002

Absolute Maximum Ratings over operating free-air temperature range (unless otherwise noted)[†]

Supply voltage, V_{DD} (see Note 1)	6 V
Operating free-air temperature range, T_A	-25°C to 70°C
Storage temperature range, T_{stg}	-25°C to 85°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds	240°C

[†] Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTE 1: All voltage values are with respect to GND.

Recommended Operating Conditions

	MIN	NOM	MAX	UNIT
Supply voltage, V_{DD}	2.7	5	5.5	V
Operating free-air temperature range, T_A	-25		70	°C

Electrical Characteristics at $V_{DD} = 5\text{ V}$, $T_A = 25^\circ\text{C}$ (unless otherwise noted)

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
V_{OH} High-level output voltage	$I_{OH} = -4\text{ mA}$	4	4.5		V
V_{OL} Low-level output voltage	$I_{OL} = 4\text{ mA}$		0.25	0.4	V
I_{DD} Supply current			2	3	mA
Full-scale frequency [‡]		500			kHz
Temperature coefficient of output frequency	Wavelength < 700nm		+150		ppm/°C
k_{SVS} Supply-voltage sensitivity	$V_{DD} = 5\text{ V} + 10\%$		+0.5		%/V

[‡] Full-scale frequency is the maximum operating frequency of the device without saturation.

Operating Characteristics at $V_{DD} = 5\text{ V}$, $T_A = 25^\circ\text{C}$

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
f_O Output frequency	$E_p = 430\ \mu\text{W}/\text{cm}^2$, $\lambda_p = 635\text{ nm}$	200	250	300	kHz
	$E_p = 0\ \mu\text{W}/\text{cm}^2$		0.4	10	Hz
Nonlinearity [§]	$f_O = 0\text{ kHz to } 10\text{ kHz}$		+0.1%		%F.S.
	$f_O = 0\text{ kHz to } 100\text{ kHz}$		+0.2%		%F.S.
Step response to full-scale step input			1 pulse of new frequency plus 1 μs		

[‡] Full-scale frequency is the maximum operating frequency of the device without saturation.

[§] Nonlinearity is defined as the deviation of f_O from a straight line between zero and full scale, expressed as a percent of full scale.

TYPICAL CHARACTERISTICS

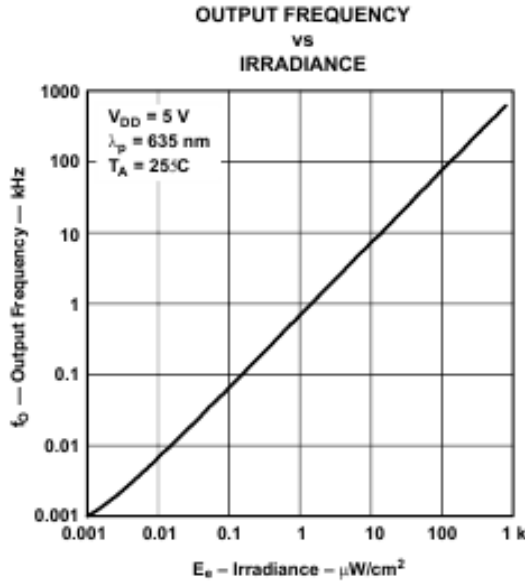


Figure 1

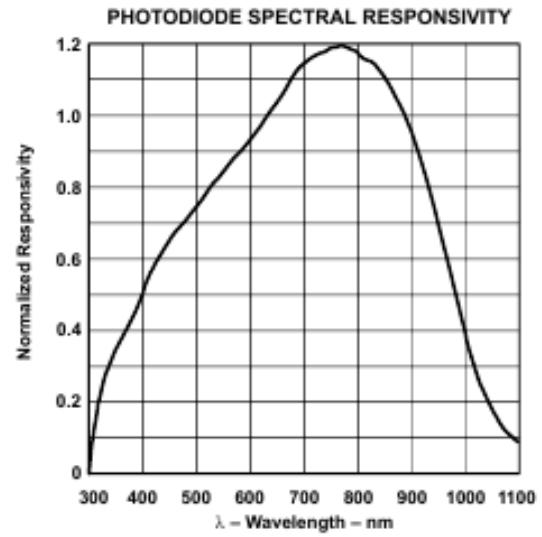


Figure 2

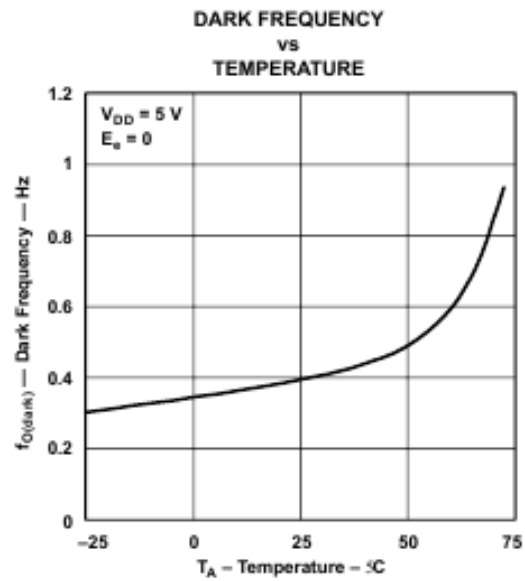


Figure 3

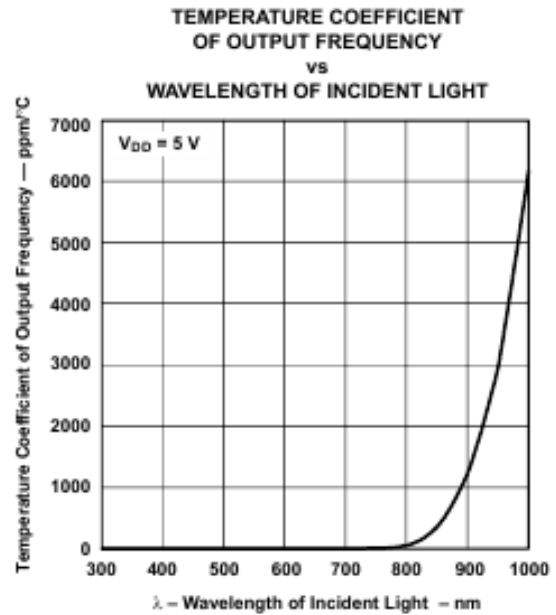


Figure 4

TSL235R LIGHT-TO-FREQUENCY CONVERTER

TAOS038 – APRIL 2002

TYPICAL CHARACTERISTICS

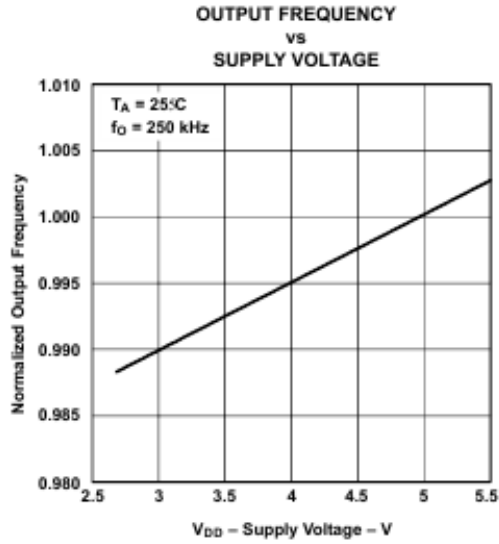


Figure 5



APPLICATION INFORMATION

Power-supply considerations

Power-supply lines must be decoupled by a 0.01- μ F to 0.1- μ F capacitor with short leads placed close to the TSL235R (Figure 6).

Output interface

The output of the device is designed to drive a standard TTL or CMOS logic input over short distances. If lines greater than 12 inches are used on the output, a buffer or line driver is recommended.

Measuring the frequency

The choice of interface and measurement technique depends on the desired resolution and data-acquisition rate. For maximum data-acquisition rate, period-measurement techniques are used.

Period measurement requires the use of a fast reference clock with available resolution directly related to reference-clock rate. The technique is employed to measure rapidly varying light levels or to make a fast measurement of a constant light source.

Maximum resolution and accuracy may be obtained using frequency-measurement, pulse-accumulation, or integration techniques. Frequency measurements provide the added benefit of averaging out random- or high-frequency variations (jitter) resulting from noise in the light signal. Resolution is limited mainly by available counter registers and allowable measurement time. Frequency measurement is well suited for slowly varying or constant light levels and for reading average light levels over short periods of time. Integration, the accumulation of pulses over a very long period of time, can be used to measure exposure — the amount of light present in an area over a given time period.

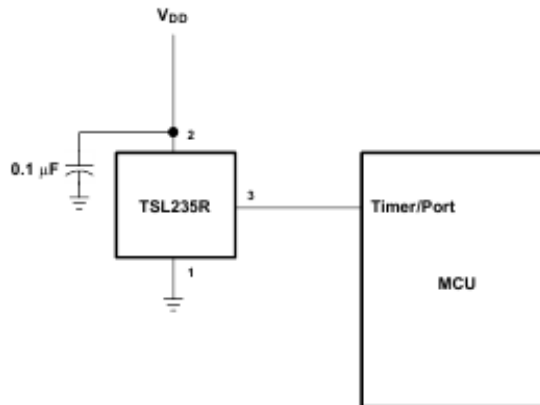
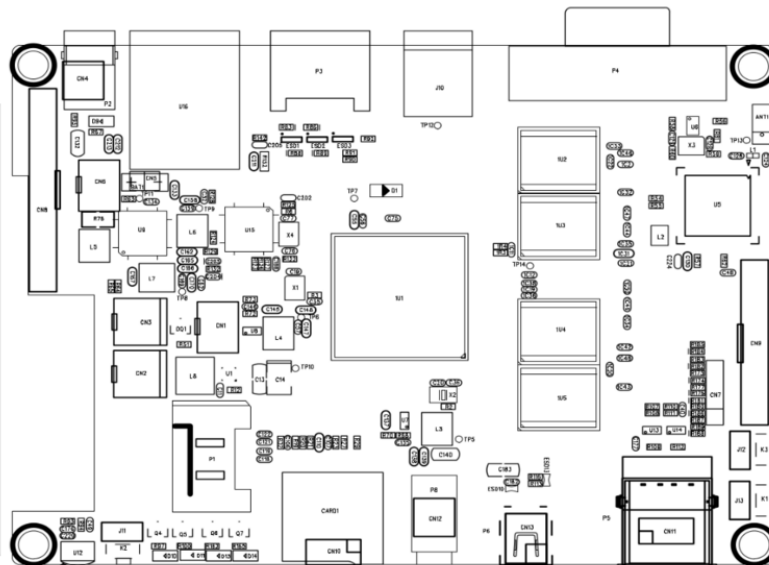


Figure 6. Typical TSL235R Interface to a Microcontroller

Anexo E

DISTRIBUCIÓN DE LOS GPIO DE LA CUBIETRUCK

CN8			
PIN	DEF	PIN	DEF
30	LRADC1	29	VCC-5V
28	LRADC0	27	CVBS
26	LINEINR	25	PB19
24	LINEINL	23	PB18
22	IR-RX	21	PB2
20	IR-TX	19	PI3
18	PI15	17	PI21
16	PI14	15	PI20
14	GND	13	GND
12	SPI0-MISO	11	SPI0-CLK
10	SPI0-MOSI	9	SPI0-CS
8	PC22	7	PC20
6	PC21	5	PC19
4	RESET#	3	AVCC
2	VCC-3V	1	GND



CN9			
PIN	DEF	PIN	DEF
1	VCC-3V	2	VCC-3V
3	PG0	4	PG3
5	PG2	6	PG1
7	PG4	8	PG5
9	PG6	10	PG7
11	PG8	12	PG9
13	PG10	14	PG11
15	GND	16	GND
17	XP	18	TVIN0
19	XN	20	TVIN1
21	YP	22	TVIN2
23	YN	24	TVIN3

CN8 (Near Ethernet connector)			
2x15 Header			
30	LRADC1	29	VCC-5V
28	LRADC0	27	CVBS
26	LINEIN-R	25	PB19(TWI1-SDA)
24	LINEIN-L	23	PB18(TWI1-SCK)
22	PB4 (IR0-RX)	21	PB2 (PWM0)
20	PB3 (IR0-TX))	19	PI3 (PWM1)
18	PI15 (PS2SDA1/EINT27)	17	PI21 (FMIN-R/PS2SDA0/UART7-RX/HSDA)
16	PI14 (PS2SCLK1/EINT26)	15	PI20 (FMIN-L/PS2SCLK0/UART7-TX/HSCL)
14	GND	13	GND
12	PB17 (SPI2-MISO/JTAG_DI0)	11	PB15(SPI2-CLK/JTAG_CK0)
10	PB16 (SPI2-MOSI/JTAG_DO0)	9	PB14 (SPI2-CS0/JTAG_MS0)
8	PC22 (SPI2-MISO/EINT15)	7	PC20 (SPI2-CLK)
6	PC21 (SPI2-MOSI/EINT14)	5	PC19 (SPI2-CS0)
4	RESET#	3	AVCC
2	3.3V	1	GND

CN9 (Near USB Ports)			
CSI1/TS/TP/TVIN			
1	3.3V	2	3.3V
3	PG0 (TS1_CLK/CSI1-PCLK)	4	PG3 (TS1_ERR/CSI1-VSYNC)
5	PG2 (TS1_SYNC/CSI1-HSYNC)	6	PG1 (TS1_DVLD/CSI1-MCLK)
7	PG4 (TS1_D0/CSI1-D0)	8	PG5 (TS1_D1/CSI1-D1)
9	PG6 (TS1_D2/CSI1-D2/UART3-TX)	10	PG7 (TS1_D3/CSI1-D3/UART3-RX)
11	PG8 (TS1_D4/CSI1-D4/UART3-RTS)	12	PG9 (TS1_D5/CSI1-D5/UART3-CTS)
13	PG10 (TS1_D6/CSI1-D6/UART4-TX)	14	PG11 (TS1_D7/CSI1-D7/UART4-RX)
15	GND	16	GND
Analog			
17	XP-I2SDO1	18	TVIN0-I2SMCLK
19	XN-I2SDO2	20	TVIN1-BTPCMCLK
21	YP-I2SDO3	22	TVIN2-BTPCMSYNC
23	XN-BTPCMIN	24	TVIN3-BTPCMOUT