"Responsabilidad con pensamiento positivo"

UNIVERSIDAD TECNOLÓGICA ISRAEL

TRABAJO DE TITULACIÓN EN OPCIÓN AL GRADO DE: INGENIERO EN ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES

TEMA:
SISTEMA DE DETECCIÓN Y ALERTA DEL ESTADO DE SOMNOLENCIA DE CONDUCTORES MEDIANTE VISIÓN ARTIFICIAL

AUTOR:
Cristian Patricio Baiza Lovato

TUTOR:
Mg. Francisco Jurado Pruna

QUITO, ECUADOR 2020

DECLARACIÓN

UNIVERSIDAD TECNOLÓGICA ISRAEL

DERECHOS DE AUTOR:

Yo **Cristian Patricio Baiza Lovato,** siendo el autor intelectual y titular del proyecto investigativo, asumiendo morales y patrimoniales de titulación, dando consentimiento con referencia al artículo: "*114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN",* declaro la total autoría del presente proyecto.

Por esta manera, el autor declara con la presente autorización, que la obra suscrita es original y que no infringe ningún derecho a terceros, responsabilizándose por cualquier tipo de reclamo que se encuentro en temas de autoría y liberando en su totalidad a la institución educativa de toda responsabilidad que esto causase.

D.M. Quito, 5 marzo del 2020

AUTOR

Cristian Patricio Baiza Lovato
C.I. 1721445730

CERTIFICACIÓN DEL TUTOR

UNIVERSIDAD TECNOLÓGICA ISRAEL

Por medio de la presente y en mi rol de tutor de investigación, certifico:

Que el presente trabajo con el tema: "SISTEMA DE DETECCIÓN Y ALERTA DEL ESTADO DE SOMNOLENCIA DE CONDUCTORES MEDIANTE VISIÓN ARTIFICIAL" el cual es desarrollado por: **Cristian Patricio Baiza Lovato**, en su calidad de estudiante que finaliza la carrera de: *"Electrónica Digital y Telecomunicaciones"*; completa los requisitos planteados por la institución y está apto para ser sometido a una posterior evaluación.

D.M. Quito, 5 de Marzo del 2020

TUTOR

……..
Ing. Francisco Jurado.

AGRADECIMIENTO

Este proyecto investigativo, le dedico principalmente a Dios por regalarme tantas inquietudes por resolver aún.

Quiero extender un agradecimiento a mis padres y mi hermana, quienes han estado siempre a mi lado apoyándome en cada paso que doy, en todo momento con un aliento que ha hecho que este logro sea mucho más satisfactorio.

Agradezco de igual manera a mi novia que siempre ha estado a mi lado en los mejores y peores momentos que he pasado, sus palabras han hecho que mi meta se pueda cumplir.

Cristian Patricio Baiza Lovato

DEDICATORIA

A Dios por sobre todas las cosas, porque nunca me ha abandonado y me ha dado fuerzas, para realizar y terminar con éxito este trabajo.

A mis padres y mi hermana, que mediante su grato aprecio han generado bendiciones y un ejemplo de lucha para poder seguir.

A mis abuelos paternos que desde pequeño siempre supieron guiarme por el mejor camino y ahora estoy logrando lo que tanto he anhelado.

A mi abuela materna porque desde que fui un niño me apoyó en cada paso como hasta el día de hoy, siempre con su cariño supo mejorar mi estado de ánimo para poder seguir adelante.

A mi novia que siempre me apoyó desde que comencé esta travesía, sus palabras de cariño y afecto han hecho que pueda alcanzar una de mis metas.

Cristian Patricio Baiza Lovato

TABLA DE CONTENIDOS

LISTA DE FIGURAS

LISTA DE TABLAS

LISTA DE GRÁFICOS

RESUMEN

El vigente trabajo de investigación es un proyecto que desarrollará un prototipo el cual esté en la capacidad de detectar un estado de somnolencia en los conductores, mismo que será detectado con un método de inteligencia artificial a quien se lo conoce como visión artificial, tomando como parámetro específico el estado que tienen los usuarios al momento de parpadear en lapsos de tiempo establecidos, detectando así un posible estado de sueño.

La solución está orientada a conductores que tengan un cierto nivel de somnolencia constante en horas de conducción, mediante un prototipo portátil diseñado con microrordenadores, por su fácil transporte y su adaptabilidad a programas de reconocimiento facial.

El proyecto diseñado con un módulo *Raspberry* pi y librerías OpenCv de reconocimiento facial, así como programado mediante el lenguaje Python; es capaz de adaptarse a diferentes conductores, tomando una secuencia de video para generar un patrón de estado de somnolencia en los conductores, que si sobrepasa un lumbral del 80% emitirá una alarma sonora y otra visual, la cual pondrá al usuario en modo de alerta, previniendo cualquier tipo de accidente que pueda causar un estado de sueño prolongado. El prototipo en cuestión usará una cámara integrada con sensores normales de movimiento, la cual dará una prevención sonora y otra visual al conductor para que vuelva a tener un estado de alerta.

Palabras clave: somnolencia, prototipo de detección de sueño, reconocimiento facial, visión artificial.

ABSTRACT

The present research work is a project that will develop a prototype which is capable of detecting a state of drowsiness in drivers, using an artificial intelligence technique known as artificial vision, based specifically on the state that users have at the time of blinking in established time periods, thus detecting a possible state of sleep.

The solution is aimed at drivers who have a certain level of constant sleepiness in driving hours, using a portable prototype designed with microcomputers, for its easy transport and its adaptability to facial recognition programs.

The project designed with a Raspberry pi module and OpenCv libraries for facial recognition, as well as programmed using the Python language; is able to adapt to different drivers, taking real-time images to generate a pattern of driver drowsiness, which must be greater than 80% to emit an audible alarm, which will put the user in alert mode, preventing in this way any type of accident that can cause a prolonged state of sleep. The prototype in question will use an integrated camera with normal motion sensors, which normally works in natural light conditions, that is; It does not work in darkness.

Keywords: drowsiness, sleep detection prototype, facial recognition, artificial vision.

INTRODUCCIÓN

Antecedentes de investigación

Según el autor Carrillo-Mora (2013), dice que: "*El acelerado ritmo de vida en la actualidad, hace que muchas personas empiecen a padecer de perturbaciones en el sueño acarreando consecuencias negativas en su vida diaria*" (Carrillo-Mora, Ramírez-Peris, & Magaña-Vázquez, 2013, pág. 40). El dormir es una función básica para el bienestar de las personas; aunque se ha visto en aumento una tendencia mundial en reducir las horas de sueño por factores externos y hábitos. En el Hospital Intermutual de Levante (2019) concluyen que: *"no dormir las horas necesarias provoca somnolencia, falta de concentración, dolor corporal, rendimiento bajo, entre otros síntomas"* (Hospital Intermutual de Levante., 2019)

Entrar en un estado de falta de sueño o somnolencia tiene un efecto negativo directo en sentidos de percepción, disminuyendo su capacidad de reacción auditiva provocando que decrezcan los reflejos de reacción del ser humano e incrementando la probabilidad de sufrir un accidente en carretera si se conduce en este estado.

Según Egas Cunalata (2017): *"numerosos estudios del sueño, están vinculados con miles de incidentes vehiculares cada año"* (Egas Cunalata, F. D., 2017). Este dato se puede corroborar con la información de la Agencia Nacional de Tránsito, en Ecuador existen registros de múltiples accidentes causados por somnolencia y fatiga por falta de sueño sobre los conductores.

Actualmente varios fabricantes de automóviles de gama alta se encuentran desarrollando Sistemas Avanzados de Conducción (SAAC) que tienen como objetivo el tener una mejor seguridad en los vehículos y de esta manera aumentar la probabilidad de prevenir accidentes. Estos sistemas detectan: peatones, señales de tránsito, estado del conductor, entre otros. (Crespín Luis, J. C., & Julián García, R. A., 2014).

Estos vehículos no son accesibles para las personas de clase media, por el valor de mantenimiento que tienen y por su elevado costo de adquisición, en general se recurre a la compra de vehículos de valor bajo, los cuales no constan de esta tecnología para el transporte diario.

El desarrollo de esta investigación se respalda en los siguientes trabajos:

En Flores (2018) quien implementó la detección de somnolencia en el conductor basado en iluminación infrarroja para seguimiento nocturno del estado del conductor (Flores, M. J., Armingol, J. M., & Escalera, A. de la.).

En Vats (2012) quien desarrolló un proyecto que determina los niveles de sueño sobre conductores, utilizando las redes neuronales y técnicas de visión artificial, determinando que es necesario de un control con al menos tres neuronas para lograr una precisión del 98% en los usuarios que presenten niveles de somnolencia (Vats, E. M., & Garg, E. A.).

En Crespín y García (2014) analizaron secuencias de grabaciones de conductores usando visión para detectar si un conductor está teniendo síntomas de falta de sueño, el procesamiento se planteó mediante una computadora (Crespín Luis, J. C., & Julián García, R. A.).

En Conlago y Yunda (2016) que realizó una aplicación sobre señales de tránsito para que sean detectadas automáticamente, mismo que utilizó técnicas artificiales de detección para analizar los espacios de color y forma en diversas condiciones: climáticas, iluminación, rotaciones y oclusiones, y determinar ante que señal se encuentra el vehículo a una distancia máxima de 20 metros; este proyecto fue desarrollado en Ecuador (Conlago Guatemal, C. R., & Yunda Sangoluisa, J. A.).

En López (2016) quien estableció un prototipo de detección de apertura de los ojos, distracción y orientación del conductor con algoritmos de visión artificial y oximetría del pulso, todo esto se procesó en una tarjeta Cuboetruck, se realizó pruebas del prototipo en las avenidas de Ambato concluyendo que la respuesta es lenta (López Romero, W. L.).

En Palma y Torres (2017) desarrollaron un sistema móvil para detectar síntomas de sueño en usuarios de transporte público, utilizando un método basado en porcentajes, donde los ojos de un usuario permanecen cerrados, más conocido por sus siglas: “PERCLOS”, obteniendo un 90% de eficiencia en la detección (Palma Jaramillo, M. A., & Torres Berrú, Y.).

Tomando como referencia a todas las investigaciones mencionadas y planteando superar los desafíos que conlleva la detección de somnolencia en los conductores, se propone desarrollar

el presente trabajo de grado, con el tema: “Sistema de detección y alerta del estado de somnolencia en conductores mediante visión artificial.”.

Planteamiento del problema

Conducir en estado somnoliento es un factor que presenta gran riesgo por los accidentes viales, y en horas nocturnas la cifra de accidentes se triplica. En datos que se presentan a nivel mundial, el conducir en un estado de falta de sueño es causante de una considerable tasa de accidentes de tránsito dejando fallecidos, pérdida de recursos y heridos. La somnolencia durante la conducción conlleva aproximadamente el mismo riesgo que tiene el consumo de sustancias alcohólicas (Dallas, M. E., 2018).

Actualmente no existe una prueba diagnóstica que determine si una fatiga por escases de sueño sea el detonante de algún tipo de accidente al conducir, muy diferente a la ingesta de alcohol que puede ser detectado en el organismo a través de diversos métodos. Para determinar si la falta de sueño fue causa de un accidente vial se hace un análisis de las variables que están en el accidente, como son: en tiempo en el que se manejaba, si el conductor viaja solo, el vehículo se sale de ruta o invade otro carril, entre otras (Mujica, J. R., Mayor, E. R., & Rojas, M. E., 2010).

Mediante la investigación, desarrollo e implementación de un prototipo que pueda detectar los niveles de somnolencia, se puede evitar un gran porcentaje de posibles accidentes, enviando señales de alerta sonoras y sensitivas al conductor para que pueda tomar las acciones respectivas como detenerse o cambiar de conductor y así salvaguardar su integridad, la de los transeúntes y demás conductores. El sistema no debe interferir con la visibilidad del conductor y debe realizar una ejecución continua durante la conducción.

Justificación

Es necesario que las personas se puedan movilizar entre sus puntos de encuentro, sean por motivos personales, de trabajo, enfermedad, ocio, etc. sabiendo que en estos trayectos no están exceptos de cualquier tipo de accidente que pueden ser producidos por varias razones. Una de ellas puede ser la somnolencia que provoca la pérdida de concentración y un decremento en

la reacción del conductor, ocasionando siniestros automovilísticos que generan lesiones, muertes y pérdidas económicas tanto a conductores, transeúntes y pasajeros.

Es por ello que se ha considerado el desarrollo de un prototipo que tenga la capacidad de detectar el estado de somnolencia en conductores, basándose en técnicas de visión artificial para alertar así a los usuarios de vehículos livianos, analizando mediante procesamiento de imágenes y cambios faciales del conductor, haciendo un enfoque en el rito de parpadeo de los ojos que tiene.

El presente proyecto busca alertar a quienes están detrás de un volante sobre el estado que produce la falta de sueño, para que así tomen un descanso y procedan con su viaje; y, en casos extremos dándole la posibilidad de generar alguna maniobra que evite un posible accidente.

Objetivo General

Desarrollar un sistema de detección y alerta del estado de somnolencia para conductores aplicando técnicas de visión artificial.

Objetivos específicos

- Determinar las principales señales de somnolencia en conductores para aplicar técnicas de detección según documentación relacionada con los sistemas avanzados de ayuda.
- Construir algoritmos de visión artificial en Open CV para la detección de estado de somnolencia en conductores.
- Construir un prototipo que proporcione señales de alerta al conductor ante la presencia de estado de somnolencia.
- Validar el desempeño del sistema en tiempo real para evaluar la velocidad y precisión del mismo ante la presencia de síntomas de somnolencia y cambios de iluminación en el transcurso del día.

Hipótesis

- La visión artificial ofrece múltiples ventajas para la detección de somnolencia ya que con pocos recursos permite obtener una percepción amplia del entorno, dándole así al sistema suficiente información para determinar si ha ocurrido la somnolencia durante la conducción.
- La detección de los signos de somnolencia temprana mediante el sistema permitirá alertar de manera eficiente de su estado al conductor dando un apoyo adicional a la seguridad del mismo.

Alcance

En el presente proyecto de investigación se realizará la implementación de un sistema de detección del estado de somnolencia para conductores mediante visión artificial con el fin de alertar al conductor y evitar accidentes de tránsito.

Para realizar el procesamiento que permite la detección del estado de somnolencia se va a ocupar un equipo con características en hardware y software capaces de soportar la ejecución de algoritmos de visión artificial.

Utilizando visión artificial se va a determinar los parámetros faciales del conductor para establecer el estado de somnolencia, el sistema de alarma electrónico se activará al detectar signos de somnolencia en el conductor.

Con el método de detección de somnolencia, se prosigue con la etapa de elaboración del prototipo y finalmente se realizarán pruebas de funcionamiento de los algoritmos desarrollados para determinar el desempeño del prototipo y corregir las falencias que pueda presentar en acción. Dichos resultados serán analizados y de acuerdo a su relevancia podrían ser presentados en un documento científico.

Descripción de los capítulos

La investigación empieza con un resumen del proyecto, seguido del capítulo introductorio que describe brevemente los objetivos, alcances, hipótesis y justificación del

proyecto. Seguido por el cuerpo de la investigación que cuenta con cuatro capítulos: el primero plantea la somnolencia, visión artificial, y los microrordenadores; el segundo trata sobre la metodología que se va a utilizar para el análisis y recolección de datos, el tercer capítulo trata sobre los detalles específicos de la propuesta, indicando las herramientas que se van a utilizar y un estudio de opciones para hallar la más óptima en cuestión de tiempos y costos. Posteriormente en el capítulo cuarto, se detalla la implementación, pruebas y análisis de resultados que tuvo como respuesta la propuesta planteada. Para finalizar, se concluye el proyecto y se realizan recomendaciones del mismo.

CAPÍTULO 1

FUNDAMENTACIÓN TEÓRICA

1.1. Somnolencia

Según el autor Mayor (2010) define la somnolencia como:

> *"La somnolencia, o la tendencia a quedarse dormido, puede ser el principal síntoma de diversas patologías. La causa más común de somnolencia es la privación de sueño. Su medición es compleja debido a sus diferentes conceptos operacionales. Los instrumentos más utilizados son los cuestionarios de auto-evaluación que miden la somnolencia subjetiva. La somnolencia excesiva tiene un impacto en la salud mental y física de la persona que lo sufre, por lo que es de suma importancia que el personal de salud pueda evaluar esta condición y determinar la causa; o derivarlos al especialista en trastornos del sueño si el caso es complejo"* (Edmundo Rosales Mayor, pág. 32).

Se puede resumir a la somnolencia como un trastorno en el cual los sentidos pierden su agudeza, siendo un estado entre la lucidez y el sueño profundo, el mismo que puede causar perturbaciones en la conciencia.

1.1.1. Factores y características

Entre los factores que causan un estado de somnolencia pueden ser primarias o secundarias. El síndrome de Kleine-Levin es un tipo de hipersomnia primaria que tiene un origen central en la falta de sueño el cual es conocido como trastorno de narcolepsia o hipersomnia idiopática.

Las causas secundarias del trastorno de sueño se pueden dividir en otros dos subgrupos: teniendo como primer subgrupo a los que se relacionan con un déficit respiratorio, el cual se lo conoce mejor por sus siglas "SAHS", cuyo comportamiento causa una privación en el sueño, lo cual es más frecuente en trabajos de noche; otro trastorno que se engloba en este primer subgrupo es el llamado "Jet Lag", el cual es una alteración al ciclo circadiano o conocido como síndrome de las piernas inquietas que hace referencia a movimiento variante de los miembros

inferiores. En el segundo de este subgrupo se menciona a las condiciones médicas causadas por traumas en el cerebro, como accidentes encefálicos, condiciones neurodegenerativas, enfermedades psicológicas como depresión o incluso algún grado de cáncer que afecte directamente a la materia gris; aquí se incluyen también algunos efectos secundarios que pueden ser causa de medicamentos recetados y que contengan un alto grado de químicos que afecten directamente al adecuado ciclo del sueño.

Entre sus características están las siguientes:

- Aumento en la dificultad de concentración, parpadeo repetitivo y párpados pesados.
- Tener pensamientos errantes y micro sueños, conocidos como: soñar despierto.
- Complejidad para recordad las últimas distancias conducidas; las salidas o señales de tráfico.
- Repetidos bostezos.
- Falta de control para tener la cabeza recta.
- Estar en alejamiento del carril principal, cambio de carril y desviación.

En la Tabla 1 se muestran varios de los tipos de trastornos del sueño y las consecuencias que estos pueden llegar a tener en una persona (Fundación CEA, 2015).

Tabla 1: Trastornos del sueño

Trastornos del sueño	Descripción	Alteraciones en la conducta
Insomnio	Una dificultad constante para mantener o iniciar el sueño.	Baja concentración, somnolencia, extrema fatiga e irritabilidad.
Narcolepsia	Ataques periódicos y sueño durante el día	Probabilidad de sufrir ataques epilépticos durante el día
Hipersomnia	Facilidad para dormirse durante el día y somnolencia prolongada en la noche	Ataques de somnolencia periódicos, bajo nivel de concentración.
Trastornos respiratorios	Alteración en las ondas respiratorias naturales, provocando somnolencia periódica	Problemas de concentración, somnolencia persistente, agotamiento.
Trastornos cardiacos	Cambio del patrón de sueño	Fatiga, irritabilidad y problemas de concentración

Fuente: (Fundación CEA, 2015)

1.1.2. Consecuencias de somnolencia al conducir

El ser humano necesita del sueño para tener una actividad normal en sus funciones diarias, siendo un mecanismo de recolección de energía y descanso natural. Cuando se trata de tareas que requieren un grado alto de concentración, como es el conducir, es necesario estar en las óptimas condiciones de descanso.

El conducir es una tarea que requiere un estado de esfuerzo alto, tanto mentalmente como físico, para ello se presentan varios síntomas que indican que el conductor está entrando en un estado de falta de sueño:

- Agotamiento, pesadez y picor constante en los ojos.
- Brazos fatigados con una combinación de dolor de cabeza.
- Fatiga física y mental.
- Signos de adormecimiento con el calor interior del carro.

Entre las alteraciones que se encuentran por causa de la somnolencia ya afectan de manera directa al conductor, son las siguientes:

Alteración en la reacción: Al entrar en un estado donde las reacciones motoras se ven alteradas, el tiempo de respuesta se ve disminuido, siendo proclive a causar choques involuntarios. Como ejemplo se puede presentar en una carretera con constante tráfico, donde los vehículos que se encuentran por delante y detrás quieren proseguir y la lenta reacción causada por somnolencia, hará que se produzca una posible colisión.

Mayor distracción y una menor concentración: La falta de sueño produce que tener una concentración fija sea más complicada en un solo punto, por lo que instintivamente se intenta buscar distracciones para mantener la concentración, siendo esto causa de problemas en vez de una oportuna solución.

Errores en la toma de decisiones lentas: Una falta de sueño y posible somnolencia, es causante de que el conductor procese la información del entorno en una escala más lenta y que reaccione con relativa dificultad, cometiendo en muchos de los casos, errores que conllevaran a accidentes.

Cambios de reacciones motoras: En el momento que el cuerpo empieza a entrar en un estado de somnolencia, los músculos de las extremidades se relajan, haciendo que sus movimientos de reacción se vean afectados, ralentizándolos con leves temblores.

Automatización de movimientos: Un mecanismo de defensa cuando se entra en un estado de sueño, es realizar movimientos automáticos, este estado del cuerpo lo acciona para una rápida respuesta, pero en un entorno cambiante como lo son las impredecibles carreteras, este tipo de acción puede ser contraproducente.

Micro sueños: En pequeños periodos de tiempo, los cuales duran apenas unos segundos después de haber tenido los primeros síntomas de somnolencia, aparecen los micro sueños, que

son instantes donde se entra a un estado transitivo de sueño ligero, ignorando los acontecimientos del entorno. El problema se agrava porque estos pequeños periodos de micro sueños no son percibidos por el conductor, haciendo pensar que nunca han sucedido y volviendo a tenerlos sin la menor noción.

Cambio en las funciones sensoriales: El cambio de estado que produce la somnolencia, afecta de manera directa a la visión, produciendo fatiga ocular, visión borrosa y falta de enfoque; así mismo los sentidos tardan más tiempo en reaccionar a estímulos externos como luces fuertes o sonidos ajenos.

Percepción alterada: Cuando los sentidos están en una transición de la lucidez a la somnolencia, los objetos que se visualizan se ven de manera errada, como las señales de tránsito, las luces de otros vehículos y los sonidos que estos producen. Y, en casos extremos hasta se pueden llegar a percibir varias alucinaciones que causarán un cambio de perspectiva y posibles accidentes.

Comportamiento cambiante: Para muchas personas, el despertar es un proceso donde tienen un traslado de emociones que puede durar unos cuantos segundos, algo similar pasa cuando se está en un estado de somnolencia, presentando conductas agresivas, de tensión y nerviosismo.

1.1.3. Accidentes por somnolencia al conducir

Se debe tener en cuenta que las alteraciones del sueño empiezan a ser más notorias en personas que superan los 50 años, una gran parte de accidentes que se registran por somnolencia son realizados por jóvenes adultos con una edad que va desde los 18 y hasta los 19 años (AMERICAN SLEEP ASSOCIATION, 2019).

Uno de los estudios que ha hecho el director en el laboratorio de antropología aplicada para la universidad de París-V, el profesor Philippe Cabon, ha probado que con un reposo de aproximadamente 20 minutos en una persona que tienen fatiga a causa del poco descanso, ha reducido de modo significativo los estados de somnolencia y aumentado el rendimiento de las personas que sufrían de este tipo de cansancio.

Para poder confirmar la teoría de que un periodo de tiempo corto puede ser beneficio para recuperar algo de sueño en las personas, se procedió al estudio de un informe de la NASA, donde muestra que con un lapso de 40 minutos de descanso en el transcurso de una jornada laboral pesada y con fatiga por falta de sueño, ha mejorado en un 34% el rendimiento de los sujetos puestos a prueba.

Dentro del distrito de América del norte, se presenta un índice del 55% en donde los accidentes se causan por culpa de conductores adormilados y que tienen menos de 25 años de edad. En el estudio de la AMERICAN SLEEP ASSOCIATION difundido en el 2019, se indicó que estar sin un descanso por sueño por mas de 18 horas, es equivalente a estar en un estado con alcohol en la sangre de 0.08%, lo que implica que está legalmente en estado de embriaguez, dejando la misma probabilidad de accidente. Esta investigación ha señalado a hombres jóvenes que trabajan por turnos, que cuentan con una licencia profesional y que son personas sin registro de trastornos del sueño a corto plazo.

1.2. Visión artificial

La visión artificial es un campo interno de la inteligencia artificial que mediante una serie de algoritmos adecuadamente utilizados con técnicas de sensores especializados que captan sectores específicos de un rostro humano, permite la obtención de información para poder ser analizada como imágenes o videos digitalizados.

El análisis de las imágenes procesadas que la visión artificial captura en un conjunto de procesos se dividen en: captación orgánica, almacenamiento de la información, interpretación digitalizada y análisis de los resultados. Mediante estas técnicas se pueden realizar los siguientes proyectos

- Aligerar las tareas de repetición realizadas por operarios.
- Automatizar el análisis de calidad de productos.
- Analizar objetos sin ningún contacto físico.
- Automatizar el proceso de inventario y producción.
- Disminuir los tiempos de producción de productos.

- Captar imágenes en tiempo real en ámbitos de seguridad para un distinto tipo de usuarios.

El proceso que hace la visión artificial para poder dar un resultado óptimo se divide en varias capas, las mismas que se analizan en la Figura 1: (Ministerio de educación de España, 2012):

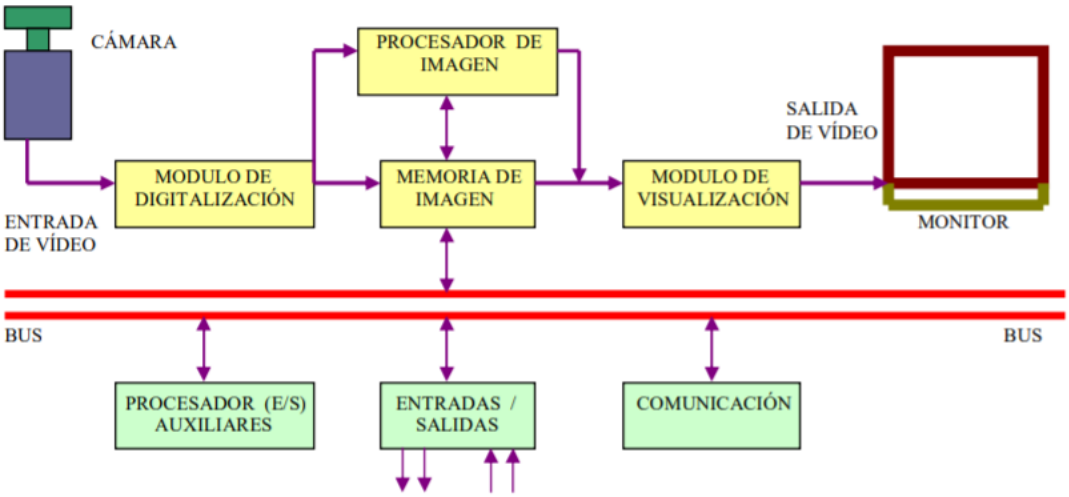


Figura 1: Diagrama de bloque de visión artificial

Fuente: (Ministerio de educación de España, 2012)

- Módulo de digitalización: convierte las señales de tipo analógico y las procesa en datos digitales.
- Memoria de imagen: almacena los datos digitalizados.
- Módulo de visualización: transforma la señal digitalizada y previamente almacenada, en una señal analógica para poder ser vista a través de una salida de video.
- Procesador de imagen: analiza las imágenes capturadas y las procesa para su posterior uso.
- Módulo de entradas/salidas: coordina las entradas de video analógicas, así como las salidas para que tengan el mismo formato.
- Comunicaciones: vía de entrada de datos, salida de respuestas o conexiones de internet.

Todo este conjunto de procesos está relacionado con la secuencia operativa de datos que tiene la visión artificial y la cual se describe en la Figura 2 (Adolfo Valdes, 2019):

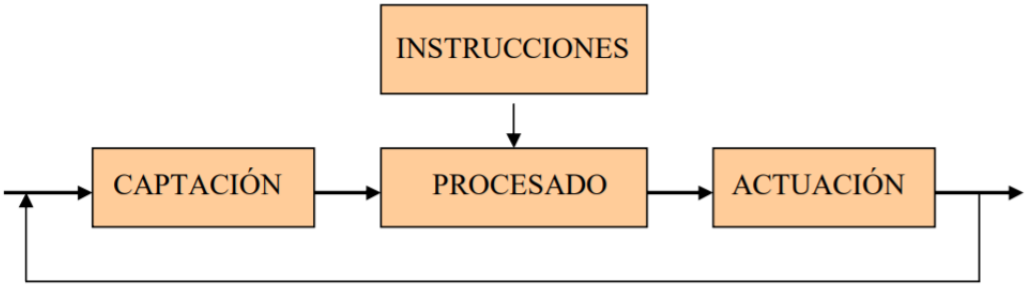


Figura 2: Diagrama operativo de visión artificial

Fuente: (Adolfo Valdes, 2019)

- Captación: entrada de la imagen para poder ser procesada y analizada.
- Instrucciones: una serie de patrones y reglas para poder procesar las imágenes y obtener un tipo de respuesta específico.
- Procesado: transformación de los datos de entrada aplicando las instrucciones solicitadas.
- Actuación: resultado de la imagen procesada y salida como una respuesta entendible para el usuario.

1.2.1. Antecedentes de la visión artificial

El nacimiento de la visión artificial fue marcado por el investigador Larry Roberts quien en el año de 1961 desarrolló un sistema el cual fue capaz de ver una estructura de bloques para después analizarla y duplicarla con una perspectiva diferente. Demostrando de esta manera el inicio de la visión artificial que mediante una cámara podía captar imágenes y procesarlas para devolver un tipo de resultado.

Después de esta primera prueba, cada investigador en el mundo lo tomó como punto de partida y empezaron a desarrollar sus propios algoritmos de detección con la visión artificial y de este modo nacieron los primeros proyectos como:

- Los primeros pasos para detección en rostros humanos.
- Registro de una misma imagen con diferentes ángulos y posiciones.
- Memorizar la misma imagen en diferentes sesiones.
- Seguimiento de uno o varios objetos en video.

1.2.2. Detección facial

Los métodos que se utilizan para el reconocimiento facial, se basan en teorías y métodos matemáticos, como patrones geométricos y análisis estadísticos de distancia con patrones simétricos. Se presentan las teorías más populares de detección:

Algoritmo de Viola Jones

En el ámbito de detección real de rostros, el método de "Viola-Jones" es una opción robusta que permite un análisis de capas. Esta secuencia de pasos lógicos es un avance en los algoritmos de detección por la clasificación individual de pixeles, lo que permite una mayor rapidez en el procesamiento y abstracción de resultados en cada fotograma analizado.

Se puede tener en cuenta 3 subdivisiones del método: 1) la posibilidad de poder evaluar las características de la imagen analizada; 2) la clasificación en método de cascada (una imagen tras otra) para escoger un conjunto de patrones, los cuales son necesarios para analizar y procesar la imagen completa; 3) y, una técnica que combina las dos anteriores y resulta una clasificación eficiente en detección al mismo tiempo que precisa por su segmentación.

Detección en cascada

El criterio primordial del método de "Viola-Jones" es el de captar una serie repetida de veces a través de un mismo datagrama, con diferentes ángulos y escalas. Mediante este análisis se considera que es ineficiente el detectar una imagen por la cantidad de datagramas que contenga, generando un cuello de botella para cualquier algoritmo que quiera procesar una imagen, ya que el sistema debe obtener, clasificar, analizar y dar un resultado por cada una de las características que conforman una imagen completa. Con el paso de la tecnología se asume que al aumentar la velocidad de procesadores generalmente se disminuiría el retraso que implica la detección de una imagen, pero el error fue que también aumentaron la calidad de las imágenes y por ende el peso de las mismas; la optimización de procesos está ligada a la disminución de análisis de imágenes para lo cual "Viola-Jones" propusieron una técnica que reduciría el período de análisis y las exigencias, el cual sería la detección en cascada.

El método de cascada para la detección de imágenes, consiste en tomar solo fragmentos de los datagramas y analizarlos por separado, se tiene en consideración que los videos constan

de segmentos que se repiten a lo largo de algunos segundos, esta disgregación de análisis por segmentos, disminuyó el nivel de procesos y aumentó la velocidad de detección. En la Figura 3 se describe la detección en cascada:

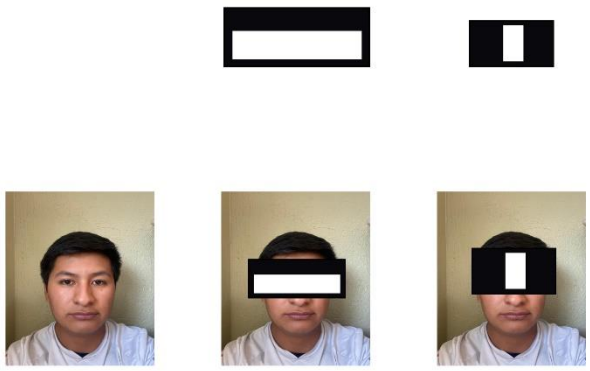


Figura 3: Detección en cascada de imágenes

Fuente: elaboración propia

Clasificación simple

La clasificación de imágenes en método simple, es un tipo de secuencia que tomó a la detección en cascada como base y la utilizó para el análisis en nodos que asemejan a la estructura de árbol. Al tener presente que una imagen se divide en segmentos de datagrama, el algoritmo de clasificación simple selecciona las diferencias primordiales del objeto a detectar (caras y no caras o parte superior del cuerpo y no parte inferior del cuerpo) dentro de un datagrama de video.

El determinar qué tipo de características se van a analizar y cuáles no, son una parte imprescindible dentro de los sistemas de detección artificial, el problema se encuentra a la hora de distinguir cuales son los aspectos que pertenecen a un objeto. Para ello se realiza una serie de variaciones y mediciones en el objeto llamadas: “características”, para que el sistema pueda relacionar las características halladas y encontrar una semejanza.

1.2.3. Herramientas de visión artificial

Las herramientas de visión artificial que se han generado a lo largo del tiempo son variadas y muchas de ellas constan de diversos algoritmos que han sido empaquetados en librerías y utilizados para diversos proyectos, se presentan a los siguientes como las opciones más estándar en el mercado:

OpenCv

Es un conjunto de algoritmos encapsulados en una biblioteca de visión artificial y aprendizaje automático “Open Source” conocido como código abierto. OpenCV fue creada para dar estructura estándar entre software de visión por computadora y aumentar el tiempo de proceso de las imágenes en productos comerciales. Al ser una librería que cuenta con una licencia Berkeley Software Distribution (BSD), la cual se da al software que sea de libre propiedad, OpenCV tiene como principal característica el brindar a las empresas su uso libre para cualquier modificación pertinente.

Esta librería cuenta con un aproximado de 2500 algoritmos programados, en los que se encuentran proyectos de visión por computadora y aprendizaje autónomo en ambientes antiguos y actuales. El uso de estos algoritmos y su propagación por ser Open Source, ha hecho que evolucione y agreguen características nuevas cada generación, entre las funciones principales se encuentran la detección de objetos, caras, acciones humanas, rastreo de objetos, modelar y reproducir nubes, entre otros muchos proyectos agregados por los programadores. Junto con grandes empresas como lo son: Yahoo, Intel, IBM, SONY, Google, Microsoft, AMD y Toyota; hacen uso abierto de la biblioteca con todas las extensiones que brinda OpenCV.

Dlib

Dlib es un kit de herramientas moderno de C++ que tiene algoritmos de aprendizaje automáticos y funciones para crear proyectos complejos en C++ para resolver problemas del mundo real. Se utiliza tanto en la industria como en la rama académica, en una serie de proyectos como: robótica, telefonía. Microchips integrados, y entornos informáticos de alto rendimiento. Al ser una librería de código abierto, es permitido usarlos en proyectos de forma gratuita.

A diferencia de muchos proyectos de código abierto, este proporciona una guía completa para cada proceso programado, también ayudando con métodos de depuración que verifican las condiciones de las funciones, detectando de esta manera la gran mayoría de los errores causados por llamar a las funciones erradamente o usar objetos de manera incorrecta. Entre las funcionalidades que incorpora están:

- Buena cobertura de prueba unitaria.
- Consta de una robusta biblioteca que es testeada con regularidad en sistemas operativos basados en: Windows, Linux y Mac OS X.
- Es independiente de bibliotecas externas para su uso. Solo se necesitan las API que proporcionan un sistema operativo listo para usar.
- No es necesario instalar ni configurar los pasos antes de poder usar la biblioteca.

OCRMax

La librería OCRMax es una herramienta de identificador de textos que ofrece la facilidad de búsqueda de topografías que han sido subidas a algún servidor en internet, haciéndola uno de los sistemas predilectos para empresas con una gestión constante de productos masivos. Entre los proyectos que sobresale OCRMax se tienen a los mercados de automoción, alimentos y bebidas, productos de consumo básico, medicamentos y productos farmacéuticos, partes electrónicas y de servicios postales. El método que utiliza esta librería se la llama In-Sight Explorer, la cual facilita un conjunto de algoritmos prediseñados en la detección de textos y que ofrece un alto nivel de confianza en la lectura y la verificación topográfica con distintos niveles de textos digitales y algunos de documentos reales, un ejemplo se presenta en la Figura 4, donde se muestra el exhaustivo mejoramiento de las mecánicas que componen los algoritmos de detección (Marketingbcnvision, 2014).

Figura 4: OCRMax imagen degradada

Fuente: (Marketingbcnvision, 2014)

MatLalb

Se define a MATLAB como un lenguaje para la informática técnica. Integra la computación, la programación y la visualización en un entorno fácil de usar donde los problemas y las soluciones se expresan en cálculos matemáticos. Entre sus usos más comunes están:

- Computación y matemáticas.
- Creación de algoritmos.
- Creación de prototipos, simulación y modelado.
- Visualización de datos, análisis y exploración.
- Gráficos de ingeniería a nivel científico.
- Implementación de interfaces gráficas para el usuario.

Se le considera a MATLAB como un sistema cuyas características son un conjunto de datos básicos en una matriz que no requiere dimensionamiento. Esta adaptabilidad es una fortaleza para resolver proyectos informáticos técnicos, en especial las que tienen funciones en vectores y matrices, esto en un tiempo más bajo que sus contrapartes como: C o Fortran.

MATLAB se escribió originalmente para proporcionar un rápido acceso a código segmentado en matrices de proyectos como: LINPACK y EISPACK, que son los primeros en aparecer usando algoritmos matriciales.

Desde su creación hasta la actualidad, MATLAB ha pasado por los escritorios de muchos usuarios, evolucionando su código base. En ámbitos académicos, es una materia estándar para la introducción a cursos de ingeniería, matemáticas y ciencias.

LabView

El nombre LabVIEW es una forma abreviada de su descripción: Laboratorio de instrumentos virtuales de ingeniería Workbench. LabVIEW es un lenguaje de programación visual, es una plataforma de diseño de sistemas y un entorno de desarrollo que tenía como objetivo permitir que se utilicen todas las formas de sistemas.

LabVIEW fue desarrollado por National Instruments como un banco de trabajo para controlar la instrumentación de prueba. Sin embargo, sus aplicaciones se han extendido mucho

más allá de la simple instrumentación de prueba a todo el campo del diseño y operación del sistema.

Este lenguaje es esencialmente un entorno que permite la programación en G: este es un lenguaje de programación gráfico creado por National Instruments que se desarrolló inicialmente para comunicarse a través de GPIB, pero desde entonces se ha actualizado considerablemente. Hoy en día G se puede utilizar para aplicaciones de prueba automatizadas, adquisición de datos generales, programación de FPGA, etc.

1.3. Microrordenadores

Una microcomputadora es una computadora con una unidad central de procesamiento como microprocesador. Diseñado para uso individual, más pequeña que una computadora central o una minicomputadora.

El término microordenador no se usa con tanta frecuencia como en los años setenta y ochenta. Ahora se refiere a las microcomputadoras como, simplemente, computadoras de tamaño pequeño que incluyen las funciones de una computadora de tamaño normal, pero con un espacio más reducido.

Las microcomputadoras personales a menudo se usan para educación y entretenimiento. Más allá de las computadoras portátiles y de escritorio, se pueden incluir consolas de videojuegos, electrónica computarizada y teléfonos inteligentes.

En el lugar de trabajo, las microcomputadoras se han utilizado para aplicaciones que incluyen procesamiento de datos y palabras, hojas de cálculo electrónicas, presentaciones profesionales y programas de gráficos, comunicaciones y sistemas de gestión de bases de datos. Se han utilizado en negocios para tareas tales como contabilidad, inventario y comunicación; en entornos médicos para registrar y recuperar datos de pacientes, administrar planes de atención médica, completar el cronograma y procesar datos; en instituciones financieras para registrar transacciones, rastrear la facturación, preparar estados financieros, nóminas y auditorías; y, en aplicaciones militares para dispositivos de entrenamiento entre otros usos.

1.3.1. Antecedentes

El término microordenador se remonta a la década de 1970. La llegada del microprocesador Intel 4004 en 1971, y más tarde el microprocesador Intel 8008 e Intel 8080 en 1972 y 1974 respectivamente, siendo el camino hacia la creación de la microcomputadora.

El primer microordenador fue el Micral, lanzado en 1973 por Réalisation d'Études Électroniques (R2E). Basado en el Intel 8008, fue la primera computadora sin kit basada en un microprocesador. En 1974, el microordenador MCM / 70 basado en Intel 8008 fue lanzado por Micro Computer Machines Inc. (más tarde conocido como MCM Computers).

Aunque lanzado después del Micral y MCM / 70, el Altair 8800 a menudo se considera el primer microordenador comercial exitoso. Lanzado en 1974, fue diseñado por Micro Instrumentation Telemetry Systems (MITS) y se basó en el microprocesador Intel 8080. Se vendió por alrededor de $400 en forma de kit, $600 ensamblados ($2,045 y $3,067 en dólares de 2018, respectivamente).

A medida que el diseño del chip del microprocesador maduró, también lo hizo la capacidad de procesamiento de los microordenadores. En la década de 1980, las microcomputadoras se usaban para algo más que juegos y recreación basada en computadora, encontrando un uso generalizado en computación personal, estaciones de trabajo y academia. En la década de 1990, las microcomputadoras se producían como asistentes digitales personales (PDA) de bolsillo, y luego llegaron en forma de teléfonos celulares y reproductores de música portátiles.

1.3.2. Tipos de microrordenadores

Raspberry pi

La Raspberry Pi es un pequeño computador con un costo bajo y de un tamaño no mayor al de una tarjeta bancaria que está diseñada para ser conectada a un monitor y utilizar periféricos normales como mouse o teclados. Es una pequeña computadora que se utiliza para iniciar en programación desde cualquier edad, en los lenguajes de Scratch y Python. La potencia almacenada con un sistema operativo ligero, le da la capacidad de realizar cualquier función que lo haría una computadora normal, incluyendo la navegación por internet, reproducción de

música o video, procesamiento de texto e incluso correr algunos juegos de video. Al poseer todas estas funciones en un tamaño reducido, tal como se describe en la Figura 5, se ha visto potenciado por proyectos académicos en una gama alta en creación digital, estaciones con cámaras de seguridad, cajas multimedia de reproducción y emulación de consolas de videojuegos. La utilización de este microrordenador se ha visto direccionada como introducción a la informática en niños y jóvenes (RASPBERRY PI FOUNDATION, 2020).

Figura 5: Microrordenador Raspberry pi

Fuente: (RASPBERRY PI FOUNDATION, 2020)

Pine H64

PINE H64 es la última computadora de placa única de Pine64 la cual está alimentada por un procesador Allwinner "H6" Quad-Core ARM Cortex A53 de 64 bits con GPU MALI T-722. El PINE H64 está equipado con una memoria de sistema LPDDR3 PC-1600 de hasta 3GB y Flash de arranque SPI de 128Mb. También hay un módulo opcional eMMC (hasta 128 GB) y una ranura microSD para el arranque. La placa está equipada con 11n Wifi / BT incorporado, 1x USB 3.0 Host, 2x USB 2.0 Host, Gigabit Ethernet, PI-2 GPIO Bus, Euler GPIO Bus, así

como muchas otras interfaces de dispositivos periféricos como UART, SPI , I2C. El último modelo se presenta en la Figura 6 (PINE H64, 2019).

Figura 6: Microrordenador Pine H64

Fuente: (PINE H64, 2019)

ROCK PI

ROCK PI es una computadora de placa única (SBC) con pequeñas dimensiones, lo cual ofrece una demanda con competidores al tiempo que aprovecha al máximo sus especificaciones. ROCK PI brinda tanto a fabricantes como a aficionados, una plataforma confiable y extremadamente capaz para construir y ajustar las ideas a la realidad.

ROCK Pi 4 es un SBC (Single Board Computer) basado en Rockchip RK3399 de Radxa. Puede ejecutar Android o algunas distribuciones de Linux. ROCK Pi 4 presenta un procesador ARM de seis núcleos, LPDDR4 3200Mb/s de doble canal de 64 bits, señal de HDMI, MIPI DSI, MIPI CSI, conector de 3.5 mm con micrófono, 802.11 ac WIFI, Bluetooth 5.0, Puerto USB, GbE LAN, Encabezado de expansión de color de 40 pines, RTC. Además, ROCK Pi 4 admite alimentación por USB PD y QC.

ROCK Pi presenta opciones de expansión amigables para el fabricante, incluida una interfaz GPIO de 40 pines que permite la interfaz con entradas de rango de botones, interruptores, sensores, LED y mucho más. ROCK Pi también cuenta con una LAN Gbit para

la red, con bus y controlador dedicados, funciona sin latencia en aplicaciones de red de carga pesada. El wifi 802.11 ac integrado ofrece conectividad WLAN 2.4G y 5G. Con Bluetooth 5.0, ROCK Pi mejora la velocidad de Bluetooth y ofrece un mayor alcance. ROCK Pi también cuenta con un host USB 3.0 y un puerto USB 3.0 OTG, cada uno de 5 Gbps/s, que funciona de forma independiente. El USB 3.0 OTG puede funcionar como un dispositivo USB como Android ADB o dispositivos USB. Su estructura de hardware se la presenta en la Figura 7 (Rockpi, 2019).

Figura 7: Microrordenador RockPi

Fuente: (Rockpi, 2019)

Le Potato AML

Le Potato es nuestra plataforma de hardware insignia con soporte para la última versión de Android 9 / TV, upstream Linux, u-boot, Kodi y más. Basado en una familia SoC compatible a largo plazo con un registro comprobado de implementaciones masivas por parte de los proveedores de contenido regionales más grandes, Google y Amazon, AML-S905X-CC es la plataforma de desarrollo perfecta para proyectos que requieren CPU de clase ARM Cortex-A de alto rendimiento, entrega y reproducción de medios 4K seguros y no seguros, Widevine CAS DRM, alta confiabilidad y baja potencia.

Libre Computer es el único proveedor de soluciones con experiencia clave en el diseño de hardware y la pila de software libre de código abierto (FOSS), tal cual se lo muestra en la Figura 8. Para familia SoC presenta el Amlogic Video Engine 10 (AVE10) capaz de manejar transmisiones H.265, H.264 y VP9 con metadatos HDR, características que lo hacen ideal para la señalización digital 4K en Linux principal y el último Android 9.0 Pie (Hernandez et al., 2020).

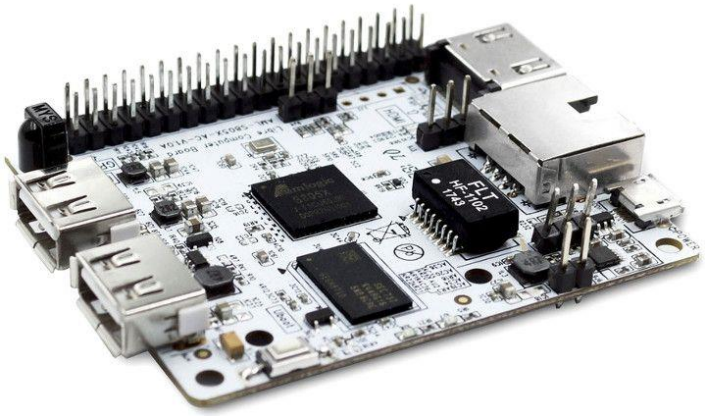


Figura 8: Microrordenador Potato AML

Fuente: (Libre Computer, 2020)

CAPÍTULO 2

MARCO METODOLÓGICO

Este proyecto está desarrollado con metodologías de investigación basadas en la observación y recolección de datos científicos, los cuales darán como resultado diversos parámetros que se utilizarán para un análisis de posibles soluciones ante el desgaste que produce la somnolencia en conductores.

Tipos de Investigación

Se pueden encontrar diversas características de investigación que pudieren ser aplicados e implementados en el estudio de una investigación como tal, para el caso del presente, existen dos parámetros o factores que determinan su desarrollo, estos son: la profundidad del estudio y el referido a las fuentes de consulta. Por lo cual se plantea lo siguiente:

2.1.1. De acuerdo con la profundidad del estudio

De acuerdo a la parte referida a la profundidad del estudio, se ha podido deducir que el método a utilizar es el Correlacionar, ya que este evalúa las similitudes que tienen dos o más variables, según el comportamiento de estas antes mencionadas, por lo que se pudo establecer un buen nivel de concordancia de esta metodología con la temática a la cual está referida el proyecto, es por ello que se llegó a determinarlo como el más idóneo para tomarlo a manera de referencia. (Hernandez et al., 2020)

2.1.2. De acuerdo con las fuentes de consulta

Ya que la fuente de consulta es un punto sobresaliente dentro del desarrollo del proyecto, se escogió como método a la investigación bibliográfica, la cual plantea, generar resultados mediante la recolección proveniente de libros, documentos escritos, etc., los cuales servirán

como referencia y fuente principal para el desarrollo de los temas inmersos dentro del proyecto investigativo.

Diseño de investigación

Este proyecto es de carácter experimental porque tiene un mayor énfasis y atención a la formulación y tratamiento de la hipótesis mediante procedimientos básicamente educativos, orientados en el procesamiento de datos informativos y los parámetros bajo los cuales esta establecidos el desarrollo del proyecto. Además, es de carácter documental porque la información requerida ha sido extraída de documentos como libros, folletos, revistas, entre otros. (Normas Apa, 2016)

Métodos

Existen algunos métodos que se pueden incluir dentro de la investigación, en el caso actual se tiene a los siguientes:

2.3.1. Método Inductivo

Este método de investigación plantea que es el estudio de fenómenos particulares para llegar al conocimiento, es por ello que se llegó a concluir que este método se apega a varios puntos dentro de la investigación.

2.3.2. Método Deductivo

Dado que existen ciertas medidas que necesitan ser estudiadas desde un contexto general y se debe descifrar como elementos separados pertenecientes a estos parámetros, se decidió establecer al método deductivo como un método de elaboración en el proceso del proyecto.

2.3.3. Método de Síntesis

Este método plantea un único proceso que debe llevar desde lo más simple para llegar a lo más complejo, por lo tanto, se concluye que este método beneficiará en el desarrollo de esta investigación.

2.3.4. Métodos Estadísticos

La definición de este método dice que la estadística se puede utilizar como una herramienta, la cual facilita la realización de tabulación y análisis de los datos para transformarlos en información para que sea de aquí de donde se extraigan resultados, conclusiones y recomendaciones; por lo cual toda la información recopilada va a entrar a un análisis de tabulación de los datos donde el método estadístico, sus funcionalidades y herramientas ingresan a formar parte del proyecto.

2.3.5. Método de Análisis

Es una parte primordial para la estructuración del proyecto planteado, tomando como su definición que este método habla de tomar un todo general y lo divide en partes, para revisarlas de manera minuciosa e identificar partes de interés que puedan generar información valiosa en el desarrollo de los temas inmersos dentro de la investigación.

2.4. Población y muestra

2.4.1. Población

La población se refiere al grupo de incidencias dentro del objeto del cual se va a hacer referencia para poder sacar los datos necesarios a ser analizados, para esto se muestra en la Tabla 2 las características de este grupo seleccionado.

El universo al que se hace referencia sobre la población de estudio, estará dirigido a una sesión de 100 pruebas realizadas en un entorno real con el prototipo desarrollado, donde se hará un sondeo de alertas de sueño, con un registro de casos positivos y posibles negativos.

Se analizarán en diferentes horas del día para poder tener un margen de datos más amplio y poder ofrecer un análisis que abarque todas las posibles incidencias en las que el prototipo pueda trabajar. Teniendo la variación de somnolencia según los parámetros presentados en el estudio.

2.4.2. Muestra

Tabla 2: Muestra de estudio

Población y Muestra	
Grupo Objetivo	Incidencias de somnolencia en un conductor
Zona Geográfica	Distrito metropolitano de Quito
Método de Investigación	Inductivo
Metodología	Observación y documentación
Tipo de recolección de datos	Personal
Zona objetiva a implementar en el análisis de resultados	100 incidencias en varias horas del día.

Fuente: elaboración propia

2.5. Técnica para análisis de datos

Es necesario dar un análisis completo a los resultados que se pretenden conseguir dentro del proyecto, realizando un seguimiento del prototipo en diversas ramas, lo cual dará datos que se pueden procesar y sacar un análisis completo que ayudará en concluir con los objetivos presentados. Por ello, en la Tabla 3 se presentan las siguientes preguntas a realizar cuando se esté en el capítulo de análisis de resultados:

Tabla 3: Preguntas de investigación

Pregunta 1: ¿En cuánto tiempo se realizó la detección de somnolencia?
Pregunta 2: ¿Las alarmas son lo suficientemente adecuadas para poner en alerta al conductor?
Pregunta 3: ¿El conductor tuvo algún problema para manejar el prototipo?
Pregunta 4: ¿Se ha tenido inconvenientes en encontrar un ángulo de visión dentro del vehículo?
Pregunta 5: ¿Ha sido incómodo para el conductor el manejar con el prototipo instalado?
Pregunta 6: ¿Se ha reducido la incidencia de estado de somnolencia en el conductor?

Fuente: elaboración propia

Cada pregunta dentro de la encuesta tiene la finalidad de análisis y conocer si la investigación está correctamente encaminada al objetivo principal, estos objetivos se los presenta en la Tabla 4.

Tabla 4: Objetivos de preguntas

Pregunta 1: El tiempo de latencia es importante para tener en cuenta si el prototipo tiene la capacidad de realizar un análisis en tiempo real.
Pregunta 2: El poder saber si el usuario al volante puede escuchar o ver las alarmas y ponerse en alerta nuevamente, es fundamental para coincidir con el objetivo de disminución de somnolencia.
Pregunta 3: Que el usuario final pueda manejar de manera óptima el prototipo desarrollado, es una tarea básica para que lo pueda poner en funcionamiento.
Pregunta 4: Para poder saber si se tiene un cansancio por falta de sueño, el ángulo donde está la cámara debe ser idóneo, caso contrario no podrá detectar los rasgos del conductor.
Pregunta 5: La respuesta esperada es que el conductor pueda tener al prototipo en todo lado al que vaya, por lo tanto, se tiene que saber si le causa alguna incomodidad en el trascurso de una jornada normal de manejo.
Pregunta 6: Nuestro resultado final es el poder disminuir en algún grado el estado de somnolencia del conductor.

Fuente: elaboración propia

Para la tabulación de datos en la primera pregunta se utilizará los métodos me medida de control central, siendo relevante el tener un rango exacto del tiempo de demora que hace el prototipo para detectar somnolencia y enviar un resultado con las alarmas visuales y auditivas.

La **media** que es la tendencia central más utilizada y se la conoce por el nombre de promedio, se calcula mediante la suma de todos los casos dividido por el número total de

repeticiones, siendo uno de los métodos más precisos para detectar el valor central en una agrupación de datos. La **mediana** es el valor o grupo de valores que se encuentran en el centro de un conjunto de datos, sin importar el rango o repeticiones que tengan los mismos, esta medida de control ayuda para el análisis centralizado de datos. La **moda** se trata del conjunto de valores que se encuentra con más frecuencia en intervalos de datos, sirve para un análisis de repeticiones y sus complicaciones se pueden dar cuando existen más de un conjunto de datos con el mismo valor.

CAPÍTULO 3

PROPUESTA

Introducción

La presente propuesta da la solución al problema de accidentes que se puedan producir por un estado de falta de sueño en conductores mediante un prototipo que pueda detectar cuando el usuario entra en este estado, dando una alerta sonora y visual en modo de alarma, para que la persona al volante tome las medidas pertinentes y evite posibles accidentes. Si se quiere detectar los síntomas que producen la somnolencia, es necesario basarse en los distintos rasgos que produce el rostro como: cabeceo y parpadeo; cuando la persona pestañea suele llevarse las manos hacia los ojos o bien permanecer con los ojos cerrados y cuando cabecea, la persona lleva su cabeza hacia adelante.

El sistema empleado para el desarrollo de esta propuesta fue un módulo *Raspberry* pi en versión 3, la cual se le debe incorporar una cámara que captará las imágenes del conductor; posteriormente para pasar todas las imágenes tomadas en tiempo real hacia el lenguaje de programación *Python*, el mismo que utilizará la librería de detección de visión artificial llamado *OpenCV*, que procesará los datos enviados y mediante algoritmos creados específicamente para este proyecto. El sistema devolverá como resultado un dato que será procesado como negativo o positivo, dependiendo si el usuario que está conduciendo entra en un rango cerca de un 80% de somnolencia; en el caso de ser una respuesta acertada, la salida de datos se traducirá a un pitido constante, así como una alarma visual para alertar al usuario.

Señales de somnolencia

La fatiga y somnolencia es una de las causas principales de los accidentes de tránsito en las carreteras, gran parte de ellos con consecuencias fatales. Por eso, el análisis previo de los síntomas que producen somnolencia dará indicadores que se puedan utilizar para determinar

este estado. Se presenta a continuación las señalas que los conductores tienen cuando entran a un estado de somnolencia:

- **Parpadeo:** cuando los niveles energéticos de una persona se encuentran en óptimo nivel, su parpadeo dura menos de un segundo. Si tarda más es posible que el individuo se encuentre cansado y con algo de somnolencia, por otra parte, si la duración es dos segundos o más, es un signo de que se encuentra fatigado y con sueño atrasado, por lo que es posible que sufra un micro sueño en cualquier momento.
- **Estado de ánimo:** un conductor que presenta un inicio de somnolencia, se ve agotado, se muestra callado, distante e incluso un poco irritable. De igual modo se mostrará torpe y lento al moverse, además de frotarse la cara y ojos constantemente.
- **Lenta reacción:** alguien cansado y con sueño tiene una capacidad de reacción mínima. La fatiga y somnolencia en las personas hacen que tengan problemas de concentración y pensamientos desorientados
- **Visión borrosa:** la fatiga puede producir visión borrosa e incluso dificultades para enfocar objetos en el campo visual.
- **Fatiga general:** los conductores que presentan un desgaste de sueño, tienen como anomalías una baja en la presión cardíaca, parpadeo lento, bostezos o cabeceo constante.

2.1.1. Cálculo del índice de somnolencia

La excesiva somnolencia diurna (ESD) es un fenómeno que cada vez se viene manifestando con mayor frecuencia en aspectos cotidianos y, por ende, deriva en una mayor demanda de atención especializada. Según el estudio realizado por la *American Sleep Disorders Associationse* (ASA), estima que entre el 10 y el 20% de la población padece de trastornos de somnolencia en actividades rutinarias. Para poder detectar este tipo de patología la ASA propone dos métodos presentados en la Tabla 5, que son necesarios tener como técnicas que ayuden a la detección de somnolencia (AMERICAN SLEEP ASSOCIATION, 2019):

Tabla 5: Índice de somnolencia

FACTOR	PERCLOS	HORN Y DONG
TEORÍA	El método que utiliza es que, si se sobrepasa el 80% de síntomas de somnolencia, se detecta como posible alerta.	El método utilizado se refiere a que si existen más de 5 fotogramas consecutivos donde se detecte que los ojos están cerrados, se lo considera como posible somnolencia.
NÚMERO DE PRUEBAS	100	100
DETECCIÓNES	91 (91%)	92 (92%)
FALSOS POSITIVOS	2 (5%)	6 (6%)
FALSOS NEGATIVOS	7 (8%)	2 (2%)

Fuente: Elaboración propia

La técnica que se utilizará para la detección de somnolencia vendrá ligara a la exactitud del método estudiado, en donde se puede apreciar que a pesar que el método HORN Y DONG tiene un porcentaje mayor, está ligado al proceso de una secuencia de fotogramas mínima. De igual modo se debe observar el número de falsos positivos y negativos que tienen ambos métodos, para poder dar como resultado que la técnica de PERCLOS consta de un margen menor de datos posiblemente errados en los falsos positivos. Por estos motivos la metodología a escoger será la de PERCLOS.

Esquema de la propuesta

Mediante un esquema integral presentado en las Figuras 9 y 10, se detalla de manera visual como es el seguimiento del prototipo para detectar un estado de somnolencia en conductores, donde se divide en dos partes: los requerimientos de hardware, que serán los dispositivos físicos que se utilizarán para construir el prototipo y; los implementos de software, que serán los programas, sistemas y librerías que se implementarán para la programación de los algoritmos de reconocimiento de somnolencia y se devolverán los resultados en modo de alerta.

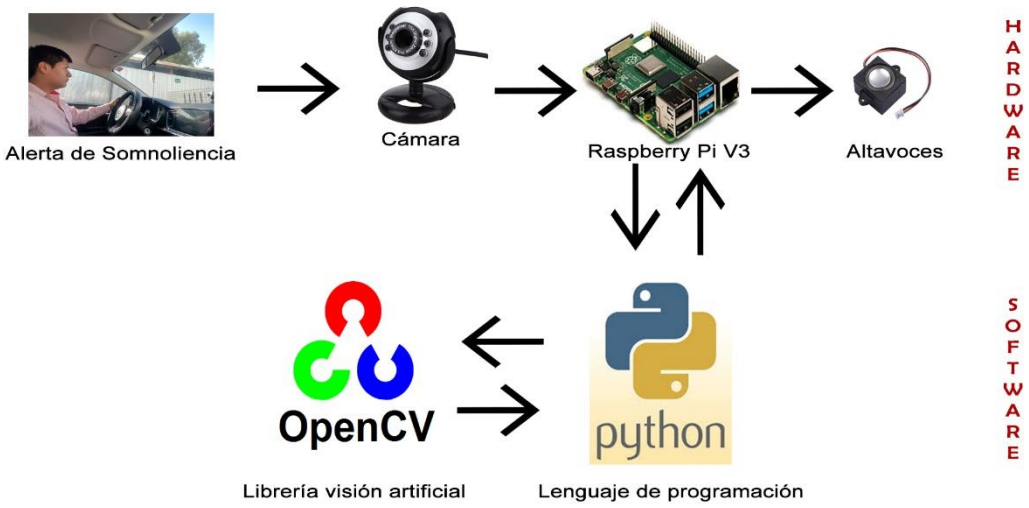


Figura 9: Esquema integral de la propuesta.

Fuente: elaboración propia

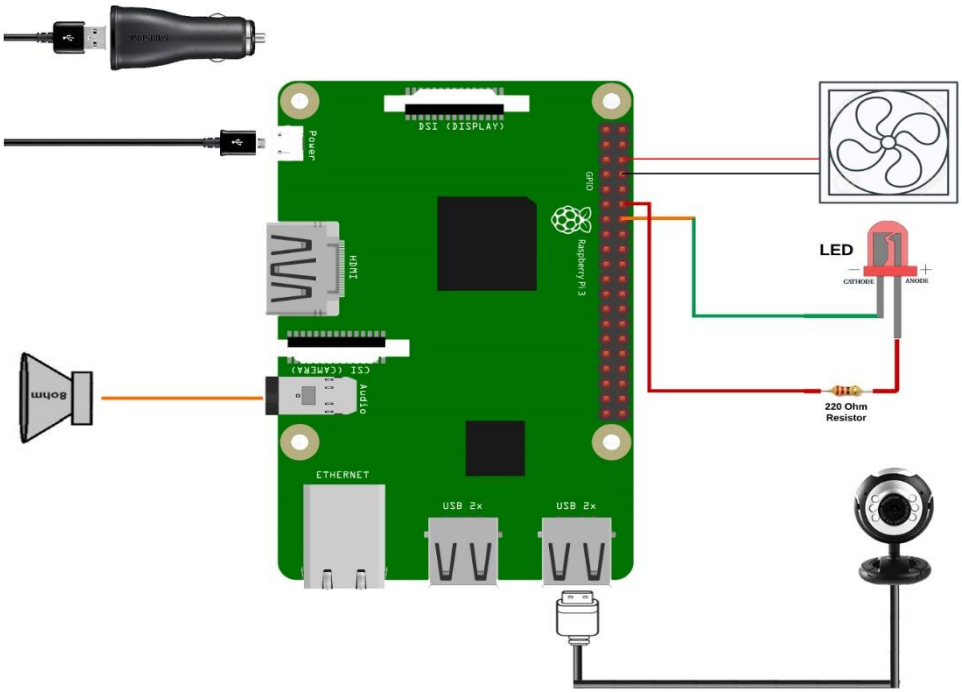


Figura 10: Diagrama de conexión

Fuente: elaboración propia

2.1.2. Raspberry Pi

Es considerado un microrordenador de bajo coste, también conocido como una mini placa de computadora (SBC), que ofrece las funcionalidades de una placa normal a un tamaño más reducido. Fue desarrollado en los países del Reino Unido por la universidad de Cambridge en el 2011 con el nombre de: *RaspBerry PI.* El principal objetivo del proyecto fue tener un prototipo en miniatura de una placa de computadora que pueda ser utilizada a bajo coste y así estimule la enseñanza de la informática en escuelas y centros de estudio. Las dimensiones de esta placa la cual empezó su comercialización desde el año 2012, es de 85 x 54 milímetros y empezó trabajando con un chip Broadcom BCM2835 el cual contaba con procesador ARM de hasta 1 GHz de velocidad, GPU VideoCore IV y hasta 512 Mb de memoria RAM.

El microrordenador más actual en su versión 3 B+, que se muestra en la Figura 11, cuenta con las siguientes características:

- Chip central "Broadcom BCM2835", con una velocidad de 700 MHz, el cual puede tener una sobre potencia de hasta 1GHz.
- Un procesador gráfico (GPU) VideoCore IV
- Una memoria RAM de 512Mb
- Un puerto RJ45 para una velocidad de internet de hasta 10/100 Mbps
- 2 puertos USB 2.0
- Un conector Jack de 3.5mm que se intercala como salida de audio y entrada de micrófono.
- Salida de video digital HDMI
- Salida analógica de video RCA
- Pines de entrada y salida para uso general
- Conector de alimentación microUSB
- Lector de tarjetas SD

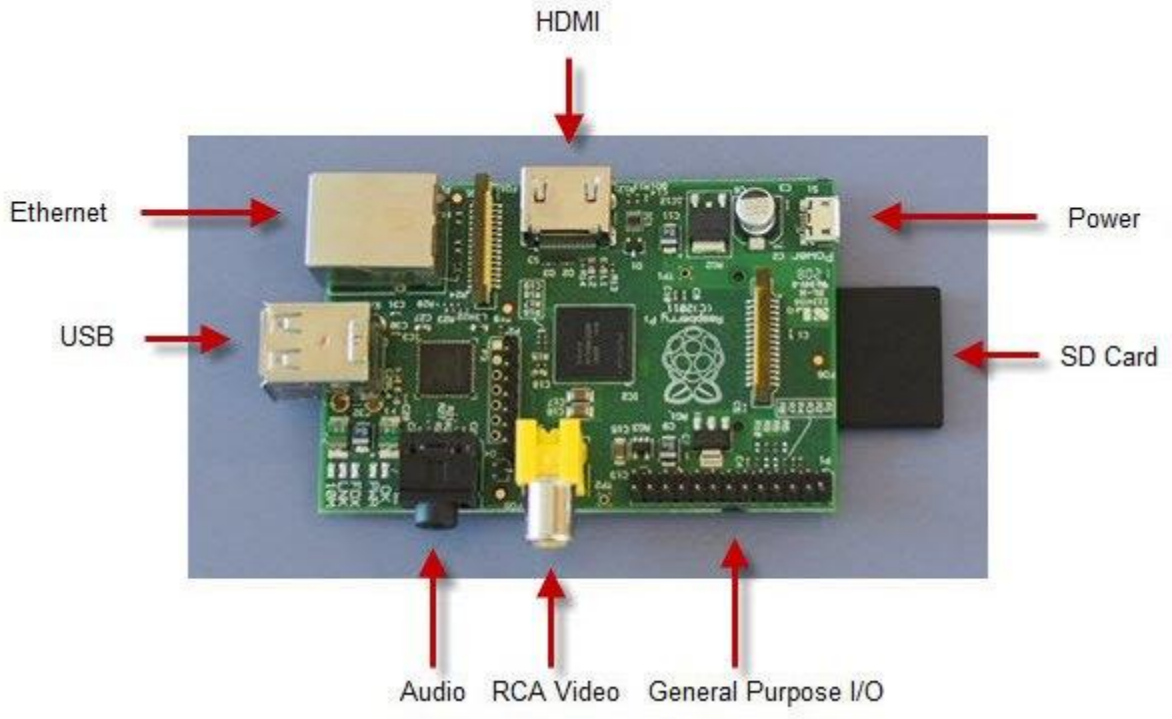


Figura 11: Esquema raspeberry pi

*Fuente(*https://histinf.blogs.upv.es/2013/12/18/raspberry-pi*)*

El diagrama GPIO presentado en la Figura 12, muestra los puertos que se pueden usar como elementos electrónicos, donde los pines 4 y 6 se utilizan para la conexión del ventilador; se toma en cuenta que los pines 12 y 14 se usan para la instalación de la salida LED (RASPBERRY PI FOUNDATION, 2020).

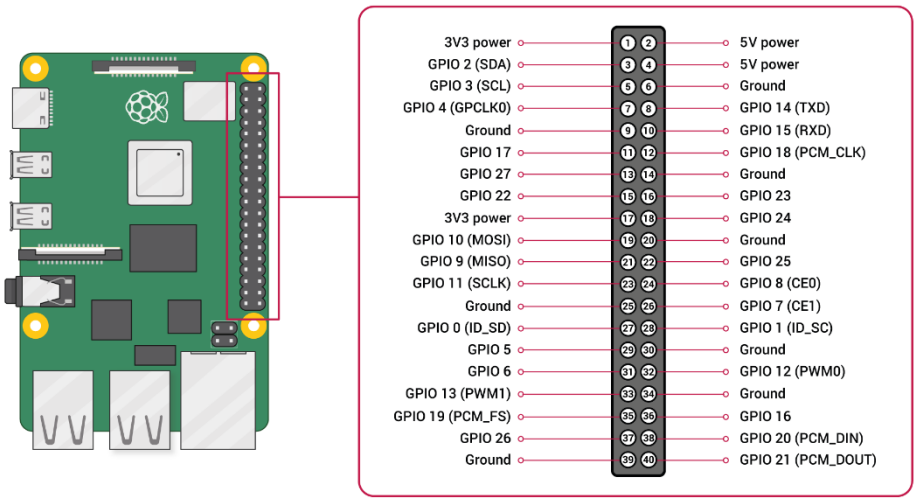


Figura 12: Diagrama de conexión de elementos

Fuente: (RASPBERRY PI FOUNDATION, 2020)

2.1.3. Cámara

Se tiene en el mercado un gran número de cámaras que se pueden integrar en los microprocesadores *Raspberry*, siendo los sistemas contra robo y atraco los más comunes en su contra parte con funciones complementarias como mecanismos contra incendios, anti-hurtos, especiales, etc, Estos sistemas ofrecen al usuario una visualización en tiempo real del área donde se instala la cámara. Una vez conectada la cámara a una red de datos es fácilmente configurable mediante un acceso Web. En un programa de video de vigilancia se considera un servidor de streaming el cual se encarga de recibir un flujo de datos (video) y distribuirlo entre los clientes conectados a este.

Cámara genérica

Un módulo que se puede integrar al microprocesador, es el de una cámara en Full HD, la cual puede tomar capturas de imágenes en movimiento a una resolución de 1080p con una calidad de 8MP (Megapíxeles). Este dispositivo que se puede instalar sin mayor inconveniente dentro de Raspberry pi y que es compatible con el sistema operativo *Raspbian*, mismo que viene por defecto como mejor opción para instalar al microrordenador que se hace mención. Dentro del puerto CSI, que viene indicado y especializado en el microrordenador, se puede hacer la

instalación del periférico de video, mismo que cuenta con varios tamaños dependiendo de la distribuidora donde se haya adquirido, siendo los más estándar: 25 mm x 23 mm x 9 mm y con un peso de 3 gramos, haciendo de esta manera un dispositivo portátil que se integra perfectamente al prototipo a diseñar y se presenta en la Figura 13 (RASPBERRY PI FOUNDATION, 2020).

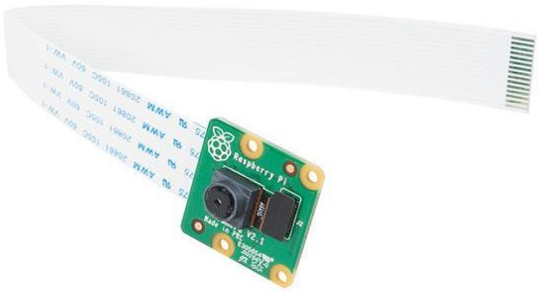


Figura 13: Cámara

Fuente: (RASPBERRY PI FOUNDATION, 2020)

2.1.4. Lenguaje de programación Python

Los lenguajes de programación son el medio por el cual los proyectos se plasman en ideas que puedan entender las máquinas, traduciendo el lenguaje normal que se implementa hacia los comandos que entiende un sistema de programación. Python cumple con estas características, al ser un lenguaje de escritura independiente de cualquier IDE o plataforma. Al ser un lenguaje que está orientado a objetos, es adaptable para realizar cualquier tipo de proyecto, sea este vía Windows, servidores, páginas web u otros. Este es interpretativo, lo que significa que no requiere de la compilación de las líneas de código para poder correr un proyecto, siendo robusto cuando se trata de proyectos con un alcance grande, controles de visión artificial y detección de imágenes.

Entre las ventajas que se tienen al usar este lenguaje, se pueden encontrar las siguientes:

- Existe una gran cantidad de librerías que han sido programas en código abierto y las cuales se pueden reutilizar, esto ayudará a encontrar una fuente de consulta y que no se tenga que empezar un proyecto desde cero.

- Al ser un lenguaje interpretativo, se pueden crear programas con menos líneas de código que sus contrapartes en JAVA o C.
- Por tratarse de un lenguaje que es de código abierto, se puede implementar en cualquier sistema operativo como: Windows, Linux, Unix y sistemas abiertos como lo es el mismo *Raspbian*.
- Es un lenguaje Open Source, que integra todas sus funciones de manera gratuita inclusive cuando se trata de macro proyectos, sean estos empresariales o para desarrollo de módulos complejos.

Al crear este lenguaje, se tuvo como objetivo el crear un sistema robusto de programación que no se vea limitado con un pago como su contraparte visual studio de Windows; generando así posibilidades de adaptar a multiplataforma como proyectos relacionados en web, estáticos o portátiles.

2.1.5. Librería OpenCv

Esta librería que da sus siglas por “Open Source Computer Vision” es un conjunto de algoritmos de código abierto y para uso gratuito en proyectos que tengan relación entre la visión artificial y software de machine learning, los cuales proveen una infraestructura con más de 47000 usuarios alrededor del mundo que mejoran a cada segundo los programas de la librería, ofreciendo así un software estable con más de 7 millones de descargas para uso profesional, empírico y académico. Esta librería cuenta con un recopilado de 2500 algoritmos, en los que se incluyen el reconocimiento facial por medio de imágenes estáticas e imágenes en movimiento; este paquete de programación permite identificar y clasificar de entre las imágenes que se les alimenta a objetos, rostros, personas, modelos en 3D y de igual forma íntegra funciones para su manipulación como extraer imágenes, enfocarse en una sola, eliminar imperfecciones, seguir el movimiento de ojos o boca y monitorizar los músculos que comprenden un rostro humano o artificial.

2.1.6. Detección de movimiento

Son cada vez más constantes las aplicaciones que utilizan la detección de movimiento basadas en inteligencia y visión artificial. En muchos de los casos se utilizan para proyectos que involucren el filtro de rostros humanos, sean para contar el número de personas en un determinado establecimiento, seguridades de alguna institución financiera, identificación facial, entre otros. Para todo este tipo de procesos, lo que se realiza es extraer el entorno que rodea a la o las personas que se pretende analizar, quitando el ruido u objetos alrededor que no son de interés para analizar.

Entre los varios métodos que se tienen para poder realizar un exacto filtro del movimiento que realiza un rostro humano, las técnicas y métodos que proveen librerías como OpenCv o Dlib, son estándares que se han puesto en marcha y perfeccionado con el pasar del tiempo y la manipulación de diversos usuarios que se han visto en la necesidad de aumentar el código para hacer de estas herramientas mucho más robustas.

Con una completa adaptabilidad de las librerías de detección de imágenes con un lenguaje que sea capaz de adaptar en tiempo real las órdenes que se les implanta, se puede realizar una sustracción de fondo optima y así quedar tan solo con las imágenes que se necesitan para hacer el proceso de detección de sueño que implica en este proyecto.

Sustracción de fondo

Esta técnica consiste en capturar una imagen o una sucesión de imágenes en movimiento y restar el ruido que se tiene en el fondo, esto se realiza separando cada fotograma en una individual y procesando el fondo que es de interés, para así obtener las imágenes de objetos o personas con las cuales se van a trabajar. Esto en tema de enfoque, se trata de separar el primer plano que se va a analizar y restarle el segundo plano o fondo, esta técnica se la percibe en la Figura 14 la cual está generando una interferencia en el procesamiento de imágenes o ruido (Luis del Valle Hernández, 2017).

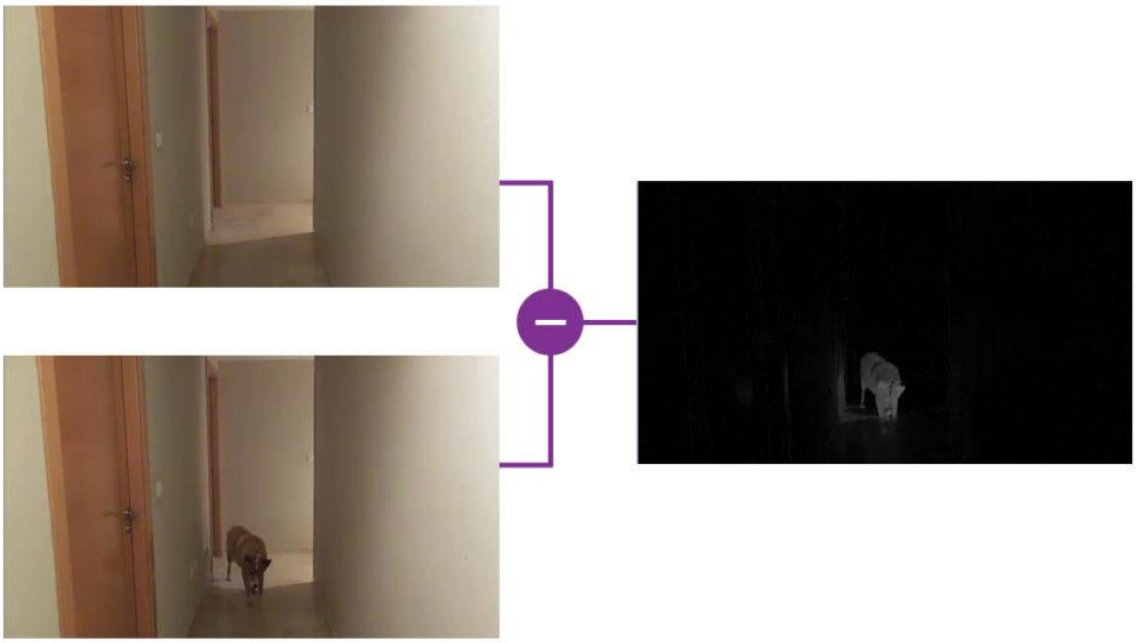


Figura 14: Detección de fondo

Fuente: (Luis del Valle Hernández, 2017)

Como resultado se tiene a una imagen con fondo diferente, de modo general el segundo plano se transforma en uno con fondo negro y el primer plano es del color real del fotograma. Esta técnica se ha incorporado en las librerías y no requiere de ningún trato en especial a las imágenes con las cuales se van a tratar, poniendo en deterioro a sensores que eran comunes ver para este tipo de detecciones. Por el lado de iluminación, cabe recalcar que los algoritmos presentes son sensibles dependiendo del tipo de cámara que se utilice, siendo más complicado el extraer fondos cuando se tratan de videos en modo nocturno, a pesar de implementarse cámaras en infrarrojos, aún sigue siendo un inconveniente la programación de sustracción en negros o fotos con poca iluminación. Por lo que es recomendable el utilizar los métodos de detección con luz, para que el fondo este adaptable a cambios de tono y saturación que serán óptimos a la hora de reducir el fondo y tener una imagen más limpia.

Sustracción con imagen en referencia

Se ha podido determinar que la sustracción de la imagen de fondo es vital para poder tener los planos necesarios que se van a utilizar en el análisis de movimiento, es por ello que se han creado algoritmos de detección constante de imágenes, las cuales capturan un fotograma y lo comparan con los demás para determinar si estos comparten un fondo similar, haciendo que el proceso de separar los planos sea dinámico y no se repitan con cada imagen a analizar.

Este proceso es muy sensible a los cambios de luz y de movimiento en un video, por lo que es recomendable tener una cámara estática en un lugar donde haya poca movilidad y una fluctuación baja de luz, se muestra un ejemplo en la Figura 15 donde se puede ver el tiempo de procesos que toma el analizar cada fotograma por separado, dando como resultado un aumento en la velocidad de detección (Luis del Valle Hernández, 2017).

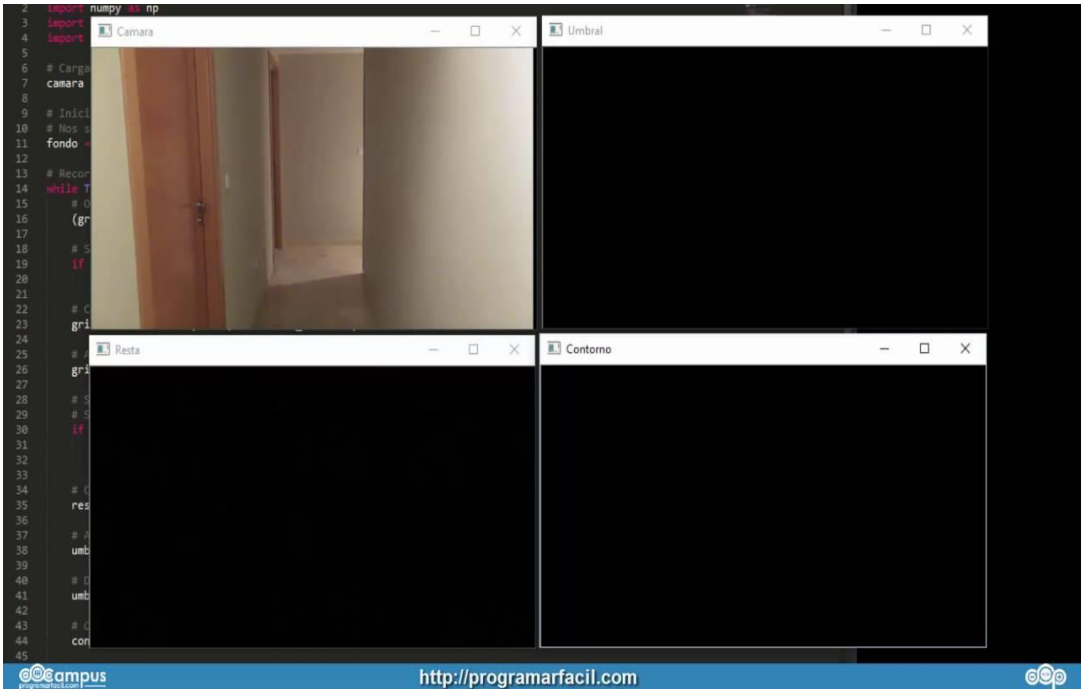


Figura 15: Sustracción con imagen en referencia

Fuente: (Luis del Valle Hernández, 2017)

Substracción con fotogramas anteriores

Esta técnica de sustracción de fondo es muy parecida con la anterior, la diferencia radica en que se toma como punto de anclaje únicamente al primer fotograma y se asume que los demás restantes van a coincidir con el primer fondo que se presentó. Este modelo de procesamiento hace que el proceso se vea disminuido drásticamente en tiempos de ejecución, sin embargo, es poco preciso por la movilidad que tienen las imágenes y ser fluctuantes en mucho de los casos tal cual se presenta en la Figura 16 (Luis del Valle Hernández, 2017).

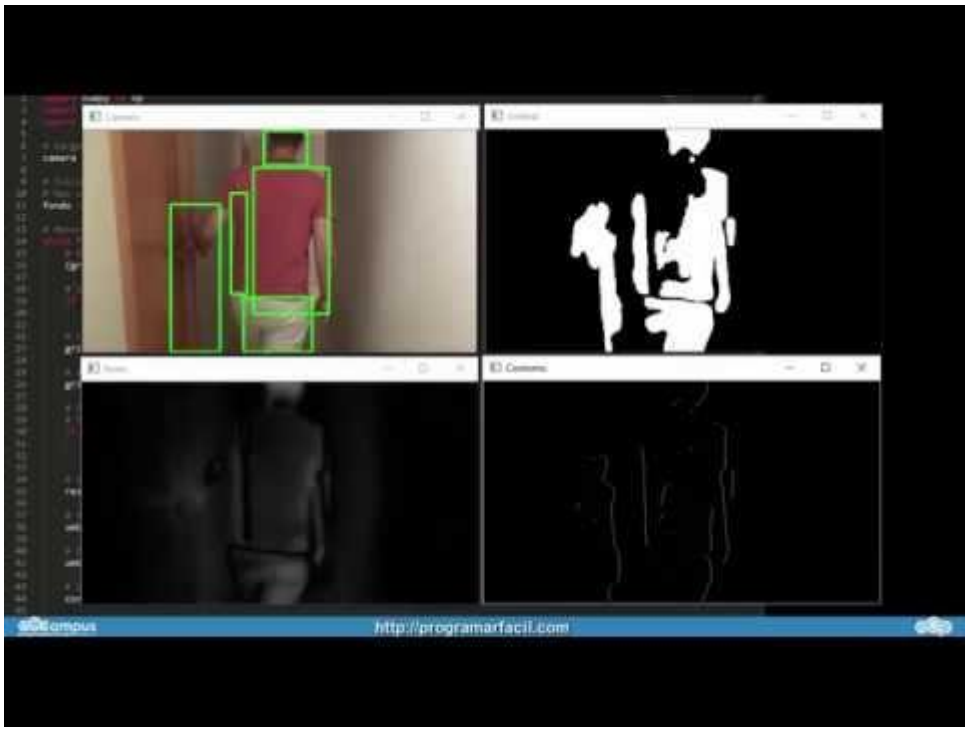


Figura 16: Substracción con fotogramas anteriores

Fuente: (Luis del Valle Hernández, 2017)

2.1.7. Fases del proceso de detección de movimiento

Una vez identificado el proceso que se realiza para un análisis estándar de imágenes estáticas o en movimiento, los siguientes pasos se proceden con la librería OpenCv para poder hacer una detección facial:

- Sustituir la imagen a una escala de grises para poder eliminar el ruido de fondo.
- Enfocar y sustituir el segundo del primer plano para poder trabajar con solo uno de ellos.
- Aplicar un filtro de umbral, que hará que se enfoque en uno de los planos seleccionado.
- Detección de los contornos, los cuales se denominan *blobs*.

Lo que se tiene que considerar son los parámetros que se deben incluir en la detección de visión artificial, ya que cada proyecto es único dependiendo de las especificaciones que tenga el mismo, del uso que se lo vaya a dar y del entorno donde este sujeto a ser implementado; de la misma manera las varias a ingresar deben optimizarse para dar el mejor rendimiento a la investigación en curso. Es por ello que no existen valores exactos o estándares cuando se trata de detección de un video en particular, las pruebas se las realiza en el transcurso del desarrollo del prototipo para ajustar el máximo rendimiento de los algoritmos implementados.

Conversión a escala de grises y eliminación de ruido

El primer paso para poder tratar cualquier tipo de imágenes, independientemente del proyecto o la finalidad que tengan las mismas, es la de transformarlas a una escala de grises; esto traerá un beneficio a la hora de separar los planos y analizarlos en forma independiente. Y es que, resulta menos complejo para los algoritmos utilizados en la detección, el procesar imágenes con una escala de grises, a diferencia de aquellas que están con la gama de colores completa.

Por otro lado, la eliminación de los efectos que provoca la iluminación y que se ve afectado por la misma cámara, es útil para tener una muestra más limpia y que el proceso de detección se vea beneficiado con una tasa baja de errores. Este proceso se lo conoce como suavizado o eliminación de ruido, el cual se realiza pixel por pixel y proporciona una calidad de imagen superior, lo que se traduce a una detección facial más precisa.

```
# Convertir la imagen a escala de grises
gris = cv2.cvtColor(fotograma, cv2.COLOR_BGR2GRAY)

# Suavizado de la imagen
gris = cv2.GaussianBlur(gris, (21, 21), 0)
```

Substracción entre el segundo plano y el primer plano

Cuando se van a separar las imágenes en dos diferentes planos, se deben tener en cuenta que, si la iluminación no es la adecuada, se corre el riesgo de tener valores negativos; lo que

representará un lento procesamiento en la detección del rostro y un posible colapso en tiempo del prototipo. Para poder evitar este tipo de contratiempos, se tienen que dar valores absolutos en nuestra programación.

```
# Resta absoluta
resta = cv2.absdiff(fondo, gris)
```

Aplicación de umbral

Este método trata de separar los pixeles que superen un cierto tipo de parámetro, este se destaca por el movimiento que tiene, dando como resultado a aquellas partes del cuerpo que están en movimiento e ignorando a aquellas partes corporales que no lo están. Para de esta manera poder enfocar los movimientos que tiene el rostro y poder compararlos con los algoritmos que están presentes en una posible somnolencia.

```
# Aplicamos el umbral a la imagen
umbral = cv2.threshold(resta, 25, 255, cv2.THRESH_BINARY)
```

Detección de contornos

La detección de contornos, la cual es conocida como separación de blobs (un conjunto de pixeles que se conectan unos con otros dando una figura estándar o similar entre las personas) es una técnica que limita los espacios de fondo y permite enfocar en un espacio delimitado. Esto facilita el proceso de ojos y bocas, puesto que, aunque cada persona es diferente entre sí, los rasgos que los componen tienen una similitud y son representados en la Figura 17 (Luis del Valle Hernández, 2017).

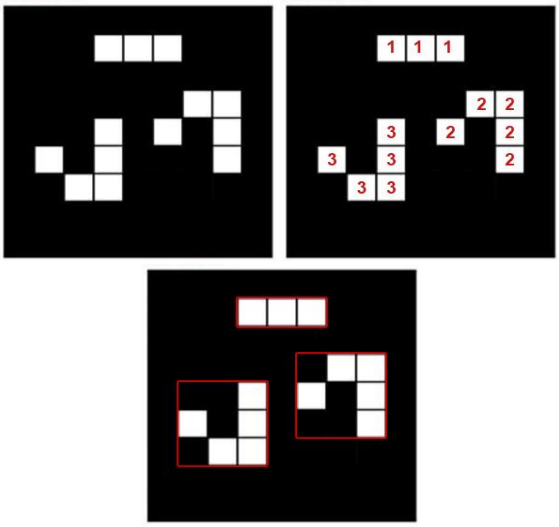


Figura 17: detección de contornos

Fuente: (Luis del Valle Hernández, 2017)

Desarrollo del algoritmo

2.1.8. Esquematización

Al tener en cuenta los factores que determinan que un conductor entra en estado de somnolencia, se procede a crear un algoritmo que sea capaz de detectar los factores que influyen en la falta de concentración y producen somnolencia. La secuencia lógica con la que contara el algoritmo, determina una disminución constante en el modo de parpadeo de los ojos del conductor, siendo este el factor que determine que está entrando a un estado de somnolencia.

El esquema secuencia que el algoritmo tendrá se describe en la Figura 18:

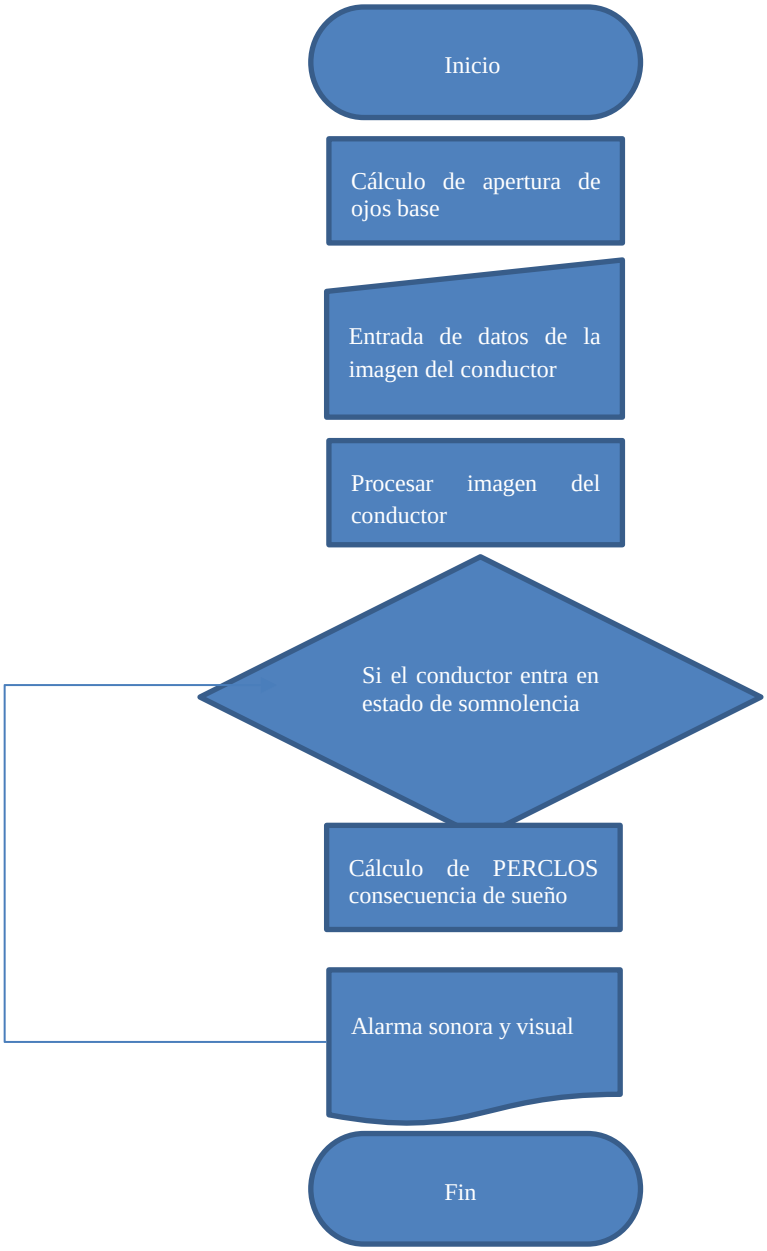


Figura 18: Diagrama de secuencia

Fuente: Elaboración propia

El algoritmo detectará el rostro del conductor enfocándose en el rostro, delimitara los ojos para poder hacer un enfoque a la latencia que tiene mientras está conduciendo, guardando y procesando el retardo de parpadeo con el estándar presentado en la metodología de PERCLOS; donde si el rango de apertura de ojos llega a ser menor del establecido por el método y este lapso

se prolonga por más de 10 fotogramas seguidos, el programa dará como resultado una alarma visual y auditiva que prevendrá al conductor.

2.1.9. Casos de uso

El proceso secuencial que realiza el sistema de detección de somnolencia se describe mediante la Tabla 6 con pasos guiados, que tiene como nombre *"Casos de uso"*, que representa los caminos a seguir con sus limitantes y acciones.

Tabla 6: Requerimiento 1

Requerimiento #1
Actor: Usuario, **Persona que utilizará el sistema de detección de somnolencia**
Identificador del Caso de uso: **CU001:** **Detectar Somnolencia en un conductor utilizando una cámara mediante el microrordenador Raspberry pi.** Se detalla el proceso necesario para la detección de somnolencia del conductor.
Secuencia de eventos
Flujo Normal: Detectar Somnolencia en un conductor utilizando cámara.
1. El usuario coloca la cámara de manera que su rostro quede cerca de la cámara del vehículo
2. La aplicación detecta su rostro.
3. La aplicación detecta sus ojos.
4. La aplicación calcula el índice de somnolencia y es mayor al 80%
5. La aplicación emite una alerta sonora y otra audiovisual.
6. El usuario debe mirar fijamente a la cámara para que detecte que ha despertado.
7. La aplicación regresa al evento 2.
Flujo Alterno 1. A partir del Evento 4
4. **La aplicación calcula el índice de somnolencia y es menor al 80%**
5. **La aplicación vuelve al evento 3**
Flujo Alterno 2. A partir del Evento 5
De no realizar el evento 6 por parte del usuario la aplicación seguirá en el evento 5

Fuente: Elaboración propia

2.1.10. Programación del algoritmo

A continuación, se presenta el algoritmo de detección de somnolencia, desarrollado en el lenguaje de programación Python con el IDE Visual Studio Code:

Carga de librerías: para la detección de rostros se necesita de librerías externas que contengan los archivos base de visión artificial. *OpenCv* será la encargada de enmarcar el rostro del conductor y, ayudada por la librería *Dlib,* la cual se hará cargo del aislamiento de los ojos. A su vez, se implementa la librería *playsound,* que se encargara de enviar la alarma sonora.

```
import dlib
import cv2
import playsound
```

Cálculo de distancia: el primer paso para poder realizar la detección es el determinar la distancia que hay entre los ojos, para lo cual se definen dos variables, que restando una de la otra, se tiene el valor exacto para determinar el valor que tiene un ojo con el otro.

```
def euclidean_dist(ptA, ptB):
    return np.linalg.norm(ptA - ptB)
```

Función de apertura ocular: se realiza el cálculo que tiene la apertura de los ojos del conductor que está en el volante, para lo cual se suman ambas aperturas verticales dadas por las variables A y B, las cuales serán divididas por la multiplicación de 2 (por ser ambos ojos) con a la distancia horizontal marcada por la variable C; a este resultado se le retorna en la variable *ear,* la cual será el punto de referencia para la comparación de somnolencia.

```
def eye_aspect_ratio(eye):
    A = euclidean_dist(eye[1], eye[5])
    B = euclidean_dist(eye[2], eye[4])

    C = euclidean_dist(eye[0], eye[3])

    ear = (A + B) / (2.0 * C)

    return ear
```

Inicialización de constantes: se declaran las constantes que serán el punto de comparación para detectar somnolencia en el conductor. En primera instancia se tiene a la apertura de ojos, la cual tendrá un valor base de 0,3 y será comparada con el parámetro *ear*

anteriormente calculado, a su vez, el contador de frames se establece con un valor de 10, el cual será el límite de fotogramas permitido antes de que se presenten las alarmas de alerta.

```
EYE_AR_THRESH = 0.3
EYE_AR_CONSEC_FRAMES = 10
```

Carga de librerías: se cargan los archivos de las librerías OpenCv y Dlib para su oportuno uso.

```
detector = cv2.CascadeClassifier(args["cascade"])
predictor = dlib.shape_predictor(args["shape_predictor"])
```

Inicio bucle principal: se inicia el primer bucle, el cual contiene todo el código para la detección de somnolencia y el mismo que se repetirá de manera indefinida mientras el sistema esté en marcha. Esta función se realiza para que el programa esté en constante detección sin un tiempo límite.

```
while True:
```

Cambio a escala de grises: cada fotograma que pasa por la cámara, se convierte a escala de grises, para que las librerías puedan detectar con mayor facilidad cada parte del rostro del conductor.

```
frame = vs.read()
frame = imutils.resize(frame, width=450)
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

Inicio del segundo bucle: se inicia el segundo loop que será repetido hasta que el enfoque de la cámara no presente cambios en el estado del conductor. El programa enfoca los ojos y apenas detecte que existe una variación en apertura, la cual está determinada por los puntos: x,y,w y h, empezará el sondeo si el patrón continúa por más de 10 fotogramas.

```
for (x, y, w, h) in rects:
        # crea un marco en los ojos
        rect = dlib.rectangle(int(x), int(y), int(x + w),
            int(y + h))
```

Pregunta de control: el programa hace la pregunta si la apertura de ojos del conductor es mejor al parámetro establecido como constante, entonces suma al contador 1 por cada fotograma en el que el usuario tenga la apertura inferior de ojos a la establecida como mínimo para considerar somnolencia.

```
if ear < EYE_AR_THRESH:
            COUNTER += 1
```

Pregunta de fotograma: si el contador de fotogramas llega a ser mayor al establecido, entonces se considera que el conductor ha permanecido mucho tiempo con los ojos entre abiertos y se afirma que está en estado de somnolencia. Para lo cual la alarma se pondrá en ON y dará como resultado alertas visuales y sonoras.

```
if COUNTER >= EYE_AR_CONSEC_FRAMES:
                if not ALARM_ON:
                    ALARM_ON = True
```

Alerta sonora: cuando el algoritmo define que está en somnolencia, envía dos alarmas para que puedan ser detectadas, la primera es una sonora que pitara por un lapso de dos segundos.

```
                pygame.mixer.music.play()
                while pygame.mixer.music.get_busy() == True: continue
```

Alerta visual: el comando GPIO permite enviar una corriente directa a los pines de salida del módulo Raspberry, siendo el número 18 en el tablero el cual prenderá la luz. Con un lapso de un segundo de latencia se prenderá la alarma visual y se volverá a apagar quedando inactiva hasta que detecte otro estado de somnolencia en el conductor.

```
            GPIO.setmode(GPIO.BCM)
            GPIO.setwarnings(False)
            GPIO.setup(18,GPIO.OUT)
            time.sleep(1)
            GPIO.output(18,GPIO.HIGH)
```

Características y ventajas

- El sistema detecta al usuario e iniciará el proceso de monitoreo de expresiones faciales para extraer características de síntomas de somnolencia: El sistema será capaz de identificar el algoritmo a utilizar, debido a que éste contiene 2 librerías de detección facial independientes (OpenCV y DLib), uno es el encargado de utilizar la detección de imágenes cuya función es el monitoreo del usuario. El otro proceso utiliza la cámara para el monitoreo del rostro. El sistema también deberá iniciar mientras el conductor

está frente al volante, el cual permitirá enviar alertas de presencia de somnolencia al usuario.

- El sistema de detección envía una respuesta al dispositivo para generar una alarma visual y otra sonora. Luego de que el sistema detecte presencia de somnolencia enviará un mensaje al dispositivo que está conectado a unas bocinas, las cuales activarán un sonido que pueda ser escuchado por el usuario y a la vez, se mostrará una alerta visual.
- El sistema podrá almacenar información sobre los algoritmos que generan somnolencia en los usuarios: coordenadas X y Y del rectángulo del ojo, ancho y alto de rectángulo del ojo, coordenadas X y Y del rectángulo de la boca, ancho y alto de rectángulo de la boca.
- El software de monitoreo cuenta con un margen de detección sin interferencias: el usuario no deberá tener ningún tipo de accesorio sobre su cuerpo para que el sistema funcione correctamente. Los dispositivos serán ubicados de tal forma que no afecten o incomoden la visibilidad del usuario mientras éste conduce. La alerta de notificación será de tipo auditiva y otra de tipo visual, para que el usuario no distraiga su mirada de la carretera.

CAPÍTULO 4

IMPLEMENTACIÓN

Inicialización

Una vez teniendo el conocimiento de la secuencia que el sistema debe seguir para poder cumplir con los objetivos del proyecto, se procede a inicializar los componentes, tanto de hardware visible en la Tabla 7, como de software mostrada en la Figura 19; para el desarrollo y puesta en marcha del prototipo.

Tabla 7: Inicialización de recursos

RECURSO	TIPO	DESCRIPCIÓN
Raspberry Pi	**HADWARE**	Se utilizó un microrordenador para realizar las pruebas de funcionamiento
Computador	**HADWARE**	Se utilizó para el desarrollo del sistema de detección.
Python	**SOFTWARE**	Lenguaje de programación para el desarrollo del sistema.
OpenCv	**SOFTWARE**	Herramienta para el desarrollo de visión artificial.
Visual Studio Code	**SOFTWARE**	IDE para el manejo del lenguaje de programación y las librerías

Fuente: Elaboración propia

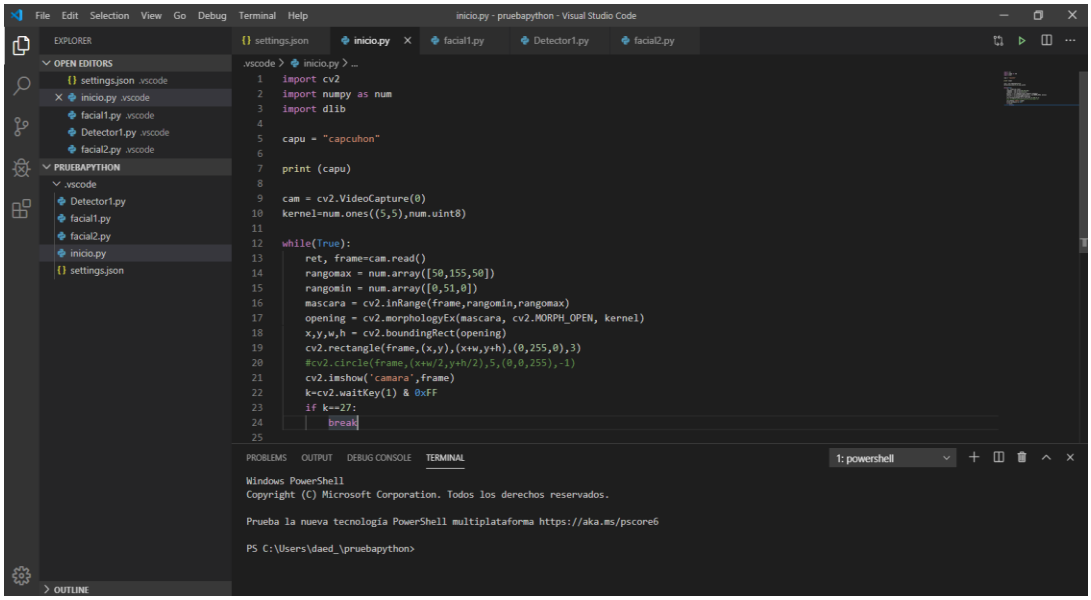


Figura 19: Inicialización en Python

Fuente: Elaboración propia

- **Librería de visión artificial OpenCv:** Utilizada para el procesamiento digital de imágenes con el código: *import cv2*
- **Librería de vectores y matrices:** Utilizada para la interpretación de imágenes transformándolas en vectores con el código: import numpy as num.
- **Librería de visión artificial DLib:** Utilizada para un procesamiento estándar de imágenes digitales, la cual interactúa con la librería OpenCv para una mejor detección en el rostro del usuario. Se la usa con el código: import dlib
- **Utilización del kernel del sistema:** Utilizada para llamar de manera física a los puertos conectados en el procesador, en el caso específico del presente proyecto, se llama al puerto de la cámara. Se usa con el siguiente código:

```
cam = cv2.VideoCapture(0)
kernel=num.ones((5,5),num.uint8)
```

- **Utilización de rangos de visión:** Utilizada para delimitar el rango de visión que tiene la cámara, haciendo un enfoque al rostro del usuario para su posterior procesamiento de datos centrándose en los ojos. Se usa con el siguiente código

```
rangomax = num.array([50,155,50])
rangomin = num.array([0,51,0])
```

- **Utilización de visión artificial**: Utilizada para detectar los movimientos de partes específicas dentro de una imagen en movimiento por medio de la cámara predeterminada. Se usa con el siguiente código, haciendo enfoque a los ojos del usuario:

```
cv2.rectangle(frame,(x,y),(x+w,y+h),(0,255,0),3)
```

- **Finalización de sentencia:** Utilizada en modo de prueba, para interrumpir el sistema al presionar en caso de que ocurra algún fallo en el algoritmo. Se utiliza con el código:

```
k=cv2.waitKey(1) & 0xFF
```

Los procesos que realiza la aplicación secuencial para detectar un estado de somnolencia se detallan en la Figura 20:

Figura 20: Secuencia de detección de somnolencia

Fuente: Elaboración propia

Detector del rostro: el programa detecta las facciones faciales para poder delimitarlas y enfocarse exclusivamente en el rostro, dejando a un lado el entorno adjunto.

Detector de los ojos: una vez delimitado el rango del video, se enfoca a las acciones que realicen los ojos, ignorando las demás partes del rostro.

Estado del conductor: mediante los algoritmos presentados sobre la somnolencia, se procede a calcular los niveles de somnolencia que tiene el conductor mediante la variación de pestañeo.

Implementación

Los implementos que son necesarios para arrancar por primera vez el prototipo son los siguientes:

- Raspberry pi (USB)
- Alimentador de 5v y 2500mA
- Cable HDMI y Monitor con entrada HDMI
- Memoria micro SD con adaptador de 16GB o más
- PC Mouse y Teclado USB
- Cable Ethernet
- Parlante de alerta
- Cámara

Al microrordenador Raspberry Pi se le debe colocar una tarjeta SD del espacio necesario para el funcionamiento del sistema operativo y los utilitarios de programación básicos (16GB de espacio es más que suficiente para el uso del prototipo de detección de somnolencia). Tanto la cámara que será utilizada para hacer la visualización del rostro del conductor, como los ventiladores y disipadores; se instalan en las ranuras pertinentes mostradas en la Figura 21.

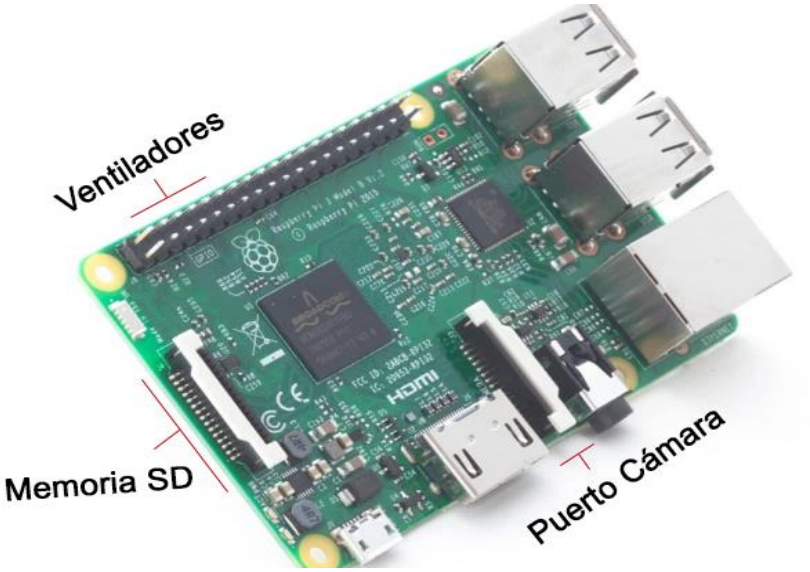


Figura 21: Componentes Raspberry pi

Fuente: Elaboración propia

Una vez conectado el dispositivo, se procede a encapsularlo dentro de un case mostrado en la Figura 22. De este modo se lo tiene seguro para su manipulación y puesta en marcha dentro del vehículo.

Figura 22: Case Raspberry pi

Fuente: Elaboración propia

Al encender el dispositivo se presenta la pantalla inicial del sistema operativo que se instaló previamente (ver en anexos “Manual de desarrollo”) y se procede a cargar el programa de detección de somnolencia que se ha desarrollado y descrito en el capítulo anterior (para ver el código completo visualizar anexos “Código fuente”). A continuación, se describen los pasos para cargar el programa desarrollado dentro del prototipo Raspberry:

- **Abrir el panel comandos - sudo raspi-config**
- **Extender la memoria SD a toda su capacidad**

- **Activar la cámara**

- **Comprobar la Cámara:**

$ raspivid -o vid1.h264 -t 20000

$ omxplayer vid1.h264

- **Instalar librerías estándar**

$ sudo apt-get update

$ sudo apt-get install build-essential cmake

$ sudo apt-get install libopenblas-dev liblapack-dev libatlas-base-dev

$ sudo apt-get install libx11-dev libgtk-3-dev

$ pip install playsound

$ pip install pyobjc

- **Instalar Librerías de reconocimiento OpenCV**

$ sudo apt-get install python3-scipy

$ sudo apt-get install espeak

$ pip install imutils

$ sudo pip3 install argparse

$ sudo pip3 install VideoStream

- **Instalar Dlib**

$ pip3 install numpy

$ pip3 install dlib

- **Exportar el programa de detección de somnolencia que se ha desarrollado**

  $ sudo python /home/pi/somnolencia.py

- **Iniciar el programa directo al iniciar el microrordenador**

  sudo python /home/pi/sample.py &

El Raspberry iniciará el programa con el boot, antes de que otros servicios comiencen. Esto hará que el programa arranque continuamente al iniciar el módulo raspberry en un proceso de booteo.

- **Reinicio del microrordenador**

  sudo reboot

Al tener el sistema de detección correctamente instalado en nuestro prototipo Raspberry pi, se coloca dentro del vehículo en una posición donde la cámara pueda detectar directamente al rostro del conductor. Se enciende y empieza con el análisis de los resultados estimados que arroje la implementación y que se muestran en la Figura 23.

Figura 23: Implementación prototipo

Fuente: Elaboración propia

Pruebas de funcionamiento

Se pretende detallar las pruebas necesarias para evaluar la aplicación las mismas que se dividen en dos pruebas de funcionalidad, y pruebas de factibilidad a continuación se detalla el plan para cada una de ellas.

- Funcionalidad.
    - Pruebas Unitarias son las cuales se realizan a los métodos y las clases dentro de la estructura de programación.
    - Pruebas de Compatibilidad: al utilizarse un único dispositivo para correr el sistema (Raspberry v3) la compatibilidad es asegurada, por los módulos que han sido implementados para que funcionen únicamente con el sistema operativo que se instala en el microrordenador.
    - Pruebas en un dispositivo: al igual que las pruebas anteriores, todos los módulos y funciones están diseñadas con el propósito de ser acopladas a un solo prototipo portátil Raspberry.
- Factibilidad.
    - Validación Pruebas con sujetos reales para determinar el margen de error de la aplicación.

3.1.1. Pruebas de funcionalidad

Las pruebas se aplicaron para validar el correcto funcionamiento del código mediante el depurador de error que provee el IDE seleccionado (Visual Studio Code) mostrado en la Figura 24 con sus clases, métodos y funciones; los cuales no deben presentar errores en una primera instancia de pruebas, antes de poder traspasarlos al dispositivo Raspberry con ejecución en tiempo real de detección de rostros.

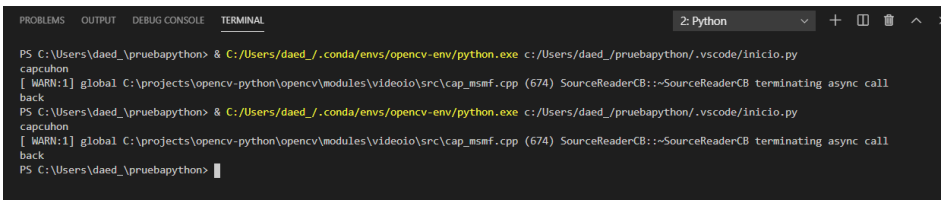


Figura 24: Prueba de funcionalidad

Fuente: Elaboración propia

3.1.2. Pruebas de factibilidad

Las diferentes características a probar se representan en la Tabla 8:

Tabla 8: Estrategia de pruebas

Alcance	Aplicación en un microrordenador Raspberry para la detección de somnolencia de un conductor aplicando visión artificial
Ítems a probar	Detección de rostro Detección de somnolencia Alarma de registro de somnolencia
Estrategia	Manejo de la aplicación por parte de los usuarios. Obtener información a partir de las pruebas realizadas
Recursos	Raspberry pi Camara ip Computador

Fuente: Elaboración propia

3.1.3. Limitaciones

- El prototipo de detección solo podrá funcionar con un módulo Raspberry pi con su cámara integrada.
- Los conductores que lo vayan a utilizar deben tener las bases de un conocimiento en manejo de dispositivos electrónicos.
- Se necesita de iluminación extra para usarlo en entornos nocturnos o con poca luz.
- El conductor tendrá que tener el rostro sin ningún artículo que oculte el área de sus ojos.
- El conductor debe colocar al prototipo en un ángulo correcto para que la cámara pueda detectar su rostro de manera directa.

3.1.4. Iluminación

Según indican las limitaciones del prototipo, el uso del mismo en factores con poca iluminación se vuelve un factor decisivo para una correcta captura del rostro del conductor, detectando de esta manera, un estado de somnolencia en ambientes nocturnos y con poca iluminación. Para dar una solución al proceso de imágenes de visión artificial en entornos

oscuros, se presenta la disposición de utilizar un sistema de luz integrado dentro del vehículo que alumbre al conductor cuando detecte que la luz es escasa. Dando de esta manera, un método practico para solucionar una de las limitantes del prototipo.

Se tomó como muestra a las llamadas: “luces de cortesía”, estas luces se suelen encontrar situadas en el techo del vehículo y se iluminan cuando el conductor o los pasajeros entran o salen del vehículo. La iluminación normalmente permanece encendida hasta que el vehículo se enciende para que los pasajeros puedan abrocharse los cinturones con seguridad.

Además, dichas luces como otras interiores pueden ayudar al conductor o a cualquiera de los pasajeros a leer mapas, poner el navegador o elegir la música que escucharán durante el trayecto. En los vehículos más modernos, se incluye una opción para poder encender o apagar dicha luz. Al tomarlas como ejemplo, se implementó una tira de LED’s en la parte superior del puesto de conductor, la cual se encenderá de manera automática cuando detecte una caída de iluminación.

Figura 25: Iluminación dentro del vehículo

Fuente: Elaboración propia

La iluminación mostrada en la Figura 25, es externa al prototipo, siendo un aditamento para facilitar la visibilidad del algoritmo en entornos nocturnos.

Costos

Para el desarrollo del presente trabajo se utilizaron recursos técnicos, económicos, humanos que contribuyeron al cumplimiento de cada uno de los objetivos planteados y los cuales se detallan en la Tabla 9:

Tabla 9: Presupuesto total

Recursos	**Subtotales**
Recursos Humanos	$ 2000,00
Recursos Técnicos y tecnológicos	$ 730,00
Recursos Materiales	$ 200,00
Subtotal	**$ 2930,00**
Imprevistos 10%	$ 293,00
Total	**$3223,00**

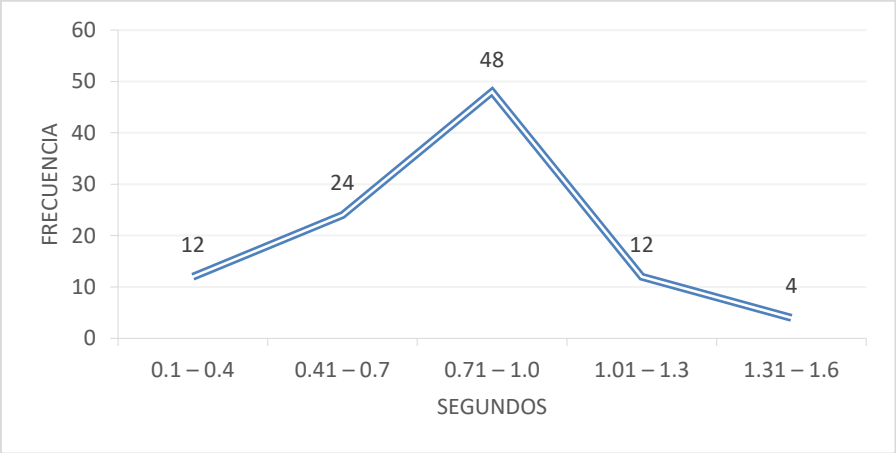
Fuente: Elaboración propia

Análisis de resultados

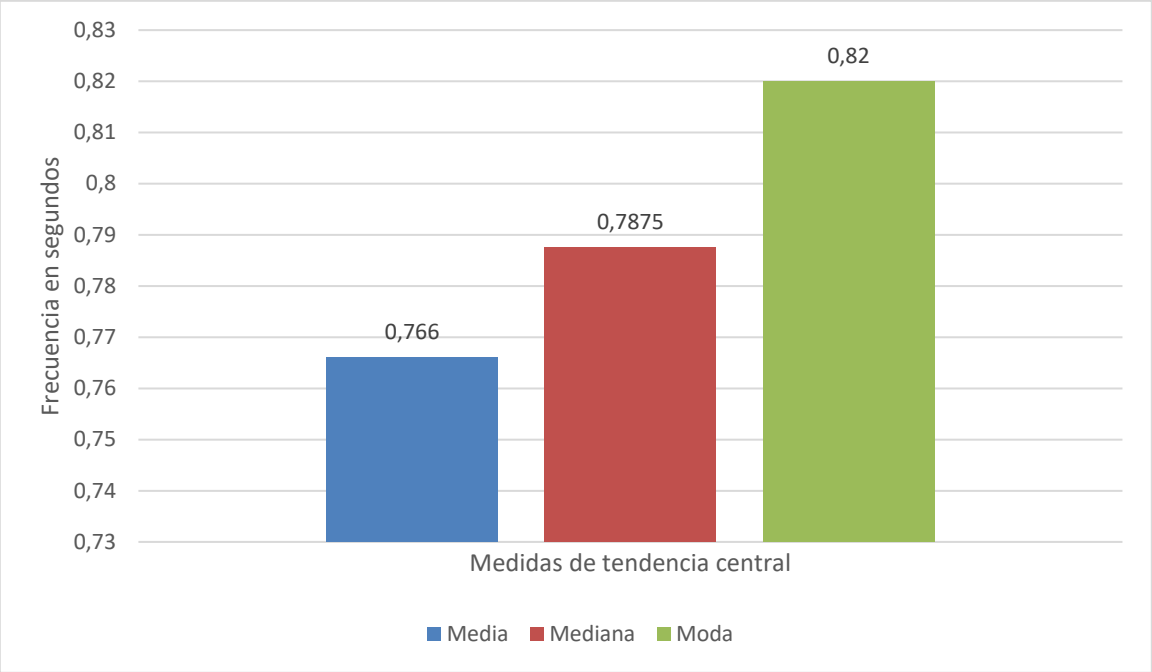
Pregunta 1: ¿En cuánto tiempo se realizó la detección de somnolencia?

Gráfico 1: Pregunta 1

Tiempo (seg)	x	f	F	xf
0.1 – 0.4	0.25	12	12	3
0.41 – 0.7	0.55	24	36	13.2
0.71 – 1.0	0.85	48	84	40.8
1.01 – 1.3	1.15	12	96	13.8
1.31 – 1.6	1.45	4	100	5.8
		100		76.6



Media	$x = \frac{\sum xf}{n}$	$x = \frac{76.6}{100}$	x = 0.766 seg
Mediana	$Me = Li + A\,(\frac{\frac{n}{2} - Fi - 1}{fi})$	$Me = 0.71 + 0.3\,(\frac{50 - 36}{48})$	Me = 0.7875 seg
Moda	Mo = $Li + A(\frac{fi - fi{-}1}{(fi - fi{-}1) + (fi - fi{+}1)})$	Mo = $0.71 + 0.3(\frac{48-24}{(48-24)+(48-12)})$	Mo = 0.82 seg



Fuente: Elaboración propia

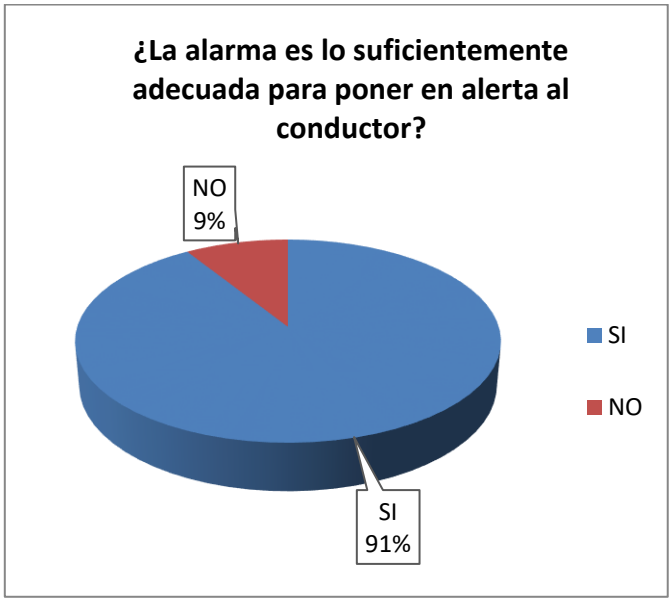
Análisis:

Para el desarrollo del proyecto existió la necesidad de tener el rango exacto de los segundos que se demora el prototipo a la hora de realizar un escaneo del rostro del usuario, enviarlo a que el programa rastree sus signos y devuelva un resultado en forma de alarma que avise al conductor que ha estado en un posible estado de somnolencia.

Los datos mostrados en las tablas y estadísticas del Gráfico 1, demuestran que las medidas de tendencia central utilizadas dan un resultado parecidas entre ellas. Donde la media tiene un promedio de 0.76 segundos de demora, la mediana cuenta con un 0.78 segundo de demora y para finalizar la moda (la que tiene el rango mayor de las tres medidas) da como resultado un 0.82 segundos de demora para que el prototipo capte señales de somnolencia y devuelva como respuesta las respectivas alarmas.

Pregunta 2: ¿Las alarmas son lo suficientemente adecuadas para poner en alerta al conductor?

Gráfico 2: Pregunta 2



SI	91	91%
NO	9	9%
TOTAL	100	100%

Fuente: Elaboración propia

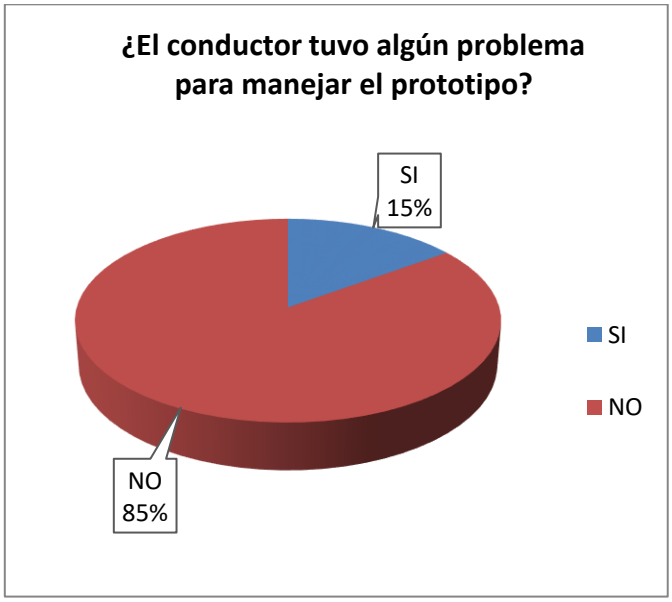
Análisis:

El objetivo principal del prototipo presentado en esta propuesta, es el de detectar y dar dos tipos de alarmas efectivas para el usuario y así prevenir accidentes que se pueden producir por un estado de somnolencia, por lo tanto, los tipos de alarmas que se dé es vital para poder tener una respuesta favorable ante el problema presentado.

Por ello se determinó un tipo de alarma que sonará y otra que parpadeará mientras el usuario este entrando en estado de somnolencia y se midió si estas alarmas son lo suficientemente notorias, o, por lo contrario, son muy tenues. En el Gráfico 2 se pueden ver que un 91% de los casos presentados, da como respuesta que las alarmas son indicadas para el conductor y lo ha alertado de manera correcta; en contra parte, tan solo un 9% de casos, han resultado como no perceptibles para el conductor.

Pregunta 3: ¿El conductor tuvo algún problema para manejar el prototipo?

Gráfico 3: Pregunta 3



SI	15	15%
NO	85	85%
TOTAL	100	100%

Fuente: Elaboración propia

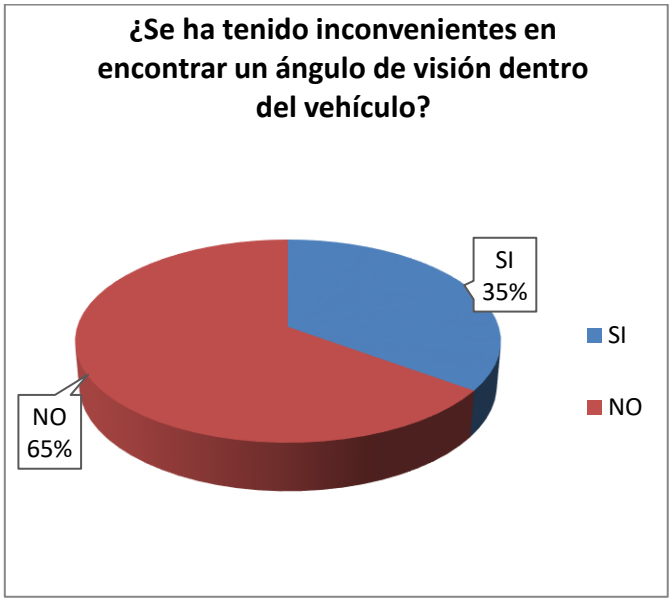
Análisis:

El presente prototipo que se implanta dentro de un vehículo para poder enfocar al rostro del conductor, es un modelo único y por lo tanto el funcionamiento de encendido y apagado del mismo, depende en gran medida de la facilidad que el conductor tenga para manejar dispositivos electrónicos. Aún con una correcta instalación de módulos que se inician de inmediato y no requieren la manipulación humana para arrancar los programas de detección, han existido algunas complicaciones por parte de los usuarios.

Se puede observar en el Gráfico 3 que un 15% de los casos en que el conductor ha tenido que manipular el prototipo, se presentó algún tipo de dificultad al accionar, encender y apagar el pototipo. En contra parte, un gran porcentaje, el cual es del 85% de casos en los que el conductor ha tenido que manipular el prototipo sin ayuda, lo ha realizado con complicaciones menores.

Pregunta 4: ¿Se ha tenido inconvenientes en encontrar un ángulo de visión dentro del vehículo?

Gráfico 4: Pregunta 4



SI	35	35%
NO	65	65%
TOTAL	100	100%

Fuente: Elaboración propia

Análisis:

Para un adecuado nivel de detección del rostro del conductor, es necesario tener un ángulo adecuado de la cámara del prototipo. En las diversas pruebas que se realizaron, uno de los inconvenientes fue el encontrar un espacio para adaptar la cámara y el armazón del prototipo sin que interfiera con el espacio determinado por los objetos personales del conductor dentro del vehículo.

Después de unos cuantos intentos, se pudo determinar en el Gráfico 4 que hay varios lugares óptimos dentro del vehículo, por ello es que el 35% de los resultados fueron pruebas que resultaron no recomendables. Y, por otro lado, el restante 65% de los casos, ya tuvieron una respuesta mucho más precisa y satisfactoria para no interferir con el espacio del prototipo al interior del vehículo.

Pregunta 5: ¿Ha sido incómodo para el conductor el manejar con el prototipo instalado?

Gráfico 5: Pregunta 5

SI	10	10%
NO	90	90%
TOTAL	100	100%

Fuente: Elaboración propia

Análisis:

Después de haber encontrado los lugares más óptimos dentro del rango de visión del prototipo y sin que interfiera con la comodidad del conductor. Se realizó un seguimiento para analizar si es adaptable en uso real al usuario mientras maneja su vehículo.

Se pudo tener como resultados en el Gráfico 5 qué, ya encontrado el ángulo correcto, la incomodidad del conductor se vio disminuida considerablemente, llegando casos en los que ni siquiera notaba la existencia de la cámara; siendo el 90% de los casos en los que no se encontró incomodidad y un 10% en los que si incomodó al usuario.

Pregunta 6: ¿Se ha reducido la incidencia de estado de somnolencia en el conductor?

Gráfico 6: Pregunta 6

SI	70	70%
NO	30	30%
TOTAL	100	100%

Fuente: Elaboración propia

Análisis:

El resultado final que se espera para la resolución del problema propuesto, es un incremento en la prevención de casos de conductores que puedan entrar en estado de somnolencia, aumentado así las posibilidades de evitar algún tipo de accidente que pueda provocar este estado en las vías de conducción.

Para poder llegar a este último análisis del Gráfico 6, se tomó en cuenta un usuario que tenga unas cuantas horas de falta de sueño, con las medidas de precaución tomadas, se realizó una prueba real durante unos cuantos minutos en un tramo sin ningún tipo de afluencia vehicular. Dando como resultado que el prototipo ponía en alerta al conductor en un 70% de los casos y el 30% restante, era pasado desapercibido.

CONCLUSIONES

- Se logró determinar una de las principales señales de somnolencia que presentan los conductores cuando entran a un estado de fatiga causante por un desequilibrio del sueño, se trata de un aumento en el tiempo de latencia a la hora de parpadear, aumentando significativamente la demora en la apertura de los ojos, manteniéndolos entreabiertos por un tiempo prolongado. Determinando esta señal la técnica aplicada para la prevención será la llamada "PERCLOS" que, según su documentación de ayuda, brinda un análisis sistemático que permite determinar rangos en la somnolencia.
- El algoritmo desarrollado se divide en el análisis de la imagen del conductor para sustraer cada fotograma y analizarlo de manera independiente, enfocando la reacción de los ojos para determinar cuándo causa un síntoma de somnolencia; si el síntoma es prolongado, el código desarrollado responderá con alarmas mediante los pines conectados directamente al módulo prototipo del Raspberry pi.
- Se pudo desarrollar un algoritmo adaptable a las expresiones faciales de los usuarios. Dando de esta manera apertura para integrar en el prototipo un sistema de alarmas que consta de una sonora a modo de pitido y otra visual a modo de parpadeo de luces; las cuales ayudan a mantener en estado de alerta al conductor cuando empieza a tener inicios de somnolencia frente al volante.
- La implementación del prototipo dio como primeros resultados una lenta respuesta de somnolencia, con las pruebas de entorno aplicadas, se mejoró notablemente el tiempo de respuesta modificando el algoritmo para que presente una velocidad mayor en la toma de video y de esta manera obtener niveles óptimos hacia síntomas de somnolencia. De la misma manera, al realizar las pruebas de entorno respectivas, se implementó una luz interior en el vehículo de prueba para que la detección pueda funcionar en horarios nocturnos.
- Al desarrollar un prototipo que sea capaz de ser portátil con la utilidad que presenta el microrordenador en su modelo de Raspberry pi, hace que se pueda implementar en cualquier vehículo y con cualquier usuario que quiera tener un soporte externo para

predecir posibles accidentes a causa de la somnolencia a la hora de conducir. De igual manera, al desligar de productos de terceros (como Smartphone o cámaras con base de datos incorporadas) se tiene el control total de hacer modificaciones y adaptaciones para cada tipo de vehículo y usuario.

- El cansancio ha hecho que se produzcan accidentes a causa de una posible somnolencia que presentan los conductos al estar largas jornadas sin sueño, el poder implementar una solución asequible y portátil, hace que el porcentaje de riesgo se vea reducido, aportando de esta manera un método adaptable que puede ser tomado en consideración para futuros proyectos.

RECOMENDACIONES

- Se recomienda incorporar el prototipo a artefactos internos del vehículo para que la alarma pueda ser más notoria, el usar tanto los parlantes y luz vehicular interno, hará que los accesorios extras que tiene el prototipo desaparezcan; a su vez, hay que tomar en cuenta que el incorporar este tipo de mejoras, será un proceso para cuando el prototipo pase las normas regulatorias de precauciones vehiculares.
- Por ser una investigación práctica con algoritmos adaptables, es posible incorporar funcionalidades de tiempo para detectar cuales son los rangos horarios que los conductos tienen para empezar a estar dentro de un estado de somnolencia, esto mediante una base de datos que registre las interacciones de visión del conductor, sus horarios y posibles causas.
- Dentro del proceso de análisis de los accidentes que pueden sufrir los conductores por estar en un estado de somnolencia, ha hecho que se demuestren que es un síntoma que puede traducirse a graves accidentes, por ello se recomienda el tener mecanismos más fuertes cuando se esté detectando una falta de concentración del conductor. Como ejemplo, se puede incorporar un freno automático cuando el usuario esté por más de unos minutos en estado de somnolencia y que las alarmas ya no le sea suficiente para entrar en concentración nuevamente.

REFERENCIAS BIBLIOGRÁFICAS

A., E. J. . (2010). *Algoritmo de reconocimiento de forma y color para una plataforma robótica.* Universidad del País Vascp.

Acosta, J. R. . (2010). *ARQUITECTURA E IMPLEMENTACIÓN DEL PROCESO DE LOCALIZACION ON-LINE En Sistema de Localización en Interiores Basado en Redes WiFi.*

Adolfo Valdes. (19 de Septiembre de 2019). *Raintech*. Obtenido de https://raintech.cl/que-es-vision-artificial/

Álvarez Romero, L. M., & Figueroa Montenegro, J. E. (11 de Noviembre de 2017). *Repositorio Institucional de la Universidad de las Fuerzas Armadas ESPE.* Obtenido de http://repositorio.espe.edu.ec/handle/21000/2643

AMERICAN SLEEP ASSOCIATION. (10 de Diciembre de 2019). *ASA.* Obtenido de https://www.sleepassociation.org/

Calero, M. J. (2010). *Sistema avanzado de asistencia a la conducción mediante visión por computador para la detección de somnolencia.* Leganés.

Camacho, J. (2009). *Medición de distancias por medio de procesamiento de imagen, triangulación, haciendo uso de cámaras de video.* Universidad de las Américas.

Carrillo-Mora, P., Ramírez-Peris, J., & Magaña-Vázquez, K. (2013). Neurobiología del sueño y su importancia: antología para el estudiante universitario. *Revista de La Facultad de Medicina*, 5–15.

Carrillo-Mora, Ramírez-Peris, & Magaña-Vázquez. (2013). Neurobiología del sueño y su importancia: antología para el estudiante universitario. *Revista de la Facultad de Medicina*.

Conlago Guatemal, C. R., & Yunda Sangoluisa, J. A. (10 de Agosto de 2016). *Sistema Automático de detección y reconocimiento de señales de tránsito en intersecciones viales para aplicaciones en vehículos inteligentes.* Obtenido de Universidad de las

Fuerzas Armadas ESPE: https://repositorio.espe.edu.ec/bitstream/21000/12242/1/T-ESPE-053557.pdf

Crespín Luis, J. C., & Julián García, R. A. (2014). *Sistema detector de somnolencia en secuencias de video de conductores manejando usando visión computacional.* Obtenido de Universidad Nacional de Trujillo: http://www.inf.unitru.edu.pe/revistas/2014/4.pdf

Dallas, M. E. (2018). *Somnolencia durante la conducción.* HealthDay.

De La Cruz, C., Carelli, R., & Gava, C. C. (2006). *Control centralizado de formacion usando una câmara omnidireccional. IV Jornadas Argentinas de Robótica.* .

Diego, L. G., & Fernando, G. B. (21 de Julio de 2012). *Universidad de Huelva.* Obtenido de http://rabida.uhu.es/dspace/bitstream/handle/10272/5501/Nuevas_aportaciones_en_algoritmos_de_planificacion.pdf?sequence=2

Edmundo Rosales Mayor. (2010). *Somnolencia: Qué es, qué la causa y cómo se mide.*

Efrain Ernesto, A. V. (2015). *Implementación de reconocimiento de objetos por color y forma en un robot móvil.* Computing Science.

Efrain Ernesto, A. V., Arturo, Z. L., & Juan, V. C. (2015). *Implementación de reconocimiento de objetos por color y forma en un robot móvil.* Research in Computing Science.

Egas Cunalata, F. D. (2017). *Sistema de vigilancia al conductor vehicular basado en técnicas de visión artificial e implementado en un Smartphone para la detección y alerta de somnolencia.* Obtenido de http://repositorio.espe.edu.ec/handle/21000/13442

España Tarira, L. G., & Oña Paredes, E. M. (2018). *Implementación de un prototipo para la detección de signos de fatiga del conductor aplicando visión artificial en un vehículo liviano en la noche.* Obtenido de Universidad Politécnica Salesiana: https://dspace.ups.edu.ec/bitstream/123456789/15181/1/UPS - ST003421.pdf

Espinola Gonzales, J. E., Asís López, M. E., & Rodríguez Sabino, V. G. (2011). *Artificial vision system for the detection of drivers drowsiness, based on the ocular behavior (Universidad Nacional "Santiago Antúnez de Mayolo").* Obtenido de

http://repositorio.unasam.edu.pe/bitstream/handle/UNASAM/2264/T033_17959973_TI.pdf?sequence=1&isAllowed=y

Flores, M. J., Armingol, J. M., & Escalera, A. de la. (2018). *Real-time drowsiness detection system for an intelligent vehicle.* Obtenido de https://doi.org/10.1109/IVS.2008.4621125

Fundación CEA. (2015). *El sueño y la fatiga en la conducción.* Madrid.

G. J. L. H. Rey de Castro J. (2004). *Cansancio y somnolencia en conductores de ómnibus y accidentes de carretera.*

Garcés, M. A. (2015). Driver drowainess detection systems: Beginning, development and future. *Revista de Ingeniería y región*, 159-168.

García Sánchez, A. (s.f.). *Sistema de control de somnolencia al volante.* Obtenido de Universidad Politécnica de Madrid: http://oa.upm.es/47421/9/TFG_ANTONIO_GARCIA_SANCHEZ.pdf.pdf

Hospital Intermutual de Levante. (17 de Junio de 2019). *La importancia del sueño.* Obtenido de https://www.hilevante.com/la-importancia-del-sueno/

INCIBE. (22 de Septiembre de 2017). *Instituto Nacional de Ciberseguridad de España.* Obtenido de https://www.certsi.es/blog/amenazas-sci

Inglett, J. E., & Rodríguez-Seda. (2017). Object transportation by cooperative robots. *SoutheastCon*, 1-6.

Irma Josefna GArcía Enríquez, M. N. (2009). Segmentación de rostro por color de la piel aplicado a detección de somnolencia en el conductor. *Congreso Nacional de Ingeniería Electrónica del Golfo CONAGOLFO*, 67-72.

J. C. G. Crespín Luis,. (2014). *Sistema detector de somnolencia en secuencias de video de conductores manejando usando visión computacional.* Obtenido de http://www.inf.unitru.edu.pe/revistas/2014/4.pdf.

Javier Andrés, L. Z., & Mario Alberto, R. V. (2016). Visión artificial y comunicación en robots cooperativos omnidireccionales. *lnqeciencia - Revista de la Facultad de Ingeniería y Ciencias Básicas*, 5-12.

Kim, Y., & Kim, B. K. . (2017). Time-Optimal Trajectory Planning Based on Dynamics for Differential-Wheeled Mobile Robots With a Geometric Corridor. *IEEE Transactions on Industrial Electronics*, 5502-2212.

La Serna Palomino, R. C. (2019). Técnicas de Segmentación en Procesamiento Digital de Imágenes. *Revista de Ingeniería de Sistemas e Informática*, 9-16.

Libre Computer. (10 de Enero de 2020). *Libre Computer* . Obtenido de https://libre.computer/products/boards/aml-s905x-cc/

López Romero, W. L. (2016). *Sistema de control del estado de somnolencia en conductores de vehículos.* Obtenido de http://repositorio.uta.edu.ec/handle/123456789/19363

López, D., Gómez-Bravo, F., Cuesta, F., & Ollero, A. . (2010). Planificación de trayectorias con el algoritmo RRT. Aplicación a robots no holónomos. *Revista Iberoamericana de Automática e Informática Industrial*, 56-67.

Luis del Valle Hernández. (10 de Septiembre de 2017). *Programar Fácil*. Obtenido de https://programarfacil.com/blog/vision-artificial/deteccion-de-movimiento-con-opencv-python/

Luis España, E. O. (2018). *Implementación de un prototipo para la detección de signos de fatiga del conductor de un vehículo liviano en la noche.* Quito: Universidad Politécnica Salesiana.

Marketingbcnvision. (2014 de Junio de 2014). *Bcnvision*. Obtenido de https://www.bcnvision.es/blog-vision-artificial/las-herramientas-mas-utilizadas-en-vision-artificial/

Ministerio de educación de España. (Febrero de 2012). *Vision Artificial.* Obtenido de http://visionartificial.fpcat.cat/wp-content/uploads/UD_1_didac_Conceptos_previos.pdf

Morales, L., Pozo, D., Rosero, J., Sandobalin, S., & & Rodríguez, M. (2014). Mapeo de Laberintos y Búsqueda de Rutas Cortas MedianteTres Mini Robots Cooperativos. *Revista Politécnica*, 101-106.

Mujica, J. R., Mayor, E. R., & Rojas, M. E. (2010). *Somnolencia y cansancio durante la conducción: accidentes de tránsito en las carreteras del Perú.* Obtenido de http://www.scielo.org.pe/scielo.php?script=sci_arttext&pid=S1728-59172009000100011

OpenCv. (Mayo de 2014). *OpenCv.* Obtenido de http://www.ecured.cu/index.php/OpenCV.

P. d. Madrid. (15 de Mayo de 2015). *Introducción a la Visión Artificial,.* Obtenido de http://www.elai.upm.es/webantigua/spain/Asignaturas/MIP_VisionArtificial/ApuntesVA/cap1IntroVA.pdf.

Palma Jaramillo, M. A., & Torres Berrú, Y. (2017). Aplicación móvil para la detección de somnolencia de un conductor aplicando visión artificial. *Revista Tecnológica - ESPOL*, 30.

PINE H64. (10 de Diciembre de 2019). *PINE H64*. Obtenido de https://www.pine64.org/pine-h64-ver-b/

Polo, E. M. (2008). *Técnicas de Localización en el Instituto de Investigación en Informática de Albacete.*

RASPBERRY PI FOUNDATION. (01 de Enero de 2020). *Raspberrypi*. Obtenido de https://www.raspberrypi.org/

Rockpi. (20 de Octubre de 2019). *Rockpi*. Obtenido de https://rockpi.org/

Segura, J. L. (2010). *Sistema de auto localización a partir de una red WIFI.*

Soria, C., Carelli, R., Kelly, R., & Zannatha, J. M. (2014). *Control de Robots Cooperativos por Medio de Vision Artificial. XVI Congreso de la Asociación Chilena de Control Automático. .*

V. M. Arévalo, J. González y G. Ambrosio. (2015). *LA LIBRERÍA DE VISIÓN ARTIFICIAL OPENCV*. Madrid.

Vats, E. M., & Garg, E. A. (2012). *Detection and security system for drowsy driver by using artificial neural nerwork technique.* Obtenido de In International Journal of Applied Science and Advance: https://pdfs.semanticscholar.org/5834/ce9ca287d18f6bef7c0360b8280e1747f70a.pdf

ANEXOS

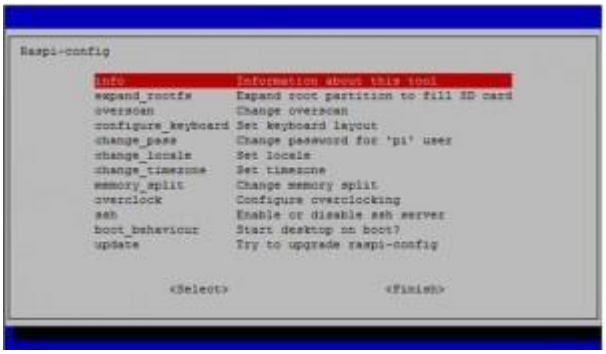
Manual de desarrollo

El instalar un sistema operativo dentro del microrordenador Raspberry pi es diferente a lo que comúnmente se utiliza (Windows, Linux, Unix). Para ello se necesita emplear un equipo aparte del que se está utilizado, el mismo que debe tener una entrada de lectura para tarjetas SD.

Con estos implementos, lo que se procede a hacer es insertar la tarjeta en la computadora y proceder a formatearla, es remendable utilizar algún programa especializado que pueda borrar los archivos de la tarjeta tanto física como lógicamente, dejando al dispositivo de almacenamiento completamente limpio y listo para instalar cualquier sistema operativo sin el menor riesgo. El programa que se utilizó fue uno de descarga gratuita llamado: "Win32 Disk Imager".

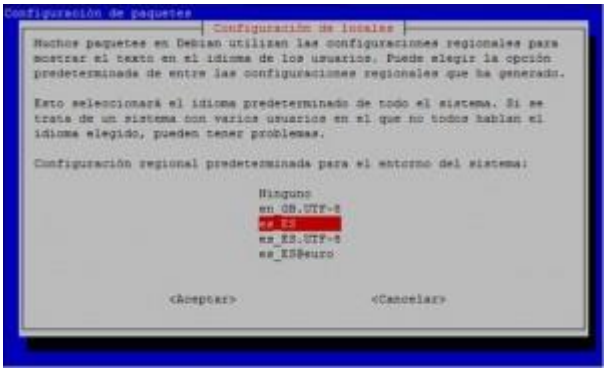
El sistema pre diseñado para los módulos Raspberry pi y con su licencia estándar es *Raspbian*, el cual es una distribución basada en el sistema operativo de código abierto Linux y que ha sido modificado para dar un ambiente ligero al microrordenador. Al ser un sistema que esta licenciado por la misma plataforma que creó el dispositivo, se la puede encontrar de manera gratuita en su página, tomando en cuenta que se actualiza cada cierto tiempo y es recomendable siempre tenerlo en la última versión por las nuevas funciones que le incorporan. Se adjunta el link de la página oficial para su oportuna descarga: "http://www.raspberrypi.org/downloads".

Lo primero que se hace al insertar la tarjeta formateada y encendido el Raspberry Pi, es la configuración del sistema operativo, la cual se la puede realizar en el ambiente gráfico o directamente con la ventana de configuración.

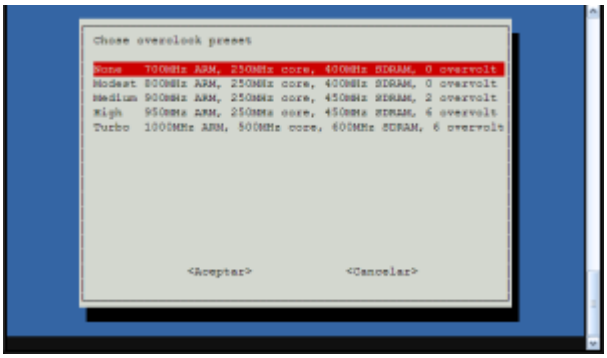


Lo primero que se debe realizar es una expansión de la tarjeta SD con la partición root, esto va a permitir que se utilice toda la capacidad de la tarjeta para poder instalar utilitarios y programas en nuestro dispositivo. Seguido de eso, se va a configurar los parámetros básicos de cualquier distribución de Linux:

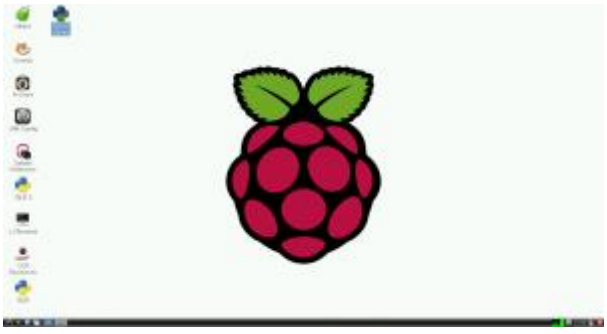
- Si se desea usar un teclado en nuestro dispositivo, se lo debe configurar conectando uno dentro de los puertos USB libres.
- Cambiar el usuario y pass del dispositivo que por defecto vienen como “PI” y “raspberry” respectivamente.
- Cambiar la zona horaria a la que se está estandarizado, en el caso de Ecuador, con la -5 que es la estándar, funcionará la hora normal.
- Con el comando de Overscan se definirá el tamaño del display a utilizar
- El protocolo SSH viene por defecto en activo por si se lo quiere desactivar.
- Si se desea realizar un Update que actualice el sistema operativo, debe estar conectado un cable de red para conectar al internet.
- Si se desea hacer Overclock para aumentar la configuración de defecto del procesador que es de 700Mhz, a una velocidad de la placa que funcionará hasta a 1Ghz en su CPU y hasta 600Mhz su memoria RAM sin ningún tipo de problemas.



Si se activa el modo “Turbo” se presenta un aumento de hasta un 52% de los procesos que realice el módulo a comparación si se lo deja con la configuración que viene por defecto.



Una vez configurado todo el entorno de trabajo del microrordenador, la instalación del sistema operativo transcurrirá sin ningún otro impedimento. Al igual que cualquier distribución de Linux, y gracias a la velocidad con la que cuenta el módulo, la instalación se demorará un tiempo relativamente corto hasta que se pueda entrar en el entorno gráfico para la utilización al 100% del prototipo.



En la instalación normal, se va a encontrar con varias librerías que ya han sido cargadas por default, así como una configuración de sonido y video estándar. Lo siguiente a realizar es la

preparación del ambiente de programación Python, que es donde se va a incorporar nuestro programa para que pueda correr en el módulo a modo portátil.

La ventaja de utilizar la versión más reciente del portal de RaspBerry, es que cuenta con una distribución incorporada del lenguaje de programación Python, el cual es pre instalado en el sistema operativo, del mismo modo que el IDLE, el cual es un entorno de desarrollo integrado para poder programar bajo el lenguaje Python.

La instalación de las librerías como OpenCv o Dlib se la realiza sin ningún contratiempo, siendo tan común como la instalación en un sistema Linux o su contra parte Windows, cabe recalcar qué, si se cuenta con una conexión a internet, la navegación por los portales de distribución de estas librerías serán de gran ayuda para descargar la versión más reciente del código, caso contrario habría que exportar a la memoria SD el módulo de programación actual.

Manual de usuario

El prototipo diseñado para la detección de somnolencia, es un dispositivo electrónico portátil, el cual se encapsula en una caja sellada de un tamaño no mayor al de un celular, para que de esta manera sea trasladado con un índice de ruptura mínimo.

Como características básicas para la manipulación del dispositivo son las siguientes:

- Mantener a una temperatura máxima de hasta 55 °C.
- Evitar poner al prototipo en lugares con mucha humedad o con constante fluctuación de polvo
- Usar solo el cargador que viene de fábrica.
- Evitar la variación de voltaje al conectar el dispositivo
- Evitar golpes o caídas

Teniendo las seguridades necesarias, se dispone a enumerar los pasos para el uso del prototipo:

1. **Desempaque**

Por defecto el prototipo viene en una caja para su transporte, al sacarlo, tener el cuidado pertinente para no arrojarlo accidentalmente. De igual modo se debe ser precavido al manipularlo, por la cámara externa, la cual se encuentra expuesta y es vulnerable a daños.

2. **Instalación**

El prototipo debe ser colocado en un lugar que no interfiera al conductor, como puede ser en la guantera o a un costado de la palanca de cambios, la cámara debe estar en una zona plana de preferencia. Una vez encontrado el sitio idóneo, se despega las cintas adhesivas del dispositivo y se lo coloca en el lugar seleccionado, teniendo en cuenta que la cámara debe apuntar directamente al rostro del usuario al volante.

3. Encendido

El cable de energía que viene incorporado en la caja de accesorios, es el único que se debe usar para conectar la energía del prototipo. Al conectar, automáticamente este encenderá. Se deben esperar unos segundos (alrededor de 40) para que carguen todos los archivos del sistema y empiece a correr.

No se requiere ningún tipo de configuración y presionar nada del prototipo, está programado para que el sistema de detección arranque apenas se prenda el dispositivo.

4. Detección

El prototipo está diseñado para que pueda dar dos alarmas, una de modo sonoro y la otra visual al momento de presenciar signos de somnolencia mediante técnicas de escaneado de rostro. Si el conductor está entrando a un estado de somnolencia, el prototipo emitirá un sonido los primeros 20 segundos, así como una luz parpadeante, el cual ira cambiando en modo de alarma hasta que el conductor vuelva a estar concentrado en conducir.

El sistema dejará de sonar automáticamente cuando el conductor vuelva a estar concentrado, por lo que no es necesario el volver a encender el prototipo.

5. **Apagado**

Al ser un prototipo portátil, el mismo se puede apagar desconectado la fuente de energía conectada con anterioridad. No se debe hacer ningún otro paso para poder apagarlo y tampoco sufrirá daños por desconéctalo directamente.

Nota: Al ser este un dispositivo piloto, se cuenta únicamente con una unidad, la cual sirve a modo de pruebas y ejemplo que el sistema y programación de detección de somnolencia es apto para poder implantarlo en masa, lo cual se debe estudiar en futuras investigaciones. Por ello se debe tener un alto grado de cuidado, aunque el prototipo se puede replicar y volver a cargar.

Codigo fuente

```python
#python drowniess_yawn.py --webcam webcam_index

from scipy.spatial import distance as dist
from imutils.video import VideoStream
from imutils import face_utils
from threading import Thread
import numpy as np
import argparse
import imutils
import time
import dlib
import cv2
import os

def alarm(msg):
    global alarm_status
    global alarm_status2
    global saying

    while alarm_status:
        print('call')
        s = 'espeak "'+msg+'"'
        os.system(s)

    if alarm_status2:
        print('call')
        saying = True
        s = 'espeak "' + msg + '"'
        os.system(s)
        saying = False

def eye_aspect_ratio(eye):
    A = dist.euclidean(eye[1], eye[5])
    B = dist.euclidean(eye[2], eye[4])

```

```
    C = dist.euclidean(eye[0], eye[3])

    ear = (A + B) / (2.0 * C)

    return ear

def final_ear(shape):
    (lStart, lEnd) = face_utils.FACIAL_LANDMARKS_IDXS["left_eye"]
    (rStart, rEnd) = face_utils.FACIAL_LANDMARKS_IDXS["right_eye"]

    leftEye = shape[lStart:lEnd]
    rightEye = shape[rStart:rEnd]

    leftEAR = eye_aspect_ratio(leftEye)
    rightEAR = eye_aspect_ratio(rightEye)

    ear = (leftEAR + rightEAR) / 2.0
    return (ear, leftEye, rightEye)

def lip_distance(shape):
    top_lip = shape[50:53]
    top_lip = np.concatenate((top_lip, shape[61:64]))

    low_lip = shape[56:59]
    low_lip = np.concatenate((low_lip, shape[65:68]))

    top_mean = np.mean(top_lip, axis=0)
    low_mean = np.mean(low_lip, axis=0)

    distance = abs(top_mean[1] - low_mean[1])
    return distance


ap = argparse.ArgumentParser()
ap.add_argument("-w", "--webcam", type=int, default=0,
```

```
                help="index of webcam on system")
args = vars(ap.parse_args())

EYE_AR_THRESH = 0.3
EYE_AR_CONSEC_FRAMES = 30
YAWN_THRESH = 20
alarm_status = False
alarm_status2 = False
saying = False
COUNTER = 0

print("-> Loading the predictor and detector...")
#detector = dlib.get_frontal_face_detector()
detector = cv2.CascadeClassifier("C:/Users/daed_/.conda/envs/opencv-env/Lib/site
predictor = dlib.shape_predictor('F:/Extras/Rasperry Pi/Documentos/Documentacion

print("-> Starting Video Stream")
vs = VideoStream(src=args["webcam"]).start()
#vs= VideoStream(usePiCamera=True).start()       //For Raspberry Pi
time.sleep(1.0)

while True:

    frame = vs.read()
    frame = imutils.resize(frame, width=450)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    #rects = detector(gray, 0)
    rects = detector.detectMultiScale(gray, scaleFactor=1.1,
        minNeighbors=5, minSize=(30, 30),
        flags=cv2.CASCADE_SCALE_IMAGE)

    #for rect in rects:
    for (x, y, w, h) in rects:
```

```
    #for rect in rects:
    for (x, y, w, h) in rects:
        rect = dlib.rectangle(int(x), int(y), int(x + w),int(y + h))

        shape = predictor(gray, rect)
        shape = face_utils.shape_to_np(shape)

        eye = final_ear(shape)
        ear = eye[0]
        leftEye = eye [1]
        rightEye = eye[2]

        distance = lip_distance(shape)

        leftEyeHull = cv2.convexHull(leftEye)
        rightEyeHull = cv2.convexHull(rightEye)
        cv2.drawContours(frame, [leftEyeHull], -1, (0, 255, 0), 1)
        cv2.drawContours(frame, [rightEyeHull], -1, (0, 255, 0), 1)

        lip = shape[48:60]
        cv2.drawContours(frame, [lip], -1, (0, 255, 0), 1)

        if ear < EYE_AR_THRESH:
            COUNTER += 1

            if COUNTER >= EYE_AR_CONSEC_FRAMES:
                if alarm_status == False:
                    alarm_status = True
                    t = Thread(target=alarm, args=('wake up sir',))
                    t.deamon = True
                    t.start()

                cv2.putText(frame, "DROWSINESS ALERT!", (10, 30),
                            cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
```

```
                cv2.putText(frame, "DROWSINESS ALERT!", (10, 30),
                            cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)

        else:
            COUNTER = 0
            alarm_status = False

        if (distance > YAWN_THRESH):
                cv2.putText(frame, "Yawn Alert", (10, 30),
                            cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
                if alarm_status2 == False and saying == False:
                    alarm_status2 = True
                    t = Thread(target=alarm, args=('take some fresh air sir',))
                    t.deamon = True
                    t.start()
        else:
            alarm_status2 = False

        cv2.putText(frame, "EAR: {:.2f}".format(ear), (300, 30),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
        cv2.putText(frame, "YAWN: {:.2f}".format(distance), (300, 60),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)


    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1) & 0xFF

    if key == ord("q"):
        break

cv2.destroyAllWindows()
vs.stop()

```