UNIVERSIDAD NACIONAL DE CHIMBORAZO

FACULTAD DE INGENIERÍA

CARRERA DE ELECTRÓNICA Y TELECOMUNICACIONES

Proyecto de Investigación previo a la obtención del título de:

Ingeniero en Electrónica y Telecomunicaciones

TRABAJO DE TITULACIÓN

"IMPLEMENTACIÓN DE UN SISTEMA DE ALERTA PARA LA PREVENCIÓN DE ACCIDENTES MEDIANTE RECONOCIMIENTO DE SEÑALES DE TRÁNSITO Y VISIÓN ARTIFICIAL. "

Autor:

Víctor Edison Cain Guambo

Tutor:

Ing. José Luis Jinez Tapia, Mgs.

Riobamba - Ecuador

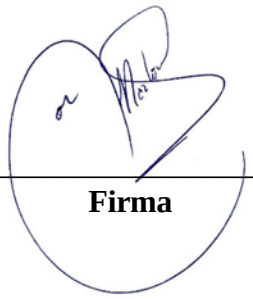
Año 2021

Los miembros del tribunal de graduación del proyecto de investigación de título: **"IMPLEMENTACIÓN DE UN SISTEMA DE ALERTA PARA LA PREVENCIÓN DE ACCIDENTES MEDIANTE RECONOCIMIENTO DE SEÑALES DE TRÁNSITO Y VISIÓN ARTIFICIAL",** presentado por: Victor Edison Cain Guambo, dirigido por: Ing. José Luis Jinez Tapia, Mgs.

Una vez escuchada la defensa oral y revisado el informe final del proyecto de investigación con fines de graduación escrito en la cual se ha constatado el cumplimiento de las observaciones realizadas, remite la presente para uso y custodia en la biblioteca de la Facultad de Ingeniería de la UNACH.

Para constancia de lo expuesto firman.

Ing. Carlos Peñafiel. Mgs.
Presidente del Tribunal **Firma**



Dr. Marlon Basantes. PhD.
Miembro del Tribunal **Firma**

Dr. Leonardo Rentería. PhD.
Miembro del Tribunal **Firma**

DECLARACIÓN EXPRESA DE TUTORÍA

En calidad de tutor del tema de investigación **"IMPLEMENTACIÓN DE UN SISTEMA DE ALERTA PARA LA PREVENCIÓN DE ACCIDENTES MEDIANTE RECONOCIMIENTO DE SEÑALES DE TRÁNSITO Y VISIÓN ARTIFICIAL"** realizado por el Sr. **Victor Edison Cain Guambo**, para optar por el título de Ingeniero en Electrónica y Telecomunicaciones, considero que reúne los requisitos y méritos suficientes para ser sustentada públicamente y evaluada por el jurado examinador que se designe.

Riobamba, Abril 2021

Ing. José Luis Jinez Tapia. Mgs

C.I. 0602899007

TUTOR:

Firma

AUTORÍA DE LA INVESTIGACIÓN

La responsabilidad del contenido de este proyecto de graduación corresponde exclusivamente a: **Victor Edison Cain Guambo, Ing. José Luis Jinez Tapia. Mgs** y el patrimonio intelectual de la misma a la Universidad Nacional de Chimborazo.

Victor Edison Cain Guambo

C.I. 0604499103

DEDICATORIA

Este proyecto va dedicado a mis padres, a mi madre, y a mi difunto padre quien fue mi principal fuente de apoyo, mi aliento cuando más necesitaba de él.

Dedico con todo corazón mi tesis a mi madre, pues sin ella no lo había logrado. Tu bendición a diario a lo largo de mi vida me protege y me lleva por el camino del bien. Por eso te dedico mi trabajo en ofrenda por tu paciencia y amor.

Victor Cain.

AGRADECIMIENTO

Agradezco a Dios, ser divino por darme la vida y guiar mis pasos, a mis padres y hermanos, porque son lo más sagrado que tengo en la vida, por ser los principales motivadores y formadores de lo que ahora soy como persona, sin ustedes y sus consejos no habría llegado hasta donde estoy. Gracias familia.

A mi tutor Ing. José Luis Jinez Tapia. Mgs por haberme brindado todos sus conocimientos en la culminación de la etapa Universitaria.

Victor Cain.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

En el presente trabajo de investigación se desarrolla un sistema de alerta para la prevención de accidentes mediante reconocimiento de señales de tránsito y visión artificial. Este es un sistema multitarea, ya que es capaz de reconocer las señales de tránsito preventivas, determinar la distancia entre el vehículo de prueba y el vehículo que se encuentran frente a él, detectar las líneas de carril y además detectar el nivel de alcohol del conductor.

El hardware para el sistema está conformado por una Raspberry PI a la cual se conectan una cámara y sensores de distancia y alcohol. Se emplea el sistema operativo Raspbian donde se desarrolla el algoritmo para el reconocimiento de imágenes y emisión de alertas. Se emplea las librerías de visión por computadora OpenCV para el procesamiento de las imágenes.

El sistema adquiere la información de las señales de tránsito y de las líneas de carril mediante la cámara conectada a la Raspberry PI. Además, para el reconocimiento de las señales preventivas se emplea algoritmos de aprendizaje como Haar Cascade, el cual permite reconocer cada señal de una manera más precisa.

Para determinar la distancia entre el vehículo y medir el nivel de alcohol del conductor se utiliza un sensor ultrasónico y un sensor MQ-3 respectivamente. Los valores de los sensores serán procesados utilizando el lenguaje Python y se emitirá una alerta en caso de superar los niveles establecidos.

Palabras claves: Raspberry PI, Sensores, Sistema de Alerta, Visión Artificial, Python

ABSTRACT

The aim of this research is to develop a warning system for the prevention of accidents by means of traffic sign recognition and artificial vision. This is a multi-tasking system, because it is capable to recognize preventive traffic signs, determine the distance between the test vehicle and the vehicle in front of it, detect lane lines and also detect the driver's alcohol level. The hardware for the system consists of a Raspberry PI which is connected to a camera and distance and alcohol sensors. The Raspbian operating system is used to develop the algorithm for image recognition and warning emission. The OpenCV computer vision libraries are used for image processing. The system acquires the information of traffic signs and lane lines through the camera connected to the Raspberry PI. In addition, learning algorithms such as Haar Cascade are used for the recognition of warning signs, which allows each sign to be recognized more accurately.To determine the distance between the vehicle and measure the driver's alcohol level, we use an ultrasonic sensor and an MQ-3 sensor, respectively. The sensor values will be processed using Python language and an alarm will be issued in case of exceeding the established levels.

KEYWORDS:

RASPBERRY PI / SENSORS / ARTIFICIAL VISION / PYTHON.

Reviewed by:

Msc. ENRIQUE GUAMBO YEROVI.

ENGLISH PROFESSOR

CI: 060180242

INTRODUCCIÓN

En los actuales periodos la electrónica y los sistemas de ayuda en la conducción están ganando cada vez más importancia, siendo estos componentes esenciales en el transcurso de un determinado viaje o ruta al que el conductor acude. La implementación de estos sistemas, en un sinnúmero de vehículos ya no se realiza por simple comodidad, sino para la reducción de un alto número de accidentes que se producen a diario en las vías, ya sea por el agotamiento o el descuido perceptible de las señales de tránsito, alertando y ayudando al conductor a una mejor percepción a las señales de tránsito, las mismas que indican prevención, información y reglamentación de las vías. (Mateo, 2017)

Existen diversos trabajos realizados correspondientes a la detección de señales de tránsito, los mismos que hacen referencias exclusiva a señales de intersección como lo son: ceda el paso y pare (Flores Calero, Conlago, Yunda , & Aldás, 2018), de igual manera la detección de señales de tránsito reglamentarias en informativas que se encuentran claramente en calles y carreteras de diferentes lugares (Chicaiza, 2017), siendo este sistema de gran aporte para los conductores y ocupantes.

En estos dispositivos de ayuda a la conducción, los sistemas de visión artificial tienen como objetivo adquirir, procesar y analizar imágenes del mundo real, en este caso las señaléticas de prevención en las vías, con el fin de generar información que pueda ser interpretada por una máquina y por el usuario. Uno de los sectores en los cuales se puede integrar la visión artificial en el sector del automovilístico, siendo esta: la detección y reconocimiento de señales de tránsito sobre una fuente de video en tiempo real. (Gamán, 2015).

El presente proyecto de investigación se orienta a la utilización de sistemas de visión por ordenador para realizar un prototipo que sea capaz de detectar y reconocer señales de tránsito preventivas, detección de líneas en la carretera, para que el conductor no pueda abandonar e invadir su carril, distancia entre automóviles para que no se produzca un posible choque, además de un sistema para la detección de nivel de alcohol, todo este procedimiento realizado en tiempo real. Con el objetivo de alertar al conductor a través de una señal sonora y en lo posible hacer más placentero el manejo sobre las vías de nuestra ciudad Riobamba – Ecuador

El siguiente documento está constituido de la siguiente manera:

- ✓ **Capítulo I**: Esta sección se encuentra conformada por los objetivos y el alcance con el que se va a alcanzar la investigación del proyecto planteado.
- ✓ **Capítulo II**: Se describe toda la información relacionada con el proyecto, iniciando con los criterios de señales de tránsito preventivas y horizontales, una descripción de lo que es la Visión Artificial y sus Aplicaciones para finalizar con las características de los elementos o materiales para el proyecto.
- ✓ **Capítulo III**: Este apartado se menciona la metodología que se utilizó para la realización del proyecto de investigación.
- ✓ **Capítulo IV:** se muestra los resultados que se obtuvo del proyecto de investigación.
- ✓ **Capítulo V:** Esta sección se finaliza con las Conclusiones y recomendaciones.

CAPÍTULO I

1.1. PLANTEAMIENTO DEL PROBLEMA

La Organización Mundial de la Salud y el informe sobre la situación mundial de la seguridad vial indican que cada año mueren cerca de 1.3 millones de personas del mundo entero, y entre 20 y 50 millones padecen traumatismos no mortales. Los accidentes de tránsito son una de las principales causas de muerte en todos los grupos etarios comprendidos entre 15 y 29 años (Peden, 2009).

En Sudamérica, el Ecuador es el segundo país con el mayor índice de muertes por accidentes de tránsito. Esta cifra la corrobora el Instituto Nacional de Estadísticas y Censo (INEC) en conjunto con la Agencia Nacional de Tránsito (ANT), según datos del (SOAT) cada 10 minutos existe un accidente de tránsito y según la gravedad del accidente una persona muera en las vías.

En la provincia de Chimborazo según la Dirección Nacional de Tránsito y Seguridad Vial, se registra un total de 3528 accidentes de tránsito, con un porcentaje del 7.87% de fallecidos, siendo como causas principales: la distracción, estado de embriaguez, etc. (ANT Chimborazo, 2015)

Este proyecto está enfocado en implementar un sistema que permita detectar señales de tránsito preventivas, las mismas que son obviadas por la mayoría de los conductores. El dispositivo permite mediante alertas un posible accidente de tránsito, ocasionados por efectos como: distracción, irrespeto (distancia respectiva entre vehículos, invasión de carriles) y alcohol, mediante un sistema de alerta automática y con la ayuda de tecnología como lo es visión artificial, que bien es cierto servirá de mucha utilidad para detectar las diferentes señales preventivas principales que se encuentran en la vía.

A diferencia de otros proyectos, se trata de implementar varios sistemas en un mismo dispositivo, como la detección de señales de tránsito preventivas y horizontales, un sistema que

permita la distancia entre vehículos, además la detección de nivel alcohol del conductor esta investigación se realizó en la ciudad de Riobamba - Ecuador

1.2. Justificación del problema

Implementar el sistema de alertas para la prevención de accidentes de tránsito, será capaz de realizar varias funciones, las cuales brindarán mayor seguridad al conductor. Este es un sistema multitarea capaz de: reconocer las señales de tránsito preventivas que se encuentran en la ciudad de Riobamba - Ecuador, determinar la distancia entre el vehículo de prueba y el vehículo que se encuentran frente a él, detectar las líneas de carril, detectar el nivel de alcohol del conductor y alertar al conductor para de esta manera evitar un accidente, pérdida de vidas y daño a la propiedad o su entorno.

1.3. OBJETIVOS

1.3.1. GENERAL

- Implementar un sistema de alerta automática para la prevención de accidentes mediante el reconocimiento de señales con visión artificial.

1.3.2. ESPECÍFICOS

- Desarrollar un código de programación que permita detectar las señales de tránsito preventivas y horizontales mediante visión artificial.
- Elaborar un sistema de seguridad que permita la detección del nivel de alcohol en el conductor y distancia entre vehículos.
- Implementar el prototipo de alerta automática para la prevención de accidentes de tránsito.

CAPÍTULO II

2.1. ESTADO DEL ARTE RELACIONADO A LA TEMATICA

2.1.1. ANTECEDENTES

En la actualidad el medio de transporte más utilizado a nivel mundial es el vehículo, se debe a la comodidad que ofrece a los usuarios frente a otros medios de transporte. En los últimos años se han desarrollado múltiples sistemas avanzados, para una conducción mucho más segura y reducir el número de accidentes de tránsito. Este sistema de reconocimiento de señales de tránsito se está incorporando a los nuevos modelos de vehículos. Estos sistemas permiten, entre muchas otras cosas, avisar al conductor del estado de la carretera, detectar obstáculos en la vía y salidas de carril y detectar las señales de tránsito que se encuentre en la carretera.

En la Universidad Simón Bolívar de Barranquilla, se realizó un trabajo que propone el reconocimiento de un objeto a través de una cámara Web, mediante el reconocimiento de patrones determine si un objeto coincida con otro, previamente almacenados en una base de datos. (Estarita, Jiménez, Brochero, Escobar, & Moreno, 2018)

En el Instituto Tecnológico y de Estudios Superiores de Occidente, desarrolla un programa para el reconocimiento facial y la detección de una señal de tránsito, en la cual procesa diferentes algoritmos para procesar imágenes, entrenamiento del clasificador en cascada con imágenes positivas y negativas. (Lara Nuñez & Mares Ruiz, 2016)

En la Universidad Autónoma de Occidente, se desarrolla un sistema de reconocimiento basado en patrones para la detección de tráfico vehicular, enfocado a las partes que existe mayor tráfico vehicular. (Arce Millan & Vasco Alzate, 2018)

El proyecto realizado por (Villalón, Torres, & Flores, 2017) investiga un sistema para la detección y reconocimiento de señales de tránsito, aledañas a intersecciones viales y un análisis, para conocer su capacidad de detección en función de la distancia el método que propone el investigador es la segmentación por color.

El trabajo realizado por (Cruzado Hernando, 2015) se basa en la detección de señales de tránsito que capte la cámara, y la segunda en el reconocimiento del mayor número posible. Todo esto se hará con la ayuda de las funciones que ya vienen implementadas en las OpenCv y sobre el sistema operativo ROS (Robot Operating System).

En la Universidad Técnica de Ambato se realiza un sistema de asistencia al conductor empleando visión artificial en vehículos de transporte público, propone la utilización de un método de reconocimiento de patrones empleando visión artificial, se emplea las técnicas de Haar Cascade para la detección de obstáculos, peatones, líneas de carril y la activación de alertas sonoras. (Pico Aponte, 2019)

En la Universidad Politécnica de Madrid se propone la utilización de métodos de visión artificial para PC como apoyo en la automoción, lo que se realiza en este trabajo de investigación es la detección de líneas de carril y vehículos utilizando los métodos más comunes como son: la transformada Hough, Canny, Haar Cascade. (Barba Guamán, 2015)

En la Universidad Técnica de Ambato se propone la realización de un sistema electrónico de alerta automática para el reconocimiento de señales de tránsito reglamentarias, preventivas e informativas en la ciudad de “Ambato”, utiliza los métodos de Viola y Jones para la detección de las señales en tiempo real. (Chicaiza, 2017)

2.2. FUNDAMENTACIÓN TEÓRICA.

2.2.1. Accidente de tránsito

Un accidente de tránsito es un suceso eventual que altera la actividad de personas, vehículos y usuarios viales que se encuentra en una carretera alterando el orden regular y ocasionando daños Cada año se pierden aproximadamente 1.35 millones de vidas como consecuencia de los accidentes de tránsito. Entre 20 millones y 50 millones de personas sufren traumatismos no mortales, y muchos de esos traumatismos provocan una discapacidad alrededor de 3500 personas fallecen en las carreteras. (Organización Mundial de la Salud, 2018)

2.2.1.1. Factores que intervienen en los accidentes de transito

En un accidente de tránsito intervienen tres factores los cuales se pueden presentar en los diversos escenarios que son:

1) Factor Humano. - los accidentes de tránsito son ocasionados normalmente por una acción irresponsable, imprudente o negligente de un conductor o peatón estos pueden ser: atropello, arrollamiento, caída de pasajero.
2) Factor mecánico del vehículo. - los accidentes de tránsito son ocasionados por los fallos o problemas mecánicos y componentes del vehículo choque estos pueden ser: estrellamiento, colisión entre otros. (Cristina1128, 2015)
3) Factor vial-ambiental. - los accidentes de tránsito pueden ser ocasionados por que la vía no esté en condiciones ya que puede estar mal estado del pavimento, falta de iluminación en la vía, falta de señalización en la vía, condiciones meteorológicas.

2.2.1.2. Principales causas de accidentes de tránsito

Los accidentes de tránsito son un problema que hay que tener en cuenta en nuestra sociedad como se ha visto en el anterior apartado las cuales son provocadas por diversos factores los cuales

se clasifican en tres grupos: factores humanos, factores mecánicos y factores ambientales dependiendo el siniestro estos factores se tendrán en cuenta.

Los factores humanos suelen ser la causa principal de los accidentes de tránsito entre el 70% y 90% de estos siniestros uno de los principales factores es el exceso de velocidad en un 20% de los accidentes mortales los principales factores humanos que pueden aparecer son:

- ✓ Conducir bajo el efecto del alcohol o drogas
- ✓ Conducir con exceso de velocidad de acuerdo con los límites legales.
- ✓ Falta de experiencia en la conducción
- ✓ Adelantar en lugares prohibidos
- ✓ Falta de conocimiento de los reglamentos de conducción, señalética, sanciones, etc.

En lo que concierne a las causas de riesgo mecánico hace referencia a que el vehículo no esté en óptimas condiciones para su circulación. Se toma en cuenta los daños que tiene el vehículo los mismos que pueden provocar un choque debido a que el conductor no podría responder de manera adecuada y evitar el accidente.

Factores ambientales, son los factores climáticos que no permiten al conductor manejar en forma óptima, estos son: la niebla, la lluvia u otras. También se debe tomar en cuenta el estado de la carretera y de las señaléticas. (Mateo, 2017)

2.2.1.3. Conducción bajo los efectos del alcohol

Conducir bajo los efectos del alcohol aumenta el riesgo de un accidente de tránsito, ya que aumenta las probabilidades de que este ocasione la muerte o traumatismos graves. El riesgo de verse involucrado en un accidente tránsito cuando la alcoholemia pasa de los 0.03 g/dl es elevado,

es necesario realizar controles exhaustivos para reducir eficazmente el número de accidentes de tránsito relacionados con la ingestión de bebidas alcohólicas.

Tabla1. Grado de Alcoholemia

Alcoholemia	**Nivel de dificultad en el tránsito**	**Efectos en la persona**	**Nivel de riesgo**
0.0	Sin dificultad	Dominio pleno de facultades para circular libremente en el tránsito.	Nulo
0.3	Moderado	Disminuye la capacidad de atender a situaciones de peligro. La respuesta a las mismas se comienza a lentificar y se hace confusa.	Bajo
0.5	Moderado a severo	Se produce la visión con dificultad de enfoque y eso ocasiona falta de atención a las señales de tránsito que no pueden ser percibidas adecuadamente.	Alto
0.8	Severo	La motricidad se ve afectada, se retardan los movimientos. Manejo agresivo y temerario por impulsos sin razonar.	Alto
1.5	Critico (no conduzca)	Estado de embriaguez importante reflejos alterados y reacción lenta e imprecisa.	Muy alto
2.5	Critico (no conduzca)	Ebriedad completa. La persona parece como “narcotizado” y confuso y le es imposible tomar decisiones con certeza.	Severo
3.0	Critico (no conduzca)	Ebriedad profunda se pierde paulatinamente la conciencia como antesala de coma y principio de riesgo de muerte.	Extremo

Fuente. (Casanova Vásquez, 2014)

2.2.2. Tipos de señales de tránsito en el Ecuador

Las señales de tránsito tienen como objetivo el de advertir a los usuarios sobre anomalías en las carreteras, regular el tránsito e informar sobre aspectos relevantes que intervienen en la circulación vial. Existe un gran número de señales de tránsito vigentes en la normativa ecuatoriana, estas son: manuales, luminosas, acústicas, verticales y horizontales

Las señales de tránsito preventivas son las que se utilizan para ayudar al movimiento seguro y ordenado tanto de personas como vehículos. Estas señales proporcionan información que pueda necesitar el conductor, previenen de peligros que no pueden ser muy evidentes o muestran información acerca de rutas, direcciones, destinos y puntos de interés. Las señales empleadas para transmitir información están formadas por la combinación de un mensaje, una forma y un color. El mensaje de la señal de tránsito puede ser una leyenda, un símbolo o un conjunto de los colores. (INEN, 2011)

2.2.2.1. Señales de tránsito horizontales

Las señales de tránsito horizontales o marcas sobre el pavimento sirven para prevenir, guiar y establecer un excelente medio de señalización para el conductor sin distraer su vista sobre el camino.

Figura 1. Señales Horizontales

Fuente. (educavial, 2015)

2.2.2.2. Señales Regulatorias

Las señales regulatorias indican a los usuarios de las vías las preferencias en el uso de las mismas, así como las prohibiciones, restricciones, obligaciones y autorizaciones existentes, cuyo incumplimiento constituye una contravención de tránsito. Estas señales son de color rojo.

Figura 2. Señales Regulatorias

Fuente. (master autoescuela, s.f.)

2.2.2.3. Señales Preventivas

Se utilizan para alertar a los conductores de potenciales peligros que se encuentren en la vía. Indican la necesidad de tomar precauciones y requieren la reducción de la velocidad de circulación o de realizar alguna otra maniobra. Estas señales son de color amarillo.

Figura 3. Señales Preventivas

Fuente. (agermoso, 2015)

2.2.2.4. Señales de Información Vial

Tienen como propósito orientar y guiar a los usuarios, indicando la información necesaria para que puedan llegar a sus destinos de la forma más simple, segura y directa posible. Los colores de estas señales son: azul, verde y naranja.

Figura 4. Señales de información vial

Fuente. (Educación Vial, 2018)

2.2.3. Asistencia a la conducción (Sistemas Avanzados de Asistencia al Conductor) (ADAS)

Estos sistemas están basados en sensores como: ultrasónicos, radares o cámaras de video que detectan las inmediaciones del vehículo ayudando al conductor hacer más cómoda la conducción o en situaciones críticas ayudan actuar de forma rápida y segura. (Mendieta, 2013)

2.2.3.1. Sistema de advertencia de abandono de carril

Advierte al conductor cuando el vehículo comienza a salirse de su carril en autopistas y carreteras principales. Esta tecnología evita accidentes ya que mediante un aviso el conductor podrá enderezar el coche para retomar el carril.

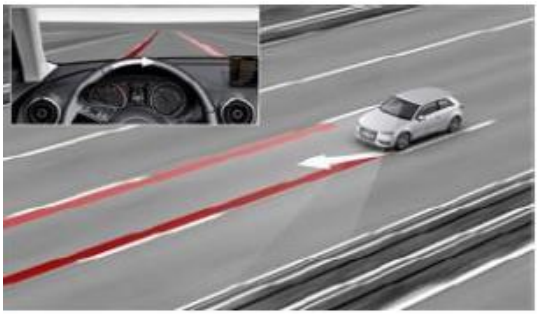


Figura 5. Abandono de carril

Fuente. (Mendieta V. A., 2013)

2.2.3.2. Distancia entre Autos

Son sistemas que mediante sensores permite al conductor tener una distancia de seguridad adecuada entre vehículos para no colisionar con el otro automóvil. Según el Reglamento de la ley de transporte terrestre y tránsito y seguridad vial existen diversas distancias en la cuales el cerebro puede reaccionar ante una posible eventualidad.

En las áreas urbanas se debe tener una distancia de 3 metros con el vehículo que lo antecede en el mismo carril, y una distancia lateral de 1.5 metros. En cambio, en las áreas rurales esta distancia estará determinada por la velocidad, estado del vehículo, condiciones ambientales, topografía de la vía y el tránsito existente al momento de la circulación.

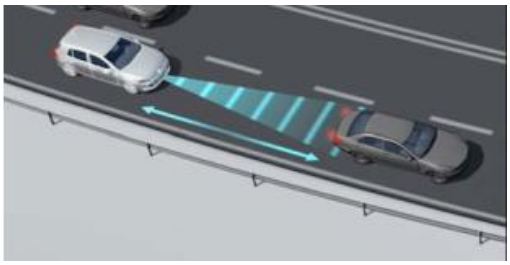


Figura 6. Distancia entre autos

Fuente. (Mendieta V. A., 2013)

2.2.4. Visión Artificial

La visión artificial también conocida como visión por computador es un subcampo de la inteligencia artificial. El propósito de la visión artificial es programar un computador para que entienda una escena o las características de la imagen.

La visión artificial la componen un conjunto de procesos destinados a realizar el análisis de imágenes. Estos procesos son: captación de imágenes, memorización de la información, procesado e interpretación de los resultados.

Con la visión artificial se pueden:

- Automatizar tareas repetitivas de inspección realizadas por operadores.
- Realizar controles de calidad de productos que no era posible verificar por métodos tradicionales.
- Realizar inspecciones de objetos sin contacto físico.

- Realizar inspección del 100% de la producción (calidad total) a gran velocidad.
- Reducir el tiempo de ciclo en procesos automatizados.
- Realizar inspecciones en procesos donde existen diversidad de piezas con cambios frecuentes de producción.

2.2.4.1. Importancia de la visión artificial

La visión artificial ha permitido el desarrollo de nuevas tecnologías en áreas económicas tales como elaboraciones de equipos, alimentos, medicamentos y del mismo modo ha pasado a ser parte inseparable de los sistemas de vigilancia y protección, se ha incrementado su eficacia para el reconocimiento de individuos, animales u objetos que se encuentren en el entorno donde se aplique la visión artificial. (Rueda Valencia, 2017)

2.2.4.2. Aplicaciones de la visión artificial

- **Reconocimiento óptico de caracteres.** Emular la capacidad del ojo humano para reconocer objetos como reconocimiento de placas de vehículos, señales de tránsito, etc.
- **Inspección y control de calidad.** Inspección visual y automática de un producto como partes de vehículos, circuitos, transistores donde se busca algún defecto.
- **Imágenes médicas.** Procesamiento de imágenes orientadas hacia el diagnostico de dolencias o enfermedades, entre ellas: radiografías, resonancias magnéticas, tomografías.
- **Reconocimiento y clasificación.** Reconocimiento de objetos y establecer características entre estos objetos como: tamaño, color y forma.

- **Seguridad vehicular.** Detección de objetos, por ejemplo: peatones en la calle, vehículos, señales de tránsito, bajo ciertas condiciones donde se utilizan técnicas de visión artificial usando radares o cámaras de video.
- **Vigilancia.** Monitoreo de intrusos, análisis de tráfico en autopistas, monitoreo en piscinas para posibles víctimas de ahogamiento.
- **Reconocimiento dactilar y biométrico.** Procesos para la autentificación automática y aplicaciones forenses.
- **Construcción de modelos 3D.** Construcción automática de modelos en 3D para uso de fotografías aéreas como los mapas de Bing. (Barba Guamán, 2015)

2.2.5. Procesamiento de imágenes

El procesamiento de imágenes se ha ido desarrollando según va avanzando la tecnología, es por esta razón que hay diversas funciones que nos permiten procesar una imagen, ya sea para mejorar la calidad, añadir algún efecto o también mezclar las imágenes. Para el procesamiento de imágenes se emplea diversos recursos como: Matlab, eclipse, OpenCV, etc.

2.2.5.1. Procesamiento de imágenes con OpenCv

OpenCV (Biblioteca de visión artificial de código abierto). Es una librería de computación visual creada por Intel, esta librería está disponible en plataformas como: Windows, Linux, Mac, Android, además cuenta con un soporte para diferentes lenguajes como: Python, Java, C/C++ entre otros. La biblioteca OpenCV puede ser usada bajo licencia BSD para proyectos escolares o comerciales, las aplicaciones de esta librería incluyen: la robótica, análisis y procesamiento de imágenes o videos, seguimiento y detección de objetos, detección y reconocimiento de rostros, reconocimiento de placas de vehículos, análisis de formas, reconstrucción 3D, realidad aumentada, y mucho más. (Tutor de Programación, 2017)

2.2.5.2. Filtrado de una imagen en OpenCV

Smoothing Images. Permite difuminar una imagen mediante filtros pasa bajo, OpenCV posee una función $cv2.filter2D$, que realiza una convolución entre un filtro Kernel y una imagen. Esto lo que hace es sacar un promedio entre los pixeles produciendo un desenfoque en la imagen. Esta función es útil para eliminar el ruido.

Image Blurring. El desenfoque de la imagen se logra convolucionando la imagen con un núcleo de filtro de paso bajo para eliminar el ruido. En realidad, elimina el contenido de alta frecuencia (por ejemplo: ruido, bordes) de la imagen, lo que hace que los bordes se vean borrosos cuando se aplica este filtro. (existen técnicas de desenfoque que no desenfocan los bordes). OpenCV aporta especialmente cuatro tipos de métodos de desenfoque.

- ✓ Averaging (Promedio)
- ✓ Gaussian Filtering (Filtrado Gaussiano)
- ✓ Median Filtering (Filtro de media)
- ✓ Bilateral Filtering (Filtrado bilateral)

2.2.5.3. Técnica de Viola y Jones

La primera técnica para la detección de objetos fue propuesta por Pablo Viola y Michael Jones, proporciona tasas de detección que son muy aceptables en tiempo real. Puede ser entrenado para el reconocimiento de varios objetos, este método fue creado para el reconocimiento de caras y puede ser implementado en OpenCV. El reconocimiento facial hoy en día es uno de los más aplicados, se encuentra en edificios, laboratorios o la búsqueda de imágenes en una base de datos. La aportación que realizaron Pablo Viola y Michael Jones fue el uso en cascada ya que se ejecutan uno detrás de otro y cada clasificador tiene un algoritmo de boosting AdaBoost. La detención

mediante cascada permite rechazar una gran parte de la imagen para solo detectar la zona donde se encuentra la cara. (Chicaiza, 2017)

2.2.6. Sensor de Distancia Ultrasónico

El sensor ultrasónico mide la distancia mediante el uso de ondas ultrasónicas. El cabezal emite una onda ultrasónica y recibe la onda reflejada que retorna desde el objeto. Los sensores ultrasónicos miden la distancia al objeto contando el tiempo entre la emisión y la recepción.(Electronilab, s.f.)

Figura 7. Sensor de distancia Ultrasónico

Fuente. (Electronilab, s.f.)

2.2.6.1. Especificaciones

- Dimensiones del circuito: 43 x 20 x 17 mm
- Tensión de alimentación: 5 Vcc
- Frecuencia de trabajo: 40 Khz
- Rango máximo: 4.5 m
- Rango mínimo: 1.7 cm
- Duración del pulso de disparo: 10 µs.
- Duración del pulso eco de salida: 100 - 25000 µs.

2.2.7. Sensor de alcohol MQ3

Es muy sensible al alcohol y de menor sensibilidad a la bencina, también es sensible a gases como GLP, hexano, CO, CH4, pero con sensibilidad muy baja, la cual se puede despreciar si hay poca concentración de estos. (Naylamp Mechatronics, s.f.)

Figura 8. Sensor de Alcohol MQ3

Fuente. (Naylamp Mechatronics, s.f.)

2.2.7.1. Especificaciones

- Voltaje de operación: 5Vcc
- Integrado amplificador LM393 con un umbral mediante potenciómetro.
- 2 pines de salida (salida analógica y salida de nivel TTL).
- Salida de nivel TTL valida de bajo nivel, se puede conectar directamente al microcontrolador
- Salida analógica de: 0~5V
- Temperatura ambiente: -10°C a 65°C.
- Humedad: ≤ 95% RH.

2.2.8. Cámara Raspberry Pi

La placa de la cámara Raspberry Pi de alta definición es compatible con todas las tarjetas Raspberry Pi. La cámara se manipula para tomar video de alta definición, así como imágenes fijas, es fácil de utilizar.

Figura 9. Cámara Raspberry Pi

Fuente. (amazon, s.f.)

2.2.8.1. Especificaciones

- Cámara de 8 megapíxeles capaz de tomar fotografías infrarrojas de 320 x 2464 píxeles
- Captura de video a resoluciones de 1080p30, 720p60 y 640 x 480p90
- Todo el software es compatible con la última versión del sistema operativo Raspbian.
- Longitud del cable: 15 cm
- Aplicaciones: Fotografía infrarroja, fotografía con poca luz, monitoreo del crecimiento de la planta, cámara de seguridad.

2.2.9. Raspberry Pi

La Raspberry pi es un pequeño ordenador que es capaz de realizar las mismas tareas que una Pc de escritorio. Fue desarrollado por la fundación Raspberry Pi (Universidad de Cambridge) en 2006, aunque su comercialización empezó en 2012 se creó con el objeto de estimular la enseñanza

de informática en las escuelas del mundo. La tarjeta tiene diversas entradas y salidas las que el usuario puede utilizar como lo requiera. (Historia de la Informática, 2013)

Figura 10. Tarjeta Raspberry Pi

Fuente. (amazon, s.f.)

2.2.9.1. Especificaciones

- Un procesador gráfico (GPU) Video Core IV
- Un módulo de 512 MB de memoria RAM (aunque originalmente al ser lanzado eran 256 MB).
- Un conector de RJ45 conectado a un integrado lan9512 –jzx de SMSC que nos proporciona conectividad a 10/100 Mbps.
- 2 buses USB 2.0
- Una salida analógica de audio estéreo por Jack de 3.5mm.
- Salida digital de video + audio HDMI
- Salida analógica de video RCA
- Pines de entrada y salida de propósito general
- Conector de alimentación micro USB
- Lector de tarjetas SD

- Un chipset Bradcom BCM2835, que contiene un procesador central (CPU) ARM 1176JZF-S a 700 MHz (el firmware incluye unos modos turbo para que el usuario pueda hacerle overclock de hasta 1 GHz sin perder la garantía).

2.2.10. Pantalla Raspberry Pi

Para visualizar los resultados se requiere de un monitor o una pantalla táctil. Actualmente en el mercado existen pantallas táctiles compatibles con las tarjetas Raspberry Pi.

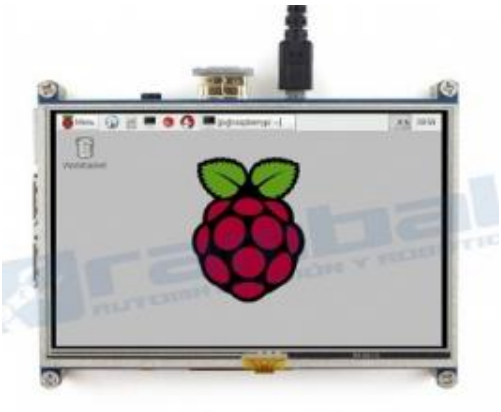


Figura 11. Pantalla raspberry Pi

Fuente. (rambal, s.f.)

2.2.10.1. Especificaciones

- Pantalla táctil Resistente
- Entrada HDMI
- Cable USB para la alimentación de 5V a 1ª
- Frecuencia de actualización: 60Hz
- Tamaño 143mm * 83mm

CAPITULO III

3.1. Metodología de la investigación

3.1.1. Tipo de investigación

3.1.1.1. Investigación Documental

Esta investigación es de tipo documental debido a que se basa en estudios anteriores realizados en tesis y artículos científicos donde desarrollan sistemas de alerta para la prevención de accidentes como son: detección de líneas de carril, reconocimiento de señales y nivel de alcohol en forma independiente. De estos estudios se extrajo las técnicas empleadas para el reconocimiento de imágenes y se desarrolló un sistema multifuncional, debido a que agrupa varios sistemas de alerta en uno solo.

3.1.1.2. Investigación Experimental

La investigación también es de tipo experimental debido a que se implementó el sistema en un vehículo de pruebas para determinar su funcionamiento. Las pruebas se realizaron en dos ambientes: en el día con luz natural y en la noche con luz artificial, además se determinó una velocidad constante del vehículo para el recorrido.

3.2. Métodos de investigación

3.2.1. Método Analítico

Este proyecto se emplea el método analítico, debido a que se analiza los diversos métodos que utiliza la visión artificial y encontrar la técnica más apropiada para el reconocimiento de imágenes, para que software funcione de la mejor manera.

3.3. Técnicas

3.3.1. Observación

En esta investigación se empleó la técnica de observación la cual consiste en recolectar información que se haya producido con respecto a las temáticas que involucran a los parámetros de análisis dentro del estudio a realizar y de esta manera obtener datos que nos ayuden a desarrollar el proyecto de investigación.

3.3.2. Fuentes de recopilación de información

El presente documento de investigación consta de fuentes informativas primarias y secundarias confiables con respecto con el desarrollo del algoritmo del sistema toda esa búsqueda se realizó en libros, tesis, revistas científicas. Todos los datos que se recolectaron tienen su respectiva fuentes o referencias.

3.4. Instrumentos de investigación

En la presente investigación se utiliza los instrumentos de videos tutoriales de sitios web, blogs, datasheets, y artículos científicos que nos ayuden con el algoritmo del sistema.

3.5. Hipótesis

¿La implementación de un sistema de alerta para la prevención de accidentes mediante el reconocimiento de señales de tránsito y visión artificial permitirá al conductor realizar sus viajes de manera segura?

3.5.1. Operacionalización de Variables

Tabla2. Operacionalización de Variables

Variable	**Concepto**	**Indicadores**	**Instrumentos Técnicos**
Variable dependiente Porcentaje efectividad del sistema de alertas	Su propósito es emitir una alarma con el objetivo de tomar precauciones ante un posible riesgo.	Efectividad del Sistema	Observación Análisis estadístico Alertas
Variable independiente Nivel de alcohol Distancia entre vehículos Porcentaje aciertos en el reconocimiento de las señales de tránsito preventivas Porcentaje de aciertos en la detección de líneas de carril	Capacidad del equipo informático de interpretar variables del exterior ya sea imágenes a través del uso de cámaras, o magnitudes físicas mediante sensores	Nivel de alcohol Distancia	Análisis de imágenes Procesamiento de imágenes Sensores Cámara

3.6. Procedimiento

Para el desarrollo del prototipo electrónico de alerta automática, se desarrolla diferentes códigos de programación para el reconocimiento de señales preventivas, líneas de carril y un sistema de seguridad de distancia y alcoholímetro. Este proyecto cuenta con una cámara y sensores conectados a la tarjeta Raspberry Pi.

- ✓ Instalación del software OpenCV en tarjeta Raspberry Pi.

- ✓ Desarrollo de un código de programación para la detección de colores y figuras en OpenCV.
- ✓ Crear un algoritmo para el reconocimiento de señales de tránsito preventivas.
- ✓ Desarrollo de un algoritmo de programación para detectar las líneas de carril.
- ✓ Diseño de un sistema de seguridad para detectar el nivel de alcohol del conductor y distancia entre vehículos.
- ✓ Pruebas de funcionamiento del prototipo para la verificación de su funcionamiento.
- ✓ Análisis de Resultados.

3.6.1. Esquema General del Sistema

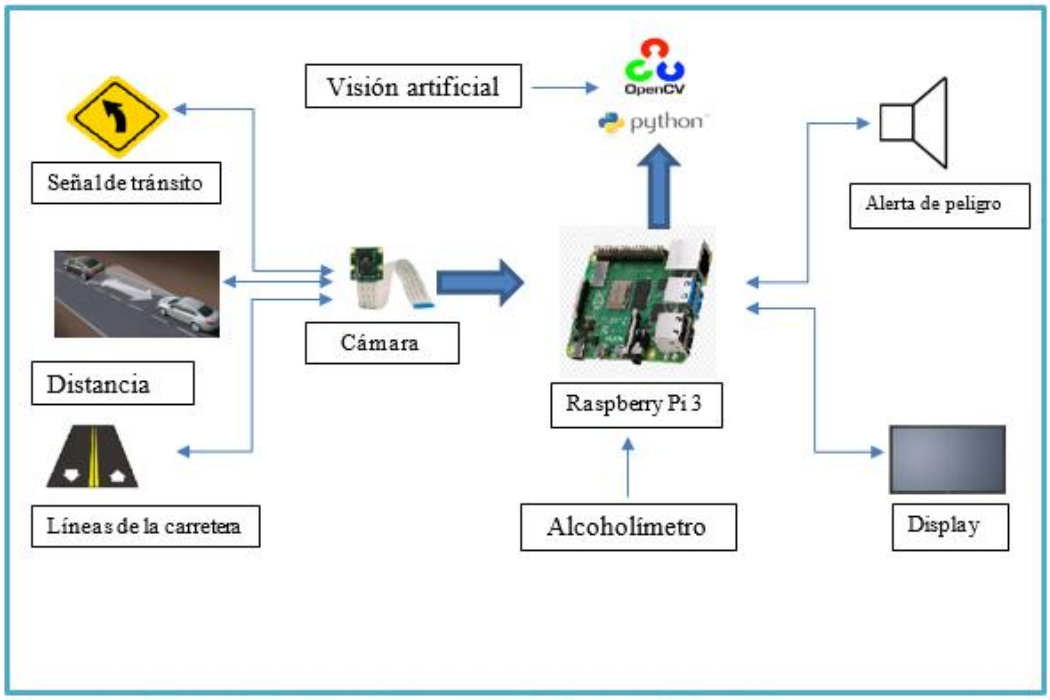


Figura 12. Diagrama de bloques del sistema

Fuente. Autor

3.7. Procedimientos y Análisis

3.7.1. Especificaciones del Software

El software utilizado para este proyecto de investigación se lo detalla a continuación para el uso correcto y poder trabajar con las librerías que se necesita.

Tabla3. Especificaciones de los softwares libres OpenCV y Python

PYTHON	
Sistema Operativo	Multiplataforma
Versión	3.8.5
Desarrollador	Guido van Rossum
Apareció	1991
OPENCV	
Sistema Operativo	GNU/Linux, Mac Os X, Microsoft Windows, Android.
Versión	4.1
Desarrollador	Intel
Apareció	2000

Fuente. Autor

3.7.2. Instalación de la librería OpenCv de Python en la tarjeta Raspberry Pi

OpenCv es una herramienta muy indispensable en el proyecto que vamos a realizar ya que nos facilita al momento de desarrollar nuestro algoritmo Una vez instalado el sistema operativo Raspberry Pi OS en la tarjeta Raspberry Pi procederemos abrir la ventana de comandos para digitalizar **python3** con este comando pretendemos ver que Python tenemos instalado en la tarjeta Raspberry Pi.

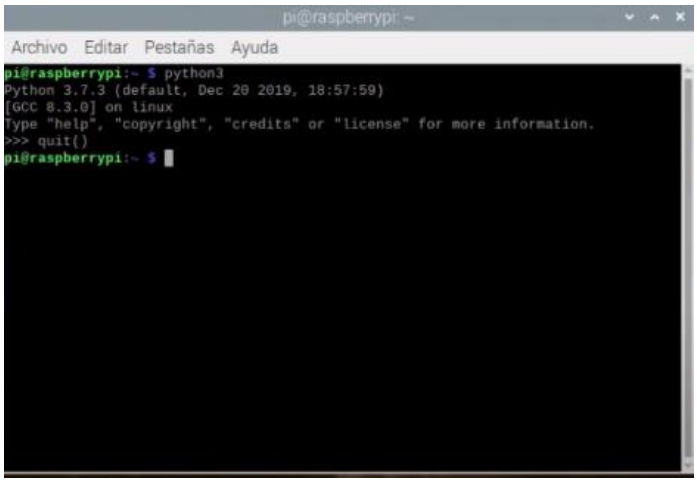


Figura 13. Verificación de python3

Fuente. Autor

Después de verificar que Python está instalado en tarjeta Raspberry Pi se procede a instalar algunos paquetes que lo realizare digitalizando un comando.

- ✓ **sudo apt-get install libhdf5-dev libhdf5-serial-dev libatlas-base-dev libjasper-dev libqtgui4 libqt4-test**

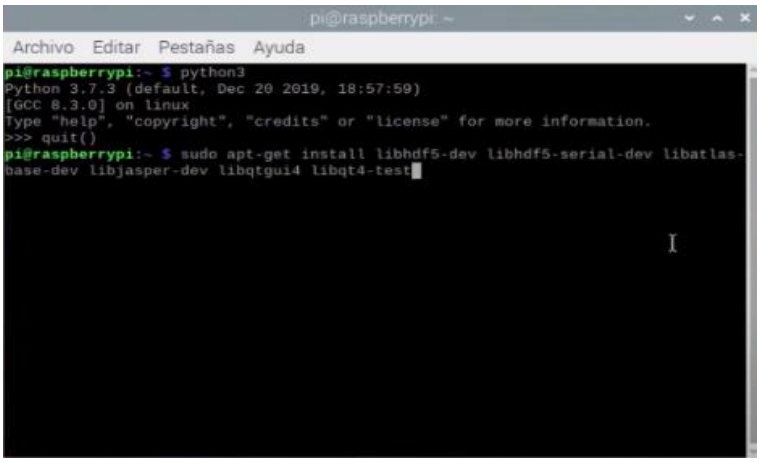


Figura 14. Instalación de Paquetes para OpenCV

Después de a ver realizado lo que antes se mencionó se procede a la instalación de OpenCV 4.1.0 con el siguiente comando.

- ✓ pip3 install OpenCv-contrib-python==4.1.0.25

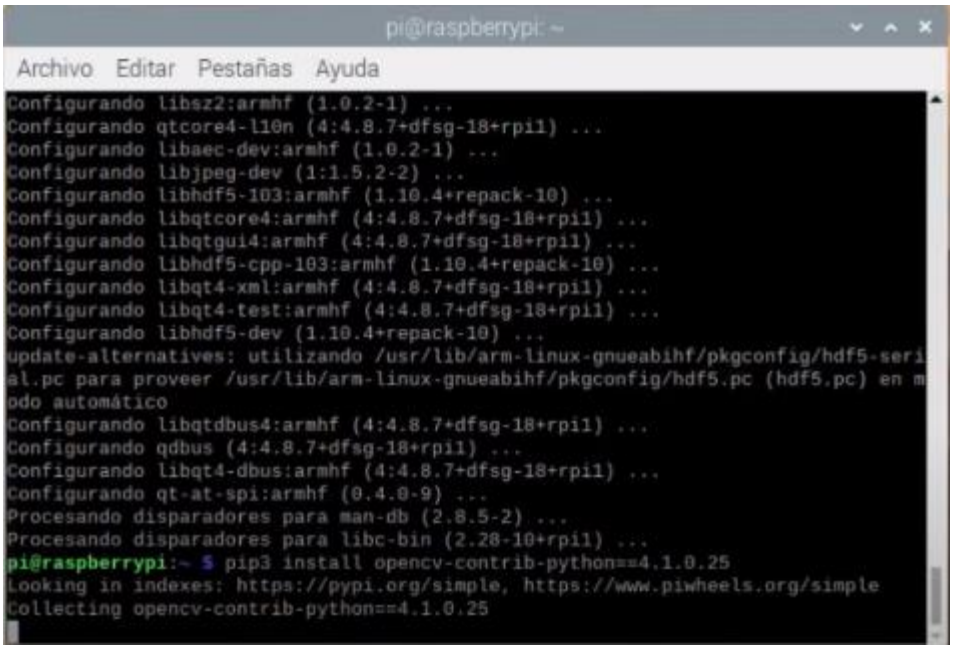


Figura 15. Instalación de OpenCV 4.1.0

Fuente. Autor

Para realizar la verificación que se instaló correctamente digitalizamos el siguiente comando **python3** para después imprimir para que nos muestre la versión de OpenCV que está instalado en la tarjeta Raspberry Pi.

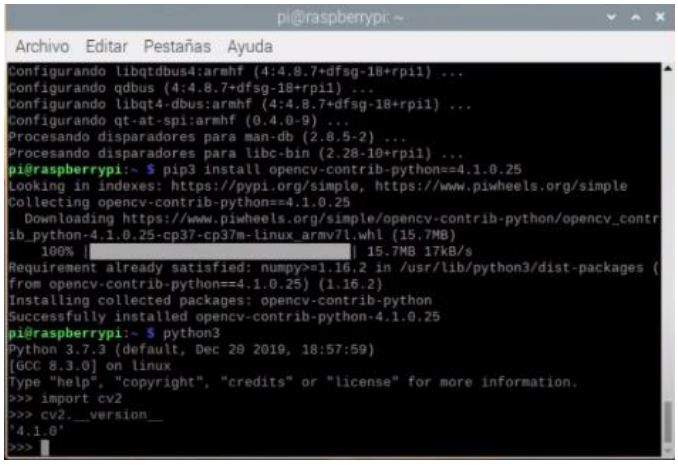


Figura 16. Verificación de la versión de OpenCV

Fuente. Autor

3.7.3. Detección de colores y figuras en OpenCV

Para el desarrollo del siguiente código de programación, nos permitirá el reconocimiento de un color en específico el cual será el amarillo, se desglosa en los siguientes pasos para la detección del color.

1) La imagen o fotograma a procesar
2) Trasformar la imagen de BGR a HSV
3) Determinar el rango del color
4) Visualización

Lo primero que se realiza es la importación de las librerías después la lectura de la imagen, también se lo realiza en tiempo real para lo cual se activa la cámara y se reconoce la imagen.

Figura 17. Imagen reconocida por la cámara

Fuente. Autor

Cuando una imagen es reconocida por OpenCV por defecto la lee en BGR por ello es necesario la trasformación a HSV (Matiz, Saturación, Brillo). Se emplea este espacio de color ya que es más sencillo el reconocimiento del color amarillo

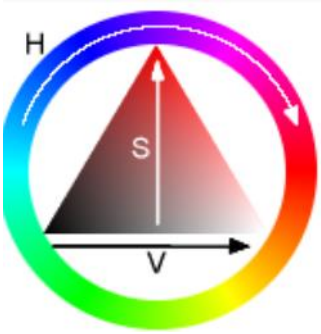


Figura 18. Espacio de color HSV

Fuente. (EcuRed, s.f.)

Se necesita dos valores límites para determinar el rango del color uno inicial y uno final, se busca H ya que S y V toman valores entre 100 a 255 y de 20 a 255 respectivamente, el valor de H toma valores de 15 y 45.

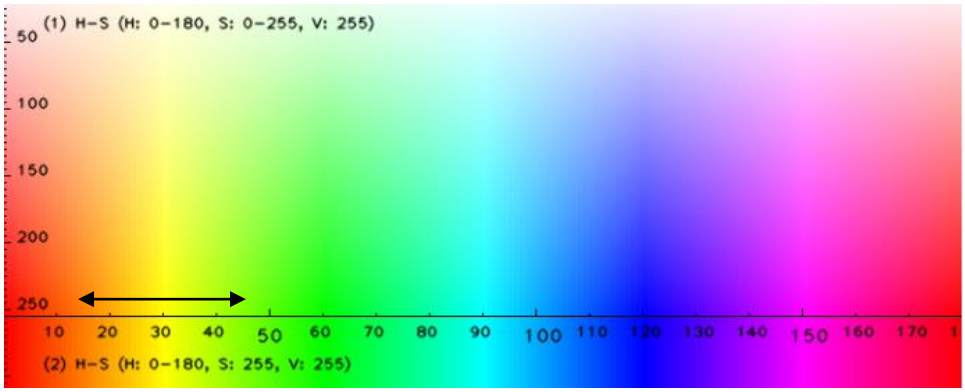


Figura 19. Rango de valores del color amarillo

Fuente. (stackoverflow, 2020)

Y por último se visualiza los resultados de la detección del color amarillo.

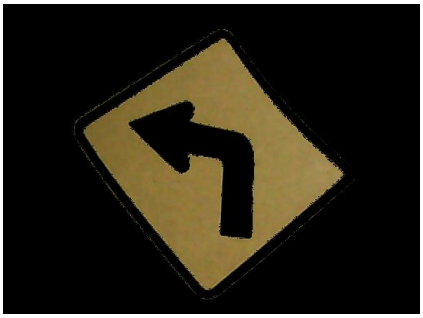


Figura 20. Detección del color amarillo

Fuente. Autor

Detección de figuras.

Lo siguiente que se realizara es la detección de bordes canny la cual nos sirve para la detección de la figura geométrica.

- ✓ Para la detección de bordes canny se siguen los siguientes pasos:
- ✓ Filtración de la imagen, utilizando el filtro gaussiano
- ✓ Cálculo de detección de bordes Canny
- ✓ Supresión de pixeles fuera del borde
- ✓ Aplicar umbral por histéresis

Las operaciones que se realiza para el filtrado es la convolución que consiste en ir recorriendo pixel a pixel una imagen con una máscara de NxN, en donde N es el número de pixeles con el que se va a trabajar. La imagen a tratar se le convoluciona con el filtro gaussiano, la convolución se realiza en las direcciones x, y.

$$G_x \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad G_y \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Una vez que se realiza la convolución de la imagen con el filtro gaussiano para eliminar el ruido se calcula el gradiente de la imagen suavizada, para determinar los pixeles donde se produce la máxima variación, se determina la dirección del vector gradiente.

$$G = \sqrt{{G_x}^2 + {G_y}^2}$$

$$\theta = arctan\left(\frac{G_x}{G_y}\right)$$

Una técnica utilizada para la supresión es la no máxima la cual permite adelgazar los bordes basándose en el gradiente. El no máxima elimina los pixeles que no son considerados como parte de un borde. Solo las líneas finas permanecerán en la imagen

Para la detección de bordes canny, se utiliza la umbralización por histéresis el cual se centra en tomar como referencia 2 umbrales, uno máximo y otro mínimo esto es para determinar si un pixel forma parte de un borde.

```
import cv2
image = cv2.imread('figurasColores.png')
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
canny = cv2.Canny(gray, 10, 150)
canny = cv2.dilate(canny, None, iterations=1)
canny = cv2.erode(canny, None, iterations=1)
#_, th = cv2.threshold(gray, 10, 255, cv2.THRESH_BINARY)
#_,cnts,_ = cv2.findContours(canny, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
cnts,_ = cv2.findContours(canny, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)# O
#cv2.drawContours(image, cnts, -1, (0,255,0), 2)
for c in cnts:
  epsilon = 0.01*cv2.arcLength(c,True)
  approx = cv2.approxPolyDP(c,epsilon,True)
  #print(len(approx))
  x,y,w,h = cv2.boundingRect(approx)
  if len(approx)==3:
    cv2.putText(image,'Triangulo', (x,y-5),1,1.5,(0,255,0),2)
  if len(approx)==4:
    aspect_ratio = float(w)/h
    print('aspect_ratio= ', aspect_ratio)
    if aspect_ratio == 1:
      cv2.putText(image,'Cuadrado', (x,y-5),1,1.5,(0,255,0),2)
    else:
      cv2.putText(image,'Rectangulo', (x,y-5),1,1.5,(0,255,0),2)
  if len(approx)==5:
    cv2.putText(image,'Pentagono', (x,y-5),1,1.5,(0,255,0),2)
  if len(approx)==6:
    cv2.putText(image,'Hexagono', (x,y-5),1,1.5,(0,255,0),2)
  if len(approx)>10:
    cv2.putText(image,'Circulo', (x,y-5),1,1.5,(0,255,0),2)
  cv2.drawContours(image, [approx], 0, (0,255,0),2)
  cv2.imshow('image',image)
  cv2.waitKey(0)
```

Figura 21. Código de programación de detención de figuras

Fuente. Autor

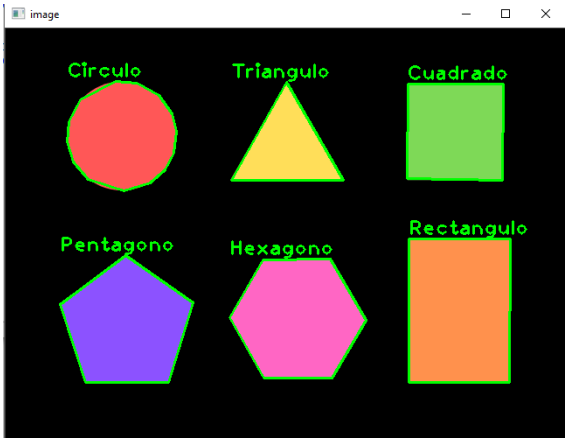


Figura 22. Reconocimiento de las figuras geométricas

Fuente. Autor

3.7.4. Detección de objetos mediante Haar Cascades en Python

La aplicación del método de Haar Cascade se divide en dos partes:

- ✓ Cálculo de características de Haar
- ✓ Creación de imágenes integrales

Para aplicar el método de Haar Cascade se requiere una gran cantidad de imágenes positivas como negativas.

Cálculo de características de Haar

Una operación Haar es el cálculo que se realiza en regiones rectangulares, ubicadas esencialmente en una ventana en específico, el cálculo que se realiza es la suma de las intensidades de pixel de cada región y la diferencia entre las sumas.

$$Valor = \sum(suma\ de\ pixeles\ en\ la\ region\ oscura)$$

$$- \sum(suma\ de\ pixeles\ en\ la\ region\ blanca)$$

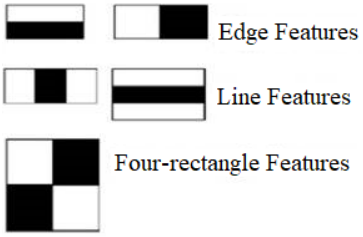


Figura 23. Clasificadores Haar

Fuente. (Mujtaba, 2020)

Creación de imágenes integrales

Las imágenes esenciales aceleran el cálculo de las características Haar, en lugar de calcular en cada pixel la imagen lo que se realiza es la creación de sub-rectángulos y crea referencias de matrices.

$$ii(x,y) = \sum_{x' \le x, y' \le y} i(x',y')$$

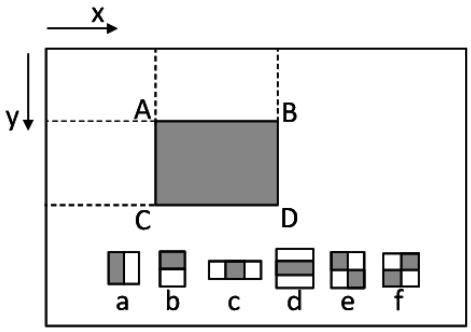
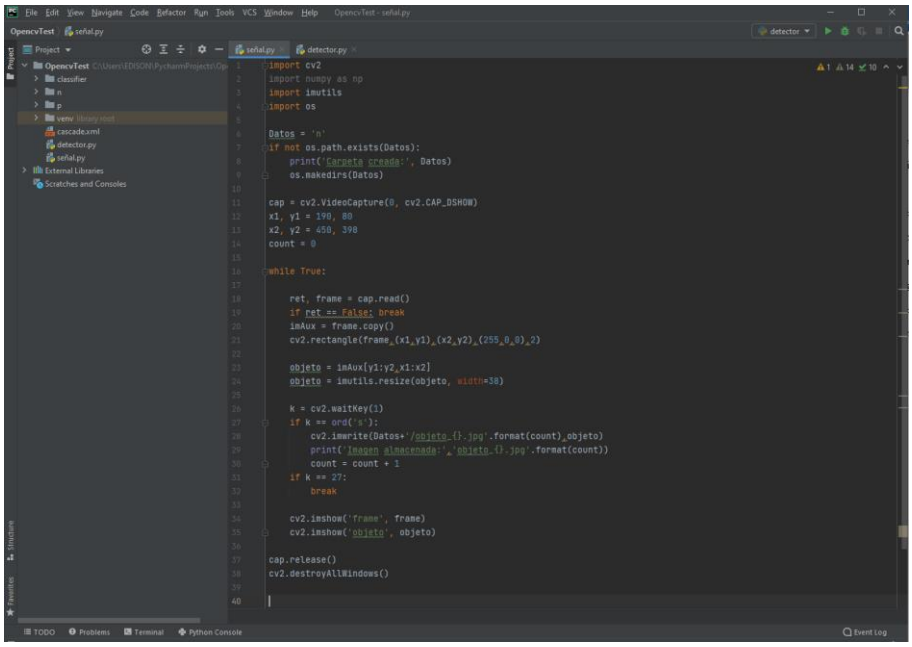


Figura 24. Creación de la imagen integral

Fuente. Autor

Para el reconocimiento de las señales de tránsito preventivas se empleó el método de Haar Cascade. Se realizó una base de datos conformada por imágenes positivas (imágenes que contengan señales de tránsito preventivas) e imágenes negativas (imágenes que no contengan las señales). Las imágenes positivas y negativas se obtuvieron empleando el código mostrado en la figura 25.



```
import cv2
import numpy as np
import imutils
import os

Datos = 'n'
if not os.path.exists(Datos):
    print('Carpeta creada:', Datos)
    os.makedirs(Datos)

cap = cv2.VideoCapture(0, cv2.CAP_DSHOW)
x1, y1 = 190, 80
x2, y2 = 450, 398
count = 0

while True:

    ret, frame = cap.read()
    if ret == False: break
    imAux = frame.copy()
    cv2.rectangle(frame, (x1,y1), (x2,y2), (255,0,0), 2)

    objeto = imAux[y1:y2,x1:x2]
    objeto = imutils.resize(objeto, width=38)

    k = cv2.waitKey(1)
    if k == ord('s'):
        cv2.imwrite(Datos+'/objeto_{}.jpg'.format(count), objeto)
        print('Imagen almacenada:','objeto_{}.jpg'.format(count))
        count = count + 1
    if k == 27:
        break

    cv2.imshow('frame', frame)
    cv2.imshow('objeto', objeto)

cap.release()
cv2.destroyAllWindows()
```

Figura 25. Código de programación para la obtención de imágenes positivas y negativas

Fuente. Autor

Después de obtener la base datos de las imágenes se entrenó el clasificador, el cual fue utilizado para reconocer las señales de tránsito preventivas empleando el método de Haar Cascade. Para entrenar el clasificador se empleó el software Cascade Trainger Gui.

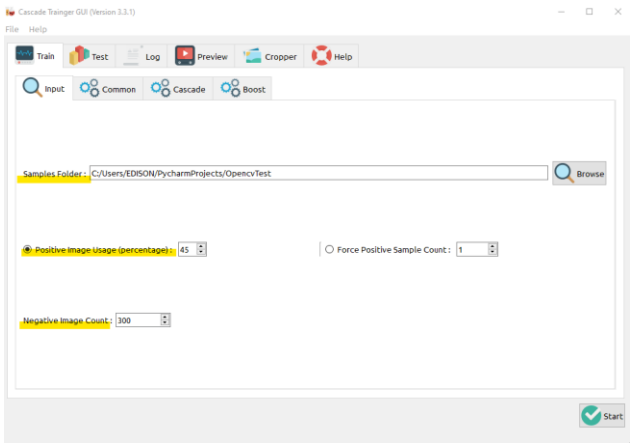


Figura 26. Programa Cascade Trainger Gui

Fuente. Autor

En la ventana que se muestra en la figura 26 se procede a configurar tres aspectos fundamentales para el entrenamiento que son:

- ✓ Abrir la ubicación donde se encuentra la carpeta de las imágenes positivas y negativas.
- ✓ Especificar el número de imágenes positivas que se obtuvo en este caso fueron 45.
- ✓ Insertar el número de imágenes negativas que se obtuvo en nuestro caso fueron 300.

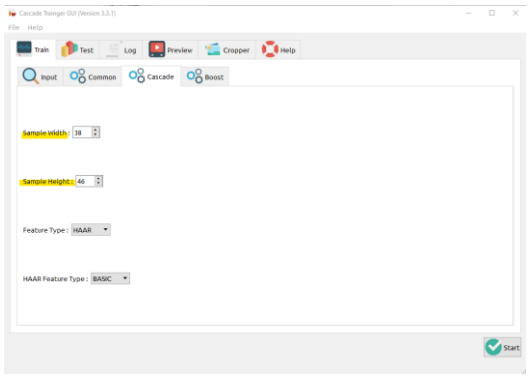


Figura 27. Cascade Trainger Gui, Cascade

Fuente. Autor

Una vez realizado el entrenamiento se genera una carpeta llamada clasiffier, donde está ubicado el clasificador nombrado en este caso como cascade.xml.

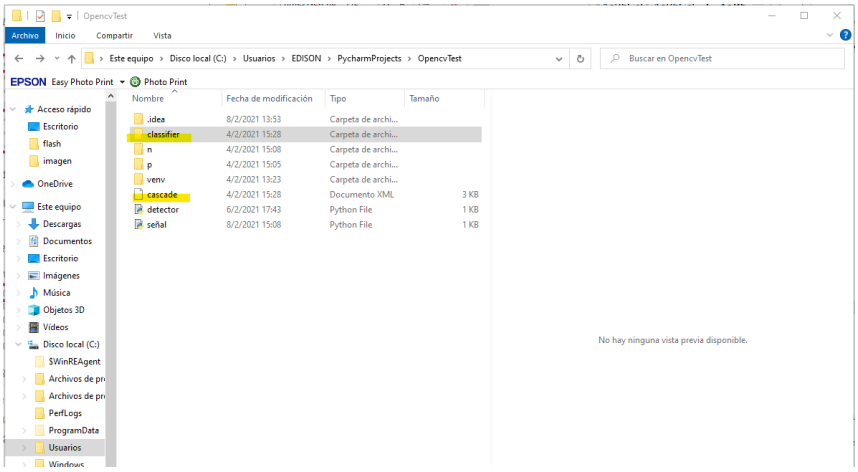


Figura 28. Archivo cascade.xml

Fuente. Autor

Empleando este clasificador se construyó el código de programación para detectar las señales de tránsito preventivas en tiempo real.

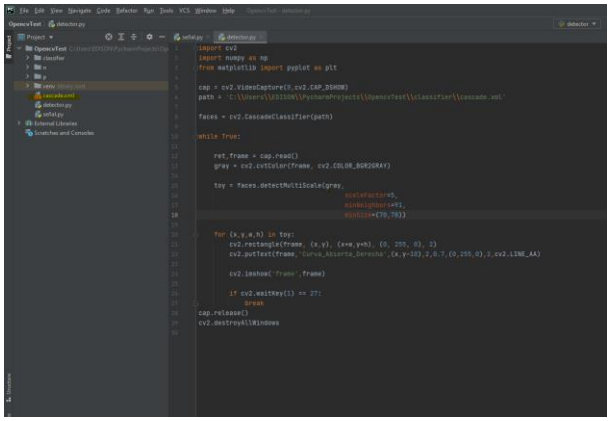


Figura 29. Código de programación para el reconocimiento de señales de tránsito preventivas

Fuente. Autor

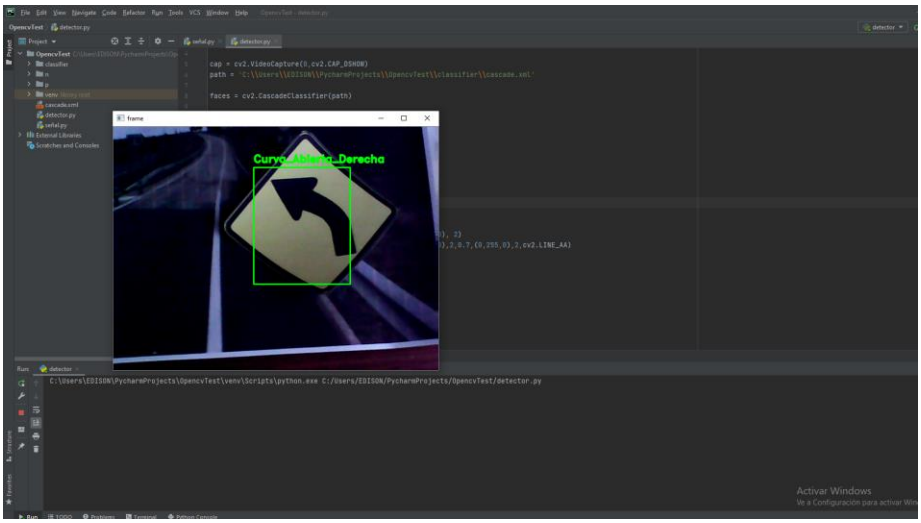


Figura 30. Resultados obtenidos

Fuente. Autor

3.7.5. Detección de líneas de carretera con OpenCV

Para detectar las líneas de carril se realizó los siguientes pasos:

1) Importación de librerías opencv, numpy, matplotlib
2) Lectura de la imagen: Lo que realizaremos es leer la imagen con la función de opencv "cv2.imread".

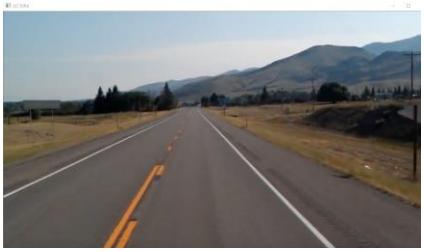


Figura 31. Lectura de la imagen a Estudiar

Fuente. Autor

3) Convertir la imagen a escala de grises por lo que se utiliza la función de opencv "cv2.cvtColor".

Figura 32. Imagen en escala de grises

Fuente. Autor

4) Filtrar el ruido: Para poder eliminar el ruido de la imagen se aplica la función de "Gaussianblur" para lo cual se debe tener en cuenta los valores del tamaño Kernel como también la desviación estándar.

Figura 33. Aplicación del filtro Blur a la imagen de escala de grises

Fuente. Autor

5) Aplicar la detención de bordes se utiliza la función "cv2.Canny" para lo cual se realiza la lectura de la imagen, transformación a escala de grises se tiene en cuenta dos valores importantes que son el umbral bajo y umbral alto el cual nos permite dibujar los bordes de la imagen en este caso nos va a dibujar las líneas de la carretera.

Tabla4. Especificación de valores Canny

Parámetros	**Valor**
Umbral bajo	**50**
Umbral alto	**150**

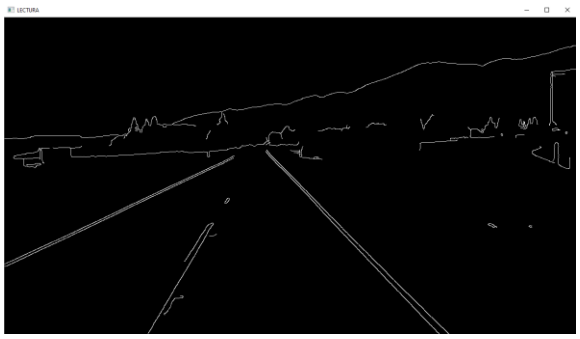


Figura 34. Detección de bordes con la función Canny

Fuente. Autor

6) Región de interés: se va a delimitar una región en (x, y) donde se dibuja un triángulo con la finalidad que dentro del triángulo solo se encuentren las líneas del carril. En la Figura 35 lo que se va a realizar es determinar los vértices del triángulo. Después de haber hallado los vértices lo que se encuentra dentro de ese triangulo es nuestra región de interés y la cual ya se detecta las líneas del carril como se puede ver en la figura 31.

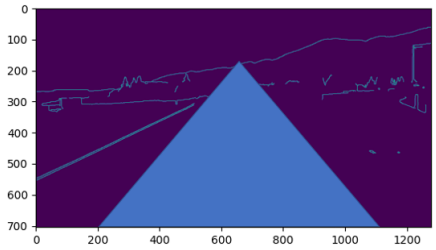


Figura 35. Región de interés

Fuente. Autor

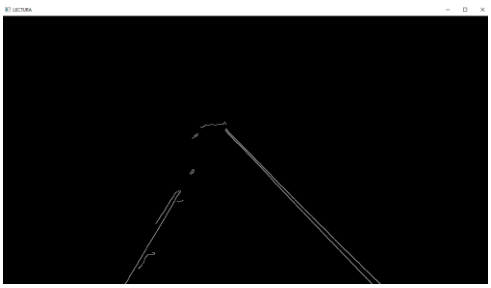


Figura 36. Detención de las líneas del carril

Fuente. Autor

7) Detección de líneas rectas para lo cual se utiliza el espacio de Hough con la función de opencv HoughlinesP

Tabla5. Parámetros para el espacio de Hough

Parámetros	**Valor**
rho(ρ)	**1**
Theta(θ)	**1° = πrad/180**
Minimun_votes_threshld	**15**
Min_line_lenght	**7**
Max_line_gap	**3**
Line_thickness	**1**

Fuente. Autor

Después de aplicar la trasformada Hough según los parámetros ya establecidos sobre la imagen se observa la detección de las líneas del carril.

Figura 37. Detección de líneas de carril

Fuente. Autor

3.7.6. Distancia de seguridad entre vehículos

Se realizó un código de programación para determinar la distancia de seguridad entre vehículos, para evitar una posible eventualidad proporcionando al conductor la capacidad de reaccionar de manera oportuna ante posibles imprevistos y así evitar accidente de tránsito.

De acuerdo al Art.175 de la ley de tránsito se determina una distancia mínima entre vehículos de 3 metros a una velocidad de 60km/h. Se realizó un dispositivo con la capacidad de emitir alertas sonoras en función a la distancia, está conformado por un sensor ultrasónico para obtener la distancia, una tarjeta Raspberry Pi 4 para procesar la información y un buzzer para emitir las alertas sonoras.

El código de programación se desarrolló en el lenguaje Python, se realizó los siguientes pasos:

- ✓ Importar las librerías GPIO y Time.
- ✓ Declarar los puertos como entrada o salida.
- ✓ Asignar las variables.
- ✓ Ejecutar el código de programación.

3.7.6.1. Diagrama de boques del funcionamiento

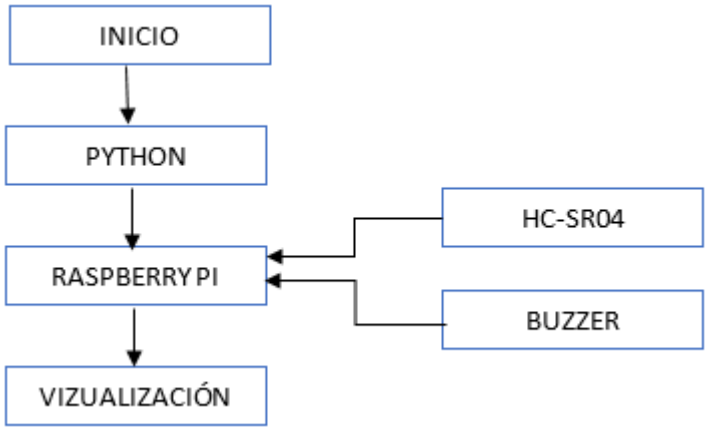


Figura 38. Diagrama de bloques de la detección de la distancia entre vehículos

Fuente. Autor

3.7.6.2. Diseño esquemático del sensor Ultrasónico

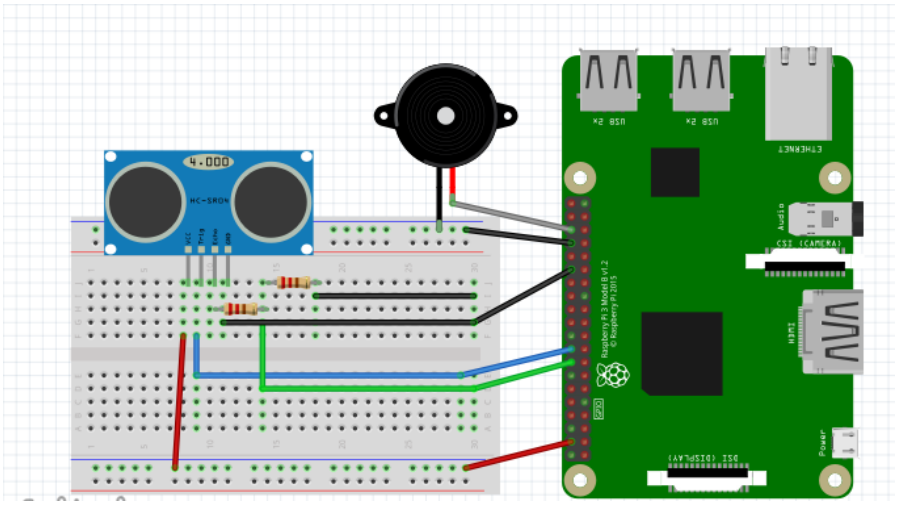
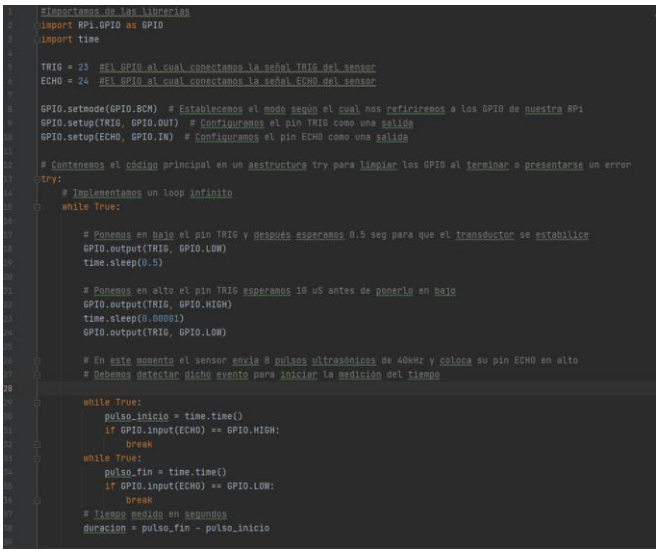


Figura 39. Conexión de Raspberry Pi y sensor ultrasónico

Fuente. Autor

Para obtener la distancia se conectó el sensor ultrasónico a la Raspberry pi en los puertos GPIO configurados como entrada. Los datos del sensor ultrasónico son procesados por código de programación realizado en el lenguaje Python y ejecutado por la tarjeta Raspberry Pi.



```
#Importamos de las librerias
import RPi.GPIO as GPIO
import time

TRIG = 23  #El GPIO al cual conectamos la señal TRIG del sensor
ECHO = 24  #El GPIO al cual conectamos la señal ECHO del sensor

GPIO.setmode(GPIO.BCM)  # Establecemos el modo según el cual nos referiremos a los GPIO de nuestra RPi
GPIO.setup(TRIG, GPIO.OUT)  # Configuramos el pin TRIG como una salida
GPIO.setup(ECHO, GPIO.IN)  # Configuramos el pin ECHO como una salida

# Contenemos el código principal en un aestructura try para limpiar los GPIO al terminar o presentarse un error
try:
    # Implementamos un loop infinito
    while True:

        # Ponemos en bajo el pin TRIG y después esperamos 0.5 seg para que el transductor se estabilice
        GPIO.output(TRIG, GPIO.LOW)
        time.sleep(0.5)

        # Ponemos en alto el pin TRIG esperamos 10 uS antes de ponerlo en bajo
        GPIO.output(TRIG, GPIO.HIGH)
        time.sleep(0.00001)
        GPIO.output(TRIG, GPIO.LOW)

        # En este momento el sensor envia 8 pulsos ultrasónicos de 40kHz y coloca su pin ECHO en alto
        # Debemos detectar dicho evento para iniciar la medición del tiempo

        while True:
            pulso_inicio = time.time()
            if GPIO.input(ECHO) == GPIO.HIGH:
                break
        while True:
            pulso_fin = time.time()
            if GPIO.input(ECHO) == GPIO.LOW:
                break
        # Tiempo medido en segundos
        duracion = pulso_fin - pulso_inicio
```

Figura 40. Código de programación para la detección de la distancia entre vehículos

Fuente. Autor

3.7.7. Detección del nivel de alcohol en un conductor

Según el código orgánico integral penal, el límite máximo establecido es de 0.3 mg/L de alcohol en la sangre (El Universo, 2020). Se desarrollo un dispositivo que permite medir el nivel de alcohol del conductor, para asegurarse que se encuentre dentro de los límites permitidos por la ley y así evitar posibles sanciones o accidentes de tránsito que se puedan ocasionar por el estado etílico del conductor.

El dispositivo está conformado por un sensor para medir el nivel de alcohol, un buzzer para emitir alertas sonoras y una tarjeta Raspberry Pi para procesar la información.

El sensor de alcohol se encuentra conectado a la raspberry Pi 4B en los puertos GPIO configurados como entrada. Los valores analógicos obtenidos del sensor no son directamente los

valores de alcohol en el aire, se debe aplicar la ecuación (2) para traducir las lecturas del sensor en miligramos por litro (mg/L).

$$alcohol = 0.4091\left(\frac{Rs}{5463}\right)^{-1.497} \quad (1)$$

Donde:

Rs es el valor de la resistencia entregada por el sensor MQ-3

- ✓ Importar las librerías GPIO y Time.
- ✓ Declarar los puertos como entrada o salida.
- ✓ Asignar las variables.
- ✓ Ejecutar el código de programación.

3.7.7.1. Diagrama de Bloques del Alcoholímetro

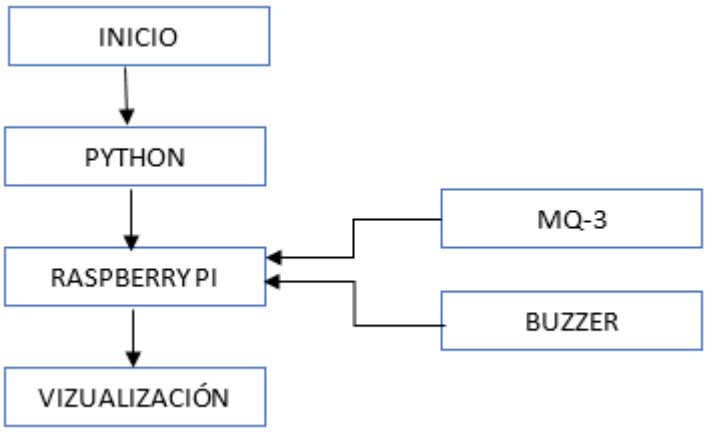


Figura 41. Diagrama de bloques para la detección del nivel de alcohol

Fuente. Autor

3.7.7.2. Diagrama esquemático del Alcoholímetro

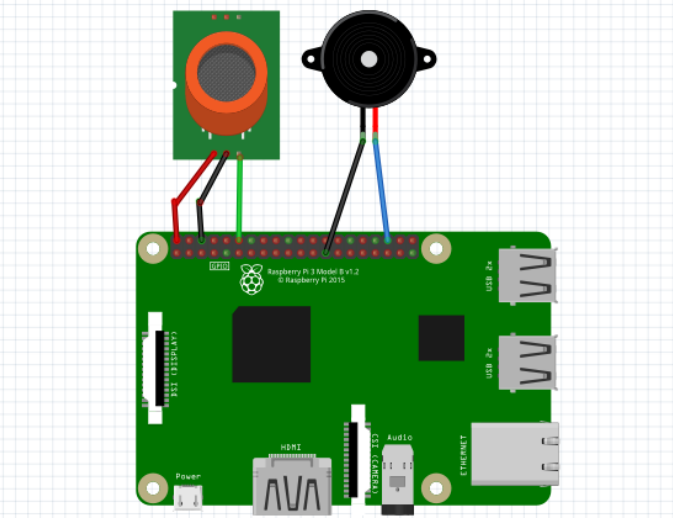


Figura 42. Conexión del sensor MQ-3 a la Raspberry Pi

Fuente. Autor

CAPÍTULO IV

4.1. Resultados y Discusión

4.1.1. Resultados experimentales

El presente capitulo se muestra las pruebas que se realiza al prototipo con la finalidad de constatar el funcionamiento de: la detención de señales de tránsito preventivas, líneas de carril, distancia entre vehículos y detención de nivel de alcohol en el conductor

4.1.1.1. Resultados de la detección de señales de tránsito preventivas

Las pruebas del dispositivo se realizaron de manera experimental, para lo cual se instaló el artefacto en el vehículo de pruebas y se realizó el recorrido de una ruta previamente establecida. Para la selección de la ruta se tomó en cuenta ciertos criterios como: cantidad de señales de tránsito preventivas, nivel de tráfico vehicular y velocidad del recorrido.

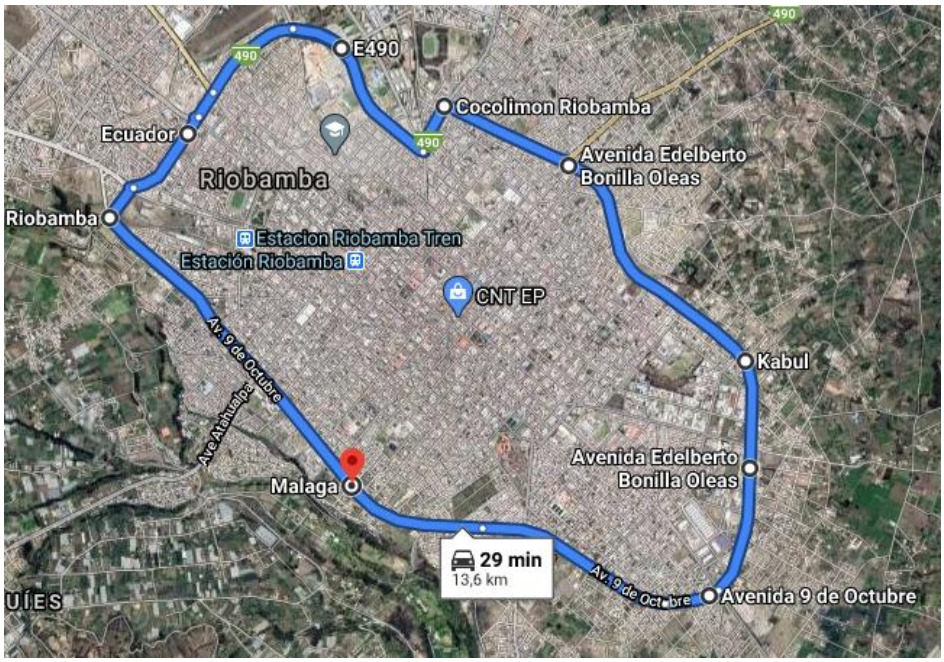


Figura 43. Recorrido realizado para la toma de Datos

Fuente. Autor

El vehículo de pruebas realizó el recorrido a una velocidad de 50Km/h, se demoró en completar el recorrido una hora, donde se detectó 126 señales de tránsito preventivas como: Aproximación de semáforos, Curva abierta izquierda, Curva abierta derecha, Reductor de velocidad, Peatones en la vía, Animales en vía, Cruze peatonal con prioridad, Aproximación a redondel, empalme lateral en curva derecha, Cruze de vías, Niños y curva cerrada ala derecha.

Estas pruebas se realizaron en dos escenarios, donde se tomó en cuenta las horas del día debido a su iluminación natural. Una correcta iluminación permite la visualización de las señales de mejor manera, caso contrario se dificulta su reconocimiento.

Escenario 1.- En este escenario la toma de datos se realizó en el horario de 12pm a 2pm

Figura 44. Escenario 1 de 12pm – 2pm

Fuente. Autor

Una vez realizado el reconocimiento de las señales de tránsito preventivas obtenidas en el escenario 1 mediante el sistema desarrollado en este proyecto, se obtuvieron los resultados mostrados en la tabla 6.

Tabla6. Porcentaje de aciertos de reconocimiento de las señales de tránsito preventivas en el escenario 1

Tipo de señal	**# de Señales**	**# de Señales detectadas**	**Porcentaje**
Aproximación de semáforo	22	18	81.8%
Curva abierta izquierda	12	10	83.3%
Reductor de velocidad	12	11	91.6%
Peatones en la vía	34	28	82.3%
Animales en la vía	2	2	100%
Cruze peatonal con prioridad	12	7	58.3%
Curva abierta a la derecha	16	16	100%
Aproximación a redondel	4	3	75%
Empalme lateral en curva derecha	4	2	50%
Cruze de vías	4	4	100%
Niños	2	2	100%
Curva cerrada derecha	2	2	100%
TOTAL	126	105	83.3%

Fuente. Autor

Escenario 2.- En este escenario la toma de datos se realizó en el horario de 4am a 6am

Figura 45. Escenario 2 de 4am – 6am

Fuente. Autor

Una vez realizado el reconocimiento de las señales de tránsito preventivas obtenidas en el escenario 2 mediante el sistema desarrollado en este proyecto, se obtuvieron los resultados mostrados en la tabla 7.

Tabla7. Porcentaje de aciertos de reconocimiento de las señales de tránsito preventivas en el escenario 2

Tipo de señal	**# de Señales**	**# de Señales detectadas**	**Porcentaje**
Aproximación de semáforo	22	13	59%
Curva abierta izquierda	12	5	41.66%
Reductor de velocidad	12	7	58.3%
Peatones en la vía	34	20	58.8%
Animales en la vía	2	0	0%
Cruze peatonal con prioridad	12	5	41.6%
Curva abierta a la derecha	16	10	62.5%
Aproximación a redondel	4	2	50%
Empalme lateral en curva derecha	4	1	25%
Cruze de vías	4	2	50%
Niños	2	0	0%
Curva cerrada derecha	2	1	50%
TOTAL	126	66	52.38%

Fuente. Autor

Tabla8. Pruebas de muestras emparejadas

Estadísticas de grupo					
	Horario	N	Media	Desv. Desviación	Desv. Error promedio
Señales Reconocidas	Madrugada	12	5,50	6,142	1,773
	Tarde	12	8,75	8,292	2,394

Prueba de muestras independientes										
		Prueba de Levene de igualdad de varianzas		prueba t para la igualdad de medias						
									95% de intervalo de confianza de la diferencia	
		F	Sig.	t	gl	Sig. (bilateral)	Diferencia de medias	Diferencia de error estándar	Inferior	Superior
Señales Reconocidas	Se asumen varianzas iguales	1,171	,291	-1,091	22	,287	-3,250	2,979	-9,428	2,928
	No se asumen varianzas iguales			-1,091	20,279	,288	-3,250	2,979	-9,458	2,958

Como p = 0.291 significa que las medias entre las señales detectadas de 12 a 14 pm y las señales detectadas de 4 a 6 am son significativamente diferentes, se concluye que, una correcta iluminación permite la visualización de las señales de mejor manera, caso contrario se dificulta su reconocimiento.

4.1.1.2. Resultados de la detección de líneas de carril

Para la realización de las líneas de carril se tomó en cuenta el área donde se va a realizar la detención de líneas de carril para lo cual se realizó un algoritmo donde ya se detalla paso a paso lo que se realizó para obtener los resultados que se muestran a continuación donde se visualiza el área de trabajo el radio de curvatura también si el conductor va en las siguientes direcciones ya que son izquierda, derecha, recto.

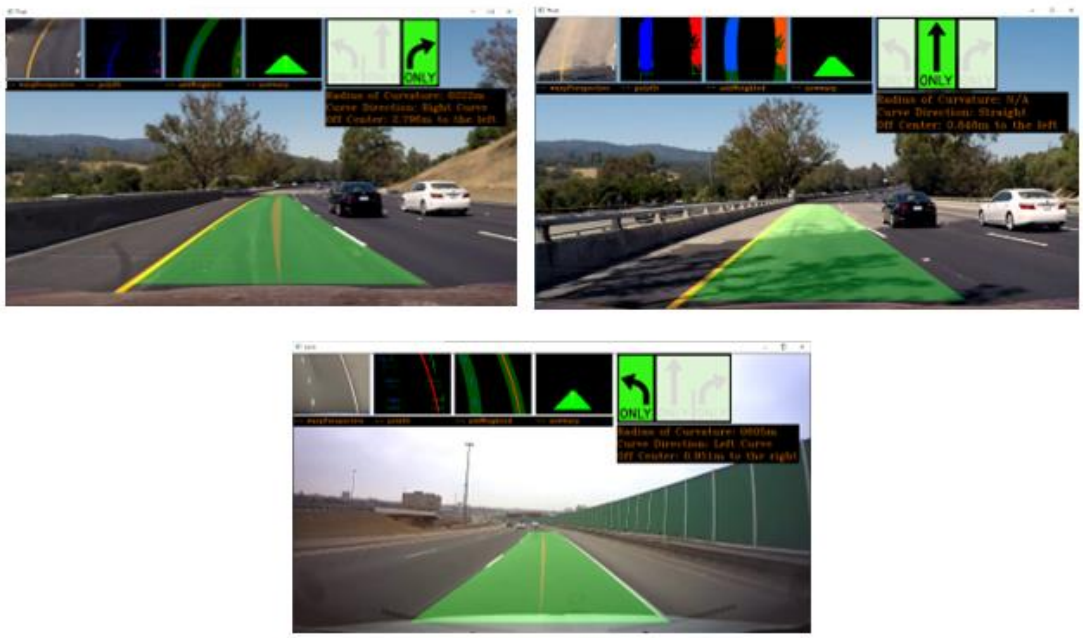


Figura 46. Resultados de la detención de carril

Fuente. Autor

4.1.1.3. Resultados de la distancia de seguridad con otro vehículo

Muestra de resultados entre valor medido del sensor y valor real.

Tabla9. Valores reales con respecto a valores obtenidos del sensor ultrasónico

Valor Real(cm)	Valor Medido(cm)	Error(cm)
50	49	1
100	96	4
150	144	6
200	191	9
250	243	7
300	290	10
350	339	11
400	388	12
450	427	23
500	470	30

Fuente. Autor

Tabla10. Estadísticas de grupo

Estadísticas de grupo					
	grupo	N	Media	Desv. Desviación	Desv. Error promedio
valor	REAL	10	275,00	151,383	47,871
	MEDIDO	10	263,70	143,488	45,375

Fuente. Autor

Tabla11. Prueba T para muestras independientes

Prueba de muestras independientes										
		Prueba de Levene de igualdad de varianzas		prueba t para la igualdad de medias						
		F	Sig.	t	gl	Sig. (bilateral)	Diferencia de medias	Diferencia de error estándar	95% de intervalo de confianza de la diferencia	
									Inferior	Superior
valor	Se asumen varianzas iguales	,034	,857	,171	18	,866	11,300	65,959	-127,274	149,874
	No se asumen varianzas iguales			,171	17,949	,866	11,300	65,959	-127,303	149,903

Fuente. Autor

Se observa que entre el valor de las medias del valor real y el valor medido son significativamente diferentes, por tanto, el sensor es válido para este sistema.

Para realizar las pruebas de funcionamiento del sistema de seguridad entre vehículos, se considera tres distancias como son: 400cm, 200cm, 100cm. Unos leds indicarán a que distancia se encuentra el vehículo.

Figura 47. Distancia sin presencia de un vehículo

Fuente. Autor

Como se puede visualizar en la figura 47, se aprecia que no se encuentra ningún vehículo por lo tanto no se activa la alerta sonora en estas circunstancias el conductor se encuentra seguro.

Figura 48. Distancia con la presencia de un vehículo

Fuente. Autor

Se visualiza en la figura 48 que el conductor se encuentra en una distancia no permitida, se activa la alerta sonora para dar a conocer al conductor sobre el rebasamiento de dicha distancia.

4.1.1.4. Resultados del nivel de alcohol en el conductor

Antes de poder conectar la raspberry Pi con el sensor MQ-3 se diseñó un divisor de voltaje ya que las señales de voltaje que nos envía el sensor son 0 – 5V y los puertos GPIO manejan un voltaje de 0 – 3.3V. Se procede hacer los cálculos con la siguiente ecuación (2)

$$alcohol = Vin * \frac{R2}{(R1 + R2)} \quad (2)$$

Vout = Voltaje de salida

Vin = Voltaje de entrada

R1 y R2 = Valores de las resistencias

Dado que conocemos los valores de los Voltajes de entrada y salida y dando un valor comercial a R1 despejamos R2 de la formula ya existente.

$$R2 = \frac{Vout * R1}{Vin - Vout}$$

$$R2 = \frac{3.3 * 10K}{5 - 3.3}$$

$$R2 = 19.41 \approx 20K$$

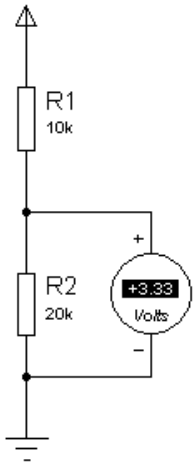


Figura 49. Diseño del divisor de voltaje

Fuente. Autor

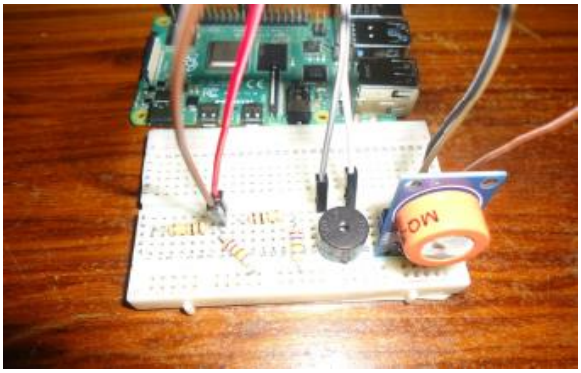


Figura 50. Montaje del circuito de alcohol

Fuente. Autor

Para comprobar el funcionamiento del dispositivo se procedió a realizar pruebas en 2 partes: con presencia de alcohol y sin presencia de alcohol

Tabla12. Pruebas de nivel de alcohol a sujetos de prueba sin presencia de alcohol.

	Prueba 1	Prueba 2	Prueba 3	Prueba 4	Prueba 5	Promedio
Sujeto 1	0.01 mg/L	0.07 mg/L	0.06 mg/L	0.06 mg/L	0.06 mg/L	0.06 mg/L
Sujeto 2	0.05 mg/L	0.05 mg/L	0.05 mg/L	0.05 mg/L	0.05 mg/L	0.05 mg/L
Sujeto 3	0.05 mg/L	0.05 mg/L	0.05 mg/L	0.04 mg/L	0.05 mg/L	0.05 mg/L
Sujeto 4	0.04 mg/L	0.04 mg/L	0.04 mg/L	0.04 mg/L	0.04 mg/L	0.04 mg/L
Sujeto 5	0.03 mg/L	0.03 mg/L	0.03 mg/L	0.03 mg/L	0.03 mg/L	0.03 mg/L

Fuente. Autor

En la tabla 12 se realiza pruebas de nivel de alcohol a cinco sujetos, los cuales se encuentran en condiciones óptimas para poder conducir.

Tabla13. Pruebas de nivel de alcohol a sujetos de prueba con presencia de alcohol.

	Prueba 1	Prueba 2	Prueba 3	Prueba 4	Prueba 5	Promedio
Sujeto 1	4.20 mg/L	4.21 mg/L	4.27 mg/L	5.10 mg/L	5.18 mg/L	4.59 mg/L
Sujeto 2	4.27 mg/L	4.30 mg/L	5.20 mg/L	5.23 mg/L	5.37 mg/L	4.87 mg/L
Sujeto 3	7.92 mg/L	7.85 mg/L	7.60 mg/L	7.42 mg/L	7.37 mg/L	7.63 mg/L
Sujeto 4	6.75 mg/L	6.71 mg/L	6.48 mg/L	6.20 mg/L	6.03 mg/L	6.43 mg/L
Sujeto 5	3.95 mg/L	3.90 mg/L	3.86 mg/L	3.80 mg/L	3.06 mg/L	3.11 mg/L

Fuente. Autor

En la tabla 13 se demuestra los datos tomados a cinco sujetos, que presentan un nivel de alcohol excesivo para lo cual no estarían aptos para conducir un vehículo.

4.2. Discusión

Los resultados encontrados en este estudio en cuanto al reconocimiento de señales de tránsito preventivas, se obtuvo un porcentaje de aciertos en el día de un 83.3% y en la noche de un 52.3% esto significa que el dispositivo, al tener la presencia de luz natural responde con un porcentaje alto al reconocimiento de señales y al encontrarse con luz artificial se obtiene un porcentaje medio. Con respecto a los resultados de la detección de líneas de carril se aplica la trasformada Hough debido a su alta eficiencia y por la facilidad de representar las líneas que se encuentren, el funcionamiento del dispositivo se encuentra en un nivel aceptable, pero siempre y cuando las líneas de carril estén claramente pintadas sobre la vía. El análisis de la distancia entre vehículos se realiza entre dos valores (real, medio), ya que se observa un valor de significancia de 0.857 esto significa que el sensor ultrasónico aplica para el sistema de alerta. Los resultados del nivel de alcohol en el conductor se lo realizan a un grupo de sujetos obteniendo un resultado promedio de 0.05 sin la presencia de alcohol, un valor promedio de 5.54 con presencia de alcohol en estas condiciones el conductor no puede manejar.

CAPÍTULO V

5.1. Conclusiones

- En este trabajo se implementó un prototipo electrónico de alerta automática para la prevención de accidentes mediante el reconocimiento de señales con visión artificial, el sistema es capaz de alertar al conductor sobre un posible suceso en la vía.
- En esta tesis se desarrolló un código de programación para el reconocimiento de señales de tránsito preventivas y horizontales, el procesamiento de imágenes está limitado debido a que mientras disminuye la luz natural será más complicado el reconocimiento de las señales.
- Se implementó un sistema de seguridad de alcohol y distancia, mediante señales sonoras alertar al conductor frente a un posible accidente tránsito.
- El aporte importante que se le ha dado a este proyecto de titulación es la utilización de visión artificial ya que nos ha facilitado el trabajo de este proyecto con sus librerías y tratamiento de imágenes y mejoras en trabajos futuros (reconocimiento de ojos, reconocimiento de tapabocas).

5.2. Recomendaciones

- La calibración de la cámara es un factor importante, para el correcto funcionamiento del sistema de alerta automática.
- Se recomienda la utilización de buena cámara (visión infrarroja, foco ajustable), para lo cual se necesita imágenes o fotogramas en buena definición, para el procesamiento de imágenes.

- Se recomienda la utilización de la Raspberry pi 4B de 8GB, debido al sistema operativo que es de 64 bits.

BIBLIOGRAFÍA

agermoso. (20 de Julio de 2015). Obtenido de http://licenciadeconducirrd.blogspot.com/2015/07/senales-de-transito.html

amazon. (s.f.). Obtenido de https://www.amazon.es/LABISTS-C%C3%A1mara-Oficial-Raspberry-Compatible/dp/B07VTMNS2B

ANT Chimborazo. (14 de Junio de 2015). Obtenido de Agencia Nacional de Tránsito: https://www.ant.gob.ec/index.php/noticias/boletines-provinciales/63-noticias-2/boletines-chimborazo/1169-ant-chimborazo-registra-una-reduccion-de-siniestros-en-un-15-en-junio-de-2015#.XeZlxb5MSUk

Arce Millan, K., & Vasco Alzate, I. (2018). *Reconocimiento de patrones en imágenes de video para el monitoreo de eventos de tráfico Vehicular.* Santiago de Cali.

Barba Guamán, L. (2015). *Utilización de métodos de visión artificial para PC como apoyo en la automoción.* Madrid.

Casanova Vásquez, M. P. (2014). *Diseño, Construccion e instalación de un alcoholímetro electrónico con dispositivo de bloqueo de un vehículo.* Riobamba.

Cazau, P. (2006). *Introducción a la investigación en Ciencias Sociales.* Buenos Aires.

Chicaiza, F. E. (2017). *Sistema Electrónico de alerta automática para el reconocimiento de señales de tránsito reglamentarias, preventivas e informativas en la ciudad de Ambato.* Ambato.

Cristina1128. (28 de Mayo de 2015). *slideshare.* Obtenido de https://www.slideshare.net/Cristina1128/ecuador-connuevatipologadeaccidentesdetrnsito

Cruzado Hernando, D. (2015). *Detección y reconocimiento de señales de tráfico.* Madrid.

EcuRed. (s.f.). Obtenido de https://www.ecured.cu/Modelo_HSV

Educación Vial. (12 de Noviembre de 2018). Obtenido de http://educacionvial201813.blogspot.com/2018/11/tipos-de-senales-de-transito-senales-de.html

educavial. (18 de Junio de 2015). *slideshare*. Obtenido de https://es.slideshare.net/educavial/seales-horizontales-49533482

El Universo. (14 de Septiembre de 2020). Obtenido de https://www.eluniverso.com/noticias/2020/09/14/nota/7977406/alguien-puede-negarse-hacerse-prueba-alcoholemia-que-dice-ley/

Electronilab. (s.f.). Obtenido de https://electronilab.co/tienda/sensor-de-distancia-de-ultrasonido-hc-sr04/

Estarita, J., Jiménez, A., Brochero, J., Escobar, H., & Moreno, S. (2018). *Sistema de Reconocimiento de objetos en tiempo real.* Colombia.

Flores Calero, M., Conlago, C., Yunda , j., & Aldás, M. (2018). *Implementación de un algoritmo para la detencción de señales de tránsito del Ecuador: Pare, Ceda el paso y Velocidad.* Ecuador.

Gamán, l. R. (2015). *Utilización de métodos de visión artificial para PC como apoyo en la automoción.* Madrid.

Historia de la Informática. (18 de Diciembre de 2013). Obtenido de https://histinf.blogs.upv.es/2013/12/18/raspberry-pi/

INEN. (2011). *Señalización Vial. Parte 1. Señalización Vertical.* Quito. Obtenido de https://www.obraspublicas.gob.ec/wp-content/uploads/downloads/2015/04/LOTAIP2015_reglamento-tecnico-ecuatoriano-rte-inen-004-1-2011.pdf

Lara Nuñez, R., & Mares Ruiz, M. (2016). *Reconocimiento de Patrones en Imágenes con un Sistema Embebido.* Jalisco.

Marcano, M. (24 de Enero de 2018). *issuu*. Obtenido de https://issuu.com/mariamarcan1996/docs/la_investigacion_experimental_pdf

master autoescuela. (s.f.). Obtenido de https://www.autoescuelamasterbolivia.com/blog/senales-verticales/

Mateo, J. E. (2017). *Implementacion de un software para la detección y reconocimiento de señales de tráfico en tiempo real a partir de un video capturado en un vehiculo en circulación.* Gandia.

Mendieta, V. A. (2013). *"Detección y Reconocimiento de Semáforos por Visión Artificial".* Madrid.

Mendieta, V. A. (2013). *Detección y Reconocimiento de Semáforos por Visión Artificial.* Madrid.

Mujtaba, H. (2 de Septiembre de 2020). *Great Learning.* Obtenido de https://www.mygreatlearning.com/blog/viola-jones-algorithm/

Naylamp Mechatronics. (s.f.). Obtenido de https://naylampmechatronics.com/blog/42_Tutorial-sensores-de-gas-MQ2-MQ3-MQ7-y-MQ13.html

Organización Mundial de la Salud. (7 de Diciembre de 2018). Obtenido de https://www.who.int/es/news-room/fact-sheets/detail/road-traffic-injuries

Peden, M. (2009). *Informe Sobre la Situación Mundial de la Seguridad Vial.* Suiza.

Pico Aponte, G. M. (2019). *Sistema Avanzado de Asistencia Al Conductor Empleando Visión Artificial en vehículos de Transporte Público.* Ambato-Ecuador.

rambal. (s.f.). Obtenido de https://rambal.com/raspberry/736-pantalla-raspberry-pi-5-in.html

Rueda Valencia, B. P. (2017). *Desarrollo de un prototipo portatil para el reconocimiento de señales dactilogicas mediante visión artificial.* Quito.

Sailema Chicaiza, F. E. (2017). *Sistema Electrónico de alerta automática para el reconocimiento de señales de tránsito reglamentarias, preventivas e informaticas en la ciudad de Ambato.* Ambato.

stackoverflow. (2020). Obtenido de https://stackoverflow.com/questions/10948589/choosing-the-correct-upper-and-lower-hsv-boundaries-for-color-detection-withcv?lq=1

Tutor de Programación. (2017). Obtenido de https://acodigo.blogspot.com/p/tutorial-opencv.html

Villalón, G., Torres, M., & Flores, M. (2017). Sistema de detencción de señales de tráfico para la localización de intersecciones viales y frenado anticipado. *Revista Iberoamericana de Automática e Informática Industrial*, 1-11.

Yanque, J. (22 de Agosto de 2016). *SCRIBD*. Obtenido de https://es.scribd.com/document/321852334/Metodologia-de-La-Investigacion-metodos-Consulta

ANEXOS

Anexo 1. Presupuesto

PRESUPUESTO DEL PROTOTIPO			
CANT	ELEMENTOS	costo unitario	PRECIOS
1	Raspberry Pi4 4GB	95	95
1	Cámara Raspberry Pi	52	52
1	Sensor MQ-3	5	5
1	Sensor ultrasónico HC-SR04	2,5	2,5
1	Pantalla Raspberry pi 5"	70	70
1	Estructura del prototipo	30	30
1	Inversor de Corriente Adaptador 12V-120V	20	20
1	Cargador Rasberry Pi 5V, 3A	10	10
30	Cables de Arduino	0,1	3
		TOTAL	287,5

Figura 51. Presupuesto del prototipo a realizar

Fuente. Autor

Anexo 2. Implementación

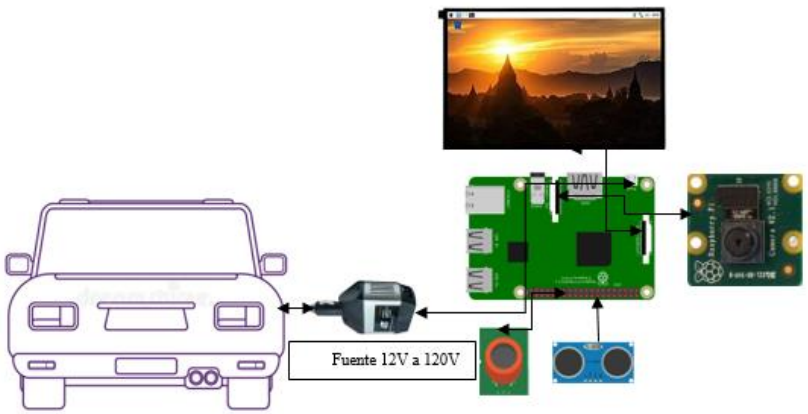


Figura 52. Implementación del prototipo en el vehículo

Fuente. Autor

Anexo 3. Adaptador inversor de 12 a 120 V

Figura 53. Inversor de energía de 12V a 120V para Autos

Fuente. Autor

CARACTERISTICAS	**VALOR**
Voltaje de entrada	12V
Voltaje de salida	120V
Voltaje de USB	5V a 0.5A
Potencia Maxima	150 W

Figura 54. Características del inversor de energía

Fuente. Autor

Anexo 4. El prototipo es de fácil utilización y montaje, tiene unas dimensiones de 14.5cm largo y de ancho 4cm

Figura 55. Implementación del prototipo en el vehículo

Fuente. Autor

Anexo 5. Vehículo con el cual se realizó las pruebas

Figura 56. Vehículo de pruebas

Fuente. Autor

Anexo 6. **Código para la toma de imágenes positivas y negativas**

```
import cv2
import numpy as np
import imutils
```

```
import os
Datos = 'n'
if not os.path.exists(Datos):
   print('Carpeta creada:', Datos)
   os.makedirs(Datos)
cap = cv2.VideoCapture(0, cv2.CAP_DSHOW)
x1, y1 = 190, 80
x2, y2 = 450, 398
count = 0
while True:
   ret, frame = cap.read()
   if ret == False: break
   imAux = frame.copy()
   cv2.rectangle(frame,(x1,y1),(x2,y2),(255,0,0),2)
   objeto = imAux[y1:y2,x1:x2]
   objeto = imutils.resize(objeto, width=38)
   k = cv2.waitKey(1)
   if k == ord('s'):
      cv2.imwrite(Datos+'/objeto_{}.jpg'.format(count),objeto)
      print('Imagen almacenada:','objeto_{}.jpg'.format(count))
      count = count + 1
   if k == 27:
      break
   cv2.imshow('frame', frame)
   cv2.imshow('objeto', objeto)
cap.release()
cv2.destroyAllWindows()
```

Anexo 7. **Código para el reconocimiento de señales preventivas**

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
cap = cv2.VideoCapture(0,cv2.CAP_DSHOW)
path = 'C:\\Users\\EDISON\\PycharmProjects\\OpencvTest\\classifier\\cascade.xml'

faces = cv2.CascadeClassifier(path)
while True:
   ret,frame = cap.read()
   gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
   toy = faces.detectMultiScale(gray,
                                scaleFactor=5,
                                minNeighbors=91,
                                minSize=(70,78))
   for (x,y,w,h) in toy:
```

```
        cv2.rectangle(frame, (x,y), (x+w,y+h), (0, 255, 0), 2)
        cv2.putText(frame,'Curva_Abierta_Derecha',(x,y-10),2,0.7,(0,255,0),2,cv2.LINE_AA)
        cv2.imshow('frame',frame)
        if cv2.waitKey(1) == 27:
            break
cap.release()
cv2.destroyAllWindows
```

Anexo 8. **Código de detención de líneas de carril**

```
import cv2
import numpy as np
def average_slope_intercept(image, lines):
    left_fit =[]
    right_fit = []
    for line in lines:
        x1, y1, x2, y2 = line.reshape(4)
        parameters = np.polyfit((x1, y1), (x2, y2), 1)
        print(parameters)
def canny(image):
    gray = cv2.cvtColor(lane_image, cv2.COLOR_RGB2GRAY)
    blur = cv2.GaussianBlur(gray, (5, 5), 0)
    canny = cv2.Canny(blur, 50, 150)
    return canny
def display_lines(image, lines):
    line_image = np.zeros_like(image)
    if lines is not None:
        x1, y1, x2, y2 = line.reshape(4)
        cv2.line(line_image, (x1, y1), (x2, y2), (255, 0, 0), 10)
    return line_image
def region_of_interest(image):
    height = image.shape[0]
    polygons = np.array([[(200, height), (1100, height), (550, 250)]])
    mask = np.zeros_like(image)
    cv2.fillPoly(mask, polygons, 255)
    masked_image = cv2.bitwise_and(image, mask)
    return masked_image
image = cv2.imread('test_image.jpg')
lane_image = np.copy(image)
canny_image = canny(lane_image)
cropped_image = region_of_interest(canny_image)
lines = cv2.HoughLinesP(cropped_image, 1, np.pi/180, 100, np.array([]), minLineLength=100,
maxLineGap=10)
averaged_lines = average_slope_intercept(lane_image, lines)
line_image = display_lines(lane_image, lines)
```

```
combo_image = cv2.addWeighted(lane_image, 0.8, line_image, 1, 1)
cv2.imshow('LECTURA', combo_image)
cv2.waitKey(0)
```

Anexo 9. **Código de distancia entre vehículos**

```
#Importamos de las librerias
import RPi.GPIO as GPIO
import time

TRIG = 23  #El GPIO al cual conectamos la señal TRIG del sensor
ECHO = 24  #El GPIO al cual conectamos la señal ECHO del sensor

GPIO.setmode(GPIO.BCM)  # Establecemos el modo según el cual nos referiremos a los GPIO de nuestra RPi
GPIO.setup(TRIG, GPIO.OUT)  # Configuramos el pin TRIG como una salida
GPIO.setup(ECHO, GPIO.IN)  # Configuramos el pin ECHO como una salida

try:
    # Implementamos un loop infinito
    while True:
        # Ponemos en bajo el pin TRIG y después esperamos 0.5 seg para que el transductor se estabilice
        GPIO.output(TRIG, GPIO.LOW)
        time.sleep(0.5)
        # Ponemos en alto el pin TRIG esperamos 10 uS antes de ponerlo en bajo
        GPIO.output(TRIG, GPIO.HIGH)
        time.sleep(0.00001)
        GPIO.output(TRIG, GPIO.LOW)
        # En este momento el sensor envía 8 pulsos ultrasónicos de 40kHz y coloca su pin ECHO en alto
        # Debemos detectar dicho evento para iniciar la medición del tiempo
        while True:
            pulso_inicio = time.time()
            if GPIO.input(ECHO) == GPIO.HIGH:
                break
        while True:
            pulso_fin = time.time()
            if GPIO.input(ECHO) == GPIO.LOW:
                break
        # Tiempo medido en segundos
        duracion = pulso_fin - pulso_inicio
        distancia = (34300 * duracion) / 2
        # Imprimimos resultado
```

```
        print("Distancia: %.2f cm" % distancia)
finally:
GPIO.cleanup()
```