ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA ELECTRÓNICA Y AUTOMATIZACIÓN

"DESARROLLO DE UN SISTEMA DE VISIÓN ARTIFICIAL BASADO EN REDES CONVOLUCIONALES PARA LA DETECCIÓN DE LAS SEÑALES DE TRÁNSITO IMPLEMENTADO SOBRE UN VEHÍCULO AUTÓNOMO"

Trabajo de titulación

Tipo: Proyecto técnico

Presentado para optar al grado académico de:

INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN

AUTOR:

ANDRÉS ALEXANDER SANANGO TUFIÑO

Riobamba – Ecuador

2022

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA ELECTRÓNICA Y AUTOMATIZACIÓN

"DESARROLLO DE UN SISTEMA DE VISIÓN ARTIFICIAL BASADO EN REDES CONVOLUCIONALES PARA LA DETECCIÓN DE LAS SEÑALES DE TRÁNSITO IMPLEMENTADO SOBRE UN VEHÍCULO AUTÓNOMO"

Trabajo de titulación

Tipo: Proyecto técnico

Presentado para optar al grado académico de:

INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN

AUTOR: ANDRÉS ALEXANDER SANANGO TUFIÑO

DIRECTOR: ING. JORGE LUIS PAUCAR SAMANIEGO

Riobamba – Ecuador

2022

Yo, Andrés Alexander Sanango Tufiño, declaro que el presente trabajo de titulación es de mi autoría y que los resultados de este son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor asumo la responsabilidad legal y académica de los contenidos de este trabajo de titulación; el patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 1 de abril de 2022

Andrés Alexander Sanango Tufiño

080348563-0

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA ELECTRÓNICA Y AUTOMATIZACIÓN

El Tribunal del Trabajo de Titulación certifica que: El trabajo de titulación; tipo: Proyecto técnico, **"DESARROLLO DE UN SISTEMA DE VISIÓN ARTIFICIAL BASADO EN REDES CONVOLUCIONALES PARA LA DETECCIÓN DE LAS SEÑALES DE TRÁNSITO IMPLEMENTADO SOBRE UN VEHÍCULO AUTÓNOMO"**, realizado por el señor **ANDRÉS ALEXANDER SANANGO TUFIÑO,** ha sido minuciosamente revisado por los Miembros del Trabajo de Titulación, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal Autoriza su presentación.

	FIRMA	**FECHA**
Ing. Edwin Vinicio Altamirano Santillán **PRESIDENTE DEL TRIBUNAL**	Firmado electrónicamente por: EDWIN VINICIO ALTAMIRANO SANTILLAN	2022-04-01
Ing. Jorge Luis Paucar Samaniego **DIRECTOR DE TRABAJO DE TITULACIÓN**	Firmado electrónicamente por: JORGE LUIS PAUCAR SAMANIEGO	2022-04-01
Ing. José Luis Morales Gordón **MIEMBRO DEL TRIBUNAL**	Jose Morales Firmado digitalmente por Jose Morales Fecha: 2022.04.06 20:52:41 -05'00'	2022-04-01

DEDICATORIA

El presente trabajo investigativo lo dedico principalmente a Dios, por ser el inspirador y darme la fuerza para continuar en este proceso de obtener uno de los anhelos más deseados.

A mis padres Luis Sanango y María Tufiño quienes con su amor, paciencia y esfuerzo me han permitido llegar a cumplir hoy un sueño más, gracias por infundir en mí el ejemplo de esfuerzo, perseverancia y valentía, de no temer las adversidades porque Dios está conmigo siempre.

A mis hermanos Jorge, Ligia y Christian por su amor y apoyo incondicional, durante todo este proceso, por estar conmigo en todo momento gracias. A toda mi familia porque con sus oraciones, lecciones y palabras de aliento hicieron de mí una excelente persona y de una u otra forma me acompañan en todos mis sueños y metas.

Finalmente quiero dedicar esta tesis a la corporación orquesta, por apoyarme cuando más las necesito, por extender su mano en momentos difíciles, por darme la oportunidad de cumplir uno de mis sueños, permitiéndome ganarme la vida haciendo lo que más amo y por el amor brindado cada día, de verdad mil gracias a todos, siempre los llevo en mi corazón.

Andrés

AGRADECIMIENTO

Quiero expresar mi gratitud a Dios, quien con su bendición y cuidado llena siempre mi vida, también a toda mi familia y amigos por estar siempre presentes.

Mi más profundo agradecimiento a todas las autoridades y personal que conforman la Escuela Superior Politécnica de Chimborazo, por confiar en mí, brindarme sus conocimientos y hacer de la experiencia educativa mi preparación para el mundo profesional.

A mis profesores en especial a los ingenieros Jorge Luis Paucar, José Luis Morales Gordón quienes con la enseñanza de sus valiosas sapiencias hicieron que pueda crecer día a día como profesional, gracias a cada uno de ustedes por su paciencia, dedicación, apoyo incondicional y amistad.

Andrés

TABLA DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE GRÁFICOS

ÍNDICE DE ANEXOS

ÍNDICE DE ABREVIATURAS

API:	Interfaz de programación de aplicaciones
DP:	Deep Learning
EM:	Ecografía Mamaria
IA:	Inteligencia Artificial
IC:	Inteligencia Computacional
EDA:	Automatización de diseño electrónico
EMI:	Interferencia electromagnética
IEEE:	Instituto de Ingenieros Eléctricos y Electrónicos
ISM:	Industrial, científico y médico
OMS:	Organización Mundial de la Salud
RN:	Red Neuronal
RNC:	Red Neuronal Convolucional
SAE:	Sociedad de Ingenieros Automotrices

RESUMEN

En el presente trabajo se describe el análisis, desarrollo y creación de un sistema de visión artificial basado en redes convolucionales para la detección de señales de tránsito, control de velocidad y detención, implementado sobre un vehículo autónomo, para lo cual se analizaron los procesos que se llevan a cabo en sistemas avanzados de asistencia al conductor, también se estudiaron los algoritmos de visión artificial. El prototipo basa su funcionamiento en el uso de software libre mediante la programación en Python con sus librerías de OpenCV y el entrenamiento de la red convolucional para el reconocimiento de patrones empleando técnicas de visión artificial como los algoritmos Haar Cascade, con el uso de estos se realiza el procesamiento necesario para la detección de las señales de tránsito y el control de la velocidad del vehículo. En el reconocimiento de estas señales se utiliza una cámara OV5647 propia de Raspberry la cual se ubica en la parte superior del vehículo, así el módulo recibe la información y la procesa mediante el algoritmo utilizando los entrenamientos descritos dando la orden de control sobre el driver L298N, mismo que controla la velocidad de desplazamiento del vehículo. En base a las pruebas realizadas se llegó a determinar que la detección de señales tiene un 99.16% de precisión, por lo tanto, se concluye que el prototipo implementado presenta un alto grado de funcionalidad a la hora de cumplir su objetivo pudiendo ser mejorado y aplicado a vehículos reales, constituyéndolo, así como un dispositivo capaz de aportar a la disminución de accidentes de tránsito. Se recomienda utilizar Raspberry pi 4B de 8GB, debido su eficiencia en procesamiento, en el caso de necesitar más recursos se puede utilizar aceleradores diseñados para aplicaciones de visión artificial con la finalidad de obtener mayor precisión y menos errores en el reconocimiento.

Palabras clave: <VISION ARTIFICIAL>, <DETECCIÓN>, <SEÑAL DE TRÁNSITO>, <CASCADE TRAINGER GUI>, <VEHÍCULO AUTÓNOMO>, <ALGORITMO>.

Firmado electrónicamente por:
HOLGER GERMAN RAMOS UVIDIA

0597-DBRA-UPT-2022

2022-04-05

SUMMARY

This paper describes the analysis, development and creation of an artificial vision system based on convolutional networks for the detection of traffic signs, speed control and detention, implemented on an autonomous vehicle, for which the processes that are carried out in advanced driver assistance systems, also the algorithms of artificial vision system were studied. The prototype bases its operation on the use of free software by programming in Python with OpenCV libraries and the training of the convolutional network for pattern recognition using techniques of artificial vision such as the Haar Cascade algorithms, with the use of these the processing necessary for the detection of traffic signals and the control of vehicle speed. On the recognition of these signals, a Raspberry OV5647 camera was used, which was placed on the top of the vehicle, so the module received the information and processed it through the algorithm using the training described giving the control order on the driver L298N, which controls the vehicle's travel speed. Based on the tests carried out, it was determined that the detection of signals has a 99.16% accuracy; therefore, it is concluded that the implemented prototype presents a high degree of functionality at the time to meet its objective, being able to be improved and applied to real vehicles, being capable of contributing to the reduction of traffic accidents. It is recommended to use 8GB Raspberry pi 4B due to its efficiency in processing; if you need more resources, you can use accelerators designed for artificial vision applications to obtain greater precision and fewer mistakes in recognition.

Keywords: <ARTIFICIAL VISION SYSTEM> <CONVOLUTIONAL NETWORKS> <TRAFFIC SIGNS> <CASCADE TRAINER GUI> <AUTONOMOUS VEHICLE>.

Firmado electrónicamente por:
LENIN IVAN LARA

INTRODUCCIÓN

El Trabajo de investigación curricular presentado a continuación describe el uso de visión artificial en un prototipo para la detección de señales de tránsito y control de velocidad, el estudio brinda una herramienta de apoyo para reducir el número de accidentes de tránsito. Actualmente los sistemas que emplean visión artificial mediante la utilización de las técnicas adecuadas, permiten la obtención, procesamiento y análisis de cualquier tipo de información especial obtenida a través de imágenes digitales, por lo que se ha considerado la utilización de esta tecnología en el proyecto obteniendo resultados satisfactorios. El Proyecto se encuentra dividido en cuatro capítulos, los cuales describen las diversas partes que componen la investigación.

El Capítulo I describe el problema centrado en los accidentes de tránsito producidos por distracción del conductor quienes al no respetar las señales de tránsito provocan accidentes generando pérdidas económicas e incluso la vida de las víctimas. Se presenta la solución al problema de acuerdo a objetivos que deben cumplirse al finalizar la investigación.

El Capítulo II expone los antecedentes investigativos del proyecto, los fundamentos teóricos necesarios para la sustentación del mismo, y el planteamiento de la solución al problema. El Capítulo III presenta el marco metodológico, se describe el análisis de factibilidad del proyecto, la descripción de los componentes del mismo, y la selección del software y hardware requerido para el desarrollo. Se diseña e implementa el prototipo del Sistema de detección de señales de tránsito y control de velocidad empleando Visión Artificial.

El Capítulo IV describe los resultados obtenidos del proyecto, protocolo de pruebas, análisis de costo y presupuesto, al final se presentan las conclusiones y recomendaciones elaboradas a partir del desarrollo del proyecto, para que puedan ser utilizadas en beneficio de futuras investigaciones.

CAPÍTULO I

1. DIAGNÓSTICO DEL PROBLEMA

En este capítulo, se realiza el planteamiento del problema en la cual se expone el asunto o cuestión que se tiene como objeto aclarar.

1.1 Antecedentes

En la actualidad existe dependencia entre el conductor y el vehículo, en donde para conducir siempre ha sido necesario que el usuario se mantenga con una concentración total para la conducción, hoy en día existen tecnologías que abren una puerta hacia la innovación y la creatividad, las cuales permitirán una gama de herramientas para desarrollar diversos sistemas siendo la visión artificial un área de investigación muy activa en donde todavía se considera muy complicado la identificación de patrones a causa de la variabilidad en la aspecto del entorno humano.

Para Néstor Torres, ingeniero informático, el problema inicia en que la inteligencia artificial no es aún autónoma ni lo será hasta dentro de mucho tiempo, no es capaz de improvisar, sigue respondiendo a una lógica matemática, a una programación, y si algo ilógico pasa, hay que reprogramar y crear nuevos algoritmos. No es comprensible para una máquina programada y basada en un algoritmo, que se deje de comprar teléfonos para adquirir papel higiénico. La inteligencia artificial puede calcular, pero Torres dice que no puede todavía hacer algo que es potestad del ser humano (TIERRA GUSQUI, y otros, 2017): improvisar para corregir. Los modelos de aprendizaje automático fueron diseñados para responder a los cambios. Pero la mayoría de estos sistemas son frágiles: funcionan mal cuando los datos de entrada difieren mucho de los datos de entrenamiento.

La seguridad es un tema muy importante para los conductores, así como también para los peatones. En la actualidad muchos vehículos cuentan con sistemas de ayuda al usuario, como lo son asistente de arranque en pendiente, auto parqueo, respuesta automática a colisiones y muchos sistemas que facilitan la conducción para los usuarios de vehículos. Pero dichos sistemas no cumplen con el requisito de tener procesamiento de imágenes o Visión Artificial.

A nivel mundial, en especial en los países desarrollados en el aspecto tecnológico; los vehículos autónomos han trascendido al uso de Visión Artificial, es decir; los vehículos autónomos son capaces de reconocer su entorno y reaccionar al mismo identificando la señalética y obstáculos presentados en la ruta.

Desde hace mucho tiempo se habla mucho de los vehículos autónomos, especialmente de los coches futuristas, pero también de aviones, submarinos, barco, etc. Estos sistemas apuntan a traer mejoras en seguridad, optimización de recursos y reducción de costos, debido a que se elimina el factor humano.

Para conseguir esto se utiliza un gran número de sensores, como: GPS, radares o cámaras, y muchas técnicas originarias de ramas diversas como son la ingeniería de control, la inteligencia y la visión artificial.

Una de las principales necesidades de estos vehículos es la de capturar una gran cantidad de información en poco tiempo, por lo que la visión artificial es perfecta, ya que una cámara puede capturar gran cantidad de datos del entorno en centésimas de segundo, pero ¿Que pueden aportar las técnicas de visión artificial a este tipo de vehículos?

A medida que la población va en aumento lo hace también el número de vehículos por lo cual es importante tomar en consideración los accidentes de tránsito que se producen, cada día alrededor de 3500 personas mueren en las carreteras. Muchos millones de personas sufren lesiones o discapacidades cada año. Los niños, los peatones, los ciclistas y los ancianos son los usuarios más vulnerables de la vía pública todo a causa del irrespeto a las señales de tránsito, el cansancio, la falta de concentración o fallos en la visión.

La OMS colabora con asociados gubernamentales y no gubernamentales en todo el mundo para prevenir los accidentes de tránsito y promover las buenas prácticas en el cumplimiento de las señales de tránsito como conducir con precaución y evitar los excesos de velocidad (PICO APONTE, y otros, 2019 pág. 128).

Uno de los grandes problemas en la actualidad es el exceso de velocidad y el irrespeto a las señales de Stop, ocasionando muchos accidentes.

Cuando hablamos de seguridad vial, América Latina sigue en primer lugar en el Rankin mundial de las regiones con las tasas de mortandad más altas por accidentes de tránsito. Es por esta razón que la seguridad vial se ha transformado en un tema de plática obligado entre los gobiernos de los países de América Latina y el Caribe.

En América Latina, la venta de sistemas de visión artificial creció en un 6% en el segundo semestre de 2013, los cuales son empleados en su mayoría en aplicaciones de control de calidad y control de procesos, siendo orientados pocos sistemas a la mejora de los vehículos autónomos, existiendo una gran concentración de uso en países como Singapur, Países Bajos, Noruega, Estados Unidos (LÓPEZ MÁRQUEZ, y otros, 2016).

Ecuador avanza en el desarrollo tecnológico, gracias a los emprendimientos de profesionales en las áreas de informática, la electrónica, la mecánica y las telecomunicaciones; sobre todo por la influencia de trabajos de grado, proyectos de investigación o tesis de grado; que impulsan las universidades, escuelas politécnicas y centros de educación superior. En el país, la detección de señales de tránsito con Visión Artificial en vehículos autónomos; es un tema, que poco o casi nada se lo ha puesto en práctica, tampoco se lo ha diseñado o implementado

1.2 Formulación del problema

¿Existe un sistema de visión artificial basado en redes convolucionales para la detección de señales de tránsito y el control de un vehículo autónomo utilizando un sistema embebido?

1.2.1 Sistematización del problema

- ¿Qué tan factible seria la implementación de un sistema de reconocimiento de señales de tránsito a través de visión artificial?
- ¿Cómo se podría adaptar un vehículo autónomo a escala para que sea capaz de controlar su velocidad y detenerse utilizando visión artificial?
- ¿Qué factores externos intervienen en el correcto funcionamiento del vehículo autónomo?
- ¿Cuáles son las restricciones por tomarse en cuenta para la correcta detección de las señales de tránsito?
- ¿Qué parámetros definen la eficiencia de un sistema de visión artificial basado en redes convolucionales que nos permitan validar su correcto funcionamiento?

1.3 Justificación del trabajo de titulación

A continuación, se presenta tanto la justificación teórica como la justificación aplicativa de este trabajo de titulación, demostrando porque es pertinente llevarlo a cabo.

1.3.1 Justificación teórica

Gracias al conocimiento adquirido durante la formación académica del transcurso de la carrera de Ingeniería en Electrónica Control y Redes Industriales de la ESPOCH se propone el presente Trabajo de Integración Curricular (TIC) bajo sus lineamientos, donde las materias de Inteligencia Artificial (Visión Artificial), programación, son la parte fundamental para desarrollar este tipo de proyecto además de reforzar he incrementar los conocimientos, ya que la investigación se la asume como una función prioritaria dentro del Visión e Inteligencia artificial.

El aporte científico de este proyecto servirá para futuros estudiantes que quieran mejorar el diseño de este trabajo de investigación o tomarlo de guía para su uso en el área de conducción autónoma

y seguridad. De igual manera se justifica técnico y tecnológicamente porque existe el hardware que presta las facilidades para cubrir cada uno de los requerimientos que se susciten durante el desarrollo, y el *software* se ha creído conveniente utilizar PYTHON como lenguaje de programación (CUEVAS JIMENEZ, y otros, 2007).

La inversión que se realizará para el desarrollo del sistema informático será solventada por el desarrollador, por lo que se empleará un gran porcentaje del trabajo intelectual adquirido durante la formación académica, asimismo al poner en ejecución la presente propuesta se estaría haciendo un ahorro considerable en lo que respecta a tiempo, y consecuentemente en la agilidad de los procesos (CUEVAS JIMENEZ, y otros, 2007).

1.3.2 Justificación aplicativa

Mucho tiempo antes de que los vehículos fueran introducidos en el mercado ya se presentaban lesiones causadas por el tráfico en las cuales se veían involucradas las personas y los medios de transporte empleados en aquel tiempo. Las cifras fueron creciendo con la aparición y proliferación de automóviles, autobuses, camiones y otros vehículos de motor (OMS, 2004).

A nivel nacional y por medio de los medios de comunicación se ha presentado que diariamente se suscita una gran cantidad de accidentes vehiculares en las vías públicas que provocan pérdidas de vidas, individuos con discapacidades permanentes y grandes pérdidas materiales. Dichos accidentes son ocasionados principalmente por imprudencia de los conductores ya que muchas veces ignoran los límites de velocidad o señales de stop en las vías (OMS, 2004), (PICO APONTE, y otros, 2019). El tema de investigación pretende ser empleado como fuente de consulta y como un apoyo a la seguridad vial ante la presente problemática.

La Visión Artificial es una herramienta útil para evitar las pérdidas humanas, económicas y materiales al prevenir accidentes debido a fallas humanas, como el irrespeto a las señales de tránsito o por fallas en la visión del conductor, con lo que la conducción será controlada por un software que pueda procesar imágenes y extraer rasgos o patrones característicos; se puede prevenir accidentes e infracciones al llevar la conducción a un entorno inteligente; propio de los algoritmos de Visión Artificial y dando un nivel de autonomía al vehículo en el control de velocidad y Stop. Existen programas de computadora como MATLAB, LABView, OpenCV, NeuroCHECK, MVIAT, APHELION, PYTHON que poseen algoritmos de procesamiento de imágenes; además existen módulos como Raspberry que permiten establecer comunicación entre los sistemas inteligentes y el móvil; al combinar la tecnología de Visión Artificial que tenemos actualmente con las comunicaciones y el control del vehículo, la seguridad para los ocupantes del vehículo puede trascender a un mejor nivel. Los beneficiarios directos del trabajo de investigación son las personas que se movilizan en vehículos y los peatones. También estudiantes y futuros

investigadores ya que al determinar cuál es el mejor algoritmo en detección de señales de tránsito, ahorraran mucho más tiempo en sus proyectos, adicionalmente. El desarrollo de este trabajo de investigación se fundamenta en la disposición de suficiente fuente bibliográfica. Procede con la ayuda de un tutor y apoyado por docentes. Además, cuenta con la existencia de una plataforma informática eficiente en algoritmos de procesamiento de imágenes.

El "*Desarrollo de un sistema de visión artificial basado en redes convolucionales para la detección de las señales de tránsito implementado sobre un vehículo autónomo*"; será factible implementarlo, en virtud del avance tecnológico en el *software* y *hardware,* las telecomunicaciones, el ancho de banda y otras categorías técnicas que facilitan la consecución de este trabajo de investigación. Vincular la tecnología de Visión Artificial con los sistemas de seguridad y reglamentos de tránsito, para buscar el bienestar del conductor y los peatones, se justifica el desarrollo de este trabajo de investigación; dado que promueve el uso de la técnica y los conocimientos para buscar la solución a un problema real, que causan pérdidas humanas, económicas y materiales desde hace mucho tiempo a las personas que son propensas a accidentes tanto por negligencia como por fallos en la visión del conductor. Otro de los aspectos importantes en el desarrollo de este proyecto es que se determina que algoritmo es el que ofrece mejores prestaciones y rendimiento.

1.4 Objetivos

1.4.1 Objetivo general

Desarrollar de un sistema de visión artificial basado en redes convolucionales para la detección de señales de tránsito, control de velocidad y detención, implementado sobre un vehículo autónomo.

1.4.2 Objetivos específicos

- Determinar las características del sistema inteligente de reconocimiento de señales de tránsito con visión artificial.
- Adaptar un vehículo autónomo a escala para que sea capaz de detectar señales de velocidad y stop utilizando un sistema de visión artificial.
- Establecer el método de control de las señales de tránsito sobre el vehículo autónomo.
- Establecer las restricciones que limitan el correcto funcionamiento del sistema.
- Diseñar un protocolo de pruebas sobre el sistema analizando parámetros que definen el funcionamiento de una red convolucional, la velocidad del vehículo autónomo y los errores cometidos.

CAPÍTULO II

2. FUNDAMENTOS TEÓRICOS

A continuación, se presenta la revisión literaria y fundamentos teóricos necesarios para la realización del proyecto.

2.1 Vehículos autónomos

Un auto autónomo es capaz de analizar el entorno en el que se encuentra y pilotar por él sin que sea necesaria intervención del conductor. También, analizan sus alrededores mediante el uso de sensores, como lo son: sistemas de posicionamiento global GPS, radares, visión computarizada, laser, sensores ultrasónicos y más. La información recolectada por los sensores es procesada por el sistema avanzado, para identificar obstáculos, señales de tránsito y seleccionar rutas de navegación apropiadas.

Los primeros coches autónomos (por tanto, verdaderamente autónomos) aparecieron en los años 80 con los proyectos Navlab y ALV de la Carnegie Mellon University (Pittsburgh, Pennsylvania) en 1984 y el proyecto EUREKA Prometheus de Mercedes Benz. Desde entonces, las compañías y grupos de investigación han enfocado en desarrollar prototipos de vehículos autónomos, siendo las más reconocidas en la actualidad las empresas Tesla Motors y Google (GUTIÉRREZ MORENO, 2009).

2.1.1 Clasificación

Al presente existen dos métodos propuestos con los cuales se llega a clasificar los vehículos autónomos: el de la National Highway Traffic Safety Administration (NHTSA) y el de la Sociedad de Ingenieros Automotrices (SAE). Estas categorizaciones son muy parecidas, por lo que solo se va a presentar la clasificación propuesta por la SAE debido a que es un organismo internacional de estandarización.

2.1.1.1 Estándar SAE J3016

Los objetivos principales del estándar J3016 publicado por SAE International tiene seis (GUTIÉRREZ MORENO, 2009):

- Identificar seis niveles de conducción automática, iniciando por “sin automatización” hasta “automatización completa”.
- Establecer una base de niveles y definiciones sobre la funcionalidad de la tecnología.

- Referir las diferencias de forma categórica para una progresión gradual mediante los distintos niveles de automatización.
- Tener presente las actuales prácticas de la industria.
- Aclarar las posibles confusiones y debe ser útil en numerosas disciplinas (ingeniería, medios legales y discurso público).
- Educar a una comunidad más amplia, explicando para cada nivel cual es el papel de los conductores en la ejecución de la tarea de conducción dinámica mientras se el sistema de conducción automática.

Los seis niveles propuestos en este estándar se establecen por la cantidad de acciones y atención requerida por el conductor del vehículo autónomo. Presentando los siguientes:

- **Nivel 0 Sin automatización:** El sistema automático solo es capaz de emitir alertas presentadas al conductor, pero no ejecuta ninguna acción sobre el vehículo.
- **Nivel 1 Ayuda al conductor:** Permite ejecuta un modo especifico de conducción por un sistema automatizado que controla la aceleración/desaceleración o la dirección del vehículo durante la conducción. El conductor debe estar preparado para tomar el control del vehículo en cualquier momento.
- **Nivel 2 Automatización parcial:** El conductor está obligado a visualizar obstáculos y eventos, y a responder si el sistema automático no es capaz de responder de forma adecuada. El sistema automático se encarga del frenado, aceleración y la dirección. El sistema puede ser desactivado inmediatamente, con lo cual el control del vehículo es por el conductor.
- **Nivel 3 Automatización condicional:** Dentro de algunos entornos condicionados como lo son las autopistas el conductor puede permitir que el vehículo autónomo tome el mando de forma segura, sin que el conductor deba prestar atención, pero tiene que estar preparado por si el sistema automático requiere su intervención.
- **Nivel 4 Alta automatización:** El sistema automático es capaz controlar el vehículo sin necesidad de la intervención del conductor, factible en casi todos los ambientes, como el mal tiempo. El conductor puede activar el sistema automático solo y solo si es seguro hacerlo. Mientras este se encuentre activado, no es necesaria la atención por parte del conductor.
- **Nivel 5 Automatización completa:** No se requiere intervención humana, tan solo basta con establecer el destino e iniciar el sistema. Esta clase de sistemas automáticos se pueden dirigir a cualquier localización geográfica en la cual sea legal conducir.

2.2 Optimización

En las décadas pasadas, el término optimización se ha referido al mundo de la informática. Sin embargo, es un concepto que también se contempla en las matemáticas, la gestión de procesos y la economía.

2.2.1 Definición

Optimización hace referencia al efecto y acción de optimizar. En términos generales, se describe como la capacidad de hacer o resolver alguna cosa de la manera más eficiente posible, utilizando la menor cantidad de recursos.

2.2.2 Técnicas de optimización

En los años 50 aparecen las técnicas de Investigación de operaciones, a partir de esto comienza a desarrollarse la metodología para su utilización. Sus referencias se basan en las investigaciones de Isaac Newton, Ackoff, Charnes, George Dantzing y Cooper, Churchman y Zimmerman.

Esta metodología se sostiene en los supuestos presentados a continuación:

- Alternativa en las decisiones.
- Posibilidades de crear una base informática.
- Posibilidades mínimas de poder utilizar los resultados.

En este proceso existe pasos para llegar a la obtención de los objetivos planteados:

- Observación e identificación del problema.
- Formulación general.
- Construcción del modelo.
- Generación de una solución.
- Prueba y evaluación de la solución.
- Implantación.
- Perfeccionamiento y desarrollo.

2.2.2.1 Observación

Se estudia el fenómeno como tal, las posibles variables, las interrelaciones que tiene, el sistema organizativo bajo el cual se halla el fenómeno, se escuchan los criterios de expertos, se examina el cumplimiento de las premisas esenciales de las técnicas de optimización, que son:

- Alternativa de decisión.

- Condiciones de linealidad o no.
- Mínimas condiciones organizativas.

Se define conceptualmente y de manera correcta cuál es el problema por resolver, se establecen hipótesis y se enuncian los objetivos, se consulta la bibliografía especializada. Se realizan contactos Inter especialistas y como último, se elabora una ficha con un pequeño resumen de la observación realizada.

2.3 Sistema inteligente

A continuación, se presenta la definición de sistemas inteligentes, las capacidades requeridas para considerar a un sistema inteligente, así como cada uno de los aspectos que conforman al mismo.

2.3.1 Definición

Un sistema inteligente es aquel capaz de solucionar problemas complejos y multidisciplinares de forma automática dando soporte a las decisiones de un técnico. pueden ser numerosas aplicaciones y de una gran variedad, como sistema inteligente como soporte de decisión en telemedicina hasta un sistema inteligente de tratamiento de datos e imágenes.

2.3.2 Capacidades requeridas

Para considerar un sistema inteligente completo, debe incluir diversas funcionalidades. Para usos prácticos usamos:

- **La inteligencia:** Como el nivel del sistema en lograr sus objetivos de los sistemas inteligentes.
- **Sistematización:** Un sistema forma parte del universo teniendo extensión limitada en tiempo y espacio. Las partes del sistema siempre tienen fuertes correlaciones con otras partes de este; que con partes fuera del sistema.
- **Objetivo:** Un objetivo es lo que el sistema inteligente quiere lograr. Suelen haber muchos niveles de objetivos, puede hacer un objetivo principal y muchos subjetivos específicos.
- **Capacidad sensorial:** Es la parte del sistema que es capaz de recibir comunicaciones del entorno. Los sentidos son necesarios para que el sistema inteligente conozca su entorno y funcione interactivamente.
- **Conceptualización:** Un concepto es el elemento físico del pensamiento. Es el almacenamiento físico de la información (en neuronas o electrones). Todos los conceptos de memoria son interdependientes en una red. La formación de conceptos implica el desarrollo de niveles de abstracción.

- **Regla de retroalimentación:** La regla de retroalimentación es el resultado de un experimento o el resultado de interpretar la propia memoria. Conectar la situación y las consecuencias de la acción.
- **Memoria:** La memoria es el almacenamiento físico de conceptos y reglas de funcionamiento. Esto incluye la experiencia del sistema.
- **Aprendizaje:** El aprendizaje es quizás la capacidad más importante de un sistema inteligente. El sistema aprende por medio de la información recibida de los sentidos. Aprenda las reglas de conducta asadas en su experiencia. El rendimiento a veces hecho al azar se almacena en su valor. La regla de retroalimentación aumentará su valor si da en el lanco. El aprendizaje implica fijar conceptos abstractos construir ejemplos concretos y crear conceptos complejos que contengan conceptos de partes de un objeto. El aprendizaje es también la capacidad de detectar relaciones (patrones) entre la parte "*situacional*" y la parte de "*situación futura"* de una regla de respuesta.

2.3.3 Métodos inteligentes

A continuación, dentro de los métodos inteligentes se describe la lógica difusa, funcionamiento y los algoritmos genéticos seleccionados por ser los principales métodos utilizados en inteligencia artificial o a su vez los más utilizados.

2.3.3.1 Lógica difusa

La lógica difusa se basa en lo relativo del evento observado como posición diferencial. Este tipo de lógica toma dos valores aleatorios, pero contextualizados y alusivos entre sí. Así, una persona que mida dos metros es alta, si anticipadamente se ha tomado el valor de persona baja y se ha establecido en un metro. Los dos valores están contextualizados a personas y alusivos a una medida métrica lineal.

2.3.3.2 Funcionamiento

La lógica difusa se adapta mejor al mundo real, incluso puede comprender y funcionar con nuestras expresiones, del tipo (hace mucho frio), (no es muy bajo), (el ritmo del corazón está un poco lento), etc.

El éxito de esta adaptación al lenguaje está en comprender los cuantificadores de cualidad para nuestras inferencias (en los ejemplos antes mencionados, mucho, muy y un poco).

En la teoría de conjuntos difusos se establece también las operaciones de diferencia, negación, unión, intersección o complemento, y otras operaciones sobre conjuntos, en los que se basa esta lógica.

Sobre cada conjunto difuso, existe asociada una función de pertenencia en los elementos, esto indica en qué medida el elemento pertenece a ese conjunto difuso. Las formas de las funciones de pertenencia más usuales son lineal, trapezoidal y curva.

Se basa en reglas heurísticas, así como: SI (antecedente) ENTONCES (consecuente), donde el consecuente y el antecedente son también conjuntos difusos, pueden ser puros o resultado de operar con ellos.

De las reglas que dispone el motor de inferencia de un sistema difuso pueden ser establecidas por expertos, también aprendidas por el sistema, usando en este caso de redes neuronales para fortalecer la toma de decisiones.

Los datos de entrada casi siempre son recogidos por sensores que miden las variables de entrada al sistema. El motor de inferencias se fundamenta en chips difusos, que están aumentando su capacidad de procesamiento de reglas año a año.

Un esquema de funcionamiento característico para un sistema difuso podría ser de la forma que se presenta a continuación en la figura 1-2.

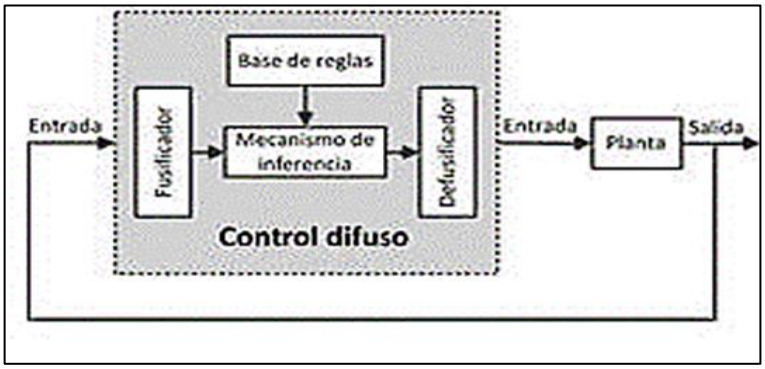


Figura 1-2: Funcionamiento de un sistema de control difuso

Realizado por: Sanango Andrés, 2022.

2.3.3.3 Algoritmos genéticos

Un algoritmo es una serie de pasos organizados que describen el proceso a seguir para resolver un problema específico.

Antena de la nave espacial ST5 de la NASA en 2006 este modo de aplicación fue descubierto por un programa de diseño de computadora evolutivo cuidadosamente diseñado para crear patrones de radiación óptimos. Se llama una antena avanzada.

En la década de 1970, impulsada por John Henry Holland, surgió una de las rutas más prometedoras hacia la inteligencia artificial, el Algoritmo Genético (GA, por sus siglas en inglés). Se nombran así porque están basados en la evolución biológica y su base genética molecular.

Estos algoritmos hacen evolucionar grupos de individuos realizando acciones aleatorias sobre grupos de individuos similares a las que ocurren en la evolución biológica (mutación genética y recombinación) y seleccionando en base a ciertos criterios para decidir qué individuos encajan mejor y cuáles sobreviven, los menos Las que caben, las que se desechan.

Los algoritmos genéticos pertenecen a los algoritmos evolutivos, que también incluyen estrategias evolutivas, programación evolutiva y programación genética. Trabajan con un conjunto de soluciones a un problema llamado fenotipo y un conjunto de individuos en una población natural, codificando información sobre cada solución en una cadena, generalmente binaria, llamada cromosoma. Los símbolos que componen la cadena se denominan genes. Cuando un cromosoma está representado por una cadena de números binarios, se le llama genotipo.

En cada generación, los cromosomas se evalúan mediante la aptitud. Las generaciones subsiguientes (nuevos cromosomas) se producen mediante la aplicación repetida de operadores genéticos, a saber, operadores de selección, cruce, mutación y reemplazo. Los algoritmos genéticos han demostrado su eficacia en los casos en los que se quiere calcular funciones no diferenciables (o derivaciones muy complejas), aunque pueden utilizarse para cualquier función.

Se debe considerar lo siguiente:

- Si la función a optimizar tiene muchos máximos/minutos locales, entonces se necesitan más iteraciones del algoritmo para "asegurar" los máximos/minutos globales.
- Si la función a optimizar contiene varios puntos con valores muy cercanos al valor óptimo, solo podemos "garantizar" encontrar uno (no necesariamente el valor óptimo).

2.4 Redes Neuronales

Son parte del campo de la IA conocido como Inteligencia Computacional (CI), ha contribuido en gran medida al éxito del reconocimiento de patrones. CI se enfoca en el estudio, diseño e implementación de modelos inspirados biológica o lingüísticamente, que incluyen, además de Redes Neuronales Artificiales (RNA, por sus siglas en inglés), algoritmos genéticos, programación evolutiva, sistemas difusos y sistemas híbridos inteligentes.

Las RNA se utilizan para construir funciones clasificación y discriminación cuando las reglas decisivas del sistema no son evidentes. También, son modelos matemáticos basados en el comportamiento de las neuronas biológicas. Aunque estos modelos son muy distintos de

simulaciones existentes del sistema nervioso de seres vivos, tienen características que les permiten aprender a partir de ejemplos además de abstraer y generalizar conceptos.

Haykin define una RNA como "*un sistema masivamente paralelo y distribuido compuesto por unidades de procesamiento simples que, naturalmente, tienden a almacenar el conocimiento adquirido a partir de la experiencia y lo hacen útil*". Imitando la computación "biológica", las RNA se caracterizan por un paralelismo masivo, un alto grado de interconectividad, resistencia al ruido y adaptabilidad al entorno (HAYKIN, 1999). Existe una gran variedad de modelos de RNA, pero en general todos tienen los siguientes componentes:

- Un conjunto de objetos de procesamiento "neuronas".
- Una regla de ejecución para cada elemento.
- Topología de interacción entre elementos de procesamiento.
- Un conjunto de reglas de propagación a través de conexiones entre elementos de procesamiento.
- Una regla de aprendizaje o algoritmo de aprendizaje de los parámetros que definen el modelo.
- El entorno en el que opera el sistema.

2.4.1 Red Neuronal Artificial (RNA)

Una parte destacada del campo científico de la Inteligencia Artificial es la que corresponde a las Redes Neuronales Artificiales (RNA), estas son aquellas redes en las que concurren elementos de procesamiento de información, que ejecutan interacciones locales y estás dependen del proceder del conjunto del sistema.

Una parte importante y destacada del campo de la ciencia de la inteligencia artificial es la que corresponde a las redes neuronales artificiales (RNA), que son aquellas redes que cuentan con elementos de procesamiento de información que interactúan localmente que dependen del comportamiento de la red neuronal, usuario y ajustes del sistema.

Las redes neuronales artificiales intentan simular el comportamiento del cerebro humano, que se caracteriza por el aprendizaje a través de la experiencia y la extracción de conocimientos generales a partir de un conjunto de datos. Estos sistemas simulan la estructura neuronal del cerebro de forma esquemática a través de programas informáticos, o modelando estructuras de procesamiento con algunas capacidades de computación paralela, o construyendo físicamente sistemas cuya arquitectura es cercana a la estructura de las redes neuronales biológicas (RIVAS, y otros, 2018).

La parte básica del sistema neuronal biológico es la neurona, que es una célula viva y por tanto contiene todos los elementos que componen una célula biológica, aunque contiene otros elementos que la diferencian. En general, las neuronas son cuerpos celulares más o menos esféricos o soma, que parten de un tronco o axón y de un denso árbol de ramas más cortas formado por dendritas. Además, los extremos axonales pueden ramificarse en sus orígenes y, a menudo, tener múltiples ramificaciones en sus extremos. La forma final de una neurona depende de la función que cumple, es decir, el lugar que ocupa en el sistema general y los estímulos que recibe.

2.4.2 Perceptrón simple

Frank Rosenblatt, un pionero del sistema neuronal más simple que puede usarse como un procesador de información eficiente. Porque algunas de sus propiedades pueden extenderse a otros sistemas nerviosos más complejos. El perceptrón se utiliza para resolver problemas linealmente separables y utiliza el modelo de McCullot-Pitts como neuronas. Este se muestra en la figura 2-2.

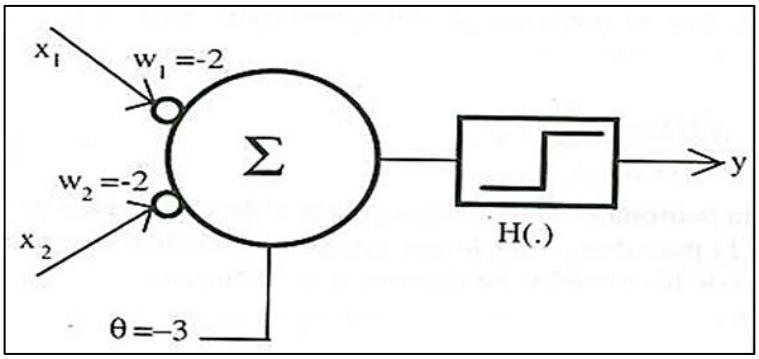


Figura 2-2: Perceptrón simple

Realizado por: Sanango Andrés, 2022.

Las características del perceptrón con las siguientes:

- Aprendizaje supervisado.
- Aprendizaje por corrección de error.
- Reconocimiento de patrones sencillos.
- Clasificación de patrones linealmente separables.

2.4.3 Perceptrón multicapa

Este modelo RNA es el más utilizado en la práctica, como la resolución de problemas de clasificación y regresión, y ha demostrado su condición de aproximador de funciones de propósito general, lo que justifica su estudio por separado y detallado (FLOREZ, y otros, 2008).

Un perceptrón multicapa es un modelo neuronal con propagación hacia adelante, basado en la organización de sus capas disjuntas de células, esta forma asegura que ninguna salida de neuronas constituye la entrada de neuronas en la misma capa o en la capa anterior, evitando así la conexión hacia atrás o la auto- Circulación. Como tal, esta arquitectura es idéntica a la de Simple Perceptrón, excepto que incluye una o más capas ocultas para aumentar la complejidad (FLOREZ, y otros, 2008). Esto se muestra en la figura 3-2.

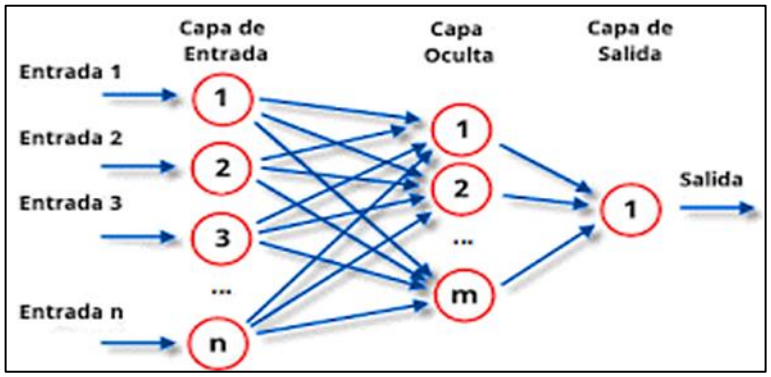


Figura 3-2: Perceptrón multicapa

Realizado por: Sanango Andrés, 2022.

2.5 Deep Learning (DL)

Buscan conjuntos de transformación directamente desde los propios datos. Este enfoque logra buenos resultados, especialmente en problemas pertenecientes al dominio de la visión artificial (CV), como la clasificación de escenas naturales y la detección de objetos. Los modelos DL también son adecuados para diferentes tareas médicas, como ayudar al diagnóstico temprano de la enfermedad de Alzheimer, la segmentación automática de lesiones causadas por la esclerosis múltiple, la detección de tumores en la ecografía mamaria (EM), y más. Sin embargo, solo unos pocos trabajos han explorado los métodos de DL en la dirección de la clasificación automática de las lesiones identificadas en la EM. métodos tradicionales de aprendizaje automático (CHANAMPE, 2019).

(ML), como las máquinas de vectores de soporte, los métodos del vecino más cercano y el análisis discriminante lineal se han utilizado como clasificadores para varios desarrollos en el diagnóstico temprano del cáncer de mama. Sin embargo, su capacidad para procesar datos en bruto es limitada (CHANAMPE, 2019).

2.5.1 Red Neuronal Convolucional (RNC)

Las redes neuronales convolucionales son análogas a la inteligencia artificial tradicional porque están compuestas por neuronas que se optimizan a sí mismas mediante el aprendizaje. Cada neurona recibe una entrada y realiza una operación de producto escalar, seguida de una función

no lineal, basada en una miríada de inteligencias artificiales. Desde el vector de imagen sin procesar de entrada hasta la salida final para la puntuación de clase, toda la red continúa expresando una función de puntuación dictada por un solo peso para obtener la mejor precisión de las imágenes capturadas.

Una de las mayores limitaciones de las formas tradicionales de redes neuronales artificiales es que tienden a tener problemas con la complejidad computacional requerida para calcular datos de imágenes. Los conjuntos de datos de referencia de aprendizaje automático comunes, como la base de datos MNIST de dígitos escritos a mano, son adecuados para la mayoría de las formas de RNA debido a sus dimensiones de imagen relativamente pequeñas de solo 28 × 28. Con este conjunto de datos, la primera capa oculta de una sola neurona contendrá 784 pesos (28 × 28 × 1, donde 1 tiene en cuenta que MNIST solo se normaliza a valores en blanco y negro), que para la mayoría de las formas de RNA es manejable.

2.6 Reconocimiento de patrones

A continuación, se introducen dos aspectos principales del reconocimiento de patrones, como su definición y las técnicas utilizadas en él, mientras que el reconocimiento de patrones se puede definir como la ciencia que se ocupa de la física o los procesos matemáticos, de ingeniería y computacionales relacionados con la física. Abstraer objetos para extraer información que permita establecer propiedades entre dichas colecciones de objetos.

2.6.1 Definición en Inteligencia Artificial

Algunas estrategias para el diseño de sistemas basados en el reconocimiento de patrones. El reconocimiento de patrones tiene como objetivo clasificar o categorizar objetos; los objetos pueden ser imágenes, audio, documentos, señales de sensores o algún otro tipo de medida disponible para una computadora. El término esquema generalmente se refiere a estos objetos o alguna representación de ellos.

En mundo moderno, el reconocimiento de patrones es parte fundamental en los sistemas que toman decisiones basados en inteligencia artificial (IA). La inteligencia artificial es el campo de la computación que se ocupa de los algoritmos que permiten que las máquinas perciban, razonen y actúen de manera similar a los humanos. (Para los lectores interesados en profundizar en la inteligencia artificial, se recomienda el curso en línea de UC Berkeley "Into Artificial Intelligence". El reconocimiento de patrones es considerado como "*una abstracción científica que involucra procesos como lo son los procesos de ingeniería, computacionales y matemáticos físicos y/o relacionados con la física, llamados esquemas, el propósito es extraer información de la cual se pueda establecer propiedades entre dichas colecciones o colecciones de objetos que*

nos permitan interpretar el mundo que nos rodea" (GÓMEZ, 2018). Normalmente, los objetos se representan por las características que se observan en ellos, en forma de vector:

- Se muestran seis gráficos, utilizando diferentes pares de características específicas en los ejes horizontal y vertical.
- La primera imagen usa el largo y ancho de los sépalos, se puede ver que las instancias de Versicolor y Virginica se superponen, pero usando el ancho de los pétalos y el largo de los sépalos se pueden separar mejor las instancias de cada clase.
- Entonces, dadas las características apropiadas, es posible encontrar funciones que distingan estas clases.
- Una función que asocia el vector de características de una instancia con la clase correspondiente se denomina función de clasificación o clasificador.

Estas instancias se muestran en la figura 4-2.

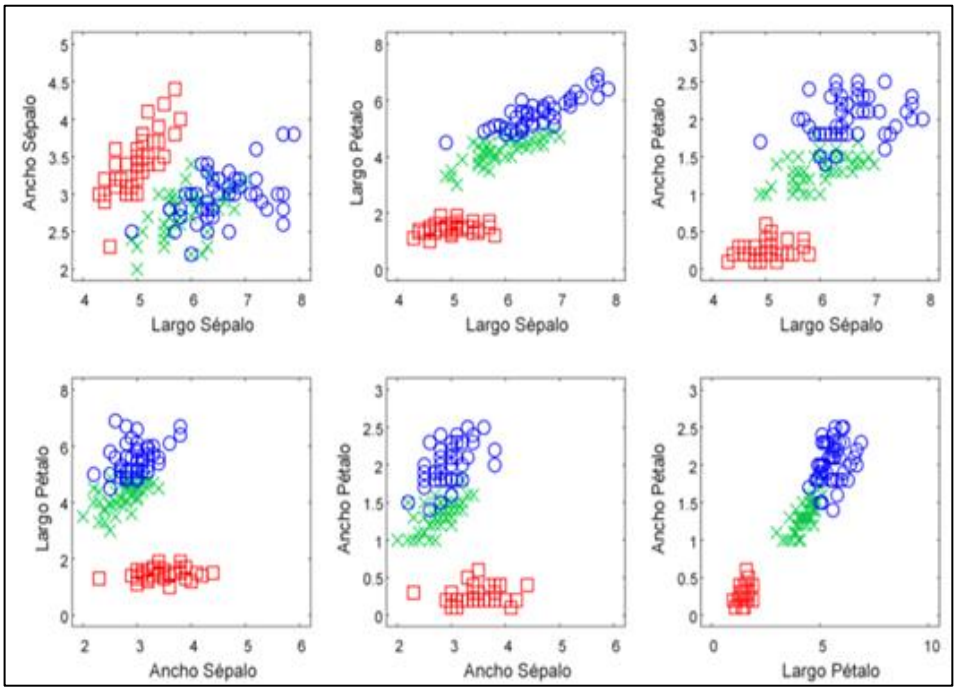


Figura 4-2: Visualizando las instancias con diferentes pares de características

Fuente: (GÓMEZ, 2018)

2.6.2 Técnicas de reconocimiento

Existen muchas técnicas para construir sistemas de reconocimiento; entre las más utilizadas se encuentran las siguientes (LÓPEZ MÁRQUEZ, y otros, 2016):

- **Estimación paramétrica:** Se basa en el acercamiento de la probabilidad condicional en que un patrón pertenezca a una clase, asumiendo una forma para dicha probabilidad y suponiendo el número de clases presentes.
- **Estimación no paramétrica:** Estos métodos ejecutan la clasificación sin asumir el número de clases ni tampoco la forma de las distribuciones de probabilidad.
- **Métodos estocásticos:** Se utilizan funciones aleatorias para diseñar funciones de clasificación.
- **Métodos basados en aprendizaje supervisado/no supervisado:** Se manejan modelos de aprendizaje de máquina para precisar a las funciones. Este aprendizaje se realiza a través de ejemplos de los patrones a clasificar, el cual es conocido como el conjunto de entrenamiento. Cuando se conoce previamente el valor de clase de los datos en el conjunto de entrenamiento, se expresa que el aprendizaje es supervisado; de lo contrario el aprendizaje se denomina "no supervisado".

2.7 Visión Artificial (VA)

Comprende muchos aspectos los cuales deben tomarse en cuenta para dominar la misma, en el presente ítem se define visión artificial y los conceptos teóricos que enmarca.

2.7.1 Definición

Es la ciencia y la tecnología que permite a las "máquinas" ver, extraer información de imágenes, resolver determinadas tareas o comprender el acontecimiento que están visualizando. En la actualidad, la visión artificial se utiliza en un amplio abanico de aplicaciones, desde el sector industrial (recuento de botellas, comprobación de defectos en líneas de montaje, interpretación de tomografías computarizadas médicas) y el sector médico (recuento y búsqueda de células), hasta el sector más sistemas complejos, permite que los robots se localicen en entornos desconocidos, reconocimiento de patrones a través de la realidad aumentada y muchas otras aplicaciones (NÚÑEZ, 2016).

Hoy en día, las técnicas de visión artificial se utilizan cada vez más en el campo del diseño de interacción a través de la interacción con superficies multitáctiles (multi-touch), la interacción con objetos tangibles y el reconocimiento de gestos corporales. Todos estos ejemplos incluyen técnicas de visión artificial. De alguna manera, en nuestro entorno, este conjunto de tecnologías nos permite diseñar interacciones para que los usuarios interactúen con las aplicaciones utilizando sus movimientos o manipulaciones de objetos (NÚÑEZ, 2016).

2.7.2 Luz visible y luz invisible

El ojo humano ve parte del espectro de toda la luz que ilumina el universo. El rango de luz que podemos ver se llama luz visible.Esto significa que hay frecuencias de luz que no podemos ver pero que existen, como el infrarrojo y el ultravioleta.

- El infrarrojo es un "color" que es invisible al ojo humano por debajo del rojo en frecuencia.

Los rayos ultravioletas son "colores" por encima del violeta que son invisibles al ojo humano. Esto se muestra en la figura 5-2 y su longitud de onda de colores en la tabla 1-2.

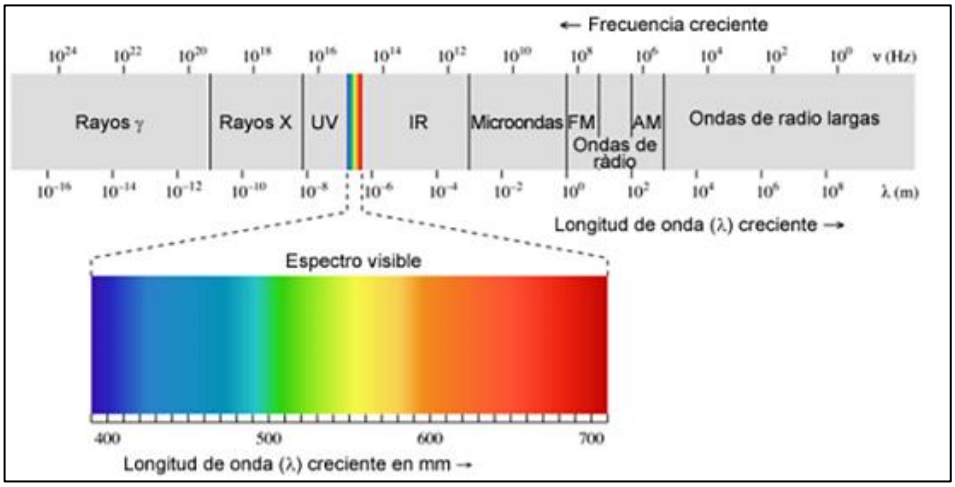


Figura 5-2: Espectrograma de la luz visible y no visible

Fuente: (FOTOGRAFÍA Y TÉCNICA, 2015)

Tabla 1-2: Longitud de onda de colores

Color	Longitud de onda
Ultravioleta	< 380 nm
Violeta	380 – 450 nm
Azul	450 – 495 nm
Verde	495 – 570 nm
Amarillo	570 – 590 nm
Naranja	590 – 620 nm
Rojo	620 – 750 nm
Infrarrojo	> 750 nm

Fuente: (FOTOGRAFÍA Y TÉCNICA, 2015)

Realizado por: Sanango Andrés, 2022.

De hecho, la mayoría de los sensores de las cámaras digitales son sensibles a la luz, pero también son (en diversos grados) sensibles a la luz infrarroja y/o ultravioleta. En este momento, la luz infrarroja no es útil para construir imágenes digitales porque nos brinda información de frecuencia que no podemos ver, por lo que no hay una representación de color subyacente. Por esta razón, la mayoría de las cámaras digitales se enfocan en construir una foto o marco con un filtro anti-IR para filtrar todas las frecuencias por debajo del espectro visible, es decir, el ruido no deseado, dejando un área que la cámara pueda capturar.

2.7.3 La imagen en movimiento

Los procesos de visión artificial son posibles gracias a las tecnologías basadas en la grabación de imágenes (cámaras, webcams), que aumentan la potencia de procesamiento de los ordenadores actuales. De hecho, la tecnología de captura comenzó en el siglo XIX, conferida con la invención del daguerrotipo (e incluso antes con el descubrimiento de la cámara oscura) y la posterior invención de la fotografía.

Descubrimientos posteriores, como la cronofotografía y otros pioneros del cine, eventualmente llevaron a la invención de la captura de imágenes en movimiento: una serie de tomas tomadas a altas frecuencias, reproducidas a la misma velocidad, para crear la ilusión de movimiento.

Actualmente, la mayoría de las tecnologías de captura de imágenes, teléfonos (cámaras) funcionan mediante el uso de sensores electrónicos (CCD y CMOS). En la primera década del siglo XXI, se estandarizó el uso de la fotografía y el video digital, grabando imágenes de alta resolución a un costo relativamente bajo y con muchos cuadros por segundo (FPS).

2.7.4 Adquisición de imágenes

A continuación, cubriremos algunas definiciones y aspectos necesarios a saber para obtener una imagen correcta, ya que se debe tener en cuenta la cámara, la iluminación y el entorno en el que se trabajará para lograr el mejor rendimiento posible de la imagen.

2.7.4.1 La cámara

Para comenzar con la visión artificial, precisamos un dispositivo sensible a la luz que permita almacenar imágenes en formato digital. En otras palabras, necesitamos una cámara web, una cámara de video o una grabadora analógica. Esto se muestra en la figura 6-2.

Figura 6-2: Cámara web

Fuente: (AMAZON, 2021a)

Las cámaras web y la mayoría de las cámaras se pueden conectar directamente a una computadora a través de un puerto USB y podemos capturar fotogramas en tiempo real. En cualquier caso, también se puede utilizar una tarjeta de captura analógica, que nos permite capturar y procesar imágenes a partir de señales de vídeo analógicas.

Una vez dentro de una instalación de inspección, se deben considerar múltiples aspectos del entorno en el que se produce la interacción, como las interrupciones de iluminación, la complejidad de la imagen u otros factores que pueden ser difíciles de realizar. Proceso de visión artificial. Recuerda que el proceso de "enseñanza" de tomar decisiones "visuales" implica muchas dificultades, por lo que es importante trabajar en un entorno donde la luz y los paisajes sean más fáciles y estables para analizarlos.

2.7.4.2 Iluminación de la escena

La detección de imágenes de la cámara o dispositivo de grabación varía según la iluminación del escenario. Un cambio suficientemente fuerte en el entorno de luz puede alterar todo el sistema informático de visión artificial de formas muy diferentes.

Por lo tanto, se debe trabajar en un entorno con un entorno de iluminación más estable tanto como sea posible.

En situaciones en las que no se dispone de un medio o de un entorno estable (como las pantallas del sistema al aire libre), el sistema de adquisición de imágenes debe adaptarse o adecuarse de alguna manera a las condiciones cambiantes a través de algoritmos adaptativos que permitan un análisis regular y secuencial de los cambios de temperatura. Iluminación ambiental y adaptación del sistema de visión a las nuevas condiciones. Algunas cámaras son especialmente sensibles a los infrarrojos porque nos permiten trabajar en entornos oscuros sin luz visible siempre que dispongamos de una fuente de luz infrarroja.

Los infrarrojos se pueden generar de diferentes maneras: hay LED infrarrojos que llenan una habitación de infrarrojos. Otro sistema menos costoso usa luz fuerte con tres filtros insertados, uno rojo, uno verde y uno azul (color sólido). El efecto de establecer un filtro rojo es hacer que la luz sea roja, es decir, todos los colores excepto el rojo se elimina de la luz filtrada. Cuando se coloca un filtro azul en este filtro, se eliminan todos los colores (incluido el rojo restante), excepto el azul (debido a la eliminación del filtro rojo, el azul ya no aparece en el filtro Luz). Y en ambos filtros, se agrega el filtro verde y los resultados son similares. Entonces, los tres filtros eliminan toda la luz visible, pero los filtros no eliminan las frecuencias infrarrojas. Lo que se hace es conseguir luz infrarroja a partir de lámparas incandescentes.

2.7.4.3 Proyecciones de video y visibilidad

En entornos interactivos, a menudo encontrará la necesidad de instalar un sistema de visión artificial en el mismo espacio escena que la proyección de video. Dado que algo se puede ver más tarde, la sofisticación de los algoritmos de visión artificial puede y son capaces de diferenciar el sistema entre proyecciones de video interactivas y los usuarios, por lo que, si dos imágenes están incrustadas en la misma sala, la manipulación suele ser a veces muy difícil. Como en el apartado anterior, podemos abordar estas dificultades gracias a nuestro conocimiento de cómo responde la luz y sus propiedades. Esto se muestra en la figura 7-2.

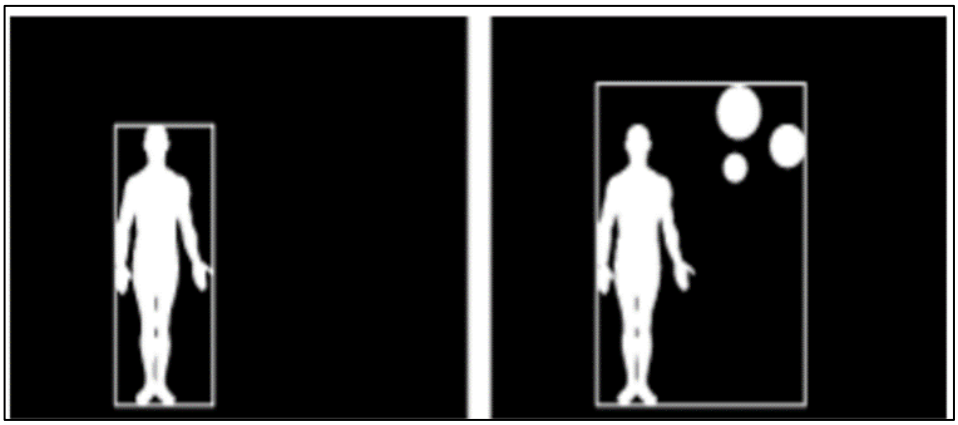


Figura 7-2: Comparación de imagen con y sin luz visible

Realizado por: Sanango Andrés, 2022.

Desde el espectro infrarrojo, la luz proyectada por el proyector es claramente visible para el ojo humano y/o el espacio digital normal. Cuando se agrega un filtro para bloquear la luz visible, solo se debe obtener y detectar la luz infrarroja. La cámara puede ver al usuario manipulando el sistema mientras su cuerpo rebota en la luz infrarroja, pero no puede ver la proyección (solo en el rango visible).

2.7.4.4 Entornos de programación

Para desarrollar aplicaciones que utilicen visión artificial, debe trabajar en un entorno de programación que pueda integrar bibliotecas OpenCV y conectarse con el resto del sistema interactivo. Aquí puede incluir entornos como MAX/MSP, procesamiento, marcos abiertos, datos puros y más. Todos estos sistemas de programación permiten generar sistemas interactivos que trabajan con gráficos, música, video, etc., mientras realizan funciones algorítmicas y humanas. En el campo industrial, médico y militar más especializado, existen numerosas alternativas y soluciones de *hardware* tangible (cámaras térmicas, cámaras lineales, salas de alta resolución o alta velocidad) y *software* en Open CV. Esto se visualiza en la figura 8-2.

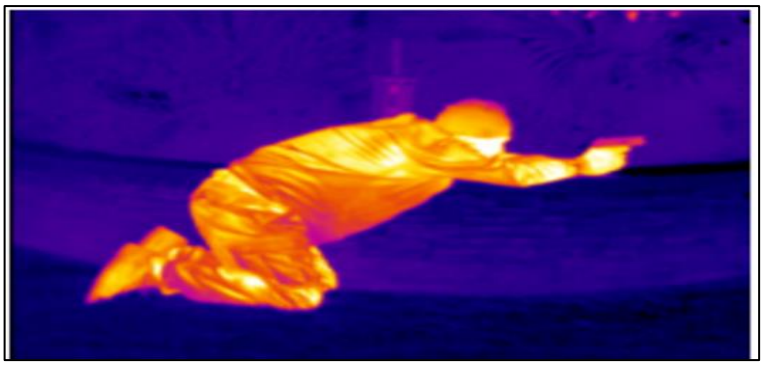


Figura 8-2: Imagen captada con mata térmica

Fuente: (X20, 2020)

CAPÍTULO III

3. MARCO METODOLÓGICO

En este capítulo se desarrolla como primer punto el diseño de hardware del prototipo. Se realiza el diseño en bloques de los módulos y dispositivos que lo conforman, también se especifica requerimientos técnicos que debe cumplir el mismo como consiguiente.

Los requerimientos de software que deben satisfacer a cada módulo del sistema están definidos en la segunda parte, eligiendo el apropiado entorno de desarrollo para la ejecución del proyecto.

3.1 Hardware

A continuación, se describen los requerimientos de hardware del prototipo, como lo son diseño y módulos electrónicos por utilizar.

3.1.1 Requerimientos para el diseño de hardware

Por medio de la revisión bibliográfica realizada en el capítulo anterior se pueden precisar los requerimientos de diseño para satisfacer el prototipo:

- Ser de costo razonable, facilidad en la implementar, fácil de manipular y que permita de forma confiable el control de la velocidad.
- Monitorear las señales de tránsito en tiempo real.
- Controlar la velocidad adecuada del vehículo durante su trayectoria.

3.1.2 Concepción de la arquitectura general

La concepción general del sistema se representa en la figura 1-3, donde se aprecia el módulo de adquisición, procesamiento y alarma que internamente se comunica mediante la arquitectura del *software.*

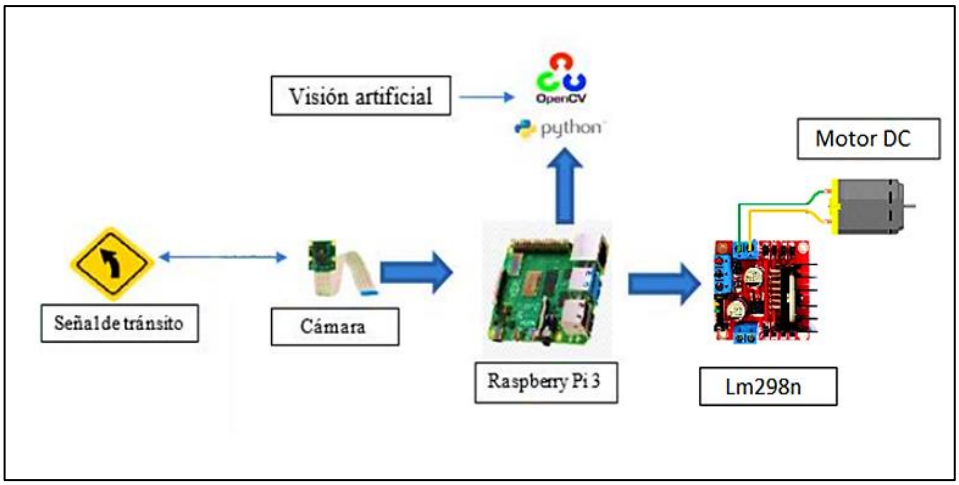


Figura 1-3: Arquitectura general

Realizado por: Sanango Andrés, 2022.

Como primer paso se tiene la adquisición de datos por medio de la cámara para obtener imágenes de las señales de tránsito, las cuales luego serán procesadas por la Raspberry, pasándolas por el software desarrollado y ejecutando la acción sobre el dispositivo Lm298n el encargado del control del motor DC.

3.1.3 Diseño de la arquitectura del módulo adquisición, procesamiento y actuadores

Una vez establecidos los requerimientos del diseño y concebida la arquitectura general del prototipo se presenta y detalla el diseño del módulo. En el gráfico 1-3, se muestra el diagrama en bloques del módulo de adquisición, procesamiento y actuador, donde se indica la interconexión de los cinco bloques que lo integran. El bloque recibe la información de la cámara de la vía, y la envía al bloque de procesamiento. Si sus valores no se encuentran dentro de los parámetros establecidos no realiza ninguna acción, caso contrario ejecutara un control sobre la velocidad.

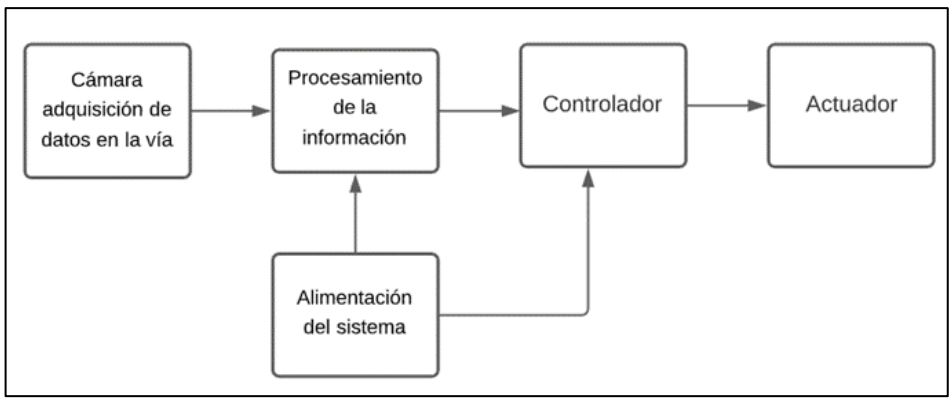


Gráfico 1-3: Diagrama de bloques del módulo de recolección y procesamiento

Realizado por: Sanango Andrés, 2022.

El bloque de procesamiento se basa en una tarjeta RASPBERRY PI 4 Modelo B de 8GB de RAM que hasta la presente fecha es la versión más actual, Utiliza un procesador quad-core ARM, posee

puertos HDMI, Ethernet, USB, conector GPIO, interfaz serial CSI-2 y DSI, los cuales son utilizados en el diseño del módulo.

3.1.4 Selección de los elementos del prototipo

A continuación, se detallan las principales características de los elementos que conforman el módulo del prototipo.

3.1.4.1 Raspberry Pi 4

En junio de 2019, se lanzó Raspberry Pi 4 Modelo B, el último de la popular línea de minicomputadoras Raspberry Pi. En comparación con la generación anterior de Raspberry Pi 3 Modelo B+, mejora la velocidad del procesador, el rendimiento multimedia, la memoria y la conectividad, al mismo tiempo que conserva la compatibilidad con versiones anteriores y un gasto de energía similar. Para los usuarios finales, Raspberry Pi 4 Modelo B ofrece un rendimiento comparable al de los sistemas de PC x86 de nivel de entrada (RASPBERRY PI, 2020a).

Las características clave de este producto incluyen un procesador quad-core de 64 bits de alto rendimiento, soporte para dos monitores con resoluciones de hasta 4K a través de un par de puertos micro HDMI, decodificación de video por hardware de hasta 4Kp60 y hasta 8 GB de RAM, LAN inalámbrica de doble banda de 2,4/5,0 GHz, Bluetooth 5.0, Gigabit Ethernet, USB 3.0 y capacidad PoE (a través del complemento PoE HAT independiente) (RASPBERRY PI, 2020a). Esto se muestra en la figura 2-3 y en la tabla 1-3.

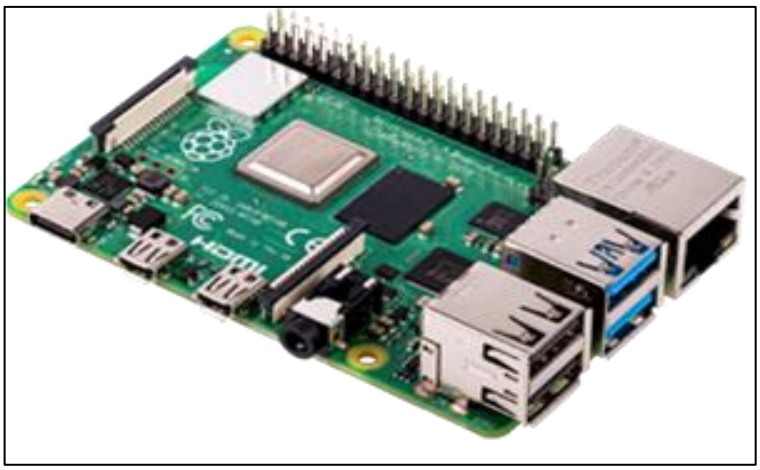


Figura 2-3: Raspberry Pi 4

Fuente: Sanango Andrés, 2022.

Tabla 1-3: Principales características de Raspberry Pi 4

Procesador:	Broadcom BCM2711, SoC de 64 bits Cortex-A72 (ARM v8) de cuatro núcleos a 1,5 GHz.
Memoria RAM:	SDRAM LPDDR4-3200 de 8GB.
Conexiones inalámbricas:	2.4 GHz y 5.0 GHz IEEE 802.11ac inalámbrica, Bluetooth 5.0, BLE.
Puerto de red:	Gigabit Ethernet.
Puertos USB:	2 puertos USB 3.0; 2 puertos USB 2.0.
Cabecera GPIO	Estándar de 40 pines Raspberry Pi (totalmente compatible con las placas anteriores).
Puertos micro-HDMI:	2, (hasta 4kp60 compatible).
Puerto de pantalla:	MIPI DSI de 2 carriles.
Puerto de cámara:	MIPI CSI de 2 carriles.
Puerto de video compuesto y audio:	estéreo de 4 polos.
Decodificador de video:	H.265 (decodificación 4kp60), H264 (decodificación 1080p60, codificación 1080p30).
Gráficos OpenGL:	3.0.
Ranura para tarjeta:	micro-SD para cargar el sistema operativo y el almacenamiento de datos.
Cargador USB:	5 V CC a través del conector USB-C (mínimo 3 A).
Pines de carga:	5 V CC a través del encabezado GPIO (mínimo 3 A).
Temperatura de funcionamiento:	0 - 50 grados C ambiente.

Realizado por: Sanango Andrés, 2022.

Se puede usar una fuente de alimentación de 2,5 A de calidad si los periféricos USB posteriores consumen un valor menor a 500 mA en total.

3.1.4.2 Cámara de visión nocturna Raspberry Pi

El módulo de cámara Raspberry Pi v2 substituyó el módulo de cámara original en abril de 2016. El módulo de cámara v2 posee un sensor Sony IMX219 de 8 megapíxeles (en comparación con el sensor OmniVision OV5647 de 5 megapíxeles de la cámara original).

El módulo de la cámara se puede usar para capturar videos y fotos de alta definición. Es fácil de usar para principiantes, pero tiene mucho que ofrecer a los usuarios avanzados si quieren ampliar sus conocimientos. Hay muchos ejemplos en línea de personas que lo usan para videos de lapso de tiempo, cámara lenta y otros dispositivos. También puede crear efectos utilizando la biblioteca que proporcionamos con su cámara (RASPBERRY PI, 2020b).

Figura 3-3: Cámara Raspberry nocturna

Fuente: (WAVESHARE, 2021)

Basta con decir que esto no es solo una mejora en la resolución, es un salto adelante en calidad de imagen, fidelidad de color y poca luz. Rendimiento. Admite en video los modos 1080p30, 720p60 y VGA90, también la captura de imágenes fijas. Se conecta al puerto CSI de la Raspberry Pi a través de un cable plano de 15 cm. La cámara funciona con los modelos de Raspberry Pi 1, 2, 3 y 4. Se puede utilizar a través de las API MMAL y V4L, y hay numerosas bibliotecas creadas para ella, incluida la biblioteca Picamera Python (RASPBERRY PI, 2020b).

Este módulo es muy versátil para usarlo en aplicaciones del hogar y también es el caso de la visión artificial.

Tabla 2-3: Principales características de la cámara nocturna Raspberry

Sensor:	OV5647 de 5 megapíxeles
Distancia de enfoque:	Ajustable
Tamaño del CCD:	1/4 de pulgada
Apertura (F):	1.8
Longitud focal:	3,6 mm
Ángulo de visión (diagonal):	75 grados
Mejor resolución del sensor:	1080p
Soporte:	4 agujeros para tornillos
Accesorios:	2 sensores infrarrojos
Potencia de salida:	3,3 V
Admite:	Conexión de LED infrarrojos y / o LED de flash de relleno
Dimensión:	25 mm x 24 mm

Realizado por: Sanango Andrés, 2022.

3.1.4.3 Driver puente h 1298n 2A

Es el dispositivo más utilizado para el control de motores DC de hasta 2 amperios. El chip L298N internamente tiene dos puentes H completos que permiten controlar 2 motores DC o un motor paso a paso bipolar/unipolar (NAYLAMPMECHATRONICS, 2021).

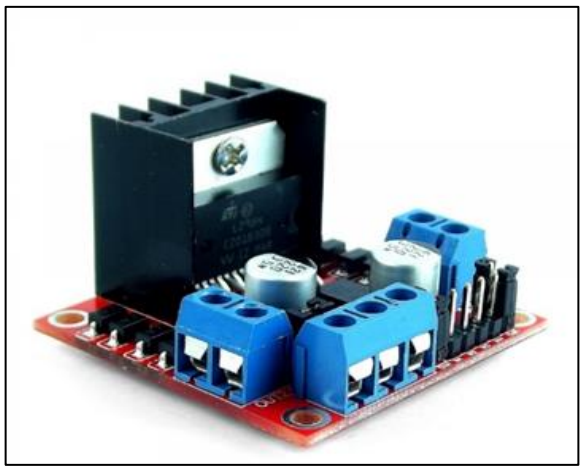


Figura 4-3: Puente h 1298n 2A

Fuente: (NAYLAMPMECHATRONICS, 2021)

Este módulo le permite controlar la dirección de rotación y la velocidad de los motores utilizando señales TTL disponibles desde microcontroladores y tarjetas de desarrollo como Arduino, Raspberry Pi o Texas Instruments Launchpads. La dirección de rotación de cada motor está controlada por dos pines, y la velocidad de rotación se puede ajustar mediante modulación de ancho de pulso (PWM).

Integra un regulador LM7805 de 5V el cual se encarga de alimentar la parte lógica del L298N, el uso de este regulador se realiza vía jumpers y puede ser utilizado para alimentar la etapa de control.

Tabla 3-3: Principales características del puente h l298n

Chip:	L298N
Canales:	2 (soporta 2 motores DC o 1 motor PAP)
Voltaje lógico:	5V
Voltaje de potencia (V motor):	5V - 35V DC
Consumo de corriente (lógico):	0 a 36mA
Capacidad de corriente:	2A (picos de hasta 3A)
Potencia máxima:	25W
Dimensiones:	43 X 43 X 27 mm
Peso:	30g

Realizado por: Sanango Andrés, 2022.

Debido al regulador de voltaje LM7805 integrado, el módulo se puede alimentar de 2 maneras. El módulo permite alimentación entre 6V y 12V DC cuando el puente de selección de 5V está conectado. Dado que el regulador está activo, el terminal etiquetado como +5V tendrá un voltaje de salida de 5V CC. Este voltaje se puede utilizar para alimentar la parte de control del módulo, ya sea un microcontrolador o un Arduino, pero recomendamos consumir no más de 500mA.

Al retirar el jumper de selección, el módulo nos da la opción de una alimentación de entre 12V a 35V DC. Por lo tanto, el regulador de 5 voltios no está funcionando, se debe conectar la bornera de +5V para alimentar la parte lógica del L298N, también se puede usar como fuente la salida de 5 voltios del Arduino.

3.1.4.4 Encoder FC-03

Con este módulo de sensor de velocidad IR con el comparador LM393, permite calcular velocidad de rotación de las ruedas del prototipo, colocando una corona dentada que gira unida a una rueda.

El funcionamiento básico de este sensor es el siguiente; Si algo ha pasado entre la ranura del sensor, crea un pulso digital en el husillo D0. Este impulso varía de 0V a 5 V y es una señal TTL digital. Así que se puede leer este impulso con Arduino.

Este sensor de codificador óptico utiliza el interruptor infrarrojo Opto MOCH22A. El dispositivo MOCH22A tiene dos partes: un transmisor infrarrojo y un receptor de IR o sensor. Entre el transmisor y el receptor IR es un espacio de espacio para máquinas tragamonedas. Los pulsos se escanean en un escaneado TTL Pulse Opamp-LM393 que puede ser interpretado por un microcontrolador como Arduino.

Se recomienda usar flancos altos / bajos para detectar pulsos. Se sugiere agregar un condensador de 100NF entre la línea de salida D0 y la Tierra como modo de filtro, para evitar el disparo falso en la interrupción. Una recomendación adicional es activar el módulo con 3,3 V.

También se utilizan en el contador de RPM (revoluciones por minuto) en motores CC / CA o como un sensor vocacional final.

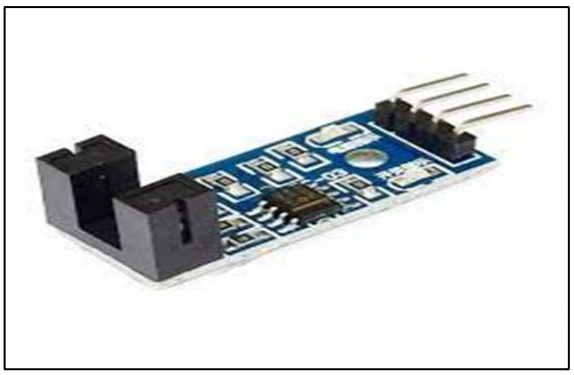


Figura 5-3: Encoder FC-03

Fuente: Sanango Andrés, 2022.

Tabla 4-3: Principales características encoder FC-03

Sensor:	MOCH22A
Voltaje de Operación:	3.3V - 5V DC
Salidas:	Analógica y Digital TTL
Modelo Placa:	FC-03 / FZ0888
Tipo de emisor:	Fotodiodo IR
Tipo de detector:	fototransistor
Longitud de onda del emisor:	950 nm (infrarrojo)
Peso:	8 gramos
Dimensiones:	3.2*1.4*0.7 cm
Ranura	5mm
Comparador Opamp:	LM393
Led indicador de alimentación	1
Led indicador de pulso	1
Salida TTL ON:	Sensor bloqueado
Salida TTL OFF:	Sensor sin bloquear

Realizado por: Sanango Andrés, 2022.

3.1.4.5 Vehículo a control remoto reciclado

En la figura 6-3, se muestra el vehículo a escala utilizado para la integración del sistema de detección de señales de tránsito, cabe recalcar que este modelo fue seleccionado tanto por ser un

juguete reciclado como por las dimensiones, las cuales son las adecuadas para poder colocar todo el sistema eléctrico dentro.

Figura 6-3: Porsche Cayenne turbo 1:24, Radio control de juguete

Fuente: (AMAZON, 2021b)

Tabla 5-3: Principales características técnicas del vehículo

Dimensiones del producto	39 x 12 x 10 cm; 400 gramos
Edad recomendada por el fabricante	6 años y más
Número de modelo	46100
Número de piezas	2
Montaje necesario	No
Escala	1:24
Necesita baterías	Sí
Incluye baterías	No
¿Mando a distancia incluido?	Sí
Tipo de mando a distancia	Funkfernsteuerung
Número de canales de radiocontrol	4
Número de canales	1
Peso del producto	400 g

Realizado por: Sanango Andrés, 2022.

3.1.5 Esquema de conexión del prototipo

Luego de identificar los diferentes componentes electrónicos que integran el prototipo, se presenta en el gráfico 2-3, el diagrama de conexión del módulo de adquisición, procesamiento y actuadores. Su elemento central es la Raspberry Pi 4 modelo B, la cual se interconecta con sus diferentes componentes como se indica a continuación:

- La cámara RaspiCam 1.3 está conectada mediante cable flexible AWM20798 de 15 pines al puerto CSI de la Raspberry Pi.
- La conexión del l298n y los demás dispositivos se muestra a continuación en Tabla 6-3

Tabla 6-3: Descripción de terminales y conexión

L298n	Motor	Fuente de alimentación L298n	Alimentación Raspberry	Raspberry Pi 2		Cámara
				# de pin	GPIO	
VCC		+5v		-	-	
GND		GND		20	GND	
IN1				16	23	
IN2				18	24	
ENA				22	25	
			USB	USB		
				MIPI CSI		MIPI CSI
OUT1	PIN1					
OUT2	PIN2					

Realizado por: Sanango Andrés, 2022.

3.1.6 Diseño electrónico

En el gráfico 2-3, se presenta el diagrama de bloques del circuito a implementar describiendo de forma rápida el proceso de conexión de cada elemento que se utiliza en el prototipo.

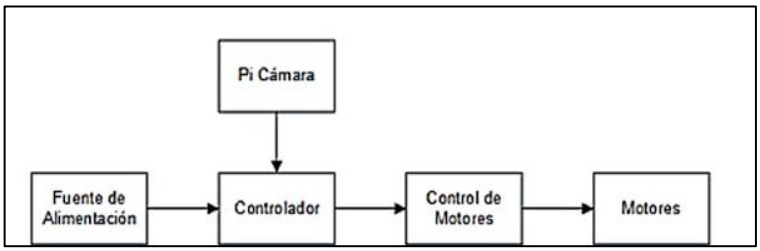


Gráfico 2-3: Diagrama de bloques del circuito

Realizado por: Sanango Andrés, 2022.

La figura 7-3, muestra una simulación más real del circuito diseñado cada uno de los pines y puertos de conexión fue especificado en la tabla 6-3.

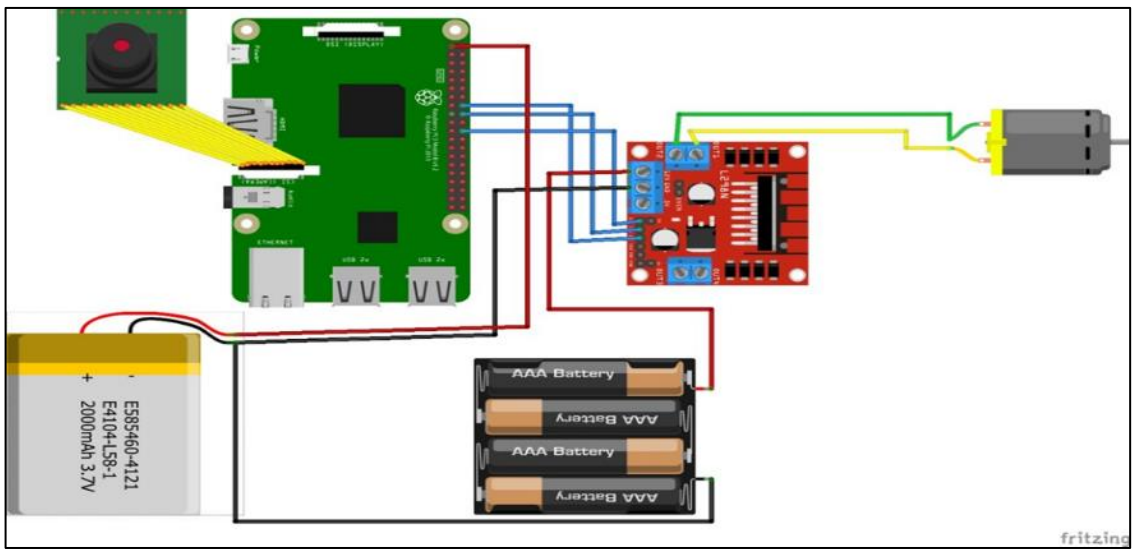


Figura 7-3: Diagrama del circuito

Fuente: Sanango Andrés, 2022.

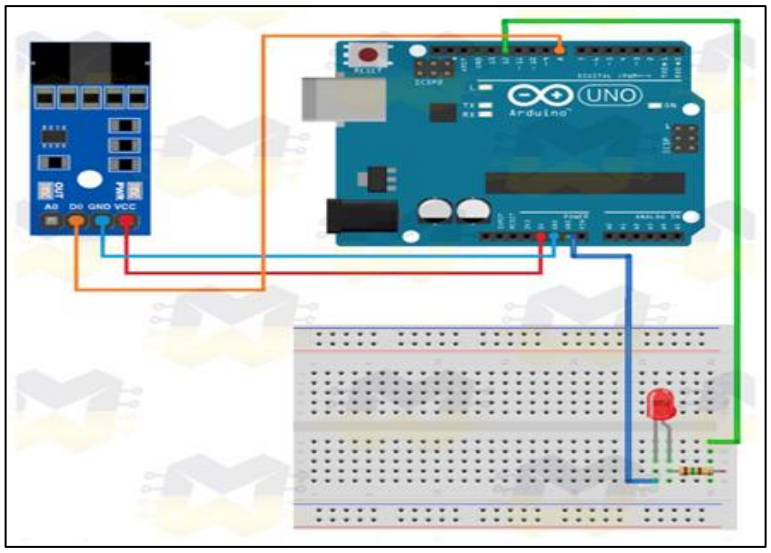


Figura 8-3: Circuito de recolección de datos de velocidad

Fuente: Sanango Andrés, 2022.

3.1.7 Prototipo ensamblado

Como se puede apreciar en figura 9-3, el circuito eléctrico y cada uno de los dispositivos se encuentran conectados de la forma especificada anteriormente, durante el ensamblaje se debe tener en cuenta las hojas de datos de cada uno de los dispositivos para evitar daños en algunos de ellos.

Todo esto es montado de la forma más adecuada para evitar roces del motor y sobrecalentamientos en la Raspberry, utilizando tornillos y pegamento para tener una mejor fijación.

En la figura 10-3, se puede apreciar el prototipo totalmente ensamblado y listo para trabajar, en el cual se coloca con un soporte impreso en 3D la cámara en la parte superior, por motivos de tener una mejor visualización del entorno para poder identificar las señales de tránsito.

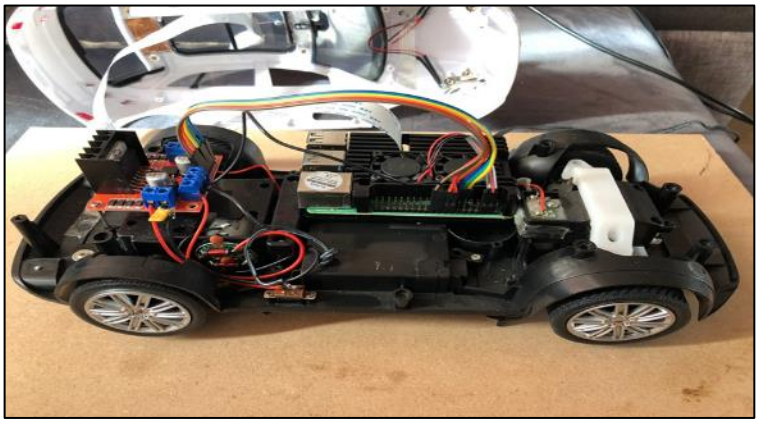


Figura 9-3: Circuito eléctrico ensamblado

Fuente: Sanango Andrés, 2022.

Figura 10-3: Prototipo ensamblado

Fuente: Sanango Andrés, 2022.

3.2 Software

Antes de iniciar con la selección del software, se debe tener claro el funcionamiento deseado del prototipo es por esto que es importante partir especificando los requerimientos del software.

3.2.1 Requerimientos del software

- Adquirir información de los fotogramas de la cámara uno y realizar los subprocesos de detección de la señal de tránsito.
- Visualizar los datos procesados por detección de las señales de tránsito en un monitor hasta realizar la programación y pruebas.
- Las señales de tránsito a reconocer son 5 (STOP,30 km/h, 50km/h, 90 km/h, 100 km/h).
- Al detectar una señal de transito se dibuja un cuadro alrededor de la misma.
- Al detectar las señales de tránsito establecidas se procede a variar la velocidad del vehículo.
- Tomar las medidas de cambio de velocidad a través de un programa Arduino extrayendo los datos de un sensor óptico.

3.2.2 Selección de software

Se indica el software escogido para el funcionamiento del prototipo.

3.2.2.1 Sistema operativo Raspbian

Raspbian es una distribución libre del sistema operativo GNU/Linux basado en Debian Wheezy (Debian 7.0) para la placa de desarrollo Raspberry Pi, su lanzamiento fue en junio de 2012. Los desarrolladores de esta iniciativa son los miembros de la Comunidad Raspbian, como

característica principal tiene un núcleo monolítico Linux con una interfaz gráfica por defecto LXDE, para plataformas de ARMv6, ARM es una arquitectura RISC (Reduced Instruction Set Computer = Ordenador con Conjunto Reducido de Instrucciones). A pesar que lleva ya muchos años lanzando sus versiones se puede decir que todavía es una plataforma en desarrollo que tiene soporte multilenguaje y cuenta con soporte técnico para todas sus versiones (RASPBIAN, 2020).

Raspbian es el sistema operativo recomendado para Raspberry Pi (optimizado para su hardware) y se basa en una distribución de GNU / Linux llamada Debian.

Para instalar Raspbian en nuestro PI de Raspberry, tenemos dos versiones; Completa con el entorno gráfico y una menor sin un ambiente gráfico:

- **Raspbian Pixel:** La versión completa con un entorno gráfico Raspbian, que es la versión de escritorio con menús, ventanas, símbolos, fondos, etc. utilizada por la mayoría de los usuarios como escritorio.
- **Raspbian Lite:** versión reducida sin entorno gráfico, que es, la versión de la versión de la consola sin gráficos. Esta opción suele estar diseñada para usuarios avanzados de Linux que utilizan la Raspberry Pi como servidor.

3.2.2.2 Python

Este es un lenguaje de scripting independiente de la plataforma independiente y orientada a objetos que está lista para ejecutar cualquier tipo de programa, desde aplicaciones de Windows hasta servidores de red o incluso páginas web. Es un lenguaje interpretado, es decir, es que el código fuente no está obligado a ejecutarlo. Ofrece ventajas, como la velocidad de desarrollo y una mejor velocidad. En los últimos años, el idioma se ha vuelto muy popular gracias a muchas razones. Por ejemplo, la cantidad de bibliotecas que usan el lenguaje en sí mismo integraron los tipos de datos y las características que ayudan a crear muchas tareas compartidas sin tener que programarlas desde el principio. Simplicidad y velocidad con cuyos programas se crean. Un programa de Python puede tener 3 a 5 líneas de código más bajo a su equivalente en Java o C. Tener la cantidad de plataformas donde podemos desarrollar, como Unix, Windows, OS / 2, Mac, amigo y otros. Además, Python es gratuito, también para fines comerciales (PYTHON, 2020).

Para mejorar la interfaz gráfica de este proyecto, se utilizó Pygame, que es un conjunto de módulos de voz de Python para desarrollar videojuegos, incluidas las bibliotecas gráficas y de audio para usar con el idioma. En cuanto a los prototipos, permiten prototipos y los resultados pueden convertirse en profesionales (PYGAME, 2020).

Python es un lenguaje de programación interpretado cuya filosofía subraya la legibilidad de su código. Este es un lenguaje de programación con múltiples fotogramas, ya que admite consultoría de objetos en la programación convincente. En menor medida, la programación funcional. Es un lenguaje de plataforma interpretado, dinámico y múltiple. Es un lenguaje de programación orientado a objetos e interpretado, es decir, el código distrae la línea de un intérprete. Además, es un lenguaje indexado que lo hace muy legible del usuario. Este idioma fue creado por Guido Van Rossum a finales de los años ochenta. Para la finalización de esta versión de Labor 3.9.0 de este lenguaje de programación se ha utilizado

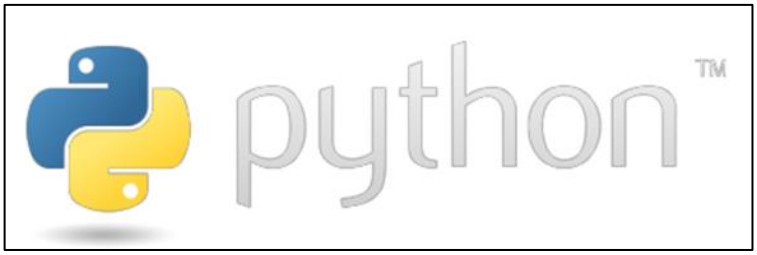


Figura 11-3: Python

Fuente: (PYTHON, 2020)

3.2.2.3 Open CV

Es una biblioteca de código abierto para la modificación de la imagen C / C y la visión calculada originalmente desarrollada por Intel. La primera versión estable se publicó en 2006. En octubre de 2009, se publicó el segundo inicio más grande: Opencv2 (WILLOGAGE, 2020).

Disponible bajo Linux, Mac y Windows. Tiene estructuras de datos básicas para ensayos con matrices y procesamiento de imágenes. Se le permite ver los datos de una manera muy simple y extraer información a partir de imágenes y vídeos. Tiene la adquisición de imágenes y presentación cuenta con más de 500 características que cubren una amplia gama de sectores en el proceso de la visión, tales como el reconocimiento de objetos, calibración de la cámara, la visión estereoscópica y la visión. Robot. Los métodos utilizados en OpenCV son: el pelo que nos puede hacer frente a la cara y las líneas de transferencia para vehículos y líneas de ferrocarril.

Abrir el ordenador, visión, detección se trata de un programa de empleo bibliotecas estadounidenses permiten hacer todo el proceso que hemos visto en un ordenador antes:

- La imagen de la adquisición a la extracción de información.

Las aplicaciones de software más libres basadas en una visión artificial basado en OpenCV. Estas bibliotecas Intel desarrolló como una expansión de las bibliotecas IPL, a partir de 1999 como un proyecto de código abierto, y publicadas en el 2007 la versión 1.0. Estas bibliotecas están en la base de algoritmos y funciones que finalmente aplicamos a nuestras aplicaciones.

3.2.2.4 Haar cascade

El clasificador en cascada es el reconocido método de Viola y Jones para la detección de rostros pues ahí tuvo sus inicios, no obstante, este método no se limita solo a la detección de rostros, sino que también es flexible a detectar gran variedad de objetos. El clasificador en cascada es un algoritmo basado en árboles, en el que Viola y Jones plasmaron las características por defecto se muestran en la figura 12-3, y se puede utilizar a todas las escalas incluso con un clasificador reforzado para calcular rápidamente cualquier objeto en una imagen de entrada.

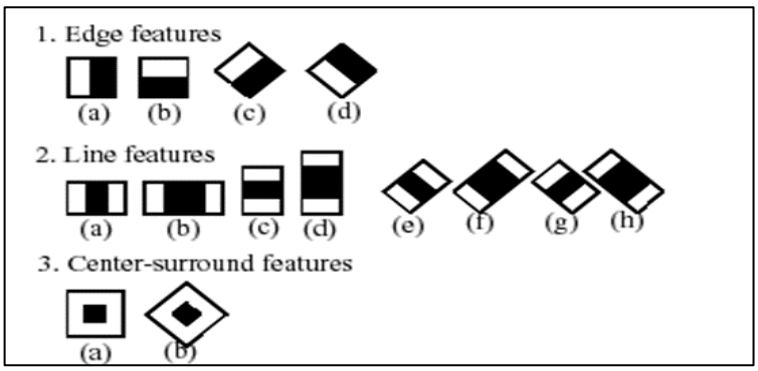


Figura 12-3: Extracción de características Haar

Fuente: Sanango Andrés, 2022.

El sistema detecta los objetos en cuestión moviendo una ventana sobre la imagen. Cada etapa del clasificador etiqueta la región específica definida por la ubicación actual de la ventana como positiva o negativa, lo que significa positivo que se encontró un objeto o negativo significa que el objeto especificado no se encontró en la imagen. Si el etiquetado da un resultado negativo, entonces la clasificación de esta región específica se completa por la presente y la ubicación de la ventana se mueve a la siguiente ubicación. Si el etiquetado da un resultado positivo, la región pasa a la siguiente etapa de clasificación como se puede ver en la figura 13-3.

El clasificador da un veredicto positivo, cuando todas las etapas, incluida la última, dan un resultado, diciendo que el objeto se encuentra en la imagen.

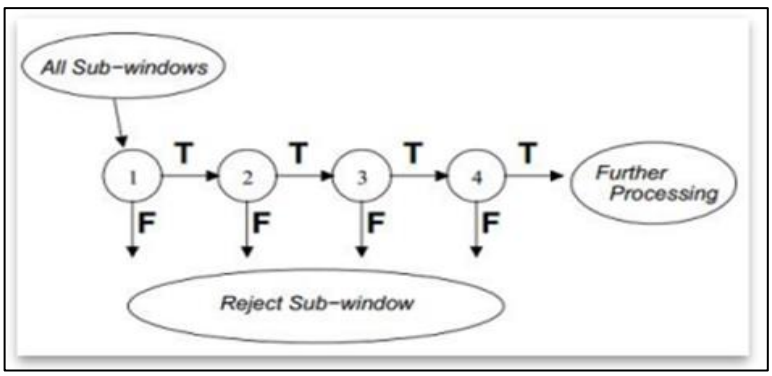


Figura 13-3: Rechazo en cascada

Fuente: Sanango Andrés, 2022.

3.2.2.5 Arquitectura del Haar cascade

Un clasificador en cascada se refiere a la concatenación de varios clasificadores dispuestos en orden sucesivo. Toma una gran cantidad de pequeñas decisiones sobre si es el objeto o no. La estructura del clasificador en cascada es de un árbol de decisión degenerado.

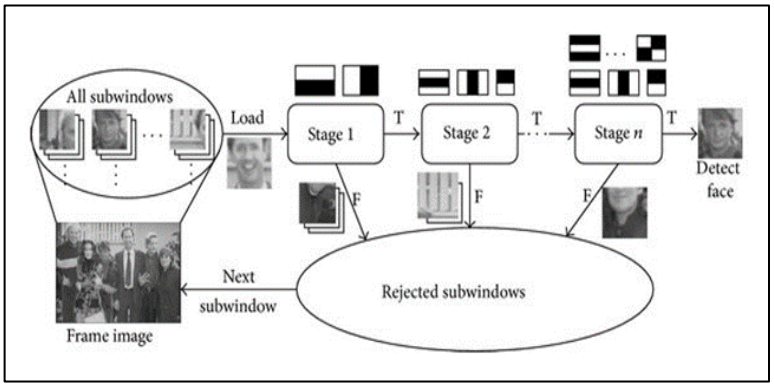


Figura 14-3: Arquitectura del clasificador en cascada

Fuente: Sanango Andrés, 2022.

3.2.2.6 Parámetros de entrada Haar cascade

Una característica del método Haar es que está definido por parámetros como la distribución, el tamaño, la orientación, regiones positivas y negativas, lo que permite la construcción de infinitos tipos. Estos parámetros, dependerán de las características del objeto a detectar, y más concretamente de la distribución de intensidad de los píxeles que componen dichas características. Por lo tanto, el objetivo al crear características de Haar es encontrar estructuras cuya estructura sea parecida a la característica que tenemos que detectar, como bordes, líneas o contornos, como en el reconocimiento de matrículas.

3.2.3 Adaptación del algoritmo Haar

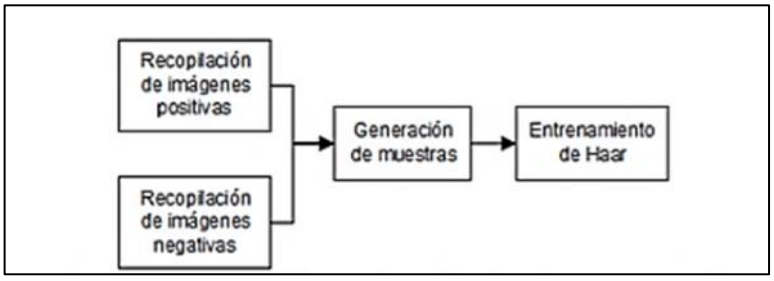


Gráfico 3-3: Diagrama de flujo para la creación del algoritmo Haar

Fuente: Sanango Andrés, 2022.

3.2.3.1 Preparando los datos de entrenamiento

Para realizar el entrenamiento es necesario contar con gran cantidad de imágenes en donde esté presente cada señal que se desea detectar (conjunto de muestras positivas) y otro conjunto de imágenes donde NO esté presente dicho objeto (conjunto de muestras negativas). Estas imágenes no deben ser muy grandes, ya que esto podría provocar lentitud en la detección por lo que se les estableció en 24X24 pixeles.

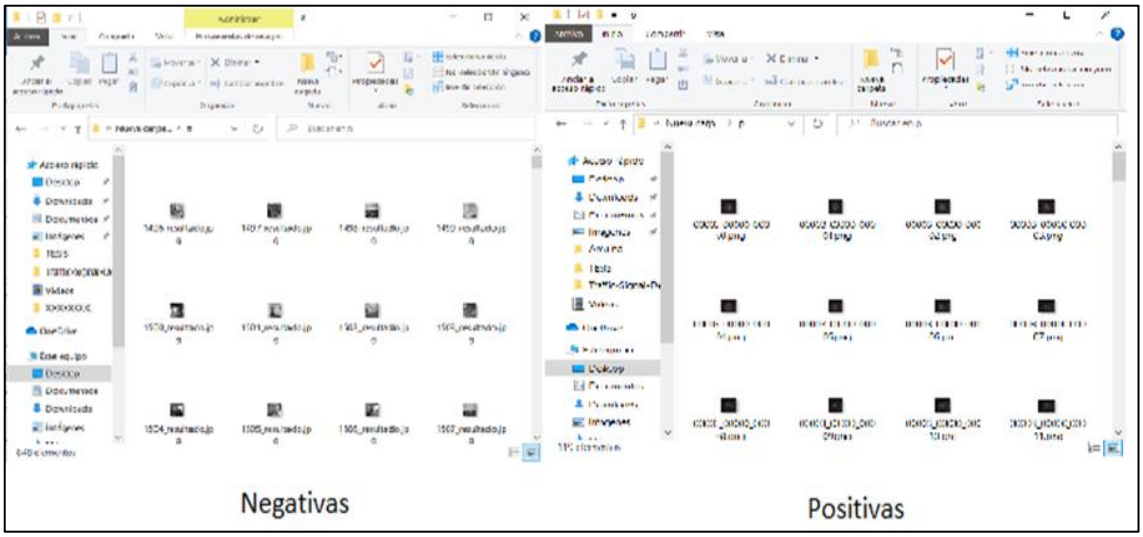


Figura 15-3: Conjuntos de imágenes positivas y negativas para el entrenamiento

Fuente: Sanango Andrés, 2022.

Para realizar el entrenamiento en Cascade Trainer GUI, es necesario tener las imágenes positivas y negativas almacenadas en dos carpetas ‘p’ y ‘n’ respectivamente como se muestra en el siguiente apartado.

3.2.3.2 Entrenamiento del clasificador en cascada

Una vez que se tiene las imágenes listas para el entrenamiento, se dirige al programa Cascade Trainer GUI, en el que se tiene la siguiente ventana:

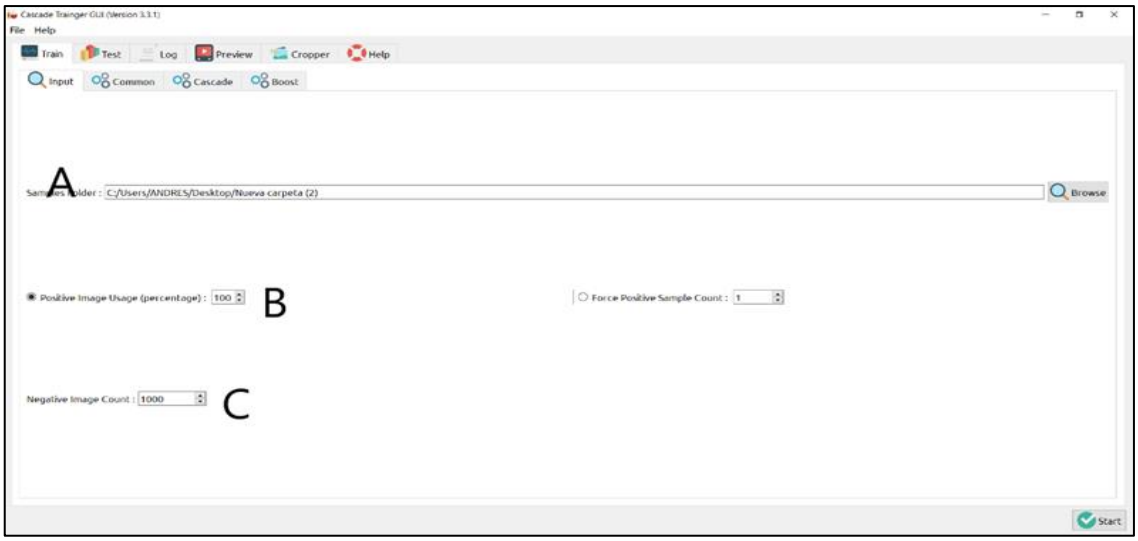


Figura 16-3: Cascade trainer

Fuente: Sanango Andrés, 2022.

- **A:** En esta sección se debe seleccionar la carpeta en donde están contenidas las carpetas p y n.
- **B:** Aquí es necesario especificar el porcentaje de imágenes positivas a usar, en este caso se estableció 600.
- **C:**Se debe especificar la cantidad de imágenes negativas a usar para el entrenamiento, en este caso 890 imágenes.

Luego se procede al aparatado Common.

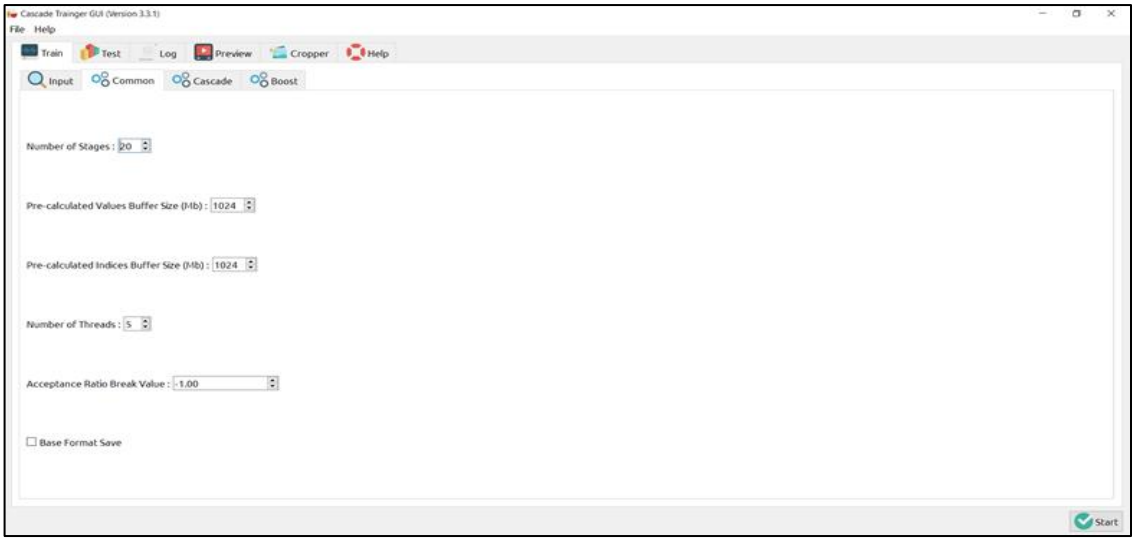


Figura 17-3: Cascade Trainer Common

Fuente: Sanango Andrés, 2022.

En este apartado no se modifica ningún valor, por lo que el número de etapas queda en 18-3.

Figura 18-3: Cascade Trainer apartado

Fuente: Sanango Andrés, 2022.

- **A:** Se ingresa el ancho que poseen las imágenes de entrenamiento, en este caso 24 pixeles.
- **B:** Se ingresa el alto igualmente, de las imágenes de entrenamiento, en este caso 24 pixeles.
- **C:** Finalmente, clic en Start y se espera a que se realice el entrenamiento. En este caso se tardó aproximadamente 4 horas.

3.2.4 Integración del entrenamiento a la Raspberry

Para poder ejecutar el clasificador entrenado en la Raspberry es necesario tener instalados en la misma Python y OpenCV con sus librerías, por motivos de eficiencia el entrenamiento es realizado en un PC de alto rendimiento ya que son mucho mas robustas que una Raspberry PI 4.

El método para ejecutar el clasificador es muy similar al que se realizaría en un PC cualquiera con Windows, ya que la Raspberry posee una interfaz gráfica del sistema operativo Raspbian.

Pasos por seguir:

- Iniciar la Raspberry PI 4
- Instalar Python
- Instalar OpenCV
- Ejecutar Python
- Cargar el programa generado en Python
- Vincular el entrenamiento con el programa diseñado
- Ejecutar el programa

3.2.5 Programación principal del prototipo

El gráfico 4-3, muestra el diagrama de flujo del programa principal del prototipo, el cual se describe a continuación.

3.2.5.1 Para la inicialización

- Declaración de las bibliotecas que se va utilizar.
- Se configura el hardware con los parámetros iniciales de uso, en este se declaran los dispositivos de entrada y salida.
- Se declaran e inicializan las variables.
- Se cargan las funciones necesarias para tener un correcto funcionamiento.
- Se cargan los entrenamientos de las 5 señales (archivos XML).

3.2.5.2 Para el lazo que se repite indefinidamente

- Se procede a la detección de la cámara y el control sobre el L298n, cada que se detecte una señal de transito que coincida con los entrenamientos cargados.

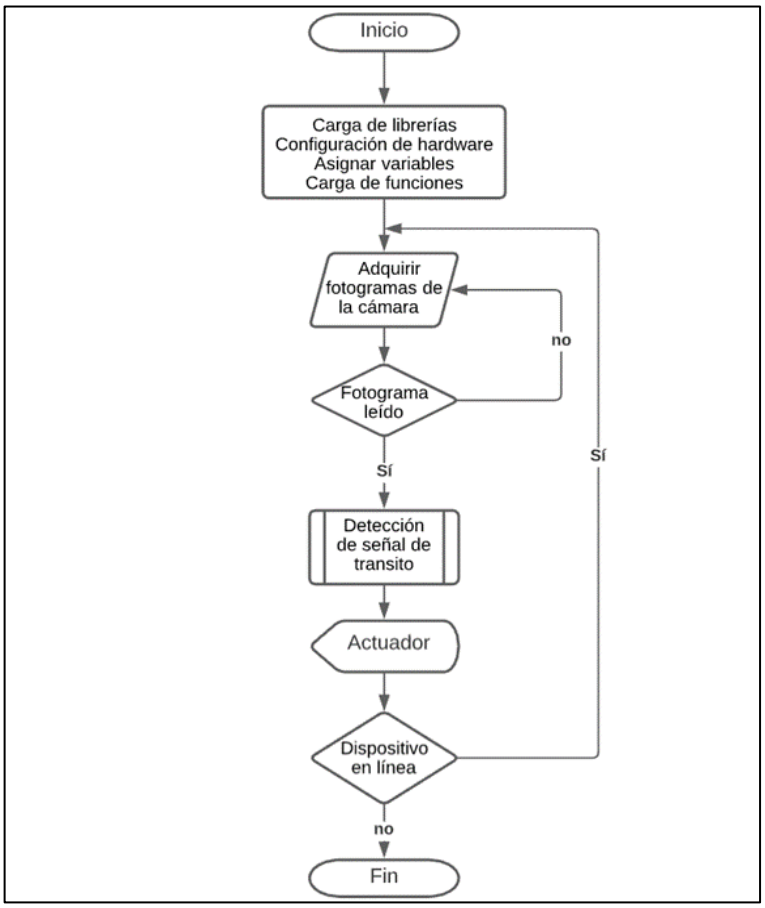


Gráfico 4-3: Diagrama de flujo del *software* para el prototipo

Fuente: Sanango Andrés, 2022.

El código del programa principal de prototipo se muestra en el anexo B.

```
import cv2
import numpy as np
import RPi.GPIO as GPIO
from time import sleep

in1 = 24
in2 = 23
en = 25
temp1=1


GPIO.setmode(GPIO.BCM)
GPIO.setup(in1,GPIO.OUT)
GPIO.setup(in2,GPIO.OUT)
GPIO.setup(en,GPIO.OUT)
GPIO.output(in1,GPIO.LOW)
GPIO.output(in2,GPIO.LOW)
p=GPIO.PWM(en,1000)
p.start(25)

SS_cascade = cv2.CascadeClassifier('stopsign_classifier.xml')
SpS_cascade = cv2.CascadeClassifier('lbpCascade.xml')
SpS1_cascade = cv2.CascadeClassifier('lbpCascade1.xml')
SpS2_cascade = cv2.CascadeClassifier('lbpCascade2.xml')
SpS3_cascade = cv2.CascadeClassifier('lbpCascade3.xml')

cap = cv2.VideoCapture(0)

#reducir la resolución para aumentar el FPS
cap.set(3,320)
cap.set(4,240)

i = 0
j = 0
k = 0
l = 0
m = 0

while True:
    ret, img = cap.read()
    ret, frame = cap.read();
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    GPIO.output(in1,GPIO.HIGH)
    GPIO.output(in2,GPIO.LOW)
    SSs = SS_cascade.detectMultiScale(gray, 1.3, 5)
    for(x,y,w,h) in SSs:
```

Figura 19-3: Parte de la programación del prototipo

Fuente: Sanango Andrés, 2022.

3.2.6 Calibración del detector

A continuación, se presenta el cómo fueron elegidos cada uno de los factores del algoritmo.

3.2.6.1 Manipulando ScaleFactor

Con el código que se ha elaborado se procede a hacer pruebas cambiando los valores de ScaleFactor en: 1.1, 1.3, 1.5, 5 y 1.01. Claro que los resultados estarán directamente relacionados también a los otros argumentos sin embargo esto sirve para darse cuenta cómo actúa este argumento.

Figura 20-3: Manipulando el argumento ScaleFactor

Fuente: Sanango Andrés, 2022.

Analizando la imagen del lado izquierda de la figura 20-3, en ella se aprecia que todas las señales que aparecen en la imagen han sido detectadas.

En la imagen derecha de la figura 20-3, al usar un ScaleFactor = 5 se ha detectado solo una señal de tránsito.

Conclusión: En esta sección solo se modificado el argumento ScaleFactor para entender mejor su funcionamiento. Cuando se utiliza 1.1 en esta imagen (se puede buscar el mejor valor para otra aplicación dependiendo del proyecto) se detecta correctamente las 5 señales, pero conforme aumenta el valor a 1.3 o 1.5 se observa que las señales se fueron perdiendo. Por otro lado, al usar 1.01 se obtienen muchísimos falsos positivos que no sirven, con lo cual se determina que el mejor valor es 1.3.

3.2.6.2 Manipulando MinNeighbors

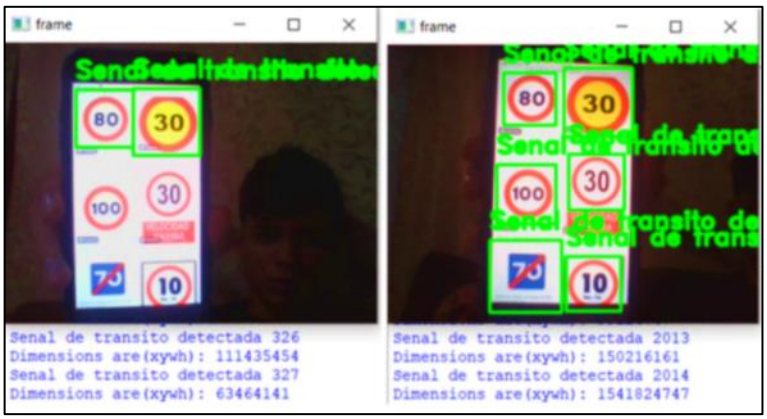


Figura 21-3: Manipulando el argumento MinNeghbors

Fuente: Sanango Andrés, 2022.

Analizando la imagen de la derecha de la figura 21-3, con MinNeighbors = 20 se tiene muchísimas más detecciones (falsos positivos) que las cuatro que deberían aparecer.

La imagen superior izquierda de la figura 21-3, con MinNeighbors = 1 se ha conseguido detectar 2 señales de los 5 que aparecen en ella.

- **Conclusión:** Este parámetro especifica el número mínimo de cuadros delimitadores o vecinos, que debe tener la señal para que se detecte como tal. Entonces se puede ver que si se le da un valor de 1 es más propenso a aceptar señales en lugares donde no los hay, mientas que, si incrementamos ese valor como a 20 por ejemplo, las detecciones descienden. Por ello se debes experimentar también con este parámetro hasta que encontrar el valor que se adapta a las necesidades.

Para este caso se seleccionó un MinNeighbors = 4 con el cual se tuvo un 100% de aciertos en la detección.

CAPÍTULO IV

4. RESULTADOS

El actual capítulo se presentan las pruebas que se realiza al prototipo con la finalidad de verificar el funcionamiento de la detención de señales de tránsito stop y velocidad, además del control de la velocidad sobre las ruedas del vehículo.

4.1 Variables

Durante el desarrollo del proyecto se determinó que las variables a ser consideradas para la extracción de datos son las siguientes:

- Numero de detecciones correctas.
- Tiempos de retardo (en los cambios de velocidad)
- Velocidad en las ruedas

4.2 Protocolo de pruebas

Las pruebas se realizaron en tres etapas:

- Mañana
- Tarde
- Noche

Durante estas etapas se realizaron pruebas de detección con 134 imágenes de diferentes señales de tránsito.

- Considerando la cantidad de luz proporcionada por el ambiente para la toma de datos.
- Presentar al prototipo las señales de tránsito.
- Verificar si se realiza la correcta detección de la señal de tránsito.
- Verificar el número de señales de tránsito detectadas por el prototipo.
- Tomar los tiempos de retardo que se tiene desde la detección hasta aplicar la orden de cambio de velocidad.
- Medir la velocidad en las ruedas.
- Verificar si la velocidad es acorde con la señal detectada.

4.3 Pruebas de detección

En la primera etapa, concluido el entrenamiento se procedió a probar el modelo para verificar su correcto funcionamiento.

Como se observa en figura 1-4, se realizó una prueba del funcionamiento del programa utilizando un entrenamiento Haar, obteniendo un resultado del 100% de detección de las señales entrenadas.

Figura 1-4: Prueba de funcionamiento del algoritmo y entrenamiento

Fuente: Sanango Andrés, 2022.

Figura 2-4: Escenario 1 8 a.m. – 11 a.m.

Fuente: Sanango Andrés, 2022.

Una vez ejecutado el reconocimiento de las señales presentadas en el escenario 1, se obtuvieron los resultados mostrados en tabla 1-4, tabla 4-4.

Tabla 1-4: Porcentaje de aciertos de reconocimiento de señales de tránsito en el escenario 1

Tipo de señal	Número de señales	Número de señales detectadas	Porcentaje
STOP	10	10	100%
10 km/h	10	10	100%
20 km/h	10	10	100%
30 km/h	10	10	100%
40 km/h	10	10	100%
50 km/h	10	10	100%
60 km/h	10	10	100%
70 km/h	10	10	100%
80 km/h	10	10	100%
90 km/h	10	10	100%
100 km/h	10	10	100%
120 km/h	10	10	100%
Reductor de velocidad	2	0	0%
Curva izquierda	2	0	0%
Curva derecha	2	0	0%
Peatones en la vía	2	0	0%
Aproximación a redondel	2	0	0%
Cruze de vías	2	0	0%
Niños	2	0	0%

Realizado por: Sanango Andrés, 2022.

Figura 3-4: Escenario 2 12 p.m. – 4 p.m.

Fuente: Sanango Andrés, 2022.

Una vez ejecutado el reconocimiento de las señales presentadas en el escenario 1, se obtuvieron los resultados mostrados en tabla 2-4, tabla 5-4.

Tabla 2-4: Porcentaje de aciertos de reconocimiento de señales de tránsito en el escenario 2

Tipo de señal	Número de señales	Número de señales detectadas	Porcentaje
STOP	10	10	100%
10 km/h	10	10	100%
20 km/h	10	10	100%
30 km/h	10	10	100%
40 km/h	10	10	100%
50 km/h	10	10	100%
60 km/h	10	10	100%
70 km/h	10	10	100%
80 km/h	10	10	100%
90 km/h	10	10	100%
100 km/h	10	10	100%
120 km/h	10	10	100%
Reductor de velocidad	2	0	0%
Curva izquierda	2	0	0%
Curva derecha	2	0	0%
Peatones en la vía	2	0	0%
Aproximación a redondel	2	0	0%
Cruze de vías	2	0	0%
Niños	2	0	0%

Realizado por: Sanango Andrés, 2022.

Figura 4-4: Escenario 3 6 p.m. – 9 p.m.

Fuente: Sanango Andrés, 2022.

Una vez ejecutado el reconocimiento de las señales presentadas en el escenario 3, se obtuvieron los resultados mostrados en tabla 3-4, tabla 6-4.

Tabla 3-4: Porcentaje de aciertos de reconocimiento de señales de tránsito en el escenario 3

Tipo de señal	Número de señales	Número de señales detectadas	Porcentaje
STOP	10	10	100%
10 km/h	10	10	100%
20 km/h	10	10	100%
30 km/h	10	10	100%
40 km/h	10	9	90%
50 km/h	10	10	100%
60 km/h	10	10	100%
70 km/h	10	10	100%
80 km/h	10	9	90%
90 km/h	10	10	100%
100 km/h	10	9	100%
120 km/h	10	9	90%
Reductor de velocidad	2	0	0%
Curva izquierda	2	0	0%
Curva derecha	2	0	0%
Peatones en la vía	2	0	0%
Aproximación a redondel	2	0	0%
Cruze de vías	2	0	0%
Niños	2	0	0%

Realizado por: Sanango Andrés, 2022.

4.4 Pruebas de velocidad y tiempos de ejecución

Las pruebas fueron realizadas de manera experimental, para lo cual se instaló un sensor óptico en una de las ruedas del vehículo y se tomó cada una de las medidas de velocidad con respecto a la detección de las señales.

Tabla 4-4: Velocidad medida y tiempos de ejecución en el escenario 1

Tipo de señal	Número de señales	Velocidad promedio Km/h	Tiempo de retardo promedio en (S)
STOP	10	0	0.7
50 km/h	10	49	0.6
90 km/h	10	89	0.5
100 km/h	10	98	0.7

Realizado por: Sanango Andrés, 2022.

Tabla 5-4: Velocidad medida y tiempos de ejecución en el escenario 2

Tipo de señal	Número de señales	Velocidad promedio Km/h	Tiempo de retardo promedio en (S)
STOP	10	0	0.7
50 km/h	10	50	0.5
90 km/h	10	90	0.6
100 km/h	10	99	0.6

Realizado por: Sanango Andrés, 2022.

Tabla 6-4: Velocidad medida y tiempos de ejecución en el escenario 3

Tipo de señal	Número de señales	Velocidad promedio Km/h	Tiempo de retardo promedio en (S)
STOP	10	0	0.7
50 km/h	10	50	0.6
90 km/h	10	89	0.7
100 km/h	10	102	0.6

Realizado por: Sanango Andrés, 2022.

4.5 Resultados y discusión

4.5.1 Resultados de la detección de señales de tránsito

En el vehículo se realizaron las pruebas de velocidad y tiempos de retardo en un laxo de 10 horas de marera intermitente, donde se presentó a nuestro dispositivo un conjunto de 134 señales de tránsito entre las cuales de encontraban: Curva izquierda, Curva derecha, Peatones en la vía, Aproximación a redondel, Reductor de velocidad, Cruze de vías, Niños, señales de velocidad (10 20 30 40 50 60 70 80 90 100 120) km/h y STOP.

Estas pruebas fueron realizadas en tres escenarios, mañana, tarde y noche, tomando en cuenta las horas del día según su iluminación natural.

Una buena iluminación permite a al prototipo visualizar de mejor manera las señales de tránsito y por lo tanto una buena detección de estas, caso contrario el reconocimiento en ineficiente.

Tabla 7-4: Pruebas de muestras emparejadas

Estadísticas de grupo				
	Horario	**Número**	**Media**	**error**
Señales reconocidas	Mañana	40	100%	0%
	Tarde	40	100%	0%
	Noche	39	97.5%	2.5%

Realizado por: Sanango Andrés, 2022.

Calculando el valor promedio de las medias obtenidas en los tres horarios antes establecidos se pudo afirmar que el prototipo es eficaz en un 99.16%

Además, se puede concluir que, al analizar las tablas presentadas anteriormente, una correcta iluminación le permite al prototipo identificar las señales de mejor manera, caso contrario se dificulta el reconocimiento en un mínimo porcentaje debido a que se a utilizando una cámara de visión nocturna.

4.6 Discusión

Los resultados inmersos en este estudio en cuanto al reconocimiento de señales de tránsito velocidad y STOP, se obtuvo un porcentaje de aciertos en el día de un 100% y en la noche de un 97.5% esto significa que el dispositivo, al tener la presencia de luz natural responde con un porcentaje alto al reconocimiento de señales y de igual manera al encontrarse con luz artificial se obtiene un porcentaje alto atribuyendo esto al tipo de cámara utilizado.

4.7 Análisis de costos

4.7.1 Presupuesto

En este apartado se detallan los valores financieros necesarios para la implementación del sistema inteligente; los costos pueden variar al cambiar los modelos y/o marcas del hardware, así como el tipo de licencias de software utilizados en el desarrollo.

4.7.1.1 Hardware

La tabla 8-4, muestra el *hardware* del sistema inteligente y su precio.

Tabla 8-4: Precio del *hardware*

Hardware	Cantidad	Precio
Ordinator Dell core i7, 2GHz, 8G RAM.	1	$0.00
Cámara Raspberry Pi3 5mp Nocturna + infrarrojos	1	$40,00
Carro a control remoto	1	$30,00
Kit Raspberry Pi4 4gb 32g + batería	1	$160,00
Total		$230,00

Realizado por: Sanango Andrés, 2022.

4.7.1.2 Software

La tabla 9-4, muestra el *software* del sistema inteligente y su precio.

Tabla 9-4: Precio del *software*

Software	Cantidad	Precio
Sistema Operativo Microsoft Windows 10	1	$0,00
Sistema Operativo Raspbian	1	$0,00
Python 3.9.0, Rcnn, YoloV2 y Yolov3, open CV	1	$0,00
Microsoft Office 2013 ESPOCH	1	$0,00
Total		$0,00

Realizado por: Sanango Andrés, 2022.

4.7.1.3 Herramientas

La tabla 10-4, muestra el hardware del sistema inteligente y su precio.

Tabla 10-4: Precio de las herramientas

Herramientas	Cantidad	Precio total
Kit de herramientas	1	$30,00
Artículos de oficina	1	$10,00
Resma de papel	2	$10,00
Extras	8	$10,00
Total		$60,00

Realizado por: Sanango Andrés, 2022.

4.7.1.4 Presupuesto general

Sumando la contribución de todas las categorías anteriores, se obtiene que el coste total del proyecto; impuestos incluidos, asciende al valor estipulado en la tabla 11-4.

Tabla 11-4: Precio de las herramientas

Concepto	Costo subtotal
Hardware	$230,00
Software	$0,00
Herramientas	$0,00
Subtotal	$230,00
Imprevistos (12%)	$27,60
Total	$257,60

Realizado por: Sanango Andrés, 2022.

4.7.1.5 Fuente de financiamiento

Todo este proyecto fue autofinanciado ya que se disponen de la mayoría de materiales lo cual redujo el costo en casi un 70% debido a que la mano de obra es proporcionada por el ejecutor del proyecto.

4.7.1.6 Comparativa

Con respecto Al dispositivo “Prototipo detector de tipos de señales de tránsito (Informativas, reglamentarias y preventivas)” sin tener efecto sobre el vehículo, este a llegando a tener un precio de $450. Por lo tanto, se determinó que el prototipo que se presenta es más económico con un precio de $257,60, presentando mejores prestaciones como lo son reconocimiento de las velocidades, mejor procesamiento, dispositivo Raspberry de última generación y control de la velocidad.

Con una diferencia de costo de $192,40.

CONCLUSIONES

- Mediante la utilización del software Cascade GUI se reduce en tiempo de entrenamiento del detector y se tiene una mejor eficiencia al utilizarlo en dispositivos con poca capacidad de procesamiento.

- Según la investigación realizada se logró determinar que las características del sistema inteligente de reconocimiento de señales de tránsito con visión artificial son, capacidad en procesamiento de imágenes, entrenamiento de la red, calidad de imagen, las cuales son primordiales para la elección de *hardware* y *software.*

- Mediante la utilización de dispositivos adecuados como lo son, driver, encoder, cámara y microcomputador, es posible adaptar un sistema de visión artificial en un vehículo a escala, logrando así la detección y reconocimiento de señales de tránsito, además de tener control sobre la velocidad del mismo, con lo cual se comprueba que es posible reconocer las señales de tránsito (STOP y velocidad) con una alta eficiencia en un vehículo a escala.

- Mediante revisión bibliográfica se logró establecer que el método adecuado para el control de la velocidad del vehículo autónomo es utilizar un driver l298n y encoder FC-03, permitiendo así conocer a velocidad y generar el PWM necesario para mantener la misma.

- Según los resultados encontrados se precisa que las variables que limitan el correcto funcionamiento del prototipo y reducen su eficacia son, encontrarse en un ambiente con poca iluminación y la capacidad del dispositivo raspberry, por ende, el procesamiento de imágenes se limita en un porcentaje mínimo afectando el reconocimiento de las señales.

- Se logró elaborar protocolo de pruebas sobre el sistema analizando parámetros que definen el funcionamiento del prototipo. En él se abordaron los principales aspectos, iluminación de la escena, velocidad del vehículo y el error llegando a determinar así la eficiencia del mismo.

- Según el análisis de resultados se puede demostrar que el prototipo propuesto tiene una precisión en la detección de señales del 99,16% valor que podría mejorar si se utiliza un acelerador Google coral USB y una cámara de mejores prestaciones.

RECOMENDACIONES

- La calibración de la cámara es un factor importante, esta debe estar enfocada correctamente, para visualizar perfectamente la escena y dado que por las noches el proceso de detección disminuye su eficiencia se debe mejorar la iluminación para así poder mejorar la precisión de 99,16% y que tienda al 100%.

- Se recomienda la utilización de una cámara con buenas características (visión infrarroja, foco ajustable), debido a que son necesarias imágenes en buena definición, para el procesamiento de imágenes.

- Utilizar Raspberry pi 4B de 8GB, debido su eficiencia en procesamiento y en el caso de necesitar más eficiencia y capacidad de procesamiento se recomienda utilizar aceleradores diseñados para visión artificial como lo es el dispositivo Google CORAL USB.

- Al adaptar el vehículo a escala se sugiere utilizar dispositivos que sean de dimensiones pequeñas y que puedan caber dentro del mismo.

- Se debe tener en cuenta la clase de motores que se va a controlar antes de elegir el dispositivo.

- Utilizar más de 500 imágenes en el entrenamiento para obtener un detector mucho más preciso.

- Se recomienda que la velocidad con la que se desplaza la señal de tránsito hacia el vehículo debe ser moderada debido a que la cámara tendrá problemas en captar la señal y esto afectaría en el reconocimiento de la misma.

BIBLIOGRAFÍA

AMAZON. Cámara Web y sus características. *Amazon.* [En línea] 2021a. [Citado el: 10 de Septiembre de 2021.] https://www.amazon.com/-/es/panorámica-privacidad-computadoras-portátiles-Conferencia/dp/B088TNCM6L.

AMAZON. Radio control de juguete del carro Porsche Cayenne. *Amazon.* [En línea] 2021b. [Citado el: 13 de Septiembre de 2021.] https://www.amazon.es/RASTAR-massG-46100-Porsche-Cayenne/dp/B007KLHRH6.

CHANAMPE, Hugo; et al. Modelo de redes neuronales convolucionales profundas para la clasificación de lesiones en ecografías mamarias. *Sedici.* [En línea] 2019. [Citado el: 09 de Septiembre de 2021.] http://sedici.unlp.edu.ar/handle/10915/77381. 978-987-3984-85-3.

CUEVAS JIMENEZ, Valdemar; & ZALDIVAR NAVARRO, Daniel. Visión por computador utilizando MATLAB y el Toolbox de procesamiento digital de imágenes. *Academia.* [En línea] 2007. [Citado el: 04 de Septiembre de 2021.] https://www.academia.edu/7448239/Visi%C3%B3n_por_Computador_utilizando_MatLAB_Y_el_Toolbox_de_Procesamiento_Digital_de_Im%C3%A1genes.

FLOREZ, Raquel; & FERNÁNDEZ, José. *Las redes neuronales artificiales fundamentos teóricos y aplicaciones prácticas.* España : Oleiros, 2008. 978-84-9745-246-5.

FOTOGRAFÍA Y TÉCNICA. La luz, materia prima para la fotografía. *Fotpgrafía y técnica.* [En línea] 2015. [Citado el: 10 de Septiembre de 2021.] http://fotografiaytecnica.blogspot.com/2015/09/la-luz-materia-prima-para-la-fotografia.html.

GÓMEZ, Pilar. El reconocimiento de patrones y su aplicación a las señales digitales. *Amexcomp.* [En línea] 2018. [Citado el: 10 de Septiembre de 2021.] http://amexcomp.mx/files/ReconocimientoPatronesAppSenalesDigitales.pdf.

GUTIÉRREZ MORENO, Jorge. Implementación de vehículo autónomo en entorno simulado. *Universidad Carlos III de Madrid.* [En línea] 2009. [Citado el: 05 de Septiembre de 2021.] https://core.ac.uk/download/pdf/288499538.pdf.

HAYKIN, Simón. *Neuronal Networks.* Canada : PEARSON, 1999. 81-7808-300-0.

LÓPEZ MÁRQUEZ, Gabriel Andrés; & ROBALINO PEÑA, Edgar Freddy. Sistema inteligente de reconocimiento de patrones con visión artificial para la alerta automática de intrusos

en las áreas de almacenamiento de las PYMES. *Universidad Técnica de Ambato.* [En línea] 2016. [Citado el: 03 de Septiembre de 2021.] https://repositorio.uta.edu.ec/jspui/handle/123456789/23065.

NAYLAMPMECHATRONICS. Driver Puente H L298N 2A. *Naylampmechatronics.* [En línea] 2021. [Citado el: 12 de Septiembre de 2021.] https://naylampmechatronics.com/drivers/11-driver-puente-h-l298n.html.

NÚÑEZ, Francisco. Diseño de un sistema de reconocimiento automático de matrículas de vehículos mediante una red neuronal convolucional. *Universidad Oberta de Catalunya.* [En línea] 2016. [Citado el: 10 de Septiembre de 2021.] http://openaccess.uoc.edu/webapps/o2/bitstream/10609/52222/7/fnunezsTFM0616memoria.pdf.

OMS. Informe mundial sobre prevención de los traumatismos causados por el tránsito. *Organización Mundial de la Salud.* [En línea] 2004. [Citado el: 05 de Septiembre de 2021.] https://www.who.int/violence_injury_prevention/publications/road_traffic/world_report/summary_es.pdf. 92 4 359131 2.

PICO APONTE, Grecia Magaly; & SALAZAR LOGROÑO, Franklin Wilfrido. Sistema avanzado de asistencia al conductor empleando visión artificial en vehículos de transporte público. *Universidad Técnica de Ambato.* [En línea] 2019. [Citado el: 02 de Septiembre de 2021.] https://repositorio.uta.edu.ec/jspui/handle/123456789/29951.

PYGAME. Interfaz gráfica Pygame. *Pygame.* [En línea] 2020. [Citado el: 16 de Septiembre de 2021.] http://www.pygame.org/hifi.html.

PYTHON. Python. *Python.* [En línea] 2020. [Citado el: 15 de Septiembre de 2021.] https://www.python.org/.

RASPBERRY PI. Paspberry Pi 4 Características principales. *Raspberry.* [En línea] 2020a. [Citado el: 10 de Septiembre de 2021.] www.raspberrypi.org.

RASPBERRY PI. Raspberry Pi 4 Cámara nocturna. *Raspberry.* [En línea] 2020b. [Citado el: 10 de Septiembre de 2021.] https://www.raspberrypi.org/products/camera-module-v2/?resellerType=home.

RASPBIAN. Sistema operativo Raspbian. *Raspbian.* [En línea] 2020. [Citado el: 14 de Septiembre de 2021.] http://www.raspbian.org/.

RIVAS, Wilmer; & MAZÓN, Bertha. Redes neuronales artificiales aplicadas al reconocimiento de patrones. *Universidad Técnica de Machala.* [En línea] 2018. [Citado el: 08 de Septiembre de 2021.] http://repositorio.utmachala.edu.ec/bitstream/48000/14223/1/Cap.1-Generalidades%20de%20las%20redes%20neuronales%20artificiales.pdf. 978-9942-24-100-9.

TIERRA GUSQUI, Juan José; & GALARZA DUCHI, Marco Vinicio. Implementa Implementación de un sistema de control y monitoreo en base al procedimiento de imágenes digitales en los sistemas de visión artificial aplicado al reconocimiento de la máquina selectora de botellas en el laboratorio de ingenieria industrial. *Escuela Superior Politécnica de Chimborazo.* [En línea] 2017. [Citado el: 01 de Septiembre de 2021.] http://dspace.espoch.edu.ec/handle/123456789/9346.

WAVESHARE. Cámara nocturna Raspberry. *Waveshare.* [En línea] 2021. [Citado el: 11 de Septiembre de 2021.] www.waveshare.com.

WILLOGAGE. Open CV. *Willowage.* [En línea] 2020. [Citado el: 16 de Septiembre de 2021.] https://www.willowgage.com/pags/software/opencev.

X20. Imágen con mara térmica. *X20.* [En línea] 2020. [Citado el: 10 de Septiembre de 2021.] www.x20.org.

ANEXOS

ANEXO A: CÓDIGO ARDUINO PARA MEDIR LA VELOCIDAD DE LAS RUEDAS

```
int encoder_pin = 2;             //Pin 2, donde se conecta el encoder
unsigned int rpm = 0;           // Revoluciones por minuto calculadas.
float velocity = 0;                //Velocidad en [Km/h]
volatile byte pulses = 0;       // Número de pulsos leidos por el Arduino en un segundo
unsigned long timeold = 0;  // Tiempo
unsigned int pulsesperturn = 20; // Número de muescas que tiene el disco del encoder.
const int wheel_diameter = 64;   // Diámetro de la rueda pequeña[mm]
static volatile unsigned long debounce = 0; // Tiempo del rebote.
////  Configuración del Arduino ////////////////////////////////////////////////////////
void setup(){
   Serial.begin(9600); // Configuración del puerto serie
   pinMode(encoder_pin, INPUT); // Configuración del pin n°2
   attachInterrupt(0, counter, RISING); // Configuración de la interrupción 0, donde esta conectado.
   pulses = 0;
   rpm = 0;
   timeold = 0;
  Serial.print("Seconds ");
  Serial.print("RPM ");
  Serial.print("Pulses ");
  Serial.println("Velocity[Km/h]");}
////  Programa principal ////////////////////////////////////////////////////////////////////
 void loop(){
   if (millis() - timeold >= 1000){  // Se actualiza cada segundo
      noInterrupts(); //Don't process interrupts during calculations
      // Desconectamos la interrupción para que no actué en esta parte del programa.
      rpm = (60 * 1000 / pulsesperturn )/ (millis() - timeold)* pulses; // Calculamos las rpm
      velocity = rpm * 3.1416 * wheel_diameter * 60 / 1000000; // Cálculo de la velocidad en [Km/h]
      timeold = millis(); // Almacenamos el tiempo actual.
      // Se envia al puerto serie el valor de tiempo, de las rpm y los pulsos.
      Serial.print(millis()/1000); Serial.print("       ");
      Serial.print(rpm,DEC); Serial.print("   ");
      Serial.print(pulses,DEC); Serial.print("     ");
      Serial.println(velocity,2);
      pulses = 0;  // Inicializamos los pulsos.
      interrupts(); // Restart the interrupt processing // Reiniciamos la interrupción
   }
  }
////Fin de programa principal ////////////////////////////////////////////////////////////////////////////////
//////////////////////////////Función que cuenta los pulsos buenos ////////////////////////////////////////
 void counter(){
  if(  digitalRead (encoder_pin) && (micros()-debounce > 500) && digitalRead (encoder_pin) ) {
// Vuelve a comprobar que el encoder envia una señal buena y luego comprueba que el
//tiempo es superior a 1000 microsegundos y vuelve a comprobar que la señal es correcta.
        debounce = micros(); // Almacena el tiempo para comprobar que no contamos el rebote que hay en la señal.
        pulses++;}  // Suma el pulso bueno que entra.
        else ; }
```

ANEXO B: CÓDIGO DEL PROGRAMA PRINCIPAL

```
import cv2
import numpy as np
import RPi.GPIO as GPIO
from time import sleep

in1 = 24
in2 = 23
en = 25
temp1=1


GPIO.setmode(GPIO.BCM)
GPIO.setup(in1,GPIO.OUT)
GPIO.setup(in2,GPIO.OUT)
GPIO.setup(en,GPIO.OUT)
GPIO.output(in1,GPIO.LOW)
GPIO.output(in2,GPIO.LOW)
p=GPIO.PWM(en,1000)
p.start(25)

SS_cascade = cv2.CascadeClassifier('stopsign_classifier.xml')
SpS_cascade = cv2.CascadeClassifier('lbpCascade.xml')
SpS1_cascade = cv2.CascadeClassifier('lbpCascade1.xml')
SpS2_cascade = cv2.CascadeClassifier('lbpCascade2.xml')
SpS3_cascade = cv2.CascadeClassifier('lbpCascade3.xml')

cap = cv2.VideoCapture(0)

#reducir la resolución para aumentar el FPS
cap.set(3,320)
cap.set(4,240)

i = 0
j = 0
k = 0
l = 0
m = 0

while True:
    ret, img = cap.read()
    ret, frame = cap.read();
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    GPIO.output(in1,GPIO.HIGH)
    GPIO.output(in2,GPIO.LOW)
    SSs = SS_cascade.detectMultiScale(gray, 1.3, 5)
    for(x,y,w,h) in SSs:
```

```
        #dibujamos un rectangulo sobre la detección
        cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
        roi_gray = gray[y:y+h, x:x+w]
        roi_color = img[y:y+h, x:x+w]
        cv2.rectangle(frame, (x,y),(x+w,y+h),(0,255,0),2)
        cv2.putText(frame,'Senal stop detectada',(x,y-10),2,0.7,(0,255,0),2,cv2.LINE_AA)
        print("Dimensions are(xywh): " + str(x) + str(y) + str(w) + str(h))
        print('Senal stop detectada ' + str(i))
        i+=1
        print("stop")
        GPIO.output(in1,GPIO.LOW)
        GPIO.output(in2,GPIO.LOW)

    SpSs = SpS_cascade.detectMultiScale(gray, 1.1, 4)
    for(a,b,c,d) in SpSs:
        #dibujamos un rectangulo sobre la detección
        cv2.rectangle(img,(a,b),(a+c, b+d),(0,255,0),2)
        roi_gray = gray[b:b+d, a:a+c]
        roi_color = img[b:b+d, a:a+c]
        cv2.rectangle(frame, (a,b),(a+c,b+d),(0,255,0),2)
        cv2.putText(frame,'Senal de transito detectada ',(a,b-10),2,0.7,(0,255,0),2,cv2.LINE_AA)
        print ("Dimensions are(xywh): " + str(a) + str(b) + str(c) + str(d))
        print ("Senal de transito detectada " + str(j))
        j+=1



        #dibujamos un rectangulo sobre la detección
    SpSs1 = SpS_cascade.detectMultiScale(gray, 1.3, 5)
    for(a,b,c,d) in SpSs:
        dibujamos un rectangulo sobre la detección
        cv2.rectangle(img,(a,b),(a+c, b+d),(0,255,0),2)
        roi_gray = gray[b:b+d, a:a+c]
        roi_color = img[b:b+d, a:a+c]
        cv2.rectangle(frame, (a,b),(a+c,b+d),(0,255,0),2)
        cv2.putText(frame,'Senal de 50 km/h ',(a,b-10),2,0.7,(0,255,0),2,cv2.LINE_AA)
        print ("Dimensions are(xywh): " + str(a) + str(b) + str(c) + str(d))
        print ("Senal de 50 km/h " + str(j))
        k+=1
        p.ChangeDutyCycle(25)
```

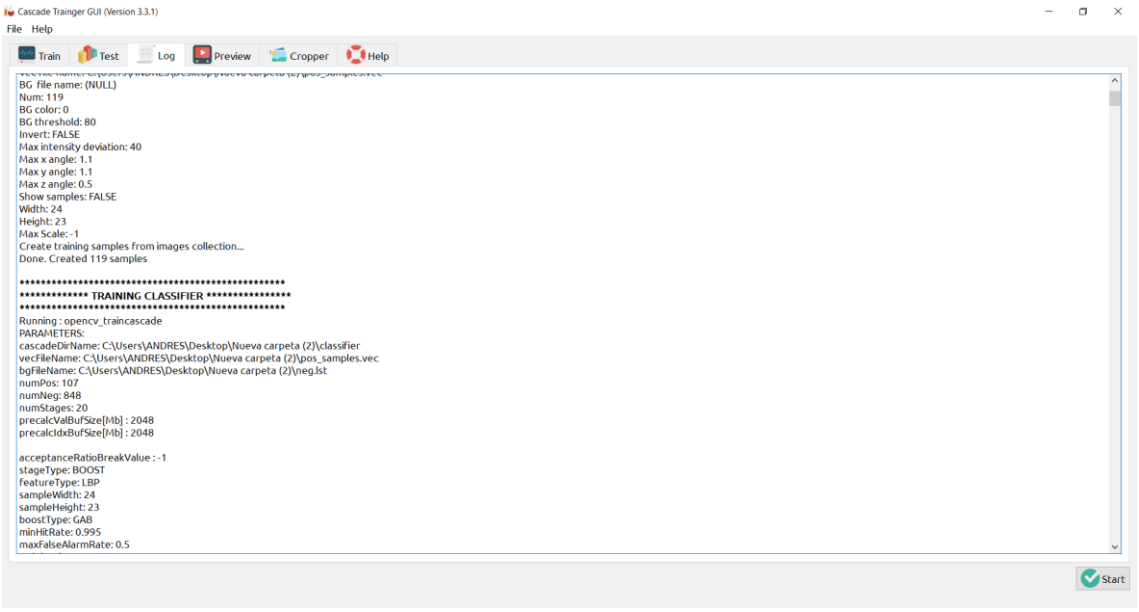
```
        #dibujamos un rectangulo sobre la detección
    SpSs2 = SpS_cascade.detectMultiScale(gray, 1.3, 5)
    for(a,b,c,d) in SpSs:
        dibujamos un rectangulo sobre la detección
        cv2.rectangle(img,(a,b),(a+c, b+d),(0,255,0),2)
        roi_gray = gray[b:b+d, a:a+c]
        roi_color = img[b:b+d, a:a+c]
        cv2.rectangle(frame, (a,b),(a+c,b+d),(0,255,0),2)
        cv2.putText(frame,'Senal de 90 km/h ',(a,b-10),2,0.7,(0,255,0),2,cv2.LINE_AA)
        print ("Dimensions are(xywh): " + str(a) + str(b) + str(c) + str(d))
        print ("Senal de 90 km/h " + str(j))
        l+=1
        p.ChangeDutyCycle(25)

        #dibujamos un rectangulo sobre la detección
    SpSs3 = SpS_cascade.detectMultiScale(gray, 1.3, 5)
    for(a,b,c,d) in SpSs:
        dibujamos un rectangulo sobre la detección
        cv2.rectangle(img,(a,b),(a+c, b+d),(0,255,0),2)
        roi_gray = gray[b:b+d, a:a+c]
        roi_color = img[b:b+d, a:a+c]
        cv2.rectangle(frame, (a,b),(a+c,b+d),(0,255,0),2)
        cv2.putText(frame,'Senal de 100 km/h ',(a,b-10),2,0.7,(0,255,0),2,cv2.LINE_AA)
        print ("Dimensions are(xywh): " + str(a) + str(b) + str(c) + str(d))
        print ("Senal de 100 km/h " + str(j))
        m+=1
        p.ChangeDutyCycle(25)

    k = cv2.waitKey(20) & 0xff
    if k == 27:
        break
    cv2.imshow('frame',frame)
    if cv2.waitKey(20) & 0xFF == ord('q'):
        break

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

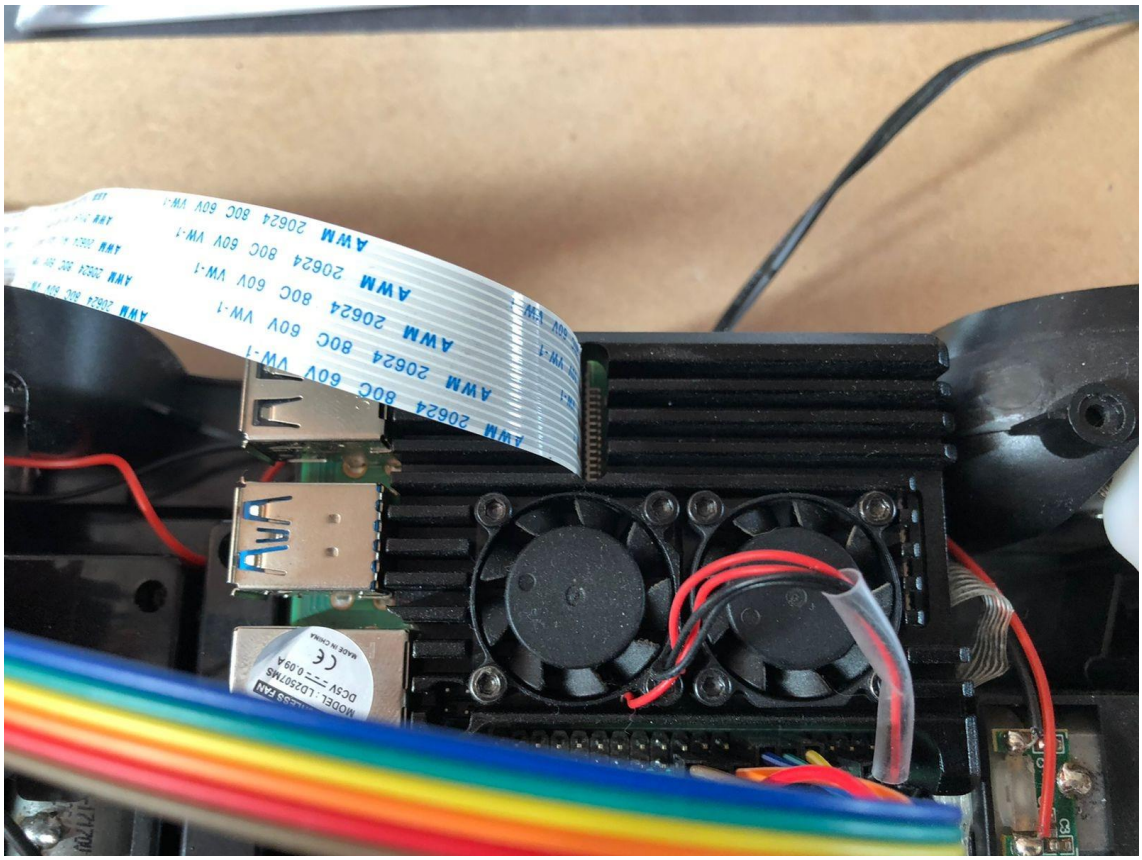
ANEXO C: ENTRENAMIENTO EN EJECUCIÓN



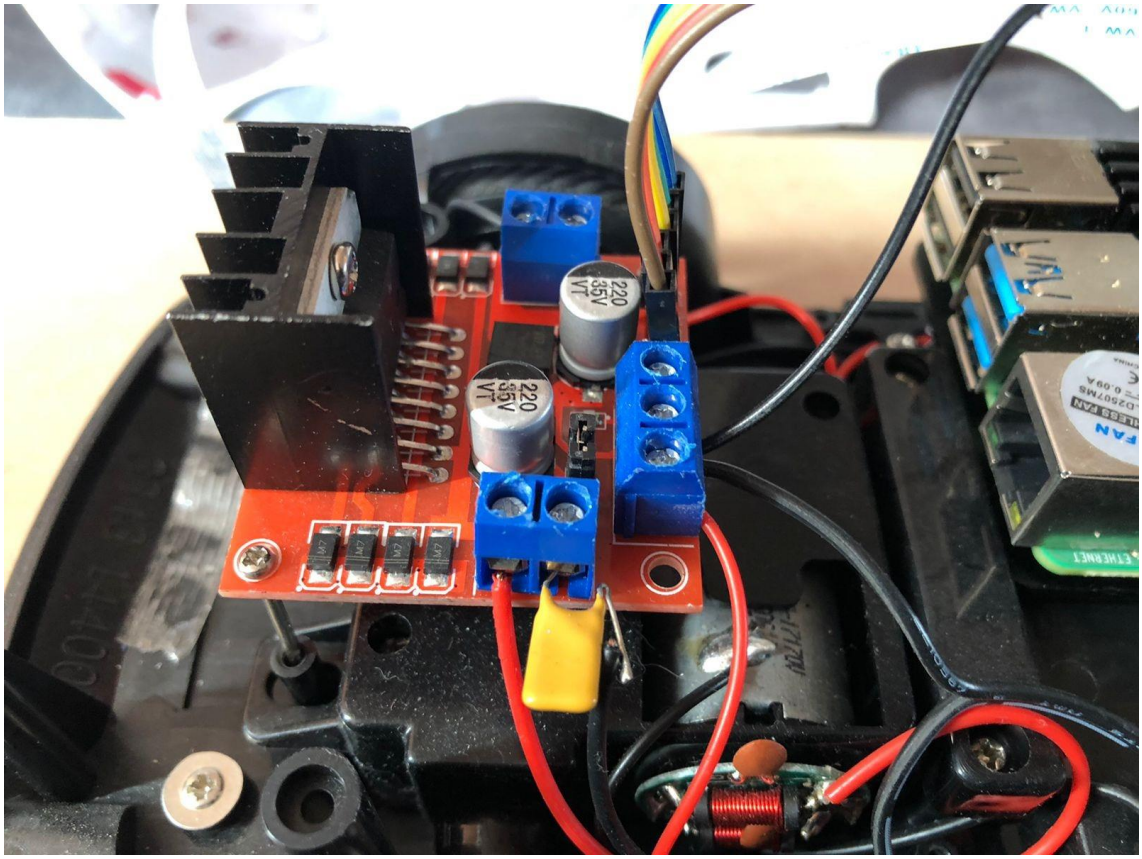
ANEXO D: PROTOTIPO TERMINADO


Macan turbo

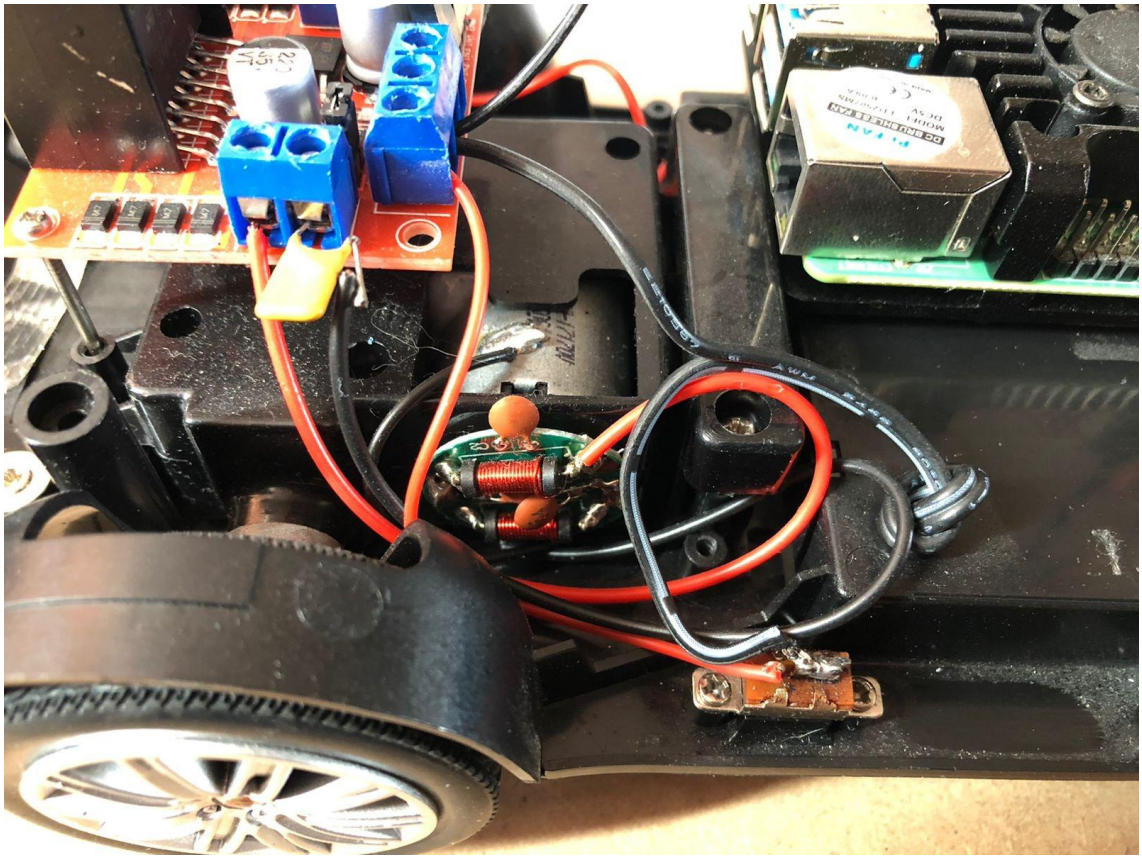
ANEXO E: DISPOSITIVO INTERNO



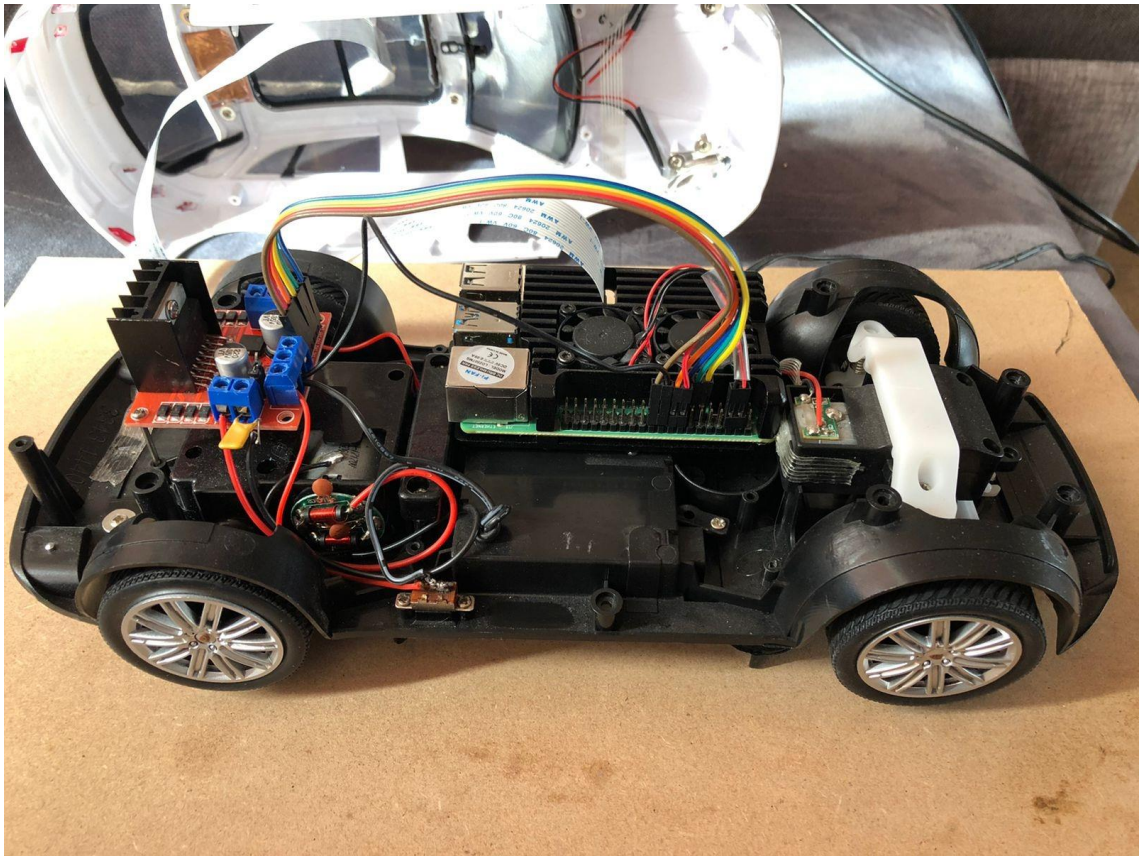
ANEXO F: DRIVER L298N



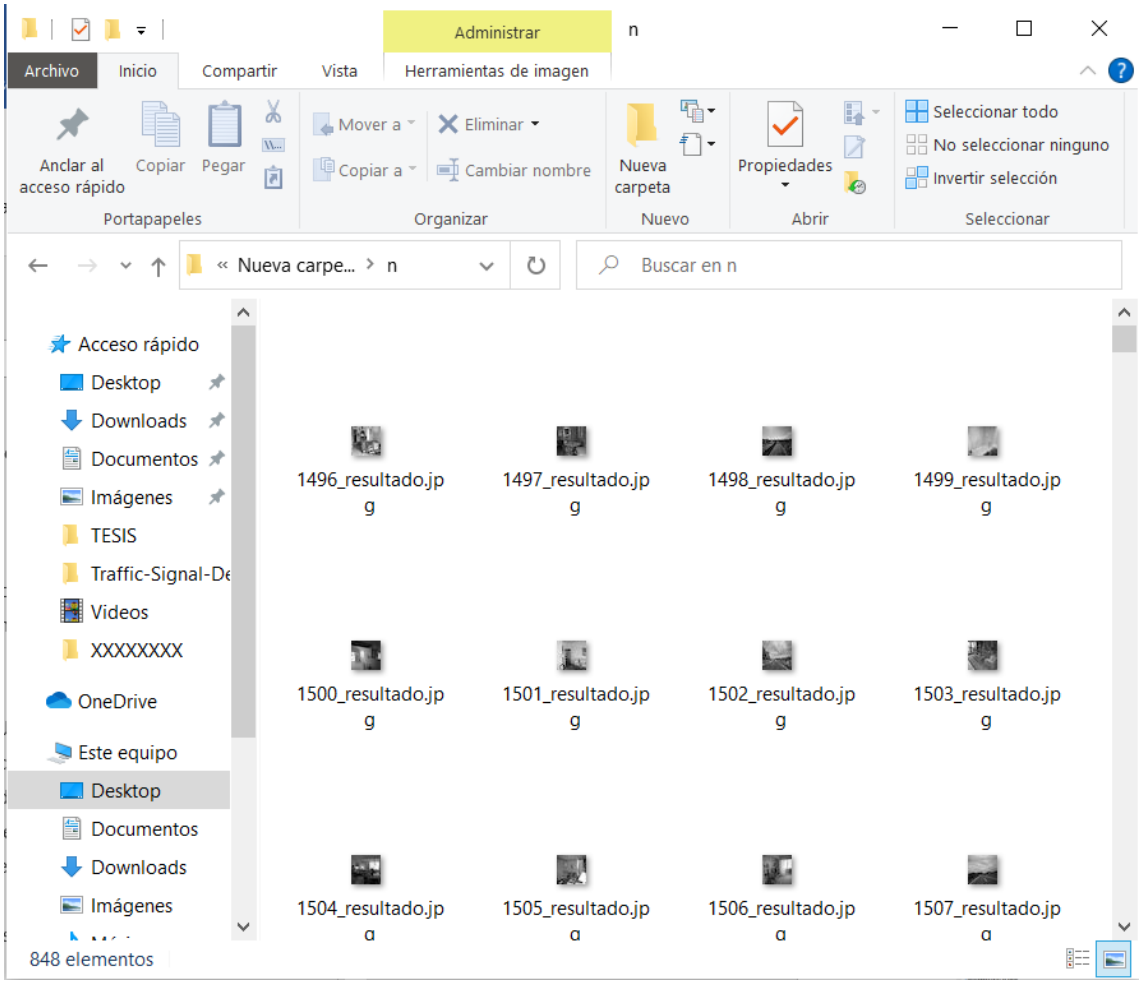
ANEXO G: CIRCUITO INTERNO



ANEXO H: CIRCUITO INTERNO DE TODO EL VEHÍCULO



ANEXO I: IMÁGENES NEGATIVAS UTILIZADAS



ANEXO J: CASCADE TRAIGER