

Research Article

Development of an Ackermann-type Mobile Platform for the Validation of Autonomous Navigation Algorithms within the ESPOCH Campus

Desarrollo de una Plataforma Móvil Tipo Ackermann para la Validación de Algoritmos de Navegación Autónoma Dentro del Campus ESPOCH

Andres Barahona*, Kleber Rios, Javier Gavilanes, and Johnny Gavilanes

Facultad de Mecánica, Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador

ORCID

Andres Barahona: 0000-0001-9949-1249

IV INTERNATIONAL
CONGRESS OF
ENGINEERING FACULTY OF
MECHANICS (IV COINTI
2021)

Corresponding Author:
Andres Barahona; email:
andres.barahona@espoch.
edu.ec

Published: 31 August 2022

Production and Hosting by
Knowledge E

© Andres Barahona
et al. This article is distributed
under the terms of the
[Creative Commons
Attribution License](#), which
permits unrestricted use and
redistribution provided that
the original author and
source are credited.

Abstract

The continual development of the automotive industry has evolved to astronomical levels, capable of promoting a new technological generation of several automated systems (both mechanical and electronic systems). As a result, a generation of autonomous vehicles – also known as intelligent vehicles – capable of avoiding human errors has emerged. In the present investigation, an Ackermann-type mobile prototype is used to validate autonomous navigation algorithms in different environmental conditions, which is developed by covering artificial vision stages such as the detection of the bicycle lane and traffic signals at scale. The control system consists of three parts – First is the detection algorithm through an artificial vision to collect information from the environment through a Pi camera that can later be processed through Python with its OpenCV tool. The second deals with the training of traffic signs (Stop and Speed Signage) through the Haar Cascade, as well as the detection of road lines through which various filters such as Canny, edge detection, and the transformation of Hough are executed on the Python platform with the OpenCV tool. The third part is the communication between the processing of the codes and their respective actuators (Motor, Servomotor) to finally collect the statistical data and validate the algorithms in the Ackerman-type mobile prototype.

Keywords: *Python, OpenCV, Canny Filter, Hough Transform, Haar Cascade.*

Resumen

El desarrollo constante de la industria automotriz evoluciona a niveles astronómicos capaz de impulsar una nueva generación tecnológica de automatización de varios los sus sistemas ya sean: mecánicos o electrónicos y así forma la generación de vehículos autónomos o llamados también vehículos inteligentes, capaces de evitar errores humanos al conducir. En la presente investigación se implementa un prototipo móvil tipo Ackermann para validar algoritmos de navegación autónoma en diferentes condiciones de su entorno que se desarrolle cubriendo etapas de visión artificial como la detección del carril de una ciclovía y señales de tránsito a escala. El sistema de control consta de tres partes, la primera parte es el algoritmo de detección por medio de visión artificial para recolectar información del entorno a través de una cámara Pi y posteriormente ser procesada a través de Python con su herramienta OpenCV. La segunda parte se encarga del entrenamiento de las señales de tránsito (Pare y Señalética de Velocidad) por medio de Haar cascade, además la detección las líneas de la carretera las cuales a través.

 OPEN ACCESS



de varios filtros como Canny, detección de bordes y la transformada de Hough todos estos filtros se lo ejecutan en la plataforma de Python con la herramienta de OpenCV, y la tercera parte es la comunicación entre el procesamiento de los códigos a sus respectivos actuadores (Motor, Servomotor) para finalmente recolectar datos estadísticos y Validar los algoritmos en el prototipo móvil tipo Ackerman.

Palabras Clave: *Python, Open, Filtro Canny, Transformada de Hough, Haar Cascade.*

1. Introducción

1.1. Antecedentes

Las empresas de automoción están inmersas en múltiples procesos competitivos en una carrera imparable hacia el desarrollo de vehículos autónomos [1]. Algunas de estas empresas que se han destacado son: Tesla, Uber, Google y Amazon; conocidas por su apoyo al desarrollo tecnológico, sin excluir a las grandes empresas del sector automotriz tradicional [2]. Por otro lado, existen centros tecnológicos y de investigación que desarrollan proyectos innovadores para la evolución constante de nuevas tecnologías y la ambición de construir el primer prototipo autónomo en vehículos, es el caso del prototipo de la Universidad Libre de Berlín a cargo del profesor Rojas (2020) y la Universidad Politécnica de Madrid con el prototipo Autopía (2020) [3]. Cada una de estas se comprometieron al desarrollo del futuro en las llamadas ciudades inteligentes donde los vehículos autónomos forman parte de los paisajes urbanos de las nuevas ciudades que están previstas llegar en el 2030 [1].

Los trabajos antes mencionados, proponen sistemas físicos combinados con modelos inteligentes, en los que el aprendizaje profundo está llamado a jugar un papel relevante. Es en esta perspectiva se propone el presente trabajo, motivado por el interés en desarrollar un prototipo a pequeña escala, que incorpore tecnologías de prototipos desarrollados por los fabricantes de automóviles antes mencionados [4]. El prototipo que se incorporará va a contar con un sistema de visión artificial capaz de leer tanto las líneas de la carretera de una ciclovía, simulando un escenario de la vía real por la cual va a conducir un vehículo autónomo y por medio de una red neuronal identificar y clasificar señales de tránsito. Se establece una comunicación conveniente entre el procesador a bordo del vehículo, en este caso una Raspberry Pi 2, y la computadora central (remota) donde se realizan la ejecución y validación de algoritmos de navegación autónoma por medio de Python y OpenCv, como procesamiento de imágenes y clasificación de objetos, utilizando la red neuronal [5].

De esta forma, el prototipo tipo Ackerman de coche inteligente llamado ESPOCH01 establece una propuesta un poco más avanzada que una simulación. Este proyecto propone un modelo que funciona con una Raspberry Pi 2, por medio de Python y Open Cv va a ser el encargado de dar las órdenes a los distintos sistemas tales como: la dirección que es controlada por un Arduino 1 que enviara la señal a un servomotor MG995 y un IBT_2 que va a controlar el giro y la potencia que va a funcionar el motor de 12 voltios.

Tomando la guía en relación con trabajos semejantes, la comunicación entre los diferentes sistemas del prototipo se realiza mediante WIFI, gracias a una dirección IP se puede controlar las diferentes acciones y observar por medio de una aplicación llamada VNC, vincular desde el ordenador al prototipo en tiempo real y ejecutar los algoritmos de control por medio de Python y OpenCV. Los algoritmos para el preprocesamiento de imágenes hacen uso de filtros tales como Filtro Canny, Escala de Grises. Y la clasificación para permitir el reconocimiento de imágenes hace uso de líneas de Hough y Haar Cascade que detectan la señal de tránsito según el entrenamiento previo que se haya realizado [5].



Figure 1

Prototipo Vehiculo Autonomo ESPOCH01.

2. Materiales y Métodos

El objetivo central consistió en desarrollar una plataforma móvil tipo Ackermann equipada con visión artificial para validar algoritmos de navegación autónoma dentro del Campus ESPOCH, consulte la Figura 2

La propuesta desarrollada fue un prototipo de vehículo a escala autónoma, que mediante el reconocimiento de imágenes identificar tanto las líneas de la ciclovía y las señales de tránsito a escala, en este caso una plataforma para su identificación y localización para poder navegar sin problemas. Como se indicó anteriormente, el prototipo consta de una serie de componentes de hardware y software. En este apartado se describen los algoritmos inteligentes enmarcados dentro de este último, dejando el resto en la descripción del prototipo [6].

2.1. Estructura

Como se observa en la Figura 2 identificamos el sistema de comunicación de cada uno de los elementos del prototipo. La particularidad del prototipo es la intercomunicación entre: la cámara pi, router y el ordenador para la ejecución de los códigos hacia la raspberry pi 2 para poder ejecutar cada una de las operaciones según los datos recibidos de la cámara pi en tiempo real según el entorno, las condiciones que se desarrolle y enviar la información hacia la raspberry para ejecutar las ordenes hacia los actuadores.

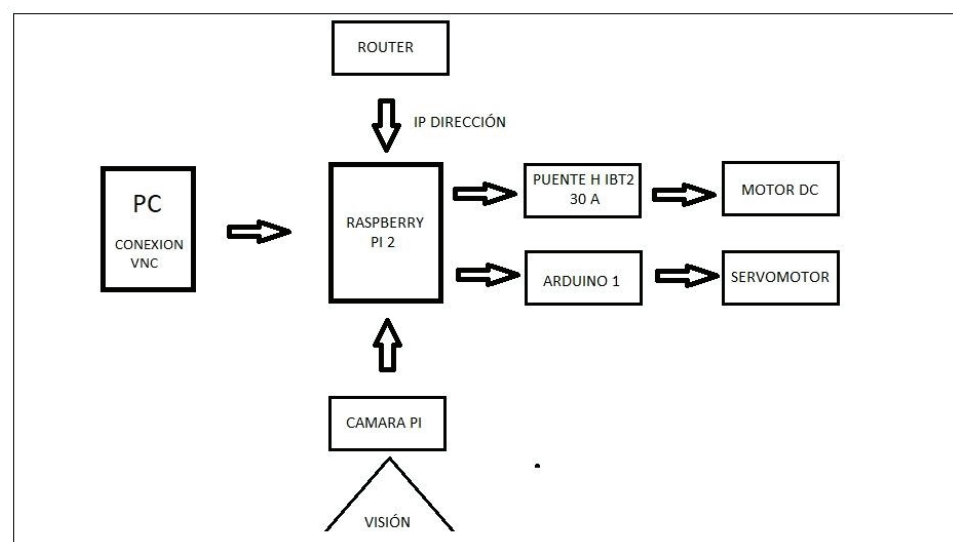


Figure 2

Estructura sistema de control prototipo autónomo.

2.2. Dirección tipo Ackermann

El prototipo tiene una dirección tipo Ackermann debido a que la mayoría de los vehículos tradicionales que existen en el Ecuador poseen este sistema con variaciones de

elementos, nuestro prototipo posee este sistema debido a que con las modificaciones que tienen los distintos vehículos, tienen el mismo principio funcionamiento [7].

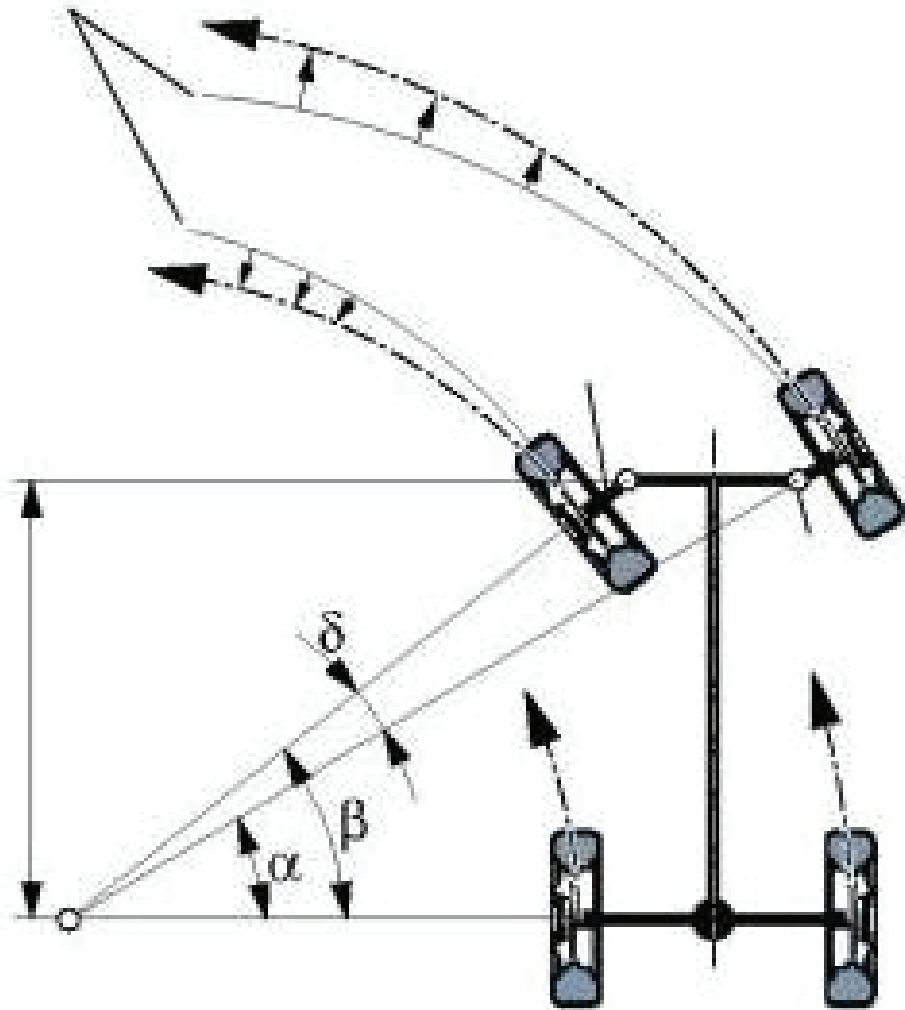


Figure 3

Principio de Ackerman.

El principio de Ackerman establece que cuando un vehículo gira en una curva, los ejes de todas las ruedas deben coincidir en un punto, el centro instantáneo de rotación, se logra una geometría de dirección óptima [7]. Para seguir este principio, el ángulo de giro de la rueda interior se hace mayor que el exterior, es decir, como se muestra en la Figura 3.

2.3. Flujograma de visión artificial prototipo tipo Ackerman

En la Figura 4 observamos el procesamiento de los diferentes filtros que atraviesa la imagen captada por medio de la cámara, convertir una imagen RGB a BRG debido a



que la imagen que capta la cámara pi no va a leer los filtros por eso se le transforma la imagen [8]. Para la detección de carriles de la ciclovía por el cual va circular el prototipo, a través de los diferentes filtros como son: filtro canny que va detectar bordes, posterior rellenar las líneas de la carretera para poder seccionar la imagen y la transformada de hough para dibujar las líneas en el contorno de las líneas de la carretera para poder detectar la posición real donde se encuentra el prototipo [9].

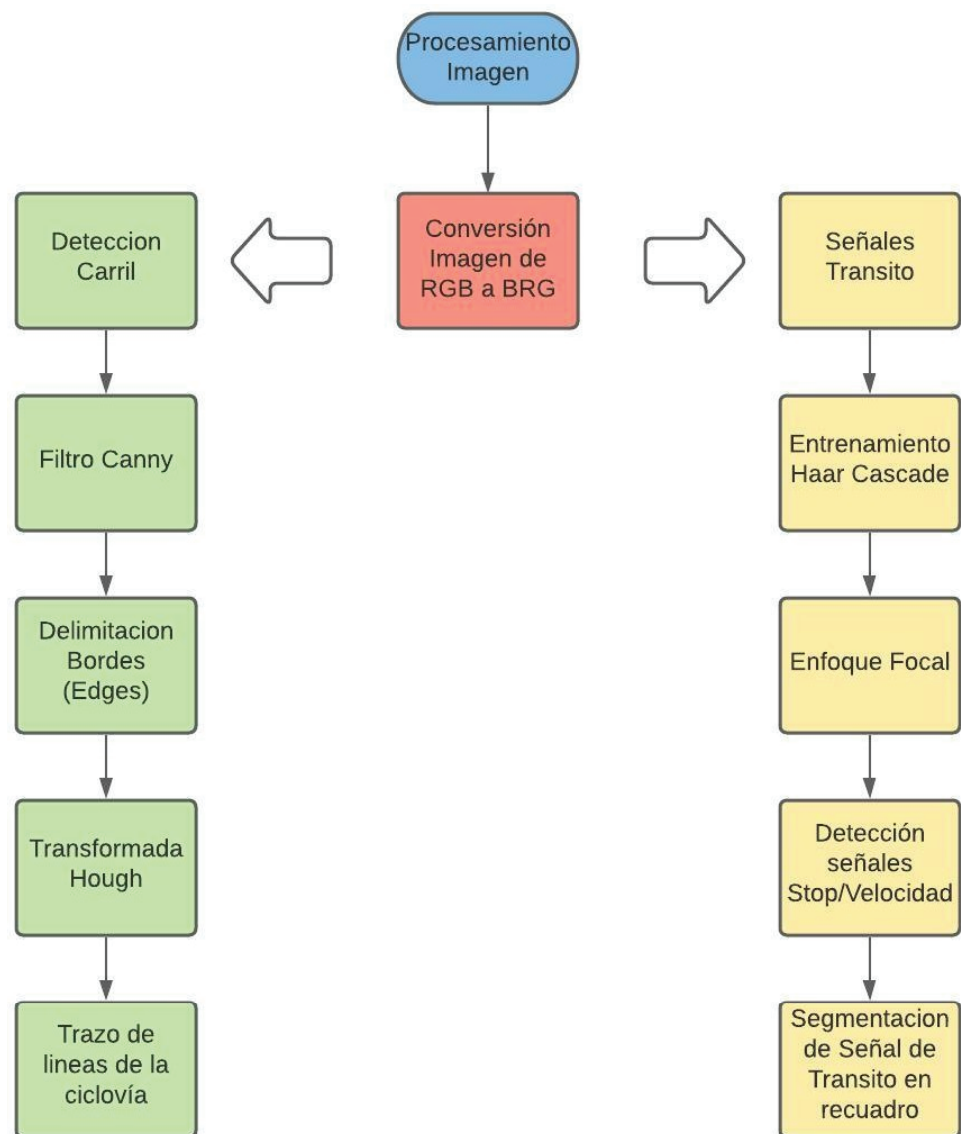


Figure 4

Diagrama de flujo Procesamiento de la imagen del prototipo ESPOCH01.



2.4. Esquema de conexión del circuito

El esquema del circuito y sus correspondientes conexiones entre: la Raspberry Pi, Arduino y Puente ibt_2 como se muestra en la Figura 5.

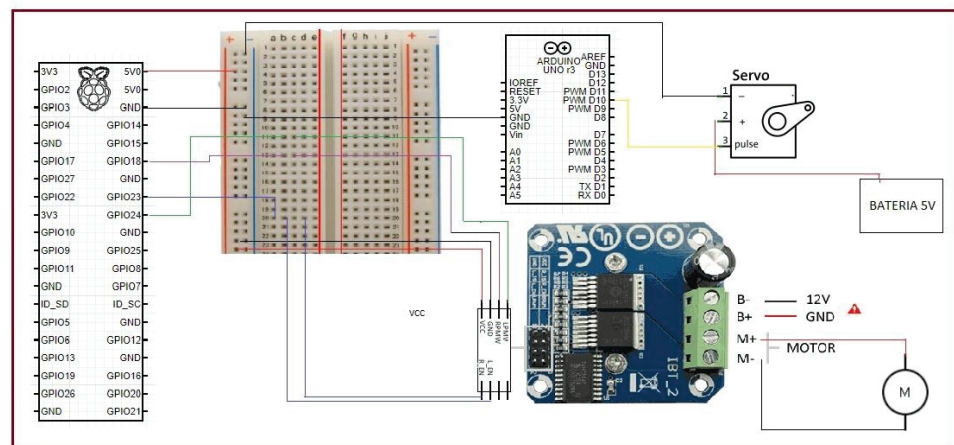


Figure 5

Diagrama de conexión prototipo ESPOCH01.

Como podemos observar en la Figura 5 el diagrama de conexión del propio ESPOCH01, tiene una batería de 12 V y una alimentación de 5V para el servomotor, además una fuente de alimentación independiente para la Raspberry, todas las GND deben tener una tierra común para un correcto funcionamiento y comunicación entre elementos.

2.5. Detección de carril en tiempo real

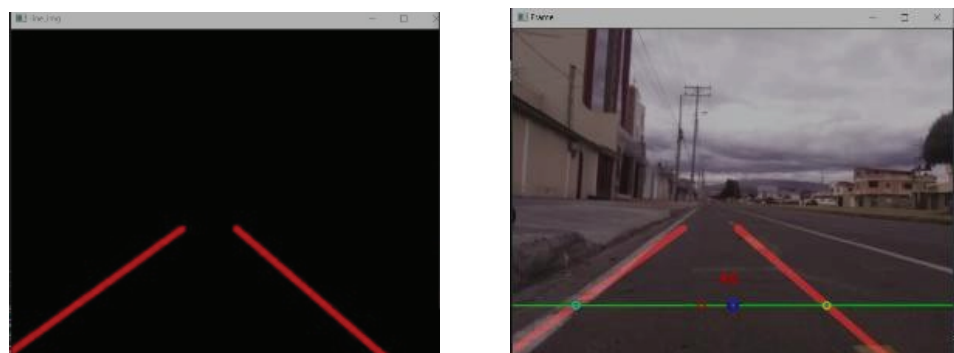


Figure 6

Detección en tiempo real del Carril con el algoritmo antes mencionado.

Como podemos observar en la Figura 6, el algoritmo de navegación autónoma funciona transformando la imagen real en una de escala de grises, luego se le transforma por medio del Filtro Canny realizamos pruebas hasta tener una imagen clara de solo las líneas, posterior a eso se realiza una transformada de las Líneas de Hough como se observa en la imagen de la izquierda [10].

Luego de obtener las líneas se saca el centro de cada uno de los carriles, luego se calcula el centro de la imagen para que la Raspberry mande la información hacia el Arduino y este envíe la señal por medio de PWM al servomotor para corregir el giro de los grados según vaya corrigiéndose constantemente del carril [10].

2.6. Detección de señales de tránsito en tiempo real



Figure 7

Detección en tiempo real de señales de Tránsito con el algoritmo.

Como podemos observar en la Figura 7, se muestra el funcionamiento del código de navegación autónoma de detección de señales de tránsito, el cual detectara en función a la cámara pi una señal de STOP, VERLOCIDAD 30 o 60 KM/H. Por medio del o HAAR CASCADE para la detección de la señal y posteriormente al corte de energía del motor para el frenado en caso de la señal de STOP y acelerar o disminuir la velocidad según la señal de tránsito que detecte.

3. Resultados

Para las estimaciones de los resultados a continuación se hicieron las pruebas en un horario del día en donde no hubo lluvia, ni condiciones de poca luz, estaba en un entorno visible con una luz tenue cálida, así mismo se hizo diferentes pruebas todas en la ciclovía dentro del campus ESPOCH

3.1. Detección de comunicación

Esta prueba consiste en un rango en donde nuestra red wifi abarque y satisfaga las necesidades de ese momento de la lectura de la cámara pi, el procesamiento y la capacidad de analizar la información de acuerdo con la complejidad de los códigos de detección por una red neuronal con Haar cascade y a su vez su posterior análisis y despliegue de señal a los actuadores, a continuación, se muestra el número de aciertos en una cantidad de metros establecidos [11].

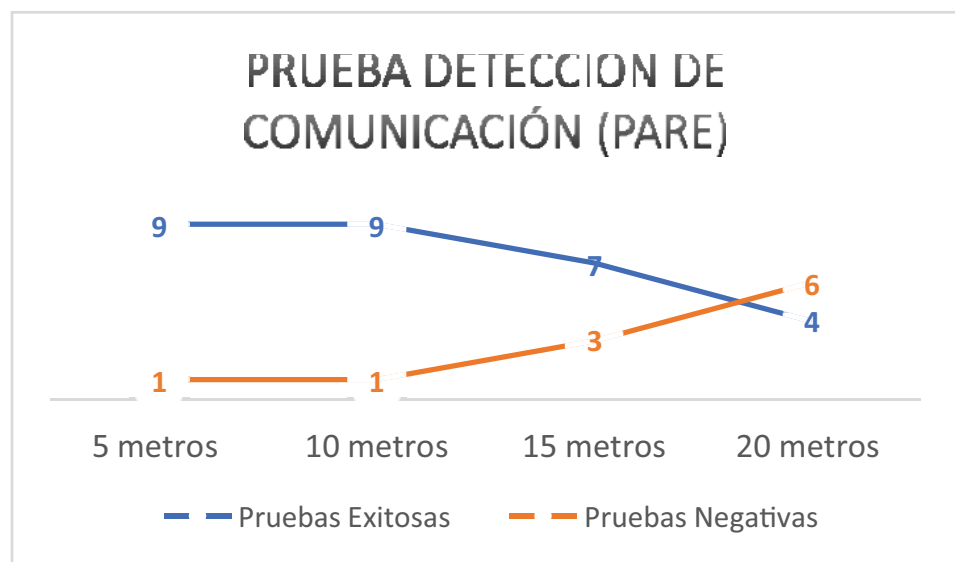


Figure 8

Gráfico de prueba de detección de comunicación exitosas y negativas.

A mayor el número de metros la señal se perderá por lo que fallará más, esto se debe básicamente a la limitante del rango de señal que proporciona el router, pero para el propósito de enseñanza este cumple con lo indicado, teniendo pocos fallos dentro de los metros adecuados [12].

3.2. Detección de carril

Esta prueba se basa en los metros capaces de avanzar detectando las líneas de carril de la ciclovía, está separada en 4 tramos en donde se hizo 10 pruebas para cada tramo. El código fue hecho para la lectura de dichas líneas semejando a escala lo que serían líneas de carretera para autos normales, así mismo se usó diferentes filtros para el procesamiento de datos como lo son los filtros Canny, Transformada de Hough [13].

Como se observa a mayor distancia falla un poco la estabilización de prototipo dentro de las líneas. Muy probablemente sea por la comunicación ya que dentro del rango de

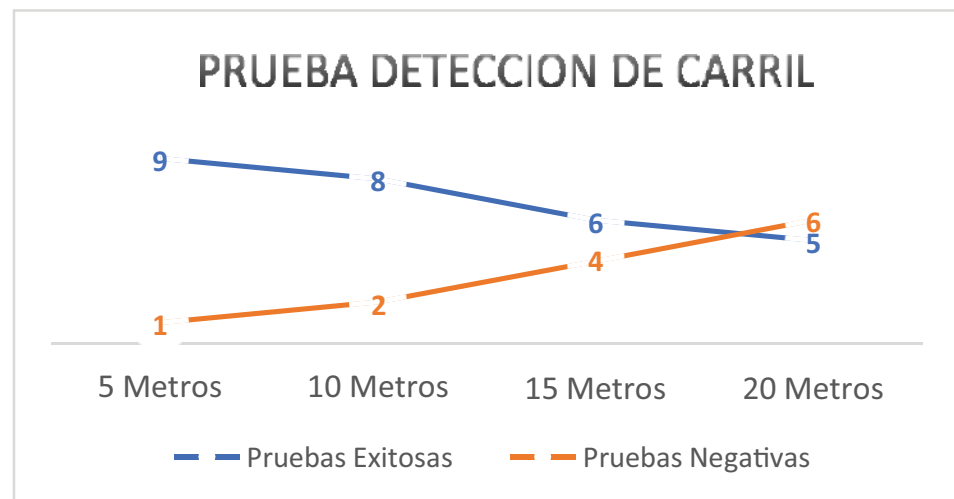


Figure 9

Gráfico de prueba de detección de carril exitosas y negativas.

los 10 metros este tiene pocas pruebas fallidas cumpliendo con el objetivo de validación del código para líneas de carril.

3.3. Enfoque focal

Esta prueba se basa en la distancia de detección de las señales de tránsito, en el caso del PARE el prototipo deberá para al detectarlo, por otra parte, en la de velocidad este cambiara su potencia de marcha al adecuado, estos dos tipos de detecciones están hechos con una red Neuronal Haar Cascade la cual para ellos anteriormente se hizo un entrenamiento con imágenes positivas y falsos positivos [13]. La prueba consiste en ver por cada tramo cuantas operaciones exitosas existen siendo aquellas las cuales las procesa correctamente [14].

Ambas tienen un rendimiento parecido siendo un promedio de 1 metro el correcto alcance para la mejor toma de datos y reconocimiento de este, esto se debe a muchos factores, pero el principal es el tipo de cámara, posición de la cámara y estado de la luz del entorno, existen cámaras con mejores prestaciones y hasta lecturas en 3D que ayudarían a extender la distancia [15]. Pero para el modelo a escala es el correcto distanciamiento de lectura validando correctamente los algoritmos con un promedio de 1 metro de lectura.

Como podemos observar en la Figura 12, realizamos una prueba de intervalo de confianza para corroborar la información adquirida en la Figura 10 y 11, por medio de la herramienta Minitab. Arrojános un intervalo de confianza del 95 % en una distancia

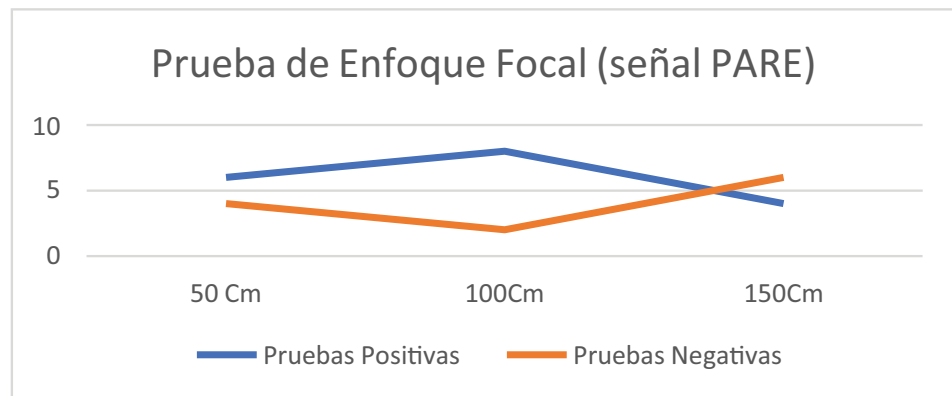


Figure 10

Gráfico de prueba de enfoque focal (señal PARE) exitosas y negativas.

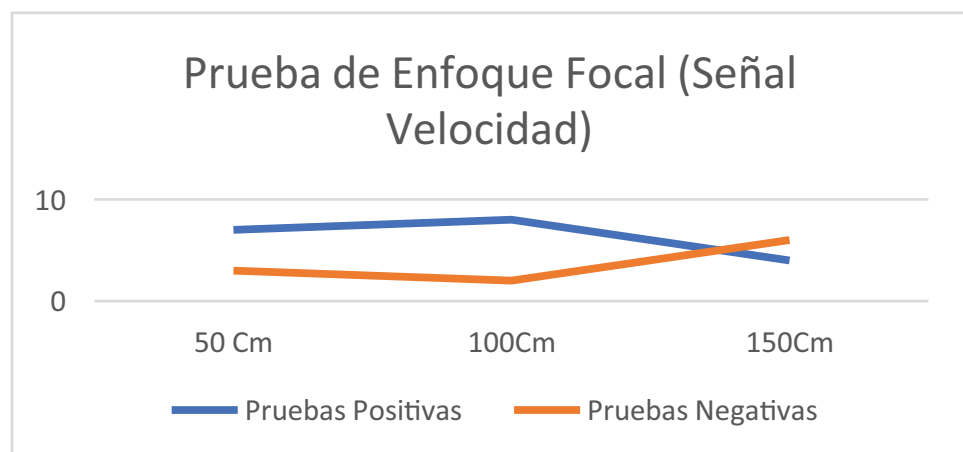


Figure 11

Gráfico de prueba de enfoque focal (señal de velocidad) exitosas y negativas.

de 14.5 a 15 cm del enfoque focal en el horario de 10 a 11 del día con condiciones ideales de luminosidad.

4. Conclusiones

1. Se presentó una plataforma móvil tipo Ackermann la cual está acoplada con funciones para la validación de algoritmos de navegación autónoma por medio de visión artificial, sistema con orientación a aplicaciones de seguimiento de líneas de carretera, detección óptima de la señal de tránsito "PARE" y de la señalética por parte de un semáforo, todo implementado con OPENCV y una Raspberry pi 2 con un procesamiento de imagen por medio de una cámara pi

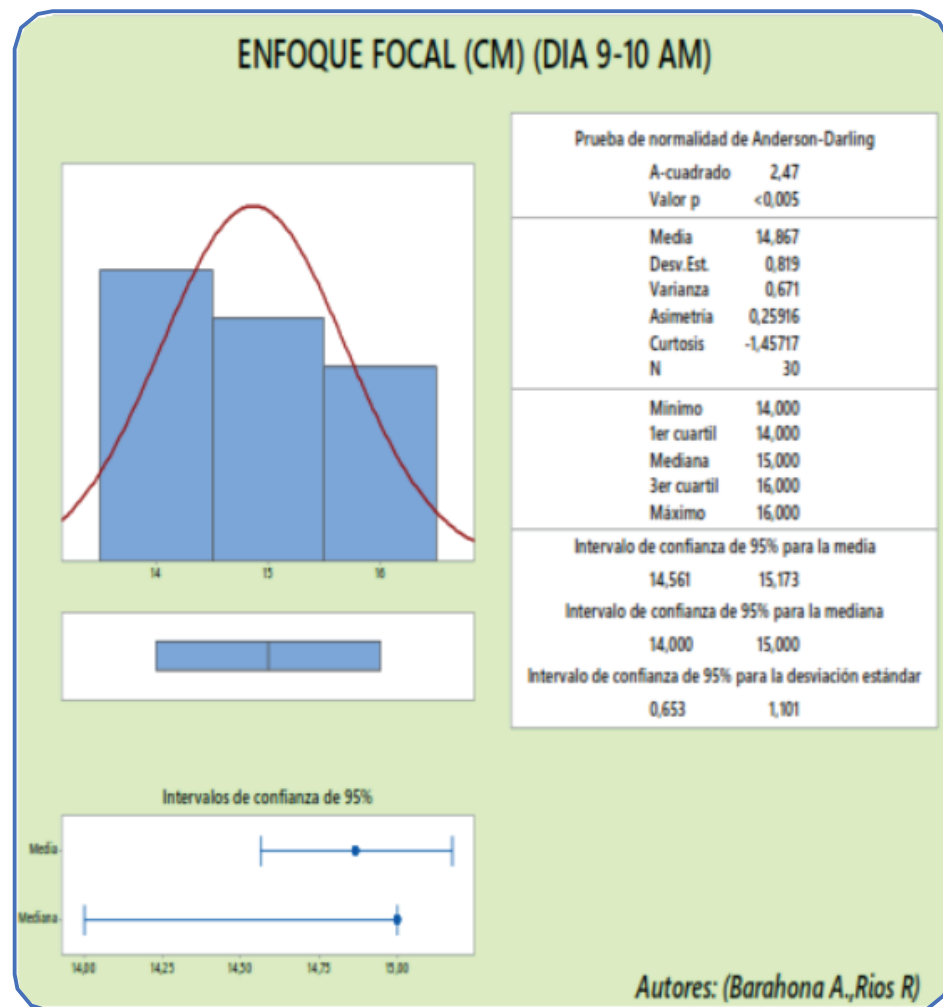


Figure 12

Enfoque focal, minitab.

2. La obtención de resultados para la detección de la señalética varía de acuerdo: al color, tipo de señalética y tipo de condiciones climáticas que se posea en el momento del procesamiento de esta, ya que el enfoque de la cámara varía a las condiciones climáticas y al tipo de comunicación que se poseen debido a la señal wifi que tiene un rango delimitado de conexión máximo de 20 metros, debido a las pruebas realizadas se promedió dándonos un resultado del 45% de eficiencia.
3. Conforme a los resultados obtenidos acerca de las pruebas del enfoque focal por medio de la herramienta estadística Minitab, realizamos pruebas a diferentes intervalos de distancia. Dándonos como resultado un intervalo de confianza del 95% a una distancia de 14.5 a 15cm de la distancia de la cámara con respecto al prototipo.



4. De acuerdo a los resultados, se ha observado que el rango óptimo de funcionamiento del prototipo es de 5 metros hasta los 10 metros, debido a la comunicación entre la Raspberry y ordenador por medio de una dirección IP para la detección de líneas de la ciclovía, mientras para las señales de tránsito el funcionamiento óptimo por medio de un enfoque focal es de 1 metro, esto se debe a la baja comunicación por parte de la red wifi y a la capacidad de la cámara en pixel, no obstante debajo de esos metros mencionados vemos una eficacia arriba del 75%

References

- [1] Punagin A, Punagin S. Analysis of lane detection techniques on structured roads using OpenCV. *International Journal for Research in Applied Science and Engineering Technology*. 2020;8:2994–3003.
- [2] Rossi A, Ahmed N, Salehin S, Choudhury TH, Sarowar G. Real-time lane detection and motion planning in Raspberry Pi and Arduino for an autonomous vehicle prototype. *arXiv [Preprint]*. 2020: arXiv200909391.
- [3] Bose S. Traffic detection using OpenCV. 2020. Available from: <https://ssrn.com/abstract=3604892>
- [4] Farag W. Real-time detection of road lane-lines for autonomous driving. *Recent Advances in Computer Science and Communications (Formerly Recent Patents on Computer Science)*. 2020;13(2):265–274.
- [5] Monterrey Cañas ÁM, Sosa Ramírez CA. Diseño de un sistema de semaforización inteligente para controlar flujo vehicular a partir de procesamiento de imágenes. Colombia: Universidad Piloto de Colombia Sede Administrativa; 2020.
- [6] Byeloborodov Y, Rashad S. Design of machine learning algorithms for behavioral prediction of objects for self-driving cars. In: *11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON) [Internet]*; 2020 Oct 28–31. p. 101–105.
- [7] Olea JAR. Vehículo autónomo de configuración Ackermann para seguimiento de carril mediante procesamiento digital de imagen. *Revista Científica Aristas*. 2020;8(15):205–210.
- [8] Chen Q, Chen Y, Zhu J, De Luca G, Zhang M, Guo Y. Traffic light and moving object detection for a guide-dog robot. *Journal of Engineering*. 2020;2020(13):675–678.
- [9] Palacios O, Enrique R. Control adaptativo para el seguimiento de trayectorias de un robot móvil en carsim [master's thesis]. México: Tecnológico Nacional de México



Campus Tuxtla Gutiérrez; 2020.

- [10] Sugue V, Jesús de J. Análisis de algoritmos de detección de características de OpenCV en Raspberry Pi [dissertation]. Spain: Universidad de Sevilla; 2020.
- [11] Morales A, Junieth R. Reconocimiento de placas vehiculares aplicando procesamiento de imágenes digitales en Python-OpenCV [master's thesis]. Nicaragua: National Autonomous University of Nicaragua; 2020.
- [12] Flores Arrieta CD. Universidad Politécnica Salesiana Sede Quito [Internet, master's thesis]. 2020. Available from: <http://dspace.ups.edu.ec/handle/123456789/18503><http://dspace.ups.edu.ec/bitstream/123456789/5081/1/UPS-CYT00109.pdf>
- [13] Bermejo JML. Características de la infraestructura que pueden favorecer la conducción asistida y automatizada. *Rev Digit del Cedex*. 2020;197:106–117.
- [14] Alonso MAP, Succar LES, Balderas HJE, Erro LE, de Tonantzintla SM. Detección de novedades visuales basada en aprendizaje profundo para manejo autónomo. Mexico: INAOE; 2020.
- [15] Llambo Yansaguano FB. Sistema electrónico de corrección del gesto deportivo en base a la biomecánica en deportistas de marcha atlética. Ecuador: Universidad Técnica de Ambato; 2020.