

## Análisis de un sistema de conducción autónoma

### Analysis of an Autonomous Driving System

Zabulón Miguel **Hernández-Olivares**<sup>1</sup>, Alberto Jorge **Rosales-Silva**<sup>2</sup>,  
Armando Adrián **Miranda-González**<sup>3</sup>, Dante **Mujica-Vargas**<sup>4</sup>,  
Ponciano Jorge **Escamilla-Ambrosio**<sup>5</sup>, Jean Marie **Vianney-Kinani**<sup>6</sup>,  
Floriberto **Ortiz-Rodríguez**<sup>7</sup>

Instituto Politécnico Nacional, MÉXICO

<sup>1</sup> ORCID: 0000-0002-5694-1446 | zhernandezo1500@alumno.ipn.mx

<sup>2</sup> ORCID: 0000-0001-8436-3025 | arosales@ipn.mx

<sup>3</sup> ORCID: 0000-0003-4673-0490 | amirandag1100@alumno.ipn.mx

<sup>5</sup> ORCID: 0000-0003-3772-3651 | pescamillaa@ipn.mx

<sup>6</sup> ORCID: 0000-0001-5237-3050 | jkinani@ipn.mx

<sup>7</sup> ORCID: 0000-0002-2608-8077 | flortiz@ipn.mx

Tecnológico Nacional de México, MÉXICO

<sup>4</sup> ORCID: 0000-0001-8665-4096 | dante.mv@cenidet.tecnm.mx

Recibido 23-03-2023, aceptado 09-05-2023.

### Resumen

Debido al progreso tecnológico en automatización e inteligencia artificial aplicados a la autonomía de vehículos, ha surgido una mayor relevancia en los niveles de asistencia a la conducción autónoma, como lo indica un informe del INEGI que muestra una disminución en el número de muertes en accidentes de tránsito entre 2016 y 2020 [1]. Por esta razón, el Instituto Mexicano de Transporte (*IMT*) se basó en lo que la Sociedad de Ingenieros Automotrices (*SAE, Society of Automotive Engineers*) ha establecido, que es una clasificación de 6 niveles de asistencia a la conducción, que abarca desde la conducción sin automatización hasta la conducción autónoma total.

Este artículo presenta el funcionamiento de un sistema de conducción autónoma en el nivel de asistencia 3, implementado Matlab Simulink, desarrollando un escenario utilizando el motor gráfico UNREAL ENGINE, el cual incluye un entorno realista con peatones, tráfico automovilístico y diferentes diseños de pistas y carreteras. Las pruebas de funcionamiento se llevaron a cabo en este entorno simulado, utilizando un algoritmo detector de objetos llamado You Only Look Once, versión 2, el cual se encarga de detectar automóviles, peatones y señalizaciones haciendo uso de una cámara y un sensor LIDAR (Detección de Luz y Rango) para ampliar el campo de visión de forma artificial.

**Palabras clave:** detección automática de objetos, conducción autónoma, inteligencia artificial, LIDAR, YOLO.

### Abstract

Due to the technological advance in automation and artificial intelligence applied to the autonomy of vehicles, it has caused the levels of assistance for autonomous driving to be more relevant since, according to the INEGI from 2016 to 2020 there is a sustained decrease in the number of deaths in traffic accidents [1]. That is why the Society of Automotive Engineers SAE standardized a classification where 6 levels of driving assistance are defined which includes driving without automation to total autonomous driving. [2] This article shows the operation of an autonomous driving system having an assistance level 3 carried out through Matlab Simulink, a scenario was developed by means of an UNREAL ENGINE graphic engine that had a realistic environment that shows pedestrians and car traffic as well as different track and road layouts. The performance tests were carried out in a simulated environment where an object detector algorithm called You Only Look Once in its version 2 is used. YOLO oversees detecting cars, pedestrians, and signs by means of a camera and a LIDAR sensor. (Light Detection and Range) to extend the field of vision artificially.

**Index terms:** artificial intelligence, autonomous driving, LIDAR, object detection algorithm, YOLO.

## I. INTRODUCCIÓN

Cada día, los vehículos autónomos se vuelven más accesibles para la población, lo que demuestra que nuestro entendimiento sobre ellos se ha vuelto más evidente y la tecnología ha evolucionado al punto en el que podemos convertirnos en pasajeros de nuestros propios vehículos. Según el *IMT* [3], en México existen niveles de conducción autónoma que muestran el alcance en el que un automóvil puede asistir al conductor. Estos niveles van desde la ausencia de automatización hasta la autonomía total del vehículo. En la Tabla 1 se presentan los niveles de clasificación. El análisis del presente trabajo se enfoca en un nivel de conducción autónoma de tipo 3, que se una conducción condicionada en la que la intervención humana se reduce.

TABLA 1  
NIVELES DE ASISTENCIA DE UN VEHÍCULO AUTÓNOMO SAE, IMT [2], [3]

Nivel	Concepto	Descripción
0	Sin automatización	No existe ningún tipo de automatización en el vehículo.
1	Asistencia del conductor	En este nivel ya existe algo de automatización, como lo es el control del volante o dirección.
2	Automatización parcial	El automóvil maneja sistemas como el volante y pedales, pero siempre bajo la supervisión del conductor.
3	Automatización de conducción condicional	En esta categoría entran en funcionamiento la velocidad constante, el asistente a la conducción y el aparcamiento automatizado.
4	Automatización de conducción alta	El automóvil maneja y conduce por sí mismo casi por completo exceptuando algunos entornos con climatología adversa como niebla, nieve y tormentas.
5	Automatización de conducción completa	Este sistema como menciona es una conducción completa, es el último nivel y aquí el conductor es un mero espectador ya que el vehículo hace todo el trabajo

En relación con lo mostrado en la Tabla 1, se ha observado un notable aumento en el uso de asistencia a la conducción en los últimos niveles. Este incremento puede atribuirse directamente a la aplicación de la inteligencia artificial (*IA*, *artificial intelligence*). La *IA* desempeña un papel fundamental en el progreso de los vehículos autónomos, ya que estos automóviles emplean una combinación de sensores, sistemas de percepción y algoritmos de aprendizaje automático para adquirir comprensión del entorno, tomar decisiones y controlar su movimiento de manera autónoma.

Los vehículos autónomos recolectan datos del entorno en el que se encuentran mediante diversas cámaras y sensores LIDAR. Estos dispositivos capturan información sobre objetos, peatones, señales de tráfico, carriles y cualquier otro elemento relevante en la vía. Posteriormente, la *IA* procesa y analiza estos datos con el objetivo de comprender el entorno y tomar decisiones en tiempo real.

#### A. Inteligencia artificial en vehículos autónomos

La implementación de una *IA* en el ámbito de los vehículos autónomos ha dado lugar al desarrollo de algoritmos de percepción que habilitan a estos vehículos para identificar y reconocer de manera precisa y confiable los objetos presentes en su entorno. Estos algoritmos se basan en técnicas de visión por computadora y aprendizaje automático, que les permiten detectar y clasificar una amplia variedad de elementos, incluyendo peatones, vehículos, ciclistas, señales de tráfico y obstáculos.

La visión artificial es parte importante de los vehículos autónomos, ya que les proporciona información sobre su entorno y de acuerdo con el algoritmo que se tenga, toma decisiones adecuadas en tiempo real. Los algoritmos de percepción emplean modelos de aprendizaje automático entrenados con grandes conjuntos de datos que contienen información de diferentes objetos. Estos modelos aprenden patrones y características relevantes, lo que les permite reconocer y comprender el entorno en función de los datos capturados por los sensores del vehículo.

Una de las tareas principales de los algoritmos de percepción es generar una representación precisa del entorno del vehículo. Esto se logra mediante la creación de un modelo gráfico que está en forma de nube de puntos, que captura la posición tridimensional de los objetos detectados en relación con el vehículo. A medida que el vehículo se desplaza, estos algoritmos actualizan constantemente el modelo de percepción para adaptarse a los cambios en el entorno y garantizar una representación actualizada y precisa.

Los algoritmos de percepción se desempeñan en la seguridad y el rendimiento de los vehículos autónomos. Su capacidad para detectar y reconocer objetos con precisión permite que el vehículo tome decisiones informadas, como evitar colisiones, ajustar la velocidad o detenerse en respuesta a las señales de tráfico. A medida que la investigación y el desarrollo en el campo de la IA y los vehículos autónomos continúan avanzando, se espera que los algoritmos de percepción se vuelvan cada vez más sofisticados y precisos, permitiendo una conducción autónoma más segura y eficiente en el futuro.

### B. Aprendizaje automático

El aprendizaje automático (*Machine Learning, ML*) es una disciplina que forma parte de la inteligencia artificial (IA) y se centra en el desarrollo de algoritmos y modelos que permiten a las computadoras aprender y tomar decisiones basadas en datos, sin requerir programación específica para tareas particulares.

El *ML* se fundamenta en la premisa de que las máquinas pueden aprender de manera autónoma a través de la experiencia y mejorar su desempeño en una tarea específica a medida que se les suministra más información. En lugar de seguir instrucciones explícitas, el *ML* permite que las computadoras analicen grandes volúmenes de datos y reconozcan patrones, teniendo la capacidad de realizar predicciones, tomar decisiones y llevar a cabo tareas sin necesidad de una intervención humana directa.

Dentro del *ML*, existen diversas técnicas de aprendizaje, siendo los más comunes el aprendizaje supervisado y el aprendizaje no supervisado. En el aprendizaje supervisado, se suministra a la máquina un conjunto de datos de entrenamiento que están previamente etiquetados o clasificados. La máquina utiliza esta información para aprender a realizar predicciones o clasificaciones cuando se le presentan nuevos datos. En contraste, en el aprendizaje no supervisado, la máquina se le proporciona un conjunto de datos sin etiquetas y su tarea consiste en descubrir patrones o estructuras en los datos de manera autónoma, sin la orientación de categorías predefinidas.

### C. Aprendizaje profundo

El aprendizaje profundo (*Deep Learning, DL*), es una rama del *ML* que se enfoca en el entrenamiento de redes neuronales artificiales profundas para procesar datos y realizar tareas de manera automática. Está inspirado en la estructura y función del cerebro humano, específicamente en las redes neuronales biológicas.

El *DL* se basa en el uso de redes neuronales artificiales compuestas por múltiples capas de nodos interconectados, que se conocen como capas ocultas. Cada capa procesa y extrae características de los datos de entrada de manera progresiva y jerárquica. El entrenamiento de una red neuronal profunda implica la alimentación de datos de entrenamiento a través de la red y la comparación de las salidas generadas por la red

con las salidas deseadas. A través de algoritmos de optimización, como la retropropagación del error, los pesos y las conexiones de la red se ajustan gradualmente para minimizar la discrepancia entre las salidas reales y las esperadas. Este proceso de entrenamiento se repite iterativamente hasta que la red logre un nivel de rendimiento satisfactorio en la tarea específica.

#### D. Redes Neuronales artificiales

Las redes neuronales artificiales (*Artificial Neural Networks, ANN*), se inspiran en las redes neuronales biológicas que se encuentran en el cerebro humano. Estas redes están compuestas por elementos que imitan el comportamiento de las neuronas biológicas en sus funciones más comunes, y están organizados de manera similar a la estructura del cerebro humano.

Desde su inicio, el desarrollo de redes neuronales artificiales ha sido impulsado por la comprensión de que el cerebro humano realiza cálculos de una manera muy diferente a la de las computadoras digitales convencionales. Mientras que las computadoras son lineales y trabajan de manera secuencial, el cerebro es altamente complejo, no lineal y paralelo.

Como se muestra en la Fig.1 las redes neuronales están formadas por un conjunto de nodos conocidos como neuronas que transmiten señales entre sí, las neuronas de la red están a su vez conectadas en capas que forman la red neuronal, en la figura se ilustra una red neuronal de 3 capas que contiene 3 neuronas de entrada, 4 neuronas en la capa oculta, y 2 neuronas de salida [4].

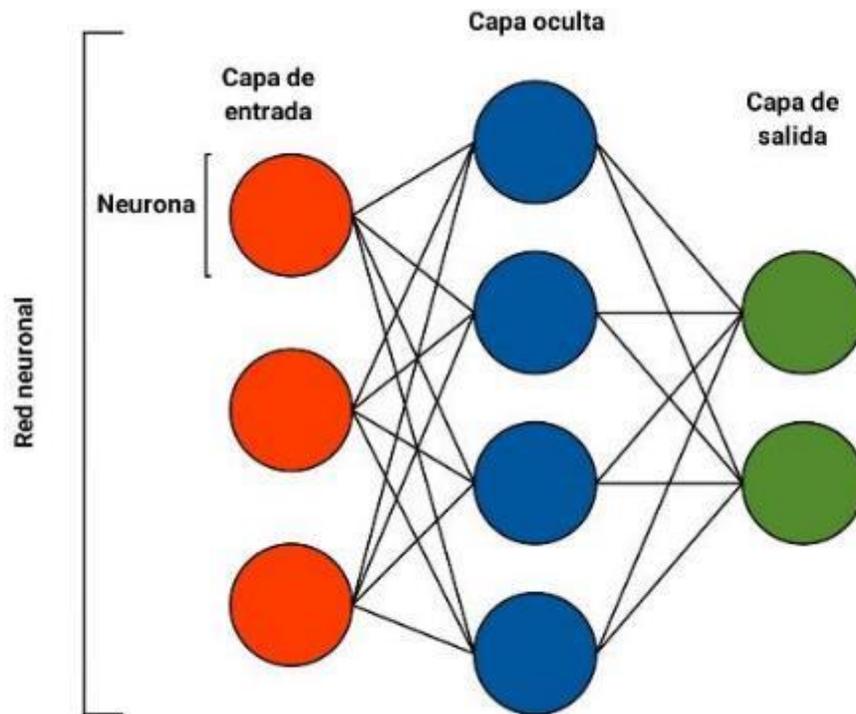


Fig. 1. Esquema general representativa de una red neuronal donde los círculos representan las neuronas y las líneas conexiones entre neuronas.

### E. Detectores de objetos

Los detectores de objetos son una herramienta importante en la visión por computadora y el procesamiento de imágenes. Estos sistemas utilizan algoritmos de aprendizaje profundo para identificar y localizar objetos específicos en una imagen o video. Los detectores de objetos se utilizan en una amplia variedad de aplicaciones, desde el reconocimiento facial, la seguridad en el hogar hasta la detección de objetos en entornos de conducción autónoma. Estos sistemas suelen utilizar redes neuronales convolucionales (CNN, Convolutional Neural Network) para analizar y clasificar características específicas de la imagen, como bordes y formas, y luego utilizar esta información para identificar objetos y su ubicación en la imagen. Los detectores de objetos son una tecnología en constante evolución, y su precisión y capacidad de detección siguen mejorando gracias al uso de técnicas avanzadas de aprendizaje automático y al aumento de la capacidad de procesamiento de los ordenadores.

La mayoría de los algoritmos de detección de objetos emplean diversas arquitecturas de redes neuronales configuradas para abordar tareas específicas, lo que implica la necesidad de un entrenamiento previo. En este artículo, se muestra el como una red neuronal detectora de objetos llamada YOLO (You Only Look Once), la cual ha sido entrenada específicamente para la conducción autónoma [5], [6]. YOLO es uno de los múltiples detectores disponibles, y en su versión 2 utiliza una red de detección de una sola etapa, lo cual le confiere una mayor velocidad en comparación con otros detectores de objetos, como Fast R-CNN y Faster R-CNN ResNet [5]. Estos últimos utilizan un enfoque de aprendizaje profundo en dos etapas. El modelo YOLOv2 ejecuta una CNN [5], [6] en una imagen de entrada para generar predicciones de la red. Posteriormente, el detector de objetos decodifica estas predicciones y genera cuadros delimitadores para el reconocimiento de la imagen.

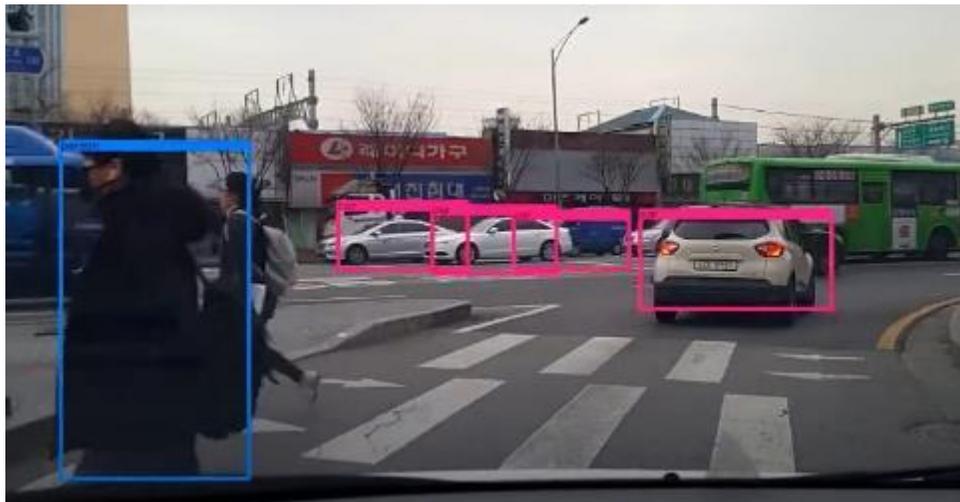


Fig. 2. Ejemplo del funcionamiento de un detector de objetos YOLOv2 en tiempo real. [6]

Como se muestra en la Tabla 2, YOLO usa la arquitectura Darknet-19 [8] que es un tipo de red neuronal convolucional CNN utilizada en el campo de la visión por computadora y el reconocimiento de objetos. Fue desarrollada por Joseph Redmon como parte del proyecto Darknet, que se enfoca en la detección de objetos en imágenes.

Darknet-19 se compone de un total de 19 capas, incluyendo capas de convolución, capas de agrupación (pooling) y capas completamente conectadas. A diferencia de otras arquitecturas populares como VGG o ResNet, Darknet-19 utiliza filtros de tamaño 3×3 en lugar de filtros más grandes, como 7×7 o 11×11. Esto

reduce el número de parámetros y, por lo tanto, disminuye la complejidad computacional de la red. La arquitectura Darknet-19 sigue el patrón general de las *CNN*, donde las capas de convolución se utilizan para extraer características a diferentes niveles de abstracción, y las capas completamente conectadas se utilizan para la clasificación final [5], [8], [9].

- Capas de convolución: Utilizan filtros de  $3 \times 3$  para convolucionar la entrada y generar mapas de características. Estas capas se encargan de extraer características a diferentes niveles de abstracción.
- Capas de agrupación (pooling): Reducen la dimensionalidad de los mapas de características, preservando las características más relevantes. Se utiliza el MaxPooling, que selecciona el valor máximo de una región.
- Capa de salida: Produce las salidas finales de la red, representando las probabilidades de clasificación de diferentes clases de objetos.

TABLA 2  
ARQUITECTURA YOLO [8]

Tipo	Filtro	Tamaño	Salida
Convolutacional	32	$3 \times 3$	$224 \times 224$
Maxpool		$2 \times 2 / 2$	$112 \times 112$
Convolutacional	64	$3 \times 3$	$112 \times 112$
Maxpool		$2 \times 2 / 2$	$56 \times 56$
Convolutacional	128	$3 \times 3$	$56 \times 56$
Convolutacional	64	$1 \times 1$	$56 \times 56$
Convolutacional	128	$3 \times 3$	$56 \times 56$
Maxpool		$2 \times 2 / 2$	$28 \times 28$
Convolutacional	256	$3 \times 3$	$28 \times 28$
Convolutacional	128	$1 \times 1$	$28 \times 28$
Convolutacional	256	$3 \times 3$	$28 \times 28$
Maxpool		$2 \times 2 / 2$	$14 \times 14$
Convolutacional	512	$3 \times 3$	$14 \times 14$
Convolutacional	256	$1 \times 1$	$14 \times 14$
Convolutacional	512	$3 \times 3$	$14 \times 14$
Convolutacional	256	$1 \times 1$	$14 \times 14$
Convolutacional	512	$3 \times 3$	$14 \times 14$
Maxpool		$2 \times 2 / 2$	$7 \times 7$
Convolutacional	1024	$3 \times 3$	$7 \times 7$
Convolutacional	512	$1 \times 1$	$7 \times 7$
Convolutacional	1024	$3 \times 3$	$7 \times 7$
Convolutacional	512	$1 \times 1$	$7 \times 7$
Convolutacional	1024	$3 \times 3$	$7 \times 7$
Convolutacional	1000	$1 \times 1$	$7 \times 7$
Av gpool		Global	1000
Softmax			

#### F. Sensores

En la conducción autónoma se depende de sensores para detectar de objetos como lo son peatones, vehículos entre otros, esto forma una parte fundamental de los vehículos pues es la forma en cómo se tiene una visión artificial del entorno. Los vehículos autónomos usan diferentes sensores, uno de ellos es el Detector de luz y rango (Light Detection And Ranging, *LIDAR*). Un sistema *LIDAR* consiste en un transmisor que genera una

señal, con una potencia y una longitud de onda específica. La luz generada por el transmisor pasa a través de un splitter, un splitter une o divide la señal óptica según sea el caso, en esta primera parte divide la señal, una parte pasa a través de un modulador óptico para transmitirla en el espacio libre y la otra parte se dirige al receptor, donde se utilizará como referencia. Después de cruzar la óptica del receptor, la señal recibida pasa a través de un filtro óptico que suaviza el ruido para el proceso de detección, después del filtro óptico, la señal pasa a través de otro splitter que en este caso une la señal recibida del objetivo con la señal de referencia del transmisor y se entrega al detector óptico, y se convierte en una señal eléctrica para procesar la información como se ilustra en la Fig. 3 [10], [11].

7

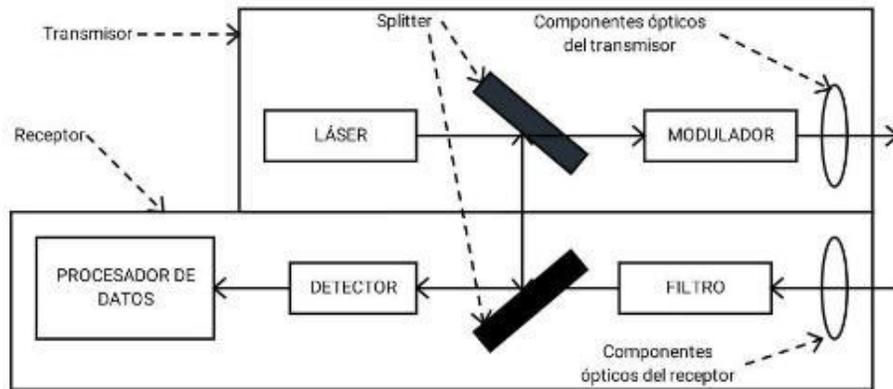


Fig. 3. Diagrama general del sensor LIDAR [10], [11].

## II. IMPLEMENTACIÓN DEL SISTEMA DE CONDUCCIÓN AUTÓNOMA

### A. Diagrama General

En la Fig.4 se ilustra un esquema general que representa el funcionamiento del sistema de conducción autónoma. En dicho diagrama se visualizan todos los bloques que lo conforman, que incluyen los siguiente:

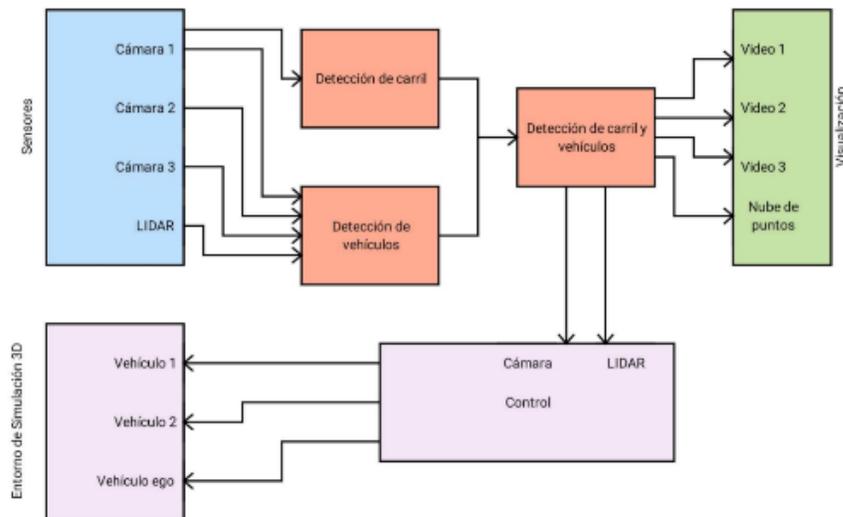


Fig. 4. Diagrama de bloques general para el funcionamiento de un vehículo autónomo.

- Entorno de simulación 3D.
- Sensores simulados.
- Bloques de detección.
- Control.
- Visualización.

### B. Entorno de simulación

∞

Se presenta Unreal Engine [12] como plataforma de desarrollo para cada uno de los entornos, debido a que es uno de los motores gráficos más avanzados disponibles para la creación de videojuegos y aplicaciones interactivas en 3D. La creación de un escenario de simulación en Unreal Engine implica el uso de diversos componentes y recursos, como la configuración de niveles del entorno, la creación de objetos, la programación de eventos y la aplicación de efectos visuales, tal como se muestra en la Fig.5.

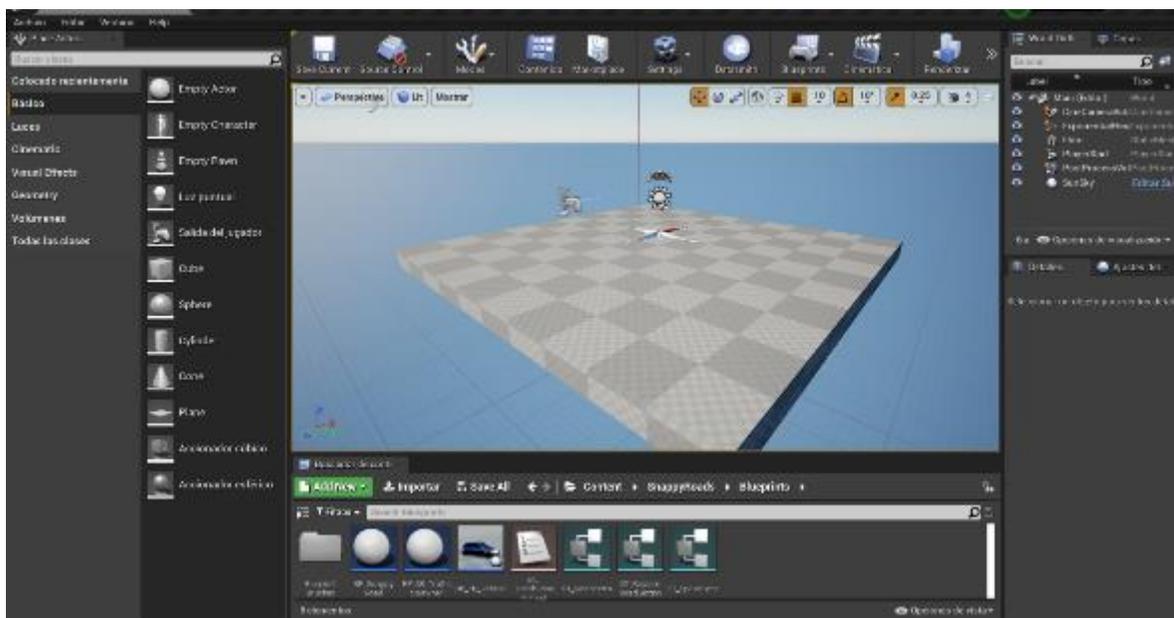


Fig. 5. Diseño en el motor gráfico UNREAL ENGINE

### C. Sensores simulados

Hay varias razones que respaldan el uso de un sensor *LIDAR* [7] [13] simulado en Simulink de MATLAB para un vehículo de conducción autónoma. Una de ellas es la capacidad de evaluar y desarrollar algoritmos de percepción, ya que el sensor *LIDAR* es importante en los sistemas de percepción de vehículos autónomos al proporcionar información tridimensional del entorno mediante la emisión de pulsos de luz. Al simular un sensor *LIDAR* en Simulink, es posible crear y probar algoritmos de percepción en un entorno controlado y reproducible. Esto facilita la evaluación y mejora de los algoritmos de detección y seguimiento de objetos antes de su implementación.

En la Fig.6 se muestra la simulación de una cámara utilizando Simulink en MATLAB para la conducción autónoma. Simular presenta múltiples beneficios, que incluyen la evaluación y desarrollo de algoritmos de visión, la mitigación de riesgos, el control y la reproducibilidad del entorno de prueba, y la flexibilidad en el desarrollo del sistema.

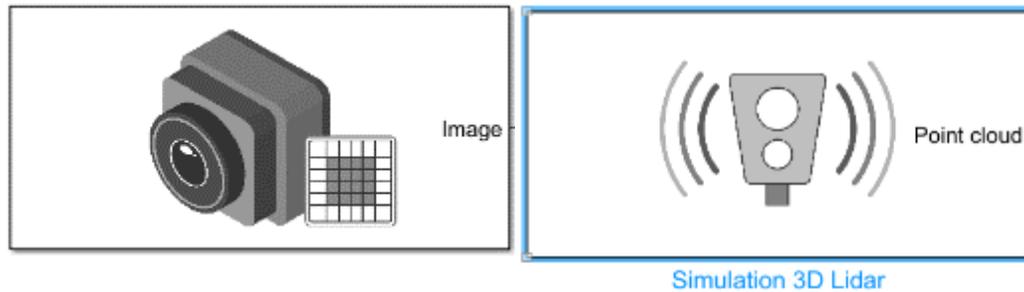


Fig. 6. Bloques del sensor LIDAR y la cámara.

#### D. Bloques de detección

YOLOv2 se utiliza en bloques de Simulink para la detección de vehículos y la detección de líneas debido a su alta precisión, rendimiento en tiempo real, capacidad de detectar múltiples clases de objetos. Su implementación simplificada y la posibilidad de utilizar modelos pre-entrenados donde se usa como base darknet19 que es una arquitectura de *CNN* utilizada en el campo de la visión por computadora y el aprendizaje profundo. En la Fig.7 se muestran los bloques utilizados y en cada uno se implementa YOLO v2.

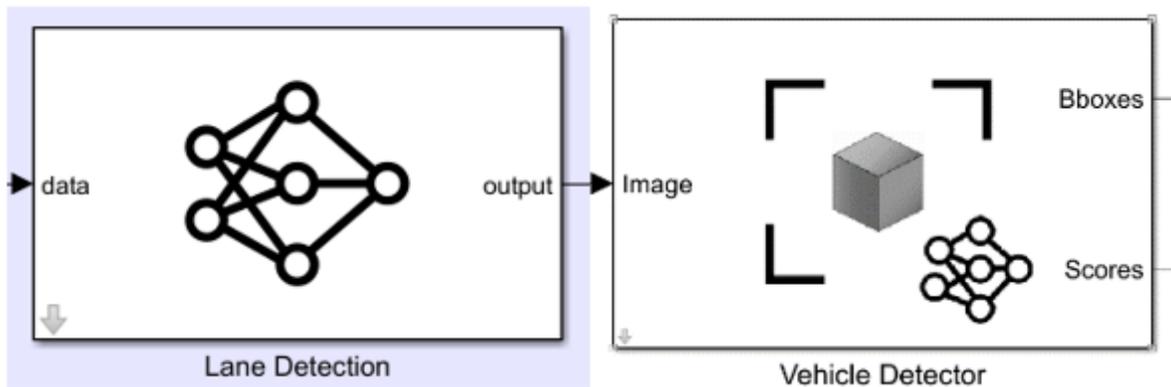


Fig. 7. Diagrama a bloques del funcionamiento de el detector de línea y de carril.

#### E. Control

Para el control, se emplea el modelo cinemático del bloque Ackermann como se muestra en la Fig.8, el cual consiste en un modelo de vehículo similar a un automóvil que utiliza un sistema de dirección Ackermann. Este modelo representa un vehículo con dos ejes separados por una distancia conocida como distancia entre ejes.

El estado del vehículo se define mediante un vector de cuatro elementos:  $[x \ y \ \theta \ \psi]$ . Este vector incluye la posición global del vehículo en términos de coordenadas  $x$  e  $y$ , la dirección del vehículo ( $\theta$ ) y el ángulo de giro ( $\psi$ ). Es importante destacar que la dirección del vehículo y la posición  $x$  e  $y$  se refieren al centro del eje trasero. Los ángulos se expresan en radianes y las posiciones globales en metros. La entrada de dirección para el vehículo se representa como  $d\psi/dt$ , que indica la tasa de cambio del ángulo de giro en radianes por segundo [8]. Esto permite controlar la dirección y el movimiento del vehículo de manera precisa.

El modelo cinemático del bloque Ackermann es ampliamente utilizado en la industria automotriz y en sistemas de control de vehículos autónomos. Proporciona una representación eficaz del comportamiento del vehículo y permite realizar cálculos para el control de su movimiento en tiempo real.

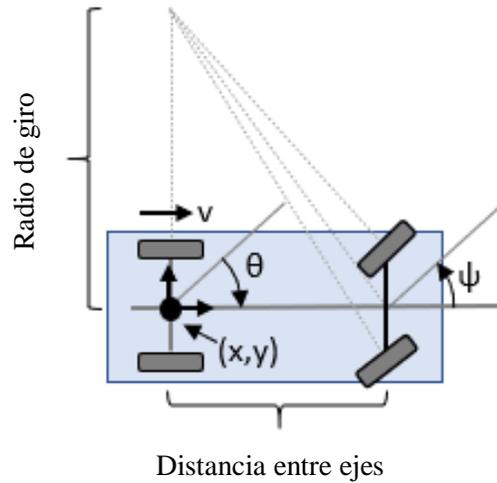


Fig. 8. Modelo cinemático de Ackermann.

Como se muestra en la Fig.9, se ocupa así también un bloque de control crucero adaptativo (*ACC*, Adaptive Cruise Control) que rastrea una velocidad establecida y mantiene una distancia segura de un vehículo líder al ajustar la aceleración longitudinal de un vehículo ego. El bloque calcula las acciones de control óptimas mientras satisface las restricciones de distancia, velocidad y aceleración seguras mediante el control predictivo del modelo *MPC*.

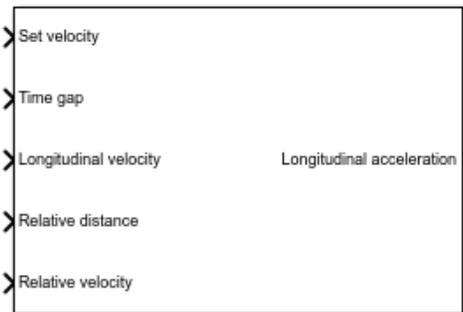


Fig. 9. Bloque del control crucero adaptativo

Un vehículo ego equipado con *ACC* tiene un sensor *LIDAR* que hace un mapeo del entorno y al vehículo precedente en el mismo carril (automóvil líder),  $[D_{rel}]$ . El sensor también mide la velocidad relativa del auto líder,  $[V_{rel}]$ . El sistema *ACC* opera en los siguientes dos modos:

- Control de velocidad: el vehículo ego viaja a una velocidad establecida por el conductor.
- Control de distancia: El vehículo ego mantiene una distancia segura del auto líder.

El sistema *ACC* decide qué modo usar en función de las mediciones del *LIDAR* en tiempo real. Por ejemplo, si el auto líder está demasiado cerca, el sistema *ACC* cambia de control de velocidad a control de espacio. De manera similar, si el auto líder está más lejos, el sistema *ACC* cambia de control de espacio a control de velocidad. En otras palabras, el sistema *ACC* hace que el vehículo ego viaje a la velocidad establecida siempre que mantenga una distancia segura, véase Fig. 10.

Si  $[D_{ref} \geq D_{segura}]$ , entonces el modo de control de velocidad está activo. El objetivo del control es rastrear la velocidad establecida por el controlador,  $[V_{inicial}]$ .

Si  $[D_{ref} < D_{segura}]$ , entonces el modo de control de espaciado está activo. El objetivo del control es mantener la distancia de seguridad,  $[D_{segura}]$ .

11

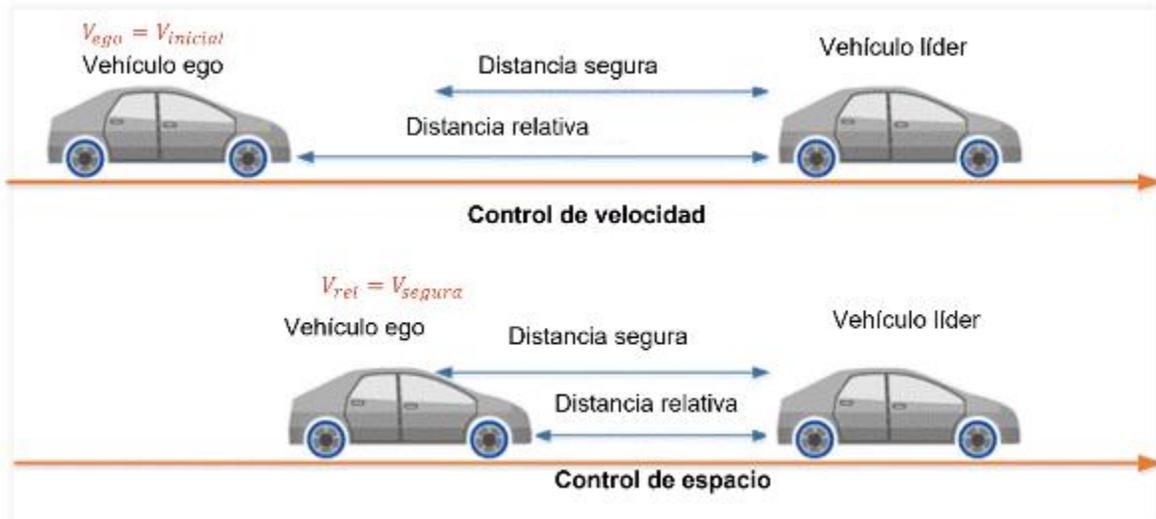


Fig. 10. Imagen gráfica del vehículo líder y el vehículo ego [13][14].

#### F. Visualización

Para la visualización de cada una de las cámaras se simulan 3 bloques de cámara en simulink de MATLAB teniendo una en el techo del vehículo y las otras 2 ubicadas en el lado izquierdo y derecho respectivamente como se muestra en la Fig.11.

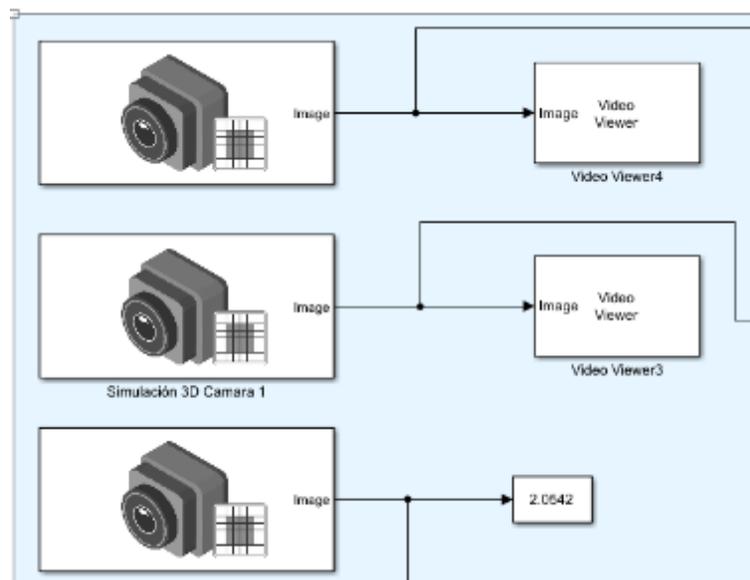


Fig. 11. Diagrama a bloques de las cámaras utilizadas.

### III. RESULTADOS EXPERIMENTALES

#### A. Escenario simulado

El motor gráfico Unreal Engine es un entorno de desarrollo utilizado para crear videojuegos en el cual también ofrece una variedad de herramientas esenciales para la construcción de simulaciones de entornos, en la Fig. 12 se ilustra el escenario realizado, se agregaron señalamientos, también se realizó una pista que contara con al menos 4 carriles para tener un mayor flujo de automóviles y el detector de objetos se encargara de reconocerlos, así también se añadieron diferentes tipos de arquitecturas como lo son edificios y vegetación y para complementar el entorno simulado se agregaron algunos peatones.

12



Fig. 12. Creación del entorno de 4 carriles y señalamientos.

#### B. Aplicación se sistema de conducción autónoma en el entorno gráfico

De acuerdo con el diagrama general en la Fig.4, configurando cada uno de los bloques que conforman el vehículo autónomo en Matlab simulink, y ya hecho el escenario, la siguiente sección lo conforman los bloques haciendo uso de YOLOv2, aquí el sensor LIDAR y la cámara monocular funcionan como la visión artificial del vehículo dándonos como resultado un reconocimiento de vehículos clasificados como se observa en la Fig.13.



Fig. 13. Detección de objetos a través de la cámara con YOLOv2.

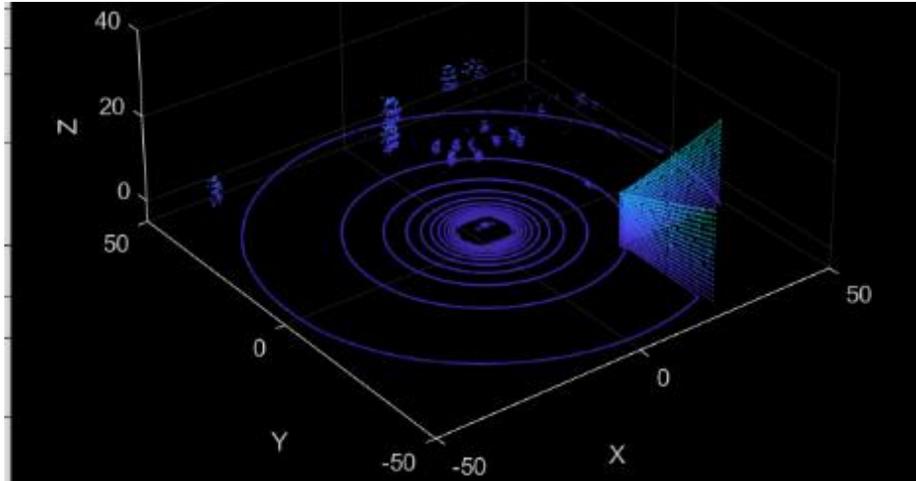


Fig. 14. Nube de puntos generada por medio de un sensor LIDAR en MATLAB.

La Fig.14 muestra una representación tridimensional de un entorno capturado por el sensor LIDAR. La nube de puntos consiste en una gran cantidad de puntos dispersos en el espacio, cada uno de los cuales representa un objeto o una parte del entorno escaneado. Los puntos varían en tamaño y densidad, creando una estructura tridimensional que refleja la geometría del entorno. La Fig. 14 muestra una vista aérea, lo que permite observar la forma y la estructura del entorno escaneado. Es posible identificar objetos tridimensionales, como paredes, árboles o cualquier otro elemento presente en el escaneo.

### C. Control del vehículo

En la Fig.15 se muestra en color amarillo la parte de control del vehículo que, de acuerdo con lo capturado, nuestro control actuara conforme al objeto reconocido. La gráfica de la Fig.15 muestra nuestro vehículo autónomo de color azul y el vehículo detectado de color amarillo, el vehículo amarillo se encuentra a una velocidad constante de alrededor de 20 km/hr mientras nuestro vehículo se encuentra a 50 km/hr eso quiere decir que nos encontramos a una velocidad mayor al otro por lo que nuestro vehículo procede a reducir su velocidad hasta poder mantenerse a una velocidad menor a la del vehículo amarillo y así evitar algún tipo de siniestro.

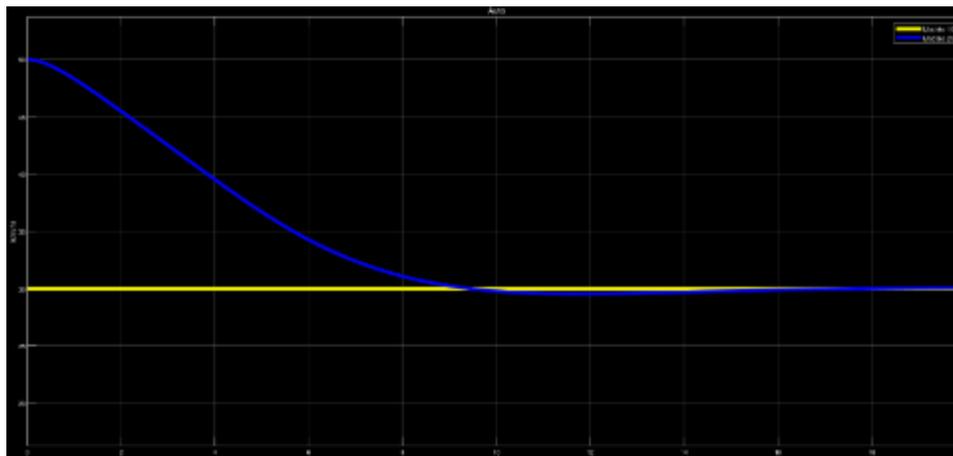


Fig. 15. Gráfica de dos vehículos en movimiento.

#### D. Cambio de carril

En la Fig.16 se muestra el otro resultado en el caso en el que el vehículo ego efectúa un rebase en lugar de reducir la velocidad, el objetivo principal al utilizar el control cinemático de Ackermann es controlar las ruedas delanteras del vehículo de manera que sigan una trayectoria específica mientras se realiza el cambio de carril. Durante un cambio de carril, el vehículo tiene que girar hacia el lado deseado sin desviarse o sobrepasar los límites de la vía. El control cinemático de Ackerman se encarga de calcular los ángulos de dirección necesarios para las ruedas delanteras, asegurando así que el vehículo siga una trayectoria suave y precisa.

14



Fig. 16. Vehículo ego en cambio de carril.

#### IV. CONCLUSIONES

En conclusión, utilizando un motor gráfico como Unreal Engine, se logra crear un entorno simulado de alta fidelidad que proporciona el escenario adecuado para desarrollar y probar sistemas de vehículos autónomos. En este trabajo, se analizó que al seguir el diagrama general y aprovechar las funciones de cada bloque, fue posible implementar con éxito un vehículo autónomo utilizando el algoritmo de detección de objetos YOLOv2 y la visión artificial proporcionada por el sensor LIDAR y las cámaras.

El vehículo autónomo exhibió un comportamiento deseado al evitar colisiones con otros vehículos en movimiento. Esto se logró gracias a la capacidad del vehículo para detectar y reconocer objetos utilizando el YOLOv2 para tomar decisiones en tiempo real. Al detectar la presencia de otro vehículo, el vehículo autónomo redujo su velocidad de manera apropiada, demostrando una respuesta acorde a lo esperado y garantizando la seguridad en el entorno simulado. Otro resultado logrado fue el utilizar el modelo cinemático de Ackermann para efectuar un rebase en lugar de reducir la velocidad

#### AGRADECIMIENTOS

Los autores agradecen al Instituto Politécnico Nacional y al Consejo Nacional de Ciencia y Tecnología de México por el apoyo en la realización de este trabajo de investigación.

## REFERENCIAS

- 15
- [1] Instituto Nacional de Estadística y Geografía (INEGI), "Georreferenciación de accidentes de tránsito en zonas urbanas", 2021, url: [https://www.inegi.org.mx/contenidos/saladeprensa/boletines/2021/accidentes/ACCIDENTES\\_2021.pdf](https://www.inegi.org.mx/contenidos/saladeprensa/boletines/2021/accidentes/ACCIDENTES_2021.pdf)
  - [2] SAE International, "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles J3016\_202104", 2022, url: [https://saemobilus.sae.org/content/j3016\\_202104](https://saemobilus.sae.org/content/j3016_202104)
  - [3] O. Flores, M. Fabela, D. Vázquez, R. Hernández. "Conducción autónoma: Implicaciones", *Publicación bimestral de divulgación externa/ Instituto Mexicano del Transporte*, no. 172, 2018, url: <https://imt.mx/resumen-boletines.html?IdArticulo=462&IdBoletin=172>
  - [4] X. Basogain Olabe, *Redes Neuronales artificiales y sus aplicaciones* (curso), Escuela Superior de Ingeniería de Bilbao, 2008.
  - [5] J. Redmon, A. Farhadi, "YOLO9000: Better, Faster, Stronger", en 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 21–26 de julio de 2017. IEEE, 2017.
  - [6] J. Redmon, S. Divvala, R. Girshick, A. Farhadi. "You only look once: Unified, real-time object detection", arXiv preprint Cornell University, 2015, url: <https://arxiv.org/abs/1506.02640>
  - [7] D. T. Nguyen, T. N. Nguyen, H. Kim, H. J. Lee, "A High-Throughput and Power-Efficient FPGA Implementation of YOLO CNN for Object Detection," in *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, vol. 27, no. 8, pp. 1861-1873, Aug. 2019, doi: <https://doi.org/10.1109/TVLSI.2019.2905242>
  - [8] J. Redmon. Darknet: Open-source neural networks in C, 2016, url: <http://pjreddie.com/darknet/>
  - [9] "YOLO: Real-Time Object Detection". Survival Strategies for the Robot Rebellion. <https://pjreddie.com/darknet/yolov2/> (accedido el 20 de diciembre de 2022).
  - [10] Y. Li, J. Ibanez-Guzman, "Lidar for Autonomous Driving: The Principles, Challenges, and Trends for Automotive Lidar and Perception Systems", *IEEE Signal Processing Magazine*, vol. 37, 2020.
  - [11] A. Bar et al., "The Vulnerability of Semantic Segmentation Networks to Adversarial Attacks in Autonomous Driving: Enhancing Extensive Environment Sensing", *IEEE Signal Processing Magazine*, vol. 38, 2021.
  - [12] "Unreal Engine | The most powerful real-time 3D creation tool". Unreal Engine. <https://www.unrealengine.com/en-US>
  - [13] MathWorks, "Highway Lane Following- MATLAB & Simulink- MathWorks América Latina", 2023.
  - [14] MathWorks, "Detect vehicles in lidar using image labels- MATLAB & simulink- mathworks américa latina", 2023.