ESTIMACIÓN DE LA VELOCIDAD DE NAVEGACIÓN SEGURA PARA VEHÍCULOS AUTÓNOMOS UTILIZANDO TÉCNICAS DE VISIÓN POR COMPUTADORA

RAMIRO SANTIAGO AVILA CHACON

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2025

ESTIMACIÓN DE LA VELOCIDAD DE NAVEGACIÓN SEGURA PARA VEHÍCULOS AUTÓNOMOS UTILIZANDO TÉCNICAS DE VISIÓN POR COMPUTADORA

RAMIRO SANTIAGO AVILA CHACON

Trabajo de Grado para optar al título de
Ingeniero de Sistemas

Director:
Hoover Fabián Rueda Chacón
Ph.D. en Ingeniería Eléctrica y Computación

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2025

DEDICATORIA

Primero a Dios por haberme dado la capacidad, salud y guía para llegar a este momento tan importante de mi formación académica.

A mis padres Alfonso Avila y Aliris Chacón quienes con su amor, esfuerzo y apoyo incondicional me ayudaron a cumplir hoy este sueño en mi vida.

Al semillero de investigación Hands-On Computer Vision.

AGRADECIMIENTOS

A mi padre Alfonso Avila, a mi madre Aliris Chacón, a mi hermanos Camilo Avila Chacón, Andres Avila Avila y mi hermana Zarela Avila Avila.

Al semillero de investigación Hands-On Computer Vision.

A mi director Hoover Rueda-Chacón, por su paciencia, tiempo, compromiso y formación tanto profesional como personal.

CONTENIDO

LISTA DE FIGURAS

pág.

LISTA DE CUADROS

pág.

RESUMEN

TÍTULO: ESTIMACIÓN DE LA VELOCIDAD DE NAVEGACIÓN SEGURA PARA VEHÍCULOS AUTÓNOMOS UTILIZANDO TÉCNICAS DE VISIÓN POR COMPUTADORA [*]

AUTOR: RAMIRO SANTIAGO AVILA CHACON [**]

PALABRAS CLAVE: Visión estéreo, vehículos autónomos, redes neuronales, estimación de profundidad, detección de objetos, velocidad de navegación segura.

DESCRIPCIÓN:

Este trabajo aborda el desarrollo de un simulador para la estimación de la velocidad de navegación segura para vehículos autónomos mediante técnicas recientes de visión por computadora. Se utilizan imágenes adquiridas con un sistema de visión estéreo para calcular la profundidad de los objetos en el entorno del vehículo. La propuesta incluye modelar las maniobras de detención y evasión de obstáculos, determinando la velocidad de navegación segura con el fin de evitar colisiones con los obstáculos detectados a diferentes distancias. Durante el desarrollo del simulador se implementa un método que combina la percepción visual con algoritmos de aprendizaje profundo, como los modelos *Realtime Detection Transformer (RT-DETRv2)* y *You Only Look Once (YOLOv11)* para la detección de objetos y el algoritmo *Neural Markov Random Field (NMRF)* para la estimación de los mapas de disparidad posteriormente convertidos en mapas de profundidad. Se consideran factores críticos como el tiempo de percepción, el tiempo de latencia y las características geométricas del vehículo para calcular la velocidad de navegación. El simulador propuesto se valida como una herramienta útil para evaluar decisiones de navegación en escenarios controlados. Los resultados muestran la velocidad de navegación segura de los vehículos autónomos para operar de forma segura optimizando las decisiones de navegación, los estados de conducción y prevención de colisiones.

---

[*] Trabajo de grado

[**] Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniería de Sistemas e Informática. Director: Hoover Fabián Rueda Chacón.

ABSTRACT

TITLE: SAFE NAVIGATION SPEED ESTIMATION FOR AUTONOMOUS VEHICLES USING COMPUTER VISION TECHNIQUES [*]

AUTHOR: RAMIRO SANTIAGO AVILA CHACON [**]

KEYWORDS: Stereo vision, autonomous vehicles, neural networks, depth estimation, object detection, safe navigation speed.

DESCRIPTION:

This work addresses the development of a simulator for estimating the safe navigation speed for autonomous vehicles using recent computer vision techniques. Images acquired with a stereo vision system are used to calculate the depth of objects in the vehicle's environment. The proposal includes modeling stopping and obstacle avoidance maneuvers, and determining the safe navigation speed in order to avoid collisions with obstacles detected at different distances. During the development of the simulator, a method is implemented that combines visual perception with deep learning algorithms, such as the *Realtime Detection Transformer (RT-DETRv2)* and *You Only Look Once (YOLOv11)* models for object detection and the *Neural Markov Random Field (NMRF)* algorithm for estimating disparity maps subsequently converted into depth maps. Critical factors such as perception time, latency time, and vehicle geometric characteristics are considered to calculate navigation speed. The proposed simulator is validated as a useful tool for evaluating navigation decisions in controlled scenarios. The results show the safe navigation speed of autonomous vehicles for safe operation by optimizing navigation decisions, driving states, and collision avoidance.

---

[*] Bachelor's Thesis

[**] Faculty of Physical-Mechanical Engineering. School of Systems Engineering & Informatics. Advisor: Hoover Fabián Rueda Chacón.

INTRODUCCIÓN

Los vehículos autónomos han revolucionado el panorama del transporte por su capacidad de operar de manera independiente y segura en diversas condiciones de tráfico y entornos[1]. Estos vehículos utilizan sistemas avanzados de percepción, control y toma de decisiones, basados en tecnologías como la inteligencia artificial y la visión por computadora[2] para navegar sin intervención humana directa. Esta autonomía requiere detectar obstáculos, interpretar señales de tráfico y planificar rutas de manera eficiente, contribuyendo no solo a la seguridad vial, sino también a la eficiencia y comodidad del transporte moderno[3].

La visión por computadora abarca aplicaciones como la clasificación de objetos, la detección de obstáculos y la restauración de imágenes[4]. En el contexto de vehículos autónomos, la capacidad de clasificación es vital para reconocer peatones y otros vehículos[5]. Además, la detección de objetos es fundamental para evitar colisiones y navegar de manera segura[6]. Los avances recientes en algoritmos han mejorado

---

1 Alonzo Kelly y Anthony (Tony) Stentz. «Rough Terrain Autonomous Mobility - Part 1: A Theoretical Analysis of Requirements». En: *Autonomous Robots* 5 (1998), págs. 129 -161.

2 Athanasios Voulodimos et al. «Deep Learning for Computer Vision: A Brief Review». En: *Intell. Neuroscience* 2018 (2018). DOI: 10.1155/2018/7068349.

3 Ryan De Iaco, Stephen L. Smith y Krzysztof Czarnecki. «Universally Safe Swerve Maneuvers for Autonomous Driving». En: *IEEE Open Journal of Intelligent Transportation Systems*. Vol. 2. 2021, págs. 482-494. DOI: 10.1109/OJITS.2021.3138953.

4 Richard Szeliski. «Computer vision: algorithms and applications». En: *Springer Nature*. 2022.

5 Jia Deng et al. «Imagenet: A large-scale hierarchical image database». En: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. Ieee. 2009, págs. 248-255.

6 Mehdi Masmoudi et al. «Object Detection Learning Techniques for Autonomous Vehicle Applications». En: *2019 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*. 2019, págs. 1-5. DOI: 10.1109/ICVES.2019.8906437.

la precisión y velocidad de la detección, haciendo posible su implementación en entornos dinámicos y complejos[7]. Esto permite a los vehículos autónomos identificar y responder rápidamente a obstáculos y otros elementos en la carretera[8].

La visión estéreo es otra técnica esencial en la visión por computadora para vehículos autónomos ya que permite estimar la profundidad del entorno[9]. Esto es importante para la navegación en terrenos irregulares y la detección de obstáculos a diferentes distancias. Los estudios han demostrado que el mapeo de terrenos basado en visión estéreo mejora significativamente la capacidad de los vehículos para navegar en entornos complejos[10].

El modelado de las maniobras de detención y evasión son aspectos críticos para la velocidad de navegación segura de los vehículos autónomos. Se han propuesto modelos basados en cámaras de enfoque variable para definir distancias de seguridad dinámicas según las condiciones del entorno[11]. Esto permite a los vehículos ajustar su velocidad en tiempo real, garantizando una operación segura en diferentes escenarios de tráfico. A pesar de estos avances, los vehículos autónomos enfrentan desafíos significativos en la percepción y toma de decisiones. La detección de obs-

---

7 Nicolas Carion et al. «End-to-End Object Detection with Transformers». En: *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I*. Berlin, Heidelberg: Springer-Verlag, 2020, págs. 213-229. DOI: 10.1007/978-3-030-58452-8_13.

8 Jingyun Liang et al. «Swinir: Image restoration using swin transformer». En: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, págs. 1833-1844.

9 Rui Fan et al. «Computer stereo vision for autonomous driving: Theory and algorithms». En: *Recent Advances in Computer Vision Applications Using Parallel Processing*. Springer, 2023, págs. 41-70.

10 Arturo Rankin, Andres Huertas y Larry Matthies. «Stereo-vision-based terrain mapping for off-road autonomous navigation». En: *Proc. SPIE 7332, Unmanned Systems Technology XI*. SPIE, 2009, pág. 733210. DOI: 10.1117/12.819099.

11 Min-Joong Kim et al. «On the Development of Autonomous Vehicle Safety Distance by an RSS Model Based on a Variable Focus Function Camera». En: *Sensors*. Vol. 21. 20. MDPI, 2021, pág. 6733. DOI: 10.3390/s21206733.

táculos en entornos complejos, la identificación precisa de objetos en condiciones de iluminación adversas y la estimación precisa de la profundidad son áreas que requieren investigación continua[12].

La elección y configuración de los sensores también juegan un papel fundamental en la capacidad de un vehículo autónomo para operar a velocidades seguras. Estudios sobre sensores adecuados para operaciones a alta velocidad han destacado el uso de cámaras, sistemas LiDAR y radares[13].

Para desarrollar y entrenar sistemas de visión por computadora para vehículos autónomos, se requieren conjuntos de datos extensos y variados[14]. Conjuntos de datos como *KITTI Stereo 2015*[15] y *DrivingStereo*[16] han proporcionado una base sólida para el entrenamiento de modelos de aprendizaje profundo en tareas de clasificación y detección de objetos[17]. El uso de técnicas avanzadas de aprendizaje, como

---

[12] Larry H. Matthies y Pierrick Grandjean. «Stochastic performance, modeling and evaluation of obstacle detectability with imaging range sensors». En: *IEEE Transactions on Robotics and Automation* 10.6 (1994), págs. 783-792. DOI: 10.1109/70.338533.

[13] Iaco, Smith y Czarnecki, ver n. 3; Anne Schneider et al. «Sensor study for high speed autonomous operations». En: *Proc. SPIE 9494, Next-Generation Robotics II; and Machine Intelligence and Bio-inspired Computation: Theory and Applications IX*. SPIE, 2015, pág. 949408. DOI: 10.1117/12.2176596.

[14] German Ros et al. «The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes». En: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, págs. 3234-3243.

[15] Moritz Menze y Andreas Geiger. «Object scene flow for autonomous vehicles». En: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, págs. 3061-3070.

[16] Guorun Yang et al. «DrivingStereo: A Large-Scale Dataset for Stereo Matching in Autonomous Driving Scenarios». En: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[17] Yiyi Liao, Jun Xie y Andreas Geiger. «KITTI-360: A Novel Dataset and Benchmarks for Urban Scene Understanding in 2D and 3D». En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45 (2021), págs. 3292-3310; Ming-Fang Chang et al. «Argoverse: 3D Tracking and Forecasting With Rich Maps». En: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, págs. 8740-8749. DOI: 10.1109/CVPR.2019.00895.

redes neuronales convolucionales (*CNN*) y transformadores de visión, ha mejorado significativamente la capacidad de los sistemas de visión por computadora para interpretar imágenes complejas[18].

En este proyecto se propone estudiar la velocidad de navegación segura para vehículos autónomos en términos de maniobras de detención y evasión de obstáculos detectados por medio de algoritmos de visión por computadora en imágenes RGB adquiridas por un sistema de visión estéreo. La capacidad de estos vehículos para interpretar y reaccionar ante su entorno de manera autónoma depende crucialmente de la precisión y velocidad de estos algoritmos[19].

Para abordar esta propuesta, el presente trabajo se enfoca en el desarrollo de un simulador interactivo que permite estimar la velocidad de navegación segura para vehículos autónomos, tomando en cuenta maniobras de detención y evasión de obstáculos detectados mediante algoritmos de visión por computadora. El simulador integra un modelo de procesamiento de imágenes estéreo para estimar el mapa de disparidad y, posteriormente, calcular el mapa de profundidad. A continuación, se implementan modelos de detección de objetos y medición de distancias desde el sistema óptico hasta los objetos detectados. El cálculo de la velocidad de navegación segura se realiza teniendo en cuenta los parámetros del entorno físico, del sistema óptico y de la geometría del vehículo, con el fin de obtener la distancia de anticipación necesaria para frenar o evadir los obstáculos. Además, se considera la medición de los ángulos de visión, el campo de visión instantáneo y la altura de los objetos.

---

[18] Vincent Dumoulin y Francesco Visin. «A guide to convolution arithmetic for deep learning». En: *arXiv preprint arXiv:1603.07285* (2016); Ashish Vaswani et al. «Attention is All you Need». En: *Advances in Neural Information Processing Systems*. Ed. por I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017.

[19] Rankin, Huertas y Matthies, ver n. 10; Matthies y Grandjean, ver n. 12.

Adicionalmente, el simulador incorpora una interfaz gráfica interactiva que permite al usuario visualizar los resultados del procesamiento de imágenes y ajustar los parámetros de percepción visual, las condiciones del entorno y la geometría del vehículo. Esta plataforma modular y dinámica ofrece una herramienta novedosa para la investigación y validación de algoritmos de visión por computadora aplicados a vehículos autónomos, permitiendo explorar y optimizar la velocidad de navegación segura bajo diversas condiciones de tráfico y entorno.

1. OBJETIVOS

Objetivo general

Desarrollar un simulador que estime la velocidad de navegación segura para vehículos autónomos en términos de maniobras de detenimiento y evasión de obstáculos detectados mediante algoritmos de visión por computadora en imágenes RGB adquiridas por un sistema de visión estéreo.

Objetivos específicos

1. Modelar matemáticamente la distancia que existe entre un obstáculo y el sistema de visión en términos de las propiedades de desaceleración del vehículo en maniobras de detenimiento y evasión.

2. Analizar las características de percepción visual de un sistema de visión estéreo respecto a la adquisición de imágenes de obstáculos en términos de las propiedades de los lentes y los sensores empleados.

3. Comparar las capacidades de detección de obstáculos de los algoritmos de detección de objetos basados en aprendizaje profundo para múltiples distancias a la que se encuentre el sistema de visión estéreo.

4. Simular la velocidad segura de navegación de un vehículo autónomo respecto a la capacidad de detección de obstáculos y su distancia hasta el sistema de visión estéreo.

2. MARCO DE REFERENCIA

2.1. Modelamiento matemático de maniobras de conducción

La distancia que existe entre un obstáculo y el observador en términos del tiempo de reacción del observador y las propiedades de desaceleración del vehículo es un aspecto crítico para la seguridad en la navegación autónoma. El tiempo de reacción incluye el procesamiento de la información visual por parte del sistema de visión por computadora y la respuesta del sistema de control del vehículo[20].

La distancia de frenado se ve afectada por muchos factores, incluyendo las condiciones de los frenos y neumáticos, las condiciones de la carretera, por ejemplo, húmeda o seca, pavimentada o sin pavimentar, y la pendiente de la carretera[21]. Los cálculos que siguen asumen carreteras pavimentadas. Particularmente se analizarán dos maniobras: la de detención y la de evasión de un obstáculo, como se observa en la Figura 1.

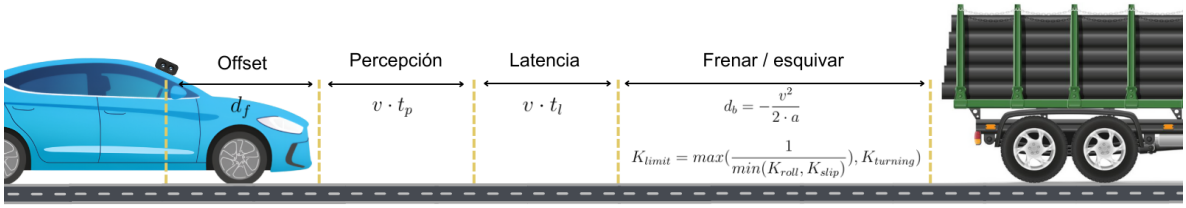


Figura 1. Distancias de seguridad a considerar para maniobras de frenado y evasión de obstáculos en función de la velocidad y el tiempo de respuesta. Elaboración propia.

- **Maniobra de detención:** Un esquema de la maniobra de detención se ilustra

[20] Aleksander Rydzewski y Pawel Czarnul. «Human awareness versus Autonomous Vehicles view: comparison of reaction times during emergencies». En: *2021 IEEE Intelligent Vehicles Symposium (IV)*. 2021, págs. 732-739. DOI: 10.1109/IV48863.2021.9575602.

[21] Schneider et al., ver n. 13.

en la Figura 2 (izquierda). Matemáticamente, la ecuación cinemática base que se utilizará para determinar la distancia recorrida durante la desaceleración está dada por la Ecuación (1),

$$v_f^2 = v_i^2 + 2 \cdot a \cdot d_b, \tag{1}$$

donde v_f es la velocidad final, v_i es la velocidad inicial, a es la desaceleración, y d_b es la distancia recorrida. La distancia recorrida durante el periodo de desaceleración puede ser calculada como

$$d_b = \frac{-v_i^2}{2 \cdot a}, \tag{2}$$

donde el término de aceleración será negativo, para obtener una distancia de frenado positiva[22].

Sin embargo, esta distancia no tiene en cuenta el tiempo requerido para procesar los datos adquiridos por el sistema de visión, es decir, determinar si hay un obstáculo obstruyendo la ruta, las latencias mecánicas/software que impiden que el vehículo frene instantáneamente, o la distancia mínima de seguridad entre el sistema de visión y la parte frontal del vehículo[23]. La Ecuación (3) combina estas características con la cinemática del vehículo, para obtener d como la distancia de anticipación de frenado:

$$d = d_f + v_i \cdot t_p + v_i \cdot t_l + \frac{-v_i^2}{2 \cdot a}. \tag{3}$$

donde d_f es la distancia entre el sensor y la parte delantera del vehículo (*off-*

---

[22] Schneider et al., ver n. 13.

[23] Schneider et al., ver n. 13.

set), t_p es el tiempo requerido para el procesamiento de datos y t_l es la latencia que se refiere al tiempo de respuesta física del vehículo para aplicar los frenos.

- **Maniobra de evasión:** Un esquema de la maniobra de evasión se muestra en la Figura 2 (derecha) y matemáticamente se representa con el límite general de curvatura para cualquier velocidad, dada por la Ecuación (4),

$$K_{limit} = \max\left(\frac{1}{\min(K_{roll}, K_{slip})}, K_{turning}\right), \tag{4}$$

donde K_{roll} es el límite de curvatura por criterios de vuelco, para evitar que el vehículo se vuelque al tomar una curva, K_{slip} es el límite de curvatura por fricción de los neumáticos y $K_{turning}$ es el ángulo de giro del vehículo[24].

En ese orden, el límite de curvatura por criterios de vuelco, se define como,

$$K_{roll} = \frac{g \cdot w_h}{COG_h \cdot v^2}, \tag{5}$$

donde g es la gravedad, w_h es la distancia horizontal entre los centros de las ruedas delanteras y traseras, COG_h es la altura del centro de gravedad y v es la velocidad del vehículo[25].

La siguiente limitación para la curvatura proviene de la fuerza de deslizamiento, que se determina por la capacidad de los neumáticos para adherirse al camino sin deslizarse. Teniendo en cuenta el coeficiente de fricción μ entre los neumáticos y la superficie de la carretera, K_{slip} se representa como,

$$K_{slip} = \frac{\mu \cdot g}{v^2}. \tag{6}$$

---

[24] Schneider et al., ver n. 13.

[25] Schneider et al., ver n. 13.

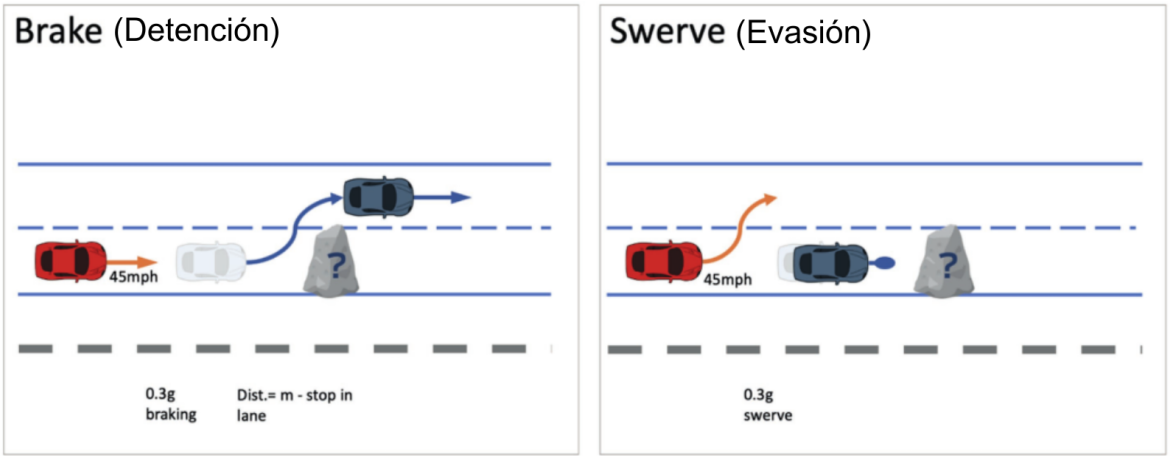


Figura 2. Maniobra de conducción de detención (*brake*) y evasión (*swerve*) para un vehículo (rojo) que viaja a 45 mph utilizando una desaceleración de 0.3g para ambas maniobras. Adaptada de Nicholas Britten et al. «Do you trust me? Driver responses to automated evasive maneuvers». En: *Frontiers in Psychology* 14 (2023). DOI: 10.3389/fpsyg.2023.1128590.

Una vez obtenemos el limite general de curvatura, así como los tiempos de percepción, latencia y la distancia mínima de seguridad entre el sistema de visión y la parte frontal del vehículo, se combinan con las características de la cinemática del vehículo, para obtener la distancia de anticipación de evasión, por medio de,

$$d_k = d_f + v_i \cdot t_p + v_i \cdot t_l + K_{limit}, \tag{7}$$

donde d_f es la distancia del offset, t_p es el tiempo de procesamiento de datos, t_l es la latencia, y K_{limit} es el límite de curvatura general.

2.2. Características de un sistema de visión

En los sistemas de visión por computadora para vehículos autónomos, varias características juegan un papel relevante en la adquisición y el procesamiento de las

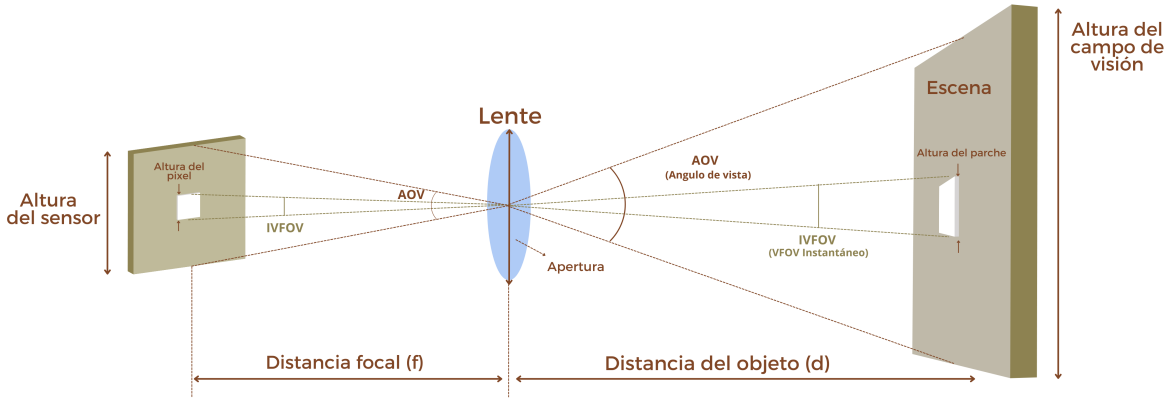


Figura 3. Representación gráfica de las propiedades de visión del sistema de percepción. En esta representación se incluye la distancia focal (f), el ángulo de visión (AOV), el campo de visión instantáneo (IFOV) y el campo de visión vertical instantáneo (*IVFOV*). Se muestran las relaciones entre la altura del sensor, el lente, la altura del parche de la escena, la distancia al objeto (d) y la altura del campo de visión en la escena. Elaboración propia.

imágenes, necesarias para la navegación segura y la detección de objetos[26]. Estas características incluyen el angulo de visión (*AOV*) y la resolución del sistema, así como las propiedades de los lentes y de los sensores empleados, como se ilustra en la Figura 3.

El AOV determina el ángulo de visión del área visible captada por el sistema de cámaras. Un AOV amplio permite cubrir una mayor área del entorno, crucial para la detección temprana de obstáculos y la planificación de rutas seguras[27]. Sin embargo existe un *trade-off* entre el AOV y la resolución de los objetos en los sensores: a mayor AOV, los objetos ocupan menos pixeles, es decir aparecen con menor re-

---

[26] Shahian-Jahromi Babak et al. «Control of autonomous ground vehicles: a brief technical review». En: *IOP Conference Series: Materials Science and Engineering* 224.1 (2017), pág. 012029. DOI: 10.1088/1757-899X/224/1/012029.

[27] Kui Yuan et al. «Field of View and Its Design for the Autonomous Driving System of Tram». En: *2021 International Conference on Information Control, Electrical Engineering and Rail Transit (ICEERT)*. 2021, págs. 236-239. DOI: 10.1109/ICEERT53919.2021.00052.

solución; y viceversa, a menor AOV mayor resolución de los objetos[28]. Además, un AOV amplio puede introducir distorsiones en la imagen y reducir la resolución en los bordes, lo cual debe ser gestionado cuidadosamente[29].

La resolución de las cámaras afecta directamente la capacidad del sistema para detectar y reconocer objetos con precisión. Una mayor resolución permite captar más detalles, mejorando la identificación de objetos pequeños o lejanos, aunque también aumenta los requerimientos de procesamiento y almacenamiento de datos. Encontrar un balance adecuado entre la resolución y la capacidad de procesamiento es esencial para mantener un rendimiento óptimo[30]. En este mismo camino, el campo de visión instantáneo (*IFOV*) es utilizado y se define como la extensión angular en el espacio del objeto que ocupa un píxel en el detector y refleja la capacidad de resolución del sistema de formación de imágenes[31].

Las propiedades de los lentes, como la distancia focal[32], la apertura y la resolución angular[33], influyen en la calidad y características de las imágenes captadas. La distancia focal determina el grado de ampliación y el AOV de la cámara. Un lente con una distancia focal corta proporciona un AOV amplio, útil para la visión general del

---

[28] Shahryar Afzal, Jiasi Chen y K. K. Ramakrishnan. «Viewing the 360° Future: Trade-Off Between User Field-of-View Prediction, Network Bandwidth, and Delay». En: *2020 29th International Conference on Computer Communications and Networks (ICCCN)*. 2020, págs. 1-11. DOI: 10.1109/ICCCN49398.2020.9209659.

[29] Fan et al., ver n. 9.

[30] Xiangyuan Zhu et al. «Cross View Capture for Stereo Image Super-Resolution». En: *IEEE Transactions on Multimedia* 24 (2022), págs. 3074-3086. DOI: 10.1109/TMM.2021.3092571.

[31] Benqi Zhang et al. «Simultaneous improvement of field-of-view and resolution in an imaging optical system». En: *Opt. Express* 29.6 (2021), págs. 9346-9362. DOI: 10.1364/OE.420222.

[32] Alonzo Kelly y Anthony Stentz. «Rough Terrain Autonomous Mobility - Part 2: An Active Vision, Predictive Control Approach». En: *Auton. Rob.* 5 (jun. de 2000). DOI: 10.1023/A:1008822205706.

[33] Kelly y Stentz, «Rough Terrain Autonomous Mobility - Part 1: A Theoretical Analysis of Requirements», ver n. 1.

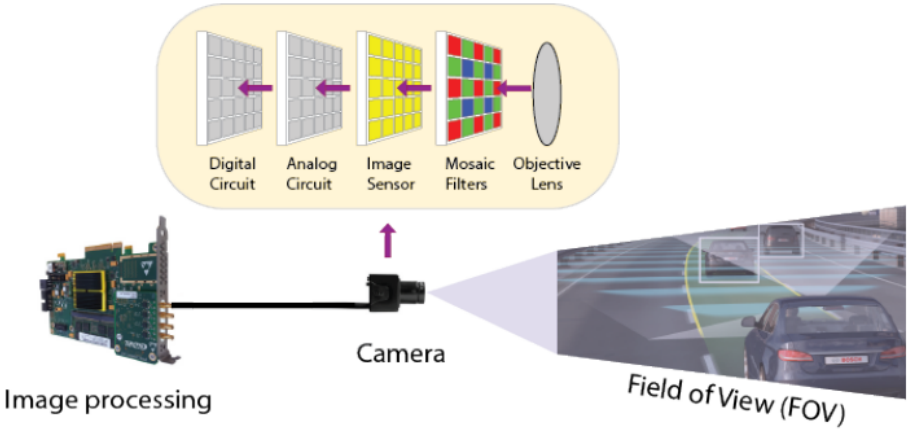


Figura 4. Componentes de los sistemas de visión. Se incluye el lente objetivo, filtros mosaico, sensor de imagen y circuitería, que representan el proceso de adquisición y procesamiento de imágenes. Adaptada de[36].

entorno, mientras que un lente con una distancia focal larga es adecuado para la detección de detalles a larga distancia[34]. La apertura del lente controla la cantidad de luz que entra en la cámara, afectando la claridad y el brillo de las imágenes. Un ajuste adecuado de estas propiedades es importante para optimizar la adquisición de imágenes en diversas condiciones de iluminación y distancia[35].

Los sensores de las cámaras convierten la luz captada por los lentes en señales electrónicas que pueden ser procesadas por el sistema de visión, como se ilustra en la Figura 3. Las propiedades de los sensores, como la sensibilidad a la luz, la relación señal-ruido, la velocidad de adquisición y su número de píxeles, determinan la calidad y la fiabilidad de las imágenes. Sensores con alta sensibilidad y baja relación señal-ruido son preferibles para entornos con condiciones de iluminación

---

[34] Abdelmoghit Zaarane et al. «Distance measurement system for autonomous vehicles using stereo camera». En: *Array* 5 (2020), pág. 100016. DOI: https://doi.org/10.1016/j.array.2020.100016.

[35] Zaarane et al., ver n. 34.

variables, como la conducción nocturna o en túneles[37]. Además, la capacidad de los sensores para captar imágenes a alta velocidad es determinante para la detección y el seguimiento de objetos[38].

La configuración efectiva de estas características es fundamental para el rendimiento general del sistema de visión por computadora en vehículos autónomos. Un sistema bien diseñado debe equilibrar el AOV, IFOV y las propiedades de los lentes y sensores.

2.3. Sistema de visión estéreo para estimación de profundidad

Las imágenes estéreo son un componente crucial en los sistemas de visión por computadora utilizados en vehículos autónomos[39]. Este enfoque implica el uso de dos o más cámaras para adquirir imágenes de la misma escena desde diferentes ángulos y representar modelos geométricos[40], como se ilustra en la Figura 5. Al analizar las diferencias entre estas imágenes, es posible calcular la profundidad y crear un mapa tridimensional del entorno[41].

---

[37] Narsimlu Kemsaram, Anweshan Das y Gijs Dubbelman. «A Stereo Perception Framework for Autonomous Vehicles». En: *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. Antwerp, Belgium, 2020, págs. 1-6. DOI: 10.1109/VTC2020-Spring48590.2020.9128899.

[38] Kelly y Stentz, «Rough Terrain Autonomous Mobility - Part 1: A Theoretical Analysis of Requirements», ver n. 1; Kemsaram, Das y Dubbelman, «A Stereo Perception Framework for Autonomous Vehicles», ver n. 37.

[39] Matteo Poggi et al. «On the synergies between machine learning and binocular stereo for depth estimation from images: a survey». En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.9 (2021), págs. 5314-5334.

[40] Narsimlu Kemsaram, Anweshan Das y Gijs Dubbelman. «Architecture Design and Development of an On-board Stereo Vision System for Cooperative Automated Vehicles». En: *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. 2020, págs. 1-8. DOI: 10.1109/ITSC45102.2020.9294435.

[41] Matthies y Grandjean, ver n. 12; Gabe Sibley, Larry Matthies y Gaurav Sukhatme. «Bias Reduction and Filter Convergence for Long Range Stereo». En: *Robotics Research*. Ed. por Sebastian

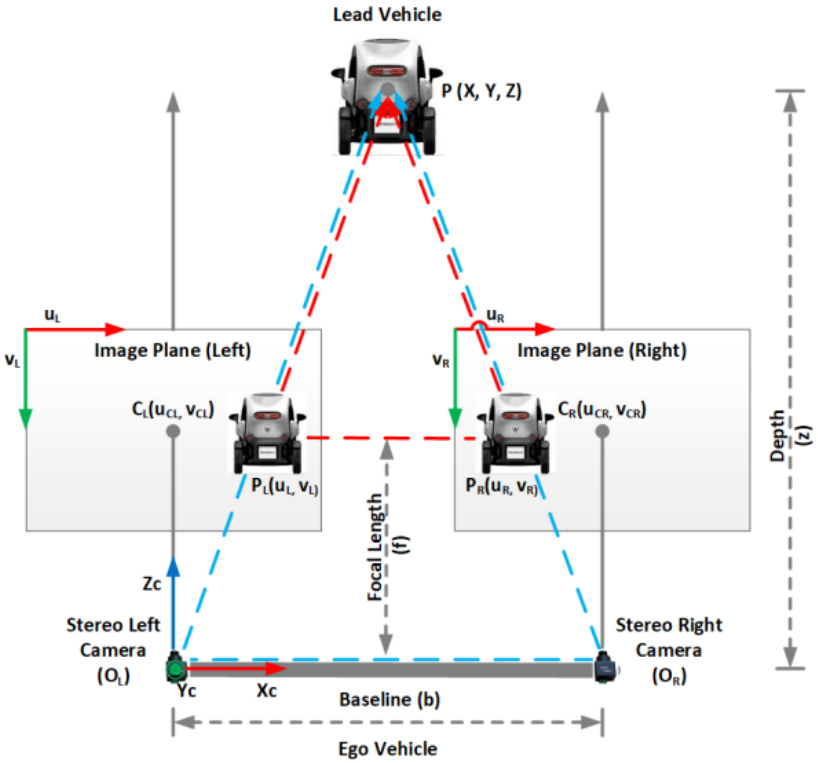


Figura 5. Modelo geométrico de visión estéreo para la estimación de la profundidad. El sistema de coordenadas de la cámara (X_c, Y_c, Z_c) es relativo al centro de proyección de la cámara izquierda. El *baseline* (b) es la distancia entre ambas cámaras. El vehículo utiliza cámaras estéreo izquierda y derecha para capturar imágenes del vehículo delantero, permitiendo así estimar su profundidad y posición relativa. Adaptada de[42].

El sistema de visión estéreo utiliza los parámetros de calibración de la cámara adquiridos del archivo de configuración del equipo, para transformar y rectificar los pares de imágenes estéreo de entrada, como se ilustra en la Figura 6. Estas imágenes se utilizan para calcular el mapa de disparidad. El mapa de disparidad se concibe en la correspondencia estéreo al realizar la correlación de píxeles entre las imágenes, que es la diferencia en la posición de un objeto en las dos imágenes. Esta disparidad se convierte en información de profundidad utilizando las propiedades geométricas del

Thrun, Rodney Brooks y Hugh Durrant-Whyte. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, págs. 285-294.

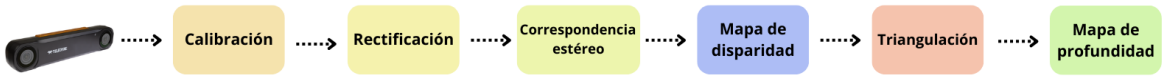


Figura 6. *Pipeline* de visión estéreo por etapas del proceso. Calibración, rectificación, correspondencia estéreo, cálculo del mapa de disparidad, triangulación y generación del mapa de profundidad. Elaboración propia.

sistema de cámaras y técnicas de triangulación[43]. Matemáticamente, y apoyados en la Figura 5, la disparidad se puede calcular como,

$$\delta = u_l - u_r, \tag{8}$$

donde u_l es la coordenada del píxel en la imagen izquierda, y u_r es la coordenada del píxel correspondiente en la imagen derecha. Esta disparidad se utiliza posteriormente para calcular la profundidad de los objetos en la escena, mediante la Ecuación (9)

$$z = \frac{f \cdot b}{\delta}, \tag{9}$$

donde z es la profundidad, f es la distancia focal de las cámaras y b es la distancia entre las dos cámaras (*baseline*).

La Figura 7 muestra las imágenes adquiridas con un sistema estéreo y el mapa de disparidad correspondiente. En la parte superior están las imágenes originales capturadas por un sistema de visión estéreo en una intersección de una ciudad, mientras que en la parte inferior, el mapa de disparidad coloreado[44].

---

[43] Fan et al., ver n. 9.

[44] Yun Xie, Shaowu Zheng y Weihua Li. «Feature-Guided Spatial Attention Upsampling for Real-Time Stereo Matching Network». En: *IEEE MultiMedia* 28.01 (2021), págs. 38-47. DOI: 10.1109/MMUL.2020.3030027.

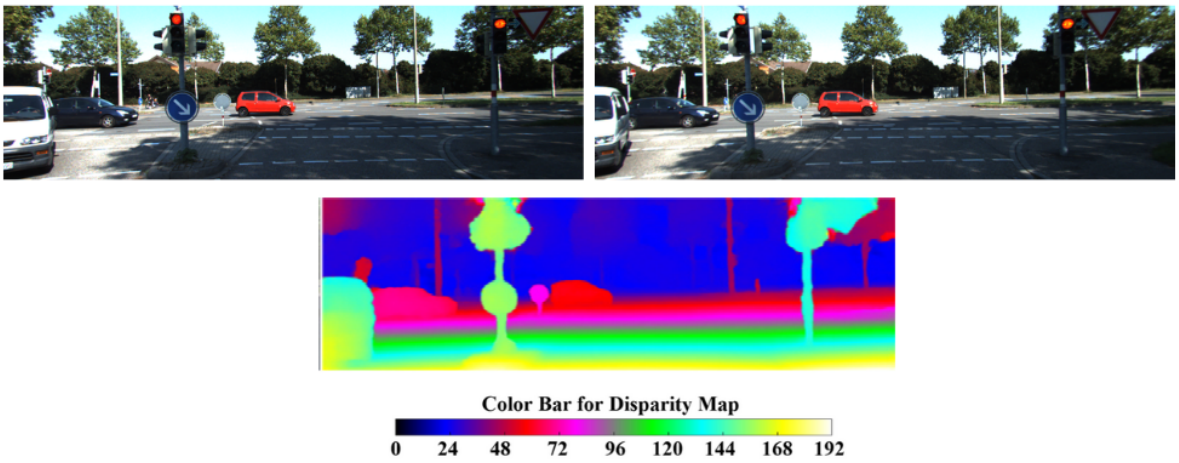


Figura 7. Imágenes adquiridas con un sistema de visión estéreo y su correspondiente mapa de disparidad. Adaptada de[45].

2.4. Algoritmos de detección de objetos

La detección precisa de objetos es fundamental para la navegación segura en vehículos autónomos ya que permite identificar peatones, coches, camiones, bicicletas, señales de tráfico y obstáculos, tal como se representa en la Figura 8. Los modelos basados en redes neuronales profundas[46] han mejorado significativamente la eficiencia y precisión en la detección de objetos al integrar técnicas avanzadas de procesamiento de imágenes y aprendizaje profundo, en particular la arquitectura *YOLO* (*You Only Look Once*)[47].

La capacidad de estos modelos para detectar diferentes tipos de objetos en escenas complejas se debe a su entrenamiento en grandes conjuntos de datos como,

---

[46] Carion et al., ver n. 7.

[47] Juan Terven, Diana-Margarita Córdova-Esparza y Julio-Alejandro Romero-González. «A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS». En: *Machine Learning and Knowledge Extraction* 5.4 (nov. de 2023), 1680–1716. DOI: 10.3390/make5040083.

por ejemplo, *KITTI Object Detection*[48]. Esto les permite generalizar bien a nuevas situaciones y mantener un alto rendimiento incluso en condiciones desafiantes como iluminación variable u oclusiones parciales[49].

La integración de técnicas avanzadas con módulos de atención[50] y transformadores de visión[51] también permite al sistema enfocarse en las partes más relevantes de la imagen. Especialmente el modelo *Detection Transformer* (*DETR*)[52], es una arquitectura que consta de tres componentes principales: una red neuronal convolucional *ResNet*[53] para la extracción de características, un transformador codificador-decodificador y una red de retroalimentación (*FFN*)[54] para las predicciones.

- ***RT-DETRv2:*** Un *transformer* de detección de objetos en tiempo real mejorado, se basa en el reciente *RT-DETR*[56], representado en la Figura 9. RT-DETRv2[57] sugiere establecer un número de puntos de muestreo para características a

---

48 Andreas Geiger, Philip Lenz y Raquel Urtasun. «Are we ready for autonomous driving? The KITTI vision benchmark suite». En: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 2012, págs. 3354-3361. DOI: 10.1109/CVPR.2012.6248074.

49 Shane Gilroy, Edward Jones y Martin Glavin. «Overcoming Occlusion in the Automotive Environment—A Review». En: *IEEE Transactions on Intelligent Transportation Systems* 22.1 (2021), págs. 23-35. DOI: 10.1109/TITS.2019.2956813.

50 Vaswani et al., ver n. 18.

51 Alexey Dosovitskiy et al. «An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale». En: *arXiv preprint arXiv:2010.11929* (2021). eprint: 2010.11929 (cs.CV).

52 Carion et al., ver n. 7.

53 Kaiming He et al. «Deep residual learning for image recognition». En: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, págs. 770-778.

54 Vaswani et al., ver n. 18.

56 Wenyu Lv et al. *RT-DETRv2: Improved Baseline with Bag-of-Freebies for Real-Time Detection Transformer*. 2024. arXiv: 2407.17140 [cs.CV].

57 Lv et al., ver n. 56.

Figura 8. Resultados de la detección de vehículos en diversos entornos urbanos y climáticos utilizando el modelo *DETR*. Las imágenes representan la efectividad del modelo en la identificación precisa de diferentes tipos de vehículos bajo varias condiciones. Adaptada de[55].

diferentes escalas dentro del módulo de atención deformable para lograr una extracción selectiva de características multiescala por parte del decodificador. Proporciona un operador de muestreo discreto opcional para reemplazar al operador `grid_sample` que es especifico de los DETR, eliminando así las restricciones de implementación típicamente asociados con los transformadores de detección DETR.

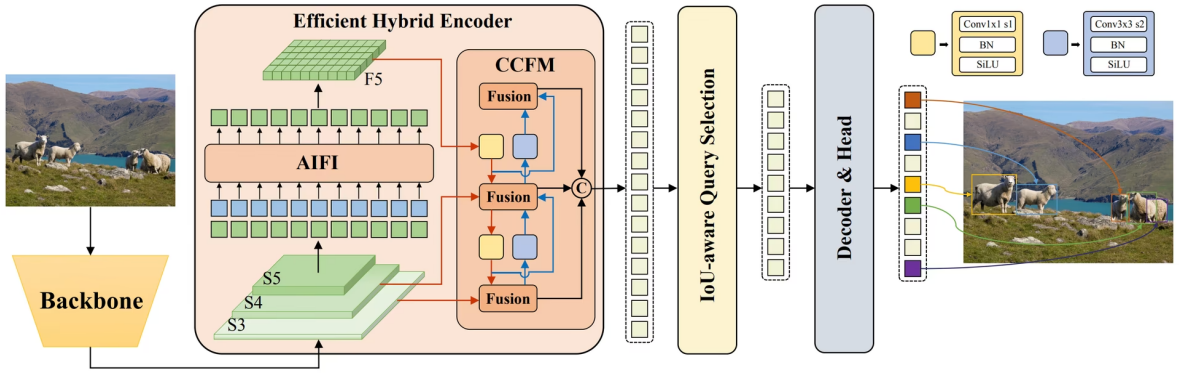


Figura 9. Arquitectura del modelo RT-DETR. Incorpora un eficaz codificador híbrido que transforma las características multiescala en una secuencia de características de imagen mediante la interacción de características intraescala (AIFI) y el módulo de fusión de características de escala cruzada (CCFM). La selección de consultas basada en IoU se emplea para seleccionar un número fijo de características de imagen que sirvan como consultas de objeto iniciales para el decodificador. Adaptada de[58].

- ***YOLOv11:*** Es uno de los modelos recientes de la familia YOLO para la detección de objetos, diseñado para mejorar el rendimiento en la extracción de características y la precisión en la detección[59]. La arquitectura de YOLO se basa en tres componentes fundamentales, como se muestra en la Figura 10: primero, la columna vertebral (*backbone*), que actúa como el extractor de características principal, utilizando redes neuronales convolucionales para transformar a la imagen en mapas de características de múltiples escalas. En segundo lugar, el cuello (*neck*) funciona como una etapa de procesamiento intermedia, agregando y refinando las representaciones de características mediante capas especializadas que permiten integrar información a diferentes escalas. Finalmente, la cabeza (*head*) es responsable de la predicción, generando los resultados finales de localización y detección de objetos a partir de los mapas de características procesados[60].

  Sobre esta arquitectura consolidada, YOLOv11[61] introduce varias innovaciones clave que optimizan su rendimiento, como el bloque *C3k2* (*Cross-Stage Partial*) con un tamaño de kernel de 2 mejora la eficiencia en la extracción de características; el módulo *SPPF* (*Spatial Pyramid Pooling Fast*), que permite una agregación de características más rápida y eficiente y el bloque *C2PSA* (*Convolutional Block with Parallel Spatial Attention*), que refuerza la atención espacial paralela para una mejor precisión en la detección[62].

---

[59] Rahima Khanam y Muhammad Hussain. *YOLOv11: An Overview of the Key Architectural Enhancements*. 2024. arXiv: 2410.17725 [cs.CV].

[60] Terven, Córdova-Esparza y Romero-González, ver n. 47.

[61] Khanam y Hussain, ver n. 59.

[62] Khanam y Hussain, ver n. 59.

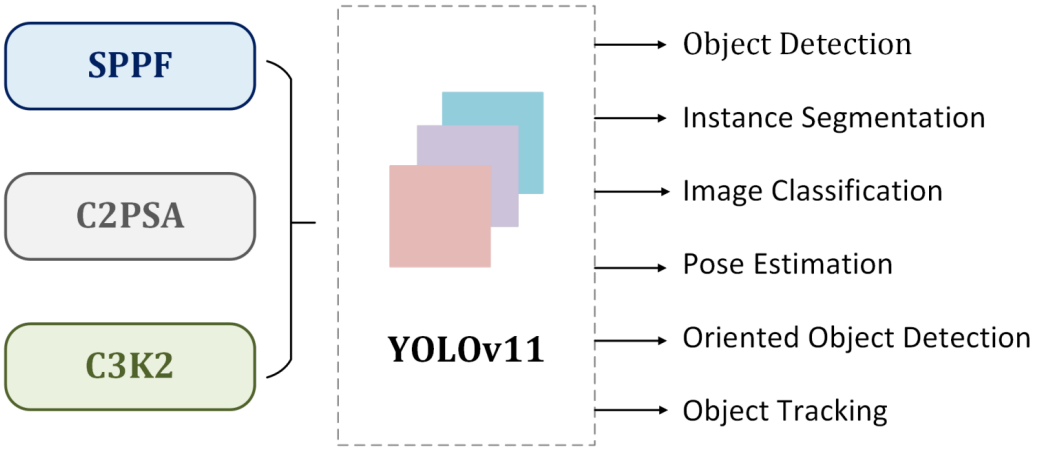


Figura 10. Arquitectura del modelo YOLOv11. Incorpora tres componentes principales: *SPPF (Spatial Pyramid Pooling Fast)*, *C2PSA (Cross Stage Partial with Parallel Spatial Attention)*, y *C3K2 (Cross Stage Partial with Kernel Size 2)*. Adaptada de[63].

2.5. Algoritmo de estimación de la disparidad

El algoritmo *Neural Markov Random Field for Stereo Matching*[64] aborda las limitaciones de los modelos *Markov Random Field (MRF)*[65] diseñados manualmente para el emparejamiento estéreo. Como se ilustra en la Figura 11, el objetivo del modelo es mejorar la precisión de la correspondencia estéreo, el proceso comienza con la etapa de extracción de características locales, donde un par de imágenes estéreo se procesan a través de una CNN. Esta red extrae características visuales en diferentes escalas, generando representaciones a resoluciones de 1/8 y 1/4 del tamaño original de la imagen. Con las características extraídas, sigue la etapa de *Disparity Proposal Network (DPN)*. En esta fase, se identifican los k valores de disparidad

---

[64] Tongfan Guan, Chen Wang y Yun-Hui Liu. «Neural Markov Random Field for Stereo Matching». En: *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun. de 2024, 5459–5469. DOI: 10.1109/cvpr52733.2024.00522.

[65] Stan Z Li. *Markov random field modeling in computer vision*. Springer Science & Business Media, 2012.

más probables para cada píxel, conocidos como modales de disparidad. Estas propuestas iniciales de disparidad se distribuyen a lo largo de la imagen y representan posibles valores de disparidad, pero de forma dispersa, ya que cada píxel tiene múltiples opciones candidatas antes de ser refinadas. A continuación, se refinan estos modales mediante N_p pasos de mensajes neuronales, lo que permite capturar coherencia espacial y mejorar la calidad de las propuestas iniciales. Seguidamente, se realiza la inferencia a través del modelo (Neural MRF). Este modelo factoriza la imagen de un grafo probabilístico donde cada nodo representa un valor de disparidad candidato, y las conexiones modelan relaciones entre etiquetas dentro del mismo píxel (*self-edges*) y entre píxeles vecinos (*neighbor-edges*). Mediante una estrategia de paso de mensajes basada en atención y aprendida de manera neuronal, se refinan las distribuciones de probabilidad asociadas a cada candidato, considerando tanto la evidencia local como la compatibilidad con los vecinos. Las predicciones se decodifican en valores de disparidad mediante una estrategia *winner-takes-all*, seleccionando para cada píxel el valor de disparidad más probable según la información inferida.

Finalmente, para mejorar la precisión y reducir errores en regiones criticas como los bordes de los objetos y áreas con poca textura, se incorpora una etapa de refinamiento de disparidad (*Disparity Refinement*). Esta fase toma la estimación de disparidad obtenida a nivel de resolución más baja y la ajusta utilizando características de mayor resolución[66].

---

[66] Guan, Wang y Liu, ver n. 64.

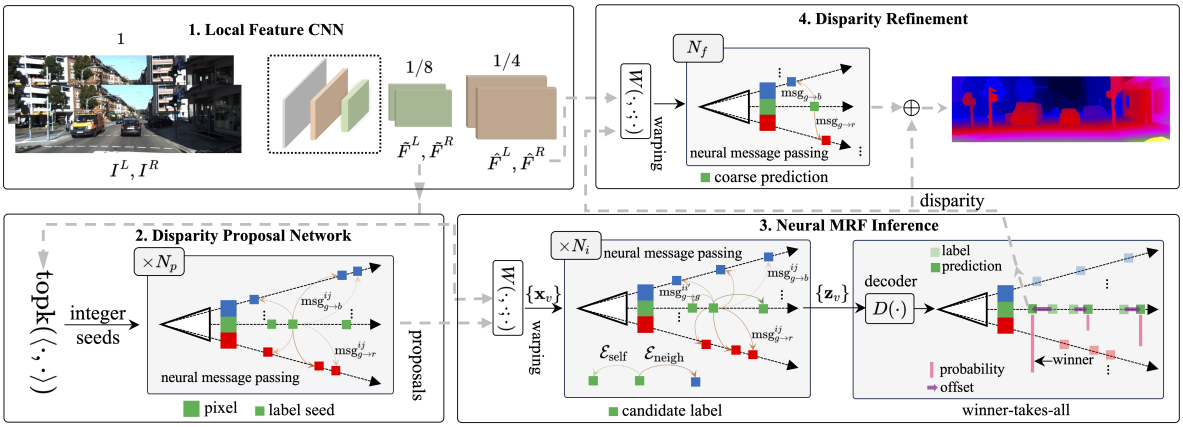


Figura 11. Arquitectura del modelo NMRF. El proceso incluye: 1) Extracción de características mediante una CNN. 2) Propuesta de disparidades candidatas por píxel con la DPN. 3) Inferencia basada en un modelo MRF neuronal. 4) Refinamiento de la disparidad a mayor resolución. Adaptada de[67].

3. MÉTODO PROPUESTO

El enfoque propuesto para estimar la velocidad de navegación segura se desarrolla en tres etapas principales: (i) procesamiento de las imágenes estéreo para estimar el mapa de profundidad, (ii) detección de objetos y medición de las distancias desde el sistema óptico hasta los objetos detectados, y (iii) cálculo de la velocidad segura y la distancia de anticipación para frenar o evadir obstáculos, dependiendo de los parámetros del sistema óptico y del vehículo en función de los objetos seleccionados. Para facilitar la interacción y visualización de los resultados durante estas etapas,

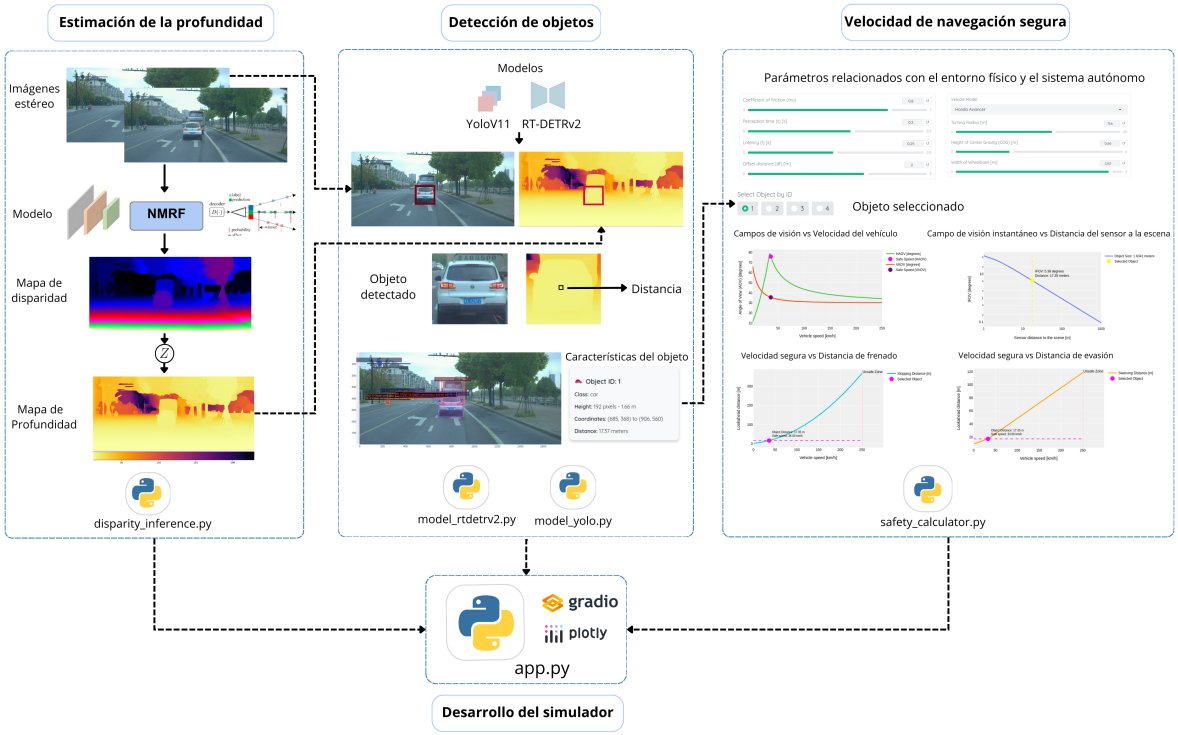


Figura 12. Método propuesto para el desarrollo de un simulador modular. Con *scripts* de Python para inferencia, carga de modelos y cálculo de la velocidad de navegación segura. Elaboración propia.

se emplea la librería Gradio[68] para crear interfaces interactivas, y Plotly[69] para la representación gráfica de los datos obtenidos.

En la Figura 12 se presenta el proceso que comienza con la inferencia de disparidad mediante el script `disparity_inference.py`, encargado de procesar imágenes adquiridas por un sistema de visión estéreo para generar mapas de profundidad que permiten estimar la distancia a los objetos, en metros. A continuación, tanto el script `model_rt-detrv2.py` como el `model_yolo.py` se encargan de la detección de objetos en la escena, proporcionando información sobre la presencia de obstáculos, su altura y ubicación en relación con el sistema óptico. Finalmente, el script `safety_calculator.py` calcula la velocidad de navegación segura y la distancia de anticipación para frenar o evadir obstáculos. Todo este proceso esta conectado al script principal `app.py`, que integra los módulos y facilita la interacción del usuario a través de interfaces interactivas.

3.1. Etapa de procesamiento de imágenes estéreo

Esta etapa se implementa mediante el script `disparity_inference.py` implementando el algoritmo NMRF descrito en la sección 2.5 para la estimación de disparidad. El script acepta pares de imágenes estéreo y opera en los conjuntos de datos KITTI Stereo 2015 y DrivingStereo. El proceso comienza cargando el modelo preentrenado con pesos ajustados en el conjunto de datos KITTI y su configuración asociada, que define parámetros críticos como la disparidad máxima evaluable. Durante la inferencia, el modelo procesa las imágenes estéreo rectificadas y genera mapas de

---

[68] Abubakar Abid et al. «Gradio: Hassle-free sharing and testing of ml models in the wild». En: *arXiv preprint arXiv:1906.02569* (2019).

[69] Elias Dabbas. *Interactive Dashboards and Data Apps with Plotly and Dash: Harness the power of a fully fledged frontend web framework in Python–no JavaScript required*. Packt Publishing Ltd, 2021.

disparidad densos. Posteriormente, se aplica la relación matemática entre la disparidad y la profundidad usando la Ecuación (9).

3.2. Etapa de detección de objetos

La etapa de detección y análisis de objetos se implementa mediante un enfoque flexible que permite elegir entre los modelos RT-DETRv2 y YOLOv11, descritos en la sección 2.4. El proceso se inicia con la carga del modelo seleccionado. En el caso de RT-DETRv2, esto se realiza a través de `model_rt-detrv2.py`, donde se utiliza una arquitectura *transformer end-to-end* para la detección de objetos.
Por otro lado, el modelo YOLOv11 se gestiona desde el script `model_yolo.py`, que implementa una red convolucional altamente optimizada entrenada con el conjunto de datos *KITTI Object Detection Evaluation 2012*[70], en partición del 80 % para entrenamiento y del 20 % para prueba.
Independientemente del modelo seleccionado, el sistema procesa la imagen estéreo izquierda original junto con el mapa de profundidad previamente generado. En el caso de RT-DETRv2, se consideran únicamente los objetos detectados con una confianza superior al 80 %[71]. Para cada objeto identificado, se calculan sus coordenadas espaciales mediante cajas delimitadoras y se extrae la información de profundidad correspondiente de la región del mapa de profundidad. La distancia asociada a cada objeto se calcula tomando la región del mapa de profundidad correspondiente al *bounding box* del objeto detectado. En esta región, se consideran únicamente los valores de profundidad válidos mayores que cero y se utiliza la mediana de estos valores para obtener una medida robusta frente a posibles valores atípicos.

---

[70] Geiger, Lenz y Urtasun, ver n. 48.

[71] Simon Wenkel et al. «Confidence Score: The Forgotten Dimension of Object Detection Performance Evaluation». En: *Sensors* 21.13 (2021). DOI: 10.3390/s21134350.

Los resultados se visualizan usando la librería Plotly, generando gráficos interactivos que muestran cada objeto detectado con su identificador único, clase, dimensiones en píxeles y distancia estimada en metros.

3.3. Etapa de cálculo de la velocidad de navegación segura

En la fase final del simulador, se implementa un análisis paramétrico multivariable para determinar las condiciones óptimas de navegación segura en la logica del script `safety_calculator.py`. Este análisis considera parámetros críticos como el coeficiente de fricción (μ) entre neumáticos y superficie, la latencia del sistema (t_l), el tiempo de percepción (t_p) del sistema de visión, así como parámetros geométricos del vehículo, tales como la distancia de offset (d_f), la altura del centro de gravedad COG_h, la distancia horizontal entre los centros de las ruedas delanteras y traseras w_h y el ángulo de giro del vehículo $K_{turning}$.
La simulación produce un panel gráfico con 4 visualizaciones interactivas esenciales para el análisis de navegación segura:

1. **Ángulo de visión (*AOV*)**: Es el ángulo que define el campo visual completo que puede capturar el sistema óptico. Se mide en radianes o grados. Se representa como,

$$AOV = 2\tan^{-1}\left(\frac{\sqrt{A}}{2f}\right) = \frac{\sqrt{A}}{f}, \tag{10}$$

   donde A es el área del pixel y f es la distancia focal. El ángulo de visión se descompone en el ángulo de visión horizontal *(HAOV)* y vertical *(VAOV)* relacionados con la velocidad a la que se desplaza el vehículo, como se puede observar en la Figura 13.

   - **Ángulo de visión horizontal (HAOV)**: Determinado por el requisito de

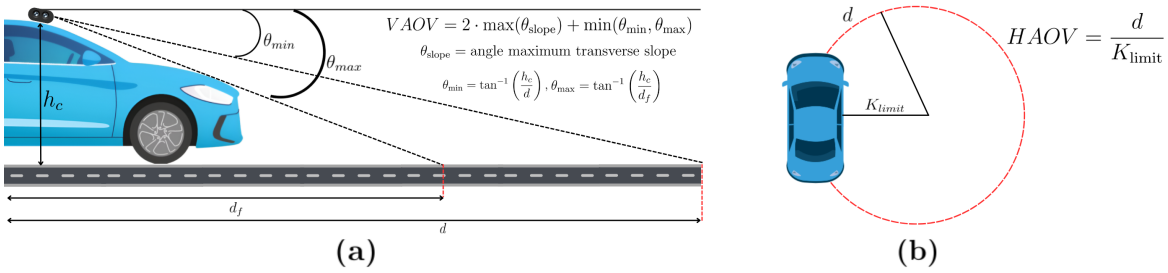


Figura 13. Representación de los ángulos de visión VAOV y HAOV. En (a) el ángulo de visión vertical VAOV muestra el campo de visión en el plano vertical, definido por los ángulos mínimo (θ_{min}), máximo (θ_{max}) y el ángulo pendiente transversal máximo (θ_{slope}) junto con la altura del sistema de visión desde el suelo h_c, la distancia de frenado d y el offset d_f. En (b) el ángulo de visión horizontal HAOV ilustra la cobertura lateral del vehículo, representando el limite de curvatura general y la distancia de frenado d. Elaboración propia.

ver en la dirección en la que se va a girar, se expresa como,

$$HAOV = \frac{d}{K_{limit}} \tag{11}$$

donde d es la distancia de anticipación para frenar y K_{limit} es el límite de curvatura general, representados en la Figura 1.

- **Ángulo de visión vertical (VAOV)**: Se define como el ángulo entre las líneas de visión superior e inferior que limitan el campo observable en dirección vertical, se expresa como,

$$VAOV = 2 \cdot \max(\theta_{slope}) + \min(\theta_{min}, \theta_{max}) \quad \begin{cases} \theta_{min} = \tan^{-1}\left(\frac{h_c}{d}\right) \\ \theta_{max} = \tan^{-1}\left(\frac{h_c}{d_f}\right) \end{cases} \tag{12}$$

donde θ_{min} es el ángulo formado entre el horizonte a la altura del sistema de visión y la posición del obstáculo, θ_{max} es el ángulo formado entre el horizonte a la altura del sistema de visión y la posición del *offset* (como parte delantera del vehículo); h_c es la altura a la que se encuentra el sistema de visión desde el suelo, como se ilustra en la Figura 13.

2. **Campo de visión instantáneo (IFOV)**: Es el ángulo subtendido por un solo píxel del sensor en un parche de la escena, se evalúa en función de la distancia y la altura del objeto, enfocándose en obstáculos, como se ilustra en la Figura 14, uno que va desde el sistema de visión hasta el suelo y otro desde el sistema de visión hasta la parte superior de un obstáculo. Matemáticamente, se expresa como:

$$IFOV = \tan^{-1}\left(\frac{h_c}{d}\right) - \tan^{-1}\left(\frac{h_c - h_p}{d}\right), \tag{13}$$

donde h_c es la altura del sistema de visión hasta el suelo, d es la distancia de anticipación y h_p es la altura del objeto. Para obtener la altura del objeto es necesario emplear la distancia de muestreo de tierra (*GSD*, por sus siglas en inglés *Ground Sampling Distance*) que permite determinar la altura del objeto detectado con los parámetros del sistema de visión, mediante,

$$GSD = \frac{d \cdot sensorSize}{f \cdot imageHeight}, \tag{14}$$

donde d es la distancia desde el sensor hasta el objeto, $sensorSize$ es el tamaño del sensor en píxeles, f es la distancia focal, $imageHeight$ es la resolución de la altura de la imagen en píxeles; estos componentes se asocian a la representación en la Figura 3.

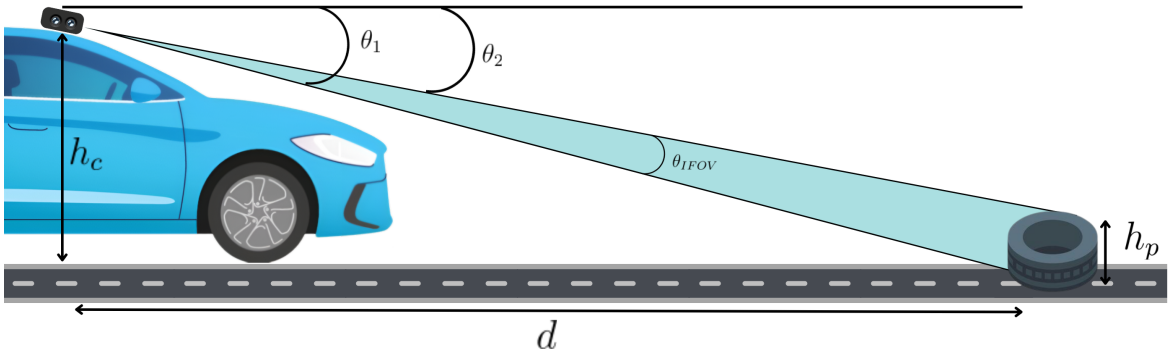


Figura 14. Representación del campo de visión instantáneo (IFOV). En esta representación se observan los ángulos de visión asociados a un obstáculo en función de la altura del sistema de visión h_c y la altura del obstáculo h_p, con la distancia (d) entre ellos. Elaboración propia.

3. **Distancia de frenado**: Integra las distancias de *offset*, percepción, latencia y frenado, permitiendo evaluar cuánto espacio necesita el vehículo para detenerse completamente a diferentes velocidades. La distancia a la que se encuentra el objeto detectado pertenece a un valor específico y se mapea frente a todos los valores de distancias de frenado obtenidas.

4. **Distancia de evasión**: Integra las distancias de offset, percepción, latencia y evasión, considerando los límites de vuelco (K_{roll}) y deslizamiento (K_{slip}) del vehículo, permitiendo determinar cuánto espacio necesita el vehículo para esquivar un obstáculo sin perder estabilidad. La distancia a la que se encuentra el objeto detectado pertenece a un valor específico y se mapea frente a todos los valores de distancias de evasión obtenidas.

5. **Tiempo de reacción**: Es la suma de los tiempos de percepción (t_p) y latencia (t_l). El tiempo de reacción (t_r) se categoriza en reacción lenta y rápida, como se puede observar en el Cuadro 1.

6. **Estados de conducción**: Se establecen según los rangos de velocidades que alcanza el vehículo, como se puede observar en el Cuadro 2.

Tipo de reacción	Tiempo de reacción
Lenta	$1.4\ s$
Rápida	$0.26\ s$

Cuadro 1. Descripción de los tiempos de reacción. Adaptado de: Aleksander Rydzewski y Pawel Czarnul. «Human awareness versus Autonomous Vehicles view: comparison of reaction times during emergencies». En: *2021 IEEE Intelligent Vehicles Symposium (IV)*. 2021, págs. 732-739. DOI: 10.1109/IV48863.2021.9575602.

Estado de conducción	Rango de velocidad
Alta Velocidad (AV)	$v > 100\ km/h$
Velocidad Rápida (VR)	$70 < v < 100\ km/h$
Velocidad Media (VM)	$40 < v < 70\ km/h$
Velocidad Baja (VB)	$20 < v < 40\ km/h$
Velocidad Muy Baja (VMB)	$v < 20\ km/h$

Cuadro 2. Descripción de los estados de conducción para diferentes rangos de velocidad. Adaptado de: Min-Joong Kim et al. «On the Development of Autonomous Vehicle Safety Distance by an RSS Model Based on a Variable Focus Function Camera». En: *Sensors* 21.20 (2021). DOI: 10.3390/s21206733

A partir de estos cálculos, el sistema realiza un análisis detallado para el cálculo de la velocidad de navegación segura y distancia de anticipación para frenar o esquivar dependiendo de los parámetros del sistema óptico y del vehículo en función de los objetos detectados y seleccionados para el análisis.

4. RESULTADOS

En esta sección se presentan los resultados obtenidos a partir del método propuesto para el cálculo de la velocidad de navegación segura y distancia de anticipación para frenar o evadir un obstáculo dependiendo de los parámetros del sistema óptico y del vehículo en función de los objetos detectados a cierta distancia.

4.1. Simulación de inferencia estéreo

En el primer módulo del simulador se presenta la interfaz de usuario, como se observa en la Figura 15, posteriormente se cargan las imágenes, como se observa en la Figura 16, para los conjuntos de datos de KITTI Stereo 2015 y DrivingStereo.

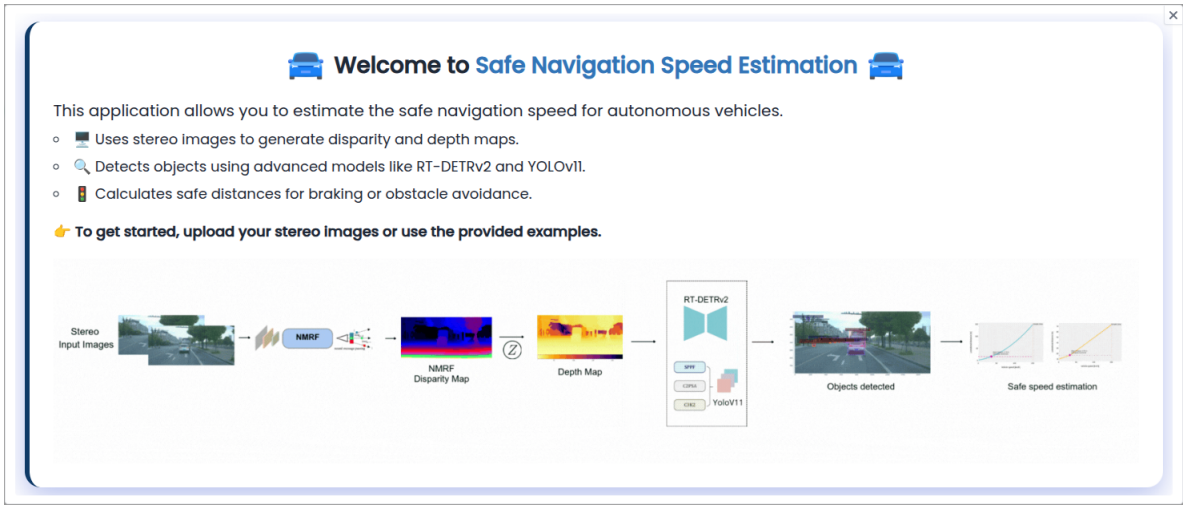


Figura 15. Interfaz de usuario de bienvenida al simulador. Se presenta la información del simulador y su paso a paso. Elaboración propia.

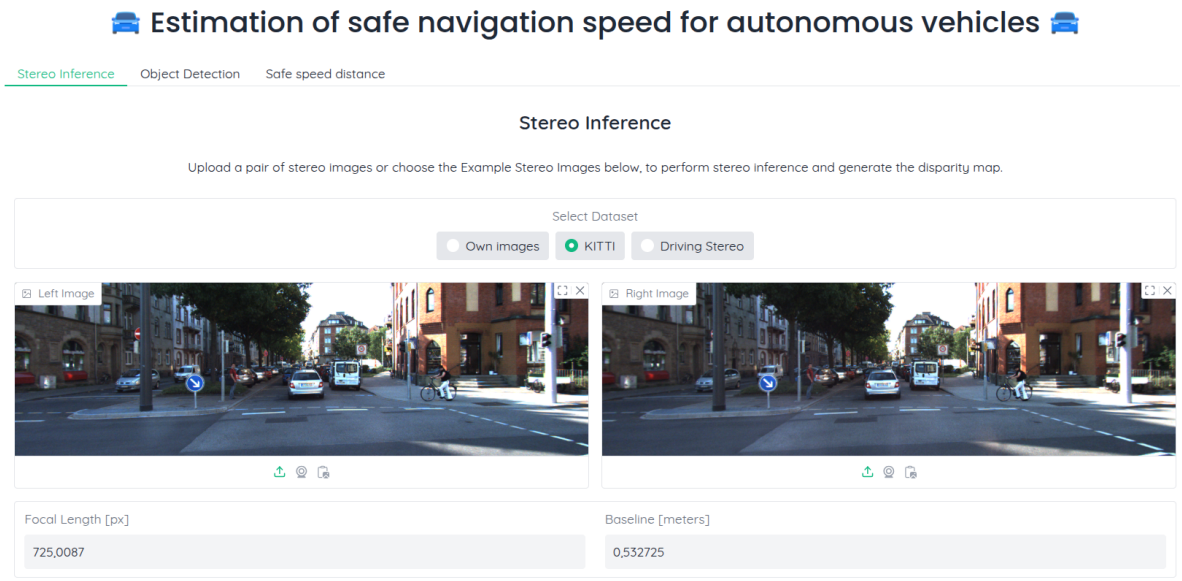


Figura 16. Interfaz de usuario desarrollada para la inferencia estéreo. Elaboración propia.

Al deslizar hacia abajo en la aplicación se permite generar el mapa de disparidad presionando en el botón "*Run Inference*", como se ilustra en la Figura 17.

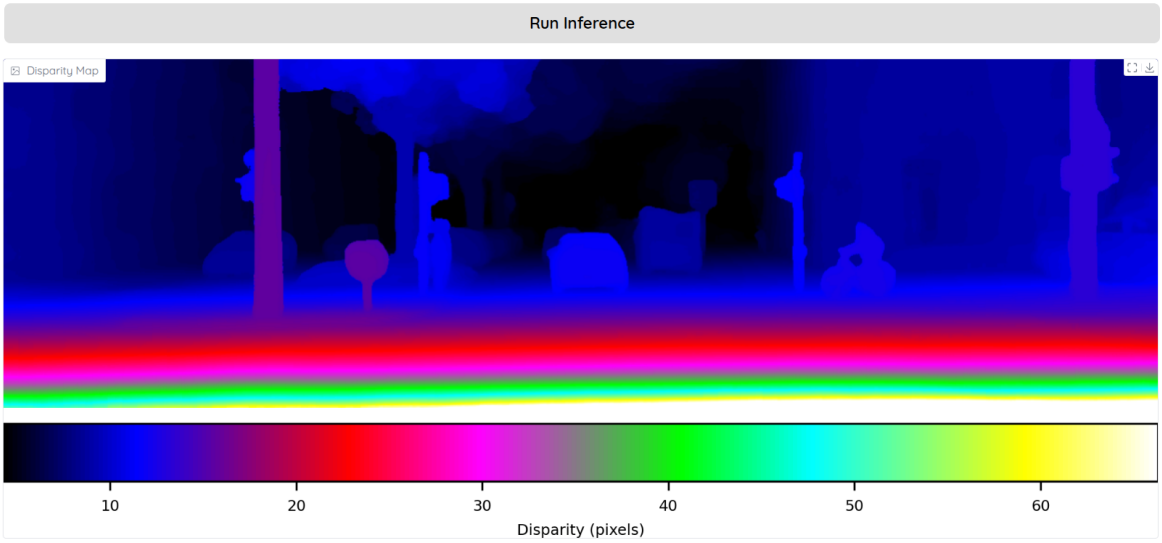


Figura 17. Mapa de disparidad generado a partir de imágenes estéreo. Los colores representan las diferencias de disparidad en la escena, donde las regiones más cálidas indican valores de disparidad bajos y las más frías valores más altos. Elaboración propia.

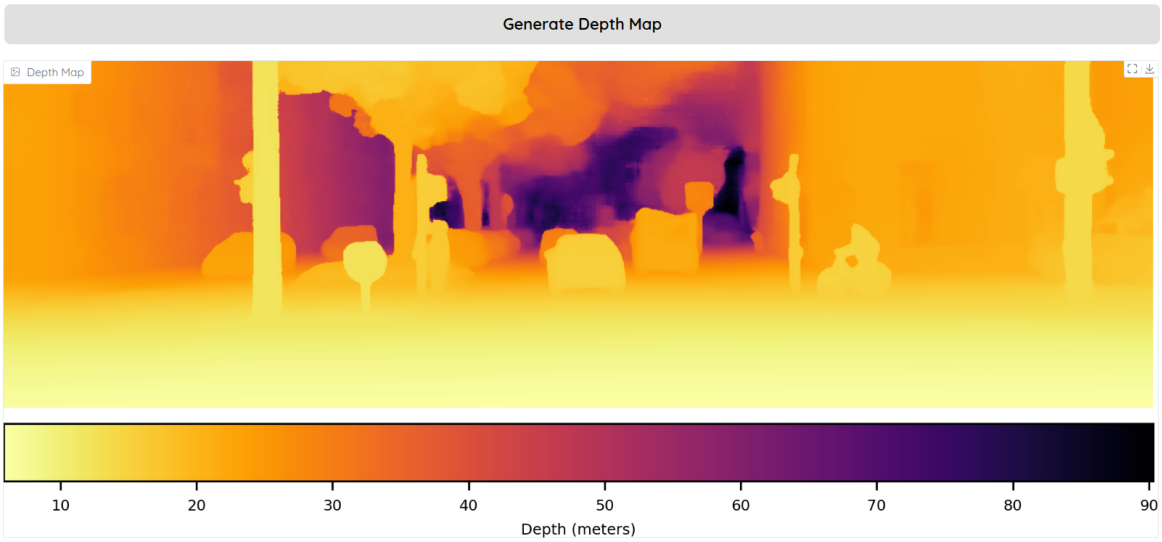


Figura 18. Mapa de profundidad generado a partir de imágenes estéreo a partir del mapa de disparidad. Los colores en la escala indican la distancia de los objetos en metros, donde los tonos más claros representan objetos más cercanos y los más oscuros objetos más lejanos. Elaboración propia.

Utilizando la Ecuación 9 sobre el mapa de disparidad se obtiene el mapa de profundidad medido en metros, como se ilustra en la Figura 18.

4.1.1. Métricas para evaluar el mapa de profundidad Las métricas para evaluar el rendimiento del mapa de profundidad son las siguientes:

- **Error medio absoluto (*MAE*)**: Definido por la ecuación:

$$\text{MAE} = \frac{1}{|\mathcal{S}|} \sum_{s} |\hat{d}_s - d_s^{gt}|, \tag{15}$$

donde d_s^{gt} es la profundidad real del píxel s, $\hat{d}_s$ es la profundidad predicha, y $|\mathcal{S}|$ es el número total de píxeles evaluados. Esta métrica tiene valores en el rango de $[0, \infty)$; se reporta en milímetros en este documento, y a menor valor, mejor

es la estimación.[72]

- **Raíz del error cuadrático medio (*RMSE*)**: Definido por la ecuación:

$$\text{RMSE} = \sqrt{\frac{1}{|\mathcal{S}|}\sum_{s}\left(\hat{d}_s - d_s^{gt}\right)^2}. \tag{16}$$

Esta métrica mide la magnitud promedio de los errores, pero penaliza más fuertemente los errores grandes debido a la elevación al cuadrado. También tiene valores en el rango de $[0, \infty)$; se reporta en milímetros en este documento, y a menor valor, mejor es la estimación.[73]

- **Error absoluto medio de la profundidad inversa (*iMAE*)**: Definido por la ecuación:

$$\text{iMAE} = \frac{1}{|\mathcal{S}|}\sum_{s}\left|\frac{1}{\hat{d}_s} - \frac{1}{d_s^{gt}}\right|. \tag{17}$$

Esta métrica mide la diferencia absoluta entre las profundidades inversas, dando mayor relevancia a los objetos cercanos. Se reporta en unidades de [1/km], y valores más bajos indican una mejor estimación.[74]

- **Error cuadrático medio de la profundidad inversa (*iRMSE*)**: Definido por la ecuación:

$$\text{iRMSE} = \sqrt{\frac{1}{|\mathcal{S}|}\sum_{s}\left(\frac{1}{\hat{d}_s} - \frac{1}{d_s^{gt}}\right)^2}. \tag{18}$$

Esta métrica penaliza más los errores grandes en objetos cercanos debido a

---

[72] Jian Qian et al. *SDformer: Efficient End-to-End Transformer for Depth Completion*. 2024. arXiv: 2409.08159 [cs.CV].

[73] Qian et al., ver n. 72.

[74] Qian et al., ver n. 72.

la elevación al cuadrado. Se reporta en unidades de [1/km], y a menor valor, mejor es la estimación.[75]

Las cuatro métricas se evalúan en cinco escenas distintas del conjunto de datos de entrenamiento y validación de KITTI Stereo 2015: Ciudad, Residencial, Carretera, Campus y Persona, así como en el conjunto de datos de prueba de DrivingStereo: Conducción, Soleado, Neblina, Nublado y Lluvioso. Para cada conjunto de datos es necesario conocer los parámetros ópticos, como se describen en el Cuadro 3.

Conjunto de datos	Imágenes	Distancia focal [px]	Baseline [m]	Resolución [px]	Tamaño del pixel [µm]
KITTI (Entrenamiento)	1130	725.0087	0.532725	1242 × 375	4.65
KITTI (Validación)	521	725.0087	0.532725	1242 × 375	4.65
DrivingStereo (Prueba)	500	2007.113	0.54	1762 × 800	5.86

Cuadro 3. Descripción de los conjuntos de datos utilizados. La distancia focal se obtiene a partir de la matriz de calibración proporcionada en cada conjunto de datos.

La profundidad predicha se obtiene a partir del procesamiento en la simulación de inferencia estéreo con el modelo NMRF, como se explica en la sección 4.1 y la profundidad real (*ground truth*) la proporciona cada conjunto de datos. Los resultados de evaluación para las cuatro métricas se pueden observar en el Cuadro 4.
El modelo NMRF implementado en el simulador tiene mejor rendimiento en la escena Ciudad, con los errores más bajos en validación. En cambio, en Campus, los errores son los más altos, lo que indica que es un entorno más difícil de procesar. La escena Persona tiene los menores errores, pero solo se evaluó en entrenamiento, ya que en validación no se encuentra disponible esta escena. Además, los errores inversos son mayores en escenas como Residencial y Campus, lo que muestra que el modelo tiene más dificultades en entornos con mucha variación de profundidad. Para el conjunto de prueba se presenta el menor MAE en un clima Nublado, el menor RMSE en un entorno urbano de Conducción, el menor iMAE y iRMSE para un

[75] Qian et al., ver n. 72.

Conjunto de datos	Escena	MAE [mm]	RMSE [mm]	iMAE [1/km]	iRMSE [1/km]
KITTI (Entrenamiento)	Ciudad	506.882	1219.603	**1.346**	**1.724**
	Residencial	442.107	1232.146	1.364	2.253
	Carretera	442.107	1232.146	1.364	2.253
	Campus	585.004	1693.675	1.914	3.004
	Persona	**295.253**	**721.806**	2.447	3.306
KITTI (Validación)	Ciudad	**335.088**	**850.060**	**1.217**	**1.653**
	Residencial	408.385	1290.457	1.662	3.974
	Carretera	505.883	1169.286	2.086	3.148
	Campus	676.914	2391.028	2.685	5.702
	Persona	–	–	–	–
DrivingStereo (Prueba)	Conducción	1677.054	**3173.304**	3.126	8.215
	Soleado	1816.341	4221.276	**2.646**	**7.160**
	Neblina	1735.812	4114.396	3.072	7.821
	Nublado	**1345.608**	4031.648	2.855	7.808
	Lluvioso	2917.907	4655.667	4.512	13.566

Cuadro 4. Comparación de las métricas de profundidad. Se realiza una comparación utilizando las métricas de profundidad en cinco escenas del conjunto de datos KITTI para entrenamiento y cuatro escenas de validación, dado que la escena 'Persona' no está disponible. Además, se incluyen cinco escenas del conjunto de datos DrivingStereo para evaluar los resultados.

clima Soleado. En la Figura 19 se observa el análisis mediante diagramas de barras.

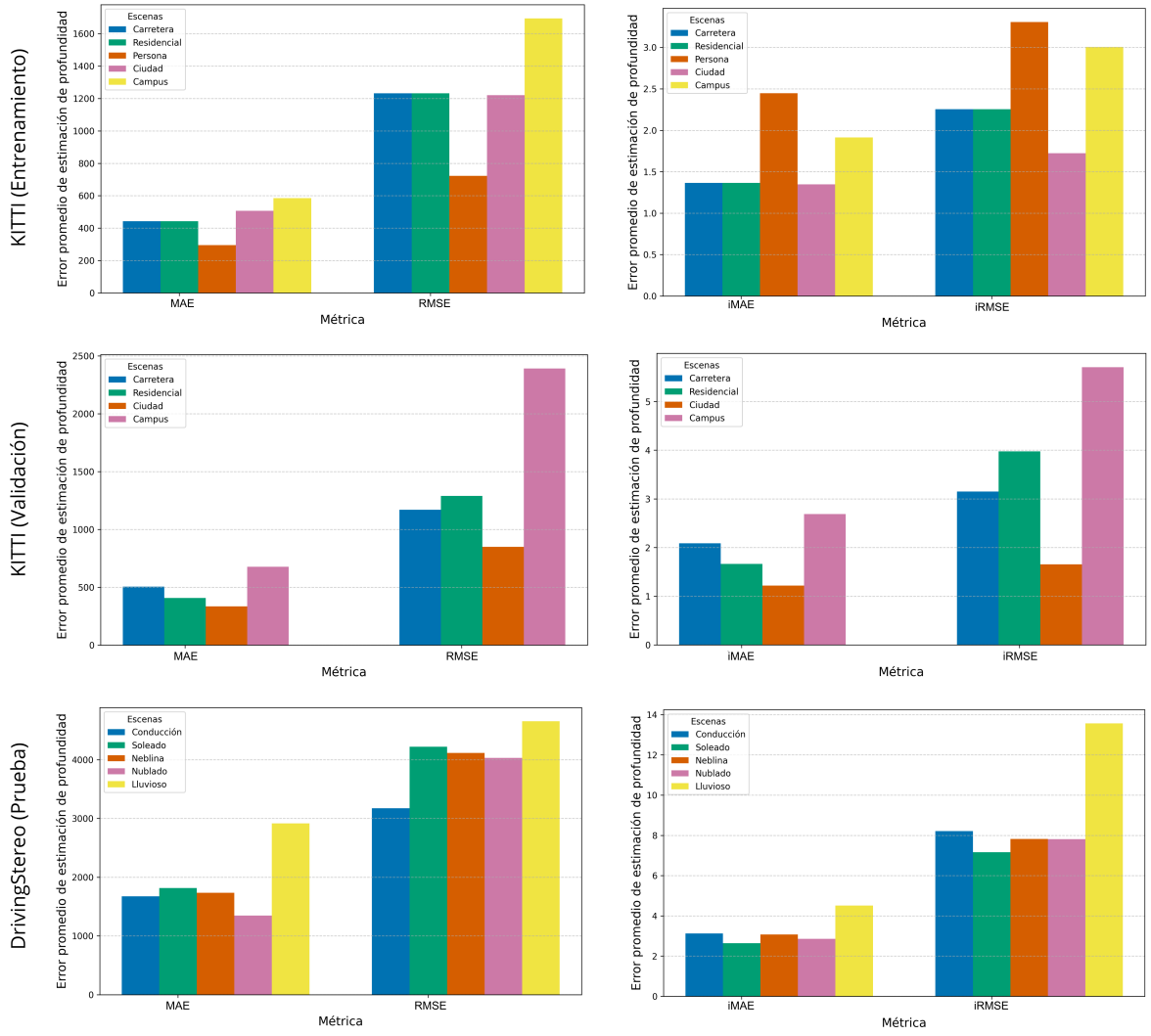


Figura 19. Gráfico de barras para la comparación de las métricas de profundidad en los conjuntos de datos KITTI Stereo 2015 y DrivingStereo.

4.2. Simulación de detección de objetos

Para el segundo módulo se presentan los resultados de la detección de objetos. En la Figura 20 se representa la detección para el modelo YOLOv11. La interfaz desarrollada con Plotly permite interactuar con la imagen de detección de objetos para acercar, alejar, ampliar, descargar y conocer las coordenadas de cada fragmento de

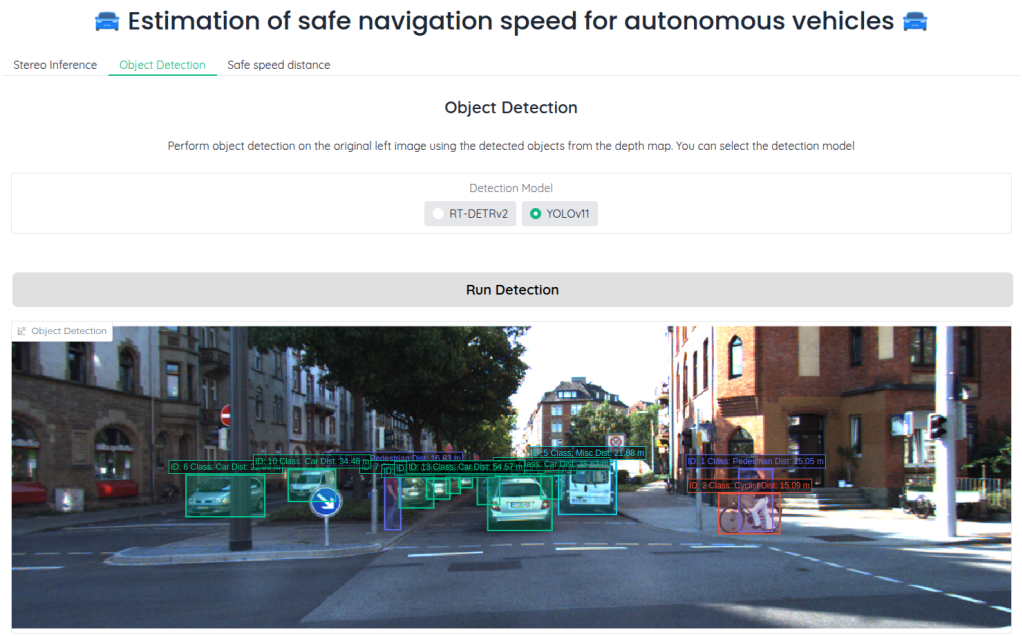


Figura 20. Detección de objetos en la imagen estéreo izquierda utilizando el modelo YOLOv11. Las cajas delimitadoras muestran la clasificación de los objetos detectados junto con sus respectivas distancias estimadas en metros, obtenidas a partir del mapa de profundidad. Elaboración propia.

la imagen, como se observa en la Figura 21.

Al concluir este módulo se presenta un resumen en tarjetas interactivas de los objetos detectados con la clases de los objetos, su altura en píxeles y metros, las coordenadas de la cajas delimitadoras y la distancia estimada en metros, como se

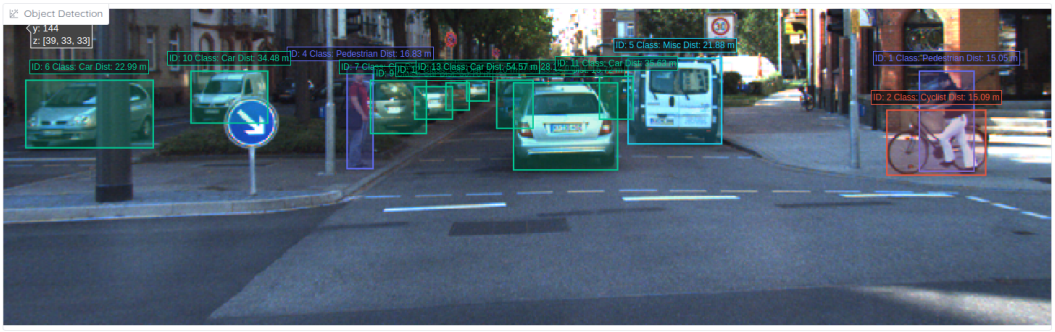


Figura 21. Interacción con la imagen generada de la detección de objetos. La interfaz permite acercar, alejar, ampliar y descargar la imagen. Elaboración propia.

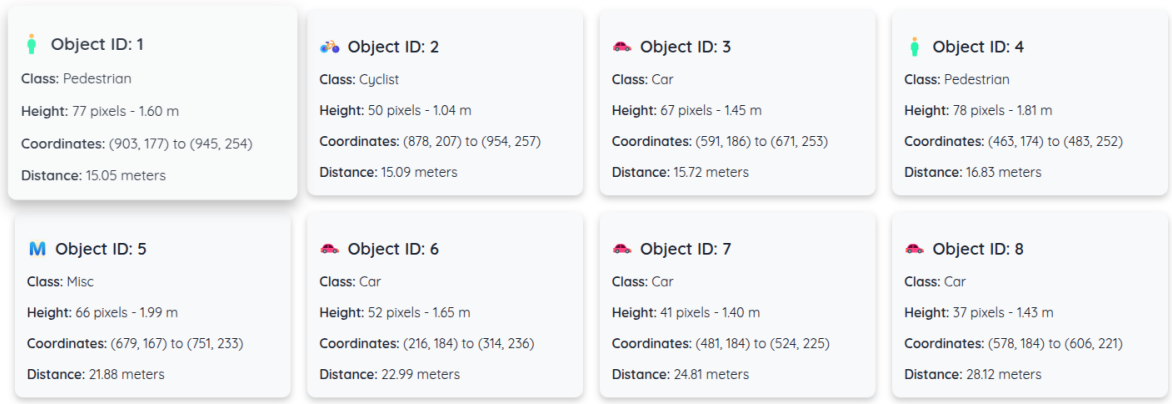


Figura 22. Resumen de los objetos detectados en la escena mediante el modelo YOLOv11. Se presentan las clases de los objetos, sus alturas en píxeles, las coordenadas de las cajas delimitadoras y las distancias estimadas en metros a partir del mapa de profundidad. Elaboración propia.

observa en la Figura 22.

4.2.1. Métricas para evaluar la detección de objetos Las métricas para evaluar la detección de objetos en ambos modelos se presentan a continuación, donde los valores a calcular dependen de la intersección sobre la unión, los verdaderos positivos TP, verdaderos negativos TN, falsos positivos FP, y falsos negativos FN, asociados a la clase de cada ejemplo del conjunto de datos:

- **Intersección sobre la unión (*IoU*)**: Métrica utilizada para evaluar la superposición entre dos volúmenes o áreas arbitrarias A y B. Se define como la razón entre el área de intersección y el área de unión de ambos cuadros delimitadores,

$$IoU = \frac{|A \cap B|}{|A \cup B|}, \tag{19}$$

donde A es el área del cuadro delimitador predicho, B es el área del cuadro delimitador real (ground truth), $|A \cap B|$ representa el área de intersección entre ambos cuadros, y $|A \cup B|$ representa el área de la unión de los dos cuadros.

Esta métrica toma valores entre 0 y 1, donde 1 indica una superposición perfecta.[76]

- **Precisión**: Mide la proporción de ejemplos clasificados positivamente que realmente pertenecen a la clase positiva[77],

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{20}$$

- **Sensibilidad**: Mide la capacidad del modelo para detectar todos los ejemplos positivos, indicando la proporción de verdaderos positivos sobre el total de ejemplos positivos reales[78],

$$\text{Sensibilidad} = \frac{TP}{TP + FN}. \tag{21}$$

- **Métrica-F1**: Es la media armónica entre precisión (20) y sensibilidad (21), proporcionando un equilibrio entre ambas métricas, especialmente útil cuando existe un desequilibrio de clases[79],

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Sensibilidad}}{\text{Precision} + \text{Sensibilidad}}. \tag{22}$$

Es oportuno señalar que para YOLOv11 el entrenamiento se realizó con un esquema

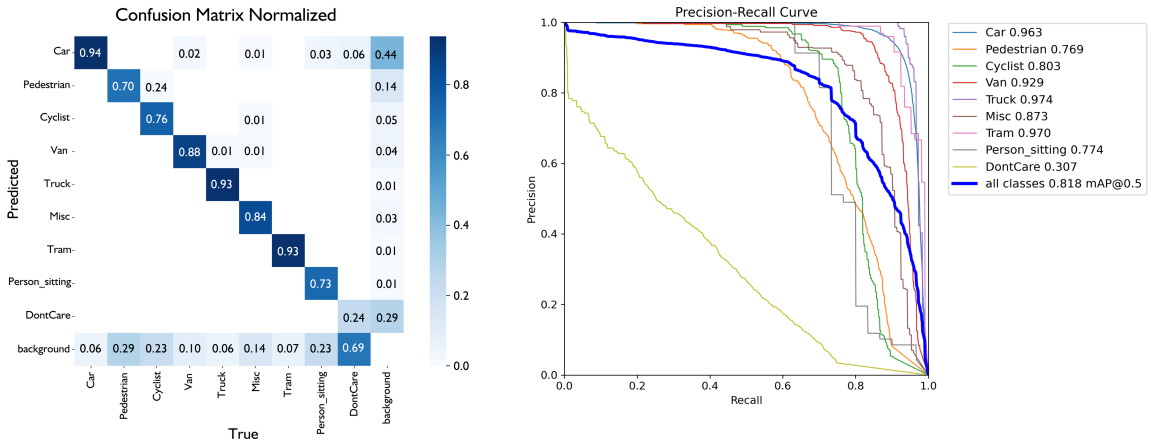
---

[76] Hamid Rezatofighi et al. *Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression*. 2019. arXiv: 1902.09630 [cs.CV].

[77] Reda Yacouby y Dustin Axman. «Probabilistic Extension of Precision, Recall, and F1 Score for More Thorough Evaluation of Classification Models». En: *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*. Ed. por Steffen Eger et al. Online: Association for Computational Linguistics, nov. de 2020, págs. 79-91. DOI: 10.18653/v1/2020.eval4nlp-1.9.

[78] Yacouby y Axman, ver n. 77.

[79] Yacouby y Axman, ver n. 77.



(a) (b)

Figura 23. Resultados de entrenamiento para el algoritmo YOLOv11. En (a) se puede observar la matriz de confusión y en (b) se puede observar la curva de Precisión - Sensibilidad. Elaboración propia.

de ajuste fino[80] estableciendo 25 épocas, una paciencia de 10 épocas sin mejora antes de detenerse, la técnica de aumento de datos *mixup* con un valor de 0.2 para mejorar la generalización al combinar imágenes aleatorias, y está configurado para detectar las clases *Car, Pedestrian, Cyclist, Van, Truck, Misc, Tram, Person sitting* y *DontCare*; en la Figura 23 se observan los resultados.

La evaluación de los modelos se realiza para tres clases: *Car*, *Pedestrian* y *Cyclist*. Para ello, se calculan los valores promedio de todas las métricas en el conjunto de prueba de KITTI Object Detection 2012. Se compararon tres modelos: *RT-DETRv2 Inference*, configurado con 101 capas ResNet y empleado en modo de inferencia, sin ajuste fino debido al alto costo computacional asociado; *YOLOv11 Inference*, que se utiliza directamente en modo de inferencia sin ajuste adicional; y *YOLOv11 Fine-Tuned*, que fue ajustado finamente para mejorar su desempeño. Los resultados se presentan en el Cuadro 5.

---

80 Katherine Tian et al. *Fine-tuning Language Models for Factuality*. 2023. arXiv: 2311.08401 [cs.CL].

Modelo	IoU	Precisión	Sensibilidad	Métrica-F1
RT-DETRv2 Inference	0.4561	0.9063	0.5954	0.7187
YOLOv11 Inference	0.4774	0.7208	0.7717	0.7454
YOLOv11 Fine-Tuned	**0.7024**	**0.9664**	**0.8285**	**0.8922**

Cuadro 5. Comparación de las métricas de detección de objetos. Para los modelos RT-DETRv2 Inference, YOLOv11 Inference y YOLOv11 Fine-Tuned en el conjunto de prueba de KITTI Object Detection Evalutaion 2012.

4.3. Simulación del cálculo de la velocidad de navegación segura

En este módulo se busca calcular la velocidad de navegación segura para frenar ó esquivar los objetos seleccionados en función de los parámetros, como, el coeficiente de fricción, tiempo de percepción, latencia del sistema, distancia de *offset*, así como características del vehículo como el modelo, el ángulo de giro, la altura del centro de gravedad y el ancho de la base de ruedas delanteras y traseras. Se ilustra en la Figura 24. Después de seleccionar el objeto, al deslizarse hacia abajo

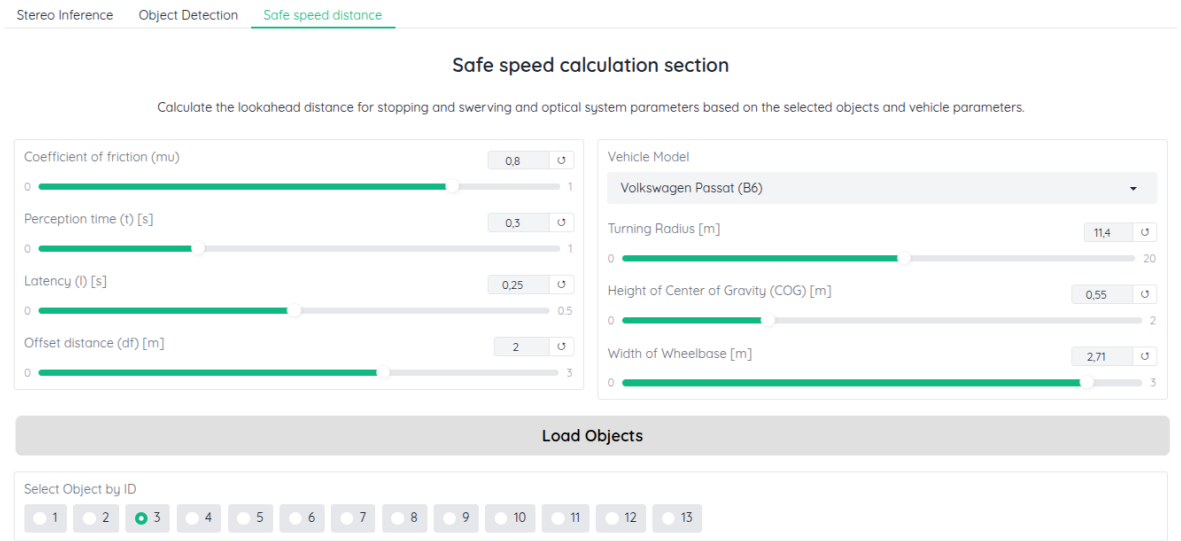


Figura 24. Interfaz de usuario para el cálculo de la velocidad de navegación segura y la distancia de anticipación para frenar o evadir obstáculos. Elaboración propia.

se despliega el panel gráfico con los resultados de la simulación, como se observa

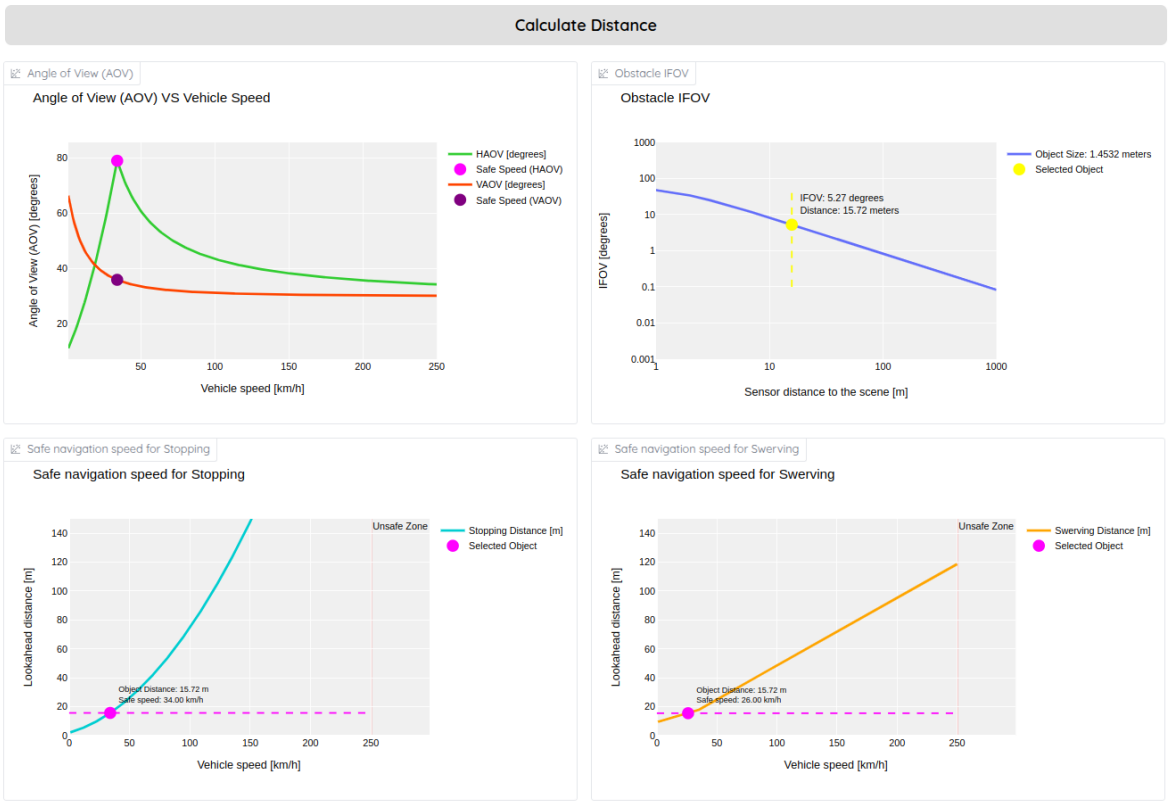


Figura 25. Panel gráfico con los resultados de la simulación. Se observan los resultados del análisis de los ángulos de visión, campo de visión instantáneo, velocidad segura para frenar y velocidad segura para esquivar. Elaboración propia.

en la Figura 25. El código fuente del simulador puede ser consultado en el siguiente enlace: `Repositorio en GitHub`.

A continuación, se describirán 3 simulaciones realizadas para analizar el comportamiento del simulador. La primera simulación consiste en comparar los objetos detectados con los modelos RT-DETRv2 Inference y YOLOv11 Fine-tuned para KITTI Stereo 2015, como se observa en el Cuadro 6. Para la misma escena ambos modelos ofrecen un rendimiento similar en la detección de objetos en condiciones urbanas, las diferencias observadas son mínimas, por lo tanto, la elección entre ambos modelos dependerá de criterios, como el costo computacional, la velocidad de carga, entre otros.

Objeto detectado	Modelo	Distancia	IFOV	VAOV	HAOV	Altura	Pixeles
Car	RT-DETRv2	13.81 m	8.11°	36.96°	67.88°	1.96 m	103 px
	YOLOv11	13.82 m	8.27°	36.96°	67.88°	2.00 m	105 px
$Pedestrian$	RT-DETRv2	5.94 m	14.86°	45.62°	29.65°	1.58 m	193 px
	YOLOv11	5.94 m	15.02°	45.62°	29.65°	1.60 m	195 px
$Cyclist$	RT-DETRv2	5.89 m	9.27°	46.58°	27.84°	0.99 m	122 px
	YOLOv11	5.89 m	9.19°	46.58°	27.84°	0.98 m	121 px

Cuadro 6. Comparación de objetos detectados con diferentes modelos. Para la misma escena urbana del conjunto de datos KITTI Stereo 2015 utilizando los modelos RT-DETRv2 Inference y YOLOv11 Fine-tuned en un Volkswagen Passat (B6).

Para la segunda simulación, se utilizó el conjunto de datos DrivingStereo y el modelo RT-DETRv2 Inference con el objetivo de obtener las velocidades de navegación segura y analizar las características de percepción visual de un sistema de visión estéreo. Este análisis se llevó a cabo bajo dos condiciones climáticas distintas: Lluviosa y Soleada. En cada una de estas condiciones, se tuvieron en cuenta variaciones en los coeficientes de fricción, los cuales afectan directamente la desaceleración del vehículo. En el Cuadro 7 se presentan los resultados obtenidos para diferentes distancias del objeto detectado. Los datos incluyen las velocidades seguras para las maniobras de frenado y esquiva, así como los valores de los ángulos de visión VAOV y HAOV, y el campo de visión instantáneo, todo en función de las condiciones climáticas y las distancias al objeto. Los coeficientes de fricción para la escena Lluviosa y la escena Soleada son de μ = 0.4 y μ = 0.8, respectivamente, lo que provoca una menor desaceleración en la primera condición. Además, los datos fueron recopilados utilizando los parámetros del vehículo Hyundai Aviancer: radio de giro de 11.6 m, altura del centro de gravedad de 0.66 m y una distancia entre los ejes de 2.82 m. La simulación empleó el modelo RT-DETRv2 para el análisis de la percepción visual.

Ahora, para analizar cómo varían los estados de conducción en función de la velocidad estimada, se utilizan como referencia los datos obtenidos en la escena Lluviosa, cuya representación se muestra en la Figura 26. El comportamiento de la velocidad

Distancia del objeto	**Escena lluviosa** ($\mu = 0.4$, $a = 3.92m/s^2$)						**Escena soleada** ($\mu = 0.8$, $a = 7.84m/s^2$)					
	Frenar	Esquivar	IFOV	VAOV	HAOV	Altura	Frenar	Esquivar	IFOV	VAOV	HAOV	Altura
$0m < d < 10m$	18 km/h	–	9.5°	39.92°	46.60°	1.623 m	19 km/h	-	6.26°	41.30°	40.80°	0.923 m
$10m < d < 20m$	24 km/h	16 km/h	5.16°	37.06°	65.84°	1.226 m	33 km/h	25 km/h	6.83°	36.22°	74.81°	1.867 m
$20m < d < 30m$	35 km/h	37 km/h	3.6°	34.23°	53.06°	1.278 m	44 km/h	43 km/h	3.73°	34.31°	65.94°	1.455 m
$30m < d < 40m$	51 km/h	66 km/h	2.03°	32.39°	44.38°	1.411 m	65 km/h	77 km/h	1.85°	32.48°	52.57°	1.248 m
$40m < d < 50m$	57 km/h	81 km/h	1.65°	31.99°	42.51°	1.388 m	77 km/h	101 km/h	1.77°	31.92°	48.48°	1.537 m
$50m < d < 60m$	59 km/h	86 km/h	1.54°	31.88°	41.98°	1.378 m	83 km/h	115 km/h	1.88°	31.70°	46.92°	1.845 m
$60m < d < 70m$	68 km/h	109 km/h	1.26°	31.49°	40.03°	1.407 m	91 km/h	133 km/h	1.26°	31.47°	45.18°	1.421 m
$70m < d < 80m$	73 km/h	125 km/h	1.08°	31.32°	39.17°	1.386 m	101 km/h	150 km/h	0.83°	31.24°	43.41°	1.106 m
$80m < d < 90m$	79 km/h	143 km/h	0.94°	31.15°	38.29°	1.373 m	110 km/h	183 km/h	1.03°	31.08°	42.12°	1.582 m
$90m < d < 100m$	84 km/h	150 km/h	0.83°	31.04°	37.66°	1.344 m	115 km/h	198 km/h	0.91°	31.00°	41.49°	1.521 m

Cuadro 7. Comparación de las velocidades de navegación segura en diferentes condiciones climáticas. Para diferentes imágenes en el conjunto de datos DrivingStereo se utilizan la escena Lluviosa con un coeficiente de fricción de μ = 0.4 y la escena Soleada con un coeficiente de fricción de μ = 0.8. Se varían las distancias en una escala de 0 a 100 metros para un tiempo de reacción moderado en ambas escenas y con los parámetros del vehículo Hyundai Aviancer con el modelo RT-DETRv2.

segura ante maniobras como el frenado o la evasión puede variar dependiendo del estado de conducción y la distancia al objeto detectado. Por ejemplo, en el estado de conducción Velocidad Rápida (VR), el rango de velocidad es adecuado para realizar una maniobra de frenado y así evitar una colisión. Sin embargo, para ejecutar una maniobra de evasión, se requiere un rango más amplio, correspondiente al estado de Alta Velocidad (AV), ya que esta maniobra alcanza una velocidad segura de hasta 125 km/h antes de ser aplicada. Un análisis similar se observa en el estado de Alta Velocidad (AV), donde el rango de velocidad no es suficiente para aplicar la maniobra de frenado de manera segura, pero sí lo es para la maniobra de evasión. Este análisis identifica los rangos de velocidad más seguros para cada tipo de maniobra.

Desde otra perspectiva, para los resultados mostrados en el Cuadro 7, analizaremos los valores correspondientes a las maniobras de frenado y evasión. Se comparan las velocidades de navegación segura para todas las distancias en ambas escenas, como se ilustra en la Figura 27. El comportamiento de las velocidades es lineal para las maniobras de evasión y cuadrático para las maniobras de detención. A medida

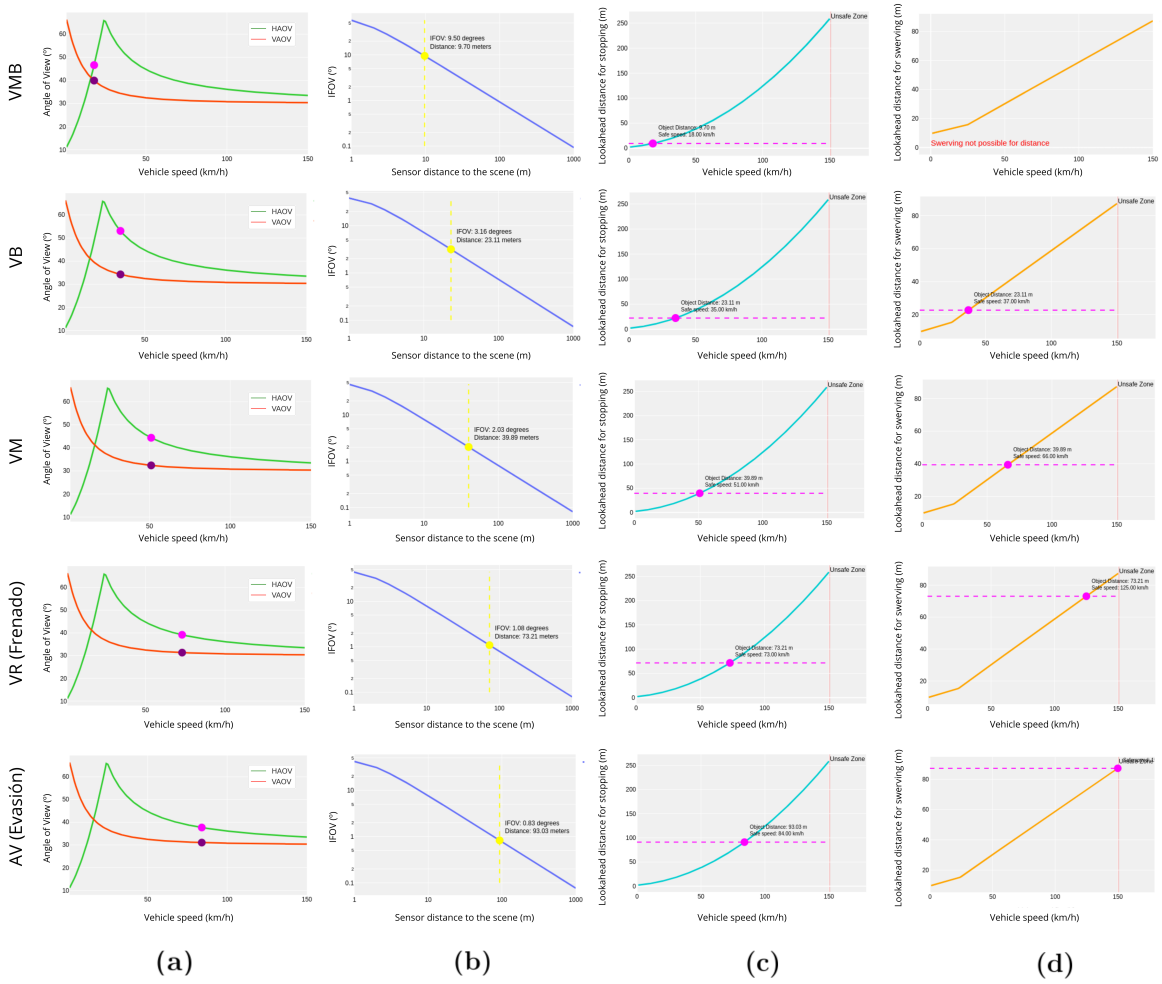


Figura 26. Análisis de los estados de conducción en función de la velocidad estimada. En la columna (a) se presentan las gráficas de los ángulos de visión VAOV y HAOV. En la columna (b) se presentan las gráficas del campo de visión instantáneo IFOV. En la columna (c) se presenta las gráficas de la velocidad segura de frenado. En la columna (d) se presentan las gráficas velocidad segura de evasión. Elaboración propia.

que aumenta la distancia al objeto, la velocidad segura también incrementa. Para distancias menores a 30 metros, es preferible aplicar la maniobra de detención, mientras que para distancias superiores a 30 metros, la maniobra de evasión comienza a ser más adecuada si se desea mantener una velocidad creciente. Este análisis es especialmente relevante en condiciones climáticas lluviosas y soleadas,

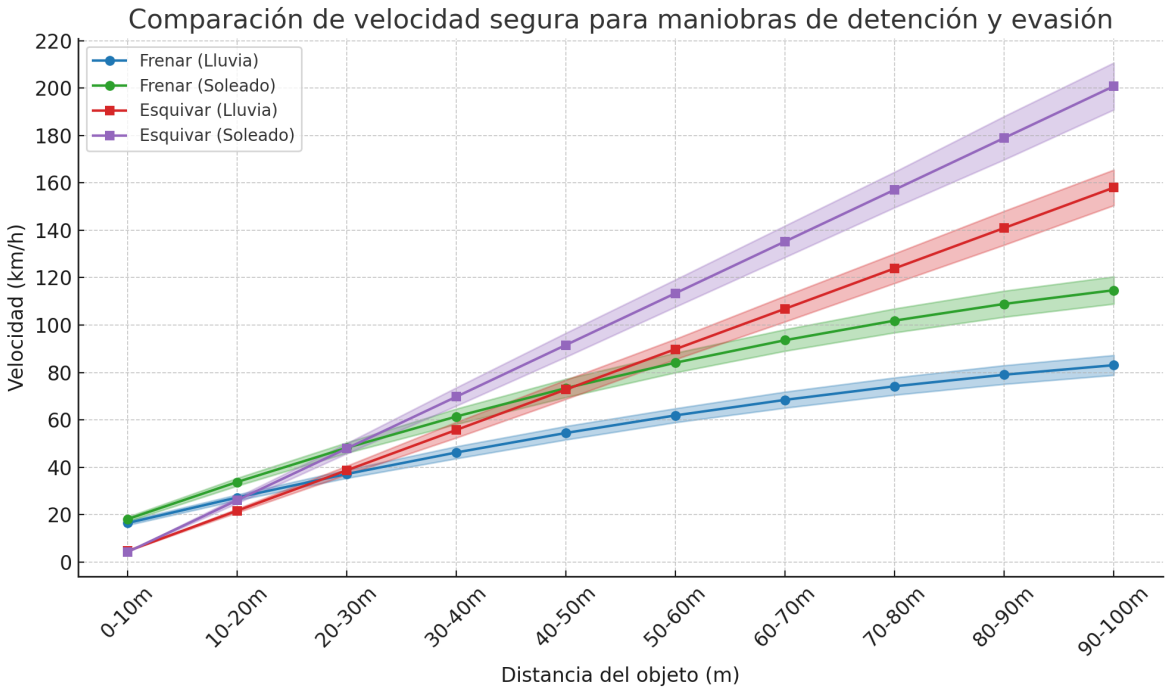


Figura 27. Comparación de velocidades seguras para maniobras de detención y evasión variando el coeficiente de fricción. Elaboración propia.

donde los coeficientes de fricción varían, afectando las decisiones de maniobra de acuerdo con las condiciones del entorno.

Para la tercera simulación, se utilizó el conjunto de datos KITTI Stereo 2015 con el modelo YoloV11 Fine-Tuned para obtener las velocidades de navegación segura y el análisis de las características de percepción visual para los tipos de reacción lenta y rápida en la misma escena, como se observa en el Cuadro 8.

Finalmente, podemos analizar los cambios de la estimación de la velocidad de navegación segura dependiendo de los tipos de reacción lenta y rápida, para las maniobras de detenimiento y evasión, como se observa en la Figura 28. El comportamiento para una reacción lenta indica que es preferible realizar la maniobra de detenimiento en una distancia mayor a 50 metros y la maniobra de evasión en una distancia menor a 50 metros para una velocidad creciente. Para la reacción rápida es preferible realizar la maniobra de evasión en una distancia mayor a 20 metros y la

Distancia del objeto	Reacción lenta $(1.4s)$						Reacción rápida $(0.26s)$					
	Frenar	Esquivar	IFOV	VAOV	HAOV	Altura	Frenar	Esquivar	IFOV	VAOV	HAOV	Altura
$0m < d < 10m$	5 km/h	-	14.38°	47.24°	26.72°	1.442 m	17 km/h	-	14.38°	46.98°	27.16°	1.442 m
$10m < d < 20m$	17 km/h	8 km/h	6.22°	36.59°	71.78°	1.597 m	40 km/h	37 km/h	6.22°	36.48°	52.93°	1.597 m
$20m < d < 30m$	31 km/h	27 km/h	3.47°	33.56°	133.34°	1.626 m	60 km/h	71 km/h	3.47°	33.54°	43.22°	1.626 m
$30m < d < 40m$	38 km/h	36 km/h	3.08°	32.83°	134.70°	1.803 m	69 km/h	91 km/h	3.08°	32.82°	40.95°	1.803 m
$40m < d < 50m$	51 km/h	53 km/h	1.82°	31.99°	106.13°	1.536 m	86 km/h	133 km/h	1.82°	31.95°	38.13°	1.536 m
$50m < d < 60m$	57 km/h	60 km/h	2.61°	31.74°	97.55°	2.489 m	92 km/h	151 km/h	2.61°	31.74°	37.41°	2.489 m
$60m < d < 70m$	65 km/h	72 km/h	1.50°	31.47°	88.69°	1.700 m	101 km/h	181 km/h	1.50°	31.48°	36.52°	1.700 m
$70m < d < 80m$	70 km/h	79 km/h	2.45°	31.33°	84.21°	3.043 m	107 km/h	199 km/h	2.45°	31.34°	36.02	3.043 m
$80m < d < 90m$	77 km/h	91 km/h	0.95°	31.18°	78.97°	1.352 m	116 km/h	229 km/h	0.95°	31.16°	35.38°	1.352 m

Cuadro 8. Comparación de las velocidades seguras para diferentes tiempos de reacción. Para diferentes imágenes en el conjunto de datos KITTI Stereo 2015 se utilizan los tipos de reacción lenta y rápida para la misma escena, se varían las distancias en una escala de 0 a 100 metros y con los parámetros del vehículo Volkswagen Passat (B6) con el modelo YOLOv11 Fine-Tuned.

maniobra de detenimiento en una distancia menor a 20 metros para una velocidad creciente.

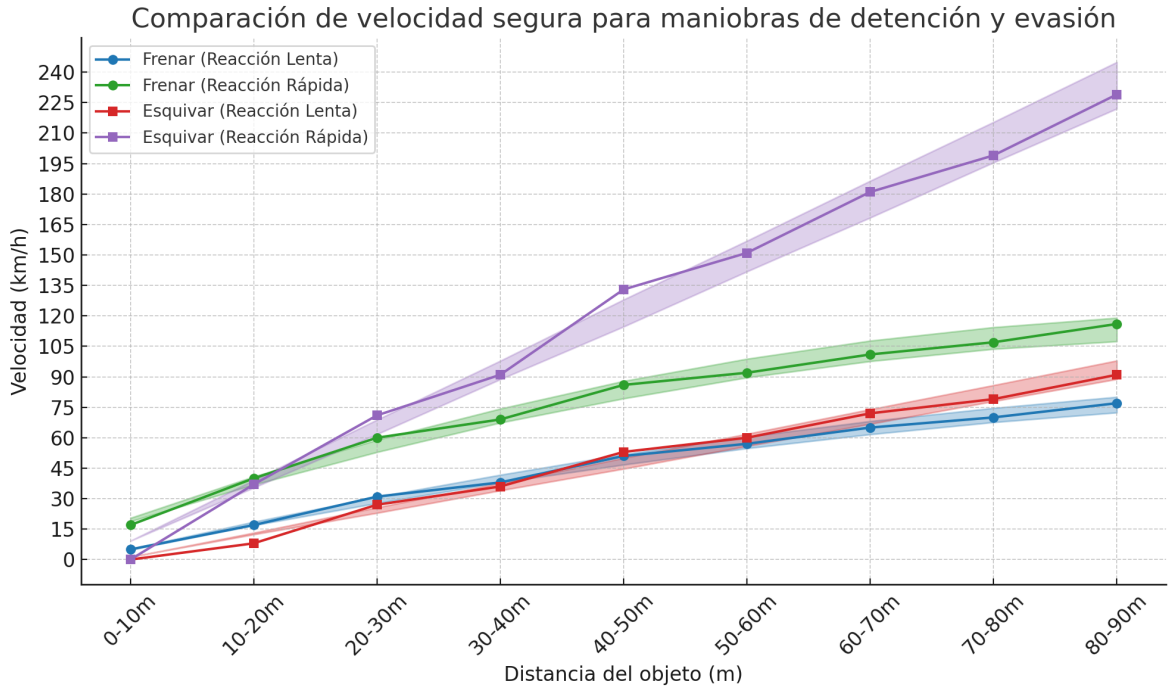


Figura 28. Comparación de velocidades seguras para maniobras de detención y evasión variando el tiempo de reacción. Elaboración propia.

5. CONCLUSIONES

En este trabajo se presentó un simulador para la estimación de la velocidad de navegación segura para vehículos autónomos en términos de maniobras de detenimiento y evasión de obstáculos detectados mediante algoritmos de visión por computadora en imágenes RGB adquiridas por un sistema de visión estéreo. El simulador permite escoger imágenes de cualquier conjuntos de datos teniendo en cuenta los valores de la distancia focal y *baseline*, generando los mapas de disparidad y profundidad con un modelo apoyado en aprendizaje profundo. Asimismo, la detección de objetos en la escena nos permite escoger modelos de vanguardia con técnicas optimizadas y de aprendizaje profundo tanto en inferencia y ajuste fino, permitiendo conocer los detalles de los objetos en términos de las propiedades de los lentes y sensores empleados. Adicionalmente, el proceso de comparación y evaluación de estos modelos nos permite conocer su funcionamiento para múltiples distancias a la que se encuentre el sistema de visión estéreo. El desarrollo de este simulador cuenta con una interfaz modular interactiva que permite parametrizar tanto las características del sistema de percepción, las condiciones del entorno, la geometría del vehículo y seleccionar un objeto detectado. El simulador ofrece a los investigadores interesados conocer la estimación de la velocidad segura respecto a los objetos detectados y su distancia hasta el sistema de visión estéreo para las maniobras de detenimiento y evasión, y cómo las características de percepción visual varían tanto para el AOV y el IFOV.

6. TRABAJO FUTURO

El simulador abre diversas oportunidades para trabajos futuros. Primero, ampliar las capacidades del simulador para abordar escenarios donde se incluyan parámetros adicionales como la curvatura de la carretera, el tipo de calzada (simple o doble), escenarios rurales con terrenos con el piso no pavimentado, condiciones climáticas como temperatura, nieve, propiedades de los neumáticos como dureza y desgaste y obstáculos ocultos como huecos y zanjas[81]. Segundo, el análisis de sensores y algoritmos de visión por computadora con técnicas para la generación de nubes de puntos[82], segmentación semántica[83], detección de objetos en 3D[84]. Tercero, el procesamiento computacional se puede optimizar para el rendimiento de procesar video con alta velocidad de fotogramas, además de incorporar un módulo de decisión final que evalúe múltiples factores para determinar la maniobra más segura.

---

[81] Kelly y Stentz, «Rough Terrain Autonomous Mobility - Part 1: A Theoretical Analysis of Requirements», ver n. 1.

[82] Yuquan Xu et al. «3D point cloud map based vehicle localization using stereo camera». En: *2017 IEEE Intelligent Vehicles Symposium (IV)*. 2017, págs. 487-492. DOI: 10.1109/IVS.2017.7995765.

[83] Zhiyuan Wu et al. «S^3M-Net: Joint Learning of Semantic Segmentation and Stereo Matching for Autonomous Driving». En: *IEEE Transactions on Intelligent Vehicles* 9.2 (feb. de 2024), 3940–3951. DOI: 10.1109/tiv.2024.3357056.

[84] Zetong Yang et al. *Visual Point Cloud Forecasting enables Scalable Autonomous Driving*. 2023. arXiv: 2312.17655 [cs.CV].

BIBLIOGRAFÍA

Abid, Abubakar et al. «Gradio: Hassle-free sharing and testing of ml models in the wild». En: *arXiv preprint arXiv:1906.02569* (2019) (vid. pág. 36).

Afzal, Shahryar, Jiasi Chen y K. K. Ramakrishnan. «Viewing the 360° Future: Trade-Off Between User Field-of-View Prediction, Network Bandwidth, and Delay». En: *2020 29th International Conference on Computer Communications and Networks (ICCCN)*. 2020, págs. 1-11. DOI: 10.1109/ICCCN49398.2020.9209659 (vid. pág. 23).

Babak, Shahian-Jahromi et al. «Control of autonomous ground vehicles: a brief technical review». En: *IOP Conference Series: Materials Science and Engineering* 224.1 (2017), pág. 012029. DOI: 10.1088/1757-899X/224/1/012029 (vid. pág. 22).

Britten, Nicholas et al. «Do you trust me? Driver responses to automated evasive maneuvers». En: *Frontiers in Psychology* 14 (2023). DOI: 10.3389/fpsyg.2023.1128590 (vid. pág. 21).

Carion, Nicolas et al. «End-to-End Object Detection with Transformers». En: *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I*. Berlin, Heidelberg: Springer-Verlag, 2020, págs. 213-229. DOI: 10.1007/978-3-030-58452-8_13 (vid. págs. 13, 28, 29).

Chang, Ming-Fang et al. «Argoverse: 3D Tracking and Forecasting With Rich Maps». En: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, págs. 8740-8749. DOI: 10.1109/CVPR.2019.00895 (vid. pág. 14).

Dabbas, Elias. *Interactive Dashboards and Data Apps with Plotly and Dash: Harness the power of a fully fledged frontend web framework in Python–no JavaScript required*. Packt Publishing Ltd, 2021 (vid. pág. 36).

Deng, Jia et al. «Imagenet: A large-scale hierarchical image database». En: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. Ieee. 2009, págs. 248-255 (vid. pág. 12).

Dosovitskiy, Alexey et al. «An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale». En: *arXiv preprint arXiv:2010.11929* (2021). eprint: 2010.11929 (cs.CV) (vid. pág. 29).

Dumoulin, Vincent y Francesco Visin. «A guide to convolution arithmetic for deep learning». En: *arXiv preprint arXiv:1603.07285* (2016) (vid. pág. 15).

Fan, Rui et al. «Computer stereo vision for autonomous driving: Theory and algorithms». En: *Recent Advances in Computer Vision Applications Using Parallel Processing*. Springer, 2023, págs. 41-70 (vid. págs. 13, 23, 27).

Geiger, Andreas, Philip Lenz y Raquel Urtasun. «Are we ready for autonomous driving? The KITTI vision benchmark suite». En: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 2012, págs. 3354-3361. DOI: 10.1109/CVPR.2012.6248074 (vid. págs. 29, 37).

Gilroy, Shane, Edward Jones y Martin Glavin. «Overcoming Occlusion in the Automotive Environment—A Review». En: *IEEE Transactions on Intelligent Transportation Systems* 22.1 (2021), págs. 23-35. DOI: 10.1109/TITS.2019.2956813 (vid. pág. 29).

Guan, Tongfan, Chen Wang y Yun-Hui Liu. «Neural Markov Random Field for Stereo Matching». En: *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun. de 2024, 5459–5469. DOI: 10.1109/cvpr52733.2024.00522 (vid. págs. 32, 33).

He, Kaiming et al. «Deep residual learning for image recognition». En: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, págs. 770-778 (vid. pág. 29).

Iaco, Ryan De, Stephen L. Smith y Krzysztof Czarnecki. «Universally Safe Swerve Maneuvers for Autonomous Driving». En: *IEEE Open Journal of Intelligent Transportation Systems*. Vol. 2. 2021, págs. 482-494. DOI: 10.1109/OJITS.2021.3138953 (vid. págs. 12, 14).

Kelly, Alonzo y Anthony Stentz. «Rough Terrain Autonomous Mobility - Part 2: An Active Vision, Predictive Control Approach». En: *Auton. Rob.* 5 (jun. de 2000). DOI: 10.1023/A:1008822205706 (vid. pág. 23).

Kelly, Alonzo y Anthony (Tony) Stentz. «Rough Terrain Autonomous Mobility - Part 1: A Theoretical Analysis of Requirements». En: *Autonomous Robots* 5 (1998), págs. 129 -161 (vid. págs. 12, 23, 25, 62).

Kemsaram, Narsimlu, Anweshan Das y Gijs Dubbelman. «A Stereo Perception Framework for Autonomous Vehicles». En: *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. Antwerp, Belgium, 2020, págs. 1-6. DOI: 10.1109/VTC2020-Spring48590.2020.9128899 (vid. pág. 25).

— «Architecture Design and Development of an On-board Stereo Vision System for Cooperative Automated Vehicles». En: *2020 IEEE 23rd International Conference*

on Intelligent Transportation Systems (ITSC). 2020, págs. 1-8. DOI: 10.1109/ITSC45102.2020.9294435 (vid. pág. 25).

Khanam, Rahima y Muhammad Hussain. *YOLOv11: An Overview of the Key Architectural Enhancements*. 2024. arXiv: 2410.17725 [cs.CV] (vid. pág. 31).

Kim, Min-Joong et al. «On the Development of Autonomous Vehicle Safety Distance by an RSS Model Based on a Variable Focus Function Camera». En: *Sensors*. Vol. 21. 20. MDPI, 2021, pág. 6733. DOI: 10.3390/s21206733 (vid. pág. 13).

— «On the Development of Autonomous Vehicle Safety Distance by an RSS Model Based on a Variable Focus Function Camera». En: *Sensors* 21.20 (2021). DOI: 10.3390/s21206733 (vid. pág. 42).

Li, Stan Z. *Markov random field modeling in computer vision*. Springer Science & Business Media, 2012 (vid. pág. 32).

Liang, Jingyun et al. «Swinir: Image restoration using swin transformer». En: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, págs. 1833-1844 (vid. pág. 13).

Liao, Yiyi, Jun Xie y Andreas Geiger. «KITTI-360: A Novel Dataset and Benchmarks for Urban Scene Understanding in 2D and 3D». En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45 (2021), págs. 3292-3310 (vid. pág. 14).

Lv, Wenyu et al. *RT-DETRv2: Improved Baseline with Bag-of-Freebies for Real-Time Detection Transformer*. 2024. arXiv: 2407.17140 [cs.CV] (vid. pág. 29).

Masmoudi, Mehdi et al. «Object Detection Learning Techniques for Autonomous Vehicle Applications». En: *2019 IEEE International Conference on Vehicular Elec-*

tronics and Safety (ICVES). 2019, págs. 1-5. DOI: 10.1109/ICVES.2019.8906437 (vid. pág. 12).

Matthies, Larry H. y Pierrick Grandjean. «Stochastic performance, modeling and evaluation of obstacle detectability with imaging range sensors». En: *IEEE Transactions on Robotics and Automation* 10.6 (1994), págs. 783-792. DOI: 10.1109/70.338533 (vid. págs. 14, 15, 25).

Menze, Moritz y Andreas Geiger. «Object scene flow for autonomous vehicles». En: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, págs. 3061-3070 (vid. pág. 14).

P., Krishnendhu S. y Prabu Mohandas. «DETR-SPP: a fine-tuned vehicle detection with transformer». En: *Multimedia Tools and Applications* 83.9 (2024), págs. 25573-25594. DOI: 10.1007/s11042-023-16502-7.

Poggi, Matteo et al. «On the synergies between machine learning and binocular stereo for depth estimation from images: a survey». En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.9 (2021), págs. 5314-5334 (vid. pág. 25).

Qian, Jian et al. *SDformer: Efficient End-to-End Transformer for Depth Completion*. 2024. arXiv: 2409.08159 [cs.CV] (vid. págs. 46, 47).

Rankin, Arturo, Andres Huertas y Larry Matthies. «Stereo-vision-based terrain mapping for off-road autonomous navigation». En: *Proc. SPIE 7332, Unmanned Systems Technology XI*. SPIE, 2009, pág. 733210. DOI: 10.1117/12.819099 (vid. págs. 13, 15).

Rezatofighi, Hamid et al. *Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression*. 2019. arXiv: 1902.09630 [cs.CV] (vid. pág. 52).

Ros, German et al. «The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes». En: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, págs. 3234-3243 (vid. pág. 14).

Rydzewski, Aleksander y Pawel Czarnul. «Human awareness versus Autonomous Vehicles view: comparison of reaction times during emergencies». En: *2021 IEEE Intelligent Vehicles Symposium (IV)*. 2021, págs. 732-739. DOI: 10.1109/IV48863.2021.9575602 (vid. págs. 18, 42).

Schneider, Anne et al. «Sensor study for high speed autonomous operations». En: *Proc. SPIE 9494, Next-Generation Robotics II; and Machine Intelligence and Bio-inspired Computation: Theory and Applications IX*. SPIE, 2015, pág. 949408. DOI: 10.1117/12.2176596 (vid. págs. 14, 18-20).

Sibley, Gabe, Larry Matthies y Gaurav Sukhatme. «Bias Reduction and Filter Convergence for Long Range Stereo». En: *Robotics Research*. Ed. por Sebastian Thrun, Rodney Brooks y Hugh Durrant-Whyte. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, págs. 285-294 (vid. pág. 25).

Szeliski, Richard. «Computer vision: algorithms and applications». En: *Springer Nature*. 2022 (vid. pág. 12).

Terven, Juan, Diana-Margarita Córdova-Esparza y Julio-Alejandro Romero-González. «A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS». En: *Machine Learning and Knowledge Ex-*

traction 5.4 (nov. de 2023), 1680–1716. DOI: 10.3390/make5040083 (vid. págs. 28, 31).

Tian, Katherine et al. *Fine-tuning Language Models for Factuality*. 2023. arXiv: 2311.08401 [cs.CL] (vid. pág. 53).

Vaswani, Ashish et al. «Attention is All you Need». En: *Advances in Neural Information Processing Systems*. Ed. por I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017 (vid. págs. 15, 29).

Voulodimos, Athanasios et al. «Deep Learning for Computer Vision: A Brief Review». En: *Intell. Neuroscience* 2018 (2018). DOI: 10.1155/2018/7068349 (vid. pág. 12).

Wenkel, Simon et al. «Confidence Score: The Forgotten Dimension of Object Detection Performance Evaluation». En: *Sensors* 21.13 (2021). DOI: 10.3390/s21134350 (vid. pág. 37).

Wu, Zhiyuan et al. «S^3M-Net: Joint Learning of Semantic Segmentation and Stereo Matching for Autonomous Driving». En: *IEEE Transactions on Intelligent Vehicles* 9.2 (feb. de 2024), 3940–3951. DOI: 10.1109/tiv.2024.3357056 (vid. pág. 62).

Xie, Yun, Shaowu Zheng y Weihua Li. «Feature-Guided Spatial Attention Upsampling for Real-Time Stereo Matching Network». En: *IEEE MultiMedia* 28.01 (2021), págs. 38-47. DOI: 10.1109/MMUL.2020.3030027 (vid. pág. 27).

Xu, Yuquan et al. «3D point cloud map based vehicle localization using stereo camera». En: *2017 IEEE Intelligent Vehicles Symposium (IV)*. 2017, págs. 487-492. DOI: 10.1109/IVS.2017.7995765 (vid. pág. 62).

Yacouby, Reda y Dustin Axman. «Probabilistic Extension of Precision, Recall, and F1 Score for More Thorough Evaluation of Classification Models». En: *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*. Ed. por Steffen Eger et al. Online: Association for Computational Linguistics, nov. de 2020, págs. 79-91. DOI: 10.18653/v1/2020.eval4nlp-1.9 (vid. pág. 52).

Yang, Guorun et al. «DrivingStereo: A Large-Scale Dataset for Stereo Matching in Autonomous Driving Scenarios». En: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019 (vid. pág. 14).

Yang, Zetong et al. *Visual Point Cloud Forecasting enables Scalable Autonomous Driving*. 2023. arXiv: 2312.17655 [cs.CV] (vid. pág. 62).

Yuan, Kui et al. «Field of View and Its Design for the Autonomous Driving System of Tram». En: *2021 International Conference on Information Control, Electrical Engineering and Rail Transit (ICEERT)*. 2021, págs. 236-239. DOI: 10.1109/ICEERT53919.2021.00052 (vid. pág. 22).

Zaarane, Abdelmoghit et al. «Distance measurement system for autonomous vehicles using stereo camera». En: *Array* 5 (2020), pág. 100016. DOI: https://doi.org/10.1016/j.array.2020.100016 (vid. pág. 24).

Zhang, Benqi et al. «Simultaneous improvement of field-of-view and resolution in an imaging optical system». En: *Opt. Express* 29.6 (2021), págs. 9346-9362. DOI: 10.1364/OE.420222 (vid. pág. 23).

Zhu, Xiangyuan et al. «Cross View Capture for Stereo Image Super-Resolution». En: *IEEE Transactions on Multimedia* 24 (2022), págs. 3074-3086. DOI: 10.1109/TMM.2021.3092571 (vid. pág. 23).