

Predicción del ángulo de giro para un auto: Implementación de dos áreas de interés Car steering prediction: Implementation of two interest areas

F. E. Martínez-Hernández ^a, J. A. Arias-Aguilar ^{a,*}, E. Macías-García ^b, O. D. Ramírez-Cárdenas ^a

^a División de Estudios de Posgrado, Universidad Tecnológica de la Mixteca, 69000, Huajuapán de León, Oaxaca, México.

^b Intelligent System Research, Intel Labs, 45017, Zapopan, Jalisco, México.

Resumen

La predicción del ángulo de dirección en vehículos inteligentes es un tema ampliamente estudiado en la literatura. Sin embargo, se requiere la intervención de un usuario para operar al vehículo, limitando el grado de autonomía. Este trabajo propone algoritmos basados en redes neuronales artificiales para determinar el ángulo de giro de un vehículo en movimiento, con aplicaciones en tiempo real para vehículos a escala en una pista de dos carriles. Utilizando una cámara frontal centrada en el vehículo se determinan dos áreas de interés: la imagen original y una versión rectificadas mediante un cambio de perspectiva. Posteriormente, se emplea la tupla de imágenes para construir dos bases de datos etiquetadas con el ángulo deseado de giro, las cuales, son empleadas para entrenar dos redes neuronales convolucionales. Una vez entrenadas, ambos modelos son implementados en tiempo real para guiar un vehículo a escala en una pista libre de obstáculos, permitiendo su navegación autónoma dentro del carril designado y observando el comportamiento en cada caso.

Palabras Clave: Aprendizaje supervisado, área de interés, CNN, sistema de dirección, vehículos inteligentes.

Abstract

Steering angle prediction in intelligent vehicles is a topic widely studied in the literature. Nevertheless, user intervention is required to operate the vehicle, limiting the degree of autonomy. This work proposes algorithms based on artificial neural networks to determinate the steering angle in a moving vehicle, with real-time applications in scaled vehicles on a two-lane track. Using a front camera centered on the vehicle, two areas of interest are determined: the original image and a rectified version by changing the perspective. Subsequently, the tuple of images is used to create two labelled datasets with the desired steering angle to train two convolutional neural networks. Once trained, both models are implemented in real-time to guide a scaled vehicle on an obstacle free road, allowing autonomous navigation in the designated lane while observing the behaviour in each case.

Keywords: Supervised learning, region of interest, CNN, steering system, intelligent vehicles.

1. Introducción

En el contexto de la conducción autónoma, existen aún áreas de oportunidad para desarrollar algoritmos que brinden a los vehículos la capacidad de ser completamente autónomos. En este sentido, de acuerdo con (Stöckle, Herrmann, Dirndorfer, & Utschick, 2020), los Sistemas Avanzados de Asistencia a la Conducción (ADAS, por su sigla en inglés) permiten al vehículo desarrollar diferentes tipos de tareas individuales tales como la detección de peatones, la activación del freno en casos de emergencia, la detección de ángulo de dirección, o el control de cruceo adaptativo. De manera similar (Brown, 2017), el brindar autonomía a los vehículos permite mejorar la mecánica de la conducción en diversas

situaciones; sin embargo, el problema se encuentra en que dicha conducción suele limitarse en evitar colisiones aisladas del vehículo usando la información disponible de los sensores, lo cual no siempre es la mejor decisión dadas las circunstancias del entorno. Por otro lado, la investigación de (Díaz, 2020) establece que la navegación autónoma requiere capturar la mayor cantidad de información posible del entorno que rodea al vehículo, a fin de tomar decisiones basadas en aspectos de seguridad, eficiencia y condiciones del entorno.

En el contexto académico y de investigación también se han desarrollado contribuciones importantes para el área, tal es el caso del proyecto BRAiVE desarrollado en la Universidad de Parma (Broggi, et al., 2015), a partir del cual se creó un sistema de navegación que permitió a un vehículo conducir de forma

*Autor para correspondencia: anibal@mixteco.utm.mx

Correo electrónico: filiberto.martinez133@gmail.com (Filiberto Eduardo Martínez-Hernández), anibal@mixteco.utm.mx (José Anibal Arias-Aguilar), edgar.macias.garcia@intel.com (Edgar Macías-García), odramirez@mixteco.utm.mx (Oscar David Ramírez-Cárdenas).

Historial del manuscrito: recibido el 13/12/2023, última versión-revisada recibida el 13/12/2023, aceptado el 09/03/2024, publicado el 22/04/2024. **DOI:** <https://doi.org/10.29057/icbi.v12iEspecial2.12260>



autónoma por 13 kilómetros. El vehículo contaba con dos escáneres láser montados en la parte delantera del vehículo, mientras que el sistema de posicionamiento empleaba un dispositivo de navegación cinética satelital; pese a brindar autonomía, fue necesario que un conductor apoyara en algunas tareas que el sistema no podía realizar por sí solo. Por otra parte, de acuerdo con (Bojarski, et al., 2016), NVIDIA logró desarrollar una red neuronal profunda que permite guiar y predecir el ángulo de dirección para la navegación autónoma, a la par de reconocer objetos que se encuentran en el camino; vehículos, postes e incluso objetos a los costados.

En el ámbito de hardware, la Universidad Libre de Berlín, Alemania, introdujo AutoNOMOS (AutoNOMOS, 2017) como plataforma física de un automóvil a escala 1:10, el cual incorporaba un modelo para simulación con propiedades similares, tanto físicas como de adquisición de datos y diseño de un auto real. Contaba con una computadora de 64 GB para el sistema y almacenamiento, un servomotor para lograr el ángulo de dirección, un motor sin escobillas para la tracción, un sensor IMU para la orientación, una cámara Intel 3D y un sensor LiDAR.

En México se han desarrollado proyectos sobre esta plataforma, como el trabajo desarrollado en (Bravo, 2018), donde se propone una metodología para implementar la navegación autónoma y evasión de obstáculos, la cual consiste en operaciones morfológicas para búsqueda de máximos locales que detecten puntos sobre la línea de carril cuantificados en un histograma desarrollado a partir de la imagen, siendo eficaz para diferentes condiciones de iluminación. Así mismo, la Universidad Tecnológica de la Mixteca (Díaz, 2020), desarrolló un algoritmo para la reconstrucción de mapas métricos generados a partir de la cámara Intel Real Sense colocada en la parte frontal superior del AutoNOMOS Mini V2.

De acuerdo con (Sandoval, 2018), el aprendizaje máquina es un área de la Inteligencia Artificial que busca generar algoritmos cuya capacidad sea la de aprender sin tener que programar explícitamente. (Calvo, Guzmán, & Ramos, 2018) describe dos tipos de aprendizaje: el aprendizaje supervisado y el aprendizaje no supervisado. El aprendizaje supervisado, dependiendo del problema, se puede dividir en clasificación o regresión, y en este caso se involucra la regresión para buscar una correlación a datos que se desea generar entre ellos (Moya, 2016). Para el caso particular del procesamiento de imágenes, las Redes Neuronales Convolucionales (CNN, por sus siglas en inglés) representan el estado del arte en lo que concierne a tareas de regresión y clasificación con imágenes.

En consecuencia, y con el objetivo de desarrollar y comparar las respuestas de un vehículo autónomo, este artículo presenta el desarrollo de algoritmos para el prototipo AutoNOMOS Mini V2. Dichos algoritmos están creados para realizar la conducción autónoma en un entorno controlado, por medio de la estimación del ángulo de dirección necesario para mantener al vehículo en movimiento en un carril designado. El resto del artículo se organiza de la siguiente manera: En la Sección II se desarrollan métodos y algoritmos para la adquisición de datos y el entrenamiento de dos modelos para la estimación del ángulo de dirección para un circuito de dos carriles. La Sección III presenta los resultados experimentales obtenidos con los modelos entrenados, mientras que en la Sección IV y V se presentan la discusión de los resultados y las conclusiones del presente trabajo.

2. Materiales y métodos

En este estudio se consideró el trabajo de (Hernández, Fernández, & Baptista, 2014) para desarrollar e implementar algoritmos de navegación autónoma. Para la adquisición de datos, se consideraron imágenes obtenidas en el prototipo físico y en simulación a través de un ambiente virtual en ROS Gazebo usando una cámara RGBD. A fin de obtener una perspectiva correcta de la escena, se decidió situar la cámara a una altura de 0.15 m sobre el suelo, y un ángulo de 20° negativos a partir de la horizontal, como se puede observar en la Figura 1.

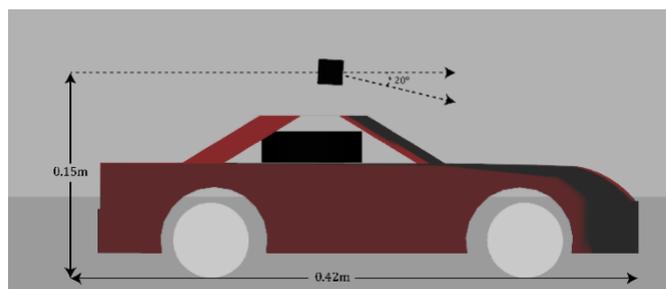


Figura 1: Configuración de la cámara de profundidad en el prototipo simulado en Gazebo.

El entorno que se eligió dentro del simulador fue una pista con 4 curvas y 2 carriles; 2 líneas continuas a las orillas para delimitar los carriles y una línea punteada central para dividir ambos sentidos. La Figura 2 muestra una vista superior de esta pista, cuyas medidas, además, son de 12 m por 9 m (cada cuadro equivalente a 1 m).

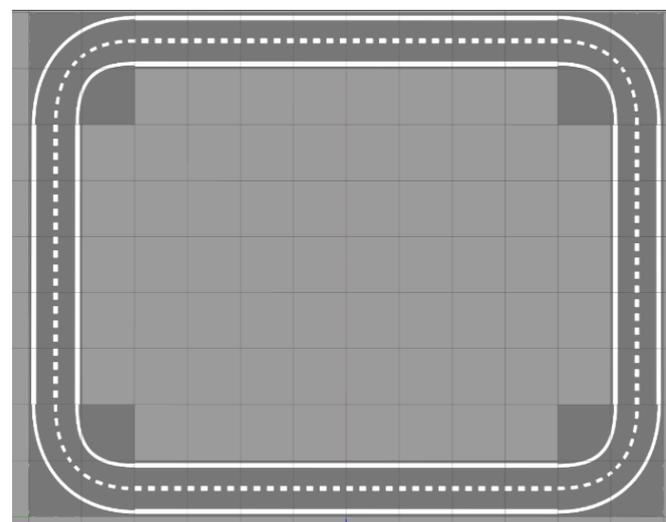


Figura 2: Vista superior de la pista en simulador.

Con el fin de optimizar el procesamiento de datos a través de las imágenes, fue necesario plantear dos áreas de interés (ROI, por sus siglas en inglés); la primera es la imagen original recibida por la cámara (Figura 3), mientras que la segunda se obtiene a través de un cambio de perspectiva que permite enfocarse en la región a nivel de piso, a la par de rectificar las líneas de los carriles (Figura 4).

La Figura 4a muestra que el área de interés se seleccionó de tal manera que pudiera ser visible tanto la línea punteada del centro que divide a los carriles como la línea continua que

limita la pista. La selección de esta área de interés se realizó con el fin de eliminar información innecesaria en la imagen, como pudiera ser el fondo que se observa en dicha figura. Por otro lado, la Figura 4b muestra el resultado de aplicar el cambio de perspectiva.



Figura 3: Imagen obtenida por la cámara de profundidad a través del simulador.

Para este caso fue necesario utilizar una transformación proyectiva en dos dimensiones. De acuerdo con (Gonzales & Woods, 2010), dicha proyección puede describirse mediante la ecuación (1) La primera matriz está representada por una matriz de similitud de 4 Grados de Libertad (GDL), la segunda por una matriz simple de corte de un GDL, la tercera es una matriz diagonal de un GDL que escala las direcciones “x” y “y” por λ y $1/\lambda$, respectivamente, mientras que la última matriz contiene un vector de elasticidad de dos GDL.

$$H = \begin{bmatrix} sR & t \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} k & 0 \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} \lambda & 0 \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} I & 0 \\ V^T & 1 \end{bmatrix}. \quad (1)$$



(a) Selección de área de interés en rojo. (b) Resultado de aplicar cambio de perspectiva.

Figura 4: Aplicación del cambio de perspectiva.

Considerando el enfoque del aprendizaje supervisado, se crearon dos bases de datos con imágenes etiquetadas (i.e., una para cada área de interés) con la información del ángulo de dirección para mantener al vehículo en un carril designado. En consecuencia, se recolectaron y almacenaron imágenes a una frecuencia de 30 Hz. (i.e., frecuencia de actualización de la cámara), etiquetadas con el ángulo de dirección adquirido a través de un mando de videojuegos (Figura 5), donde el potenciómetro L daba la velocidad del modelo y el potenciómetro R el ángulo de dirección.



Figura 5: Mando utilizado para el entrenamiento y creación de la base de datos.

La Figura 6 muestra que el ángulo de dirección máximo que el prototipo podía alcanzar en la simulación se encuentra entre -22° y 22° . Sin embargo, dado el formato necesario para utilizar los ángulos en el prototipo real, estos datos fueron convertidos a cantidades de Modulación por Ancho de Pulso (PWM, por sus siglas en inglés), que fueron normalizados posteriormente en el rango entre 0 y 1, donde 0 conservaba su valor y el valor máximo 180 correspondía a un valor unitario. Mientras que la velocidad máxima del modelo era de 0.377 metros por segundo (m/s).

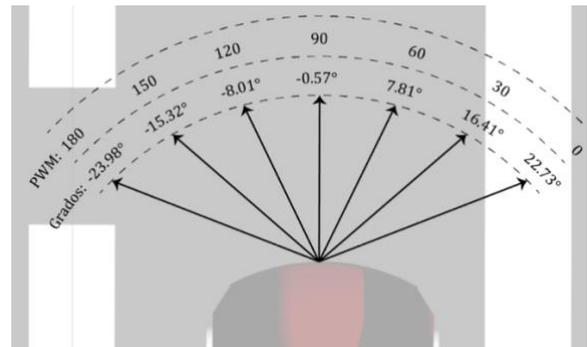


Figura 6: Ángulo de dirección obtenido del modelo.

Una vez que las imágenes fueron adquiridas y etiquetadas, se generaron dos bases de datos; la primera con las imágenes nominales de la cámara sin cambios (ROI 1), mientras que la segunda (ROI 2) con la transformación proyectiva. La Tabla 1 señala las frecuencias de actualización para cada dato, así como el total de imágenes guardadas por cada área de interés y, en consecuencia, las imágenes guardadas para cada carril del circuito (carril interno y externo).

Una vez que las bases de datos contenían un banco suficiente de imágenes (como lo muestra la Tabla 1), se separaron en conjuntos de entrenamiento y validación, a fin de entrenar un modelo de regresión para recibir como entrada la imagen adquirida por la cámara y generar como salida la predicción del ángulo de giro. Por lo tanto, se propuso que el conjunto de entrenamiento fuese del 90% del total de imágenes en ambas bases de datos, mientras que el 10% restante formó parte del conjunto de validación. En consecuencia, el conjunto de entrenamiento para la primera base de datos fue de 9'464 datos, mientras que, para la segunda de 7'484, por lo tanto, la evaluación constó de 1'051 y 831 datos, respectivamente.

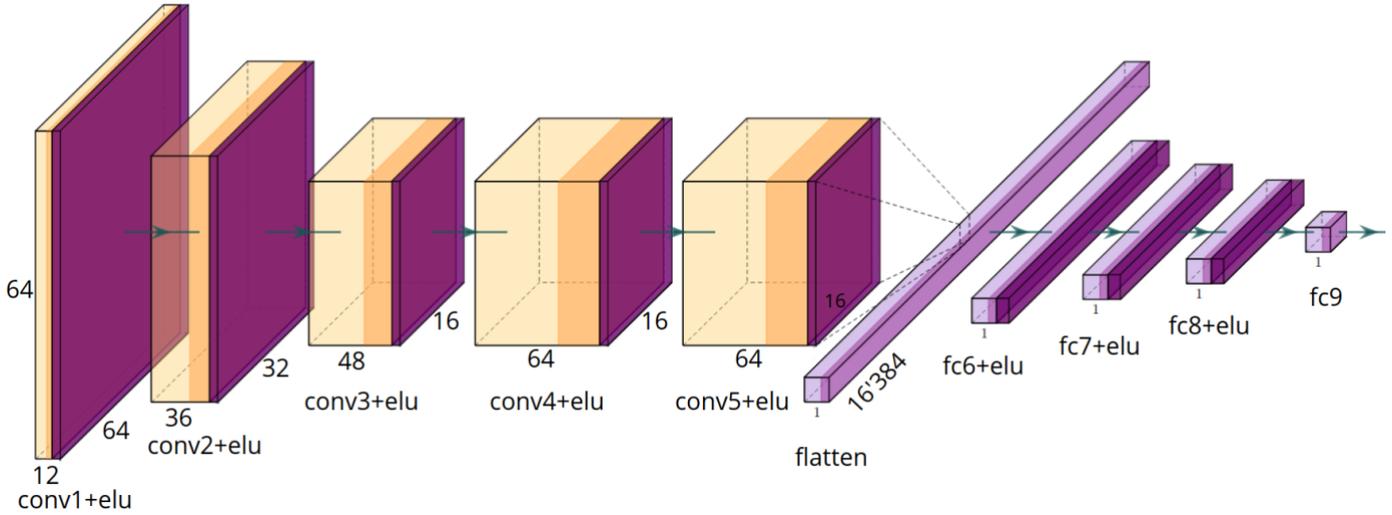


Figura 7: Topologías propuestas para la CNN.

Tabla 1: Características de las bases de datos

Opción	Frecuencia de actualización	Variables de la base de datos
Imagen nominal	30 Hz	-
Área de interés 1 (ROI 1)	20 Hz	10'515
Carril interno		4'956
Carril externo		5'559
Ángulo de dirección ROI 1	20 Hz	10'515
Área de interés 2 (ROI 2)	20 Hz	8'315
Carril interno		4'087
Carril externo		4'228
Ángulo de dirección ROI 2	20 Hz	8'315

La estructura de una red neuronal convolucional se organiza en planos que se denominan mapas de características. En el contexto de este estudio se propuso una topología de red neuronal convolucional completamente conectada con la forma mostrada en la Figura 7, que cuenta con 5 capas convolucionales (conv), un aplanado de los datos (flatten), 4 capas ocultas completamente conectadas (fc) y una salida (la única sin función de activación elu).

Las características de esta topología son descritas en la Tabla 2, considerando como entrada las imágenes correspondientes a alguna de las zonas de interés, y como salida el ángulo de dirección. Si bien la topología es la misma para ambas bases de datos, la Tabla 2 describe la cantidad de neuronas usadas para cada capa y por cada base de datos.

Tabla 2: Datos de capas internas para las topologías de las CNN

Neuronas en capas internas	fc6+elu	fc7+elu	fc8+elu	fc9
ROI 1	13	8	5	1
ROI 2	10	5	3	1

Para entrenar los modelos se empleó la función de costo denominada error cuadrático medio (MSE, por sus siglas en inglés), el cual está definido por la ecuación (2), donde M es el número total de datos.

Por cada iteración se obtiene la diferencia entre la estimación del modelo y la predicción deseada, a fin de emplear el gradiente de la función de costo para ajustar los pesos del modelo.

$$MSE = \frac{1}{M} \sum_{i=1}^M (real_i - estimado_i)^2. \quad (2)$$

3. Resultados experimentales

Tomando en cuenta las consideraciones anteriores, se obtuvieron datos correspondientes a cada área de interés. En este aspecto, la Figura 8 representa los datos recogidos después de entrenar las topologías de las redes para diferentes épocas (i.e., eje x), así como su correspondiente error cuadrático medio o, dicho de otra manera, su pérdida (i.e., eje y).

Esta información está dividida en dos partes: usando la imagen original adquirida por la cámara (ROI 1), y la imagen con transformación proyectiva (ROI 2). Cada área de interés y cada época fue evaluada contra un banco de imágenes que no se utilizó para el entrenamiento. Dicho repositorio corresponde al 10% del banco de imágenes total para cada área de interés descrita en la Tabla 1.

Como puede observarse en la Figura 8, el punto de sobreajuste fue diferente para ambos casos: Para la base de datos con la imagen completa (ROI 1), por ejemplo, fueron necesarias 23 épocas, mientras que para el área de interés (ROI 2) fueron necesarias únicamente 7.

El menor error alcanzado entre las dos áreas de interés (en términos del PWM) se logró con el cambio de perspectiva (ROI 2), con un valor de 0.0162 para entrenamiento y de 0.014 para la validación.

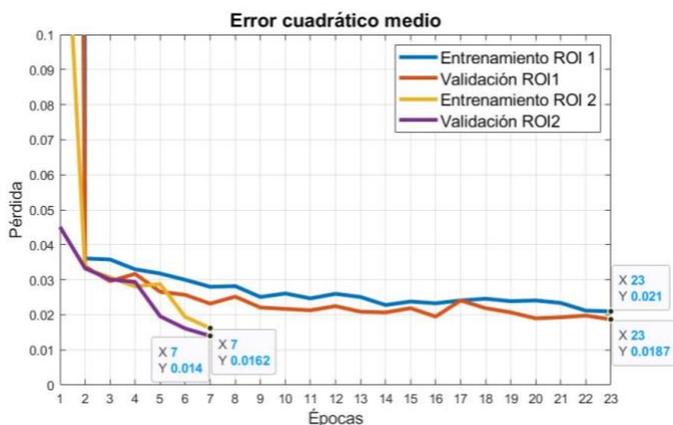


Figura 8: Gráfica de pérdida para las áreas de interés.

Por otro lado, la

Figura 9 muestra la trayectoria y el comportamiento del modelo al pasar por cada una de las cuatro esquinas en su respectivo carril, siendo la trayectoria de color rojo el ROI 1 y la trayectoria azul con el ROI 2. Aunado a esto, la Figura 10 y la Figura 11 muestran las gráficas que describen la predicción de las dos CNN para cada carril (interno y externo) en una vuelta a la pista, en ellas se muestra el ángulo de dirección a partir del cual se hizo el entrenamiento (línea negra), mientras que de color rojo y azul se muestran las predicciones de las CNN (ROI 1 y ROI 2 respectivamente).

En este mismo orden de ideas, también es necesario resaltar que al inicio el entrenamiento se realizó con imágenes correspondientes a un solo carril; sin embargo, la falta de información del carril restante hacía que el modelo perdiera el control en los tramos largos de la pista y, si bien la red sí aprendía, ésta se confundía, por lo que al agregar la información de ambos carriles este error desapareció.

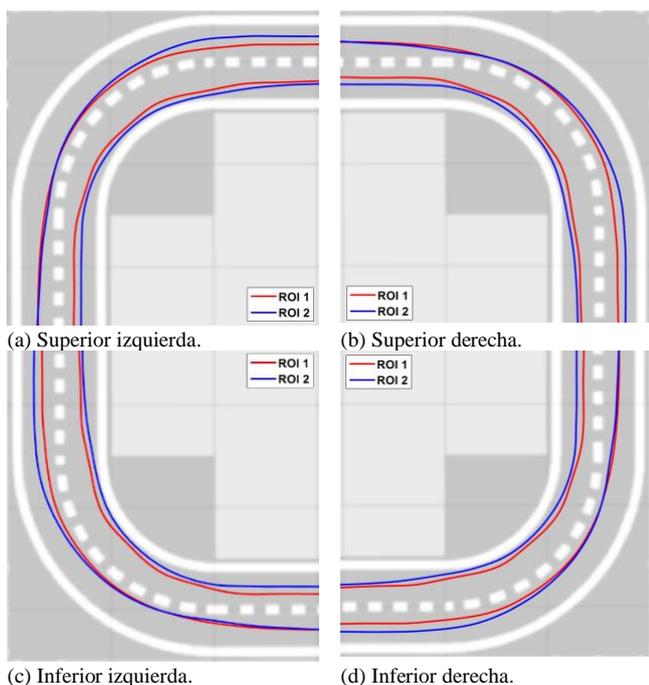


Figura 9: Gráfica de pérdida para las áreas de interés.

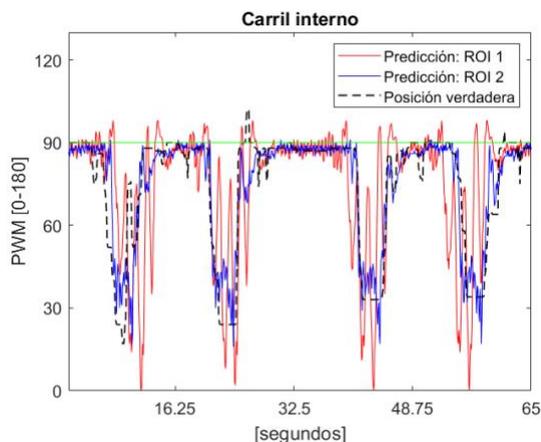


Figura 10: Gráfica de la predicción de los modelos para el ángulo de dirección en el carril interno.

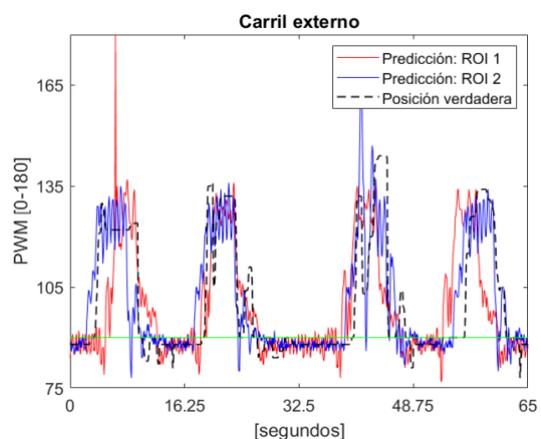


Figura 11: Gráfica de la predicción de los modelos para el ángulo de dirección en el carril externo.

4. Discusión

Los resultados muestran que por cada iteración el error disminuye, lo que demuestra que la red neuronal está aprendiendo a tomar la decisión de asignar un ángulo de dirección por cada imagen mostrada que es extraída de la cámara, sin importar el área de interés. Del mismo modo, es notorio que el entrenamiento utilizando el área de interés con el cambio de perspectiva (ROI 2) logró el menor error que se pudo obtener con un menor número de iteraciones. Lo cual, puede deberse a que dicha representación contiene una menor cantidad de información irrelevante en comparación con la primera representación (ROI 1), al enfocarse únicamente en la región a nivel de piso.

Como puede verse en la Figura 9, el carril interno utilizando el ROI 1 tiende a hacer movimientos oscilatorios, que a diferencia del ROI 2, este tiene un movimiento más suave. Aunado a esto, la Figura 10 y la Figura 11 muestran que el ROI 2 tiende a tener una mayor correlación a la referencia que la predicción del ROI 1. Aun cuando es evidente que el cambio de perspectiva (ROI 2) es mejor para predecir el ángulo, puede no siempre ser la mejor opción en función de las cualidades deseadas de navegación. Ya que la información presente por encima del nivel de piso puede ser útil para agregar cualidades al modelo, como puede ser la identificación de señales, cambio de carril, o simplemente para orientarse.

Es importante mencionar que los resultados mostrados en este estudio tienen cierta similitud con el trabajo relacionado en el área, a pesar de que las soluciones propuestas son un tanto diferentes. Por ejemplo, la investigación presentada en (Du, Dung, & Hai, 2020) utiliza un robot diferencial real para predecir el ángulo de dirección mediante la transformada de Hough. Sin embargo, y a diferencia del estudio presentado en este artículo, no se utiliza la línea divisoria entre carriles puesto que solamente se utiliza un carril. En este sentido, a pesar de que se ha presentado una solución simulada, se ha podido comprobar que es posible realizar la predicción sin hacer uso de una transformada de Hough, así como extrapolar el modelo a un robot tipo Ackerman.

De manera similar, la investigación presentada en (Thieng, Dong, & Ha, 2018) introduce un entorno de simulación que también utiliza un único carril. En este caso la red no solamente predice el ángulo de dirección, sino que también la velocidad, lo que resulta útil para la navegación durante las curvas. Es importante hacer notar que, a pesar de que no se menciona el valor del error obtenido, los resultados obtenidos son similares a los presentados en este artículo. Por otro lado, los resultados aquí presentados son preliminares y de ninguna manera pueden generalizarse para otras evaluaciones similares. De modo que, se continuará con el entrenamiento de la red neuronal con el objetivo de incrementar los datos para complementar a otro tipo de robots o dispositivos.

Material multimedia adicional referente a la experimentación realizada se puede consultar por medio del siguiente enlace:

https://drive.google.com/drive/folders/1r940ajdC8Sk95gVbeURsG9GsuG_fQJe.

5. Conclusiones

Uno de los principales problemas para el aprendizaje profundo es la selección de características para el aprendizaje de un modelo, es por ello que en este trabajo se propusieron dos áreas de interés para la estimación del ángulo de dirección requerido para mantener a un vehículo dentro de un carril designado; la primera, considerando la imagen original adquirida a través de una cámara frontal (ROI 1), y la segunda, a través de un cambio de perspectiva de la misma imagen (ROI 2). Los resultados experimentales demuestran que la segunda representación permite al modelo generalizar con una menor cantidad de iteraciones y un mejor resultado cuantitativo.

Como trabajo futuro se considerará introducir algoritmos de evasión de obstáculos adicionales tales como la evasión de

otros autos u otro tipo de objetos en el camino, pistas con diferentes tipos de características, y otros tipos de modelos de aprendizaje como las redes recurrentes. Finalmente, también se podrían proponer modelos basados en autos con diferente con diferentes tipos de instrumentación, así como el aumento de datos por medio de otro tipo de sensores como LiDAR.

Agradecimientos

Al Consejo Nacional de Humanidades, Ciencia Y Tecnología (CONAHCYT), por el apoyo económico otorgado durante la realización de este artículo.

Referencias

- AutoNOMOS. (2017). *AutomodelCar*. Obtenido de <https://github.com/AutoModelCar/AutoModelCarWiki/wiki/Hardware>
- Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Zhang, J. (2016). End to end learning for self-driving cars. *arXiv:1604.07316*.
- Bravo, C. (2018). *Navegación autónoma de un robot tipo automóvil en pista de carreras con obstáculos*. Ciudad de México: Tesis de Maestría en Ciencias e Ingeniería de Computo, Instituto Politécnico Nacional.
- Broggi, A., Cerri, P., Debattisti, S., Chiara, M., Panciroli, M., y Prioletti, A. (2015). Proud—public road urban driverless-car test. *IEEE In Transactions on intelligent transportation systems*, volumen 16, 3508-3519.
- Brown, B. (2017). The social life of autonomous cars. *IEEE In Computer*, volumen 50, 92-96.
- Calvo, J., Guzmán, M., y Ramos, D. (2018). Machine learning, una pieza clave en la transformación de los modelos de negocio. *Management Solutions*, 20-22.
- Diaz, E. (2020). *Generación del mapa métrico de una pista de competencias para la navegación del robot móvil autonomos mini V2*. Huajuapán de León: Tesis de maestría en Robótica, Universidad Tecnológica de la Mixteca.
- Du, K., Dung, K., y Hai, T. (2020). Vision based steering angle estimation for autonomous vehicles. *International Conference on Advanced Technologies for Communications (ATC)* (pp. 187-192). IEEE.
- Gonzales, R., y Woods, R. (2010). *Procesamiento Digital de imágenes*, 3ed. Pearson.
- Hernández, R., Fernández, C., y Baptista, P. (2014). *Metodología de la investigación*, 6ta Edición. Mc Graw-Hill.
- Martínez, J. (2021). Error cuadrático medio para regresión. Obtenido de <https://www.iartificial.net/error-cuadratico-medio-para-regresion/>
- Moya, R. (2016). *jarroba*. Obtenido de <https://jarroba.com/que-es-el-clustering/>
- Sandoval, L. (2018). Algoritmos de aprendizaje automático para análisis y predicción de datos. *Revista Tecnológica*, volumen 11, 36–40.
- Stöckle, C., Herrmann, S., Dirndorfer, T., y Utschick, W. (2020). Automated Vehicular Safety Systems: Robust function and sensor design. *IEEE Signal Processing: Autonomous driving*, 24-32.
- Thieng, M., Dong, T., y Ha, M. (2018). Navigating Self-Driving Vehicles Using Convolutional Neural Network. *4th International Conference on Green Technology and Sustainable Development (GTSD)* (pp. 607-610). IEEE.