Reconocimiento de objetos en entorno urbano para la conducción autónoma

Juan Pablo González M.
División de Ingenierías
Universidad de Guanajuato
Salamanca, Gto, México
jp.gonzalezmendoza@ugto.mx

Felipe Trujillo R.
División de Ingenierías
Universidad de Guanajuato
Salamanca, Gto, México
fdj.trujillo@ugto.mx

***Abstract*—En este artículo se presenta la implementación de un sistema de reconocimiento de objetos en un entorno urbano. Se reconocen cuatro tipos de clases diferentes, las cuales son: peatón, señal de alto, señal de límite de velocidad y autómovil. El objetivo de reconocer estos elementos es para dotar a un vehículo autónomo de la capacidad de identificar y reaccionar ante ellos para reducir los accidentes viales en este tipo de vehículos. Para realizar el sistema de reconocimento se hace uso del modelo YOLO versión 5. Se utilizaron 100 imágenes por clase con 90 para entrenamiento y 10 para clasificación. Con esta implementación se obtuvo una tasa de reconocimiento del 89%.**

***Index Terms*—Reconocimiento de objetos, vehículos autónomos, YOLOv5, accidentes viales, visión por computadora.**

I. Introducción

La visión por computadora es un área interesante la cual se puede aplicar en múltiples campos de estudio. Actualmente existe un auge en el uso de la visión por computadora con su introducción en los vehículos autónomos (VA). Sin embargo, este desarrollo no ha sido tarea fácil ya que si hacemos un poco de historia, el primer prototipo se ha estado desarrollando desde el año de 1939 donde en una feria se presento el primer caso de VA que funcionaba con circuitos eléctricos. Desde aquella época se observó los altos niveles de cálculo que se debían de realizar para efectuar un manejo óptimo y seguro. No fue hasta el año de 1994 donde se realizaron experimentos con dos automóviles en París que recorrieron una larga distancia sin la intervención de un piloto [1]. Pero esta navegación autónoma es solo una parte del problema a resolver por un vehículo autónomo. La otra parte es evitar que haya accidentes de transito ya que la Organización Mundial de la Salud (OMS) estima que cada año mueren en el mundo cerca de 1,3 millones de personas en accidentes de tránsito. Debido a estos accidentes, entre 20 y 50 millones padecen traumatismos no mortales causantes de discapacidad. Los accidentes viales, además, constituyen una de las principales causas de mortalidad, principalmente entre personas de entre 15 y 19 años [2]. Actualmente con el avance tecnológico y con la introducción a la inteligencia artificial, los VA se perfilan como una nueva clase de transportes donde su tendencia en las conductas viales será un gran impacto ya que mejorará el porcentaje de error humano, sin embargo, no es perfecto y los accidentes viales estarán siempre presentes y cualquiera podrá ser vulnerable por el comportamiento de múltiples factores en tiempo real y he aquí la importancia de reducir la probabilidad de riesgo en este ambiente.

II. Antecedentes

En el rubro de la seguridad vial los datos que genera la Policía Federal de México, la Comisión Nacional de Seguridad (CNS) reporta que las causas de los accidentes en las carreteras federales, alrededor del 80% de las veces se deben al conductor, 7% al vehículo, 9% a los agentes naturales y solo el 4% al camino [3]. En comparación con otro estudio respecto a la seguridad vial [4] demuestra que el error humano fue la única causa en el 57% de todos los accidentes viales y además contribuye a estos en más del 90%. Por el contrario, solamente el 2.4% se debió a fallas mecánicas y el 4.7% se estima a factores ambientales, en ambos casos afirman que el humano contiene el mayor porcentaje de error en este tipo de accidentes.

Las principales causas de accidentes de tránsito son las siguientes:

- Conducir bajo los efectos del alcohol, medicinas y estupefacientes.
- Realizar maniobras imprudentes y de omisión por parte del conductor, por ejemplo; no respetar los señalamientos viales
- Conducir a exceso de velocidad (produciendo vuelcos, salida del automóvil de la carretera, derrapes).
- Salud física del conductor (ceguera, daltonismo, sordera).
- Conducir con fatiga, cansancio o con sueño [3].

Dentro de una proyección de la industria tecnológica respecto a los VA, se estima que en los próximos 10 años hasta que estos automóviles se popularicen en el mercado, la reducción de gases de efecto invernadero será entre 2 y 4 puntos porcentuales, los ciudadanos viajarán más seguros mientras podrán realizar otras tareas dentro de los vehículos cuando circulan, además aumentará la movilidad de los ciudadanos. Las ciudades necesitaran nuevos planes urbanísticos al disponer de más espacios, las carreteras también se verán afectadas, se podrán transportar personas con capacidades reducidas, en definitiva, más calidad de vida para las personas [5].

La organización SAE international (Society of Automotive Engineers) describe 6 niveles para la consideración a una automatizaciǿn de vehículos. Los primeros tres niveles dependen de un conductor donde su aumento se caracteriza por la asistencia de monitoreo para su buen manejo, los últimos tres niveles se refieren a un sistema autónomo en el cual el cinco se refiere a una automatización total [6]. La clasificación de accidentes para nuestro caso de estudio es importante y analizar lo que debería ser crítico para mejorar los sistemas autónomos. Para el caso de estudio se enfoca en las cuatro causas principales que son choques, salidas del camino, volcadura y atropellamiento. En todas estas causas los choques comúnmente se deben al exceso de velocidad de los conductores afectando la estabilidad y posible salida del camino, por otra parte, las muertes de peatones han aumentado un 51% desde que alcanzaron su punto más bajo en 2009 y representan el 17% de las muertes por accidentes [7].

Ahora bien, dentro en los antecedentes para los algoritmos de detección los modelos de dos etapas tuvieron su arranque con la red RCNN. Esta red sustituyó los algoritmos de extracción de características tradicionales como HOG o SIFT por una red que utilizó para dicha extracción (backbone) y lo combinó con un algoritmo de proposición de regiones que las características se clasifican usando un algoritmo SVM. Por otro lado los modelos de una etapa comenzaron con el modelo OverFeat que integraban la clasificación y la localización del objeto en una única arquitectura de red. Este modelo era más rápido que su homólogo en los de dos etapas pero era menos preciso. Y así llegamos hasta el modelo YOLO que evolucionó del modelo YOLOv1 al modelo v5 a día de hoy. Este modelo entraría dentro de este grupo, donde también encontraríamos el modelo SSD (single shot multibox detector) [8]. En este caso de estudio se analizan las velocidades altas, peatones, vehículos y señales de alto las cuales serán detectadas con el algoritmo YOLOv5.

III. Metodología

Usualmente los VA comparten su funcionamiento esencial en tres elementos comunes que son: sensores como interpretes entre el mundo físico y el movimiento o dinámica del entorno, las máquinas que se encargan de los cálculos y procesamiento y finalmente los actuadores, todo esto cuando se encuentran en un ambiente urbano se rodean de información que a cada fracción de segundo se generan en cantidades masivas; por ejemplo, peatones, semáforos, calles, automóviles que rodean, señales de tránsito, etc. La importancia en la detección y clasificación correcta en tiempo real es indispensable para el adecuado funcionamiento y así evitar accidentes que perjudiquen a la salud de los pasajeros o personas del entorno.

A. Machine learning

Teniendo grandes cantidades de datos (Big data) su correcta clasificación e interpretación es de suma importancia y por ello existen algoritmos de inteligencia artificial específicamente el ”machine learning” o ”lenguaje automatizado” que ayudan a esta clase de procesamiento. Una ventaja al usar esta clase de algoritmos es el reconocimiento de la estructura de datos y poder entregar una predicción. Estos algoritmos de machine learning son diseñados para entrenarse con múltiples datos y así poder determinar las relaciones que existen entre estos, a esta etapa se le denomina ”entrenamiento” y finalmente entregan resultados que dependen de nuevos grupos de datos utilizados en el entrenamiento y así poder tener un porcentaje de aprendizaje adecuado [9].

B. Deep learning

Ahora bien el aprendizaje profundo o Deep Learning (DL) que es una rama del machine learning, permite que modelos computacionales compuestos por varias capas de procesamiento puedan aprender representaciones sobre datos con múltiples niveles de abstracción y mediante este concepto descubrir grandes volúmenes de datos [10].

C. CNN

Las redes neuronales convolucionales o ”Convolutional Neural Networks” (CNN) son producto del deep learning las cuales son útiles en el campo de investigación de visión artificial, dado su buen desempeño en problemas de reconocimiento e interpretación en imágenes y vídeo [11]. El propósito de las redes convolucionales es la extracción de características dentro de una imagen, posteriormente clasifica los objetos. Su aprendizaje lo logra a través de capas, estas optimizan y ajustan los factores a analizar para minimizar el porcentaje de error [12].

D. YOLOv5

YOLO es un algoritmo con licencia abierta utilizado en detección de objetos de última generación, en el campo de la visión por computadora previamente a este existían diferentes maneras de detectar objetos, por ejemplo; sliding windows donde eran un poco ineficientes, luego algoritmos basados en R CNN, Fast R CNN y Faster R CNN [13], pero en el 2015 se inventó YOLO donde su nivel de optimización y eficiencia en la detección fue superior. Con un rápido desarrollo en el campo de detección de objetos gracias a las CNN se tienen algoritmos avanzados y en constante actualización, en este caso de estudio se empleará el algoritmo You Only Look Once (YOLO) el motivo de usar este algoritmo es por su alto rendimiento y optimización de recursos, también es amigable para cualquier usuario con curiosidad en el campo de la visión artificial, con una gran eficacia en la detección de objetos [14]. En YOLO se toma la detección de objetos como un problema único de regresión, una sola red convolucional que predice simultáneamente múltiples cuadros delimitadores que enmarcan los objetos en la imagen y predice probabilidades condicionales por cada clase para cada uno de estos cuadros delimitadores. La red neuronal puede lograr una velocidad de ejecución de 45 a 155 fotogramas por segundo (fps) en computadoras de propósito general, aunque con coste computacional aún limitado para algunos usuarios. A diferencia de otros métodos de detección con inteligencia artificial, YOLO trabaja sobre la imagen globalmente, por

ello su codificación es implícita respecto a la información contextual, modela el tamaño de la imagen y forma de objetos como apariencia. Actualmente YOLO tiene cinco versiones y en su funcionamiento se utilizan 24 capas convolucionales seguidas por 2 capas conectadas y utiliza capas de reducción 1x1 seguidas por capas convolucionales 3x3 [14]. El diseño de YOLO permite el trabajo en tiempo real manteniendo una alta precisión media. El sistema divide la imagen de entrada en una cuadrícula S × S. Si el centro de un objeto cae en una celda de la cuadrícula, esa celda de la cuadrícula es responsable de detectar ese objeto. Cada celda de la cuadrícula predice cuadros delimitadores y coloca puntos de confianza para esas cajas. YOLO por default contiene una amplia detección de 80 clases utilizando el dataset COCO, pero en este caso se vuelven a utilizar los pesos con los nuevos enfoques de las clases que se definieron.

Fig. 1. Funcionamiento de YOLOv5

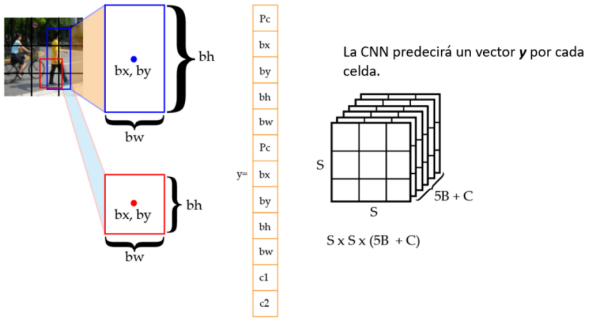

Fig. 2. Funcionamiento de YOLOv5 múltiples cajas delimitadoras

E. Dataset

Para entrenar el algoritmo previamente se debe tener un volumen de imágenes del objeto que se quiere detectar, el volumen depende de cada algoritmo. Las imágenes que tengan el objeto a detectar deben tener alguna ruta específica, usualmente se le denomina ”images” dentro de ella se separan en dos subconjuntos de imágenes. Es importante que se tenga una homogeneidad en cada una de las clases (objetos) a detectar, ya que ayuda a que el algoritmo tenga los mismos parámetros para su entrenamiento.

- El primer conjunto de imágenes se van a utilizar para el entrenamiento del algoritmo ”train”
- En este subconjunto usualmente llamado ”value” se van a comparar prediciendo los errores y los resultados de la predicción

Las imágenes fueron descargadas de internet de manera aleatoria y renombradas acorde a la clase y número de imagen. Finalmente se extrajeron 100 imágenes por clase en sus carpetas respectivas.

F. Etiquetado de imágenes

Para que el objeto sea detectado, el usuario debe indicar la posición del objeto en una imagen también conocido como ”Bounding box”, existen distintas alternativas para poder obtener estas coordenadas de puntos (x,y) en 4 vértices de la caja. En este caso se realizan pruebas de selección con labelImg (fig. 3) es muy intuitivo y lo único a realizar es hacer la selección del objeto y exportar el archivo en una carpeta denominada ”labels”. Aquí la paciencia es muy importante ya que todas las imágenes (como de entrenamiento y validación) deben ser etiquetadas, el formato que requiere YOLO es con extensión ”txt”. YOLO es estricto a cuanto límites de predicción dentro un bounding box, se debe a la celda de la cuadricula que predice otras dos, en alguna de estas existe la restricción de especificar únicamente a la clase seleccionada; por ejemplo, cuando hay demasiados peatones, al estar cercanos el sistema entrena con las delimitaciones que hemos escogido y esas fronteras de predicción es el error del modelo a detectar. También le ayuda a predecir con cuadros delimitadores de datos que ya previamente son entrenados para dar un mayor índice de efectividad.

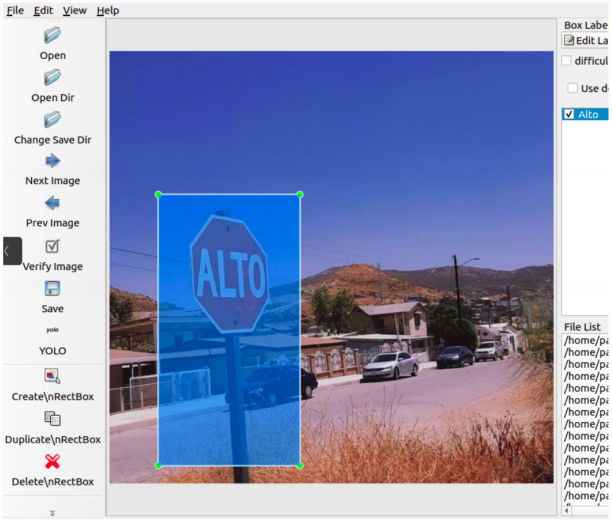

Fig. 3. Etiquetado de clase ”Alto”

G. Entrenamiento

Algunas veces los usuarios no tienen la capacidad de procesamiento para el entrenamiento del algoritmo, ante esa dificultad en este proyecto se utiliza Colaboratory, también

llamado ”Colab”, que es un entorno interactivo denominado ”cuaderno de Colab” donde te permite ejecutar y programar en Python desde tu navegador web con las siguientes ventajas:

- Funciona con una cuenta Google
- No requiere instalación o configuración de alto nivel
- Se puede compartir y colaborar de manera sencilla
- Dada su interacción y fácil GUI con el usuario, es muy intuitivo y poderoso en tiempo de ejecución

Colab puede facilitar tu trabajo, ya seas estudiante, científico de datos o investigador de IA, sus aplicaciones pueden ser muy amplias y cualquier persona conectada a internet puede crear grandes cosas. La ventaja de utilizar los pesos ya entrenados en primer plano afectan al tiempo y eficiencia de detección, la razón es que al tener una serie de parámetros como identificación de formas, texturas, y profundidad el algoritmo puede identificar de una manera más rápida solamente con los datos obtenidos de una imagen [15], de ahí el nombre que define al algoritmo ”Solamente ver una vez”. Al tener los archivos de programación, imágenes etiquetadas y el conjunto de imágenes ahora es momento de poder entrenar el modelo. Existen múltiples repositorios en Github para poder entrenar, en este caso nos basamos en el modelo YOLOv5 representando la investigación de código libre de Ultralytics sobre métodos de IA, incorporando a una investigación de más de 150 colaboradores con un conjunto de datos entrenados de COCO. Al ser de código abierto, los usuarios pueden realizar mejoras constantemente para diversificar su funcionalidad, en este caso se reprimieron las 80 clases ya entrenadas, al cual se le modificó el conjunto de clases y añadiendo las 4 clases de este problema, ahora bien como se menciona antes el algoritmo no hace todo desde un inicio, al descargar todos los pesos, solo configuramos las últimas dos capas de la red y esto ayuda a que el modelo realice una predicción porque ya tiene cargado ciertas características. En la programación el lenguaje principal es Python y al ejecutar YOLOv5 se tienen diferentes carpetas y la modificación es en la carpeta denominada ”data” en esta parte se modifica el archivo ”coco128.yaml” anexando las 4 clases de este proyecto ordenadas respecto a las etiquetas, también se modificaron las rutas para las carpetas de entrenamiento y validación, además que el entrenamiento se realizo con un batch de 9 y 200 épocas con los pesos de YOLOv5.

IV. Resultados

De acuerdo a la matriz de confusión y a las curvas de predicción con un total de 4 clases se obtuvo una efectividad de 89% es una buena predicción comparada el número de imágenes utilizadas para el entrenamiento y validación. Al utilizar 100 imágenes por clase y tomar el 10 para validación es un buen porcentaje de predicción a muy bajo coste. Las épocas entrenadas fueron 200 y terminó el proceso aproximadamente en 150 minutos, las cuales desde la época 20 tendieron a un comportamiento estable. Para la clase ”Limits” y ”Alto” llego a una detección del 100%, esto es importante ya que los VA podrán tener una ubicación del señalamiento de tránsito para igualar a la velocidad o detenerse en dado caso

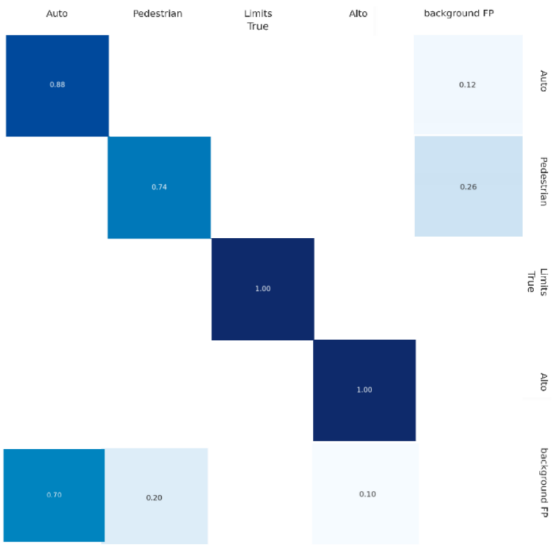

Fig. 4. Matriz de confusión

haya sido detectado, para la clase peatón hubo baja efectividad respecto a las otras tres clases con un 74% de efectividad. En los sistemas inteligentes se requiere de estrategias con alta precisión y exactitud que respondan a eventos en tiempo real. Sin embargo, en ocasiones el procesamiento y sincronización entre sensores y dispositivos hacen complicada y muy robusta esta tarea. En este trabajo se procesaron únicamente imágenes y vídeos, las cuales proporcionan una solución a los casos estudiados. Estos valores favorables son comparados con un estudio similar donde se realizaron pruebas con una base de 3000 datos con 256 para entrenamiento y 759 para validación programada con la librería Keras con clase de procesamiento tipo categorical donde se obtuvo una presición de 94.51% para la clase Auto y 49.70% para la clase Peatón [16]. Las pruebas generales en este proyecto fueron realizadas con una PC de 8gb RAM, NVIDIA GEFORCEGTX y un procesador i5 de 7th generación.

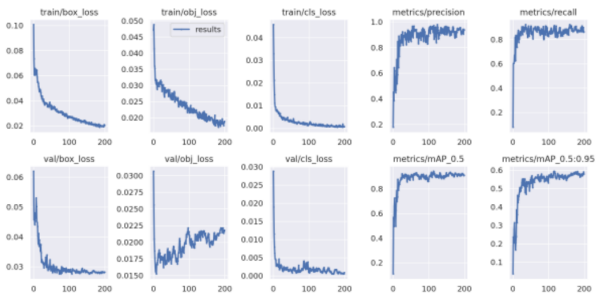

Fig. 5. Resultados generales

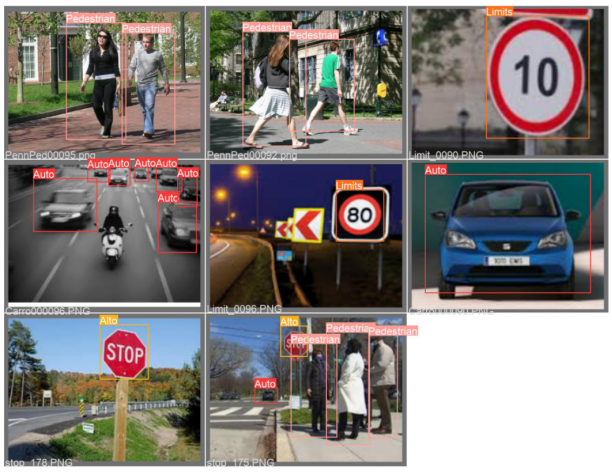

Fig. 6. Detección de objetos

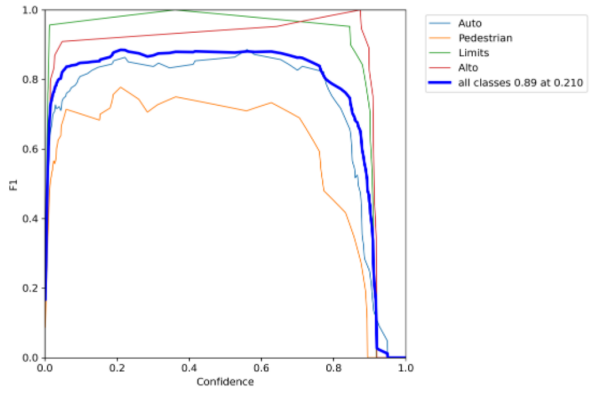

Fig. 7. Resultados de confianza

Conclusión

El algoritmo empleado en este proyecto brinda de múltiples herramientas para experimentar con problemas de detección artificial al igual que el entorno Colab para su procesamiento. Dados los resultados y comparando con el estudio "Caracterización de imágenes para aplicación en sistemas inteligentes" (que es un estudio similar) el sistema propuesto tiene una mejor efectividad en predicción, esto lo logro con menores épocas y con menor volumen de datos, sin embargo los resultados aún deben ser mejorados para beneficiar la seguridad de los pasajeros aumentando los datos, las clases y pruebas, a su vez, en estas principales detecciones se deben de realizar pruebas en tiempo real. Como parte de la discusión de trabajos futuros, con este proyecto se da inicio a un conjunto de análisis de detección de objetos en vehículos autónomos para beneficiar a personas con capacidad reducida.

References

[1] Y. Li, M. Díaz, S. Morantes, y Y. Dorati, Vehículos autónomos: Innovación en la logística urbana, Rev-RIC, vol. 4, n.º 1, pp. 34-39, oct. 2018.

[2] Traumatismos causados por el tránsito. (s. f.). WHO — World Health Organization. https://www.who.int/es/news-room/fact-sheets/detail/road-traffic-injuries

[3] Comisión Nacional de Seguridad. (2015, 10 de septiembre). ::: CNS :::. http://www.cns.gob.mx

[4] A. T. Akmatova, "ROAD ACCIDENTS.", Theoretical Applied Science, vol. 84, n.º 04, pp. 833–835, abril de 2020.

[5] A. Valero Matas, A. De la Barrera, "The Autonomous Car: A better future?," Universidad de Valladolid,Madrid, Palencia, Tech. Rep, 1989-8487, 2019

[6] SAE. Society of Automotive Engineers. On-Road Automated Vehicle Standards Committee, 2014. Tax-onomy and definitions for terms related to on-road motor vehicle automated driving systems.

[7] M. Favarò, N. Nader, S. O. Eurich, M. Tripp y N. Varadaraju, "Examining accident reports involving autonomous vehicles in California", PLOS ONE, vol. 12, n.º 9, septiembre de 2017, art. n.º e0184952. Accedido el 17 de octubre de 2021.

[8] A. Vidal, "visión artificial aplicada a los sistemas de transportes inteligentes: aplicaciones prácticas", tesis, Departamento de arquitectura y tecnología de computadores, Universidad del País Vasco, 2020.

[9] Benjelloun, F.-Z., Ait Lahcen, A. y Belfkih, S. (2018). Big Data technologies: A survey. Journal of King Saud University - Computer and Information Sciences, 30(4), 431–448. https://doi.org/10.1016/j.jksuci.2017.06.001

[10] Y. Bengio, A. Courville y P. Vincent, "Representation Learning: A Review and New Perspectives", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, n.º 8, pp. 1798–1828, agosto de 2013.

[11] M. V. Valueva, N. N. Nagornov, P. A. Lyakhov, G. V. Valuev y N. I. Chervyakov, "Application of the residue number system to reduce hardware costs of the convolutional neural network implementation", Mathematics and Computers in Simulation, vol. 177, pp. 232–243, noviembre de 2020.

[12] H. Habibi Aghdam y E. Jahani Heravi, "Traffic Sign Detection and Recognition", en Guide to Convolutional Neural Networks, Cham: Springer International Publishing, 2017, pp. 1–14.

[13] A. Bochkovskiy, C. Wang, H. Mark Liao, "Optimal Speed and Accuracy of Object Detection" 23 Apr 2020

[14] J. Redmon, S. Divvala, R. Girshick y A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection", en 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 de junio de 2016. IEEE, 2016.

[15] L. Wang, J. Shi, G. Song y I.-f. Shen, "Object Detection Combining Recognition and Segmentation", en Computer Vision – ACCV 2007, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 189–199.

[16] J. P. . González Mendoza, E. de J. . Gasca Laguna, F. J. . Serrano Martínez, J. . Duarte Jasso, y D. L. . Almanza Ojeda, Caracterización de imágenes para aplicaciones en sistemas inteligentes, JC, vol. 10, sep. 2021.