Toward a New Dataset: Mexican Traffic Signs "ReWaIn-MTS" for Detection

Daniela Bolaños-Flores, Tania Ramirez-delreal, Hamurabi Gamboa-Rosales, and Guadalupe Gutierrez-Esparza

***Abstract*—Various factors on the road can endanger the safety of drivers or pedestrians and cause high-impact accidents while driving. This is why traffic signs, as essential elements, provide crucial information on the condition of the road during the trip. We introduce the ReWaIn-MTS dataset, a practical tool that can support and advance research on traffic sign detection and classification in Mexico. Its applications are theoretical and have implications in the real world in autonomous conduction or assist driving. Convolutional neural networks (CNNs) have shown outstanding detection results compared to conventional methods. In this work, we used CNN-based machine learning techniques to categorize and detect Mexican traffic signs. The dataset, focused on traffic signs on the Mexican territory within the main urban roads in eight different cities, contains 2,283 road elements divided into 37 classes to train and validate algorithms. The mean Average Precision (mAP) metric compares the performance in state-of-the-art detection methods, particularly YOLOv5, YOLOv8, YOLOv11 and the Transformer RT-DETR.**

Link to graphical and video abstracts, and to code: https://latamt.ieeer9.org/index.php/transactions/article/view/9614

***Index Terms*—Traffic Sign Detection, Mexican Traffic Signs, YOLOv5, YOLOv8, YOLOv11, RT-DETR Trasformer.**

I. Introduction

THE contribution of traffic signs is to provide details on the route, mainly indications related to restrictions, limitations, warnings, and other valuable navigation announcements [1]. The signs show encoded information depending on the shape, color, and figures contained. Mexican traffic signs are classified as vertical signs, horizontal signs, and safety devices that complement each other [2].

The Mexican norm [2] has two types of signals: vertical and horizontal. The vertical signals are in a pole and the horizontal signals are on the road. The vertical ones have three categories: preventive, restrictive, and informative. This work is focused on vertical ones. In addition, a color code indicates different signals, such as prevention (yellow), services and tourist information (blue), restriction, general information and recommendation (white), construction zone (orange), stop and prohibition (red), destination information (green) and school crossing (fluorescent lemon green).

The signage allows the driver to have warnings for safer driving. However, on some occasions, their omission can cause accidents, for example, by ignoring the permitted speed limits [3].

Technological development has significantly advanced, so it has made significant contributions to society and various areas of engineering, such as interpreting visual information in machines, using computational techniques for image recognition, and obtaining promising results in different fields of application.

In the automotive domain, deep learning architectures have been implemented, where road element detection models are proposed and developed; later, they can be used in intelligent driving cars, allowing the driver to regard signs or details that are in the duration of routes or urban areas to detect signals that have not been observed. Driver Assistance Systems help reduce car accidents by automating tasks [4], for example, traffic sign recognition.

Intelligent cars include autopilot functions that allow users to identify stop signs and traffic lights from a certain distance as long as they are under active supervision [5]. The disadvantage is the high cost of acquiring one of these intelligent units; in addition, the signals that the vehicle detects and recognizes are merely focused on American and European standards, which limits the area where these functions are implemented.

Previously, research works focused on the recognition and detection of traffic signs using machine learning have been presented in the literature [6]–[9], obtaining satisfactory results when testing them; the disadvantage of these is that the region to which it is oriented is attached to foreign signals, so it is not possible to implement it in Mexican territory because there is variability between Mexican and foreign signs including design, color, and symbology.

Likewise, deep learning-based models have been carried out [10], where they carry out recognition and detection for road signs within the Mexican sector; in this work, we address the problem of having a large diversity of classes in Mexican signs traffic, so it is biased to specific signals and certain regions of the country, therefore, when testing the model, it cannot act on possible cases.

Our contributions are summarized as follows:

The associate editor coordinating the review of this manuscript and approving it for publication was Gladston Moreira (*Corresponding author: Tania Aglae Ramirez-delreal*).

D. Bolaños-Flores, and H. Gamboa-Rosales are with Universidad Autónoma de Zacatecas, Zacatecas, Mexico (e-mails: 42106547@uaz.edu.mx, and hamurabigr@uaz.edu.mx).

Tania Aglae Ramirez-delreal is with Secretaría de Ciencia, Humanidades, Tecnología e Innovación and with Centro de Investigación en Ciencias de Información Geoespacial, Aguascalientes, Mexico (e-mail: tramirez@centrogeo.edu.mx).

G. Gutierrez-Esparza is with Secretaría de Ciencia, Humanidades, Tecnología e Innovación and with Clinical Research, National Institute of Cardiology 'Ignacio Chávez', Mexico City, Mexico (e-mail: ggutierreze@secihti.mx).

- We introduce a novel dataset with restricted, warning, and informative Mexican traffic signs ("ReWaIn-MTS") of eight cities in the country. The set contains 1439 images with 2283 annotations for 37 classes.
- We compare the results on Mexican traffic sign detection performance among the YOLOv5, YOLOv8, YOLOv11, and Transformer RT-DETR architectures using the mean Average Precision (mAP) metric.

The paper is organized as follows. In Section II, the related work is presented. Section III describes the methods utilized and the details of the proposed dataset. The experiments and results are given in Section IV. Ultimately, the conclusions are presented in Section V.

II. Related Works

Diverse architectures have been proposed to detect objects, particularly one- or two-stage models and transformers with attention mechanisms; the applications are in various fields, including industrial and everyday activities. [11].

Convolution Neural Networks (CNN) are widely used to detect objects, particularly CNNs adding a region proposal [12]; these techniques are based on two-stage algorithms, and the main disadvantage is time processing.

Instead, the one-stage algorithms use fewer computational resources to generate the object's classification probability and coordinates. One of these structures is YOLO (You Only Look Once) [13], which has different versions.

Finally, the emerging Transformer [14] has allowed the use of this architecture in object detection, combining with CNN for the classification that appears on DETR [15] and other configurations using the attention mechanism, and enhancing RT-DETR to achieve inference speeds that allow its use in real-time applications [16].

A. Traffic Signal Detection and Datasets

In general, in the published papers related to Traffic Sign Recognition (TSR) and Traffic Sign Detection (TSD), a wide, well-known dataset has been used mainly the following can be noted:

- High-Resolution Remote Sensing Detection (HRRSD) [17]: This group contains 13 categories where only 11% belong to Asian traffic signs.
- Tsinghua-Tencent 100K (TT100K) [18]: This set contains 128 classes where only 30,000 focus on Asian traffic signs.
- German Traffic Sign Recognition Benchmark (GTSRB) [19]: This set has 43 categories and approximately 51,839 images with German road elements.

Some authors use the TT100K dataset to generate a baseline in the traffic sign recognition task. Liu et al. [20] developed a model that recognizes signs of limited proportion, blurry, and complex within a natural environment through a modified ResNeXt50 architecture. Li et al. [21] perform a model that recognizes and classifies signs containing numerical digits within a car route using a neural network based on deep learning, modified and proposed by themselves. Furthermore, Liang et al. [22] implemented a multiscale Sparse R (MSR) CNN.

Recent works used the GTSRB dataset to compare performance. Cao et al. [23] performed the recognition and detection of traffic signs using the LeNet-5 CNN architecture and another modified tool to improve the safety of the individual while driving, providing constant development of vehicle driving assistants. In contrast, Vennelakanti et al. [24] proposed a CNN Ensemble to recognize and detect circular and triangular signals from Belgium and Germany.

Other authors stand out for the realization of architectures with different CNN configurations ([25]–[27]); their main purposes are detection and recognition with better speed or in real-time. Some works use detectors such as YOLOv4, Faster R-CNN, and VGG with modifications [28]–[30].

Liu et al. developed [20] an intelligent traffic sign detection and recognition model based on YOLO. Within the system testing phase, the HRRSD dataset was used with 21,761 images with categories such as bridges, vehicles, and airplanes, among others; it is worth mentioning that this dataset is not focused on traffic signs if not used to evaluate the capacity of the proposed algorithm.

Another work with a specific Swedish dataset (STSD) is used, which contains the number of 19,236 images used to create the model, where the Mask R-CNN algorithm (Resnet-50) was used [31].

Among the related works, they use hybrid datasets; they carry out a combination of signals. Siniosoglou et al. [32] propose Auto-Encoder, which uses a modified dataset to strengthen the dataset.

In addition, Huang et al. [33] propose combined architectures; they use CCTSDB and TT100K with 4,000 images in total and the learning models to detect SSD300 + VGG16, Faster RCNN and YOLOv3.

In addition, the recent emergence of transformers has allowed the implementation of traffic signal detection [34], [34].

The datasets and works are focused on traffic signs concerning a specific area; therefore, if a model for the detection and recognition of Mexican traffic signs were developed, they would not obtain satisfactory results since the categories, shapes, and colors of these are different from those provided, except for using the MTSD (Mexican Traffic Sign Dataset) [10] since it could recognize some Mexican signals, but is left with the limitation that it would not detect or identify most of them, because the data are reduced to three cities localized in the north of the country, and they only have 11 classes.

Compared to previous datasets, ReWaIn-MTS offers broader coverage and better contextual relevance for Mexican environments. GTSRB [19] includes 43 classes from Germany and TT100K [18] offers 128 Chinese sign categories, though with class imbalance. MTSD [10], focused on Mexico, is limited to three northern cities and 11 classes. In contrast, ReWaIn-MTS includes 37 classes in eight cities in central and southern Mexico. Its combined use of physical and virtual data sources in real-world conditions improves its applicability to Latin American contexts.

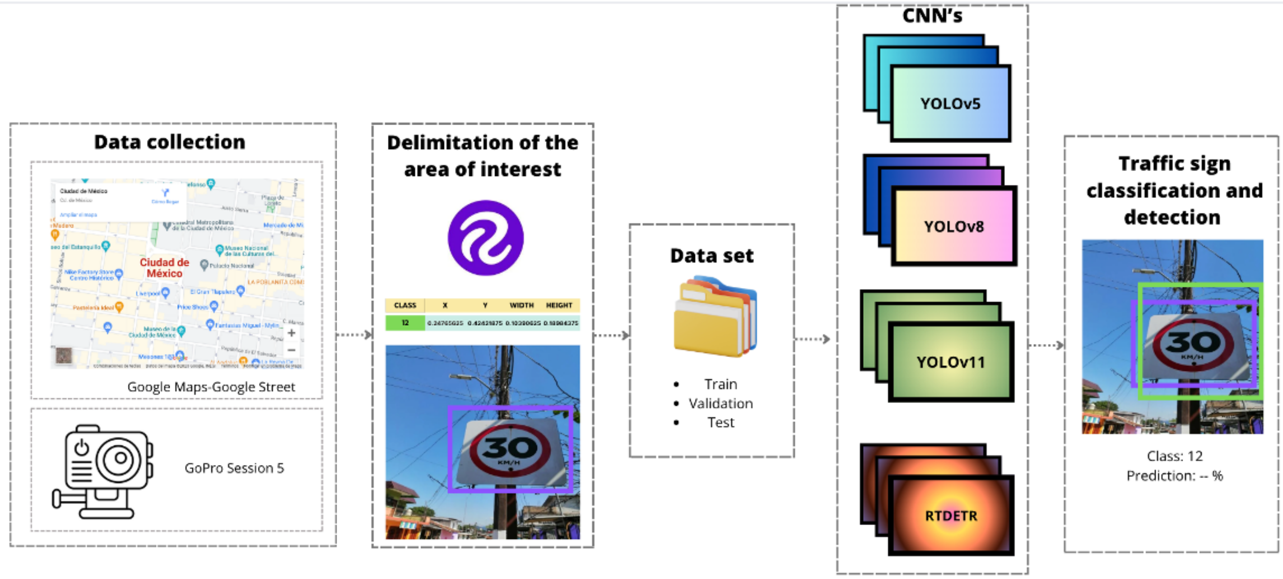

Fig. 1. Methodology carried out to carry out the creation of the model, the ground truth is shown in the purple box, and the detection for bounding box is shown in green.

III. Methodology

Fig. 1 illustrates the methodology used in creating the traffic sign detection and recognition model, along with the data collection to deliver a dataset with Mexican traffic signals, employing three convolutional neural network architectures: YOLOv5, YOLOv8, YOLOv11 and RT-DETR.

Initially, data acquisition was conducted in two ways: the first involved using a video camera mounted on a vehicle that traveled the urban roads of the city. For the second method, Google's tools were utilized, enabling the visualization of various routes in several cities throughout the Mexican region.

After completion of the aforementioned task, the next step involves defining the region of interest. The signs locations in the proposed images are annotated using the labeling tools provided by Robotflow software [35]. This annotation aims to export the data into the appropriate formats for the annotations of the bounding boxes, generating separate sets for training, validation, and testing.

The subsequent step involves training the computational model. In this stage, the previously mentioned networks, specialized in object recognition and detection, are utilized, significantly contributing to achieving the primary objective. This process occurs in various execution environments and is divided into individual segments. Ultimately, the models created for each architecture are obtained, and an assessment and comparison of their performance is performed with other related works.

In our study, YOLOv5, YOLOv8, YOLOv11 and RT-DETR are chosen based on their distinctive characteristics and proven effectiveness in object detection tasks.

YOLOv5 was selected because of its efficiency and performance in real-time object detection; this model is used for applications with fast and accurate predictions.

YOLOv8 was selected for its architectural improvements over its predecessors, particularly for its accuracy of feature extraction and detection. At the time of our study, YOLOv8 offered performance in detecting objects in various environments, in addition to high speed.

YOLOv11 was selected due to the new incorporations that improves and optimize detection accuracy without sacrificing speed. This allows for more accurate object identification, especially in complex scenes or with small, overlapping objects.

RT-DETR (Real-Time Detection Transformer) represents a significant evolution in object detection by combining the accuracy of transformers with real-time efficiency, overcoming the limitations of both traditional CNN-based approaches and slower transformers such as DETR.

A. *YOLOv5*

The YOLOv5 architecture (You Only Look Once version 5) represents a significant iteration within the series of YOLO algorithms [36], designed to address object detection in images using a convolutional neural network (CNN). This specific version, known as YOLOv5, has introduced substantial improvements in accuracy and efficiency compared to its predecessors.

In contrast to traditional approaches that divide the detection process into multiple stages, YOLOv5 performs object detection and classification in a single pass. This is achieved through a convolutional neural network that takes an input image and directly generates predictions of the objects present and their locations.

As described by Terven J. and Cordova-Esparza D. [37], this architecture comprises three main aspects. Firstly, the Backbone Network utilizes a Convolutional Neural Network architecture to process the most relevant features of the input image to reduce computational load and memory usage. The second part focuses on the Spatial Pyramid Pooling Fast Aggregation, also known as (SPPF), which accelerates network calculations by pooling features from different scales into a fixed-size feature map. Finally, there are the Predictor Heads, which resemble the third version of the You Only Look Once line.

B. YOLOv8

The YOLOv8 architecture (You Only Look Once version 8) [38], focuses on improving accuracy, efficiency, object detection, and image classification time through a convolutional neural network (CNN) and computer vision techniques.

This architecture involves modifying the network structure with changes in its feature-fusion modules, enabling improved precision in the detection task. Furthermore, the structure of the neural network is characterized by its lightweight design, which contributes to a reduced processing time [37].

The main elements of the system include three key components. Firstly, similar to the previous one, we have the Backbone Network, which plays a fundamental role in YOLOv8. It extracts features from input data and recognizes high-level image attributes.

The second phase, the Neck Network, aims to connect the previous phase to the leading network. Its primary function is to reduce the size of feature maps and enhance their quality, striking a balance between accuracy and speed.

Finally, the last stage is called the Head Network, consisting of five detection modules and a prediction layer, allowing the prediction of the class of objects and location in the image [37].

C. YOLOv11

The YOLOv11 backbone network employs initial convolutional layers for downsampling, combined with newly introduced C3k2 blocks, which significantly accelerate the feature extraction stage. The structural arrangement allows for a gradual reduction in spatial dimensions, while the increase in channel depth captures more complex patterns [39].

YOLOv11 integrates multiple C3k2 blocks to process features, employing a flexible architecture that can be adapted according to the c3k parameter settings. Later stages use CBS (Convolution-BatchNorm-SiLU) layers to further stabilize and normalize the feature maps, ensuring robust predictions through effective non-linear activation functions.

D. RT-DETR

The general architecture of DETR (End-to-End Object Detection with Transformers) is characterized by its remarkable simplicity, making it highly adaptable to various deep learning frameworks. To require fewer lines of code for implementation, its contribution significantly impacts fields dedicated to object detection and classification. This architecture consists of the following components: a convolutional neural network (CNN) backbone for feature extraction, an encoder-decoder transformer, and a feed-forward network, also known as (FFN), used for the final detection prediction [15].

The DETR is the antecedent of RT-DETR, the last one considers the classification and position of the detection, this architecture employs an encoder that efficiently processes multi-scale features by decoupling intra-scale interaction and inter-scale fusion. This architecture reduces computational costs and enables real-time detection, it also uses query selection that considers Intersection over Union (IoU), improving detection accuracy [16].

E. Mean Average Precision (mAP)

The mAP is the metric commonly used to measure the performance of object detection models; which for a set of detections is the mean over classes, the AP is given by the area under the precision/recall curve [40].

IV. Experiments and Results

This section explains the experimentation related to the collected dataset and tests the architectures for traffic sign detection.

A. Dataset

In this work, we present a new dataset for Mexican sign detection. The following paragraphs explain the details of the dataset.

1) Scope: The dataset consists of vertical restrictive, warning and informative traffic signs, following the guidelines of the latest version of the Mexican Road Signaling Manual [2].

2) Geographical Coverage: For the selection of the location of destination information signs, they are located in a successive location from the highest to the lowest hierarchy, first, the state capitals, second, the cities of commercial or tourist importance, then the border ports, fourth, the sites of tourist interest, and finally the municipal capitals and towns.

The selected cities were Mexico City, Monterrey, Puebla, San Pedro Cholula, Zacatecas, Veracruz, Córdoba and Misantla. Due to representative of different locations through north and south.

3) Image Specifications: The images have an average dimension of 965x591 pixels per image. The images contain at least one traffic sign and up to nine signs per image. It is important to note that traffic signs are located in uncontrolled environments in hours with light. Our dataset comprises 2,283 annotations that cover 37 different labels of traffic signs.

4) Annotations: Annotation is carried out through the Robotflow interface in an image from the dataset. Once the labeling phase is completed, the dataset is exported to the required format with bounded boxes. In this case, the bounded boxes are used by YOLO or RT-DETR; this enables the training model to acquire this information effectively for learning purposes.

5) Data Collection Methodology: For the first image acquisition stage, we used a Xiaomi POCO X3 smartphone equipped with a camera with dimensions of 3472x4624 pixels and a focal length of 25 mm. Additionally, we used a GoPro Session 5 video camera, which captured photos with specifications of 3648x2736 pixels and a focal length of 3 mm. This

TABLE I
CONTENTS OF THE REWAIN-MTS

	Signal type	Images ReWaIn-MTS Total, Train-Valid-Test
	Restrictive signs	
1	No parking	563, 399-99-65
2	Speed limit	265, 183-53-29
3	Bus Lane	112, 86-19-7
4	Yield	95, 56-25-14
5	Stop	82, 54-20-8
6	Parking	74, 53-13-8
7	No motorcycle	53, 31-18-4
8	Keep Left Or Keep Right	39, 23-10-6
9	Turning left prohibited	37, 23-10-6
10	Height Limit	26, 20-5-1
11	Driving straight ahead prohibited	26, 17-6-3
12	Sealt belt	24, 14-7-3
13	Turning around prohibited	18, 15-3-0
14	Trucks Prohibited	13, 9-3-1
15	Turn right	13, 7-4-2
16	Do not stop	12, 8-3-1
17	Turning right prohibited	11, 8-2-1
18	Cell phone use prohibited	10, 5-4-1
19	Mandatory right	7, 3-4-0
20	Passing right mandatory	6, 5-0-1
21	Return departure to unlevel	6, 4-1-1
22	Pedestrians prohibited	4, 3-1-0
23	Return	4, 3-1-0
24	Divided highway circulation	4, 3-1-0
	Warning Signs	
25	Curve indicator	184, 105-75-4
26	Crosswalk	228, 164-42-22
27	Transit incorporation	56, 40-10-6
28	Exit	51, 39-8-4
29	Curve	44, 31-7-6
30	Side road intersection	31, 18-10-3
31	Speed reducer	24, 16-7-1
32	Asymmetric narrowing	6, 4-1-1
33	Roundabout	5, 4-1-0
	Informative signs	
34	Bike zone	63, 50-10-3
35	Bus stop	52, 38-11-3
36	Doctor	15, 9-5-1
37	Disable person	20, 17-3-0
Total		2,283

camera was placed in the center of the dashboard of a vehicle to provide a better view of the surroundings while driving through various streets, avenues, parks, and urban areas in Zacatecas.

Due to the diversity of traffic signs in different cities, we proceeded to a second data collection phase using the Google Maps application and its Google Street View tool. This valuable tool allowed us to virtually visit the main avenues and streets in various cities.

In addition, a delineation of the traffic sign area was carried out within the image. Robotflow software was used to perform this task and labeling as mentioned in the previous text.

After the collection phase, 1439 images were obtained. Table I displays the classes of traffic signs in the dataset, their respective categories, and the total annotations per section.

6) Application Impact: The use of this new dataset allows us to train new strategies for driving assistance, and it is possible to carry out the same methodology for other countries with similar road conditions. In addition, trained models can be implemented in real-world applications in Mexico.

B. Experimental Details

For this study, multiple comparative tests were conducted with three novel convolutional neural network architectures designed for object detection and recognition to identify and perceive traffic signals within the Mexican national territory.

The selected architectures for these evaluations are as follows: You Only Look Once version 5 (YOLOv5) [36], [41], You Only Look Once version 8 (YOLOv8) [38], [42], You

Only Look Once version 11 (YOLOv11) [39], and Real Time Detection Transformer (RT-DETR) [16].

Each underwent training in different execution scenarios. For the YOLOv5 and YOLOv8 architectures, the Google tool called Google Colab Pro was used, which enables the execution of complex programming tasks by providing access to Graphics Processing Units (GPUs) and Tensor Processing Units (TPUs).

As for the requirements used for these two initial architectures, utilizing a Tesla T4 GPU with 25 GB of dedicated RAM, 4 CPUs, and the code implementation is based on Python version 3.10.6 with CUDA version 11.6, and we used the PyTorch library 2.0.1.

For the RT-DETR architecture, a different configuration was used due to its longer execution time. The computational system used is as follows: we employed the Conda environment 23.1.0 on a Windows 10 OS, powered by an AMD Ryzen 7 3700X 8-Core 3.60 GHz processor with 16 GB of RAM and an NVIDIA GeForce RTX 2060 GPU with 6 GB of dedicated RAM. Similarly, the code implementation is based on Python version 3.7.13, with CUDA version 11.3.

Within the training process of the three different architectures. The dataset, “ReWaIn-MTS,” contains at least 1,439 images, including photographs collected personally and through virtual navigation tools, observed in their original form without any modifications. The input images have an average size of 964x591 pixels.

It is important to note that to perform model training using the YOLOv5 [36], YOLOv8 [38], YOLOv11 [39] and RT-DETR [16], the following parameters are followed: the default input image size is 640x640 pixels, and it utilizes the default parameters set on ultralytics to obtain results for the reference and baseline in the proposal of the ReWaIn-MTS dataset. All architectures used 100 epochs for training.

C. Experimental Results

Likewise, the “ReWaIn-MTS” data was utilized. The experiment was carried out in 100 epochs, demonstrating an overall performance of 84.82% in mAP50 and 75.54% in mAP50-95 for YOLOv5.

In the second execution, we utilized the YOLOv8 architecture. Similarly, we subjected this architecture to tests using the previously accessible dataset.

Furthermore, the dataset “ReWaIn-MTS,” run was completed in 100 epochs, as it reached its peak performance, displaying 96.21% in mAP50 and 67.66% in mAP50-95.

The YOLOv11 architecture after 100 epochs we obtained 92.64% in mAP50 and 65.30% in mAP50-95.

YOLOv8’s improved performance may be due to the fact that it has been extensively tested, tuned, and optimized by the community and its developers (Ultralytics), while YOLOv11 is the most recent and may still be in its experimental stages, without as much refinement.

The final architecture, RT-DETR, was chosen due to its widespread development in 2024, as Zhao et al. [16]. This architecture stands out for its distinctive characteristics, as it does not belong to any previous YOLO architectures. This makes it particularly interesting to detect and recognize Mexican traffic signs.

Similarly, to carry out the model training, we configured the execution environment following the specifications, as it requires more processing time.

With the dataset, “ReWaIn-MTS,” a performance of 93.12% in mAP50 and 64.12% in mAP50-95 was achieved in 100 epochs. Furthermore, detailed results are presented in Table II.

D. Visualization of Detection and Localization

Table II highlights the best performance achieved by the model utilizing the YOLOv8 architecture. Fig. 2 presents the predictions made by this traffic sign recognition and detection model, detailing the precise location of objects through bounding boxes generated with a confidence interval greater than 0.9.

Sixteen images were randomly selected from the test section. The model also demonstrates its ability to distinguish between unique and repeated bounding boxes, thus highlighting its effectiveness in identifying bounding boxes for various traffic signs, which contributes significantly to achieving the study’s general and specific objectives.

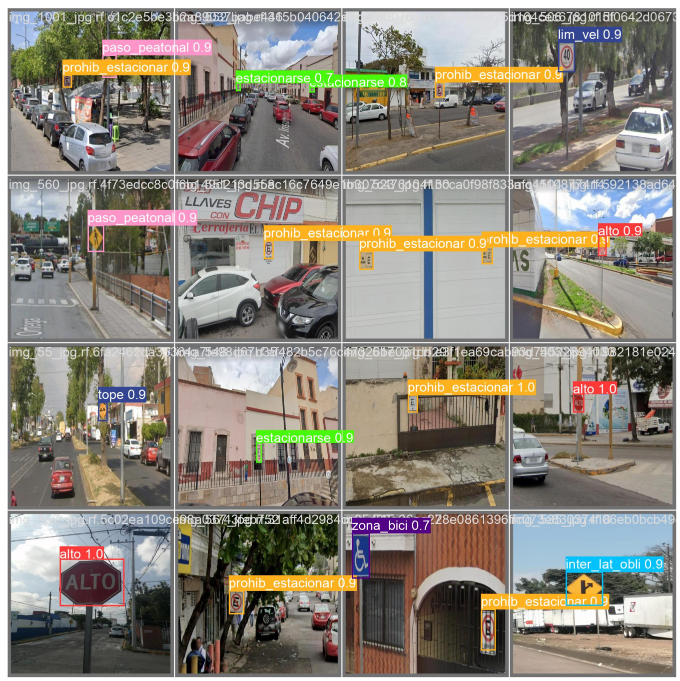

Fig. 2. Example of detection and classification using the model developed with YOLOv8 and applied to the ReWaIn-MTS dataset.

V. Conclusions and Future Works

This work proposes a dataset of Mexican traffic signs, complete with bounding-box annotations. The dataset includes images featuring multiple signs per picture and includes 2,283 annotations across 37 distinct classes of traffic signs, all collected from the urban roads of eight Mexican cities.

Future directions can reach OCR recognition in destination information and specific speed velocity. Providing traffic sign

TABLE II
RESULTS OBTAINED BY MODELS TRAINED WITH THE REWAIN-MTS DATASET USING THE YOLOV5, YOLOV8, YOLOV11 AND RT-DETR ARCHITECTURES

Dataset	Images	Architecture	mAP%50	mAP%50-95
ReWaIn MTS	1,439	YOLOv5 [36]	0.848217	0.755429
ReWaIn MTS	1,439	YOLOv8 [38]	0.962157	0.676624
ReWaIn MTS	1,439	YOLOv11 [39]	0.926451	0.653095
ReWaIn MTS	1,439	RT-DETR [15]	0.931235	0.641282

detections in Mexico to monitor, supervise and recommend locations for new traffic signs.

In future work, a plan is to balance the database to guarantee a fair representation of classes, seeking to achieve a more profitable model during training. In addition, approaches of modification in architectures specialized in object recognition and detection are being explored to improve model performance.

Ultimately, there is concern about using the proposed database to train an intelligent model capable of recognition and detection, not only in images but also in real time, implemented within a complete system inside a car.

The dataset used in this work is publicly available and properly referenced in the text, and the proposed is in https://github.com/taniaglae/ReWaIn-MTS.

REFERENCES

[1] A. S. Alturki, "Traffic sign detection and recognition using adaptive threshold segmentation with fuzzy neural network classification," in *2018 International Symposium on Networks, Computers and Communications (ISNCC)*. IEEE, 2018, doi=10.1109/ISNCC.2018.8531070., pp. 1–7.

[2] Secretaría de Infraestructura, Comunicaciones y Transportes, "Manual de señalización vial y dispositivos de seguridad," p. 770, 2014. [Online]. Available: https://lc.cx/sLiS3Z

[3] H. Wells, "The techno-fix versus the fair cop: Procedural (in) justice and automated speed limit enforcement," *The British Journal of Criminology*, vol. 48, no. 6, pp. 798–817, 2008, doi:10.1093/bjc/azn058.

[4] W. Zhao, S. Gong, D. Zhao, F. Liu, N. Sze, M. Quddus, and H. Huang, "Developing a new integrated advanced driver assistance system in a connected vehicle environment," *Expert Systems with Applications*, p. 121733, 2023, doi:10.1016/j.eswa.2023.121733.

[5] S. Nordhoff, J. D. Lee, S. C. Calvert, S. Berge, M. Hagenzieker, and R. Happee, "(mis-) use of standard autopilot and full self-driving (fsd) beta: Results from interviews with users of tesla's fsd beta," *Frontiers in psychology*, vol. 14, p. 1101520, 2023, doi:10.3389/fpsyg.2023.1101520.

[6] F. Almutairy, T. Alshaabi, J. Nelson, and S. Wshah, "Arts: Automotive repository of traffic signs for the united states," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 1, pp. 457–465, 2019, doi:10.1109/TITS.2019.2958486.

[7] U. Dubey and R. K. Chaurasiya, "Efficient traffic sign recognition using clahe-based image enhancement and resnet cnn architectures," *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, vol. 15, no. 4, pp. 1–19, 2021, doi:10.4018/IJCINI.295811.

[8] X. Gao, L. Chen, K. Wang, X. Xiong, H. Wang, and Y. Li, "Improved traffic sign detection algorithm based on faster r-cnn," *Applied Sciences*, vol. 12, no. 18, p. 8948, 2022, doi:10.3390/app12188948.

[9] R. Lahmyed, M. E. Ansari, and Z. Kerkaou, "Automatic road sign detection and recognition based on neural network," *Soft Computing*, pp. 1–22, 2022, doi:10.1007/s00500-021-06726-w.

[10] R. Castruita, M. Carlos, O. O. Vergara, V. Cruz, and H. Ochoa, "Mexican traffic sign detection and classification using deep learning," *Expert Systems with Applications*, vol. 202, p. 117247, 2022, doi:10.1016/j.eswa.2022.117247.

[11] E. Arkin, N. Yadikar, X. Xu, A. Aysa, and K. Ubul, "A survey: Object detection methods from cnn to transformer," *Multimedia Tools and Applications*, vol. 82, no. 14, pp. 21 353–21 383, 2023, doi:10.1007/s11042-022-13801-3.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, doi:10.1109/CVPR.2014.81, pp. 580–587.

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, doi:10.1109/CVPR.2016.91, pp. 779–788.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017, doi:10.5555/3295222.3295349.

[15] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*. Springer, 2020, doi:10.1007/978-3-030-58452-8_13, pp. 213–229.

[16] Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, Y. Liu, and J. Chen, "Detrs beat yolos on real-time object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 16 965–16 974.

[17] Y. Zhang, Y. Yuan, Y. Feng, and X. Lu, "Hierarchical and robust convolutional neural network for very high-resolution remote sensing object detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 8, pp. 5535–5548, 2019, doi:10.1109/TGRS.2019.2900302.

[18] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, "Traffic-sign detection and classification in the wild," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, doi:10.1109/CVPR.2016.232, pp. 2110–2118.

[19] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel, "Detection of traffic signs in real-world images: The german traffic sign detection benchmark," in *The 2013 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2013, doi:10.1109/IJCNN.2013.6706807., pp. 1–8.

[20] Y. Liu, G. Shi, Y. Li, and Z. Zhao, "M-yolo: Traffic sign detection algorithm applicable to complex scenarios," *Symmetry*, vol. 14, no. 5, pp. 1–21, 2022, doi:10.3390/sym14050952.

[21] Z. Li, M. Chen, Y. He, L. Xie, and H. Su, "An efficient framework for detection and recognition of numerical traffic signs," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, doi:10.1109/ICASSP43922.2022.9747406, pp. 2235–2239.

[22] T. Liang, H. Bao, W. Pan, and F. Pan, "Traffic sign detection via improved sparse R-CNN for autonomous vehicles," *Journal of Advanced Transportation*, vol. 2022, pp. 1–16, 2022, doi:10.1155/2022/3825532.

[23] J. Cao, C. Song, S. Peng, F. Xiao, and S. Song, "Improved traffic sign detection and recognition algorithm for intelligent vehicles," *Sensors*, vol. 19, no. 18, p. 4021, 2019, doi:10.3390/s19184021.

[24] A. Vennelakanti, S. Shreya, R. Rajendran, D. Sarkar, D. Muddegowda, and P. Hanagal, "Traffic sign detection and recognition using a cnn ensemble," in *2019 IEEE international conference on consumer electronics (ICCE)*. IEEE, 2019, doi:10.1109/ICCE.2019.8662019., pp. 1–4.

[25] A. Barodi, A. Bajit, A. Zemmouri, M. Benbrahim, and A. Tamtaoui, "Improved deep learning performance for real-time traffic sign detection and recognition applicable to intelligent transportation systems," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 5, 2022, doi:10.14569/ijacsa.2022.0130582.

[26] N. Youssouf, "Traffic sign detection and recognition with faster-

RCNN and YOLOV4," *Available at SSRN 4012734*, 2022, doi:10.1016/j.heliyon.2022.e11792.

[27] S. Ramkumar, G. Ramanathan, and K. Sridhar, "Traffic sign detection and recognition using cnn," *ECS Transactions*, vol. 107, no. 1, p. 17447, 2022, doi:10.1149/10701.17447ecst.

[28] Y. Yao, L. Han, C. Du, X. Xu, and X. Jiang, "Traffic sign detection algorithm based on improved yolov4-tiny," *Signal Processing: Image Communication*, vol. 107, p. 116783, 2022, doi:10.1016/j.image.2022.116783.

[29] U. Kamal, T. I. Tonmoy, S. Das, and M. K. Hasan, "Automatic traffic sign detection and recognition using segu-net and a modified tversky loss function with l1-constraint," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1467–1479, 2019, doi:10.1109/TITS.2019.2911727.

[30] E. Güney and C. BayiLlmiş, "An implementation of traffic signs and road objects detection using faster r-cnn," *Sakarya University Journal of Computer and Information Sciences*, vol. 5, no. 2, pp. 216–224, 2022, doi:10.35377/saucis.05.02.1073355.

[31] D. Tabernik and D. Skočaj, "Deep learning for large-scale traffic-sign detection and recognition," *IEEE transactions on intelligent transportation systems*, vol. 21, no. 4, pp. 1427–1440, 2019, doi:10.1109/TITS.2019.2913588.

[32] I. Siniosoglou, P. Sarigiannidis, Y. Spyridis, A. Khadka, G. Efstathopoulos, and T. Lagkas, "Synthetic traffic signs dataset for traffic sign detection & recognition in distributed smart systems," in *2021 17th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE, 2021, doi:10.1109/DCOSS52077.2021.00056, pp. 302–308.

[33] H. Huang, Q. Liang, D. Luo, D. H. Lee *et al.*, "Attention-enhanced one-stage algorithm for traffic sign detection and recognition," *Journal of Sensors*, vol. 2022, 2022, doi:10.1155/2022/3705256.

[34] H. Wang, "Traffic sign recognition with vision transformers," in *Proceedings of the 6th International Conference on Information System and Data Mining*, 2022, doi:10.1145/3546157.3546166, pp. 55–61.

[35] B. Dwyer, J. Nelson, J. Solawetz, and et. al., "Roboflow (Version 1.0)," 2022, accessed: 2023-10-13. [Online]. Available: https://roboflow.com/annotate

[36] G. Jocher, A. Stoken, J. Borovec, NanoCode012, ChristopherSTAN, L. Changyu, Laughing, tkianai, A. Hogan, lorenzomammana, yxNONG, AlexWang1900, L. Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, F. Ingham, Frederik, Guilhen, Hatovix, J. Poznanski, J. Fang, L. Yu, changyu98, M. Wang, N. Gupta, O. Akhtar, PetrDvoracek, and P. Rai, "ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements," *software*, Oct. 2020,doi:10.5281/zenodo.4154370.

[37] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, "A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas," *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680–1716, 2023, doi:10.3390/make5040083.

[38] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-time flying object detection with yolov8," *arXiv preprint arXiv:2305.09972*, 2023, doi:10.48550/arXiv.2305.09972.

[39] R. Khanam and M. Hussain, "Yolov11: An overview of the key architectural enhancements," *arXiv preprint arXiv:2410.17725*, 2024.

[40] P. Henderson and V. Ferrari, "End-to-end training of object class detectors for mean average precision," in *Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part V 13*. Springer, 2017, doi: 10.1007/978-3-319-54193-8_13, pp. 198–213.

[41] G. Jocher, "Ultralytics yolov5," 2020. [Online]. Available: https://github.com/ultralytics/yolov5

[42] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics yolov8," 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

Daniela Bolaños-Flores holds a Master's in Information Processing Sciences from the Autonomous University of Zacatecas. Her main areas of interest include deep learning, computer vision, image processing, automation, and electricity. Her academic and professional focus is on applying advanced technologies to improve intelligent and automated systems.

Tania A. Ramirez-delReal received her Ph.D. in science and technology from the Universidad de Guadalajara in 2017. She is currently a researcher at the Centro de Investigación en Ciencias de Información Geoespacial (CentroGeo) as a Researcher for Mexico in SECIHTI (Secretaria de Ciencia, Humanidades, Tecnologia e Innovacion). Her research interests are vision and language, digital image processing, pattern recognition, machine learning, optical interferometry, and intelligent control.

Hamurabi Gamboa-Rosales received a bachelor's degree in electronics and communications engineering from the Faculty of Engineering, University of Guadalajara, in 2000, a master's degree in electrical engineering from the University of Guanajuato, in 2003, with a focus on digital signal processing, and a Ph.D. degree in voice processing from the Technical University of Dresden, Germany, in 2010. He is currently working as a Professor and Researcher with the Academic Unit of Electrical Engineering, Autonomous University of Zacatecas, Mexico, in the area of research digital signal processing.

Guadalupe O. Gutiérrez-Esparza holds a Ph.D. in Computer Science and serves as a Researcher for Mexico in SECIHTI (Secretaría de Ciencia, Humanidades, Tecnología e Innovación). She is currently commissioned by the Instituto Nacional de Cardiología Ignacio Chávez, where she manages the data system and implements machine learning models for the Tlalpan 2020 project, focused on analyzing key factors associated with hypertension in the Mexican population. Her work focuses on artificial intelligence, predictive modeling, and data science applications in healthcare. She has contributed to the advancement of knowledge on cardiovascular risk, metabolic syndrome, and AI-based emotion detection.