A New Hybrid Approach Based on Discrete Wavelet Transform and Deep Learning for Traffic Sign Recognition in Autonomous Vehicles

Rim Trabelsi[1], Khaled Nouri[2]
Advanced Systems Laboratory-Tunisia Polytechnic School,
University of Carthage, 2078 La Marsa, Tunis, Tunisia[1]
Tunisia Polytechnic School, University of Carthage, 2078 La Marsa, Tunis, Tunisia[2]

***Abstract*—The rapid advancement of autonomous vehicles has led to the widespread integration of advanced driver assistance systems, significantly improving vehicle control, safety, and compliance with traffic regulations. A crucial aspect of these systems is the reliable detection and recognition of traffic signs, which play a key role in managing urban traffic flow and ensuring road safety. However, traffic sign recognition remains a challenging task due to varying lighting conditions, occlusions, and diverse sign appearances. This paper presents a novel hybrid approach for efficient traffic sign recognition tailored to the needs of autonomous driving. The proposed method combines the Discrete Wavelet Transform for robust feature extraction with the powerful classification capabilities of Convolutional Neural Networks within a Deep Learning framework. The DWT effectively captures essential image characteristics while reducing noise and irrelevant details, providing a compact yet informative feature set for the CNN classifier. Extensive experiments were conducted to evaluate the performance of the system in real-world conditions. The proposed approach achieved an impressive recognition precision of 98%, demonstrating its ability to interpret and respond to traffic signs with high reliability. The results confirm the method's robustness, real-time efficiency, and suitability for deployment in intelligent transportation systems and autonomous vehicles. Overall, this study highlights the complementary strengths of DWT and CNN within the broader context of Deep Learning, offering a significant improvement over conventional traffic sign recognition techniques. The proposed system represents a promising step toward enhancing the perception capabilities of autonomous vehicles, contributing to safer and more reliable navigation in complex traffic environments.**

***Keywords*—Safety; discrete wavelet transform; traffic sign recognition; autonomous vehicles; deep learning**

I. Introduction

Scene understanding is a fundamental challenge in computer vision, encompassing critical tasks such as object detection, classification, and semantic segmentation across diverse environments [1]. Recent advances in deep learning have significantly improved performance in these areas, enabling breakthroughs in autonomous driving and robotic vision [2]. Among these tasks, Traffic Sign Recognition (TSR) has attracted particular attention because of its direct impact on road safety and navigation efficiency. Accurate and real-time TSR is essential for both human drivers and autonomous systems, especially in unfamiliar or complex driving scenarios [3]. Reliable TSR ensures that vehicles can interpret and respond appropriately to traffic regulations, which is a cornerstone for the safe deployment of intelligent transportation systems.

Despite considerable progress, TSR remains a challenging problem due to both environmental and technical factors. Environmental conditions such as varying illumination, adverse weather (e.g. rain, fog, or low-light scenarios), and occlusions often reduce detection reliability. Furthermore, traffic signs exhibit significant variability in appearance, including differences in scale, shape, and color across countries, which complicates the recognition task. On the technical side, many existing approaches suffer from limitations in computational efficiency or fail to generalize well to real-world conditions. Although intelligent vision-based systems have attempted to address these issues using advanced image processing techniques and multi-sensor fusion [4], scalability and real-time performance continue to be open challenges. These limitations highlight the need for hybrid methods that combine robustness, efficiency, and adaptability.

To address these challenges, this study proposes a novel hybrid model that integrates the Discrete Wavelet Transform (DWT) with a Convolutional Neural Network (CNN) for traffic sign recognition. DWT is used as a preprocessing step to decompose traffic sign images into multiple frequency bands, thereby preserving critical structural features while suppressing noise. CNNs, on the other hand, excel at hierarchical feature learning and robust classification. By combining these two techniques, the proposed approach leverages their complementary strengths: DWT enhances the quality of input features, while the CNN ensures accurate and efficient classification. This synergy enables improved recognition performance, particularly under adverse conditions, while maintaining real-time processing capability.

The contributions of this work are as follows: First, a DWT-based preprocessing framework is proposed to extract relevant features by capturing both low and high-frequency information, which enhances contour and detail representation while effectively reducing noise, thereby improving robustness under adverse conditions such as rain or low-light environments; second, an optimized CNN architecture is designed specifically for traffic sign recognition, striking a balance between accuracy and computational efficiency; finally, comprehensive evaluations demonstrate that the proposed approach outperforms state-of-the-art methods, particularly in challenging and adverse environments.

The remainder of this paper is structured as follows: Section II reviews related work and key challenges in TSR; Sec-

tion III details the proposed DWT-CNN methodology; Section IV presents the experimental setup, results, and comparisons; and Section V concludes with a discussion of limitations and future research directions.

II. Related Work

Recent advances in Traffic Sign Recognition (TSR) have focused on improving detection accuracy, robustness under adverse conditions, and computational efficiency. Deep learning-based approaches, particularly those leveraging convolutional neural networks (CNNs) and attention mechanisms, have dominated the field, while hybrid methods integrating traditional image processing techniques have gained traction for enhanced feature extraction. Despite these advances, most methods still face challenges in balancing accuracy, robustness, and real-time performance, especially under adverse visual conditions or on embedded platforms.

A. Deep Learning: Based Approaches

CNN-based architectures remain the cornerstone of modern TSR systems due to their hierarchical feature learning capabilities. Early successes such as Faster R-CNN and SSD laid the groundwork for robust object detection pipelines. Modified Faster R-CNN models incorporating Feature Pyramid Networks (FPN) improve small and distant sign detection [5], but they remain computationally intensive, limiting their use in real-time embedded systems. SSD variants offer real-time detection with reasonable accuracy, yet they can struggle with very small or occluded signs. YOLO models, including Sign-YOLO (attention-based YOLOv7) [6] and YOLOv8 [7], achieve an excellent speed-accuracy trade-off, yet attention-based variants often increase model complexity and may require additional tuning for low-light or motion-blurred scenarios.

Transformers, such as Vision Transformers (ViTs) [8] and Swin Transformers [9], capture long-range dependencies and complex scene context effectively, improving recognition in crowded or cluttered scenes. However, their high computational cost and large memory footprint hinder deployment on low-power devices, motivating lightweight or hybrid variants.

Overall, deep learning approaches are powerful but often trade-off robustness, speed, and hardware efficiency, motivating research into methods like DWT-CNN that aim to preserve structural features while remaining efficient.

B. Hybrid and Preprocessing-Enhanced Methods

To mitigate issues such as illumination variations, noise, and motion blur, hybrid methods combining deep learning with traditional image processing have been explored. Discrete Wavelet Transform (DWT) preprocessing enhances feature robustness by decomposing images into multi-resolution subbands [10], preserving structural details while filtering noise. Studies [11] show that wavelet-based denoising improves detection in low-visibility conditions, though the additional preprocessing step may slightly increase inference time.

Multi-scale methods like SADANet [12] improve detection across varying sign sizes but can add network complexity. Multi-sensor fusion approaches, combining LiDAR and camera data [13], [14], reduce false positives in adverse weather, yet introduce high computational requirements and challenges in real-time deployment.

In comparison, the proposed DWT-CNN integrates wavelet preprocessing directly into the CNN pipeline, achieving robustness under adverse conditions while maintaining real-time performance. This positions the proposed approach as a balanced solution that addresses key limitations of both pure deep learning and hybrid methods, providing a strong foundation for real-world traffic sign recognition systems.

III. Proposed Method

In this work, this study proposes a novel hybrid architecture that integrates DWT with a CNN for robust and efficient Traffic Sign Recognition (TSR). The core idea is to leverage the multi-resolution analysis capability of DWT to enhance feature robustness under challenging conditions such as noise, motion blur, and lighting variations, while preserving a lightweight design suitable for real-time applications.

The proposed traffic sign recognition method is based on a hybrid architecture that combines the strengths of the Discrete Wavelet Transform (DWT) and Convolutional Neural Networks (CNNs) to ensure both robustness and efficiency in real-world scenarios. The overall pipeline is designed to enhance feature extraction under challenging conditions such as noise, motion blur, and varying illumination, while maintaining the low computational cost required for real-time deployment. The system begins by preparing and preprocessing a diverse set of traffic sign images to ensure uniform input and reduce variability, where input images are standardized to 128×128 pixels, and converted to grayscale to optimize computational efficiency. Next, the images are passed through a DWT-based transformation to extract multi-resolution frequency components, allowing the model to focus on structural details and suppress irrelevant noise. These enhanced representations are then fed into a lightweight CNN specifically designed for traffic sign classification. The model is trained with optimized strategies to prevent overfitting and maximize generalization. Finally, extensive evaluations are conducted to assess the system's accuracy, speed, and robustness under various environmental conditions. A comprehensive overview of the model is visually presented in Fig. 1 and the following subsections describe each phase of the proposed method in detail.

1) Preprocessing pipeline: Traffic signs suffer two main degradation types: natural and human-induced. Natural degradation is typically caused by long-term exposure to ultraviolet (UV) radiation and the use of retro-reflective materials, which may alter the sign's color and reduce its visibility. On the other hand, human-related degradation can take many forms, affecting the shape, color, components, or even the entire structure of the sign. To ensure accurate recognition of such signs despite potential deterioration, robust image preprocessing is essential. In the initial stage, the input images are standardized to improve compatibility with the Convolutional Neural Network (CNN). This includes resizing all images to a fixed dimension (640×640 pixels), then converting them to grayscale to reduce computational complexity while retaining edge and structural information.

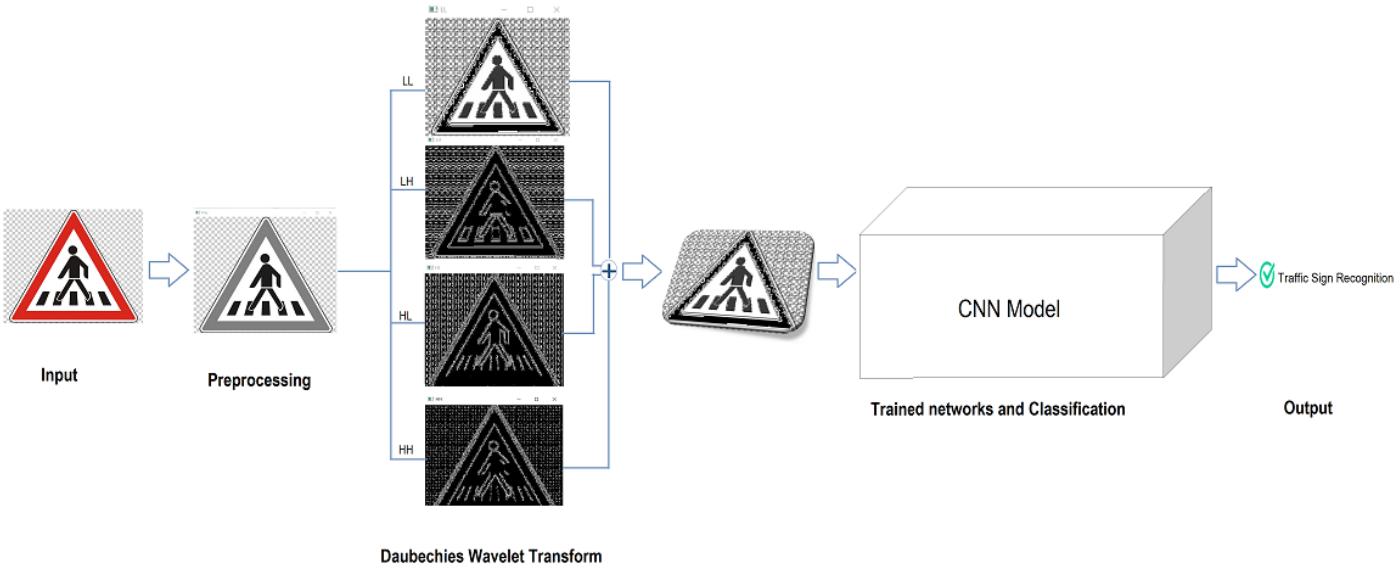

Fig. 1. Proposed TSR system based on wavelet transform and CNN: (a) Preprocessing, (b) Daubechies DWT: Decomposition Sub-bands, (c) Feature Fusion, and (d) CNN model.

2) DWT-Based feature enhancement: The DWT addresses key challenges in traffic sign recognition such as noise robustness and multi-scale feature extraction by decomposing images into frequency sub-bands. Unlike traditional methods, it preserves spatial and frequency information simultaneously, making it ideal for analyzing degraded signs [15].

There are two main types of wavelet transforms: the Continuous Wavelet Transform (CWT), designed for continuous signal decomposition, and the Discrete Wavelet Transform (DWT), commonly used in practical applications for analyzing both signals and images. The DWT decomposes input data into frequency sub-bands and is favored for its computational efficiency and simplicity.

Mathematically, the DWT is defined by Eq. 1, where $x(k)$ is a discrete-time signal with k samples and ψ is the discrete mother wavelet used for decomposition level i:

$$DWT_{ik} = \sum x(k)\psi_{i,k}(t) \tag{1}$$

Here, $\psi_{i,k}$ are scaled and translated versions of the mother wavelet $\psi(x)$:

$$\psi_{i,k}(x) = 2^{-i/2}\psi(2^{-i}x - k) \tag{2}$$

where i and k represent the dilation and translation parameters, respectively.

Various mother wavelets can be employed, such as Haar, Daubechies, Meyer, and Mexican Hat, depending on the characteristics of the application [16]. In this study, the Daubechies wavelet is employed due to its orthogonality and compact support, making it particularly effective for discrete signal processing.

To compute the DWT, the signal is passed through a low-pass filter $g[n]$ and a high-pass filter $h[n]$, as shown in Eq. 3:

$$y[n] = (x * g)[n] = \sum_{k=-\infty}^{\infty} x[k]g[n-k] \tag{3}$$

The Daubechies wavelet is adopted in this study due to its compact support and orthogonality, which efficiently capture localized signal features. The one-level decomposition (Fig. 2) partitions the image into:

- LL: Low-frequency components (approximation)
- LH/HL: (horizontal/vertical edges)
- HH: High-frequency components (diagonal details)

This decomposition process is illustrated in Fig. 2. The maximum decomposition level depends on the number of samples and the sampling frequency f_s. The frequency ranges for approximation and detail sub-bands at level i are given by:

$$A_i = \left[0, \frac{f_s}{2^{i+1}}\right], \tag{4}$$

$$D_i = \left[\frac{f_s}{2^{i+1}}, \frac{f_s}{2^1}\right]. \tag{5}$$

The 1-level decomposition was chosen to balance computational cost and feature granularity, as higher levels showed diminishing returns in preliminary experiments.

Instead of feeding raw images into the CNN, the decomposed sub-bands (LL, LH, HL, HH) are used as inputs, as illustrated in Fig. 3. This allows the network to learn more robust and localized features, particularly edges and shape patterns, which are crucial for accurate traffic sign recognition.

Using only the LL sub-band as input to a CNN may result in the loss of crucial information present in other sub-bands

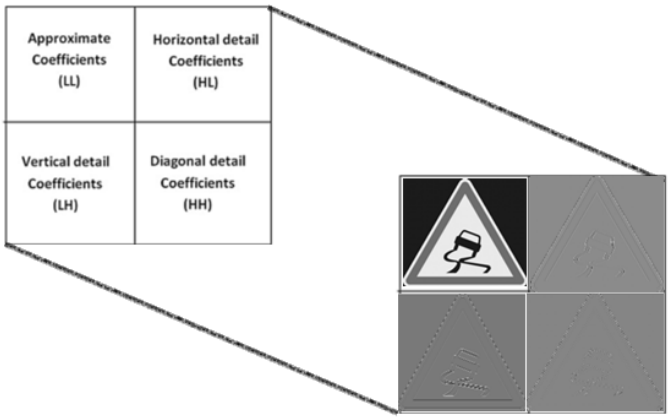

Fig. 2. Wavelet decomposition coefficients for the first level.

(LH, HL, and HH), which contain important details. By including all sub-bands as input, the model can manipulate details at various frequencies, ensuring a comprehensive representation of the image. The LH sub-band captures intermediate-frequency details, such as textures and edges, the HL sub-band contains high-frequency information, such as small details and color variations, and the HH sub-band contains very high-frequency information, such as very fine details and color variations. This approach enhances the model's ability to capture important image features, leading to improved performance and accuracy.

Fig. 3 presents the results obtained from each of the four sub-bands: LL, LH, HL, and HH.

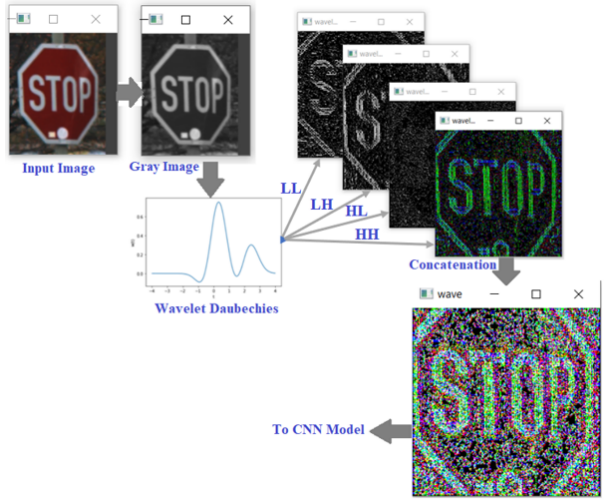

Fig. 3. Wavelet sub-band decomposition and concatenation (LL, LH, HL, HH).

In this work, a one-level 2D DWT decomposition is applied to each input image prior to classification to enhance the CNN's performance and improve feature representation.

3) CNN architecture: In this study, a DWT-based Traffic Sign Recognition System is introduced to enhance the feature extraction capability, improve detection performance, and reduce memory consumption and overall model size. As illustrated in Fig. 4, the system leverages the sub-bands generated by the Discrete Wavelet Transform (DWT). The main objective of this DWT-based approach is to extract robust and discriminative features from the DWT outputs and the backbone network, thereby improving detection accuracy under various conditions. Additionally, this mechanism guides the network's attention toward the most informative spatial regions particularly useful when traffic signs appear in cluttered or complex scenes.

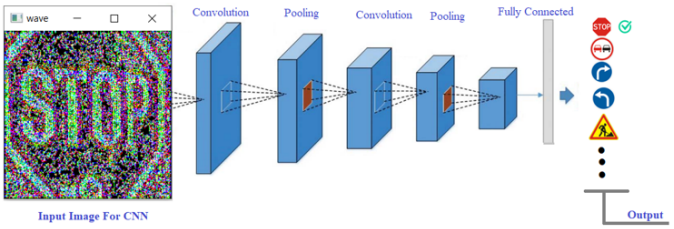

Fig. 4. Overall architecture of the proposed framework for TSR.

The convolutional layer [17] is a fundamental building block in CNNs and typically consists of three sequential components: convolution, activation, and pooling. In image processing tasks, the input is treated as a two-dimensional array, and the convolution operation applies a filter of size $S \times T$ over the input of size $M \times N$, producing an output of the same dimensions. The operation is defined as:

$$U^p_{mn} = \sum_{s=0}^{S-1} \sum_{t=0}^{T-1} h^{p-1}_{st} \cdot O^{p-1}_{m+s,n+t} + \theta^p_{mn} \tag{6}$$

Here, U^p_{mn} denotes the pre-activation value at location (m, n) in layer p, h^{p-1}_{st} is the (s, t)-th weight of the filter, $O^{p-1}_{m+s,n+t}$ is the corresponding input from the previous layer, and θ^p_{mn} is the bias term.

Subsequently, an activation function is applied. For instance, using the ReLU activation, the post-activation output becomes:

$$O^p_{mn} = \text{ReLU}(U^p_{mn}) \tag{7}$$

In the activation function layer, where no parameters are trained, the derivative for error backpropagation is calculated as follows:

$$\begin{aligned} \frac{\partial E}{\partial O^{p-1}_{mn}} &= \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \frac{\partial E}{\partial U^p_{mn}} \cdot \frac{\partial U^p_{mn}}{\partial O^{p-1}_{mn}} \\ &= \frac{\partial E}{\partial U^p_{mn}} \cdot \frac{\partial ReLU(O^{p-1}_{mn})}{\partial O^{p-1}_{mn}} \\ &= \begin{cases} \frac{\partial E}{\partial U^p_{mn}} & (O^{p-1}_{mn} > 0) \\ 0 & (O^{p-1}_{mn} \leq 0) \end{cases} \end{aligned} \tag{8}$$

When a pooling layer follows a convolutional layer, it aggregates spatial information within localized regions of the feature map to reduce dimensionality and retain salient

features. The general form of the pooling operation at position (m, n) in the p-th layer is defined as:

$$U_{mn}^{p} = \left(\frac{1}{S \times T} \sum_{(i,j) \in D_{mn}} (O_{ij}^{p-1})^{g} \right)^{\frac{1}{g}} \tag{9}$$

In this equation, O_{ij}^{p-1} denotes the output of the unit at position (i, j) in the $(p-1)$-th layer, and D_{mn} represents the pooling window of size $S \times T$ centered around (m, n). The parameters S and T are typically chosen to match the dimensions of the convolutional filter. The parameter g controls the type of pooling: setting $g = 1$ yields average pooling, while $g \to \infty$ approximates max pooling.

For example, setting $g = 1$ results in the average pooling operation, where the output is the arithmetic mean of the values within the pooling region:

$$U_{mn}^{p} = \frac{1}{S \times T} \sum_{(i,j) \in D_{mn}} O_{ij}^{p-1} \tag{10}$$

The averaging operation in the pooling layer reduces the spatial dimensions of the input while preserving essential features, thereby improving computational efficiency. When the pooling parameter g tends toward infinity, the operation becomes equivalent to max pooling, where the output at position (m, n) in the p-th layer is determined by selecting the maximum value within the pooling window D_{mn}. In this case, the pooling operation simplifies to:

$$U_{mn}^{p} = \max_{(i,j) \in D_{mn}} \{O_{ij}^{p-1}\} \tag{11}$$

Pooling layers, whether average or max, perform spatial downsampling to reduce feature map resolution while retaining discriminative information. Max pooling preserves the most dominant activations, while average pooling computes the mean of local regions. Since pooling layers do not involve trainable parameters, backpropagation in these layers is limited to the propagation of gradients from the current layer p to the previous layer $p-1$.

For average pooling, the gradient of the loss function E with respect to the output O_{mn}^{p-1} of the previous layer is computed by equally distributing the gradient across the pooling window. The corresponding gradient expression is given by:

$$\begin{aligned} \frac{\partial E}{\partial O_{mn}^{p-1}} &= \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \frac{\partial E}{\partial U_{mn}^{p}} \cdot \frac{\partial U_{mn}^{p}}{\partial O_{mn}^{p-1}} \\ &= \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \frac{\partial E}{\partial U_{mn}^{p}} \cdot \left(\frac{1}{S \times T} \sum_{(i,j) \in D_{mn}} \frac{\partial O_{ij}^{p-1}}{\partial O_{mn}^{p-1}} \right) \end{aligned} \tag{12}$$

In contrast, backpropagation through max pooling involves identifying the location of the maximum value within each pooling region. The gradient is passed only to the unit that contributed the maximum activation. This can be formulated as:

$$\begin{aligned} \frac{\partial E}{\partial O_{mn}^{p-1}} &= \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \frac{\partial E}{\partial U_{mn}^{p}} \cdot \frac{\partial U_{mn}^{p}}{\partial O_{mn}^{p-1}} \\ &= \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \frac{\partial E}{\partial U_{mn}^{p}} \cdot \frac{\partial}{\partial O_{mn}^{p-1}} \left(\max_{(i,j) \in D_{mn}} \{O_{ij}^{p-1}\} \right) \end{aligned} \tag{13}$$

In addition to pooling, normalization layers are often employed to stabilize and accelerate the training process. A common form of normalization is defined as:

$$U_{mn}^{p} = \frac{O_{mn}^{p-1} - \mu_{mn}}{\sqrt{c + \sigma_{mn}^{2}}} \tag{14}$$

where U_{mn}^{p} is the normalized output at position (m, n), O_{mn}^{p-1} is the input value from the previous layer, μ_{mn} and σ_{mn}^{2} are the local mean and variance computed over the pooling window D_{mn}, and c is a small constant added for numerical stability. These statistical quantities are computed as:

$$\mu_{mn} = \frac{1}{S \times T} \sum_{(i,j) \in D_{mn}} O_{ij}^{p-1} \tag{15}$$

$$\sigma_{mn}^{2} = \frac{1}{S \times T} \sum_{(i,j) \in D_{mn}} (O_{ij}^{p-1} - \mu_{mn})^{2} \tag{16}$$

These normalization steps help maintain consistent activation distributions across layers, which contributes to faster convergence and improved generalization during training.

4) System workflow overview: Fig. 4 illustrates the overall architecture of the proposed CNN, highlighting the sequence of convolution, activation, pooling, and normalization layers.

The pseudocode in Algorithm 1 describes the core stages of the proposed DWT-based traffic sign recognition system. It begins by converting the input image to grayscale and initializing all key components of the model, including a DWT layer, a convolutional backbone network, a neck module, and a detection head. During training, the image undergoes preprocessing and is passed through the DWT layer to extract frequency-based features. These features are then processed by the convolutional backbone, which comprises several layers of convolution, batch normalization, ReLU activations, dropout, and max pooling. The refined feature maps are passed through the neck and then to the detection head, which predicts both class probabilities and bounding box coordinates. The training loop optimizes the model using a combination of classification and localization losses. During inference, the input image follows the same processing pipeline, leading to the final prediction of bounding box coordinates and the corresponding class label.

A key originality of the proposed architecture lies in the explicit integration of the DWT as the first processing stage of the network. Unlike conventional CNN-based TSR systems that rely solely on spatial-domain features, the model leverages

Algorithm 1: Pseudocode of DWT-Based Traffic Sign Recognition System

```
Input: input_image
Output: bounding_box, class
Convert input_image to grayscale;
Initialize model components:;
    Define DWT layer (discrete wavelet transform);
    Define CNN backbone with convolutional, batchnorm, ReLU, dropout, max-pooling layers;
    Define Neck: convolutional layer with ReLU;
    Define Head: fully-connected layers for classification and bounding box regression;
Define data augmentation pipeline;
Load training dataset from CSV and image directory;
Initialize optimizer and loss functions (CrossEntropy and MSE);
while training not converged do
    Read batch of images, labels, and bounding boxes;
    Move data to device (GPU or CPU);
    Zero gradients;
    Forward pass:;
        Apply DWT to input images;
        Pass through CNN backbone:;
            - Conv + BatchNorm + ReLU + Dropout (64 filters);
            - Conv + BatchNorm + ReLU + Dropout (128 filters);
            - MaxPooling;
            - Conv + BatchNorm + ReLU + Dropout (256 filters);
            - MaxPooling;
        Pass through Neck (Conv + ReLU);
        Pass through Detection Head (FC for class and bbox);
    Compute classification and bounding box loss;
    Total loss = classification loss + bbox loss;
    Backpropagate loss and update model weights;
    Log average loss per epoch;
Save trained model weights to file;
Plot loss curve over epochs;
while testing do
    Read input image;
    Convert image to grayscale;
    Forward pass:;
        Apply DWT to input image;
        Pass through CNN backbone;
        Pass through Neck and Detection Head;
    Apply non-maximum suppression to filter redundant detections;
```

frequency-domain information to enhance robustness against noise, illumination changes, and motion blur. This design allows the backbone to operate on enriched multi-resolution representations, reducing the loss of fine structural details typically discarded in early convolutional layers. Moreover, by embedding the wavelet decomposition directly into the training pipeline rather than using it as a separate preprocessing step, the system jointly optimizes spatial and frequency features, ensuring better generalization under adverse conditions. This tight coupling of DWT with CNN layers distinguishes the proposed approach from existing hybrid methods and constitutes the main source of its improved performance and resilience.

IV. Results and Evaluation

This section demonstrates the effectiveness of the proposed DWT-CNN framework. First, the datasets used for training and evaluation are presented. Next, the implementation setup and evaluation protocol are detailed. Finally, a comparative analysis between the proposed method and other state-of-the-art approaches is provided.

1) Traffic signs dataset: The German Traffic Sign Recognition Benchmark (GTSRB) is a popular dataset for traffic sign recognition research [18] due to its large size (over 50,000 images of 43 different classes of traffic signs), diverse classes shown in Fig. 5 (including speed limits, priority, prohibitory, and warning signs), high-quality annotations, and wide use as a benchmark. Additionally, the dataset is publicly available and free to use, making it easily accessible for researchers. These factors have contributed to GTSRB's popularity as a benchmark for evaluating traffic sign recognition algorithms in recent years.

2) Implementation details: The proposed DWT-CNN model was implemented and evaluated using the German Traffic Sign Recognition Benchmark (GTSRB). For this study, a selected subset of GTSRB subclasses particularly relevant to real-world traffic scenarios was used. Each input image was resized to a resolution of 640×640 pixels for both the training and testing phases.

The experiments were conducted on a high-performance machine equipped with an Intel 13th Gen Core(TM) i7-13650HX CPU clocked at 2.60 GHz, 32 GB of RAM, and an NVIDIA GeForce RTX 4060 GPU with 8 GB of memory. The implementation was developed with CUDA version 12.7 and cuDNN version 9.1.0 for GPU acceleration.

This hardware configuration ensured efficient training and inference, especially for high-resolution input images and large-scale datasets.

The model was implemented using Python and PyTorch. The training process incorporated standard data augmentation techniques, including random rotation, horizontal flipping, and color jittering, to enhance generalization. Optimization was carried out using the Adam optimizer with an empirically tuned learning rate. The loss function combined cross-entropy for classification with mean squared error for bounding box regression.

3) Evaluation metrics: To evaluate the accuracy of the proposed method, this study employs standard metrics widely used in object detection and classification, including precision, recall, accuracy, and mean average precision (mAP). These metrics provide comprehensive insights into the method's performance in terms of correctness, completeness, and effectiveness, and also allow for fair comparisons with existing approaches in the literature.

Precision measures the proportion of correctly predicted positive samples out of all positive predictions, reflecting the model's accuracy in identifying relevant objects. Formally, it is defined as follows [19]:

Fig. 5. Summary of the dataset classes.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{17}$$

where:

- TP (True Positive): Correctly predicted object,
- FP (False Positive): Incorrectly predicted object.

Recall, also known as *sensitivity* or *true positive rate*, measures the ability of the model to detect all relevant instances of a class. It is computed as [20]:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{18}$$

where FN (False Negative) refers to instances where the model failed to detect existing objects.

Accuracy assesses the overall correctness of the model's predictions, considering both positive and negative classifications:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{19}$$

where TN (True Negative) represents the correct prediction of the absence of objects.

Finally, mean average precision (mAP) is a key performance indicator in object detection tasks. It is computed by averaging the average precision (AveP) across all object classes:

$$\text{mAP} = \frac{1}{n} \sum_{k=1}^{n} \text{AveP}_k \tag{20}$$

where AveP_k is the average precision for class k, and n is the total number of classes.

In summary, relying on these validation metrics is fundamental to rigorously assessing the reliability of the proposed method. They ensure that the evaluation is not biased toward a single criterion but instead provides a holistic view of detection performance, which is essential for benchmarking against other state-of-the-art models.

4) Result and Analysis: As evidenced by the confusion matrix in Fig. 6, the model achieves high classification accuracy across diverse sign categories, with minimal misclassifications even for visually similar classes. The synergy of DWT's noise resilience and the CNN's learning capability ensures reliable performance under real-world challenges such as occlusions and weather distortions.

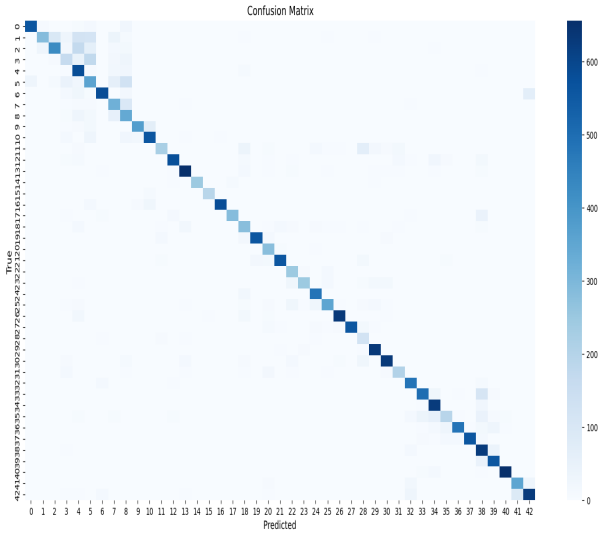

Fig. 6. Confusion matrix showing the performance of the proposed classification model.

Rigorous validation is essential in traffic sign recognition, where errors can directly affect road safety. Metrics such as precision, recall, F1-score, and mean average precision (mAP) not only quantify accuracy but also reveal the robustness of the model under varying conditions. High precision reduces false positives, which is critical to avoid incorrect driving actions, while high recall minimizes false negatives, ensuring that no critical sign is overlooked. These complementary measures together validate both the correctness and completeness of the proposed DWT-CNN model.

To assess the efficacy of the DWT-CNN model, this study conducts experiments on the GTSRB dataset, which comprises 43 subclasses grouped into four superclasses: prohibitory, mandatory, danger, and other. As shown in Table I, the model achieves strong performance across these categories, with par-

ticularly high accuracy for critical prohibitory and mandatory signs (e.g. stop: F1-score 0.96, speed_limit_20: 0.98). The results demonstrate the effectiveness of combining DWT's noise-robust feature extraction with CNN's discriminative learning, particularly in challenging real-world conditions.

TABLE I. VALIDATION OF THE PROPOSED METHOD ON SUB-CLASSES OF THE GTSRB

Class	Precision	Recall	F1-Score
0: speed_limit_20	0.98	0.97	0.98
1: speed_limit_30	0.86	0.90	0.88
2: speed_limit_50	0.78	0.97	0.86
3: speed_limit_60	0.97	0.97	0.97
4: speed_limit_70	0.98	0.96	0.97
5: speed_limit_80	0.93	0.87	0.90
6: restriction_ends_80	0.95	0.93	0.94
7: speed_limit_100	0.89	0.81	0.85
8: speed_limit_120	0.96	0.76	0.85
9: no_overtaking	0.91	0.78	0.84
10: no_overtaking_trucks	0.83	0.85	0.84
11: priority_at_next	0.83	0.72	0.77
12: priority_road	0.95	0.84	0.89
13: give_way	0.91	0.91	0.91
14: stop	1.00	0.92	0.96
15: no_traffic_both_ways	0.97	0.91	0.94
16: no_trucks	0.97	0.90	0.93
17: no_entry	0.96	0.79	0.87
18: danger	0.83	0.72	0.77
19: bend_left	0.98	0.93	0.95
20: bend_right	0.98	0.81	0.89
21: bend_double	0.95	0.94	0.95
22: bumpy road	0.97	0.96	0.97
23: slippery	0.97	0.90	0.93
24: road_narrows	0.88	0.84	0.86
25: construction	0.87	0.74	0.80
26: traffic_signal	1.00	0.99	0.99
27: pedestrian_crossing	1.00	0.92	0.96
28: children_crossing	0.96	0.86	0.91
29: bicycles	0.98	0.97	0.98
30: snow	0.99	0.98	0.99
31: animals	0.97	0.96	0.96
32: restriction_ends	0.97	0.87	0.92
33: go_right	0.95	0.98	0.96
34: go_left	0.99	0.97	0.98
35: go_straight	0.97	0.88	0.92
36: go_straight_right	0.98	0.97	0.98
37: go_straight_left	1.00	0.93	0.96
38: keep_right	0.97	0.94	0.96
39: keep_left	0.97	0.88	0.92
40: roundabout	1.00	0.92	0.96
41: restriction_ends_overtaking	0.98	0.96	0.97
42: restriction_ends_overtaking_trucks	0.96	0.91	0.93

Table II compares the proposed DWT-CNN model to state-of-the-art detectors based on key performance metrics: mean Average Precision (mAP), inference speed (FPS), memory usage (MB), and model size (number of parameters in millions). The DWT-CNN achieves a mAP of 0.97, outperforming all baseline models, including Faster R-CNN variants (ranging from 0.9062 to 0.9508) and YOLOv7/YOLOv8 (0.92–0.93), while maintaining real-time performance with 88.83 FPS. This is significantly faster than two-stage detectors such as Faster R-CNN (8.11–17.08 FPS) and even lightweight models like SSD MobileNet (66.03 FPS). Furthermore, the model accomplishes this with only 3.6 million parameters—substantially fewer than YOLOv7 (70.31M) and ResNet-based architectures (12.89–62.38M). Although the memory footprint (4493.43 MB) is higher than that of SSD MobileNet (94.70 MB), the trade-off is justified by a substantial gain in accuracy (+35.36% mAP) and speed (+22.8 FPS). These results highlight DWT-CNN's strong balance between accuracy and efficiency, making it well-suited for real-time traffic sign recognition tasks where high precision is essential.

Visual results in Fig. 7 illustrate the model's ability to accurately detect and classify traffic signs under challenging conditions from the GTSRB dataset. The test images include difficult scenarios such as low resolution, motion blur, occlusion, and poor lighting. Despite these adverse conditions, the proposed DWT-CNN model maintains high prediction accuracy, demonstrating both its robustness and strong generalization capabilities.

Table III presents a comparative analysis of the average precision and recall obtained by the proposed DWT-CNN model and several state-of-the-art traffic sign detection (TSD) approaches. The method achieves a precision of 0.98 and a recall of 0.97, matching YOLOv8 in precision while outperforming it by 3 percentage points in recall ($0.94 \rightarrow 0.97$). The DWT-CNN also surpasses two-stage detectors such as Faster R-CNN by 8% in precision and 10% in recall, and SADANet by 16% on both metrics. Compared to other single-stage models like YOLOv7 and YOLOv5, the model maintains highly competitive precision (e.g. YOLOv7: 0.97) and offers a substantial improvement in recall (YOLOv5: +14%). These results demonstrate the effectiveness of integrating Discrete Wavelet Transform (DWT) for noise-robust feature extraction with CNN-based classification, leading to a significant reduction in false negatives,a critical factor for traffic sign detection in safety-critical applications such as autonomous driving.

V. CONCLUSION

In this work, this study proposed a novel DWT-CNN architecture for traffic sign recognition, combining the robustness of Discrete Wavelet Transform-based feature extraction with the discriminative power of convolutional neural networks. Extensive experiments on the GTSRB dataset demonstrated that the model outperforms several state-of-the-art detectors, achieving high precision (98%) and recall (97%) while maintaining real-time inference speeds (88 FPS). The DWT-CNN showed strong resilience under adverse visual conditions such as blur, occlusion, and low lighting, confirming its robustness and generalization capability. These results highlight the model's suitability for real-world applications, particularly in safety-critical environments like autonomous driving systems. Future work will focus on deploying the model on embedded platforms and improving its performance in scenarios involving high-speed moving targets. Beyond these perspectives, several open challenges remain. For instance, how the DWT-CNN would perform when integrated into large-scale multimodal perception systems, or how it could adapt to unseen traffic sign classes across different countries, are questions that deserve further investigation. Addressing such issues could inspire new research directions and strengthen the impact of wavelet-based deep learning in intelligent transportation systems.

ACKNOWLEDGMENT

The authors would like to dedicate this research to their families for their continuous support and understanding. In particular, this work is dedicated to the memory of my father, Mr. Khalifa Trabelsi, in deep gratitude for his unwavering love, guidance, and inspiration.

TABLE II. PERFORMANCE COMPARISON OF THE PROPOSED DWT-CNN MODEL WITH STATE-OF-THE-ART OBJECT DETECTORS ACROSS KEY EVALUATION METRICS

Model	mAP (%)	FPS	Memory (MB)	Parameters (10^6)
Faster R-CNN ResNet 101 [21]	95.08	8.11	6134.71	62.38
Faster R-CNN ResNet 50 [21]	91.52	9.61	5226.45	43.34
Faster R-CNN Inception V2 [21]	90.62	17.08	2175.21	12.89
SSD MobileNet [21]	61.64	66.03	94.70	94.70
Zang et al. [22]	93.36	62	–	–
YOLOv7 [23]	93.00	35	321.67	70.31
YOLOv8 [24]	92.00	50	321.67	60
DWT-CNN (Ours)	0.97	88.83	4493.43	3,6

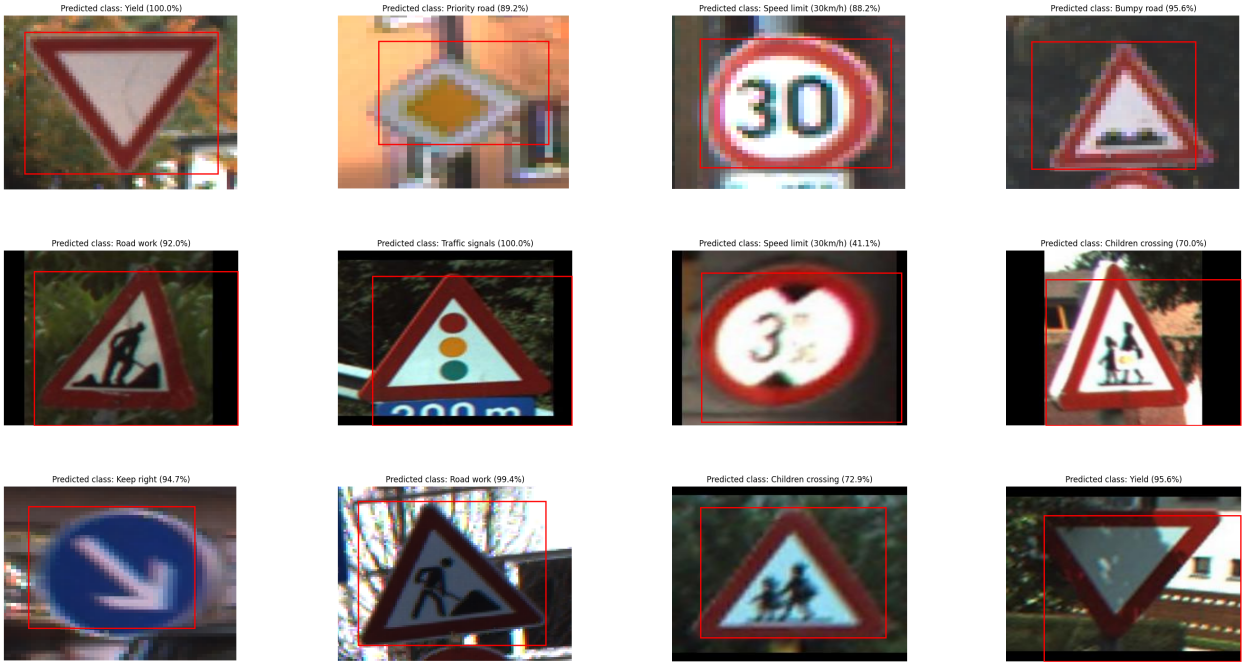

Fig. 7. Predicted class and confidence score for each image under various conditions using the proposed DWT-CNN model.

TABLE III. COMPARISON OF AVERAGE PRECISION AND RECALL OF THE PROPOSED METHOD WITH OTHER TSD APPROACHES

Model	Precision	Recall
SADANet [25]	0.82	0.81
Faster R-CNN [26]	0.90	0.87
YOLOv7 [23]	0.97	0.96
YOLOv8 [24]	0.98	0.94
Yolov5 [27]	0.85	0.83
DWT-CNN (Ours)	0.98	0.97

REFERENCES

[1] XIE, Guohuan, HESHAM, Syed Ariff Syed, GUO, Wenya, et al. A comprehensive survey on video scene parsing: advances, challenges, and prospects. [en ligne] arXiv preprint arXiv:2506.13552, 2025.

[2] SAH, Chandan Kumar, SHAW, Ankit Kumar, LIAN, Xiaoli, et al. Advancing autonomous vehicle intelligence: deep learning and multimodal LLM for traffic sign recognition and robust lane detection. [en ligne] arXiv preprint arXiv:2503.06313, 2025.

[3] HUYNH, Kha Tu, LE, Thi Phuong Linh, TRAN, Thien Khai, et al. A deep learning model of traffic signs in panoramic images detection. Intelligent Automation & Soft Computing, vol. 37, no. 1, 2023.

[4] MANI, Murali Krishnan, RAJAGOPAL, Sonaa, KAVITHA, D., et al. Deep learning-based traffic sign detection and recognition for autonomous vehicles. Digital Twin and Blockchain for Smart Cities, 2024, p. 407-428.

[5] SURESHA, R., MANOHAR, N., KUMAR, G. Ajay, et al. Recent advancement in small traffic sign detection: approaches and dataset. IEEE Access, 2024.

[6] MAHADSHETTI, Ruturaj, KIM, Jinsul, UM, Tai-Won. Sign-YOLO: traffic sign detection using attention-based YOLOv7. IEEE Access, vol. 12, p. 132689-132700, 2024.

[7] SOYLU, Emel, SOYLU, Tuncay. A performance comparison of YOLOv8 models for traffic sign detection in the Robotaxi-full scale autonomous vehicle competition. Multimedia Tools and Applications, vol. 83, no. 8, p. 25005-25035, 2024.

[8] DOSOVITSKIY, Alexey, BEYER, Lucas, KOLESNIKOV, Alexander, et al. An image is worth 16x16 words: transformers for image recognition at scale. [en ligne] arXiv preprint arXiv:2010.11929, 2020.

[9] LIU, Ze, LIN, Yutong, CAO, Yue, et al. Swin transformer: hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, p. 10012-10022.

[10] TRABELSI, Rim, NOURI, Khaled, AMMARI, Imen. Enhancing traffic sign recognition through Daubechies discrete wavelet transform and convolutional neural networks. In: 2023 IEEE International Conference on Advanced Systems and Emergent Technologies (IC_ASET), 2023, p. 1-6.

[11] LIU, Zhanwen, SHEN, Chao, QI, Mingyuan, et al. SADANet: integrating scale-aware and domain adaptive for traffic sign detection. IEEE Access, vol. 8, p. 77920-77933, 2020.

[12] SURESHA, R., MANOHAR, N., KUMAR, G. Ajay, et al. Recent advancement in small traffic sign detection: approaches and dataset. IEEE Access, 2024.

[13] CHEN, Peng, ZHAO, Xinyu, ZENG, Lina, et al. A review of research on SLAM technology based on the fusion of LiDAR and vision. Sensors, vol. 25, no. 5, p. 1447, 2025.

[14] WANG, Zhangjing, WU, Yu, NIU, Qingqing. Multi-sensor fusion in automated driving: a survey. IEEE Access, vol. 8, p. 2847-2868, 2019.

[15] TRABELSI, Rim, NOURI, Khaled. Boosting facial recognition accuracy through Daubechies wavelet techniques and deep convolutional models. In: 2024 International Symposium of Systems, Advanced Technologies and Knowledge (ISSATK), 2024, p. 1-6.

[16] GALAN-URIBE, Ervin, AMEZQUITA-SANCHEZ, Juan P., MORALES-VELAZQUEZ, Luis. Supervised machine-learning methodology for industrial robot positional health using artificial neural networks, discrete wavelet transform, and nonlinear indicators. Sensors, vol. 23, no. 6, p. 3213, 2023.

[17] YAGAWA, Genki, OISHI, Atsuya. Computational mechanics with deep learning: an introduction. Springer Nature, 2022.

[18] Belgium Traffic Sign Dataset. [en ligne] Disponible sur: https://btsd.ethz.ch/shareddata et https://www.kaggle.com/datasets/gtsrb-german-traffic-sign

[19] ZHU, Wen, ZENG, Nancy, WANG, Ning, et al. Sensitivity, specificity, accuracy, associated confidence interval and ROC analysis with practical SAS implementations. NESUG Proceedings: Health Care and Life Sciences, vol. 19, p. 67, 2010.

[20] POWERS, David. Ailab. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness & correlation. J. Mach. Learn. Technol., vol. 2, no. 22293981, p. 01, 2011.

[21] ARCOS-GARCÍA, Álvaro, ALVAREZ-GARCIA, Juan A., SORIA-MORILLO, Luis M. Evaluation of deep neural networks for traffic sign detection systems. Neurocomputing, vol. 316, p. 332-344, 2018.

[22] ZHANG, Jianming, LV, Yaru, TAO, Jiajun, et al. A robust real-time anchor-free traffic sign detector with one-level feature. IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 8, no. 2, p. 1437-1451, 2024.

[23] WANG, Chien-Yao, BOCHKOVSKIY, Alexey, LIAO, Hong-Yuan Mark. YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, p. 7464-7475.

[24] FARHAT, Wajdi, RHAIEM, Olfa Ben, FAIEDH, Hassene, et al. YOLO-TSR: a novel YOLOv8-based network for robust traffic sign recognition. Transportation Research Record, 2025.

[25] LIU, Zhanwen, SHEN, Chao, QI, Mingyuan, et al. SADANet: integrating scale-aware and domain adaptive for traffic sign detection. IEEE Access, vol. 8, p. 77920-77933, 2020.

[26] SHAO, Faming, WANG, Xinqing, MENG, Fanjie, et al. Improved faster R-CNN traffic sign detection based on a second region of interest and highly possible regions proposal network. Sensors, vol. 19, no. 10, p. 2288, 2019.

[27] ORTATAŞ, Fatma Nur, KAYA, Mahir. Performance evaluation of YOLOv5, YOLOv7, and YOLOv8 models in traffic sign detection. In: 2023 8th International Conference on Computer Science and Engineering (UBMK), 2023, p. 151-156.