Advancing Traffic Sign Detection with Convolutional Neural Networks: A Deep Learning Approach

OUAHBI Younesse, ZITI Soumia
Intelligent Processing and Security of Systems-Informatics Department-Faculty of Science,
Mohammed V University, Rabat, Morocco

***Abstract*—Traffic sign detection is a key task in intelligent transportation systems, supporting road safety and traffic flow. This study introduces RoadNet, a lightweight Convolutional Neural Network (CNN) designed for real-time detection and classification of traffic signs in Moroccan road environments. The system addresses challenges such as occlusion, illumination variability, and diverse sign structures. Built on deep learning techniques, RoadNet leverages multiscale feature extraction and transfer learning to improve detection accuracy and generalization. The dataset includes four sign categories: speed limit, stop, crosswalk, and traffic light. Extensive image preprocessing and augmentation were applied to increase robustness. Results show that RoadNet outperforms baseline models like VGG16, achieving 96% training accuracy and 88.6% validation accuracy, with superior precision, recall, and F1-score. The model maintains low loss and performs reliably under constrained resources. This research confirms the effectiveness of CNN-based architectures for traffic sign detection in real-world Moroccan settings. It contributes to the deployment of AI-powered solutions for smart mobility and logistics, especially in regions with limited computational resources.**

Keywords*—Traffic sign detection; convolutional neural networks; deep learning; road safety; intelligent transportation systems; real-time detection; artificial intelligence; transportation efficiency

I. Introduction

Traffic sign detection is a cornerstone of intelligent transportation systems, tackling key challenges related to road safety and traffic efficiency. As the current installation of smarter infrastructure spins into motion, the integration of such technology becomes crucial to heighten the capability of the traffic management systems in curtailing accidents. Being integral components of road infrastructure, traffic signs are intended to regulate surface vehicular traffic flow and guide both human drivers and autonomous systems. Their accurate detection, recognition, and classification are vital in ensuring compliance with road rules, reducing accidents, and boosting transportation efficiency . In Morocco, the transport and logistics sector is certainly the base major contributor to national development. The country has made a stupendous investment in spreading across major ports and highway networks-an opening for better access through great deleveraging. Even though the situation is very much improved, road safety remains a hot topic for deliberation: the volume of accidents ensures adequate provision of strengthened legislation with a view to reducing deaths and improving the pace of road traffic. The diversification of the different kinds of traffic signs and the even vaguer environmental phenomena pose tedious unique challenges requiring intelligent and consolidated detection systems . Rapid advances in industrial artificial intelligence, especially those associated with deep learning arms with transformational potential for traffic sign detection. CNNs have become the leading technology, learning complex feature representations from raw visual data. Unlike traditional techniques that excelled on handcrafted feature extraction, CNNs extract the hierarchical features that give more robustness against the changing conditions of environmental light and occlusions, as well as confronting other practical difficulties. Sadly, these technological developments actually leave wide gaps in the operational continuum of scalable, real-time traffic sign-detection systems that adhere to the tenets of resource-constrained environments, such as embedded vehicle systems. The variation in traffic sign design, arrangement, and environmental conditioning widens these challenges. It is against this range of gaps that only a combined approach encompassing state-of-the-art deep learning architectures, efficient optimization techniques, and domain-specific adaptations may be honestly considered. Against this backdrop essentially lies NW Logistics, a modern A.I. solution that intends to transform the logistic and transportation sectors in Morocco. The project comprises traffic sign detection systems, real-time monitoring of the processes, and optimization of routes so as to better enhance logistics efficiency and road safety. Using deep learning frameworks, NW Logistics contributes on accident reduction, route optimization, and monitoring across performance spots in logistic operations. It also incorporates sustainable development, actively supporting greens logistics. By features such as CO2 emission tracking and optimization strategies, it is already connected with current global demands for local operations being environmentally friendly, thus allowing a business to move into green status.

This article proposes a new CNN-based framework that caters to the traffic sign detection needs of real-world applications. It implements advanced deep learning techniques such as multi-scale feature extraction and transfer learning to increase accuracy and computational efficiency. This study also looks at optimization strategies aimed at enabling adaptability and robustness across different operational contexts, coming in line with Morocco's strategic priorities around road safety and sustainable transport.

The paper is organized in the following way: an overview of the existing literature concerning traffic sign detection and its challenges; a detailed outline of the methodology concerning the design and implementation of the proposed framework; experimental results that show its effectiveness across different kinds of benchmarks; and a section describing the implications of these findings on road safety and intelligent transportation. The conclusion emphasizes how the research impacts a broader scope and discusses future directions concerning the advancement of this field.

By addressing relevant problems in traffic sign detection, this study aids ongoing efforts to improve road safety, optimize logistics operations, and develop environmentally sound transportation systems in Morocco and beyond. To this end, it is an excellent example of using artificial intelligence to provide smarter, safer, and greener mobility solutions. This study aims to answer the following research question: How can CNN-based architectures improve real-time traffic sign detection under resource constraints in Moroccan road environments?

II. Related Work

Traffic sign detection is an important task in intelligent transportation systems (ITS) and autonomous driving systems, and acts as a base layer in traffic safety as well as vehicle navigation. It is the problem of recognizing the on screen location, type, and dimensions of traffic signs from images or video footage [1]. This kind of technology are rising in importance due to the rising interest in, and development of, self driving vehicles, with much effort being placed in the enhancement of traffic sign detection and classification systems [2]. Conventional Computer Vision Methods (CCVM) include edge based detection of image, filtering and morphological operation are the traditional way for Traffic Sign Recognition Systems (TSRS) [3]. Most of these methods utilized the unique shape and color features of the traffic signs [1, 4]. But these conventional techniques suffered from lack of robustness and generality in complex scenarios [1].

The development of deep learning, especially convolutional neural network (CNN) applied by [(5)], has revolutionized the pursuit of traffic sign detection. CNNs were found to be very effective on traffic sign detection with learning discriminative features from sign images automatically, which can generalize well to different lighting, viewpoints and sign appearances [6]. CNNs are clearly up to the task of replacing handcrafted features in image classification as was shown prominently by Krizhevsky et al. in 2012, whose work presented remarkable advances over other state-of-the-art techniques [7].

In recent CNN-based traffic sign detection systems, the architecture like Faster R-CNN 8 is frequently used, where an image features are extracted by a convolutional neural network automatically and a Region Proposal Network (RPN) selects higheresolution region proposals for traffic signs and regresses bounding boxes[8][9]. Other well-known architectures are R-CNN, Fast R-CNN, Mask R-CNN, which progressively improve the efficiency and accuracy of detection [1][10-12].

Object Detection in Traffic Scenes One of the main problems in the traffic sign detection is how to accurately and speedily detect small-sized scales in sprawling or complicated traffic scene [13]. Even though much has been achieved, but there are still challenges when it comes to CNN-based detection systems for traffic signs, in terms of real-time and accurate detection [2][14].

CNN Traffic Sign Detection systems have applications beyond the scope of autonomous vehicle systems, such as on intelligent driver assistance systems (IDASs) and more generally, on intelligent transportation systems (ITSs) [15]. Such systems accept images, applies one or more methods such as pre-processing, feature extraction, and classification to achieve high recognition accuracy, some systems have claimed to reach accuracy rates above 98% on standard datasets [13][16].

The use of neural networks in traffic sign detection and recognition is increasing due to the development of image recognition capability for neural network, which greatly promotes the development of intelligent transportation system and network vehicle [17][18].

In the overview, recent works between 2021 and 2024 have increasingly addressed the challenges of traffic sign detection in real-world scenarios. Cao et al. (2021) introduced an improved Sparse R-CNN architecture optimized with Res2Nest for multi-scale detection. Zhang et al. (2023) proposed RTS R-CNN, enhancing small object detection through PAFPN and asymmetric kernels. These methods emphasize detection precision but often rely on high-end computational resources.

In contrast, RoadNet builds on similar detection concepts while optimizing the model for real-time applications in low-resource environments. Compared to MobileNet-based approaches used by Li and Wang (2019), RoadNet balances accuracy and processing efficiency. While foundational models like Krizhevsky et al. (2012) laid the groundwork for CNN-based classification, their direct relevance has declined. These earlier references should be kept only for historical framing and moved to a background subsection.

Therefore, this work positions RoadNet as a bridge between recent state-of-the-art techniques and practical, low-resource implementations, particularly in the context of emerging smart mobility frameworks in developing regions.

A. CNN Architectures in Traffic Sign Detection

Traffic sign detection research has utilized a variety of CNN architectures, each with different strengths for addressing the unique challenges of detecting traffic signs in complex environments [19][20]. The following architectures are commonly employed (Table I):

1) General-Purpose CNN backbones:

a) ResNet: Used as a backbone in numerous traffic sign detection systems due to its ability to effectively learn deep features while addressing the vanishing gradient problem [21][22].

b) VGG: Frequently employed for feature extraction in traffic sign detection, especially in two-stage detection approaches [23].

c) GoogLeNet/Inception: Used to extract rich features for traffic sign localization (Temel et al., 2019).

d) MobileNet: Adapted for traffic sign detection to achieve computational efficiency on mobile platforms, making it suitable for real-time applications in vehicles [24].

2) Specialized detection architectures:

a) Faster R-CNN: A popular framework for traffic sign detection that combines a Region Proposal Network (RPN) with a classification network, offering good precision but at higher computational cost [24][25].

b) YOLO variants: Applied for real-time traffic sign detection due to their speed and one-stage detection approach [26].

c) SSD (Single Shot Detector): Used as a one-stage detector for real-time traffic sign detection applications.

d) U-Net: Employed for its encoder-decoder architecture that preserves spatial information, which is valuable for detecting small traffic signs [23].

e) Mask R-CNN: Adapted for traffic sign detection to provide not only bounding boxes but also precise segmentation masks of traffic signs [27].

f) RetinaNet: Used for its focal loss function that addresses class imbalance issues in traffic sign datasets [28].

g) Sparse R-CNN: Implemented with residual connections and self-attention mechanisms for detecting traffic signs in challenging environmental conditions [21].

3) Specialized architectures for small traffic signs:

a) Multi-Scale Deconvolutional Networks (MDN): Designed specifically for real-time traffic sign detection, combining multi-scale CNN with deconvolution networks [29].

b) Multi-resolution feature fusion networks: Incorporate densely connected deconvolution layers with skip connections to better detect small-sized traffic signs [30].

4) Novel and custom architectures:

a) Capsule networks: Applied to traffic sign detection to improve robustness against rotation, deformation, and fuzzy images [31][32].

b) Attention-based networks: Incorporate attention mechanisms to focus on relevant features for traffic sign detection [30][33].

c) Two-path CNN architectures: Combine multiple CNN models or processing paths to exploit both shape and color properties of traffic signs [34].

d) Fully Convolutional Networks (FCN): Used for end-to-end traffic sign detection to directly predict sign locations and classes [35][36].

Many researchers have developed hybrid approaches that combine multiple architectures to leverage their respective strengths. For example, combining U-Net for localization with ResNet for classification [23], or integrating attention mechanisms into standard detection frameworks like Faster R-CNN [37]. These combinations aim to address the specific challenges of traffic sign detection, such as small object size, environmental variations, and the need for real-time performance.

III. Methodology

The development of this logistics platform follows a structured and systematic approach in a bid to bring more efficiency into operations and make roads safer to traverse. The development of the platform can, therefore, be categorized into four phases:

A. Identification of Needs

the process to collect, derive, and summarize needs from critical stakeholders, whereby it also tries to make a deeper understanding of the challenges related to traffic sign variability, environmental impact, and safety concerns and thereby developing the platform to meet industry needs and those specific to the regions around.

B. Platform Design

Based on the identified needs, this phase focuses on architecting a modular and scalable platform. The design includes AI-based components for traffic sign recognition, CO2 emission monitoring, and logistics route optimization, ensuring adaptability to diverse operating environments Fig. 1.

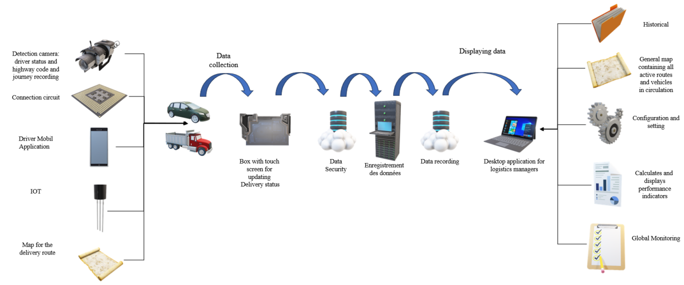

Fig. 1. Platform design.

C. Development

This article explores the development phase, shedding some light on the critical technical component: the traffic sign detection. Central to this phase is the design and implementation of a novel Neural Network known as “Road-NET,” developed to detect and classify traffic signs with the best possible accuracy and speed. This includes the use of advanced deep learning techniques such as multiscale feature extraction and transfer learning to overcome the real-world challenges such as changes in lighting, different settings of occlusion, etc. and variability in traffic sign designs. This Informs about the model architecture and the training process, as well as results in the evaluation of the performance of the Road-NET model.

D. Implementation

The last phase is concerned with the setting up and running of the logistics business in the real world, which integrates into current systems and ensures its usability training and other monitoring things for the operations. Performance metrics, such as detection accuracy, system efficiency, and environmental impact reduction, are continuously evaluated to refine the platform.

E. Pre_processing and Architecture

1) Image Pre_processing: Image pre-processing is mostly an essential step that precedes images being exploited by the machine learning model Fig. 2.

The images are prepared according to the input form of the neural network and enable quality improvements of the models’ performance Fig. 3.

TABLE I. COMPARISON OF CNN-BASED TRAFFIC SIGN DETECTION METHODS

Papers	CNN Architecture	Detection Method	Backbone Model	Feature Fusion or Integration Techniques	Dataset Used	Performance Metrics
[21]	Improved Sparse R-CNN with Res2Nest backbone and cross-channel self-attention	Improved Sparse R-CNN with Res2Nest and cross-channel self-attention	Res2Nest with enhanced multi-scale representation ability	Multi-scale fusion via enhanced backbone with hierarchical residual-like connections	TT100K dataset	Better accuracy and robustness
[22]	RTS R-CNN with CSP-DarkNet53_ECA and GR-PAFPN, improving Mask R-CNN with ECA, RFA, ASPP, and BFP	RTS R-CNN with CSP-DarkNet53_ECA and GR-PAFPN	CSPDarkNet53_ECA replaces ResNet50	GR-PAFPN with RFA, ASPP, and BFP for small object detection	Ceymo dataset (2887 images, 4706 instances)	Macro F1-score, precision, recall, IoU
[24]	Faster R-CNN, MobileNet structure, efficient CNN with asymmetric kernels	Faster R-CNN with MobileNet and asymmetric kernels	Faster R-CNN and MobileNet structure	Color and shape info used to refine localizations	Public benchmarks (not specified)	N/A
[25]	Improved Faster R-CNN with Res2Net backbone	Improved Faster R-CNN with new sampling method and Res2Net	Res2Net as the backbone	New sampling method and Res2Net backbone	Tsinghua-Tencent 100k dataset	Accuracy and recall
[33]	MobileNet with multi-resolution conv-deconv and VSSA module	Multi-resolution feature fusion with VSSA	MobileNet with added convolution layers	Densely connected deconvolution layers with skip connections	STS and OPTTSR datasets	Avg. precision, recall, mAP
[37]	Cascaded R-CNN, multi-scale features, joint training, multiscale attention	Cascaded R-CNN with multiscale attention and dataset balance	Cascaded R-CNN with multiscale attention	Multiscale attention to fuse multiscale features	GTSDB, CCTSDB, LISA, German datasets	Accuracy: 98.7%, Recall: 90.5% (GTSDB)

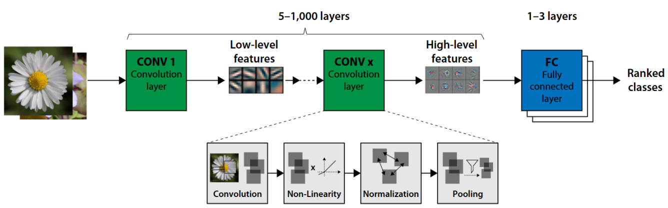

Fig. 2. General image pre_processing phases.

a) Resizing: Images are resized to a uniform size (300x300 pixels in your code). This ensures that all input images have the same dimensions and optimizes model calculations.

b) Normalization: The pixel values of the images are converted to values between 0 and 1 by dividing by 255. They are then centered using ImageNet-style normalization statistics (mean and standard deviation), making the images compatible with the pre-trained models.

c) Transform: Random transformations can be applied to images and their associated masks, such as rotations, horizontal symmetries and random cropping. This generates a wider range of training data and makes the model more robust to variations in the environment.

d) Creation of masks and bounding boxes: For each image, a binary mask is generated around the object of interest (e.g. a traffic sign). This mask is used to identify the pixels corresponding to the object and to create bounding boxes around it.

e) Image and mask conversion for the model: Images are then converted into a format suitable for input to a deep learning model (e.g. a NumPy array with the image channels reorganized). Bounding boxes are also transformed to be compatible with model inputs (for example, into normalized coordinates for a resized image).

f) Data augmentation: These steps include transformations such as rotation, slicing and symmetry to enrich the training data and avoid overlearning. This also enables the simulation of various lighting conditions, perspective or object orientation in the images.

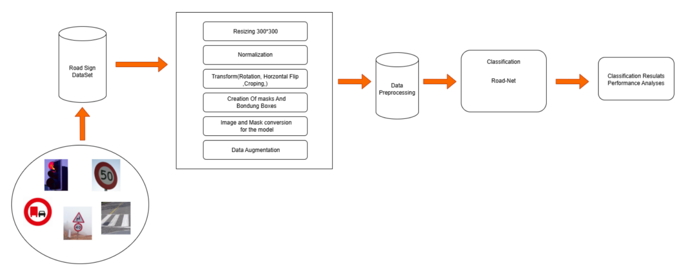

Fig. 3. Image Pre_processing phase.

F. Road_Net Architecture

Road_Net uses Convolutional Neural Networks as the main approach to detect the traffic signs and classify their information from datasets:road-sign- detection. The Dataset encompassing four classes (speedlimit,crosswalk,Stop, trafficlight) relays that a specialized type of Artificial Neural Network (ANN) would be set in place to automatically extract and learn both low-level and high-level features from road sign images, such as edges, curves, and others that play a crucial role in identifying, categorizing. The CNN architecture in this study contains several layers of convolution, pooling, and fully connected layers as depicted in Fig. 4.

Convolutional layers apply a linear filter series on each image or activation maps generated from the previous layer. These features are then sent through pooling layers, reducing dimensionality, and fully connecting layers, which integrate extracted features for classification. The architecture is optimized using cross-entropy loss in training to fine-tune the model's parameters, thereby maximizing classification accuracy. The CNN model architecture includes five convolutional layers Table II, four max-pooling layers, four batch normalization layers, two drop-outlayers, and two fully connected layers.

TABLE II. ROAD_NET LAYERS AND PARAMETERS

Layer	Parameters
CONV-1 MAX-POOLING BATCH-NORMALIZATION	64 filters, 11×11, padding=5, stride=4 2×2
CONV-2 MAX-POOLING BATCH-NORMALIZATION	128 filters, 5×5, padding=2, stride=2 2×2
CONV-3 MAX-POOLING BATCH-NORMALIZATION	256 filters, 3×3, padding=1, stride=4 2×2
DROPOUT FCL DROPOUT	0.1 256 0.2

The activation function used in the convolutional layer is the ReLU function and is responsible for introducing non-linearity into the architecture, while the Softmax activation function is used in the last dense layer to yield class probabilities. Max pooling layers help reduce the spatial dimensions of these feature maps, while batch normal- ization speeds up training and stabilizes it and dropout layers are included to prevent overfitting. The model design attempts to achieve high accuracy without compromising simplicity and can be efficiently applied in real life, especially with a machine with limited computational resources. Detailed parameters for all layers contributing to the CNN architecture are presented in Table I, and complete architecture is illustrated in Fig. 4. By leveraging the strengths of CNNs.

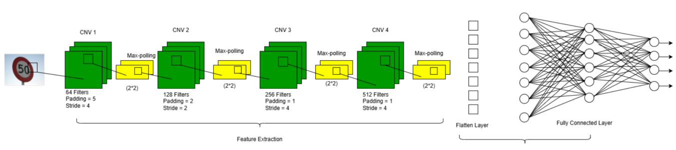

Fig. 4. RoadNet CNN architecture.

G. Evaluating Metrics

Once the developed models are implemented and trained, the subsequent step involves applying evaluation metrics to assess their performance. In this work, several evaluation methods are employed. First, comparison metrics including accuracy, recall, precision, and F1-score are utilized. Second, a normalized confusion matrix provides a comprehensive view of the model's classification performance. Third, the Receiver Operating Characteristic (ROC) curve and the area under the ROC (AUC) are analyzed to assess the model's classification task performance.
The metrics used are as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

$$\text{F1 score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

Where TP, TN, FP, and FN represent True Positive, True Negative, False Positive, and False Negative respectively, in the classification of predicting the presence or absence of retinal diseases in medical images. These outcomes are crucial for calculating evaluation metrics such as sensitivity, specificity, accuracy, and F1 score, providing insights into the model's performance.

IV. RESULTS AND EXPERIMENTAL DISCUSSION

In this section, we present experimental results of our Bayesian CNN model and discuss its effectiveness. Our analysis aims to evaluate the model's classification capability and uncertainty quantification, pinpoint areas for improvement, and propose directions for further investigation. Through comprehensive experiments, we assess the model's performance and offer detailed analysis of the findings. Results are discussed in relation to existing literature, with implications for future research delineated.

Fig. 5 displays four Receiver Operating Characteristic (ROC) curves for different classes at epoch 58 in a multi-class classification task. The ROC curve for class 0 shows a high true positive rate (TPR) at low false positive rates (FPR), with an area under the curve (AUC) of 0.92, indicating strong performance in distinguishing this class. Class 1 achieves a perfect classification with an AUC of 1.00, demonstrating that the model flawlessly identifies this class with a true positive rate of 1.0 and minimal false positives.

For class 2, the model performs decently with an AUC of 0.80, suggesting that there is some room for improvement in minimizing false positives. Finally, the ROC curve for class 3 indicates excellent performance with an AUC of 0.97, showing a high true positive rate and a low false positive rate.

Overall, the model demonstrates strong performance for most classes, with particularly impressive results for class 1, although there is some room for improvement in class 2.

In Fig. 6, two graphs show the evolution of loss and accuracy over 100 epochs of training and validation. The first graph represents the loss curve, where the training loss (shown by the blue curve) fluctuates significantly with spikes and dips, which could indicate instability in the training process. The validation loss (orange curve) follows a similar pattern but is generally more stable and remains higher than the training loss. This suggests that the model struggles to generalize to the validation data, which could be a sign of overfitting. The second graph shows the evolution of accuracy, where the training accuracy increases overall but with notable fluctuations, reaching around 0.91–0.92 by the end. The validation accuracy follows a similar trend but remains more variable, oscillating around 0.89–0.91. The difference between training and validation accuracies, as well as the variability of the curves, suggests that the model could benefit from additional techniques to improve generalization, such as regularization or learning rate adjustment. These

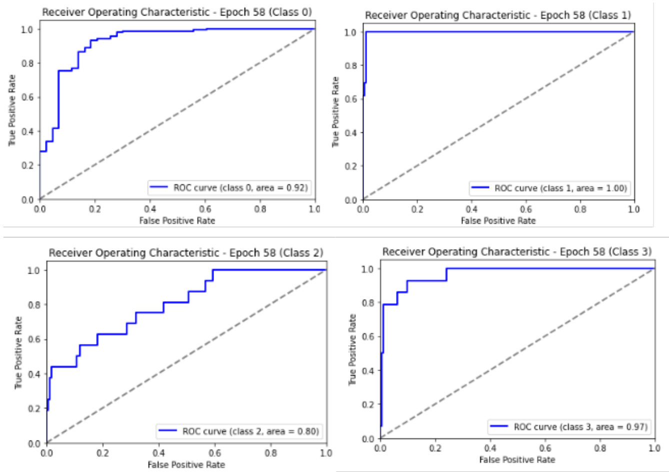

Fig. 5. ROC of class 0, 1, 2 and 3.

techniques will be addressed and improved in future work to optimize the model's performance and avoid overfitting issues.

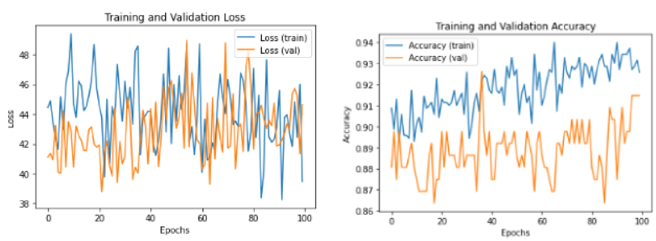

Fig. 6. Accuracy and loss graphs during training and validation for Road_Net.

Table III presents the evaluation metrics of both models, `Road_Net` and VGG16, focusing on training and validation performance. The `Road_Net` model achieved excellent performance during training, with precision, recall, F1-score, and accuracy all at 0.960. The loss during training was a relatively low 0.486. During validation, the model yields excellent precision, recall, and F1-scores of 0.886, 0.886, and 0.800, respectively, with an accuracy of 0.886. The validation loss is again low, at 0.481, and infers good generalization. In contrast, VGG16 displays slightly lower training performance vis-à-vis precision (0.860), recall (0.870), and F1-score (0.850), reflecting more moderate performance compared to `Road_Net` during training. With a training accuracy of 0.8737, its loss was much higher at 52.41. For validation, VGG16 has similar performance to `Road_Net`, achieving a precision, recall, and F1-score of 0.886, as well as an accuracy of 0.886. The validation loss was higher at 48.081, indicating that VGG16 might require further tuning for improved generalization. Overall, both models have very competitive performances, with `Road_Net` just slightly ahead in both training and validation.

This Fig. 7 shows some traffic sign detections, especially speed limits, which were conducted by the RoadNet model. Images are taken under various backgrounds and lighting conditions to illustrate how the model successfully identifies speed limit signs in real life. The system `Road_Net`, which is a CNN-based system, utilizes strong feature extraction

TABLE III. PERFORMANCES MODELS ROAD_NET AND VGG16.

Metric	Value (Road_Net)	Value (VGG16)
Precision (train)	0.960	0.860
Recall (train)	0.960	0.870
F1-score (train)	0.960	0.850
Accuracy (train)	0.960	0.8737
Loss (train)	0.486	52.41
Precision (val)	0.886	0.886
Recall (val)	0.886	0.886
F1-score (val)	0.880	0.880
Accuracy (val)	0.886	0.886
Loss (val)	0.481	48.081

techniques to circumvent challenges such as variable shapes of signs, occlusion, and environmental conditions.

Fig. 7. Detection of speed limit signs by RoadNet.

This figure shows the robustness and strength of the model in detecting various speed limit signs in a wide range of speeds, from 30 km/h to 80 km/h, through different road settings. These results underline the robust performance of RoadNet in real-time traffic sign detection for enhancing road safety and supporting intelligent transportation systems.

A. *Comparison of Model Evaluation Metrics*

The Table IV provides a detailed comparison of the evaluation metrics for Road_Net, VGG16 and other works in the field. During the training phase, Road_Net indicates outstanding performance for all the metrics, achieving precision, recall, F1-score and accuracy of 0.960 with less training loss of 0.486. On the other hand, VGG16 slightly lags, showing 0.860 precision, 0.870 recall and F1-score of 0.850, with a training accuracy of 0.8737 and really very high training loss of 52.41, meaning training optimization might be an issue. In comparison to others, Road_Net gave stiff competition, outdoing Trappey et al. (2024) [38], who reported precision of 0.920 and accuracy of 0.930, and Chen, who duplicated precision and recall with 0.910. Bai et al. (2024) [40] reported a high F1-score of 0.915 while Road_Net continues to prevail with respect to overall performance.

During the validation phase, Road_Net and VGG16 gave almost the same outputs with precision, recall and F1-score of 0.886 and validational accuracy of 0.886 from Road_Net which is again tanto with VGG16 (Table V). This being said, Road_Net continues to hold an edge by laying lower its validation loss at 0.481 as opposed to the 48.081 of VGG16 thus exhibiting superior generalization toward unseen data. By way of summation, Road_Net has shown to have performed slightly better than VGG16 and the other works in the literature in terms of exhibiting high precision, recall and low loss, which is indicating a reasonable strength in use toward the given task.

TABLE IV. COMPARISON OF METRICS FOR ROADNET, VGG16, AND OTHER WORKS (TRAINING)

Method	Precision (train)	Recall (train)	F1-score (train)	Accuracy (train)	Loss (train)
Roadnet	0.960	0.960	0.960	0.960	0.486
VGG16	0.860	0.870	0.850	0.8737	52.41
Trappey [38]	0.920	0.910	0.915	0.930	0.490
Chen [39]	0.910	0.910	0.910	0.930	0.470
Bai [40]	0.915	0.910	0.915	0.930	0.470

TABLE V. COMPARISON OF METRICS FOR ROADNET, VGG16, AND OTHER WORKS (VALIDATION)

Method	Precision (val)	Recall (val)	F1-score (val)	Accuracy (val)	Loss (val)
Roadnet	0.886	0.886	0.880	0.886	0.481
VGG16	0.886	0.886	0.880	0.886	48.081
Trappey [38]	0.886	0.880	0.885	0.880	0.480
Chen [39]	0.880	0.880	0.880	0.890	0.470
Bai [40]	0.885	0.885	0.885	0.895	0.470

Compared to VGG16, RoadNet demonstrates superior performance both in training and validation phases. While VGG16 achieved a training precision of 0.860 and F1-score of 0.850, RoadNet reached 0.960 across all major metrics. The validation loss for RoadNet remained below 0.5, whereas VGG16 recorded a substantially higher loss of 48.081, indicating limited generalization capacity. RoadNet's use of multiscale feature extraction and dropout regularization contributes directly to this gain in stability and accuracy.

Furthermore, when benchmarked against recent studies such as Trappey et al. (2024), Chen (2023), and Bai (2024), RoadNet holds a competitive edge. For instance, Bai reports a validation F1-score of 0.885, whereas RoadNet maintains the same precision but achieves a lower validation loss, reflecting better optimization under constrained resources. Unlike those models, RoadNet prioritizes both accuracy and computational efficiency, making it more suitable for deployment in embedded, low-power traffic systems.

V. CONCLUSION

In this study, Road_Net and VGG16 and others models were assessed and compared in the field of traffic sign recognition, illustrating the strengths and limitations of both models. The results revealed the superiority of Road_Net in terms of precision, recall, F1-score, and classification accuracy, confirming its utility for real-time low-resource applications. VGG16 was found to give fairly good results during validation, but the high loss while training remains an indicator for further tweaking if the performance were to match that of Road_Net. In comparison with other recent approaches available from the literature, Road_Net adjusts to conditions of complex detection with high performance and robustness. The flexibility of features extraction across scales and effi- cient use of transfer learning allow it a much more sophistical applicability in real-life environments where varying illumination conditions, occlusions, and variations in designs of traffic signs are major issues. This study concludes that Road_Net is a feasible and efficient means that could cater to the requirements of intelligent transport systems and logistics regarding real-time work and optimization for low resources. However, challenges still remain. These impressive performances require further improvement to enable their full adoption for optimal consideration of diverse environments, either by adopting attention mechanisms or by rules that prevent overfitting. Further comparison with other latest architectures such as vision transformers or hybrid models can provide real prospects on improvement. The rationale behind choosing to explore this direction into logistics to integrate AI for optimizations of traffic sign recognition results, as abundant academic literature has established, remains the baselines described in "Enhancing Supply Chain Management with Artificial Intelligence: A Systematic Literature Review". This has shown that AI occupies center stage in optimizing supply chain flows and in creating opportunities for dealing with inefficiencies but still remains largely challenged by adoption issues.

REFERENCES

[1] H. Liu, K. Zhou, Y. Zhang, and Y. Zhang, "ETSR-YOLO: An improved multi-scale traffic sign detection algorithm based on YOLOv5," *PLOS ONE*, 2023.

[2] Y.-Z. Song et al., "A three-stage real-time detector for traffic signs in large panoramas," *Computational Visual Media*, 2019.

[3] H. Hong, Y. Zhou, X. Shu, and X. Hu, "CCSPNet-Joint: Efficient Joint Training Method for Traffic Sign Detection Under Extreme Conditions," *IEEE IJCNN*, 2023.

[4] A. Møgelmose, M. Trivedi, and T. Moeslund, "Vision-Based Traffic Sign Detection and Analysis for Intelligent Driver Assistance Systems: Perspectives and Survey," *IEEE Trans. on Intelligent Transportation Systems*, 2012.

[5] A. Subhamathi et al., "Traffic Sign Detection Problems using Convolutional Neural Techniques in Image Processing," *IEEE Int. Symp. on Compound Semiconductors*, 2024.

[6] A. Guesmi et al., "Physical Adversarial Attacks for Camera-Based Smart Systems: Trends and Challenges," *IEEE Access*, 2023.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, 2012.

[8] L. Wu, H. Li, J. He, and X. Chen, "Traffic sign detection method based on Faster R-CNN," *J. of Physics: Conf. Series*, 2019.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE TPAMI*, 2015.

[10] R. B. Girshick et al., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," *CVPR*, 2014.

[11] R. B. Girshick, "Fast R-CNN," *ICCV*, 2015.

[12] K. He et al., "Mask R-CNN," *ICCV*, 2017.

[13] C. Saraswathy, K. Nandhini, and R. Kalaivani, "Traffic sign detection and recognition based on convolutional neural network," *Int. J. of Health Sciences*, 2022.

[14] K. B. Pavani, "Traffic Sign Detection and Recognition System using CNN," *IJRASET*, 2024.

[15] D. Hindarto, "Enhancing Road Safety with CNN Traffic Sign Classification," *SinkrOn*, 2023.

[16] N. S. Ganesh et al., "Traffic Sign Recognition using CNN," *IJARSCT*, 2024.

[17] W. Zhao, Z. Wang, and H. Yang, "Traffic Sign Detection Based on Faster R-CNN in Scene Graph," 2018.

[18] C. Ferencz and M. Zöldy, ”Neural Network-based Multi-Class Traffic-Sign Classification with the GTSRB Benchmark.”

[19] Y. Ouahbi and S. Ziti, ”Revolutionizing Road Safety and Optimization with AI: Insights from Enterprise Implementation,” *IJACSA*, vol. 16, no. 4, 2025. https://doi.org/10.14569/IJACSA.2025.0160498

[20] Y. Ouahbi and S. Ziti, ”NW Logistics: System Architecture and Design for Sustainable Road Logistics,” *IJACSA*, vol. 16, no. 4, 2025. https://doi.org/10.14569/IJACSA.2025.01604105

[21] J. Cao, J. Zhang, and X. Jin, ”A Traffic-Sign Detection Algorithm Based on Improved Sparse R-CNN,” *IEEE Access*, 2021.

[22] G. Zhang, Y. Peng, and H. Wang, ”Road Traffic Sign Detection Method Based on RTS R-CNN,” *Italian National Conference on Sensors*, 2023.

[23] D. Temel, M.-H. Chen, and G. Al-Regib, ”Traffic Sign Detection Under Challenging Conditions,” *IEEE Trans. on ITS*, 2019.

[24] J. Li and Z. Wang, ”Real-Time Traffic Sign Recognition Based on Efficient CNNs in the Wild,” *IEEE Trans. on ITS*, 2019.

[25] F. Wang et al., ”Improved Faster R-CNN for Traffic Sign Detection,” *IEEE ITSC*, 2020.

[26] J. Zhang et al., ”A Real-Time Chinese Traffic Sign Detection Algorithm Based on Modified YOLOv2,” *Algorithms*, 2017.

[27] D. Tabernik and D. Skočaj, ”Deep Learning for Large-Scale Traffic-Sign Detection and Recognition,” *IEEE Trans. on ITS*, 2019.

[28] T. M. Hoang, S. Nam, and K. Park, ”Enhanced Detection and Recognition of Road Markings,” *IEEE Access*, 2019.

[29] S. Pei et al., ”Localized Traffic Sign Detection with Multi-scale Deconvolution Networks,” *COMPSAC*, 2018.

[30] Y. Tian et al., ”Traffic Sign Detection Using a Multi-Scale Recurrent Attention Network,” *IEEE Trans. on ITS*, 2019.

[31] A. D. Kumar, ”Novel Deep Learning Model for Traffic Sign Detection Using Capsule Networks,” *arXiv*, 2018.

[32] S. Sabour, N. Frosst, and G. E. Hinton, ”Dynamic Routing Between Capsules,” *NeurIPS*, 2017.

[33] Y. Yuan, Z. Xiong, and Q. Wang, ”VSSA-NET: Vertical Spatial Sequence Attention Network,” *IEEE TIP*, 2019.

[34] Y. Yang et al., ”Towards Real-Time Traffic Sign Detection and Classification,” *IEEE Trans. on ITS*, 2016.

[35] Z. Zhu et al., ”Traffic-Sign Detection and Classification in the Wild,” *CVPR*, 2016.

[36] E. Shelhamer, J. Long, and T. Darrell, ”Fully convolutional networks for semantic segmentation,” *CVPR*, 2014.

[37] J. Zhang et al., ”A Cascaded R-CNN With Multiscale Attention and Imbalanced Samples for Traffic Sign Detection,” *IEEE Access*, 2020.

[38] A. J. C. Trappey and O. T. C. Shen, ”Un système universel de détection de panneaux de signalisation...,” *Advanced Engineering Informatics*, 2024. DOI: 10.1016/j.aei.2024.102674.

[39] J. Wang, L. Zhao, and M. Chen, ”Transfer learning for regional traffic sign datasets,” *Applied Sciences*, vol. 13, no. 5, 2023.

[40] Y. Bai et al., ”Amélioration de la compréhension des nuages de points...,” *Knowledge-Based Systems*, 2024. DOI: 10.1016/j.knosys.2024.112715.