# Traffic Sign Recognition system using CNN

Jeevitha.P[1], Thiyanadurai C[2], Ariharan V[3], Selvakumar K[4], Vijayaragavan S[5]
[1]*Assistant Professor, Department of Computer Science and Engineering, Maha Barathi Engineering College (Affiliated to Anna University), Chinnasalem (Tk), Kallakurichi (Dt)-606 201.*
[2,3,4,5]*UG Student, Department of Computer Science and Engineering, Maha Barathi Engineering College (Affiliated to Anna University), Chinnasalem (Tk), Kallakurichi (Dt)-606 201.*

***Abstract:*** **A dynamic component of intelligent transportation systems, traffic sign recognition is essential to sophisticated driver-assistance systems and driverless cars.This study proposes a CNN-based TSR system that integrates preprocessing, segmentation, feature extraction, and classification to accurately detect and recognize traffic signs in real-world environments. In the preprocessing phase, techniques such as Gaussian filtering for noise reduction, histogram equalization for contrast enhancement, and grayscale conversion are applied to improve image quality. The segmentation process utilizes Otsu's thresholding to isolate traffic signs from the background. For feature extraction a Gray Level Co-occurrence Matrix, deep learning-based methods using convolutional layers capture essential features like shape, texture, and color patterns. Finally, in the classification stage, a Convolutional Neural Network to accurately categorize traffic signs. To evaluate system performance, various performance metrics such as accuracy 95.5%, precision 94.8%, recall 95.9%, F1-score 96.2%, and Specificity 96.8% are utilized. The combination of image processing and deep learning techniques enhances the system's efficiency and reliability, making it suitable for real-time traffic sign recognition in autonomous vehicles and smart traffic management systems.**

***Keywords:*** **Traffic Sign Recognition, Gaussian filtering, Gray Level Co-occurrence Matrix, Convolutional Neural Network, Otsu's thresholding.**

## I.INTRODUCTION

Since artificial intelligence has become more prevalent in recent years, the vehicle-aided driving system has upgraded its earlier driving mode. The system prevents car accidents caused by driver weariness by rapidly reminding drivers to make accurate operations based on real-time road condition information. [1]. A traffic sign recognition system that can identify and comprehend the various traffic signs and assist the vehicles in which they are installed in making decisions and following the laws can be created using neural networks [2]. Therefore, it is challenging to identify a single procedure that generalizes an accurate detection, and developing a trustworthy real-time TSR continues to be an intimidating task. The identification and comprehension of restrictive indications will be the main goals of this study. The lack of such databases, photos, and innovative techniques for traffic sign recognition serves as our inspiration [3].

To address this issue, they first train cooperatively for accurate recognition models using privacy-preserving federated learning without sharing raw traffic sign data. [4]. However, because of the restricted energy and computational resources of the majority of gadgets, making it challenging for cars [5].

*A. Objectives*

To develop a system accomplished of accurately detecting and recognizing various traffic signs from images captured in real-world environments, ensuring reliable performance under varying conditions.

To utilize segmentation methods like Otsu's thresholding to isolate traffic signs from the background, enabling the system to focus on relevant objects and reduce the impact of unnecessary noise.

## II.RELATED WORK

To convey information to vehicles, traffic signs are crucial. Therefore, understanding traffic signs is crucial for road safety, and not doing so could lead to incidents. For the past few decades, traffic sign detection has been a focus of study. A reliable traffic sign detecting system is still a ways off, but the first steps are real-time and precise detections [6]. The traffic sign finding system is taught to identify and interpret a range of traffic indicators, including yield signs, and pedestrian crossings. [7]. The suggested technique extracts traffic sign information from vehicle panorama photos using a transformer encoder-decoder architecture in conjunction with convolutional neural networks and a semi-supervised learning approach [8]. Furthermore, a Swin-Transformer-based Cross-Stage Partial module is intended to improve the identification accuracy of small-scale traffic signs by capturing contextual information around them. [9]. after removing the illumination factors in the first phase using color-based histogram equalizations [10]. According to experimental data, the authors' approach considerably improves the detection system's overall performance by achieving notable gains of 3.2% on the mAP50 metric when compared to YOLOv8s while retaining a high detection speed [11]. This novel aims to methodically examine the YOLO object identification algorithm from five pertinent perspectives of this technology when it comes to traffic sign detection and recognition systems. [12]. additionally, the backbone network of YOLOv5 is replaced with GhostDarkNet53, which is based on the Ghost module, enhancing the model's real-time performance [13]. The suggested approach increases inference efficiency by 11.2% when compared to the original YOLO, according to experimental findings on the GTSDB dataset. [14The modified LeNet-5 obtained accuracy rates of 99.12% and 99.78%, respectively [15].

## III.PROPOSED METHODOLOGY

The methodology follows a structured pipeline that includes preprocessing, segmentation, feature extraction, and classification. These images are then passed through a preprocessing phase where Gaussian filtering is applied to remove noise, histogram equalization is used to enhance contrast, and grayscale conversion simplifies the image for further processing. Once segmented, the system moves to the feature extraction stage, where two key methods are employed: Gray Level Co-occurrence Matrix and Convolutional Neural Networks.

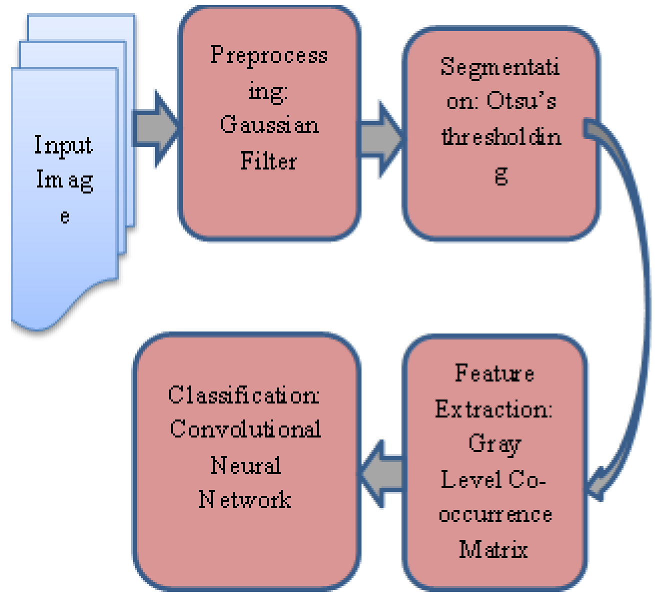


Fig.1 Proposed Architecture Diagram

### A. Pre-Processing: Gaussian Filtering

The following equation defines the Gaussian kernel, which is represented by $g_Q$, as a continuous function:

$$g_q[\mathrm{m}, \mathrm{n}] = \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2}\left(\frac{m^2+n^2}{\sigma^2}\right)}, \qquad (1)$$

Where the standard deviation is denoted by σ. It determines the weights with which pixels influence the filtering outcome by convolving the picture with a Gaussian kernel.

Basic Optimizations

According to a three-sigma rule of thumb, K should approximately equal 2πσ ≈ 6σ if the mask is K × K in size. Approximately 99.73% of the Gaussian function's non-zero values would be captured by it. Convolution will then take on the following shape.

$$y[i,j] = \sum_{m=-M}^{M} \sum_{n=-M}^{M} g_\sigma [m,n] x[i-m, j-n], \qquad (2)$$

Where K = 2M + 1, x [i, j] represents the value of the (i, j)-th pixel of image x, and y [i, j] represents the value of the (i, j)-th pixel of output image y. Figure 2 below shows an example of how discrete convolution operates.

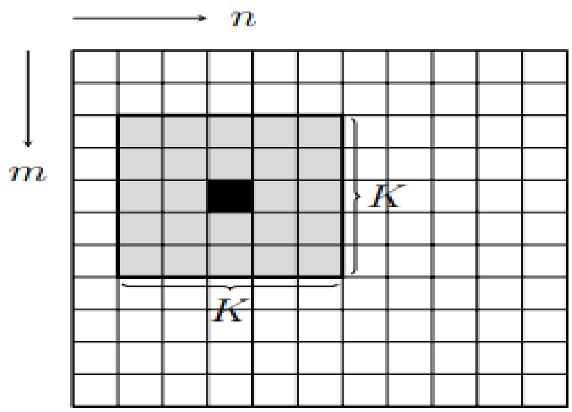


Fig. 2 Convolution of an image with the kernel of size 5×5

The values of the output picture y, or the convolution result, are obtained by applying this process to each image pixel x. There are two common ways to address such a border problem.

*B. Segmentation: Otsu's thresholding*

In a Traffic Sign Recognition (TSR) system, segmentation is a crucial step for accurately isolating traffic signs from complex backgrounds. Otsu's thresholding is applied to convert the grayscale image into a binary image, allowing clear differentiation between the traffic sign and the background. After segmentation, morphological operations such as erosion, dilation, opening, and closing are employed to refine the extracted sign regions and remove unwanted noise.

It helps refine the shape by shrinking the boundaries of the detected region. The erosion function is defined as:

$$E[I(i,j)] = Min\{I(x+y) * y \in s\} \qquad (3)$$

Where **s** represents the structural element used for transformation. This process removes isolated noise pixels, ensuring only significant traffic sign regions remain.

Dilation is applied to expand the detected traffic sign region, reinforcing weak edges and restoring missing parts of the sign structure.

The function is defined as:

$$D[I(x,y)] = Max\{I(x-y) * y \in s\} \qquad (4)$$

This operation is crucial for enhancing object dependencies, ensuring the traffic sign structure remains intact while suppressing minor background irregularities.

Morphological opening is performed to remove small bright noise pixels while preserving the overall shape of the traffic sign. It is represented as:

$$Mor_{ope[\,I\,(x,y)]} = E\,[I\,(x,y)] \qquad (5)$$

This final segmentation step isolates the traffic sign region, ensuring precise detection before feature extraction and classification using a Convolutional Neural Network**.**

*C. Feature Extraction: Gray Level Co-occurrence Matrix*

The GLCM provides mathematical texture-based features that enhance the recognition of traffic signs. GLCM computes the spatial relationship between pixel intensities in an image, capturing essential textural patterns that distinguish different traffic signs.

*A. Mean*

The mean represents the average intensity value across the entire traffic sign image. It provides an overall brightness measure.

$$\mu = \frac{\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)}{NXN} \qquad (6)$$

Whereas,

$\mu - Mean\ intensity,$ $f(x,y) -$ $Pixel\ intensity\ at\ position,$ $N -$ $Total\ number\ of\ pixels.$

*B. Variance*

Variance measures the intensity variation within the traffic sign, which helps in identifying different colors and patterns. A high variance indicates strong texture variations, such as those found in warning or prohibition signs.

$$Variance(V) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1}(x-\mu)^2 f(x,y) \qquad (7)$$

*C.Standard Deviation*

Standard deviation represents the dispersion of pixel intensities in the traffic sign image. This feature is useful in detecting sharp edges and contrasting regions in traffic signs. It is calculated as:

$$Standard\ Diviation(SD) = \sqrt{\frac{\sum |x-\mu|^2}{N}} \qquad (8)$$

*D.Energy*

Energy measures texture strength in a sign image. Traffic signs with clear patterns and well-defined regions exhibit high energy values.

$$Energy = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} |C|^2 \quad (9)$$

*E. Dissimilarity*

Dissimilarity measures the degree of variation in pixel intensity. High dissimilarity values indicate highly detailed traffic signs with multiple elements.

$$D = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} |x - y|^2$$
(10)

*D.Classification: Convolutional Neural Network (CNN)*

CNN has dominated the field of machine vision. Typical CNN hidden layer components include normalizing, pooling, convolution, and complete connectivity layers.

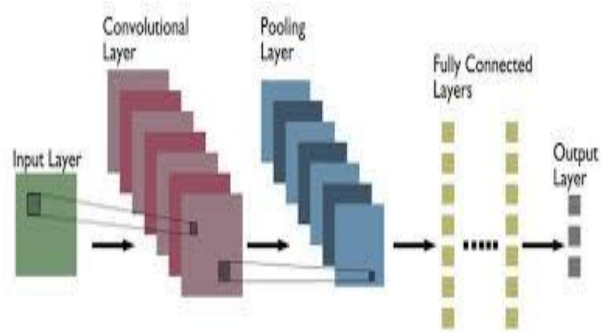


Fig. 3 Convolutional Neural Network Architecture

Matrix vectors multiplied form the basis of data and weight representation. Each layer has a unique set of properties for a group of pictures. Early CNN layers, for instance, will pick up basic characteristics like boundaries, bright spots, dark areas, shapes, etc.

The CNN learning process may make use of vector calculus. Assume $y$ is z ∈ R if z is a scalar and R H is a vector. A vector is defined as follows: the partial derivative of z with respect to y is a vector if z is a function of $y$.

$$(\partial z \ \partial yi) = \left(\frac{\partial z}{\partial yi}\right) \quad (11)$$

Convolutional, pooling, fully connected, and normalizing layers are the layers that make up CNNs. Several layers are added to these properties in order to capture complex data representations.

## IV.RESULT & DISCUSSION

The proposed CNN-based Traffic Sign Recognition system was evaluated using benchmark datasets, demonstrating high accuracy and robustness in diverse conditions. The model achieved a high accuracy rate, indicating its strong ability to classify traffic signs correctly. The F1-score confirmed a balanced performance between precision and recall, while specificity highlighted the system's effectiveness in distinguishing traffic signs from non-sign elements.

*A. Performance Metrics*

Table 1. Performance Metrics

| S.No | Metrics | Formulas |
|---|---|---|
| 1 | Accuracy | $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$ |
| 2 | Precision | $Precision = \frac{TP}{TP+FP}$ |
| 3 | Recall | $Recall = \frac{TP}{TP+FN}$ |
| 4 | Specificity | $Specificity = \frac{TN}{TN+FP}$ |
| 5 | FI-Score | $FI - Score = 2 \times \frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$ |

*B. Dataset Description*

The dataset used to categorize the various traffic sign classifications is available here. There are around 120 photos in each of the 58 classes. The corresponding description of the traffic sign class is contained in the labels.csv file. The assignment of these class IDs with descriptions is modifiable. To obtain respectable Val accuracy, we may utilize the standard CNN model.

*C. Comparison Results*

Table.2 Evaluation of Performance Metrics

| Performance Metrics | Accuracy | Precision | Recall | FI-Score | Specificity |
|---|---|---|---|---|---|
| Proposed CNN | 95.5% | 94.8% | 95.9% | 96.2% | 96.8% |
| Support Vector Machine | 94.3% | 93.2% | 93.0% | 93.7% | 94.1% |
| k-Nearest Neighbors | 92.8% | 91.9% | 92.5% | 92.2% | 93.6% |
| Random Forest | 90.5% | 89.8% | 90.2% | 90.0% | 91.3% |
| Decision Tree | 88.9% | 88.1% | 88.5% | 88.3% | 89.7% |

Table 2. Compares the performance of five machine learning models based on various metrics. The Proposed CNN outperforms the others with the highest accuracy (95.5%), precision (94.8%), recall (95.9%), F1-score (96.2%), and specificity (96.8%), demonstrating a balanced and effective classification capability.

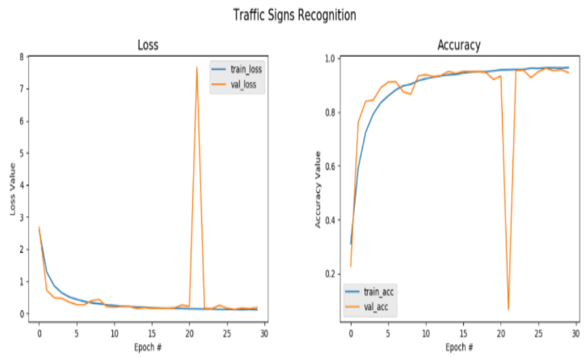


Fig. 4 Evaluation of accuracy for Proposed CNN, SVM, KNN, RF, DT

Figure 4 illustrates the evaluation of accuracy across five different machine learning models: Proposed CNN, SVM, and KNN, RF, and DT. The Proposed CNN achieves the highest accuracy at 95.5%, indicating its superior performance in correctly classifying instances. The SVM follows with a slightly lower accuracy of 94.3%, still demonstrating strong performance. KNN shows an accuracy of 92.8%, while Random Forest and Decision Tree have lower accuracy values of 90.5% and 88.9%, respectively.

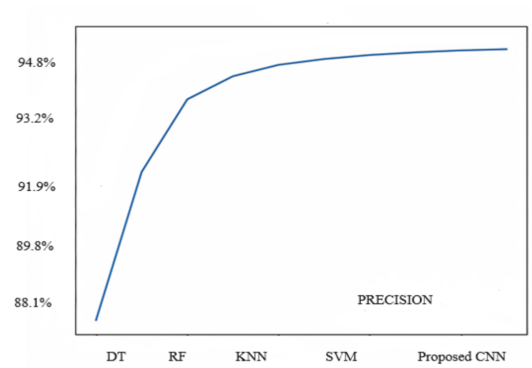


Fig. 5 Evaluation of Precision for Proposed CNN, SVM, KNN, RF, DT

Figure 5 illustrates the evaluation of precision for several machine learning models, including CNN, SVM, KNN, RF, and DT. The precision scores for these models are presented as percentages, with CNN achieving the highest precision at 94.8%, followed by SVM at 93.2%, KNN at 91.9%, RF at 89.8%, and DT at 88.1%. These results demonstrate that CNN outperforms the other models in terms of precision, indicating its superior ability to correctly identify positive instances among the predicted positives.

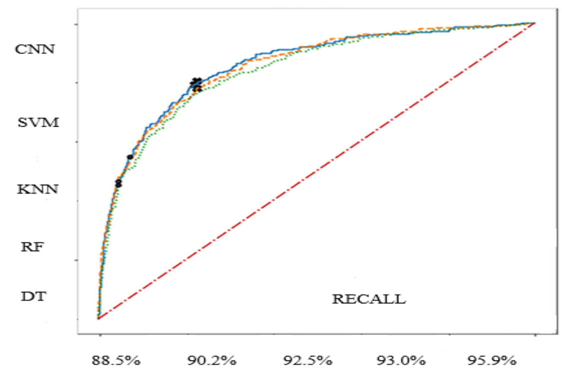


Fig. 6 Evaluation of Recall for Proposed CNN, SVM, KNN, RF, DT

Figure 6 presents the evaluation of recall for the proposed models, including Convolutional Neural Networks, SVM, KNN, RF, and DT. The recall scores are reported as percentages, with CNN achieving the highest recall at 95.9%, followed by SVM at 93.0%, KNN at 92.5%, RF at 90.2%, and DT at 88.5%. The results show that CNN excels in this regard, accurately identifying a higher proportion of relevant instances compared to the other models

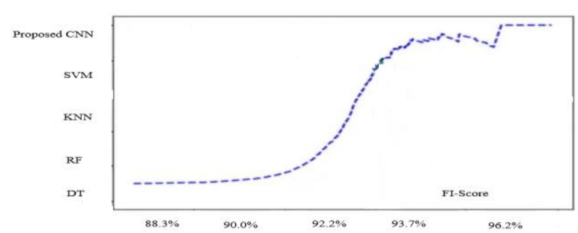


Fig. 7 Evaluation of FI-Score for Proposed CNN, SVM, KNN, RF, DT

Figure 7 showcases the evaluation of the F1-Score for the proposed models, which include Convolutional Neural Networks, SVM, KNN, RF, and DT. In this evaluation, CNN achieves the highest F1-Score at 96.2%, followed by SVM at 93.7%, KNN at 92.2%, RF at 90.0%, and DT at 88.3%.

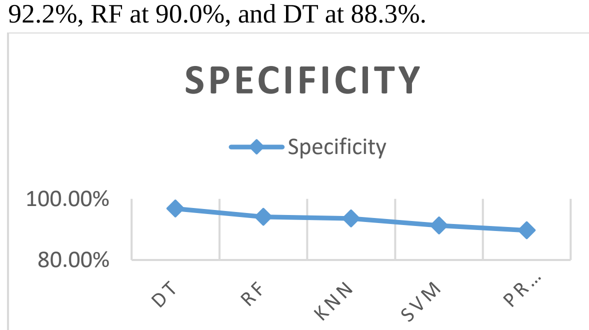


Figure 8. Evaluation of Specificity for Proposed CNN, SVM, KNN, RF, DT

Figure 8 displays the evaluation of specificity for the proposed models, including Convolutional Neural Networks, SVM, KNN, RF, and DT. CNN leads with the highest specificity at 96.8%, followed by SVM at 94.1%, KNN at 93.6%, RF at 91.3%, and DT at 89.7%

## V.CONCLUSION

In conclusion, the proposed CNN-based Traffic Sign Recognition system effectively integrates preprocessing, segmentation, feature extraction, and classification to accurately detect and recognize traffic signs in real-world environments. Through techniques like Gaussian filtering, histogram equalization, and Otsu's thresholding, the system enhances image quality and isolates relevant features. The DL-based feature extraction, utilizing CNN, captures critical attributes such as shape, texture, and color patterns for precise classification. With impressive performance metrics accuracy of 95.5%, precision of 94.8%, recall of 95.9%, F1-score of 96.2%, and specificity of 96.8% the system demonstrates high reliability and efficiency.

## REFERENCES


[1] Zhu, Y., & Yan, W. Q. (2022). Traffic sign recognition based on deep learning. Multimedia Tools and Applications, 81(13), 17779-17791.

[2] Singh, Karan, and Nikita Malik. "CNN based approach for traffic sign recognition system." Advanced Journal of Graduate Research 11, no. 1 (2022): 23-33.

[3] Dewi, Christine, Rung-Ching Chen, Yan-Ting Liu, and Shao-Kuo Tai. "Synthetic Data generation using DCGAN for improved traffic sign recognition." Neural Computing and Applications 34, no. 24 (2022): 21465-21480.

[4] Xie, Kan, Zhe Zhang, Bo Li, Jiawen Kang, Dusit Niyato, Shengli Xie, and Yi Wu. "Efficient federated learning with spike neural networks for traffic sign recognition." IEEE Transactions on Vehicular Technology 71, no. 9 (2022): 9980-9992.

[5] Megalingam, Rajesh Kannan, Kondareddy Thanigundala, Sreevatsava Reddy Musani, Hemanth Nidamanuru, and Lokesh Gadde. "Indian traffic sign detection and recognition using deep learning." International journal of transportation science and technology 12, no. 3 (2023): 683-699.

[6] Manawadu, Mayura, and Udaya Wijenayake. "Voice-assisted real-time traffic sign recognition system using convolutional neural network." arXiv preprint arXiv: 2404.07807 (2024).

[7] Khalifa, Alaa A., Walaa M. Alayed, Hesham M. Elbadawy, and Rowayda A. Sadek. "Real-Time Navigation Roads: Lightweight and Efficient Convolutional Neural Network for Arabic Traffic Sign Recognition in Intelligent Transportation System." Applied Sciences 14, no. 9 (2024): 3903.

[8] Chen, Siyun, Zhenxin Zhang, Liqiang Zhang, Rixing He, Zhen Li, Mengbing Xu, and Hao Ma. "A semi-supervised learning framework combining cnn and multi-scale transformer for traffic sign detection and recognition." IEEE Internet of Things Journal (2024).

[9] Qian, Yue Jing, and Bo Wang. "TSDet: A new method for traffic sign detection based on YOLOv5-SwinT." IET Image Processing 18, no. 4 (2024): 875-885.

[10] Suresha, R., and N. Manohar. "Two-Stage Traffic Sign Classification System." Procedia Computer Science 235 (2024): 2703-2715.

[11] Du, Songjie, Weiguo Pan, Nuoya Li, Songyin Dai, Bingxin Xu, Hongzhe Liu, Cheng Xu, and Xuewei Li. "TSD-YOLO: Small traffic sign detection based on improved YOLO v8." IET Image Processing 18, no. 11 (2024): 2884-2898.

[12] Flores-Calero, Marco, César A. Astudillo, Diego Guevara, Jessica Maza, Bryan S. Lita, Bryan Defaz, Juan S. Ante, David Zabala-Blanco, and José María Armingol Moreno. "Traffic sign detection and recognition using YOLO object detection algorithm: A systematic review." Mathematics 12, no. 2 (2024): 297.

[13] Sun, Xiaoqiang, Kuankuan Liu, Long Chen, Yingfeng Cai, and Hai Wang. "LLTH-YOLOv5: a real-time traffic sign detection algorithm for low-light scenes." Automotive Innovation 7, no. 1 (2024): 121-137.

[14] Zhang, Jingyu, Wenqing Zhang, Chaoyi Tan, Xiangtian Li, and Qianyi Sun. "YOLO-PPA based efficient traffic sign detection for cruise control in autonomous driving." In Proceedings of

the 2024 International Conference on Industrial Automation and Robotics, pp. 8-13. 2024.

[15] Kheder, Mohammed Qader, and Aree Ali Mohammed. "Real-time traffic monitoring system using IoT-aided robotics and deep learning techniques." Kuwait Journal of Science 51, no. 1 (2024): 100153.