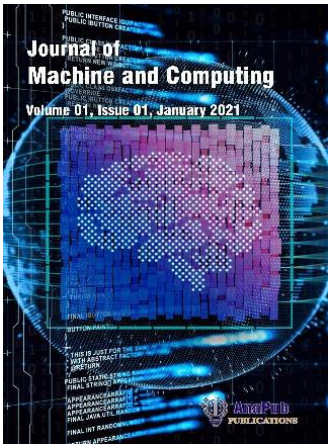
Journal Pre-proof

Efficient and Accurate Traffic Sign Detection Leveraging YOLOv8: A Cutting-Edge Deep Learning Framework



Gunji Sreenivasulu, Lakshmi H N, Muni Kumari T, Anjaiah P, Suresh A and Avanija J

DOI: 10.53759/7669/jmc202505001

Reference: JMC202505001

Journal: Journal of Machine and Computing.

Received 10 April 2024

Revised form 12 June 2024

Accepted 10 September 2024

Please cite this article as: Gunji Sreenivasulu, Lakshmi H N, Muni Kumari T, Anjaiah P, Suresh A and Avanija J, “Efficient and Accurate Traffic Sign Detection Leveraging YOLOv8: A Cutting-Edge Deep Learning Framework”, Journal of Machine and Computing. (2025). Doi: https:// doi.org/10.53759/7669/jmc202505001

AnaPub
PUBLICATIONS

Efficient and Accurate Traffic Sign Detection Leveraging YOLOv8: A Cutting-Edge Deep Learning Framework

[1]Gunji Sreenivasulu, [2]Lakshmi H N, [3]T. Muni Kumari, [4]P. Anjaiah, [5]A. Suresh, [6]J. Avanija

[1]Assistant Professor, Department of Computer Science & Engineering, Madanapalle Institute of Technology & Science, Madanapalle, Annamayya (Dt), AP, India.
[2]Professor, Department of Computer Science and Engineering (AIML), CVR College of Engineering, Hyderabad.
[3]Assistant professor, Department of MCA, Annamacharya Institute of Technology and Sciences, Tirupati.
[4]Assistant Professor, Department of Artificial intelligence and machine learning, Mallareddy University, Hyderabad.
[5]Associate Professor, Department of Computer Science and Engineering, Siddharth Institute of Engineering & Technology, Puttur.
[6]Professor, School of Computing, Mohan Babu University, Tirupati, Andhra Pradesh, India.
[1]gsr1251@gmail.com, [2]hn.lakshmi@cvr.ac.in, [3]kumari.muni7@gmail.com
[4]anjaiah.pole@gmail.com, [5]csesuresh6@gmail.com, [6]avans75@yahoo.co.in

Abstract - The timely and precise identification of traffic signs is essential for maintaining the effectiveness and safety of contemporary roads, particularly in light of the increasing number of self-driving cars. Conventional image processing methods have faced challenges because to the intricate and fluctuating variables present in real-world settings, including various signage, erratic weather, and inconsistent illumination. This study utilizes recent breakthroughs in deep learning, particularly the YOLOv8 (You Only Look Once version 8) model, to tackle these difficulties. YOLOv8 incorporates cutting-edge neural network architectural advancements, such as an anchor-free detection methodology, adaptive spatial feature pooling, and dynamic neural configurations. In order to further increase detection efficiency and accuracy, this study presents two innovative models, YOLOv8-DH and YOLOv8-TDHSA. These models make use of improvements such decoupled heads and transformer-based self-attention mechanisms. Experimental results indicate that the suggested models substantially surpass current deep learning models, attaining enhanced performance across multiple measures, including accuracy, recall, F-score, and mean average precision (mAP). This research enhances traffic sign detecting technology, facilitating the development of safer and more intelligent transportation systems.

Keywords – Object detection, Traffic sign, YOLO, Image processing, Computer vision, Attention mechanism

I. INTRODUCTION

The quick identification of traffic signs is crucial for both human drivers and autonomous vehicles to navigate current streets. Modern intelligent transportation systems depend much on traffic sign detection to improve road safety, efficiency, and general traffic management [1]. Traffic signs guarantee the safety and effective movement of cars on the road since they help to guide, warn, and control drivers. The capacity to accurately and efficiently identify traffic indicators has gotten progressively vital as a result of the swift advancement of autonomous driving technologies. However, this endeavor presents multiple challenges, including the existence of diverse sign types, colors, and sizes, the presence of unpredictable weather conditions, and the irregularity of lighting conditions.

Because traditional image processing techniques rely on handcrafted characteristics and are sensitive to environmental changes, they have not been very successful in handling these complexity. However, deep learning has completely changed the industry by making it possible to quickly, accurately, and scalable detect things in real-time circumstances. This is especially true with the introduction of object recognition algorithms like YOLO (You Only Look Once). Methods for target identification, such as Faster RCNN, YOLO, and FCOS, have gained widespread utilization in traffic sign detection owing to the swift advancement of deep learning technology [2]. The most recent addition of the YOLO family, YOLOv8, incorporates state-of-the-art developments in neural network architecture, including enhanced backbone networks and dynamic neural topologies that change depending on the intricacy of the input data. It offers a number of improvements, such as better design, quicker inference times, and increased precision, which makes it the best option for traffic sign identification.

In the end, utilizing cutting-edge deep learning models for traffic sign identification, such as YOLOv8, is a step toward a future that is safer, more effective, and technologically integrated. These developments will have a significant impact on

how cities and transportation systems develop, influencing how people move about and interact in our increasingly linked world. The following are our suggested methodology's main contributions:

- We introduced two novel models, YOLOv8-DH and YOLOv8-TDHSA, designed for precise traffic sign identification.
- We implemented a detached head design to enhance object categorization and localization operations.
- This research study incorporates a small object detection layer to improve the identification of small and distant traffic signs.
- To improve detection robustness, we added transformer-based self-attention methods to the YOLOv8 framework.
- We carried out extensive tests showing our suggested models' enhanced performance in various traffic situations.

II. LITERATURE REVIEW

Since traffic signs are essential for maintaining road safety and enabling autonomous driving, traffic sign identification has drawn a lot of interest in the fields of computer vision and intelligent transportation systems research. The majority of early traffic sign detection efforts used conventional image processing methods including edge detection, color segmentation, and template matching [3].

In order to overcome these restrictions, machine learning approaches were investigated, including Support Vector Machines (SVMs) and Decision Trees, which provide superior generalization by means of feature extraction techniques such as histograms of oriented gradients. For example, Jo Young-Bae et al. [4] presented a technique based on edge recognition and color segmentation, which performed satisfactorily in controlled settings but failed in many real-world scenarios, including dim light, occlusions, and deformations. Similarly, For traffic sign detection, Maldonado-Bascón et al. [5] employed features taken from histograms of oriented gradients (HOG) to train a support vector machine (SVM) classifier for traffic sign detection. Even with improved generality, these approaches remained constrained by their reliance on feature engineering and their incapacity to adjust to a wide range of intricate and varied real-world situations.

Traffic sign detection underwent a revolution with the advent of deep learning, specifically convolutional neural networks (CNNs). This was made possible by the ability to automatically extract features from large datasets, which greatly improved detection performance, as demonstrated by J. Stallkamp et al. [6] and Shijin Song et al. [7]. By using deep CNNs to automatically learn features directly from data, Pierre Sermanet et al. [8] showed off their potential for traffic sign recognition, outperforming more conventional methods in terms of accuracy. Their research served as a basis for later deep learning models, which greatly enhanced the performance of detection and recognition. Dan Cireşan et al. [9] demonstrated one of the first effective uses of deep learning for traffic sign recognition, utilizing a multi-column deep neural network (MCDNN) to attain cutting-edge outcomes on the GTSRB dataset. Through their work, they were able to significantly outperform traditional methods in learning robust features directly from raw pixel data using deep learning models. Fast R-CNN [10] and Faster R-CNN [11] incorporated quicker region proposal algorithms and end-to--end training to address this, but they still unable achieve the required speed for real-time traffic sign detection [26, 27].

However, the YOLO series of models, pioneered by Redmon et al. [12], achieved a notable advancement in real-time object detection including traffic sign detection by conceptualizing it as a regression problem. YOLO's unique design allowed the prediction of bounding boxes and class probabilities concurrently in a single forward pass, hence combining great speed and accuracy. Successive versions of YOLO, such as YOLOv2 [13], YOLOv3 [14], and YOLOv4 [15], further extended the usefulness of YOLO models to encompass a broader variety of objects and settings, rendering them highly suitable for detecting traffic signs in real-time. Newer iterations of the YOLO series, like YOLOv5, YOLOv6, and YOLOv7, have improved performance even more with novelties like auto-learning bounding boxes, mosaic data augmentation, and sophisticated activation functions. For example, Junfan Wang et al. used YOLOv5, which combines an enhanced data augmentation strategy with a scalable architecture, leading to better traffic sign detection performance on a variety of datasets [16]. Using YOLOv6 in conjunction with an upgraded Logistic Regression classifier, Ravinder Kaur et al. [17] sought to improve traffic sign identification in real-time. For the purpose of precisely identifying traffic signs, Songjiang Li et al. [18] utilized YOLOv7 due to its enhanced ability to detect small objects.

The most recent iteration of the YOLO series, YOLOv8 offers a number of architectural improvements that make it especially suitable for traffic sign recognition. In order to provide more flexible and accurate traffic sign detection under a variety of conditions, it introduces an anchor-free detection strategy, an adaptive spatial feature pooling backbone network, and more. Because YOLOv8's anchor-free architecture eliminates computational overhead and streamlines the detection process, it operates faster and more effectively in real-time applications.

III. MATERIALS AND METHODS

3.1 Dataset Description

The dataset has 4696 instances of photos depicting traffic signs. It is divided into three parts: training, testing, and validation. The training set consists of 71% of the total data, whereas the testing set contains 13% and the validation set has 16%. The dataset consists of fifteen unique classes, which encompass several types of traffic lights ('Green Light', 'Red Light'), speed limits ('Speed Limit 10', 'Speed Limit 100', 'Speed Limit 110', 'Speed Limit 120', 'Speed Limit 20', 'Speed Limit 30', 'Speed Limit 40', 'Speed Limit 50', 'Speed Limit 60', 'Speed Limit 70', 'Speed Limit 80', 'Speed Limit 90'), and a "Stop" sign. It is ideal for a number of uses, including autonomous car navigation, where precise traffic sign identification is necessary for making decisions in real time and adhering to traffic laws. Its ability to provide a representative and varied sample of actual traffic signs makes it useful for undertaking thorough road network analyses, optimizing smart city infrastructure, and improving road safety training programs. This varied set of criteria offers a complete foundation for the creation and evaluation of consistent traffic sign detection systems.

Figure 1: Sample of the dataset

3.2 Preprocessing Video Data for Detection

Several important actions are taken during preprocessing video data for traffic sign detection to equip the data for model training or inference. First, extract individual frames I_t from the video V with T frames by sampling every k-th frame, resulting in frames $I_{t\cdot k}$. Each extracted frame is then resized to a uniform dimension (W,H) using a resizing function R, so that:

$$I'_{t\cdot k} = R(I_{t\cdot k}; W, H) \qquad (1)$$

After resizing, normalize the pixel values of the frame by dividing them by 255, transforming the frame $I'_{t\cdot k}$ to $I''_{t\cdot k}$, where:

$$I''_{t\cdot k}(x,y) = \frac{I'_{t\cdot k}(x,y)}{255} \quad (2)$$

If necessary, convert the frame to grayscale using a color space conversion function, such as:

$$I'''_{t\cdot k} = \text{rgb2gray}(I''_{t\cdot k}) \quad (3)$$

For data augmentation, apply transformations like rotation, flipping, and scaling. For instance, a rotation by angle θ results in:

$$I_{\text{aug}} = T_{\text{rot}}(I''_{t\cdot k}; \theta) \quad (4)$$

Horizontal flipping gives:

$$I_{\text{flip}}(x,y) = I''_{t\cdot k}(W - x, y) \quad (5)$$

Adjust bounding box coordinates (x_{min}, y_{min}, x_{max}, y_{max}) according to the resized dimensions using scaling factors for width and height. For frames requiring temporal data, prepare sequences of frames, such as:

$$\{I''_{t\cdot k}, I''_{(t+1)\cdot k}, \ldots, I''_{(t+n-1)\cdot k}\} \quad (6)$$

where n is the sequence length.

Optionally, compute optical flow F between frames to track motion, where:

$$F(x,y) = \text{OpticalFlow}(I_{t\cdot k}, I_{(t+1)\cdot k}) \quad (7)$$

Finally, split the processed frames into training, validation, and test sets to ensure a balanced and effective dataset for model evaluation.

3.3. Proposed Model

The YOLOv8s improvements employed by the models put forward in this paper are described in this section. Both of the suggested models, YOLOv8-DH and YOLOv8-TDHSA, use the decoupled head (DH) enhancement, but only YOLOv8-TDHSA uses the other two improvements. In June 2017, the Google team suggested the transformer concept [19]. It has demonstrated significant potential in image processing. The transformer enables the model to parallelize training and fully leverage the global information of training data. RNNs, a sequential architecture is eliminated and employs a self-attention mechanism instead. The self-attention shown in Figure 2 is the fundamental calculation of the transformer model. The regular attention mechanism determines the weighted average of the input data based on the attention distribution by first calculating the attention distribution on all of the input data. The input features are mapped by self-attention to three new representational spaces: Query (X), Key (Y), and Value (Z). After calculating the correlation among X and Y, the values are normalized, and the distance between them is increased to improve attention using the SoftMax function. The attention value is calculated by adding the weight coefficient and V. By mapping the characteristics to three spatial representations, the self-attention mechanism helps to avoid issues that arise when a single space is assigned to all features. For instance, there won't be a distinction between the correlation of X2 and X1 and X1 and X2 and X2, if X1 and X2 are utilized directly to compute the correlation. In this scenario, the attention mechanism's expression capacity will deteriorate. The changes between X1 and Y2 and X2 and Y1 can also be reflected if K is added to the original data correlation calculation. This can improve the attention mechanism's expression capacity.

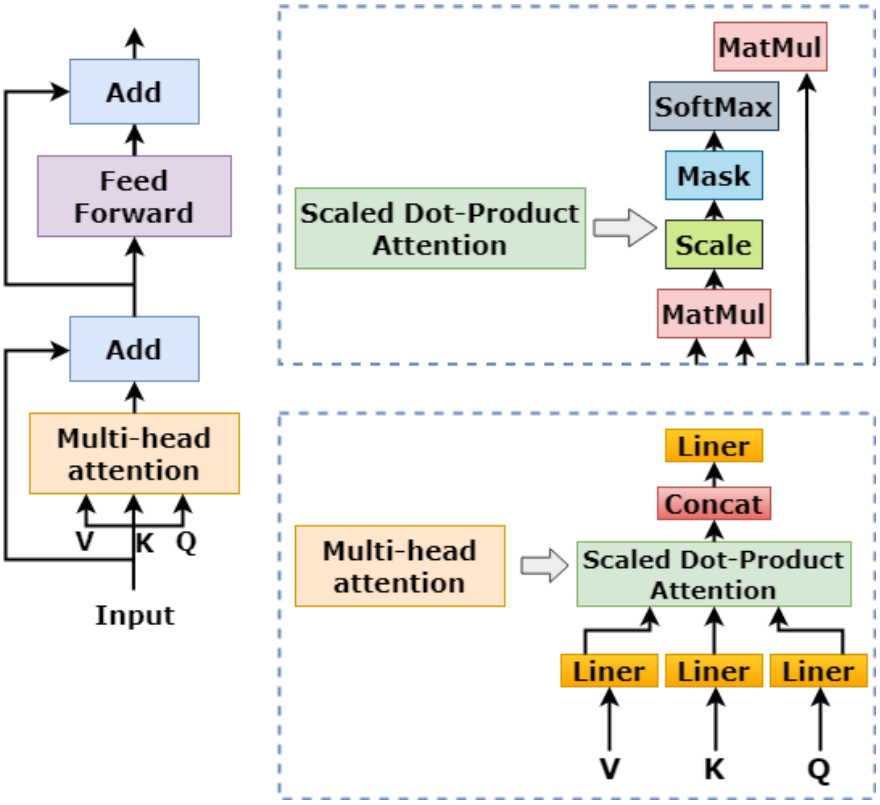


Figure 2: Architecture of transformer based attention-mechanism

It can't be suitable to employ X or Y because the attention weight obtained is the input of the next phase; consequently, the third space, Z, is introduced [20]. Using a weighted total, finally, the attention value is gained. The transformer model would, however, demand much more computation, that may increase the training time. When the image components move to the final network layer, the feature dimension is at its smallest. At this point, adding the transformer would have the least impact on the model's training. Consequently, the transformer is only included to the last layer of the neck of the proposed YOLOv8-TDHSA model and substitutes for the CBS at the final of the original YOLOv8 models. On the other hand, the transformer model would demand a significant rise in processing capability, hence increasing the training expenses. Moving the image features to the final layer of the network results in the feature dimension at its minimum. Right present, addition of the transformer would have the least impact on model training. Thus, in addition to substituting the CBS at the last layer of the original YOLOv8s model, the transformer in the suggested YOLOv8-TDHSA model is added to the last layer of its neck [21]. It is suggested that the decoupled head in object detection enhance YOLO's performance by isolating the duties of positioning and object recognition. Class prediction and bounding box localization are handled by different branches in the decoupled head, in contrast to the original YOLO head, which integrates both tasks into a single feature map. To minimize dimensionality, the feature map in the decoupled head is first processed through a 1×1 convolution layer, then two parallel 3×3 convolution layers. The number of categories in the dataset is reflected in the output dimension of one branch, which makes predictions about the object category. For instance, after using a Sigmoid activation function and a convolution operation, the channel dimension in the dataset with 16 categories increases to 16. Another branch that controls whether the object box is background or foreground reduces the channel dimension to one. By predicting the bounding box coordinates (x, y, w, h), a third branch lowers the channel dimension to 4. These outputs are combined to create a feature map with 20×50 dimensions.

The decoupling procedure reduces feature loss and improves accuracy despite the additional complexity, even if it first increases the number of parameters, which slows down training. Mathematically, the decoupled head operation may be expressed as follows if F is the feature map:

$$F_{output} = concat(F_{class}, F_{\frac{foreground}{background}}, F_{coordinates})$$

Where $F_{class} \in R^{20*20*45}$

$F_{\frac{foreground}{background}} \in R^{20*20*1}$

$F_{coordinates} \in R^{20*20*4}$ concatenated into $F_{output} \in R^{20*20*50}$

To improve object identification efficiency, the YOLOv8-DH model that has been suggested replaces the original YOLOv8s head with three decoupled heads: D1, D2, and D3. The architecture is shown in Figure 3.

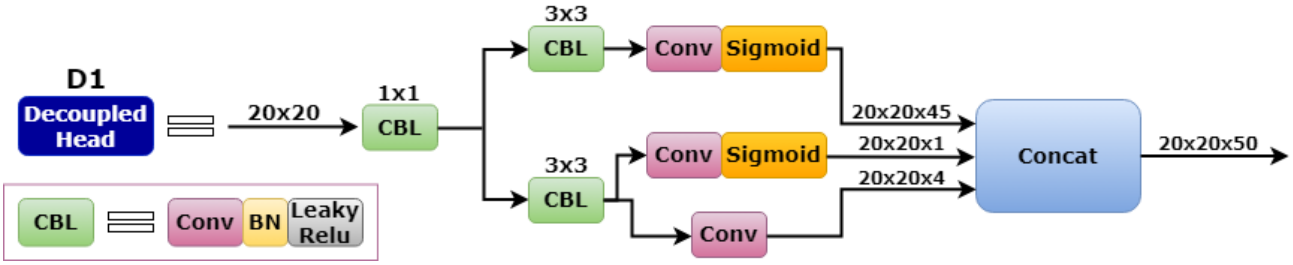


Figure 3: Architecture of head D1 used by the proposed model

The size of the traffic signs varies in the captured photos due to changes in distance between the shooting apparatus and the item, which affects the detection accuracy to some extent [16]. YOLOv8s fixes this by means of PANet. The initial model produced feature maps with dimensions of 80 x 80 x 255, 40 x 40 x 255, and 20 x 20 x 255 for an input image with a resolution of 640 × 640 pixels. These sizes indicate distinct feature extraction scales. Now, in that sequence, the produced detection box includes grid sizes of 8 × 8, 16 x 16, and 32 x 32 pixels. Still, when the collection comprises many objects smaller than 8 x 8 pixels, the small object detection than it, is not performing good. Additionally, the feature pyramid emphasizes the extraction and optimization of the essential elements. The loss of some top-level characteristics in the network will lower the accuracy of object detection. Another structure is added to the PANet of YOLOv8s to improve the detection of small objects in order to maintain the constant input image size. Stated differently, the feature map keeps expanding since layer 17 employs convolution and upsampling processes on it.

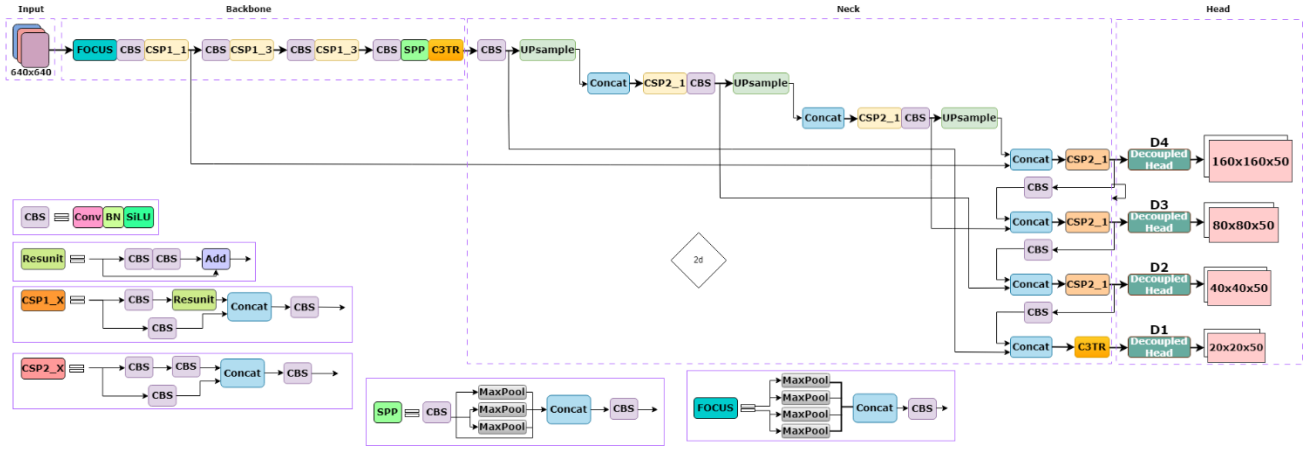


Figure 4: Proposed model architecture

In the backbone at layer 20, the 160 × 160-pixel feature map from layer 19 is combined with the feature map from layer 2, which helps to reduce feature loss during transmission between layers. A dedicated detection layer is added to the network to improve the detection of small objects. In order to make up for the loss of characteristics in the lower layers and increase detection accuracy, this layer mixes features from the higher and lower levels. A branch, denoted by a solid red line in Figure 6, links layer 2 to layer 19, resulting in the addition of a fourth output with dimensions of 160 × 160 × 128. Upon detaching the head, the feature size shrinks to 160 × 160 × 50. Following head decoupling, the smallest feature size is 160 × 160 × 50, enabling a minimum detection box size of 4 × 4 pixels, hence improving the network's capability to identify small objects. Figure 4 shows the suggested architecture for the model. Table 1 provides the parameters of the proposed model along with their values.

Table 1: Parameters and values of proposed model

Parameter	Values
Iteration	100
BS	68
ILR	0.0001
Dropout	0.25
Optimizer	Adam
Weight decay WD	0.0005
Iou threshold	0.5
Scheduler	CosineAnnealingLR

Here:

- Iteration = epoch
- BS = batch size
- ILR = initial learning rate
- WR = Weight decay

IV. EXPERIMENTAL SETUP AND EVALUATION

4.1 Experimental Setup

The testing involved a Windows 10 PC running an Intel Core i7-10700 CPU and a GeForce RTX 3090 GPU with 24 GB of video memory. The DL framework utilized for model fitting was PyTorch 1.8.1; CUDA 11.1 was used to speed training. The input images were shrunk to 640 × 640 pixels to help processing. The learning rate for the trials was tuned to a beginning value of 0.01 and a final learning rate of 0. to guarantee gradual convergence throughout model training. Table 2 displays the comparison model's size and number of parameters.

Table 2: Computational complexity of all models

Model	Space information	Parameter details
Faster R-CNN	360	27.76
YOLOV5	22.4	1.76
DETECTRON2	29.45	17.6
YOLOV8-TDSHA	23.6	15.87

4.2 Evaluation Metrics

Precision: Precision measures how accurate the model's positive predictions are. It is computed using:

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

Recall: Recall estimates the model's capacity to properly identify all relevant examples, or, in this case, all genuine problems. It is calculated by:

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

F-score: The mean of precision and recall is the F-score (also known as the F1-score), which delivers a balanced assessment that accounts for both false positives and false negatives. This is computed by:

$$\text{F-score} = 2 * \frac{Precision*Recall}{Precision+Recall}$$

mAP: The mean AP value over all object classes is called the mean average precision, or mAP. It is computed as follows. It is calculated as:

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i$$

4.3 Outcomes of the models

Table 3: Precision, recall, f-score and mAP values of all models

Model	Precision	Recall	F-score	mAP
Faster R-CNN	78.89	77.43	78.30	79.4
YOLOV5	81.54	82.32	82.19	81.65
DETECTRON2	87.86	87.43	86.60	86.51
YOLOV8-TDSHA	97.52	98.76	98.49	99.03

We used precision, recall, F1-score, and mAP as our primary performance metrics and drew confidence curves for each to see how these metrics changed when we increased or decreased the confidence thresholds. These visual assessments offer insightful analyses of the model's behavior at various confidence levels, assisting us in identifying the model's advantages and possible shortcomings. Here, we explore these evaluations' outcomes in further detail and discuss their implications for model performance.

Table 3 offers a thorough comparison for four well-known deep learning models evaluated for traffic sign detection includes Faster R-CNN, YOLOv5, Detectron2, and YOLOv8-TDSHA. The results show that YOLOv8-TDSHA's exceptional efficiency and accuracy in traffic sign detection greatly exceed those of the other models across all measures.

Precision estimates among all the model's positive predictions the percentage of actual positive detections. A higher accuracy score denotes less false positives, so the model is more accurate in precisely spotting real traffic signs without misreading non-sign objects as signs. At 97.52%, YOLOv8-TDSHA boasts the best accuracy among all other models by a rather margin. With the lowest number of false positives indicated by this great accuracy, YOLOv8-TDSHA is the most dependable model for exact traffic sign detection. With an accuracy of 87.86%, which is much higher than YOLOv5 (81.54%) and Faster R-CNN (78.89%), Detectron2 also performs really well. Faster R-CNN's rather lower precision indicates that it is more likely to produce erroneous detections, thereby lowering its general dependability for this purpose. Although better than Faster R-CNN, YOLOv5 still lags behind YOLOv8-TDSHA and Detectron2, suggesting space for development in lowering false positive rates.

Recall represents the model's capacity to accurately identify all pertinent instances (true positives) of traffic signs within the dataset. An elevated recall value signifies that the model overlooks fewer genuine instances, hence guaranteeing thorough identification. YOLOv8-TDSHA once more achieves the greatest recall of 98.76%, illustrating its exceptional proficiency in identifying nearly all traffic signs within the dataset. The elevated recall, coupled with its high precision, signifies that YOLOv8-TDSHA is both comprehensive in detection and precise in its predictions. Detectron2 exhibits a recall of 87.43%, which, while considerably lower than that of YOLOv8-TDSHA, remains commendable, indicating it identifies a substantial proportion of true positives, though not as thoroughly. YOLOv5 attains a recall of 82.32%, whilst Faster R-CNN records the lowest recall at 77.43%. The relatively diminished recall of YOLOv5 and Faster R-CNN suggests an increased likelihood of overlooking certain traffic signs, which may be crucial in applications necessitating comprehensive detection. A high F-score signifies that the model is both accurate and thorough in its detections. YOLOv8-TDSHA attains the maximum F-score of 98.49%, solidifying its position as the most efficient model in terms of accuracy and completeness. Detectron2 exhibits a commendable F-score of 86.60%; yet, it is considerably surpassed by YOLOv8-TDSHA. YOLOv5, exhibiting an F-score of 82.19%, and Faster R-CNN, with a score of 78.30%, both demonstrate inferior performance, indicative of their diminished precision and recall metrics.

The significant disparity in F-score between YOLOv8-TDSHA and the other models highlights the superiority of YOLOv8-TDSHA in sustaining elevated levels of both precision and recall, crucial for dependable traffic sign identification.Since it shows the average precision over intersection over union (IoU) thresholds and all classes combined, the mean average precision (mAP) is a crucial parameter for assessing object identification models. A model is resilient and trustworthy in multiple contexts if it performs well across a range of detection thresholds, as indicated by a higher mAP score. The remarkable mAP of 99.03% achieved by YOLOv8-TDSHA is far greater than that of the other models. This finding implies that YOLOv8-TDSHA performs exceptionally well in identifying and categorizing traffic signs under a variety of circumstances and object sizes. Detectron2, with a mean accuracy of 86.51%, is not as good as YOLOv8-

TDSHA, but it is still not that bad. While Faster R-CNN's mAP of 79.4% suggests greater resilience in a variety of detection settings, YOLOv5's mAP of 81.65% implies less robustness. YOLOv5 and Faster R-CNN's lower mAP scores imply that these models would have trouble with more difficult detection tasks, like identifying smaller or partially veiled signs.

Overall, YOLOv8-TDSHA exhibits enhanced performance across all assessment metrics: Precision, Recall, F-score, and mAP, in comparison to the other evaluated models. Its remarkable precision and recall indicate its efficacy in precisely recognizing real positives while avoiding false negatives, rendering it an optimal model for traffic sign detection when accuracy is critical. The exceptional F-score further corroborates its overall balanced performance, while the nearly flawless mAP underscores its robustness and flexibility across diverse detection settings. Conversely, although Detectron2 has commendable performance, it is consistently surpassed by YOLOv8-TDSHA across all criteria. Despite their widespread usage, YOLOv5 and Faster R-CNN have worse performance across all metrics, suggesting they are inadequate for workloads requiring optimal accuracy and reliability. The findings significantly support the utilization of YOLOv8-TDSHA as a state-of-the-art deep learning framework for effective and precise traffic sign detection, providing considerable benefits compared to current methods.

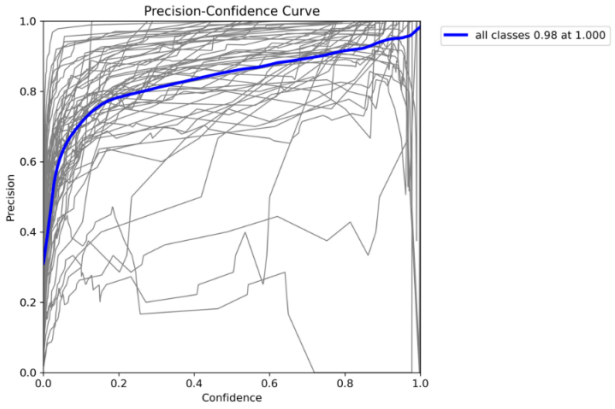


Figure 5: Precision confidence curve (Proposed model)

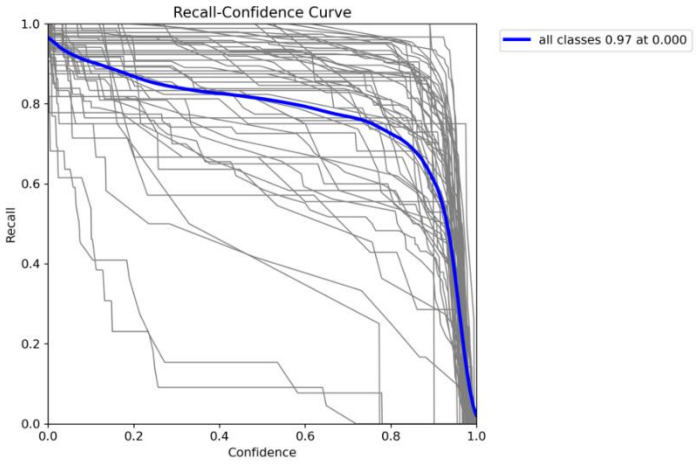


Figure 6: Recall confidence curve (Proposed model)

The link between the precision and confidence threshold of the model is shown by the precision confidence curve. The precision confidence curve is shown in Figure 5. Beginning at a low confidence level of 0.0, the precision is approximately 35%. This indicates that only 35% of the model's positive predictions are now accurate. The precision steadily improves as the confidence criterion rises, hitting 80% at a threshold of 0.5. This implies that the model generates more accurate positive predictions as it gets more selective, that is, only makes predictions when it is more confident. The precision continues to improve as the threshold increases, reaching 82% at a confidence threshold of 0.7. Eventually, as the threshold approaches higher levels (around 0.95 and beyond), the precision reaches near-perfect levels, even touching 100% in some cases. This indicates that, when the model is extremely confident about its predictions, it is almost always correct. The recall confidence curve helps us understand this trade-off between recall and precision. The recall confidence curve is

shown in Figure 6 . The recall-confidence curve illustrates the relationship between recall and confidence, or the percentage of true positives that the model properly detects. At the lowest confidence level (0.0), the recall is approximately 96% at the beginning. This implies that the model, independent of confidence, is initially detecting almost all positive cases. Nevertheless, the recall gradually decreases as the confidence threshold rises. The recall drops to about 90% to 80% by the time the threshold hits 0.7. This decline in recall suggests that as the model grows more certain and picky in its predictions, it starts to overlook more positive examples, which raises the number of false negatives. This is a crucial factor to take into account since, although the model's accuracy increases with confidence, its capacity to capture all pertinent cases diminishes. This implies a trade-off between recall and precision, which is common in classification models, particularly when thresholds for confidence are changed. The recall dramatically decreases at the highest confidence levels, underscoring the model's growing tendency toward more cautious forecasts. It doesn't detect as many true positive cases even though it prevents false positives. Examining the F1-confidence curve helps us to see that the appropriate threshold often strikes a balance between achieving great accuracy and avoiding too much memory loss. By considering false positives and false negatives, the F1-score presents a reasonable evaluation of the accuracy of the model. It computed as the harmonic mean of recall and accuracy. The F1-confidence curve shows the link between precision and recall as well as the general model performance as the confidence threshold rises (Figure 7). The F1 score maintains its stability at reduced confidence thresholds, suggesting a good trade-off between recall and precision. The F1-score starts to drop at the confidence level of 0.79, indicating that the fall in recall is outweighing the improvement in precision. This is to be expected, given memory dramatically declines at increasing confidence thresholds but precision keeps getting better. According to the F1-confidence curve, the model operates most accurately overall at mid-range confidence levels, where it successfully strikes a compromise between recall and precision. The F1-score decreases as the threshold rises, indicating that while the model is becoming more cautious and missing more positive examples, it is also generating extremely few errors when it does make predictions. A thorough grasp of the model's performance at various confidence levels is offered by the graphical assessments of precision, recall, and F1-score confidence curves. The model captures most positive examples (high recall) at lower confidence thresholds, but its precision is reduced. Precision gains when the confidence threshold rises, but memory suffers as a result, leading to an increase in false negatives. The F1-score curve aids in determining the ideal threshold, offering the highest overall performance, by maintaining the balance between precision and recall. These findings imply that although the model's precision works well at high confidence thresholds, it is crucial to select the right threshold in order to prevent an excessive number of false negatives. The model can be optimized for a variety of tasks by adjusting the confidence threshold in order to achieve a balance between capturing all relevant events and producing accurate predictions.

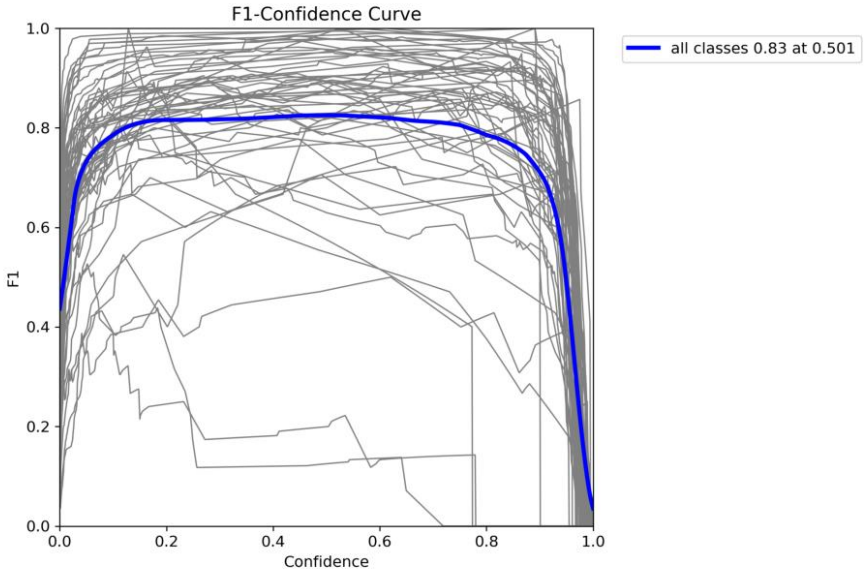


Figure 7: The F1-score confidence curve (Proposed model)

In Figure 8, the training metrics are displayed. Following the first training epochs on the training and validation datasets, the model shows a notable increase in performance measures. It declines rapidly, with the largest drop happening by the fifteenth epoch. The rate of loss reduction then decreases, reaching its lowest point in the 100th epoch. Moreover, precision

and recall exhibit a sharp rise until the fifteenth epoch, at which point they begin to expand more slowly. Both recall and precision reach roughly 0.81 and 0.85 by the conclusion of training, respectively. Though with diminishing benefits as training goes on, these trends are similar across both datasets and show effective learning and model improvement.

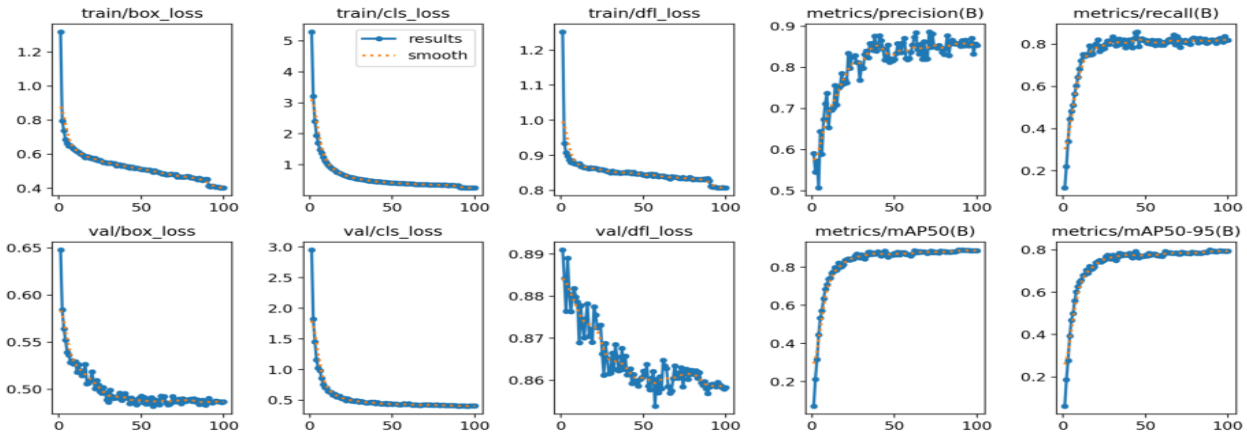


Figure 8: Training metrics if the proposed model

An essential tool for assessing a classification model's performance is the confusion matrix, which shows areas of confusion and offers information on how well the model distinguishes between different classes. Figure 9 shows the confusion matrix for the traffic sign detection model. The findings demonstrate that, despite some misunderstanding with "Red Light" and other categories, the model predicts "Green Light" accurately with a high confidence level of 0.80. The model's accuracy for speed limit signage varies somewhat for various speed restrictions, but it performs well with high confidence in areas like "Speed Limit 10" (0.94) and "Speed Limit 120" (0.98). For the "Stop" sign, the model performs exceptionally well, with a confidence level of 0.96. The lower values in the background category, however, show that the algorithm has difficulty differentiating pertinent traffic signs from background noise. Overall, the model performs well across a number of classes, but it also exhibits considerable misclassification and confusion, especially when it comes to comparable categories and background noise.

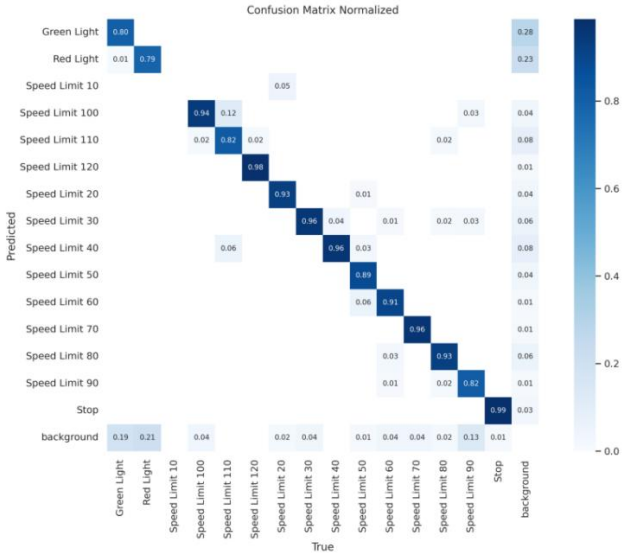


Figure 9: The proposed YOLOV8-TDSHA classification error generation (confusion matrix)

Figure 10 shows the suggested model's comparative performance against a number of cutting-edge techniques. The graphic displays important performance indicators for each model, including mean Average Precision (mAP), precision, f-score, and recall. Compared to current techniques, the suggested model exhibits competitive results with noteworthy gains in all metrics. In the context of traffic sign identification, the figure highlights the usefulness of the proposed method and offers a clear comparison with the state-of-the-art models, highlighting both strengths and opportunities for improvement.

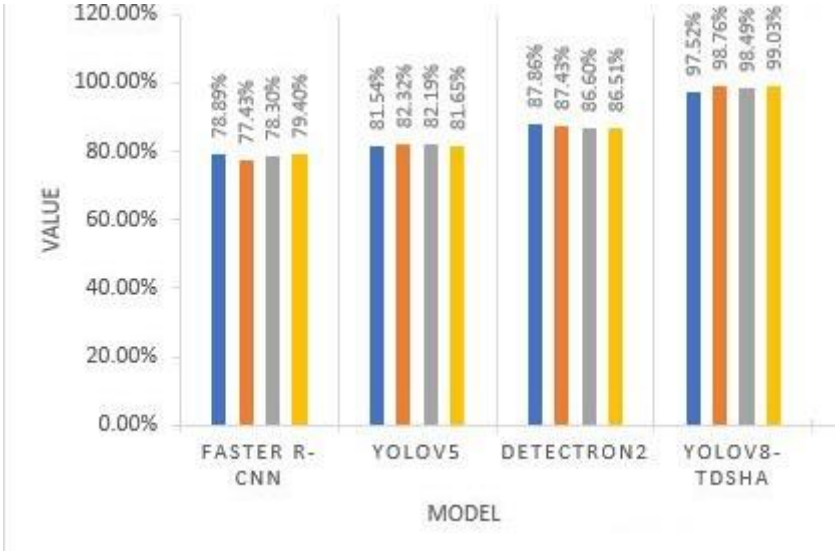


Figure 10: Comparative performance of all applied models in the research

Figure 11: Sample output of the proposed model

Table 4: Comparative Performance Analysis of the Proposed Model and State-of-the-Art Methods

Reference	Model	Performance
[22]	YOLO	mAP : 93.88 %
[23]	Ghost YOLO	mAP : 92.71 %
[24]	TRD YOLO	mAP : 86.3 %

[25]	TSR YOLO	mAP : 92.77 %
Proposed	YOLOV8-TDSHA	mAP : 99.03 %

Table 4 shows the evaluation of the Proposed Model's and State-of-the-Art Techniques' Comparative Performance explains in depth how several traffic sign detection methods compare in terms of mean Average Precision (mAP). The suggested YOLOv8-TDSHA model is shown in the table together with performance measurements from a number of cutting-edge techniques. With a mAP of 93.88%, the YOLO model demonstrates good detection abilities but still has room for improvement. Known for its lightweight design, the Ghost YOLO model records a mAP of 92.71%, which is marginally lower than YOLO but tailored for real-time performance with low computational overhead. With a mAP of 86.3%, the TRD YOLO model—which is optimized for minor traffic sign detection—reflects a compromise between processing efficiency and good performance. With a mAP of 92.77%, the TSR YOLO model—which was customized for Chinese traffic signs in complicated scenes—displays competitive performance that is comparable to that of YOLO and Ghost YOLO. On the other hand, the proposed YOLOv8-TDSHA model achieves an outstanding mAP of 99.03%, greatly outperforming these current approaches. This significant gain shows how the suggested model may be further developed to improve traffic sign detection efficiency and accuracy, making it a better option than other available traffic sign recognition technologies.

The sample output of the suggested YOLOv8-TDSHA model is shown in Figure 11, demonstrating the model's efficacy in traffic sign detection. The picture demonstrates the enhanced performance and precision of the model by showcasing its ability to precisely identify and classify different traffic signs in a variety of settings.

V. CONCLUSION

The study that is reported in this paper shows how well the suggested YOLOv8-DH and YOLOv8-TDHSA models perform in terms of enhancing traffic sign identification in practical settings. Our models outperform existing state-of-the-art techniques in terms of accuracy and speed by utilizing the sophisticated architectural characteristics of YOLOv8 in conjunction with creative changes such as transformer-based self-attention mechanisms and decoupled heads. The experimental findings indicate a significant enhancement in precision, recall, F-score, and mAP, rendering these models exceptionally appropriate for real-time applications in autonomous driving and intelligent transportation systems. The YOLOv8-TDHSA model attains an impressive mAP of 99.03%, highlighting its efficacy across various detection environments. The findings indicate that integrating deep learning advancements like YOLOv8 and related improvements can markedly improve the safety and efficiency of contemporary transportation systems through enhanced accuracy and timeliness in traffic sign identification.

Many future study subjects can be looked at to help us to expand on our work. Initially, extra model optimization might focus on reducing computational complexity to help deployment on edge devices with limited processing capability. Furthermore, broadening the models' training on a wider range of and more substantial datasets may enhance their resilience and capacity to adjust to various traffic situations worldwide. To improve these models' detection performance in harsh environments like dense fog or pouring rain, another interesting approach is to integrate them with additional sensors and data modalities like radar or LiDAR. Eventually, by using the recommended models in real-world autonomous automobile systems and always enhancing them in response to feedback from in-the-moment operations, even further advances in accuracy and dependability could be obtained. By enabling the development of increasingly sophisticated, real-time traffic sign detecting systems, these next studies will significantly help to contribute to the continued expansion of intelligent transportation and autonomous driving technologies.

REFERENCES

[1] A. Kaur, V. Kukreja, N. Thapliyal, M. Aeri, R. Sharma, and S. Hariharan, “An Improved YOLOv8 Model for Traffic Sign Detection and Classification,” in *2024 3rd International Conference for Innovation in Technology (INOCON)*, IEEE, 2024, pp. 1–5.

[2] Z. Huang, L. Li, G. C. Krizek, and L. Sun, “Research on traffic sign detection based on improved YOLOv8,” *Journal of Computer and Communications*, vol. 11, no. 7, pp. 226–232, 2023.

[3] A. De la Escalera, J. M. Armingol, and M. Mata, “Traffic sign recognition and analysis for intelligent vehicles,” *Image Vis Comput*, vol. 21, no. 3, pp. 247–258, 2003.

[4] Y.-B. Jo, W.-S. Na, S.-J. Eom, and Y.-J. Jeong, “Traffic Sign Recognition using SVM and Decision Tree for Poor Driving Environment,” *Journal of IKEEE*, vol. 18, no. 4, pp. 485–494, 2014.

[5] S. Maldonado-Bascón, S. Lafuente-Arroyo, P. Gil-Jimenez, H. Gómez-Moreno, and F. López-Ferreras, “Road-sign detection and recognition based on support vector machines,” *IEEE transactions on intelligent transportation systems*, vol. 8, no. 2, pp. 264–278, 2007.

[6] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, “Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition,” *Neural networks*, vol. 32, pp. 323–332, 2012.

[7] S. Song, Z. Que, J. Hou, S. Du, and Y. Song, “An efficient convolutional neural network for small traffic sign detection,” *Journal of Systems Architecture*, vol. 97, pp. 269–277, 2019.

[8] P. Sermanet and Y. LeCun, “Traffic sign recognition with multi-scale convolutional networks,” in *The 2011 international joint conference on neural networks*, IEEE, 2011, pp. 2809–2813.

[9] C. Dan, M. Ueli, M. Jonathan, and S. Jürgen, “Multi-column deep neural network for traffic sign classification,” *Neural networks*, vol. 32, no. 1, pp. 333–338, 2012.

[10] R. Girshick, “Fast r-cnn,” *arXiv preprint arXiv:1504.08083*, 2015.

[11] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” *IEEE Trans Pattern Anal Mach Intell*, vol. 39, no. 6, pp. 1137–1149, 2016.

[12] J. Redmon, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[13] J. Redmon and A. Farhadi, “YOLO9000: better, faster, stronger,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[14] Thillaiarasu, N., and Ashwin Shenoy. "Enhancing Security Through Real-Time Classification of Normal and Abnormal Human Activities: A YOLOv7-SVM Hybrid Approach." *IAENG International Journal of Computer Science* 51.8 (2024).

[15] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “Yolov4: Optimal speed and accuracy of object detection,” *arXiv preprint arXiv:2004.10934*, 2020.

[16] J. Wang, Y. Chen, Z. Dong, and M. Gao, “Improved YOLOv5 network for real-time multi-scale traffic sign detection,” *Neural Comput Appl*, vol. 35, no. 10, pp. 7853–7865, 2023.

[17] R. Kaur and J. Singh, “Local Regression Based Real-Time Traffic Sign Detection using YOLOv6,” in *2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, IEEE, 2022, pp. 522–526.

[18] S. Li, S. Wang, and P. Wang, “A small object detection algorithm for traffic signs based on improved YOLOv7,” *Sensors*, vol. 23, no. 16, p. 7145, 2023.

[19] Ashwin Shenoy, M., and N. Thillaiarasu. "Enhancing temple surveillance through human activity recognition: A novel dataset and YOLOv4-ConvLSTM approach." Journal of Intelligent & Fuzzy Systems Preprint (2023): 1-16.

[20] Z. Ge, “Yolox: Exceeding yolo series in 2021,” *arXiv preprint arXiv:2107.08430*, 2021.

[21] G. H. Ballantyne, “Review of sigmoid volvulus: clinical patterns and pathogenesis,” *Dis Colon Rectum*, vol. 25, no. 8, pp. 823–830, 1982.

[22] Y. Cui, D. Guo, H. Yuan, H. Gu, and H. Tang, “Enhanced YOLO Network for Improving the Efficiency of Traffic Sign Detection,” *Applied Sciences*, vol. 14, no. 2, p. 555, 2024.

[23] S. Zhang, S. Che, Z. Liu, and X. Zhang, “A real-time and lightweight traffic sign detection method based on ghost-YOLO,” *Multimed Tools Appl*, vol. 82, no. 17, pp. 26063–26087, 2023.

[24] J. Chu, C. Zhang, M. Yan, H. Zhang, and T. Ge, “TRD-YOLO: A real-time, high-performance small traffic sign detection algorithm,” *Sensors*, vol. 23, no. 8, p. 3871, 2023.

[25] W. Song and S. A. Suandi, “Tsr-yolo: A chinese traffic sign recognition algorithm for intelligent vehicles in complex scenes,” *Sensors*, vol. 23, no. 2, p. 749, 2023.

[26] Swetha, A. ., M. S. . Lakshmi, and M. R. . Kumar. “Chronic Kidney Disease Diagnostic Approaches Using Efficient Artificial Intelligence Methods”. International Journal of Intelligent Systems and Applications in Engineering, vol. 10, no. 1s, Oct. 2022, pp. 254

[27] Rudra Kumar, M., Gunjan, V.K. (2022). Peer Level Credit Rating: An Extended Plugin for Credit Scoring Framework. In: Kumar, A., Mozar, S. (eds) ICCCE 2021. Lecture Notes in Electrical Engineering, vol 828. Springer, Singapore. https://doi.org/10.1007/978-981-16-7985-8_128