# Development of An Artificial Neural Network-Based System to Detect Lane and Roadside Traffic Signs

Viraj Sonawane[1], Balasaheb Agarkar[2], Sachin Chaudhari[3]

Research Scholar, Department of E&TC Engineering, Sanjivani College of Engineering, Kopargaon, Maharashtra, India[1]

Faculty Member, Department of E&TC Engineering, Sanjivani College of Engineering, Kopargaon, Maharashtra, India[2, 3]

***Abstract*—Ensuring road safety requires timely and accurate detection of lanes and roadside traffic signs, which are essential for assisting drivers and reducing accident risks. However, robust simultaneous detection remains challenging due to variations in illumination, occlusions, complex backgrounds, and diverse road conditions. This research introduces a novel vision-based system for lane detection and roadside traffic sign recognition using advanced artificial neural network architectures. In the pre-processing stage, Contrast Limited Adaptive Histogram Equalization (CLAHE) is applied to enhance image visibility under varying lighting conditions. For classification, a Self-Artificial Attentive Gated Neuro-Recurrent Unit (SAAG-NRU) is proposed, a novel architecture integrating deep artificial neural network (ANN) layers, Gated Recurrent Units (GRU), and a self-attention mechanism, enabling sequential refinement of features and prioritization of the most safety-relevant cues. In addition, YOLOv11 was utilized to train and detect traffic signs exclusively. For lane detection, a lane mask-based approach is implemented to accurately segment and highlight lane boundaries. The trained model was integrated into a Tkinter-based Graphical User Interface (GUI) to visualize real-time detection outputs effectively. Implemented using Python-based tools, the framework is validated against benchmark lane and traffic signs, demonstrating better performance compared to existing systems in terms of lane detection, achieving 91.11% precision, and traffic sign detection, attaining 96.88% precision. Therefore, the proposed system delivers fast, accurate, and robust simultaneous lane and traffic sign detection, significantly improving real-time road safety and driver assistance.**

***Keywords*—Lane detection; roadside traffic sign recognition; contrast limited adaptive histogram equalization (CLAHE); self-artificial attentive gated neuro recurrent unit (SAAG-NRU); graphical user interface (GUI)***

## I. Introduction

The fast advancement of intelligent transportation systems has led to a considerable amount of interest in the development of advanced driver assistance systems that can guarantee the safety of the roads and allow the navigation of autonomous vehicles [1]. The major perception tasks that are unavoidable for safe driving include recognition of lanes and their adjoining traffic signs, which in turn help in holding the vehicle under control, issuing warnings in case of lane departure, and controlling the signs by allowing or not allowing the passing of vehicles [2]. The accurate reading of roadway cues is even more crucial during difficult driving conditions like poor lighting, shadows, rain, glare, worn-out lane markings, and occluded traffic signs, where traditional vision-based techniques are often found lacking [3].

Lane markings and roadside signs are the primary visual aids that indicate the safest way for drivers and self-driving cars to handle the roads by showing rules, warnings, and directions [4]. Lane markings not only determine the limits of a car's road but also control traffic, keep cars in their lanes, and prevent them from accidentally changing lanes, especially when moving fast or in a complicated road situation [5]. Roadside signs, which comprise regulatory, warning, and informational signs, give out ten critical instructions such as speed limits, turns, pedestrian crossings, and hazard alerts, thereby allowing for timely decision-making for safe driving. Lane recognition and road sign detection work together and are the basis of intelligent transportation systems and autonomous driving functions, thus ensuring accurate vehicle localization, route planning, and collision prevention in a variety of road conditions [6-8].

Among the different key features, marking lanes and detecting roadside signals are key features of current intelligent transportation systems and self-driving cars, since they allow vehicles to recognize the geometry and the rules of the roads instantly [9]. Extracting lanes means detecting the limits and the crest of the road, thereby facilitating turning and preventing wrong exits in a lane, even if the conditions are very difficult, like low light, faded paint, shadows, or obstruction [10]. Road sign detection and recognition extract visual information from traffic signs, like speed limits, warnings, direction indicators, and pedestrian alerts, to help with safe and legal driving. Vehicles with lane and road sign detection working together have a thorough awareness of their environment; hence, the accuracy and speed of their navigation, decision-making, and overall safety in road use are significantly increased even in dynamic and unpredictable surroundings [11].

Conventional computer-vision algorithms depend heavily on handcrafted features and are highly sensitive to scene variations, camera perspective, and environmental noise. A real-time traffic sign recognition system was created that could recognize more kinds of traffic signs for intelligent vehicles [12]. The method is a combination of Haar cascade detection and an attention-based deep convolutional neural network (CNN) classifier that is trained on the GTSRB dataset. The system performed well and attained greater accuracy on testing and F1 measure. Nevertheless, the performance of the system was reduced in situations with difficult lighting and motion blur, and its dependence on the GTSRB dataset also restricted its applicability to different real-world scenarios. The condition of pavement lane markings and developing a maintenance strategy based on priority were evaluated [13]. The findings indicated that white lanes were the most affected by deterioration among all road types, whereas those in the urban freeway area had the lowest defect ratios. The research is restricted to a single metropolitan road network, three road classifications, and lane-level averaging resolution. These constraints hinder their practical deployment in real-world scenarios. Therefore, this research introduces a novel vision-based system for lane detection and roadside traffic sign recognition using Self- Artificial Attentive Gated Neuro-Recurrent Unit (SAAG-NRU) that addresses these drawbacks.

### A. Contribution of this Research

This research intends to propose and introduce a novel vision-based system for lane and roadside traffic sign detection. Initially, the Traffic Signs Dataset in YOLO format is gathered, followed by data preprocessing using Contrast Limited Adaptive Histogram Equalization (CLAHE). Then, the model classification uses the SAAG-NRU model that leverages deep artificial neural network (ANN) layers, Gated Recurrent Units (GRU), and a self-attention mechanism that enables sequential refinement of features and prioritization of the most safety-relevant cues. Afterwards, create the Lane Mask and Traffic Sign Model. Although, Graphical User Interface (GUI) is used with Tkinter for Lane and Traffic Sign Detection. Then, the performance efficiency of the proposed framework is analyzed using classification parameter metrics and GUI visualization.

## II. Literature Review

This part outlines the important research that has concentrated on enhancing lane and Traffic Sign Detection using vision-based and Deep Learning (DL) techniques; refer to Table I. To address these limitations, the proposed method integrated CLAHE enhancement, YOLOv11 detection, and SAAG-NRU classification with lane mask-based segmentation, ensuring real-time, high-precision, and robust detection across diverse road and illumination conditions.

TABLE I. Summary of Existing DL and Vision-Based Approaches for Lane and Roadside Traffic Sign Detection

| Ref | Purpose | Approach | Findings | Constraints |
|---|---|---|---|---|
| [14] | Developed a low-cost real-time pedestrian and priority sign detection system to improve road safety. | Trained a Single Shot Detector-MobileNet model on a custom road dataset. | The system identified pedestrians, pedestrian crossings, and stop signs with low latency. | Reduced performance in low visibility and occlusion. |
| [15] | Developed the ELVD ensemble CNN to improve traffic sign recognition and reduce overfitting. | Trained an ensemble of LeNet, VGGNet, and DropoutNet. | Achieved better accuracy, 0.059% loss, and faster prediction than individual baseline models. | Dataset imbalance, limited real-world testing, and varying image quality affected class predictions. |
| [16] | Developed a lane detection framework capable of handling varied weather conditions for ADAS. | Employed a two-tier Deep CNN with Local Vector Pattern and optimized Flight Straight of Moth Search. | Achieved higher accuracy and faster computation. | Performance decreased in extreme weather and complex road scenarios. |
| [17] | Developed a DL system to recognize road surface traffic signs and distinguish left/right markings. | Trained Yolo V2, V3, V4, and V4-tiny models. | YOLOv4 (No Flip) achieved the best performance and 66.12% IoU, outperforming other models. | Performance depended on dataset representativeness. |
| [18] | Developed a DL system to identify and recognize traffic signs and signals for safer self-driving vehicles. | Employed multi-task CNNs with pre-trained models and a region of interest module. | Achieved greater accuracy and successfully tested real-time detection on highways. | Limited real-world testing. |
| [19] | Developed a computer vision model to detect turning lanes along with roadway data collection. | Employed a YOLOv3-based turning lane detection model. | Accomplished 80.4% accuracy and detected 13,800 turning lane characteristics. | Accuracy was affected by image quality, faded markings, and complex intersections. |
| [20] | Developed a DL–based TSR system to enhance road security by identifying and classifying the traffic signs. | Used RetinaNet for recognition and DenseNet-121 for classification. | Attained high recognition and classification accuracy, outperforming existing models. | Performance may drop in poor lighting or occluded signs. |
| [21] | Developed a fast and accurate lane recognition system for intelligent vehicles in complex road conditions. | Employed Gaussian blur, Sobel edge detection, Otsu thresholding, and Hough Transform for lane detection. | Achieved high lane detection accuracy with fast processing. | Less effective in noisy, occluded, or low-light conditions. |
| [22] | Developed a system to detect obstacles and lane states. | Trained LaneScanNET with ODN and LDN fused by OLFN. | Achieved 75.28% obstacle detection accuracy and better lane status accuracy. | The limited dataset and not extensively tested. |
| [23] | Developed a lightweight framework for real-time lane detection and path forecast. | Designed DSUNet for a path-forecasting approach to form a CNN-path prediction (CNN-PP) simulation. | DSUNet was smaller and achieved lower lateral errors. | Evaluation relied on limited real-road testing and specific driving scenarios. |
| [24] | Developed an IoT-based framework for reliable lane recognition beneath artificial lighting within the tunnels and highways. | Implemented a three-module system with illumination-invariant lane detection on board and cloud processing during detection failure. | Achieved better lane-keeping rates in tunnels and highways with 31 ms/frame processing time. | Depended on network connectivity for cloud assistance and was tested only on limited road environments. |
| [25] | Developed a DL-based system to improve lane and traffic sign recognition under road circumstances. | Employed transfer learning on YOLOv8. | Attained 90.6% mAP at 117 fps for lane and 81.3% mAP at 56 fps for traffic sign detection. | Performance declined in extreme weather and heavily eroded lane conditions. |

Existing systems still lack simultaneous multi-component recognition and show reduced reliability in low visibility, occlusion, extreme weather, and eroded lane markings [14, 16, 20, 21]. Performance also declined due to dataset imbalance, limited real-road validation, and dependence on environment-specific training [15, 18, 23]. Accuracy declines in complicated scenarios, and reliability in adverse weather conditions continue to be challenges [14, 16, 19, 20, 21]. It is uncommon to try lane and traffic sign detection simultaneously, and most attempts are made on limited datasets [22]. Performance declines due to imbalanced datasets, inadequate validation for real-world settings, and limited resources [15, 17, 24].

## III. Proposed Methodology

Research introducing a novel vision-based structure for lane and roadside traffic sign detection. The process contains the collection of the traffic signs dataset in YOLO format, data processing using CLAHE, implementing the SAAG-NRU model, Lane Mask creation, traffic sign model creation, and results.

### A. Data Collection

The Traffic Signs Dataset in YOLO format is gathered for this research, and it was sourced from the Kaggle platform. Link: https://www.kaggle.com/datasets/valentynsichkar/traffic-signs-dataset-in-yolo-format. The dataset consists of 741 .jpg image files and corresponding .txt annotation files (YOLO format): every image has a matching annotation file.

### B. Preprocessing

The intention of preprocessing is to enhance image quality, remove noise, and improve feature visibility for accurate lane and sign detection. This preprocessing step utilizes CLAHE, which improves contrast under varying lighting conditions by adaptively enhancing brightness levels without amplifying noise in important structural regions. It consists of five primary processes. Initially, the image is split into equal-sized rectangular segments, and then histogram modifications are carried out in every segment. Histogram modification involves histogram construction and clipping, along with redistribution. The mapping operation is subsequently obtained by employing the Cumulative Distribution Function (CDF) of a clipped histogram. At last, a bilinear interpolation method is implemented among the segments to eliminate the potential block distortions. CLAHE differs from conventional Histogram Equalization (HE) by restricting the contrast through a clipping point that truncates the peak values in every block's histogram. The clipped pixels are reallocated to every gray level. A higher clip point leads to greater contrast enhancement. The clipping point is computed in the following manner as stated in Eq. (1).

$$\beta = \frac{N}{M}\left(1 + \frac{\alpha}{100} T_{max}\right) \tag{1}$$

In Eq. (1), $N$ denotes the number of pixels in every segment, $M$ constitutes the segment dynamic range, $T_{max}$ denotes the maximal slope, and a indicates the clipping factor. Whenever $\alpha$ is near to zero, the clipping point is $N/M$, hence, the pixel in this segment is constant. As $\alpha$ approaches 100, the contrast becomes significantly improved to a larger degree. Therefore, the clipping point is the most important component to modify the enhancement of contrast. According to the CDF, it obtains a mapping operation for remapping the gray levels within the image segments in the following manner, as stated in Eq. (2) and (3).

$$cdf(k) = \sum_{l=0}^{k} p\, df(k) \tag{2}$$

$$S(k) = cdf(k) \times k_{max} \tag{3}$$

where, $S(k)$ indicates the actual process of remapping. Here, $k$ signifies the grayscale level of the pixel, and $k_{max}$ holds the value of the pixel highest in this segment. The CDF of the reallocated histogram in every segment obtains distinct remapping operations. To prevent segmentation distortions, every pixel's value is interpolated from the mapping operation in the neighboring segments. Therefore, CLAHE produces clearer, high-contrast images with improved lane edges and sign details, enabling more stable and accurate detection performance.

### C. Model Classification Using Self-Attentive Gated Neuro-Recurrent Unit (SAAG-NRU)

The SAAG-NRU approach utilizes for model classification that integrates Deep ANN layers, GRU, and a self-attention mechanism, enabling sequential refinement of features and prioritization of the most safety-relevant cues. Also, the proposed SAAG-NRU model trains traffic sign and lane detection. The self-attention mechanism prioritizes safety-critical features by weighting important regions, enhancing robustness under challenging visual conditions. The deep ANN extracts rich spatial features by learning complex visual patterns essential for lane and traffic sign detection. GRU captures temporal dependencies across consecutive frames, stabilizing predictions and improving sequential decision accuracy.

*1) Self-attention mechanism*: The self-attention mechanism selectively emphasizes high-value features while suppressing inappropriate background data, improving focus and interpretability in complex environments. The scaled dot-product attention technique is employed for self-attention mechanisms. This enables the framework to allocate significance scores to various portions of the input order, thus capturing dependencies and interactions between components. As shown in Fig. 1, the result of the final hidden layer is integrated with the outcome of the self-attention layer to produce a hybrid-based hidden layer that includes both acquired characteristics and contextual perceptions.

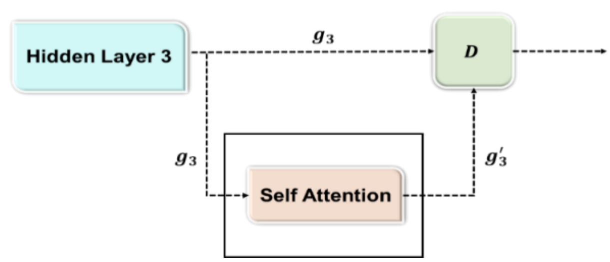


Fig. 1. Framework for hybrid hidden layer.

The hybrid layer comprises a hidden layer as shown in Fig. 1 and an activation operation defined by Eq. (5).

$$g_2 = Output\ of\ hidden\ layer\ 2$$

$$b = \sum x_l g_{2l} + a \tag{4}$$

$$g_3 = z(b) \tag{5}$$

The hidden layer's output vector $g_3$ is transferred to the self-attention layer and is linearly transformed into a vector $g_{3b}$ by employing a weight matrix $x_x$, Eq. (6). Following that, employing scaled dot-product attention, the key $(l)$, query $(r)$, and value $(u)$ vectors are computed, as stated in Eq. (7), (8), and (9).

$$g_{3b} = g_3 x_x + a_1 \tag{6}$$

$$r = g_{3b} x_r + a_r \tag{7}$$

$$l = g_{3b} x_l + a_l \tag{8}$$

$$u = g_{3b} x_u + a_u \tag{9}$$

where, $x_r$, $x_l$, and $x_u$ represent weight matrices, whereas $a_r$, $a_l$, and $a_u$ denote bias vectors. The dimension of the $r, l, and\ u$ vectors is defined by the attention unit's selection (16, 32, or 64). After calculating $r, l, and\ u$, the attention vector $\alpha$ is computed by employing a 'softmax' operation. To compute $\alpha$ value, the query and the key vector's transpose are multiplied and scaled from zero to one as stated in Eq. (10).

$$\alpha = softmax\left(\frac{r^* l^S}{\sqrt{c_l}}\right) \tag{10}$$

The $g_3'$ vector is computed through the multiplication of $(\alpha)$ the attention vector and the value vector, which is represented in Eq. (11).

$$g_3' = \alpha u \tag{11}$$

The vector $g_3'$ indicates the mechanism of self-attention output embedded in a specific hidden layer. Finally, $g_3'$ gets combined with the $hidden\ layer$ 3 resultant vector, as seen in Fig. 2 and explained in Eq. (12). The combined outcome is the last contextualized vector, which is transmitted to the resultant layer.

$$g_e = concat(g_3', g_3) \tag{12}$$

It generates attention-weighted features highlighting crucial regions and concludes that self-attention significantly boosts detection precision and situational awareness.

*2) Deep artificial neural network (ANN) layers*: The purpose of Deep ANN is to learn hierarchical spatial representations from images, enabling strong feature extraction for downstream classification and segmentation tasks in an automated way. The expression for the regularized input-to-output relationship is represented mathematically as follows:

$$\eta = x_0^{(2)} + \sum_{i=1}^{m} \left(\frac{x_i^{(2)}}{1+\exp(-\propto u_i)}\right) \tag{13}$$

Wherein,

$$u_i = x_{0,i}^{(1)} + \sum_{j=1}^{7} (x_{j,i}^{(1)} \xi_j) \tag{14}$$

In Eq. (13) and (14), $x_{0,i}^{(1)}$, and $x_0^{(2)}$ are thresholds, $x_{j,i}^{(1)}$, and $x_i^{(2)}$ are synaptic weights, $\alpha$ indicates the activation function's shape parameter, $j$ denotes a neuron index in the input layer, $i$ signifies the neuron indices in the hidden layers, and $m$ represents the total number of neurons within the hidden layer. The proper number of neurons within the hidden layer needs to be established.

To train the ANN, $M$ records have been selected at random from the database. An error residual is represented in the following Eq. (17):

$$F = \sum_{l=1}^{M} (\eta_l^{(p)} - \eta_l^{(c)})^2 \tag{15}$$

$\eta_l^{(c)}$ represents the target value that demonstrates the data's normalized compressive strength; the outcome of the ANN is $\eta_l^{(p)}$, while, $l$ represents the sequential number. The underline for $l$ indicates that the samples are renumbered after being randomly selected from the data. The training procedure involves identifying a set of appropriate thresholds and synaptic weights. Initially, all thresholds and synaptic weights are set to smaller random integers.

The error residual $(F)$ depends on the thresholds and synaptic weights. It employs the steepest descent approach to update their estimated values and lead them toward a better solution:

$$x_{j,i}^{(1),new} = x_{j,i}^{(1),old} - \mu \frac{\partial F}{\partial x_{j,i}^{(1)}} \tag{16}$$

$$x_i^{(2),new} = x_i^{(2),old} - \mu \frac{\partial F}{\partial x_i^{(2)}} \tag{17}$$

In Eq. (16) and (17), $\mu$ represents the correction step parameter. It produces rich spatial embeddings, enabling accurate recognition, and concludes that ANN layers provide foundational feature strength for the overall system.

*3) Gated Recurrent Units (GRU):* The purpose of GRU is to model temporal continuity between frames, preserving historical context and reducing detection noise during dynamic road scene transitions. The GRU has a significant advantage over other architectures because it needs fewer parameters, trains faster, and needs less data to generalize. Fig. 2 illustrates the architecture of the GRU approach.

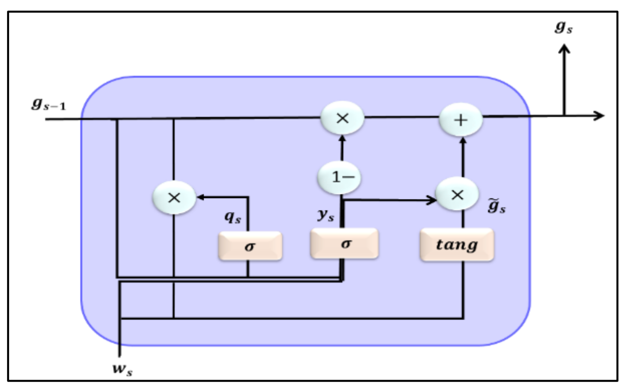


Fig. 2. The internal architecture of the GRU mechanism.

The updating and resetting gates generate intermediate values $y_s$ and $q_s$, respectively, whereas the ultimate memory of the general-use unit stores the outcome as $g_s$. The update indicates the quantity of preceding input $(w_s)$ and outcome $(g_{s-1})$, which are communicated to the following cell, determined by the weight $(x_s)$. The reset gate establishes how much data is removed from the memory. The subsequent equations represent the most important expressions that define the functioning of the GRU model.

$$y_s = \sigma\left(X_y.[g_{s-1}, w_s]\right) \quad (18)$$

$$q_s = \sigma\left(X_q.[g_{s-1}, w_s]\right) \quad (19)$$

$$\tilde{g}_s = tang\left(X_g.[q_s * g_{s-1}, w_s]\right) \quad (20)$$

$$g_s = (1 - y_s) * g_{s-1} + y_s * \tilde{g}_s \quad (21)$$

In Eq. (18)-(21), $X_y$ represents the updating gate, $X_q$ denotes the resetting gate, and $X_g$ specifies the candidate activation weight matrices. The function · signifies matrix-based multiplication, $*$ denotes component-wise multiplication. The operations $\sigma$ is the sigmoid and $tang$ is the hyperbolic tangent activation operation. Therefore, the GRU generates temporally consistent feature vectors, and concludes that the GRU effectively stabilizes predictions and enhances reliability during continuous driving scenarios. Prior to classification, CLAHE-enhanced images are processed by YOLOv11 for localizing traffic sign regions, and by the lane-mask module to segment lane regions. Each detected region is then encoded into a feature vector. This vector embeds four equal-length groups, providing the recurrent and self-attention components, which are processed by SAAG-NRU. The hyperparameter details of the proposed SAAG-NRU are provided in Table II.

TABLE II. HYPERPARAMETER CONFIGURATION FOR SAAG-NRU MODEL

| Category | Hyperparameter | Value / Range | Description |
|---|---|---|---|
| Architecture | Input Vector dimension ($D_x$) | 768 | Dimensionality of input feature vector |
| Architecture | Hidden units (H) | 256 | Size of recurrent hidden state |
| Architecture | Attention dimension (A) | 128 | Size of artificial attention vector |
| Architecture | Number of SAAG-NRU layers | 1–3 | Stacked recurrent layers |
| Attention | Memory window size (K) | 4 | Number of past hidden states used for self-attention |
| Attention | Attention scaling (α) | 1.0 | Temperature/scaling factor for attention scores |
| Gating | Update gate activation | Sigmoid | Controls memory retention |
| Gating | Reset gate activation | Sigmoid | Controls past state influence |
| Gating | Attention gate activation | Sigmoid | Balances candidate state and attention vector |
| Regularization | Dropout rate | 0.1–0.3 | Applied after hidden state update |
| Regularization | Weight decay | 1e-5–1e-4 | L2 regularization to prevent overfitting |
| Optimization | Optimizer | Adam | Gradient-based optimizer |
| Optimization | Learning rate | 1e-3 | Initial learning rate |
| Optimization | Batch size | 16–64 | Number of sequences per batch |
| Training | Number of epochs | 50–200 | Total training iterations |
| Training | Gradient clipping | 1.0 | Prevents gradient explosion |

The proposed SAAG-NRU algorithm is represented in Algorithm 1. The hybrid benefit of SAAG-NRU lies in its ability to fuse spatial feature learning (from deep ANN layers), temporal pattern understanding (from GRU), and selective feature prioritization (from self-attention), allowing the model to simultaneously extract rich visual details, track changes across frames, and focus only on the most safety-relevant information, resulting in higher accuracy and robustness than using these components individually.

**Algorithm 1:** SAAG-NRU for lane detection and roadside traffic sign recognition

```
BEGIN SAAG-NRU
1: Initialize h₀ ← [0.0, 0.0]
2: FOR t = 1 TO T DO
3:   Read input x_t
4:   u_t ← CONCATENATE(h_{t−1}, x_t)
 // ---------------- USER-GATED RECURRENCE ---------
5:     z_t ← σ(Wz · u_t)
6:     r_t ← σ(Wr · u_t)
7:    h̃_t ← tanh(Wc · CONCATENATE (r_t ⊙ h_{t−1}, x_t))
      // ------------- SELF-ARTIFICIAL ATTENTION ------
8:     a_t ← tanh(Wa · u_t)
9:     g_t ← σ(Wg · u_t)
  // ---------------- ATTENTIVE GATED UPDATE -----
10:    h_t ← (1 − z_t) ⊙ h_{t−1}
          + z_t ⊙ ( g_t ⊙ h̃_t + (1 − g_t) ⊙ a_t )
11: END FOR
12: RETURN {h₁, h₂}
END SAAG-NRU
```

## D. Lane Mask Creation

Lane mask creation is used to isolate the road region to accurately detect lane-related colours and markings. It utilizes K-means clustering that segments the image pixels into meaningful groups, helping separate lane regions from the background for improved lane boundary detection, and morphology that removes noise, fills gaps, as well as refines detected lane regions to produce smoother, continuous lane boundaries for accurate segmentation.

The k-means clustering technique separates a given sample collection into $l$ groups based on the distance among the samples that makes the points in the cluster as near as feasible, while the distance among the clusters as far as feasible. The technique is divided into 2 phases that are independent. In the initial step, $l$ centers are chosen at random, with a predetermined value for $l$.

The following stage seeks to categorize every data point to the closest center. This iterative procedure continues until the criteria function achieves its minimum. It is assumed that the clusters are separated into $D_1, D_2, \ldots, D_l$, and the purpose is to reduce the squared error $F$, which is demonstrated in Eq. (22).

$$F = \sum_{j=1}^{l} \sum_{s \in D_j} \left\| s - v_j \right\|_2^2 \quad (22)$$

$\mu_j$ represents the mean vectors of the cluster $D_j$, also known as the centroid. The expression is presented in Eq. (23).

$$v_j = \frac{1}{|D_j|}\sum_{s\in D_j} s \tag{23}$$

The Euclidean distance is primarily employed to compute the distance between every data source and the cluster center. The Euclidean distance $c(w_j, z_j)$ between one vector $w = (w_1, w_2, \dots w_m)$ and another vector $z = (z_1, z_2, \dots z_m)$ is expressed as Eq. (24).

$$c(w, z) = [\sum_{j=1}^{m}(w_j - z_j)^2]^{1/2} \tag{24}$$

This k-means clustering approach is simplistic in concept, easiest to implement, and converges quickly. The clustering impact is stronger, and the technique's interpretability is enhanced. K-means offers pixel clusters, and it enhances lane isolation by separating road surface and lane markings into distinct color-based segments.

Morphology plays an important role in lane detection as it cleans up broken lane segments affected by shadows, blurriness, or clutter. Morphological operations make the detected lanes look continuous and stable by smoothing the edges and enhancing the line structures, which in turn makes it easier for the model or post-processing algorithms to interpret the lanes accurately. Thus, morphology not only gives cleaner masks but also strengthens the lane shapes by removing scattered noise and creating coherent, connected lane boundary structures.

### E. Traffic Sign Model Creation

In the case of Train and Test Path updating, it alters the file paths that are specified in train.txt and test.txt by changing the respective initial directory locations to the new base directory. It takes the image file name only for each line in the input files, adds it to the new dataset path that is specified, and then the altered complete paths are written to the new output files (train_updated.txt and test_updated.txt). This ensures that the training and testing lists point to the correct folder structure required for traffic-sign model training.

For YAML File Creation, this automates the creation and use of a YAML configuration file for training a YOLOv11 traffic-sign detection system. YOLOv11 aims to detect and categorize road signs in real-time by extracting features from labeled images for the purpose of either fully or partly driven cars. To begin with, it creates a traffic_sign.yaml file that has the locations of the new train/validation/test text files along with the number of classes (nc=1) and the name of the class. Next, the pretrained YOLOv11-medium model (yolo11m.pt) is downloaded, and its performance is evaluated using the YAML dataset; finally, the trained model is converted to ONNX format for deployment. Therefore, the YOLOv11 model not only detects traffic signs but also does so at a very high confidence level, so it displays bounding boxes and makes class predictions that support dependable detection in real-time.

### F. Tkinter-Based GUI for Integrated Lane and Traffic Sign Detection

The trained model has been built into a Tkinter-based GUI to effectively visualize the real-time detection outputs. Hence, an interactive platform is provided for easy user interpretation, visualizing lane identification as well as traffic-sign detection results in real-time. The Tkinter-based GUI combines lane and traffic-sign recognition into one interactive structure. When the user uploads an image, the program uses pre-generated lane masks to overlay lane boundaries and simultaneously detects traffic signs with the YOLOv11 ONNX model. Every recognized sign gets cropped and compared with a collection of categorized sign templates for identification, and its type is shown as a labeled bounding box. The GUI provides a clean, real-time visualization environment for unified lane highlighting and traffic sign classification.

## IV. Results And Discussion

The present research effectively introduced a novel vision-based system for lane and roadside traffic sign detection using SAAG-NRU. This section includes the experimental setup, assessment outcomes of comparative analysis, and a Tkinter-based GUI.

### A. Experimental Configuration

The experimental setup for this research focuses on achieving high-performance real-time lane and roadside traffic sign detection. Hardware and software components are carefully selected to ensure fast computation, efficient model training, and seamless GUI integration, as represented in Table III.

TABLE III. Experimental Setup for the Lane and Traffic Sign Detection System

| Category | Component | Specifications |
|---|---|---|
| Hardware | Processing Unit | Intel i5 CPU with 32 GB RAM |
| | Graphics Unit | NVIDIA RTX 3060 GPU |
| | OS | Windows 11 64-bit |
| | Storage | 1 TB SSD |
| Software | Programming Language | Python 3.10 |
| | GUI Framework | Tkinter-based real-time interface |
| | Supporting Libraries | OpenCV, PyTorch, NumPy, Matplotlib, YAML, Ultralytics |

### B. Ablation Study

To ensure the significance of each component, the study was designed with ANN, ANN + GRU, and ANN + Attention combinations. Reproducibility was ensured with the use of random seed 42. GRU with a base ANN model does help with lane detection with its ability to store historical context. It also helps in reducing detection noise in the traffic sign detection task.

Self-attention improves performance by 4.59, 5.75, and 6.05 for precision, recall, and F1-score in the lane detection task. Its ability to remove the background clutter helped the traffic sign detection task significantly.

SAAG-NRU benefits from the individual abilities of GRU and self-attention for both tasks. Details of the ablation study are enumerated in Table IV.

TABLE IV. ABLATION STUDY FOR THE LANE AND TRAFFIC SIGN DETECTION SYSTEM

| Task | Model | Classification Parameter | | | |
|---|---|---|---|---|---|
| | | Precision | Recall | F1-Score | (mAP) |
| Lane | ANN only | 84.73 | 84.39 | 84.06 | 89.14 |
| | ANN + GRU | 85.03 | 85.39 | 86.39 | 89.91 |
| | ANN + Attention | 89.32 | 90.14 | 90.11 | 89.76 |
| | SAAG-NRU | 91.11 | 92.17 | 91.57 | 98.8 |
| Raod Side Traffic Sign | ANN only | 94.12 | 94.1 | 94.06 | 94.87 |
| | ANN + GRU | 95.77 | 95.77 | 95.12 | 95.97 |
| | ANN + Attention | 96.56 | 96.55 | 97.06 | 97.02 |
| | SAAG-NRU | 96.88 | 97.5 | 97.14 | 98.3 |

### C. Comparative Analysis

The comparative analysis compares the proposed method against existing methods by using evaluation metrics that ensure the effectiveness of lane identification and roadside traffic-sign detection. The proposed SAAG-NRU model effectively trains the road sign and lane detection. YOLOv8 [25] is the existing method for this research that includes lane and road sign detection. The utilized performance metrics, such as precision, recall, F1-score, Mean Average Precision (mAP), and Inference Time/Processing Time, are as per standard definitions.

TABLE V. COMPARATIVE NUMERICAL OUTCOMES OF PROPOSED AND EXISTING METHODS USING PARAMETERS

| Classification Parameter | Yolov8 [25] | | SAAG-NRU [Proposed] | |
|---|---|---|---|---|
| | Lane Detection | Road Sign Detection | Lane Detection | Road Sign Detection |
| Precision | 89.0 | 84.7 | 91.11 | 96.88 |
| Recall | 85.5 | 71.4 | 92.17 | 97.50 |
| F1-Score | 87.0 | 77.0 | 91.57 | 97.14 |
| (mAP) | 90.6 | 81.3 | 98.80 | 98.30 |
| Inference Time/Processing Time | 117 (Inference Time) | 56 (Processing Time) | 24.96 (Inference Time) | 2.48 (Processing Time) |

The comparative evaluation between YOLOv8 [25] and the proposed SAAG-NRU framework demonstrates substantial improvements across all classification metrics for both lane detection and roadside traffic sign recognition. For lane detection, the proposed model achieves a Precision of 91.11% compared to 89.0% in YOLOv8 [25], indicating a reduction in false positive lane predictions. Similarly, Recall improves from 85.5% to 92.17%, reflecting enhanced sensitivity in identifying true lane markings under varying illumination and road conditions. This improvement is further confirmed by the F1-Score increase from 87.0% to 91.57%, signifying balanced enhancement in both precision and recall, while achieving a better mAP outcome (98.80). Table V represents the results of existing and proposed methods.

In road sign detection, the performance gain is even more significant. Precision increases from 84.7% to 96.88%, while Recall rises from 71.4% to 97.50%, indicating that the proposed SAAG-NRU architecture effectively minimizes both missed detections and false alarms. The F1-Score improves from 77.0% to 97.14%, demonstrating superior classification reliability. Furthermore, the mAP improves from 81.3% to 98.30% for road sign detection. These results confirm that the integration of sequential refinement significantly enhances spatial-temporal feature prioritization, leading to more robust and accurate detection in complex driving environments.

In addition to accuracy improvements, the proposed model achieves substantial gains in computational efficiency. The existing YOLOv8 [25] model requires 117 ms for inference and 56 ms for processing, which limits real-time deployment capabilities in high-speed driving scenarios. In contrast, the proposed SAAG-NRU framework reduces inference time to 24.96 ms and processing time to only 2.48 ms. This reduction represents a significant acceleration in response time, enabling near real-time frame analysis with improved throughput. The lower processing latency ensures that detected lane boundaries and traffic signs are updated rapidly within the GUI interface. The efficiency improvement is attributed to optimized feature prioritization via self-attention, reduced redundant computations, and streamlined model integration. Consequently, the proposed system not only enhances detection accuracy but also ensures practical feasibility for real-world vehicular applications where both precision and speed are critical.

### D. Visualization of Tkinter-Based GUI

The Tkinter-based interface developed is quite user-friendly and allows for the unified detection of both lanes and traffic signs. The GUI of the application is not only simple but also very clean and contains the choices of uploading an input image and start the prediction process, as shown in Fig. 3(a). The moment the user picks an image from the dataset, an on-screen notification alerts him/her of the successful upload, as shown in Fig. 3(b). After that, when the user clicks the “Predict Lane & Traffic Sign” button, the system at the same time extracts the lane boundaries using the pre-generated lane mask that corresponds to the input frame and detects traffic signs using the trained YOLOv11-ONNX model. The recognized lane markings are displayed in green, while traffic signs are marked with red bounding boxes, labeled according to template-matching with the sign database, as shown in Fig. 3(c). The output has been annotated and is made visible to the user within the GUI, which concurrently depicts both the lane guidance and recognized roadside signs in a single integrated visualization. This user-friendly GUI also serves as a platform for showcasing the capability and practical usability of the proposed detection system.

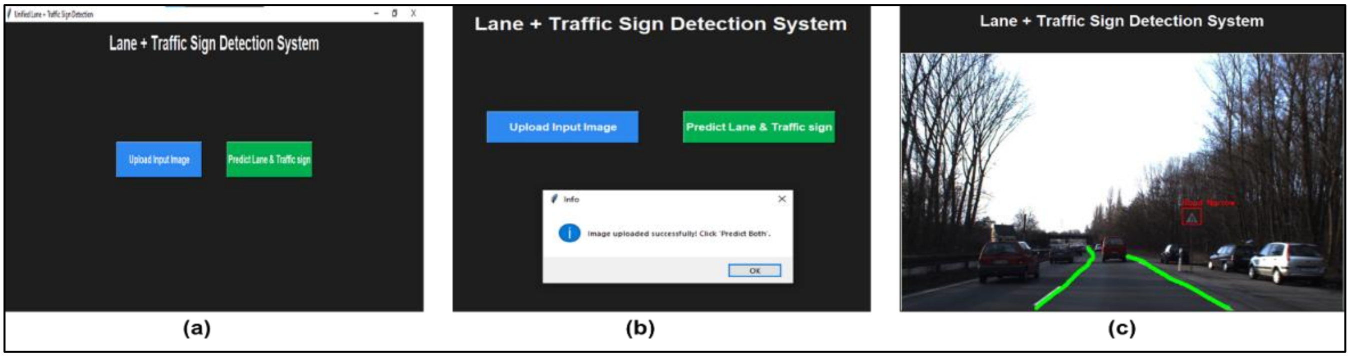


Fig. 3. Visual representation of (a) Tkinter GUI interface showing image upload and prediction, (b) Confirmation dialog indicating successful upload of the selected input image, (c) Final detection output displaying lane boundaries and recognized traffic signs on the road scene.

### E. Discussion

The present research effectively introduced the novel vision-based system for lane and roadside traffic sign detection using SAAG-NRU. For this research, YOLOv8 was the existing method. YOLOv8 [25] shows strong performance for individual object detection, but its accuracy decreases under low illumination, partial occlusions, and complex road backgrounds, leading to missed or false detections. It also lacks efficient multi-task capability for simultaneous lane and traffic sign detection. Additionally, its inference time was relatively high for real-time deployment. The proposed SAAG-NRU with YOLOv11 overcomes these drawbacks by enhancing feature refinement through self-attention, improving recognition under challenging environments, and enabling dual detection of lanes and signs.

A traffic sign detector using the YOLOv8 framework reported 97.8% precision and 96.4% recall [2]. YOLOv4 reported 66.12% IoU for road-surface sign identification [17]. Turning-lane feature identification from aerial images has an accuracy of 80.4% [19]. Accuracy of 91.36% for lane detection in LaneScanNET [22]. Even though these models are relevant, the proposed model does not establish a comparative basis, as the dataset for each is different.

## V. Conclusion

The primary core of the present research is to introduce a novel vision-based system for lane detection and roadside traffic sign recognition that was effectively established. Initially, the Traffic Signs Dataset in YOLO format was gathered, and data preprocessing was done using CLAHE. Then, the model classification is utilized SAAG-NRU model that enables sequential refinement of features and prioritization of the most safety-relevant cues. Afterwards, the Lane Mask and Traffic Sign Model is successfully created. The GUI used Tkinter for visualizing the performance of the Lane and Traffic Sign Detection. The performance efficiency of the proposed model was compared with YOLOv8 existing method using classification performance metrics. The proposed SAAG-NRU model for lane detection achieved 91.11% precision, 92.17% recall, 91.57% F1-score, and 98.80% mAP. The proposed SAAG-NRU model for roadside traffic sign detection attained 96.88% precision, 97.50% recall, 97.14% F1-score, and 98.30% mAP, with 24.96 ms inference and 2.48ms processing time, ensuring highly accurate real-time recognition performance that outperforms the existing method. Therefore, the proposed system delivers fast, accurate, and robust simultaneous lane and traffic sign detection, significantly improving real-time road safety and driver assistance. Performance decreases in heavy rain or on unmarked roads; future work will integrate LiDAR fusion and transformer-based learning for superior reliability.

## References

[1] Alsaleh, A., 2025. Toward a conceptual model to improve the user experience of a sustainable and secure intelligent transport system. *Acta Psychologica*, *255*, p.104892. https://doi.org/10.1016/j.actpsy.2025.104892

[2] Kandasamy, K., Natarajan, Y., Sri Preethaa, K.R. and Ali, A.A.Y., 2024. A robust TrafficSignNet algorithm for enhanced traffic sign recognition in autonomous vehicles under varying light conditions. *Neural Processing Letters*, *56*(5), p.241. https://doi.org/10.1007/s11063-024-11693-y

[3] Samad, A., 2025. Robust image processing framework for real-time detection of road potholes under environmental variability. *Mechatron. Intell Transp. Syst*, *4*(2), pp.92-103. https://doi.org/10.56578/mits040204

[4] Huang, A., Derakhshan, S., Madrid-Carvajal, J., Nosrat Nezami, F., Wächter, M.A., Pipa, G. and König, P., 2024. Enhancing Safety in Autonomous Vehicles: The Impact of Auditory and Visual Warning Signals on Driver Behavior and Situational Awareness. *Vehicles*, *6*(3), pp.1613-1636. https://doi.org/10.3390/vehicles6030076

[5] Munir, F., Azam, S., Jeon, M., Lee, B.G. and Pedrycz, W., 2021. LDNet: End-to-end lane marking detection approach using a dynamic vision sensor. *IEEE Transactions on Intelligent Transportation Systems*, *23*(7), pp.9318-9334. https://doi.org/10.1109/TITS.2021.3102479

[6] Cao, W., Wu, Y., Chakraborty, C., Li, D., Zhao, L. and Ghosh, S.K., 2022. Sustainable and transferable traffic sign recognition for intelligent transportation systems. *IEEE Transactions on Intelligent Transportation Systems*, *24*(12), pp.15784-15794. https://doi.org/10.1109/TITS.2022.3215572

[7] Khalifa, A.A., Alayed, W.M., Elbadawy, H.M. and Sadek, R.A., 2024. Real-time navigation roads: lightweight and efficient convolutional neural network (LE-CNN) for Arabic traffic sign recognition in intelligent transportation systems (ITS). *Applied Sciences*, *14*(9), p.3903. https://doi.org/10.3390/app14093903

[8] Elassy, M., Al-Hattab, M., Takruri, M. and Badawi, S., 2024. Intelligent transportation systems for sustainable smart cities. *Transportation Engineering*, *16*, p.100252. https://doi.org/10.1016/j.treng.2024.100252

[9] Tonec Vrančić, M., Škorput, P. and Vidović, K., 2023. An advanced driver information system at critical points in the multimodal traffic network. *Sustainability*, *16*(1), p.372. https://doi.org/10.3390/su16010372

[10] Sultana, S., Ahmed, B., Paul, M., Islam, M.R. and Ahmad, S., 2023. Vision-based robust lane detection and tracking in challenging

conditions. *IEEE Access*, *11*, pp.67938-67955. https://doi.org/10.1109/ACCESS.2023.3292128

[11] Teng, X., Huang, L., Shen, Z. and Li, W., 2025. Improving intelligent perception and decision optimization of pedestrian crossing scenarios in autonomous driving environments through large visual language models. *Scientific Reports*, *15*(1), p.31283. https://doi.org/10.1038/s41598-025-14827-x

[12] Triki, N., Karray, M. and Ksantini, M., 2023. A real-time traffic sign recognition method using a new attention-based deep convolutional neural network for smart vehicles. *Applied Sciences*, *13*(8), p.4793. https://doi.org/10.3390/app13084793

[13] Lee, S. and Cho, B.H., 2023. Evaluating Pavement Lane Markings in Metropolitan Road Networks with a Vehicle-Mounted Retroreflectometer and AI-Based Image Processing Techniques. *Remote Sensing*, *15*(7), p.1812. https://doi.org/10.3390/rs15071812

[14] Sarvajcz, K., Ari, L. and Menyhart, J., 2024. Ai on the road: nvidia jetson nano-powered computer vision-based system for real-time pedestrian and priority sign detection. *Applied Sciences*, *14*(4), p.1440. https://doi.org/10.3390/app14041440

[15] Shanmugavel, A.B., Ellappan, V., Mahendran, A., Subramanian, M., Lakshmanan, R. and Mazzara, M., 2023. A novel ensemble based reduced overfitting model with convolutional neural network for traffic sign recognition system. *Electronics*, *12*(4), p.926. https://doi.org/10.3390/electronics12040926

[16] Dewangan, D.K. and Sahu, S.P., 2023. Lane detection in intelligent vehicle system using optimal 2-tier deep convolutional neural network. *Multimedia tools and applications*, *82*(5), pp.7293-7317. https://doi.org/10.1007/s11042-022-13425-7

[17] Dewi, C., Chen, R.C., Zhuang, Y.C., Jiang, X. and Yu, H., 2023. Recognizing road surface traffic signs based on YOLO models considering image flips. *Big data and cognitive computing*, *7*(1), p.54. https://doi.org/10.3390/bdcc7010054

[18] Alawaji, K., Hedjar, R. and Zuair, M., 2024. Traffic sign recognition using multi-task deep learning for self-driving vehicles. *Sensors*, *24*(11), p.3282. https://doi.org/10.3390/s24113282

[19] Antwi, R.B., Takyi, S., Kimollo, M., Karaer, A., Ozguven, E.E., Moses, R., Dulebenets, M.A. and Sando, T., 2025. Computer vision-based model for detecting turning lane features on Florida's public roadways from aerial images. *Transportation Planning and Technology*, *48*(5), pp.1113-1144. https://doi.org/10.1080/03081060.2024.2386614

[20] Prakash, A.J. and Sruthy, S., 2024. Enhancing traffic sign recognition (TSR) by classifying deep learning models to promote road safety. *Signal, Image and Video Processing*, *18*(5), pp.4713-4729. https://doi.org/10.1007/s11760-024-03108-1

[21] Javeed, M.A., Ghaffar, M.A., Ashraf, M.A., Zubair, N., Metwally, A.S.M., Tag-Eldin, E.M., Bocchetta, P., Javed, M.S. and Jiang, X., 2023. Lane line detection and object scene segmentation using otsu thresholding and the fast hough transform for intelligent vehicles in complex road conditions. *Electronics*, *12*(5), p.1079. https://doi.org/10.3390/electronics12051079

[22] Perumal, P.S., Wang, Y., Sujasree, M., Tulshain, S., Bhutani, S., Suriyah, M.K. and Raju, V.U.K., 2023. LaneScanNET: A deep-learning approach for simultaneous detection of obstacle-lane states for autonomous driving systems. *Expert Systems with Applications*, *233*, p.120970. https://doi.org/10.1016/j.eswa.2023.120970

[23] Lee, D.H. and Liu, J.L., 2023. End-to-end deep learning of lane detection and path prediction for real-time autonomous driving. *Signal, Image and Video Processing*, *17*(1), pp.199-205. https://doi.org/10.1007/s11760-022-02222-2

[24] Ghanem, S., Kanungo, P., Panda, G., Satapathy, S.C. and Sharma, R., 2023. Lane detection under artificial colored light in tunnels and on highways: an IoT-based framework for smart city infrastructure. *Complex & Intelligent Systems*, *9*(4), pp.3601-3612. https://doi.org/10.1007/s40747-021-00381-2

[25] Yaamini, H.G., Swathi, K.J., Manohar, N. and Kumar, A., 2025. Lane and Traffic Sign Detection for Autonomous Vehicles: Addressing Challenges on Indian Road Conditions. *MethodsX*, *14*, p.103178. https://doi.org/10.1016/j.mex.2025.103178