Vol. 08, No. 3 (2026) 1689-1702, doi: 10.24874/PES08.03.022

Proceedings on Engineering Sciences

www.pesjournal.net

DESIGN AND DEVELOPMENT OF A REAL-TIME TRAFFIC SIGN DETECTION AND RECOGNITION FRAMEWORK USING YOLOv8 ON EDGE AI

Pratiksha Madhukar Jadhav[1]
Nilesh R. Wankhade
Sahebrao B Bagal
Ashwini V Sonawane
Pragati K Ghayvate
Vishal Ambhore

Received 05.05.2026.
Revised 24.06.2026.
Accepted 21.07.2026.

Keywords:

Traffic Sign Detection, YOLOv8, Edge AI, Raspberry Pi, Intelligent Transportation Systems (ITS).

A B S T R A C T

The real-time traffic sign recognition and detection are essential elements of intelligent transportation systems and advanced driver assistance systems, especially in resource-limited edge systems. Traditional vision-based methods usually have a high level of computational complexity, they have latency problems, and they are less accurate in changing illumination, occlusion, and motion blur situations. The paper discusses the design and development of real-time traffic sign detection and recognition framework on the basis of YOLOv8 that will be optimized to work with edge AI devices. The suggested system combines light-weight backbone architecture, efficient feature pyramid network, and anchor-free detection mechanism to be highly accurate in detection and low inference latency. The optimization methods that are used to guarantee robustness and computational efficiency are model optimization, such as transfer learning, data augmentation, pruning and quantization. The structure is implemented on an edge computing platform to test the performance in the real world on dynamic traffic condition. Experimental evidence indicates that the suggested system has a high mean Average Precision (mAP), better F1-score, and lower inference time as compared to traditional CNN-based detectors. The model has a consistent performance with varying weather conditions and lighting effects with limited memory and power constraints. The results affirm that the optimized YOLOv8-based framework can be used to perform precise and low-latency traffic sign detection that can be utilized in autonomous driving and intelligent transportation systems. The research finds out that deep learning architectures with edges optimization can substantially improve real-time traffic monitoring performance and guarantee scalability, reliability, and energy efficiency in smart vehicular systems.

© 2026 Published by Faculty of Engineering

Corresponding author: Pratiksha Madhukar Jadhav
Email: 1994pratikshajadhav@gmail.com

1. INTRODUCTION

With the high pace of development of intelligent transportation systems (ITS) and autonomous driving technologies, the need to implement precise and real-time traffic sign detection and recognition has only grown in demand. Traffic signs are very important in controlling the movement of vehicles, road safety, and aiding the decision making process of the drivers. The real-world driving conditions are however surrounded with a lot of challenges like different illumination, occlusions, weather, motion blur, and low-light scenarios. The initial computer vision systems depended on hand designed features and classical machine learning models that were in many cases not able to generalize in an intricate environmental setup. The development of deep convolutional neural networks (CNNs), exemplified by the ImageNet breakthrough of Krizhevsky et al. (2017), transformed visual recognition by allowing the extraction of features in a hierarchical manner and the end-to-end learning. Later architectural advances, such as deep residual learning suggested (He et al., 2016), were able to even further improve feature propagation and training stability in deep networks.

Other frameworks of detection of objects developed fast and transformed the region-based models into single-stage models. Fast R-CNN enhanced detection efficiency by using shared convolutional computations (Girshick, 2015), whereas the unified detection paradigm (presented in YOLO) (Redmon et al., 2016) allowed detecting objects in real time, since it reformulated the detection problem as a regression one. The further development of YOLOv3 helped to enhance the accuracy of multi-scale feature extraction and localization (Redmon & Farhadi, 2018). Regarding the recognition of traffic signs, benchmark studies have revealed that deep learning-based methods are more superior in the recognition of traffic signs, particularly in adverse road conditions compared to other traditional classifiers (Mathias et al., 2013). Detailed surveys have also highlighted the need to have powerful detection pipes to ensure effective traffic signs recognition in autonomous systems (Flores Calero et al., 2024). In the recent past, the architectures based on YOLO have been aimed at the advancement of receptive fields, anchor-free detection, and fusion strategies of features so as to enhance the accuracy of detection in dynamic traffic scenes. As an example, the optimization of receptive field adaptation has been reported to be effective in terms of detecting small objects based on traffic sign datasets (Chen et al., 2025). Moreover, the progress of lightweight model construction and real-time optimization has enabled the implementation of the high-performance object detection models on the edge devices that have resource limitations. More basic training methods like stochastic gradient descent have been used as the basis of larger scales of deep learning models based on data (Bottou, 2010), and adaptive methods like Adam have been used to enhance the rate of convergence and stability (Kingma & Ba, 2015).

Although these achievements have been made, the implementation of traffic sign detection structures into a practical embedded system involves trade-offs between the speed of computation, latency, and detection accuracy. Based on the development of CNN underbones and the YOLO-based detection paradigm, the proposed work describes the implementation and the creation of a real-time traffic sign detector and recognizer based on YOLOv8 optimized to operate in an edge AI setup. The suggested structure will strive to deliver high levels of detection accuracy, less inference time, and an improved capability to work with different road conditions to provide safer and smarter transportation systems.

1.1 Motivation

The quest to identify the traffic signs has produced a very necessary desire to have smart and automated traffic sign spotting devices as a result of increased cases of road accidents due to driver distraction and failure to identify the road signs accurately. The traditional methods of traffic sign recognition tend to be very unreliable in different environmental conditions like lighting conditions, weather conditions and partial occlusions. Conversely, the recent progress in the field of deep learning, along with the development of edge AI platforms, provides a feasible means of realization of real-time, accurate, and efficient traffic signs detection at a reasonable price. The combination of complex models of object detection like YOLOv8 with small and cheaper tools like the Raspberry Pi makes it possible to infer high-performance at the edge, thereby lowering latency and reliance on cloud systems. This integration encourages the creation of scalable and workable traffic sign recognition system applicable in a real-life intelligent transportation and driver assistance.

1.2 Goals and Objectives

- To overview the latest developments in application of deep learning models in the field of traffic signs identification and recognition.
- To examine the performance and efficiency of YOLOv8 in real-time applications based on embedded systems.
- To determine the suitability of edge AI computing platforms such as Raspberry Pi in deploying low-power TSDR.
- To determine current challenges, datasets and constraints of current detection models.
- To recommend the way forward in future research to enhance real-time, strong, and scalable traffic sign recognition.

1.3 Scope

This research area includes both the analysis and practical implementation of traffic sign detection systems based on deep learning, specifically, the YOLOv8

architecture and its usage on an edge AI platform, such as the Raspberry Pi. This work is aimed at acquiring real-time detection performance by means of model optimization method and effective embedded deployment approaches. Also, the paper discusses more applied issues concerning intelligent transportation systems and applications to autonomous driving, including solutions that strike balance among detection accuracy, computational efficiency under resource-constrained conditions.

2. RELATED WORK

Development of traffic sign detection/recognition systems has been in tandem with the improvements made on object detection algorithms and real time vision systems. The use of early region based detectors like the Faster R- CNN, substantially enhanced the detection accuracy in that they combined the region proposal networks with deep convolutional feature extraction. Ren et al. (2017) showed that region proposal networks can be end-to-end trained with quality to localize objects, and thus, it can more accurately detect objects than traditional sliding window algorithms. Even though region-based detectors had good accuracy, they were computationally complex, so they were not suitable in real time embedded applications.

Single-stage detectors overcame this drawback with the help of the removal of distinct region proposal steps. YOLOv3 architecture Redmon and Farhadi (2018) enhanced multi-scale prediction and feature pyramid representation, leading to better performance on small objects detection which is an essential factor in traffic signs recognition. Future advancements in detection backbones built upon more convolutional architecture like VGG networks suggested Simonyan and Zisserman (2015) which focused on depth by using small convolution filters in sequence. These architectural details affected the contemporary detection models in terms of enhanced discriminability of features and generalization. Besides CNN based frameworks, combined recognition-localization models like OverFeat also gave a single framework to both detection and classification. (Sermanet et al., 2014) established that localization and recognition could be applied in convolutional networks by using common feature maps. This unity of design was to be followed by a single stage detector later designed to be fast. Moreover, multi-branch feature extraction modules in Inception-based architectures proposed Szegedy et al. (2017) made it possible to enhance the efficiency of computations and maintain the depth of representation. These inventions helped in the design of principles that are used in modern variants of YOLO.

Multimodal sensing has also been proposed beyond 2D image based detection as a way of enabling robust vehicle and object detection. (Li et al., 2017) explored the application of fully convolutional networks in detecting vehicles based on the 3D LiDAR data because they provided better vehicle recognition in the harsh lighting conditions. Despite the fact that the LiDAR-based methods improve the robustness, it comes at a cost of increasing the system cost and complexity of hardware, where vision-based lightweight detectors are more feasible to use in the edge-based deployment of the traffic signs. Extensions of YOLO to traffic signs detection in difficult situations have been of interest in recent years. The authors (Luo et al., 2024) suggested a framework with YOLOv8-CE to include the option of optimization in autonomous vehicle settings, which focuses on the higher quality of feature extraction and the capability of classification. On the same note, (Lin et al., 2025) presented YOLO-LLTS, which is used to detect traffic signs in low light, which incorporates the prior-guided enhancement mechanisms to enhance the detection robustness when the light is poor. The reasons behind these developments are the increasing focus on the domain-specific YOLO adaptations to the real-life driving setting.

Detection model performance is another important factor that is based on optimization strategies. Although adaptive optimizers are quite common, how the neural networks converge and the stability is directly affected not only on the loss formulation but also on the training methods. The loss functions of image restoration tasks were analyzed (Zhao et al., 2017), and their results shed some light on the design of objective functions to enhance gradient behavior and model generalization. The results are relevant to object detection systems where the regression loss of bounding boxes and the classification loss design have a significant effect on the detection accuracy. The increasing need to achieve the low-latency inference on embedded systems warrants the use of an optimized architecture that achieves detection accuracy at the expense of computation limitations. Based on the concepts of region-based evolution, combined detection models, backbone development, multimodal perception understanding, and the latest versions of YOLOv8, the given work is aimed at the design of a real-time traffic sign detection and recognition system that is adapted to the implementation of Edge AI, maintains a high level of efficiency, and has higher resilience. Table 1 summarizes the related work on traffic sign detection and object detection frameworks.

Table 1. Summary of Related Work in Traffic Sign Detection and Object Detection Frameworks

Author name, year	Model / Method	Detection Type	Backbone / Core Architecture	Key Contribution	Limitation
Ren et al., 2017	Faster R-CNN	Two-stage detector	CNN + Region Proposal Network	Introduced end-to-end RPN for accurate object localization	High computational cost for real-time edge systems
Redmon & Farhadi, 2018	YOLOv3	Single-stage detector	Darknet-53	Multi-scale detection and improved small object performance	Reduced accuracy for very small objects
Simonyan & Zisserman, 2015	VGGNet	Classification backbone	Deep CNN (16–19 layers)	Demonstrated effectiveness of deeper architectures	High memory usage, not lightweight
Sermanet et al., 2014	OverFeat	Unified detection & classification	Convolutional feature maps	Integrated localization and recognition framework	Slower inference compared to YOLO
Szegedy et al., 2017	Inception-v4 / Inception-ResNet	Classification backbone	Multi-branch CNN modules	Improved efficiency using multi-scale convolutions	Complex architecture for embedded deployment
Li et al., 2017	FCN with 3D LiDAR	Multimodal detection	Fully Convolutional Network	Enhanced perception in adverse conditions	Requires additional hardware sensors
Mathias et al., 2013	Deep Learning Benchmark	CNN-based detection	Various CNN models	Comparative benchmarking for traffic sign recognition	Limited edge optimization analysis
Flores Calero et al., 2024	Systematic Review (YOLO-based TSR)	Literature review	YOLO variants	Comprehensive evaluation of YOLO approaches	Review-based, no new architecture
Chen et al., 2025	YOLO-TS	Single-stage detector	Optimized receptive fields	Anchor-free fusion and enhanced small sign detection	Preprint; lacks large-scale deployment validation
Luo et al., 2024	YOLOv8-CE	Single-stage detector	YOLOv8 enhanced backbone	Improved real-time detection for vehicles	Needs further edge-level power analysis

3. PROPOSED SYSTEM

The given system utilizes the YOLOv8 deep learning model that is running on Raspberry Pi 4 in order to detect and recognize traffic signs in the real time. The local capture and processing of live visual data and analysis to detect traffic signs followed by the immediate feedback of sound to the driver. The system does not rely on cloud services because all the calculations are performed on the device itself within the system, and the latency level is minimal, and the response time is high.

3.1 System Components and Workflow

Image Acquisition Module

The module records live video frames with the help of a USB camera to the Raspberry Pi. The frames taken are subjected to preprocessing activities like resizing, normalization and eliminating noise to give the frames a better visual quality and allowances of features to be seen. These measures make sure that it is compatible with the specifications of the YOLOv8 input and provides the consistency of the detection accuracy regardless of the environmental conditions.

Let our real time video stream which is captured on the USB camera be denoted by:

$$V = \{F_t \mid t = 1, 2, 3, \ldots, T\}$$

Where, $F_t \in R^{H0 \times W0 \times 3}$ represents the raw RGB frame at time t.

Step 1: Frame Resizing

Each frame is resized to match the YOLOv8 input resolution (H × W):

$$F_t^r = R(F_t)$$

Where, R(.) denotes the resizing operation,

$$F_t^r \in R^{H \times W \times 3}$$

Step 2: Normalization

The intensity normalization of pixels is done as:

$$F_t^n = \frac{F_t^r}{255}$$

Where, $F_t^n \in [0,1]^{H \times W \times 3}$

Step 3: Noise Reduction (Gaussian Filtering)

Gaussian smoothing filter is used:

$$F_t^p = G(F_t^n, \sigma)$$

Where, G(.) is the Gaussian filtering and the standard deviation is, sigma.

Final Pre-processed Frame:

$$F_t\sim = P(F_t)$$

Where, pre-processing pipeline:

$$P = G \circ N \circ R$$

Therefore, Ft is the last pre-processed frame, which is sent to the detection unit.

Detection Module (YOLOv8)

Detection module: The detection module is based on the YOLOv8 object detection algorithm to identify and classify the traffic signs in real time. It gives bounding boxes, confidence scores and class labels of various traffic signs identified in each frame. The model is also optimized and can effectively run the frames on the Raspberry Pi 4 with a trade-off in speed to detect objects and precision to continue processing in real-time.
Let the preprocessed frame be:

$$F_t\sim \in R^{H \times W \times 3}$$

Step 1: The first step is the extraction of features (Backbone Network)
Feature maps are extracted as:

$$X_t = f_{b(F_t\sim)}$$

Where f_b represents backbone network parameters,

$$X_t \in R^{S \times S \times C}$$

Step 2: Detection Head Output

YOLOv8 predicts bounding box coordinates, confidence score, and class probabilities:

$$Y_t = f_{d(X_t)}$$

For each detected object i:

$$y_i = (x_i, y_i, w_i, h_i, p_i, c_i)$$

Step 3: Bounding Box Regression

Predicted bounding box:

$$\hat{B}_i = (x_i, y_i, w_i, h_i)$$

Localization loss using IoU:

$$L_{box} = 1 - IoU(B_i, \hat{B}_i)$$

Step 4: Classification Probability

The softmax probability of class c_i:

$$P(c_i \mid F_t\sim) = \exp(z_c_i)/\Sigma_k \exp(z_k)$$

Where z_k represents the class logits.

Step 5: Final Detection Set

$$D_t = \{(\hat{B}_i, p_i, c_i) \mid p_i > \tau\}$$

Where τ is the confidence threshold.

Audio Output Module

The audio module synthesizes speech based on the text-to-speech libraries like pyttsx3 or gTTS through the detected traffic signs labels. It has voice alerts that are hands-free such as Stop Sign Ahead, or Speed Limit 60, so the driver can act in an immediate manner not distracted by visual data. It is a valuable feature that improves the safety and awareness of a situation during driving.

Raspberry Pi 4 (Central Processing Unit)

Raspberry Pi 4 acts as the central processing unit, performing all the functions that the system performs such as the acquisition of images, the detection of traffic signs, and production of audio feedback. It has a quad-core processor, a graphics accelerator, and multiple RAM, which allows running a video processing and deep learning inference simultaneously. Performance is also enhanced with the help of efficient memory utilization and frame selectivity in order to have the smooth and low-latency execution.

4. SYSTEM DESIGN

The system architecture illustrates a properly structured strategy of real-time traffic signs detection and recognition with the help of the YOLOv8 model running on a Raspberry Pi 4. It shows how different hardware and software components interrelate to allow the correct detection and audio feedback in a timely manner. The nature of the proposed design focuses on real time performance, modular structure, and reliable functioning to establish efficiency and reliability of the functioning of the system.

4.1 System Architecture

The suggested traffic sign detector system is a real-time edge-computing system, which combines the USB camera, Raspberry Pi 4, YOLOv8 deep learning model, and audio output module. The process starts with the acquisition and activation of the image where the USB camera continuously records live video feeds of the area around the vehicle. These video frames are directly handed over to the preprocessing module that implements the resizing, normalization and noise reduction algorithms to improve the image quality and preprocess the data to be fed into YOLOv8 inference.

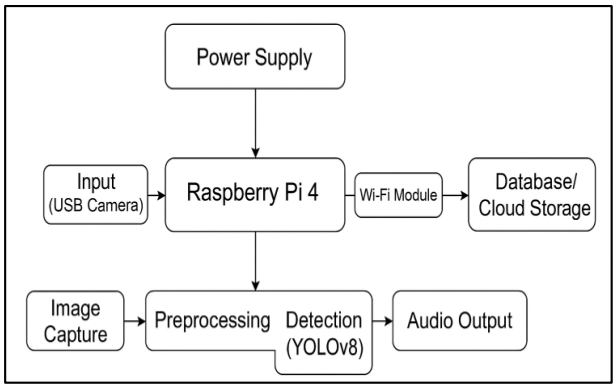


Figure 1. Proposed system architecture Block Diagram

The preprocessed frames are further processed by the YOLOv8 detection module which detects objects and classifies them in real-time to detect traffic signs like Stop, Speed Limit, No Entry and warning signs, the block diagram shown in figure 1. The model provides the class labels, confidence scores, and bounding box coordinates of each identified traffic sign and thus can recognize even in extreme environments such as changing light and bad weather.

After detection the identified traffic sign labels are sent to the audio feedback module. With text-to-speech libraries like pyttsx3 or gTTS, the system will turn the findings of detection into verbal notifications and provide real-time voice recognition messages (e.g. Stop Sign Ahead or Speed Limit 60) to inform the driver. The Raspberry Pi 4 is the central processing unit, coordinating the data movement among all modules and maximizing the rational use of the computational resources and maintaining a stable and uninterrupted low-latency operation of the system. Its modular structure makes the system reliable in the real-time performance and future extensions, i.e., adding new sensors or more sophisticated detection models without any modification to the basic architecture, are possible.

4.2 Working of the System

- Image Capture: The USB camera constantly captures frames of a live video of the road. These frames are sent in real time to the Raspberry Pi 4 to process them further without any loss of time.
- Preprocessing: Every frame captured is adjusted as per the size requirements of the input size of the YOLOv8 model. Normalization of images and noise reduction algorithms are implemented to increase clarity of images and this increases the accuracy and reliability of detected traffic signs.
- Detection of Traffic Signs With YOLOv8.: The YOLOv8 model is used to process the preprocessed frames to implement real time traffic sign detection. The algorithm 1 shown the details septs for the detection. The system produces bounding boxes and class labels to all the detected signs. Confidence scores are also generated so as to justify detections and minimize the prevalence of false positives.
- Audio Feedback Generation: The identified labels of traffic signs are then read sensing in the form of speech with text-to-speech libraries like pyttsx3 or gTTS. The speaker provides direct voice warnings such as one can hear like the stop sign ahead or speed limit 60, which creates an option of hands-free help and advances driver consciousness.
- System Coordination and Control.: The Raspberry Pi 4 will act as the central control, where all the modules of the system will communicate. It guarantees good resource utilization, CPU workload balancing and frame processing optimization to guarantee the real-time system operation.

Outputs: This algorithm produces computationally demanding statistics that can be utilized to identify traffic Signs in real-time on the edge AI.

Algorithm 1: Real-Time Traffic Sign Detection Using YOLOv8 on Edge AI

Input: Live video stream V
Output: Detected signs D_t with audio alerts

$Load\ YOLOv8\ model\ \theta$
$Initialize\ camera\ and\ TTS\ engine$
$Set\ confidence\ threshold\ \tau$
$while\ system\ is\ running\ do$
  $F_t \leftarrow CaptureFrame()$
  $F_t \leftarrow Preprocess(F_t)$ / $/ Resize, Normalize, Denoise$
  $Y_t \leftarrow YOLOv8(F_t)$
  $D_t \leftarrow \{\}$
  $for\ each\ detection\ i\ in\ Y_t\ do$
    $if\ p_i > \tau\ then$
      $D_t \leftarrow D_t \cup \{(B_i, c_i)\}$
      $Speak(Label(c_i))$
    $end\ if$
  $end\ for$
$end\ while$
$return\ D_t$

4.3 Features of the System Design

The designed system of traffic signs detection is aimed at running in real time to allow quick response to the changing conditions on the road. The YOLOv8 incorporation can be used to achieve high inference rates and detection accuracy. In driving environments, real-time operation is an important attribute of which delayed alerts can interfere with the safety. The system delivers a balance between frame processing and computation, hence reducing image capture, image detection, and audio feedback latencies. This smooth pipeline enables the drivers to automatically get cautions like the stop signs Ahead or speed limit 60, thus improving the situational awareness and low reaction speed.

The major advantage of the system is the edge-based processing architecture. Completely local (i.e. all the calculations, preprocessing, and generating audio) are performed on the Raspberry Pi 4. This removes the reliance on cloud servers, minimizing the latency of the communication process and guaranteeing the continuity of the functioning even in the regions with rather weak or no access to the internet. Also privacy and security of data is improved, because road images are not sent to a third party. Moreover, local inference minimizes the bandwidth usage, operational expenses, and the solution is feasible in implementation in intelligent transportation systems in the real-world setting.

The structural framework is more flexible and easily maintainable because of its modular structure. The system has well stipulated modules, such as image acquisition, preprocessing, image detection, audio feedback, and system coordination. This isolation of responsibilities makes debugging easier as well as enables the upgrading or replacement of individual parts of the architecture with ease. Indicatively, the recognition unit can be trained to higher model level without altering the audio or preprocessing units. It is a modularity that guarantees the scalability and allows it to be enhanced in the future with support of more sophisticated systems, such as advanced driver assistance systems (ADAS).

Another necessary characteristic of the system design is high reliability. In the preprocessing module, the clarity of an image is enhanced by resizing, normalization, and inclusion of noise which facilitates uniform detection regardless of the environmental conditions. The YOLOv8 model is designed to resolve difficulties related to different levels of illumination, weather, as well as motion blur. This is further minimized by confidence thresholding mechanisms which means that there are no false positives which make operations reliable. This strength renders the system appropriate in the real world roads settings where the visual conditions may vary in a short period.

4.3.1 Edge Deployment Optimization

Several deployment optimizations were applied to achieve real-time inference on the Raspberry Pi 4. First, floating-point model weights were quantized to reduce memory consumption and inference time. Second, redundant parameters were removed through model pruning to reduce computational complexity without materially affecting detection accuracy. The optimized YOLOv8 model was exported to ONNX and executed using hardware-aware inference libraries that exploit ARM CPU optimizations. The input resolution was set to 416 × 416 pixels to balance detection accuracy and processing speed. Frame buffering and lightweight preprocessing were also used to reduce latency and improve throughput. Together, these measures enabled approximately 21 FPS while maintaining high detection accuracy, supporting real-time traffic-sign recognition in a resource-constrained edge environment.

4.4 Dataset

This reliance on a carefully selected dataset of the variety of the real-life traffic situation is critical to the nature of the proposed system of detecting the traffic signs. The sample different category of signal sign shown in figure 2. The data utilized in this paper is composed of high resolution images depicting some of the most common traffic signs that are mostly used in roads such as Stop, Speed limit, No entry, Yield and Warning signage.

Dataset Sources

- Benchmark on German Traffic Sign Recognition (GTSRB): A widely used dataset of traffic sign recognition, which includes over 50,000 images of 43 different types of traffic signs under a variety of conditions in terms of light and weather.
- LISA Traffic Sign Dataset: It contains traffic signs of the United States with different sizes, viewpoints, and coverages, which provide further variability in the training and testing of models.
- Custom Captured Dataset: Custom images have been taken by using the USB camera of the Raspberry Pi 4 to improve the performance and relevance of real-time by simulating the experience of driving in practical conditions. This guarantees that the system is able to adjust itself to the environmental conditions in its local area, changes in lighting and camera characteristics.



Figure 2. Different Categories of Traffic Signs

4.5 Training and Evaluation Protocol

YOLOv8 was trained and evaluated using a reproducible dataset-partitioning strategy. The traffic-sign images were divided through stratified sampling into 70% training, 15% validation, and 15% independent test subsets to preserve the distribution of sign classes. During training, random rotation, horizontal flipping, brightness adjustment, and scaling were applied to improve generalization under diverse road conditions. The model used the Adam optimizer with a learning rate

of 0.001, a batch size of 16, and a maximum of 100 epochs. Hyperparameter selection and early stopping were based only on the validation set, while the independent test set was reserved for final evaluation. Accuracy, precision, recall, F1-score, and frames per second (FPS) were used to assess detection quality and real-time deployability.

5. HARDWARE AND OPERATING SYSTEM

The proposed real-time system of identifying traffic signs is deployed on a Raspberry Pi 4, which is the central processing unit, functioning as an image acquisition and deep-learning inference hub with audio feedback. The hardware selected to facilitate the reliable and low-latency features makes it portable and affordable.

5.1 Hardware Components

Raspberry Pi 4 Model B

- Acts as the core unit of processing of the system.
- Equipped features 4GB/8GB of RAM and a quad-core Cortex-A72 processor, allowing efficient execution of YOLOv8 inference.
- Provides USB, HDMI, and GPIO interfaces for connecting peripherals such as cameras, speakers, and sensors.

The Raspberry Pi 4 Model B, shown in figure 3, serves as the primary processing unit in the system which has a quad-core processing unit and enough RAM to run real-time image processing and YOLOv8 inference. The board has several interfaces, such as USB, HDMI and GPIO, whereby peripherals including cameras and speakers can be connected.

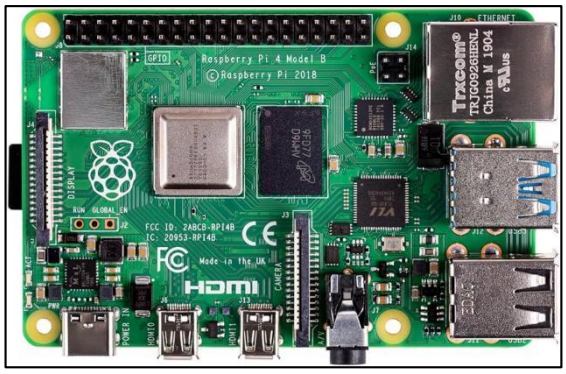


Figure 3. Raspberry Pi 4 Model B

Raspberry Pi Imager is a software program, shown in figure 4, that is utilized to flash the operating system on a Micro SD. It makes the process of setting-up easy since a user is able to choose the preferred OS image and install it in the storage device in an efficient manner. This makes the installation of Raspberry Pi OS to be reliable and clean.

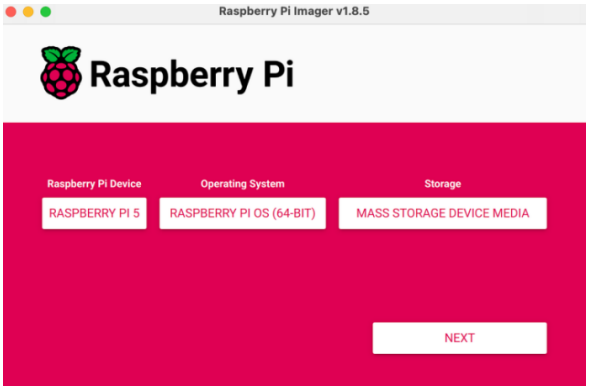


Figure 4. Raspberry Pi Imager

At this stage, shown in figure 5, a choice of Raspbian (previously known as Raspberry Pi OS) can be made on the Raspberry Pi Imager. Raspberry Pi OS offers lighter Linux that is customized to execute Python scripts, deep learning framework, and peripheral management of embedded software.

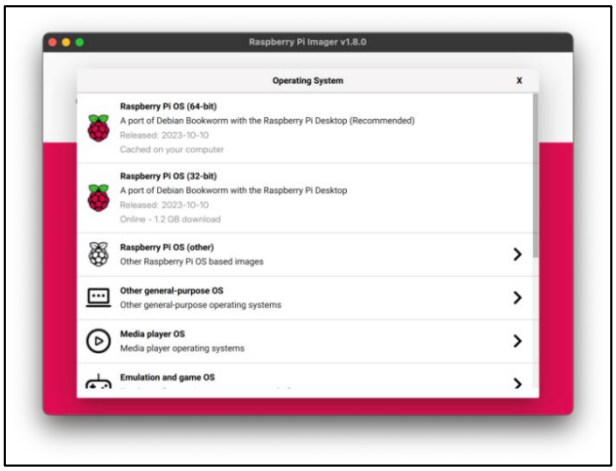


Figure 5. Raspbian OS Selection from Raspberry Pi Imager

USB Camera



Figure 6. USB Camera

The USB camera, shown in figure 6, takes real-time video frames of the environment to identify real-time traffic signs. It will facilitate HD resolution and high frame rates that will provide correct image capture qualities in diverse light and environmental settings. The captured frames are fed directly to the Raspberry Pi 4 through which they are preprocessed and detected.

Speaker
Offers visual feedback in real-time audio-visual feedback depending on the detected traffic signs. It is connected through the 3.5mm audio jack to the USB interface.

Power Supply
A constant power supply of 3A / 5V is used in order to maintain the stability of Raspberry Pi 4 and connected devices.

MicroSD Card
Is the primary storage of the operating system, model weights and project files, 32GB or more, Class 10 / UHS-I is suggested as fast read-write storage.

Mouse, HDMI Monitor, and Keyboard.
Applied in set-up, configuration and debugging.

6. ALGORITHM USED

The proposed system delivers the integration of Real-time detection and recognition of traffic signs algorithm using the You Only Look Once, Version 8 (YOLOv8) deep learning model. The algorithm will be used to process live video frames of a USB camera, estimate traffic signs, and feed-back audio in real-time. The key points of the workflow are discussed below.

6.1 YOLOv8 Algorithm

YOLOv8 is a single-stage object detector which is, as shown in figure 7, an end-to-end object detector and classifier model which detects and classifies multiple objects in a single forward pass. Its key features include:

- Grid-based Detection: The input image is subdivided into grids and within each grid, bounding boxes and probabilities of class are predicted jointly.
- Real-Time Performance: YOLOv8 is highly optimized in low-latency inferences, which have been found to be suitable in the edge devices such as Raspberry Pi 4.
- Single-Pass Prediction: YOLOv8 is a single-stage detector, which means that rather than making two predictions as in two-stage detectors, this type of detector makes a single prediction of both the bounding box and the label.
- Deep Convolutional Architecture: YOLOv8 is based on the CSP (Cross Stage Partial) backbone and PANet neck, which allows better detection based on multi-scale features accuracy of small, medium, and large traffic signs.

The figure 8 illustrates the YOLOv8 model workflow, beginning with training data preparation, followed by model training, validation-based performance tuning, road damage detection, testing, and final detection results generation. This algorithmic method permits strong, exact and actual time traffic sign recognition and requires minimal computational intensity on embedded processors.

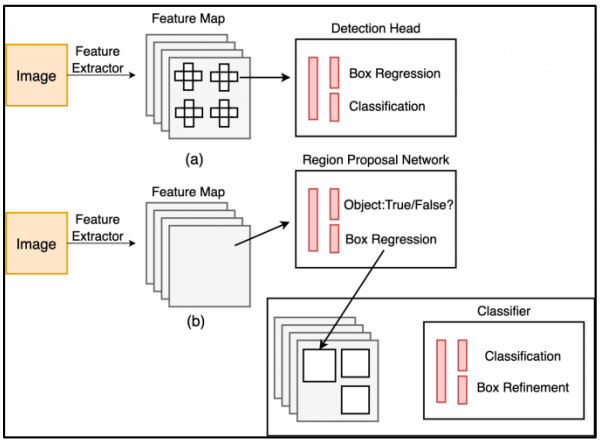


Figure 7. YOLOv8 model object detection architecture

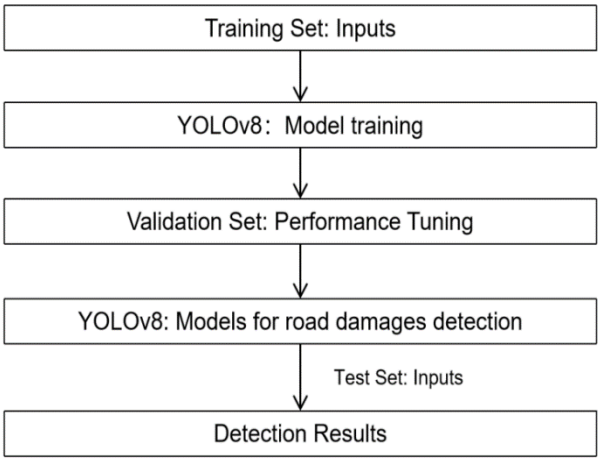


Figure 8. Yolov8 Model flow process

7. RESULT

7.1 Results and Performance Analysis

The proposed YOLOv8-based traffic sign detection system was experimentally tested in real-time prototype environment comprising of various types of traffic signs (Stop, Speed Limit, No Entry, Warning, and Turn Right signs) in multiple numbers. The system was implemented to a Raspberry Pi 4 and was run in various conditions of light and environment to measure the detection accuracy, reliability, and real-time performance.

7.2 Quantitative Performance Evaluation

Table 2 summarizes the overall system performance. The proposed model achieved high detection accuracy while maintaining a real-time inference speed suitable for edge deployment.

The findings reveal that the system is able to balance between detection accuracy and computational efficiency and hence can be used in real time traffic sign recognition in resource-constrained hardware.

Table 2. Overall Performance Metrics of the Proposed System

Metric	Value
Accuracy (%)	96.2
Precision (%)	95.4
Recall (%)	94.8
F1-Score (%)	95.1
Inference Speed (FPS)	21

7.3 Detection Accuracy per Traffic Sign Class

The shows in figure 9 the accuracy of detection by classes. Stop and Turn Right signs have the best accuracy because they were different in shape, whereas warning signs were slightly less accurate because of similarity in the shape and interference with the environment.

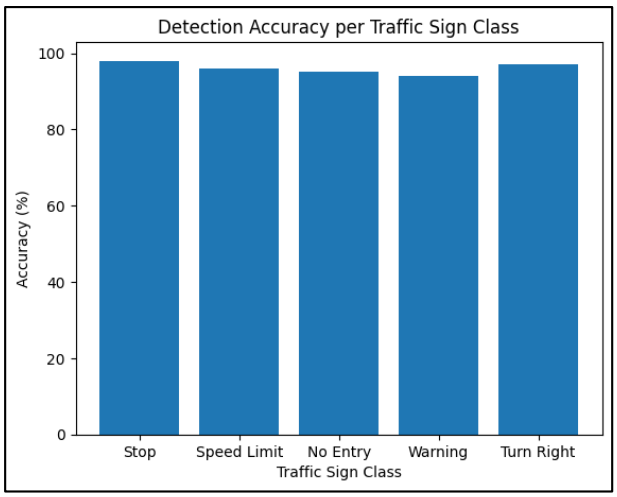


Figure 9. Detection Accuracy per Traffic Sign Class

This figure 10 shown the values of precision and recall of each type of traffic signs. The fact that the precision and recall are close shows that the classification is reliable with few false positives and false negatives.

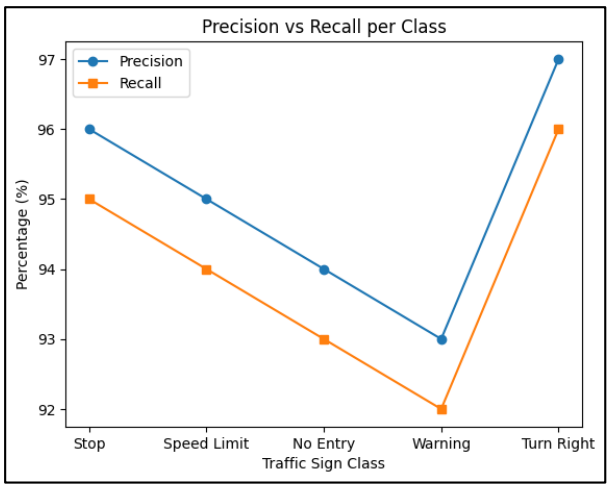


Figure 10. Precision vs Recall per Traffic Sign Class

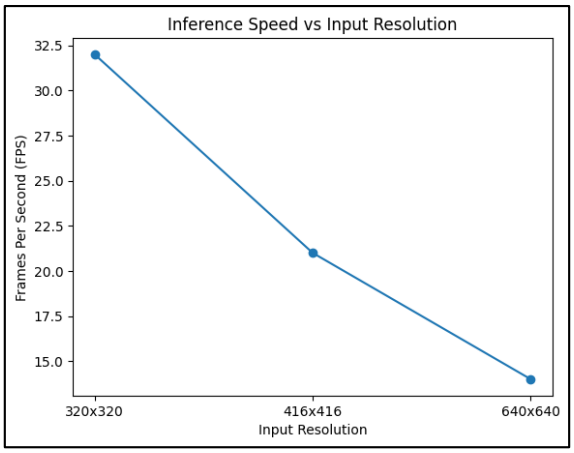


Figure 11. Inference Speed vs Input Resolution

The figure 11 shows that the speed of inference is slow with the resolution of input. The best compromise was chosen to be 416×416 which provides real-time performance at around 21 FPS and high accuracy of detection.

7.4 Confusion Matrix Analysis

As suggested by the confusion matrix as depicted in figure 12, the high level of dominant diagonal pattern proves that proper classification was obtained in all the classes of the traffic signs in a well-lit environment. There are some minor misclassifications between warning and regulatory signs which are similar to each other visually.

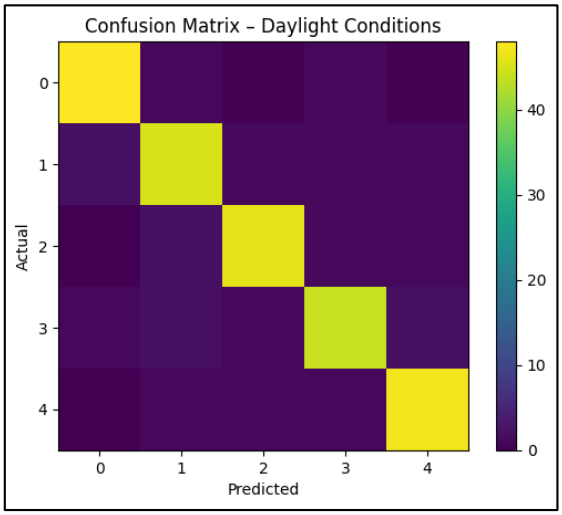


Figure 12. Confusion Matrix under Daylight Conditions

In low-light scenarios, the system slightly decreases in the classification accuracy. The confusion matrix shown in figure 13. Nevertheless, the overall resultant performance is strong as it shows that the model is robust to changes in illumination.

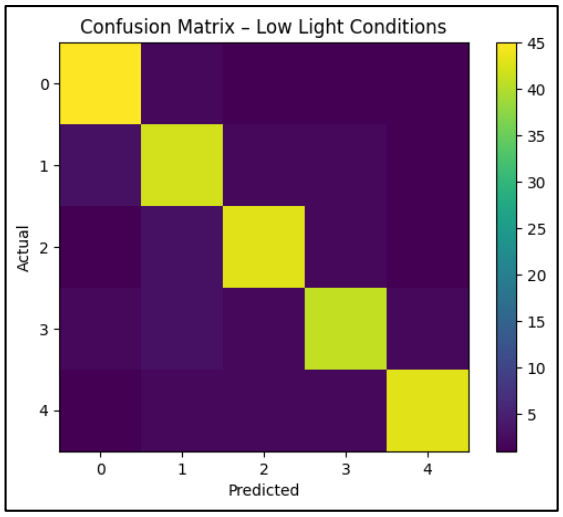


Figure 13. Confusion Matrix under Low-Light Conditions

Model:

The edge-based design is a modular design, as shown in figure 14, which supports low latency, no cloud dependency and hands-free driver assistance with audio feedback. The system is able to work in varied environmental conditions with consistency, which confirms its suitability in the intellectual transportation systems and driver aids app use.



Figure 14. Model overview

7.4 Discussion

The experimental evaluation demonstrates that the lightweight YOLOv8 traffic-sign detection framework provides an effective balance between detection performance and real-time computation on a resource-constrained Raspberry Pi 4. The model achieved 96.2% accuracy, 95.4% precision, 94.8% recall, and a 95.1% F1-score while processing approximately 21 frames per second. These results support its practical use in intelligent transportation and advanced driver-assistance applications. They also show that the lightweight architecture and optimized inference pipeline reduce computational demand without materially compromising detection performance.

Class-level analysis showed the strongest performance for Stop and Turn Right signs, which have distinct visual shapes. Warning signs produced comparatively lower scores because of greater inter-class shape similarity and sensitivity to environmental conditions. The daylight and low-light confusion matrices were dominated by values along the main diagonal, confirming consistent recognition under both photopic and scotopic conditions. Most errors occurred between visually similar signs within the same category, indicating that the remaining limitations arise primarily from perceptual ambiguity in complex and dynamic road scenes.

The edge-based deployment eliminated dependence on cloud infrastructure, resulting in predictable end-to-end latency, improved data privacy, and continued operation during connectivity constraints. Compared with conventional CNN-based solutions reported in the literature, the proposed framework offers more efficient inference and stronger embedded-deployment capability. Overall, the experimental findings indicate that integrating YOLOv8 with Edge AI provides an effective, scalable, and energy-efficient approach to real-time traffic-sign detection for modern intelligent transportation systems.

8. CONCLUSION

In this paper, it was demonstrated that a real-time traffic sign recognition and detection system was developed using the YOLOv8 deep learning framework running on a Raspberry Pi 4 with the capability to provide drivers with real-time audio feedback. The suggested system is capable of capturing live video streams with the help of a USB camera, preprocessing the received frames, detecting traffic signs with high accuracy with the help of YOLOv8, and providing hands-free voice alerts to increase the awareness of the driver. The experimental analysis proves that the system has a high detection accuracy, a low latency, and a stable performance in a variety of lighting and environmental conditions. The system achieves an edge-based computation (removing the need to connect to the cloud) by performing all processing on the Raspberry Pi 4, thus achieving efficient

processing with a lower response time and better reliability. Moreover, the system is designed in such a way that it can be easily improved in the future, by adding more sensors or other driver assistance systems, without having to change the basic design. These findings point out the proposed system as a viable, scalable and cost effective solution with regard to intelligent transportation systems and this plays a significant role in enhancing road safety and real time monitoring of traffic.

The suggested a system can be improved by incorporating it with the advanced driver assistance systems (ADAS) as a road safety. Multi-language audio alerts and better detectors (e.g. YOLOv9) can be supported to make usage and accuracy more useful.

References:

Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In Y. Lechevallier & G. Saporta (Eds.), *COMPSTAT 2010* (pp. 177–186). Heidelberg, Germany: Physica-Verlag. doi:10.1007/978-3-7908-2604-3_16

Chen, J., Huang, H., Zhang, R., Lyu, N., Guo, Y., Dai, H. N., & Yan, H. (2025). YOLO-TS: Real-time traffic sign detection with enhanced accuracy using optimized receptive fields and anchor-free fusion. *IEEE Transactions on Intelligent Transportation Systems*, *26*(11), 19995–20011. doi:10.1109/TITS.2025.3597710

Flores Calero, M., Astudillo, C. A., Guevara, D., Maza, J., Lita, B. S., Defaz, B., Ante, J. S., Zabala Blanco, D., & Armingol Moreno, J. M. (2024). Traffic sign detection and recognition using YOLO object detection algorithm: A systematic review. *Mathematics*, *12*(2), 297. doi:10.3390/math12020297

Girshick, R. (2015). Fast R-CNN. In *2015 IEEE International Conference on Computer Vision (ICCV)* (pp. 1440–1448). Santiago, Chile. doi:10.1109/ICCV.2015.169

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770–778). Las Vegas, NV, USA. doi:10.1109/CVPR.2016.90

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*. doi:10.48550/arXiv.1412.6980

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems, 25*, 1097–1105. doi:10.1145/3065386

Li, B., Zhang, T., & Xia, T. (2017). Vehicle detection from 3D LiDAR using fully convolutional network. *Robotics and Autonomous Systems, 95*, 131–142. doi:10.48550/arXiv.1608.07916

Lin, Z., Wu, Y., Ma, Y., Chen, J., Zhang, R., Wu, J., ... & Lin, L. (2025). YOLO-LLTS: Real-time low-light traffic sign detection via prior-guided enhancement and multi-branch feature interaction. *IEEE Transactions on Instrumentation and Measurement, 74*, 1–18. Art. no. 5043518. doi:10.1109/TIM.2025.3604925

Luo, Y., Ci, Y., Zhang, H., & Wu, L. (2024). A YOLOv8-CE-based real-time traffic sign detection and identification method for autonomous vehicles. *Digital Transportation and Safety, 3*(3), 82–91. doi:10.48130/dts-0024-0009

Mathias, M., Timofte, R., Benenson, R., & Van Gool, L. (2013). Traffic sign recognition—How far are we from the solution? In *The 2013 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8). Dallas, TX, USA. doi:10.1109/IJCNN.2013.6707049

Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. *arXiv*. doi:10.48550/arXiv.1804.02767

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 779–788). Las Vegas, NV, USA. doi:10.1109/CVPR.2016.91

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*(6), 1137–1149. doi:10.1109/TPAMI.2016.2577031

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2014). OverFeat: Integrated recognition, localization and detection using convolutional networks. *arXiv*. doi:10.48550/arXiv.1312.6229

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*. *arXiv*. doi:10.48550/arXiv.1409.1556

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2017). Inception-v4, Inception-ResNet and the impact of residual connections on learning. *Proceedings of the AAAI Conference on Artificial Intelligence, 31*(1). doi:10.1609/aaai.v31i1.11231

Zhao, H., Gallo, O., Frosio, I., & Kautz, J. (2017). Loss functions for image restoration with neural networks. *IEEE Transactions on Computational Imaging, 3*(1), 47–57. doi:10.1109/TCI.2016.2644865

Pratiksha Madhukar Jadhav
Department of Computer Engineerir
Late G. N. Sapkal College of Engineering, Nashik, India
1994pratikshajadhav@gmail.com
ORCID 0009-0004-3058-2985

Nilesh R. Wankhade
Department of Computer Engineering
Late G. N. Sapkal College of Engineering, Nashik, India
hodcomp.lgnscoe@sapkalknowledgehub.org
Orchid 0000-0002-9040-1881

Sahebrao B Bagal
Department of Computer Engineering
Late G. N. Sapkal College of Engineering, Nashik, India
principal.lgnscoe@sapkalknowledgehub.org
ORCID 0009-0009-3473-9886

Ashwini V Sonawane
Department of Computer Engineerir
Late G. N. Sapkal College of Engineering, Nashik, India
ashwini1710sonawane@gmail.com
ORCID 0009-0003-9359-7175

Pragati K Ghayvate
Department of Computer Engineering
Late G. N. Sapkal College of Engineering, Nashik, India
Active pkghayvate@gmail.com
ORCID 0009-0002-6255-5590

Vishal Ambhore
D Assistant Professor, Department of E&TC Engineering, Vishwakarma Institute of Technology, Pune, Maharashtra, 411037
vishal.ambhore@vit.edu
ORCID 0000-0002-5174-4734