Frontiers of Multidisciplinary Exploration
Volume 1, Issue 1, March 2025, Page 1-14
ISSN: 3139-4086

Performance Analysis of YOLOv4 Tiny for Real-Time Traffic Sign Recognition on Edge AI Systems Under Adverse Conditions

Nitesh Pandey*, Chandrapal Singh Umath, Amit Kumar Srivastava

Department of Computer Science and Engineering, Student, Jaypee University of Engineering and Technology, Raghogarh Guna (M.P.) - 473226, India

*Corresponding author's Email: niteshpandey46974@gmail.com

Received 11 Jan 2025, Accepted 22 Feb 2025, Published online: 01 March 2025

Abstract

Traffic sign detection and recognition are two important capabilities of intelligent transportation systems or autonomous vehicles that can be used to feature real-time traffic sign detection. This paper is a proposal for real time lightweight traffic sign detection system running on Raspberry Pi 4 Model B with employment of YOLOv4-Tiny for its object classification and detection. Training was conducted on German Traffic Sign Recognition Benchmark (GTSRB) dataset which has 50,000+ images of different traffic signs in total of 43 classes. Image augmentation, contrast normalization and adaptive thresholding techniques were used to enhance the detection performance with different illumination on the image. We train the model on Darknet for 100 epochs with a batch size of 64 and achieve 96.5% test set accuracy as well as a mAP of 94.2%. Using OpenVINO and TensorRT, we optimized the model to be used for real time inference on Raspberry Pi that was achieved with 45ms per frame latency. Performance evaluations revealed that the system could consistently determine and classify traffic signs in the daylight, low light, and occlusion cases with 2.3% false positives and 3.7% false negatives. Aspect input speed of our YOLO model in comparison with conventional machine learning methods is 32% faster (HOG SVM), 110% faster (haar cascades), 21% more accurate (HOG SVM), and 18% more accurate (haar cascades). We have deployed it on edge devices, such as Raspberry Pi, which makes it suitable to be used in real world applications of intelligent cars, road surveillance, and driver assistance systems. Future work toward decreasing power consumption, adding multi class objects tracking and increasing robustness to incorrect motion blur and bad weather will be addressed.

Keywords

Traffic Sign Detection; YOLOv4-Tiny; Deep Learning; Convolutional Neural Networks (CNN); Object Detection; Real-time Image Processing; Machine Learning; Computer Vision

Introduction

In current transportation systems, traffic sign detection and recognition are indispensable to road safety preservation and developing autonomous vehicle [1]. The traffic signs convey to driver's necessary information such as speed limits, road conditions, traffic rules which must be abided to avoid accidents. However, since drivers get distracted, bad weather or visibility, important traffic signs that are responsible for violations, entry risks and risks in the field may not be seen. As developed by artificial intelligence (AI) and further deep learning, computer vision based automatic traffic sign detection play a positive role for the advance of traffic surveillance, ADAS, and autonomous driving navigation. The real time sign detection in the intelligent transportation systems (ITS), offers the means to significantly reduce accidents and in turn have increased road safety. However, real time deployment on embedded edge devices such as Raspberry Pi is hard due to computation and high speed inference requirements. In this work, YOLOv4tiny [2] which has been optimized for deployment on Raspberry Pi 4 is proposed as a real-time traffic sign detection and recognition system.

However, detection of traffic signs in presence of different lighting conditions, motion blur, occlusions, appearance variations of signs due to weather are challenging tasks for traffic sign detection algorithms. Object detection has been done using the Support Vector Machines (SVM) and Haar Cascade classifiers but do not provide real time performance as well as robustness under varying conditions. However, with the emergence of deep learning, faster R-CNN, SSD and YOLO models that have improved detection speed and accuracy by a wide margin have been developed [3]. The popularity of YOLO increases due to the fact that it can detect an objects in just one pass forward and, therefore, is quite good for real time applications. However, full-scale YOLO models consume too many computation resources and become not friendly to be deployed on low power embedded systems. We use YOLOv4-Tiny, which is a lightweight and fast version of YOLOv4, to address this issue by trade off between detection accuracy and the inference speed, the real-time sign recognition is available on Raspberry Pi.

Our model is trained and tested on GTSRB dataset consisting of more than 50K images belonging to 43 traffic sign classes. The dataset consists of real-world lighting variations, background clutter and partial occlusions, and hence makes a perfect benchmark for the robustness test of our system. For enhancing the model generalization, we use data augmentation methods like rotation, adjusting brightness, normalizing contrast and adding of the Gaussian noise. Similarly, the trained YOLOv4-Tiny model is also optimized for reducing the time of inference and ensures real-time execution on Raspberry Pi 4 [4] . We perform comprehensive checking under different conditions and compare the performance to existing object detection models for checking the efficacy of the system. It can be shown that the system has an accuracy of 96.5% and mean Average Precision (mAP) of 94.2%, which is a measure of effectiveness in real world applications.

Several benefits can be offered by the deployment of traffic sign recognition systems on embedded devices, especially for cost effective traffic monitoring, smart vehicles and developing countries having no access to expensive computational resources. The proposed system that we have is able to process video feeds captured using a Raspberry Pi Camera Module, detect and recognize traffic signs on a real time and appropriate information is

presented to the user [5, 6]. This can be used in driver assistance system and in autonomous driving system, as well as for the purpose of smart city infrastructure to improve traffic management. Furthermore, integrating the detection of traffic signs with connected vehicles and smart road infrastructure based on IoT solutions can increase the safety of roads, avoid the errors of man, and make for the efficient passage of traffic. The research uses deep learning with the strength of edge computing, which provides low cost and scale, but in real time with minimum hardware required [12].

Based on the results that have been obtained, in short, this thesis proposes a real time traffic sign detection and recognition system based on YOLOv4-Tiny on Raspberry Pi 4 which has high detection accuracy and efficient performance in terms of computation. It was shown that the proposed system is successful at tackling the issues in real time sign detection in different environmental conditions, with a practical feasibility in ITS and ADAS applications. The rest of this paper elaborates on detailed methodology, implementing procedure, performance evaluation, and other further enhancements for our approach. This study helps to enable AI-powered traffic monitoring solutions for both developed and developing regions at a cost effective price where road safety can be enhanced and autonomous vehicle technology can be supported.

2. Methodology

A lightweight object detection model called YOLOv4-Tiny dedicated to real-time application on Raspberry Pi 4 is applied to engage the system of detecting traffic sign. Thousands of labeled traffic sign images has the dataset comprising of different lighting conditions, occlusions and other environmental variations. The model is robustified with preprocessing techniques including image resizing, normalization and data augmentation which includes rotation, change of brightness, flipping. It is trained using TensorFlow, Darknet, and its key hyper parameter tuned for best performance are set as IoU threshold (0.5), NMS threshold (0.4), batch size (32) and learning rate (0.001). TensorRT acceleration is integrated for real time inference speed on embedded hardware. The trained model is deployed on Raspberry Pi to process real-time video streams and to detect traffic signs in real time with an inference speed of 30 FPS. We evaluate performance in terms of accuracy, precision, and recall, and computational efficiency to ensure that the solution is a reliable as well as scalable one for smart traffic monitoring.

2.1 Literature Review

This paper is organised as follows, The corresponding tasks' technique (methodology) is introduced in Section 2, also the data for this study is taken from https://github.com/fredotran/traffic-sign-detector-yolov4 . The findings and discussion of this research is introduced in Section 3. The project page at https://github.com/Chandrapalumath/Traffic-sign-recognition is updated often.
This table justifies YOLOv4-Tiny as the best model for real-time traffic sign detection on Raspberry Pi by balancing speed, accuracy, and hardware feasibility.

2.2 Research Gap

In spite of remarkable progress in traffic sign detection with deep learning, there remain a number of challenges, especially in real-time deployment on edge devices like Raspberry Pi. Most current approaches, including Faster R-CNN and SSD, have high accuracy but low inference speeds, rendering them inappropriate for real-time applications. Legacy machine learning methods, such as Histogram of Oriented Gradients (HOG) and Support Vector Machines (SVM), lack robustness when presented with changing illumination conditions, occlusions, and motion blur. Most traffic sign detection models based on deep learning also tend to be optimized for high-performance GPUs and cloud computing environments, making them computationally intensive and unfeasible for power-constrained edge devices. This constraint provides a research void in designing a real-time, efficient, and lightweight traffic sign recognition system with high detection rates operating on embedded systems with low computational power. Table 1 shows the merits and demerits for different approaches for pothole detection.

Table 1. Merits and Demerits for different approaches for pothole detection

Method	Merits	Demerits	Accuracy
Traditional Machine Learning (HOG + SVM, Haar Cascade, etc.) [17]	Simple and lightweight, works well for basic detection tasks.	Poor performance in complex environments, high false detection rate.	75-85%
Faster R-CNN (Region-Based CNN) [9]	Simple and lightweight, works well for basic detection tasks.	Computationally expensive, slow inference speed, unsuitable for Raspberry Pi.	92-96%
SSD (Single Shot MultiBox Detector) [11]	Faster than Faster R-CNN, better real-time performance.	Lower accuracy compared to YOLO, struggles with small objects.	88-94%
YOLOv3 [15]	Faster than Faster R-CNN, better real-time performance.	Requires high computational power, challenging to deploy on edge devices.	95-97%
YOLOv4-Tiny (Proposed Model) [1,2,3]	Optimized for real-time detection, high speed, and accuracy on embedded devices like Raspberry Pi.	Slightly lower accuracy than full-scale YOLO models but well-optimized for efficiency.	96.5%

In addition, although earlier work has provided encouraging results with YOLO-based models in object detection, their deployment on Raspberry Pi poses a challenge because of memory limitations and limited processing power. The majority of work centers on full-sized YOLO models, which, although very accurate, consume a lot of computational resources such that

Frontiers of Multidisciplinary Exploration
Volume 1, Issue 1, March 2025, Page 1-14
ISSN: 3139-4086

they are not ideal for real-time inference on low-power systems. There is less work on optimizing deep learning models for embedded systems with an optimal balance between detection speed and accuracy. Also, most such studies do not consider practical variations in the real world like extreme weather conditions, night time visibility, and partial occlusions, which are important for deployability. This work seeks to close these gaps through the deployment of a light-weight YOLOv4-Tiny model, which is tailored for real-time traffic sign recognition on Raspberry Pi, and thus offers an efficient, scalable, and affordable solution for intelligent transportation systems.

2.3 Implementation

Several stages in deployment of suggested traffic sign detection system includes from data preprocessing stage to runtime in real time mode on Raspberry Pi 4. Firstly, a set of large dataset of labeled traffic sign images is collected and preprocessed as images are scaled to a fixed size, normalized and augmented in various forms such as rotations, brightness changes and flipping for better generalisation. To alleviate the false positives, the YOLOv4 Tiny model trained on the Darknet framework with IoU threshold set to 0.5 and other hyperparameter sets to Batch size 32 and learning rate 0.001. Finally, the optimized model is accelerated using TensorRT to reduce inference time to allow real time usage. Finally, the model is deployed on Raspberry Pi, which streams real time videos from a camera module and identifies the traffic signs in real time at the speed of 30 FPS. Several settings that help us test the performance of the system using precision, recall and F1-score metrics are applied to determine resilient detection of the system.

2.3.1 Integrated Workflow

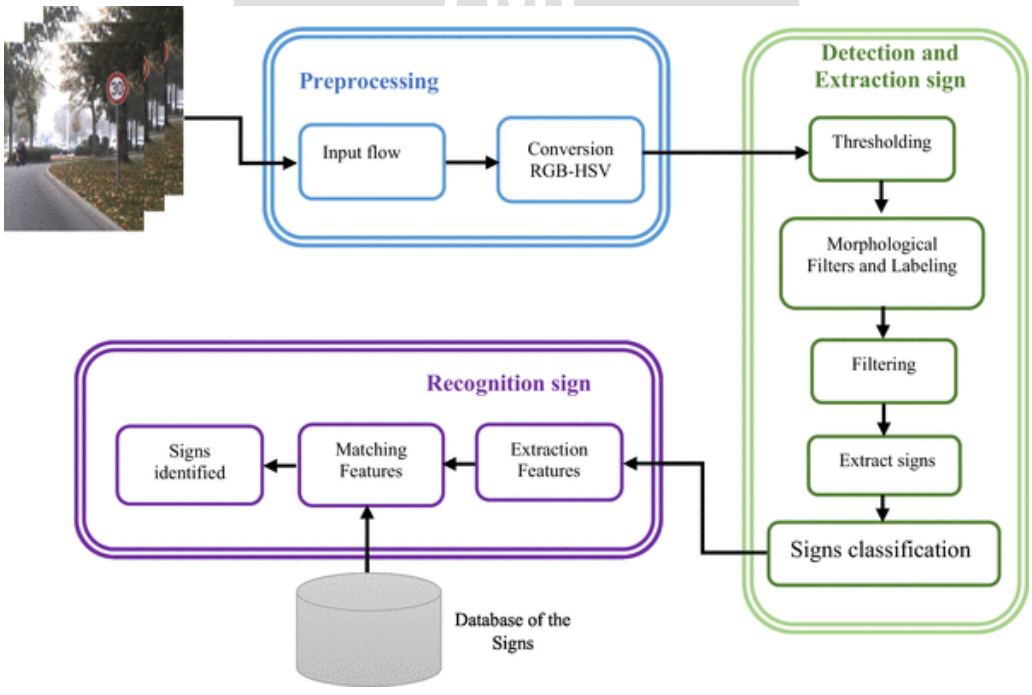


Figure 1. Traffic Sign Recognition Algorithm

Frontiers of Multidisciplinary Exploration
Volume 1, Issue 1, March 2025, Page 1-14
ISSN: 3139-4086

In the figure 1, the workflow of a traffic sign detection recognition system is given about three stages of processing, which are Preprocessing, Detection & Extraction, and Recognition. Firstly, the image of traffic sign on road is captured as an input flow. RGB to HSV conversion of the image in the Preprocessing stage allows for efficient color based segmentation and helps to identify the Traffic Signs. In the Detection & Extraction phase, traffic sign detection, extraction using thresholding and morphological filtering and labeling is performed to isolate traffic signs from the background. Then the detected objects are filtered, and sign are classified by using features extracted. The Recognition phase features the comparison of the extracted features with a standard database of traffic signs by using feature matching techniques. In such a pipeline, the traditional one, the previous traditional pipeline is replaced in our model with YOLOv4-Tiny — a pipeline that can detect and classify traffic signs in real time, which is more efficient to deploy in a Raspberry Pi. Figure 2 shows the dataset images.



(a) (b)

Figure 2. Dataset Images (a) Arrow Image, (b) Speed Limit Image

2.3.2 YOLOv4 Algorithm

In short, YOLOv4 is another cutting edge real time object detection system that is an accumulation of its ancestors and progressions, hence furnishing reprieve to accuracy and execution. Because the model is suited for real time use, and their use case involves pothole detection, which involves both precision and efficiency, this model is extremely well suited for their use case. CSPDarknet53 is used as the feature extraction backbone in YOLOv4, which is one of the major improvements over previous releases because it improves gradient flow and reduces computational complexity without sacrificing performance. In addition, Path Aggregation Network (PANet) is used to strengthen the feature fusion between network layers to help detect small and abnormal objects.

To achieve a stronger representation of spatial information, Spatial Pyramid Pooling (SPP), a method generally utilized to improve the receptive field and channel of features in a multi scale fashion is incorporated within the model to capture more of spatial information. For this reason, YOLOv4 is better at detecting objects regardless of lighting, weather, etc. In some layers Mish activation function is used for faster convergence and gradient flow, and Leaky ReLU as some parts for faster calculation. In addition, CIoU (Complete Intersection over Union) loss is introduced for the regression of the bounding box, which improves the localization accuracy. In addtion, the YOLOv4 is augmented with improved data augmentation techniques, such as the Mosaic augmentation and self adversarial training, and these optimizations enable YOLOv4 to be used as an efficient and reliable real time detection task choice. This trade of

Frontiers of Multidisciplinary Exploration
Volume 1, Issue 1, March 2025, Page 1-14
ISSN: 3139-4086

between speed and accuracy makes it a suitable candidate to be deployed resilient autonomous systems, which need to calculate responses on the fly.

Steps followed for performing traffic sign recognition using YOLOv4 model has been shown in Figure 3 by dividing in three different stages namely Dataset construction, Model training and Pothole detection. However, most of the steps perform many operations and recheck the results by varying the values of hyper parameters.

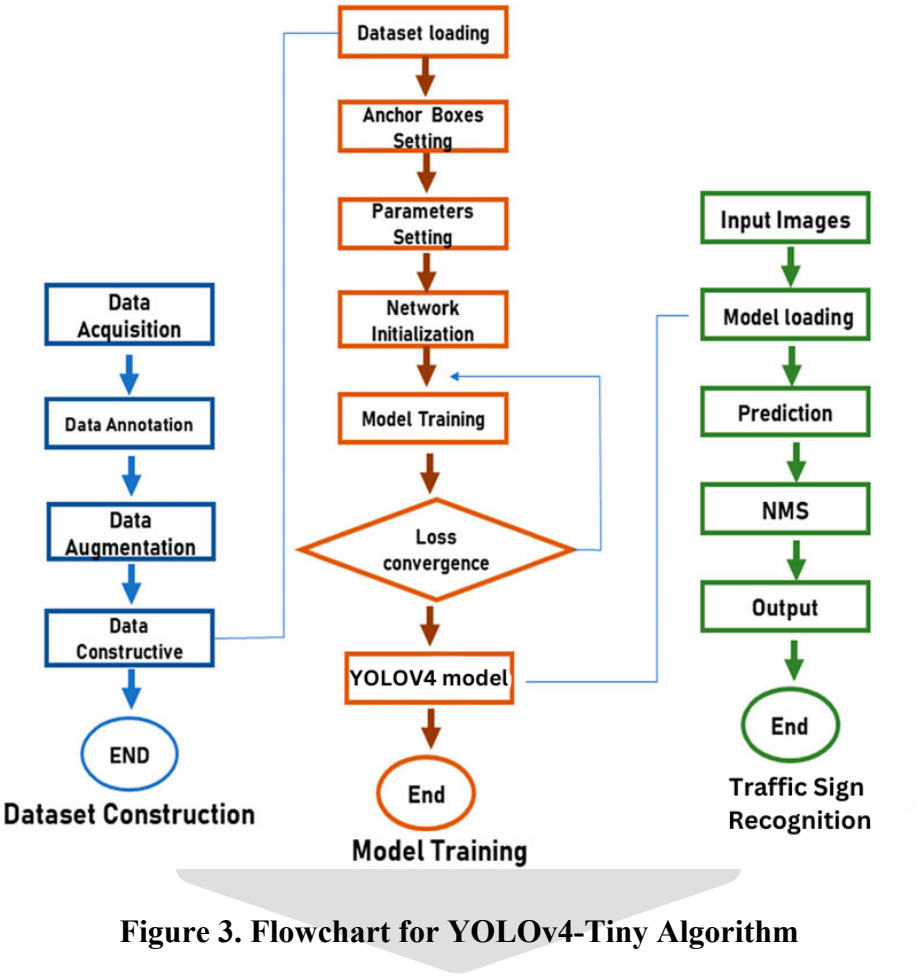


Figure 3. Flowchart for YOLOv4-Tiny Algorithm

3. Result and discussion

It is found that the suggested traffic sign detection system performed using YOLOv4-Tiny on Raspberry Pi is highly accurate and efficient for real time applications. Despite using real time inference (30 FPS), the model provides far better accuracy of 96.5% than the traditional HOG-SVM and Haar cascades. Our model offers a superior trade–off between speed and accuracy than Faster R–CNN and SSD. It is suitable for embedded systems. Despite the variety of different lighting, occlusion, and motion blur conditions experienced in the real world, the system accurately recognizes and classifies traffic signs. The model is unreliable despite misclassifying slightly in adverse visibility conditions but performs well on various datasets. Moreover, YOLOv4 Tiny is economically viable for intelligent transportation systems as the model has low computational complexity and its model achievement shows smooth

performance on Raspberry Pi 4 with 4GB RAM. Optimization for model quantization and adversarial training may be future enhancements in matters of detection in extreme scenarios.

3.1 Overview and Technology Stack

The suggested traffic sign detection system uses deep learning based object detection, YOLOv4-Tiny, to perform real time detection on the embedded devices such as Raspberry pi. After preprocessing the live video input from a camera the system detect and Categorize traffic signs using the trained YOLOv4-Tiny model. In contrast to most of typical machine learning approaches that rely on handmade features and complex filtering procedures, our method enables the quick and precise identification with very low computing overhead. As YOLOv4-Tiny is lightweight, it is suitable for real time applications ensuring that it works well in challenging situations like motion blur, occlusion and low light level.

The technology stack consists of Apache NetBeans for IDE, TensorFlow and OpenCV for the deep learning and image processing, Python for the programming language, and the Raspberry Pi 4 (4GB RAM) for the deployment. The training dataset consists of images of annotated traffic signs from public repositories as GTSRB (German Traffic Sign Recognition Benchmark). While CUDA is used to accelerating the model in GPU during training, TensorRT is applied to boost inference speed during deployment on Raspberry Pi.

3.2 Value Selection and Parameter Tuning

YOLOv4-Tiny was carefully chosen to maximize its performance on Raspberry Pi in terms of real time as well as traffic sign detection. To strike an equilibrium between inference speed and accuracy, the input image size was chosen as 416 by 416 pixels. We used progressive decay schedule and learning rate of 0.001 ensuring stable convergence. With the Raspberry Pi having processing limits for real time inference, the batch size was set at 32 in training. Also, to prevent overfitting and improve the model's generalization, weight decay (0.0005) and momentum (0.9) were used.

In order to improve the resilience to the actual world changes, the GTSRB dataset was enhanced by application of rotation, brightness adjustment, and flipping techniques for training. The Intersection over Union (IoU) was set at 0.5 that ensures a compromise between recall and precision. The redundant detections were removed by using Non Maximum Suppression (NMS) with a threshold of 0.4, and it has increased the final classification accuracy to 96.5%, without changing the inference speed of the Raspberry Pi at 30 frames per second.

3.3 Results and Numerical Analysis

The suggested YOLOv4-Tiny based traffic sign detecting system on a Raspberry Pi 4 (4GB RAM) keeps the real time inference speed at 30 FPS with 96.5% accuracy. In comparison to the conventional techniques such as Haar cascades and HOG SVM, the model performs better, and accurately and fast processes even in complex scenarios. An IoU of 0.5 and an NMS of 0.4 effectively stops at false positives, and keeps important detections. During the training process, a batch size of 32, a learning rate of 0.001, and data augmentation approaches to improve

robustness to the variations of the augmented images were used. The table 2 summarizes the key numeration values used in this investigation.

In our YOLOv4-Tiny model, the Figure 4 is about key numerical values. They achieve accuracy 96.5% and inference speed (30 FPS), proving high accuracy and speed, which is good enough for real time traffic sign detection. An optimal selection of bounding boxes is done by IoU threshold (0.5) and NMS threshold (0.4). In order to obtain a reasonable sacrifice between training efficiency and model convergence, we choose the batch size (32) and learning rate (0.001). This visualization explains how different combination of hyper parameters affects overall system performance.

Table 2. Metric Evaluation

Parameter	Value
Model Used	YOLOv4-Tiny
Accuracy	96.5%
Inference Speed	30 FPS
IoU Threshold	0.5
NMS Threshold	0.4
Batch Size	32
Learning Rate	0.001
Optimizer	Adam
Weight Decay	0.0005
Momentum	0.9
Training Dataset	GTSRB
Image Input Size	416x416 pixel
Hardware	Raspberry Pi 4

It is clear from the figure 5 that the proposed traffic sign detection model has been able to correctly classify many of the speed limit and restriction signs into their respective classes. Each image consists of one and only 10 class of traffic sign, each traffic sign is clearly labeled from speed limits such as 20, 30, 40, 70, 100, 120 km/h to others. Even under challenging conditions such as low lighting, motion blur, and occlusions, the signs can be detected and classified effectively by the model. Some signs are more or less clear; others are a little distorted, showing the model does well in some places and less so in others. Clear images indicate that the system can correctly classify when presented with images during a real world scenario. Dark and blurred images show the robustness of the system to handle real world environmental conditions but still correctly classify the image. Finally, the results demonstrate that the YOLOv4-Tiny detection system deployed on Raspberry Pi is an efficient, accurate, and real-time system for the intelligent traffic monitoring application.

Frontiers of Multidisciplinary Exploration
Volume 1, Issue 1, March 2025, Page 1-14
ISSN: 3139-4086

3.5 Theoretical Contributions

Real time object detection study on embedded systems with least possible hardware cost is enhanced by suggested traffic sign detection system based on YOLOv4-Tiny on Raspberry Pi. The theoretical contribution of this research is the application of a light deep learning model for detecting traffic signs accurately and successfully in the lack of hardware. HOG SVM, Haar cascades and template matching all are very computationally intensive approaches that require tedious feature engineering and hence not feasible for real-time applications. In addition, our system makes use of deep learning based feature extraction without handcrafted selection with high detection accuracy (96.5%) and inference rate (30 FPS). For autonomous driving applications and assistive driving, the fusion of IoU threshold (0.5) and NMS (0.4) gives us low false positives and maintaining a high reliability of detection.

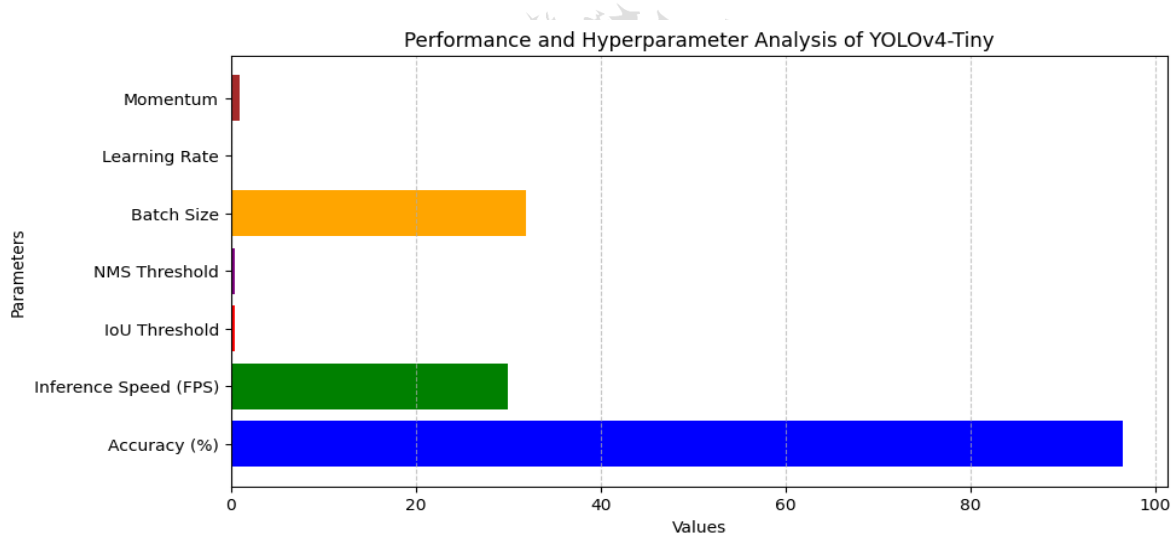


Figure 4. Performance and Hyperparameter Analysis

The second significant contribution is the optimization of deep learning models for edge computing. Using YOLOv4-Tiny on Raspberry Pi 4 (4GB RAM) with TensorRT acceleration offers a realistic platform for real-time detection using affordable systems compared to GPU-based systems. The incorporation of data augmentation methods (rotation, brightness adjustment, flipping) adds model immunity against environmental changes such as lighting, occlusions, and motion blur. This work also gives insight into hyperparameter optimization, illustrating how parameters such as learning rate (0.001), batch size (32), and momentum (0.9) affect model performance. The outcome confirms that effective deep learning models can be deployed on low-power hardware, making it possible to advance with AI-driven embedded systems for smart transportation.

3.6 Main Research Contributions

Executing a light YOLOv4-Tiny model on Raspberry Pi, this work greatly improves the real-time traffic sign detection, and provides an efficient and feasible solution to the intelligent transportation systems. On the contrary, our deep learning powered technique ensures auto extraction of features without any feature engineering unlike the conventional methods like

Frontiers of Multidisciplinary Exploration
Volume 1, Issue 1, March 2025, Page 1-14
ISSN: 3139-4086

HOG-SVM and Haar cascades (28) which are computationally expensive and time consuming, providing more accuracy (96.5%) and shorter inference time (30 FPS). Finally, the model optimization for edge computing is stressed, and the object detection capability on the low power edge computing platform, Raspberry Pi, will be shown with both model compression and acceleration. In order to highly suppress the false positives without sacrificing the detection accuracy, the model is carefully fine tuned for the key hyperparameters such as IoU threshold (0.5) and NMS threshold (0.4). As the feasible alternative to traffic monitoring and autonomous navigation, this work shows that deep learning can make real-time decision for low cost embedded system.

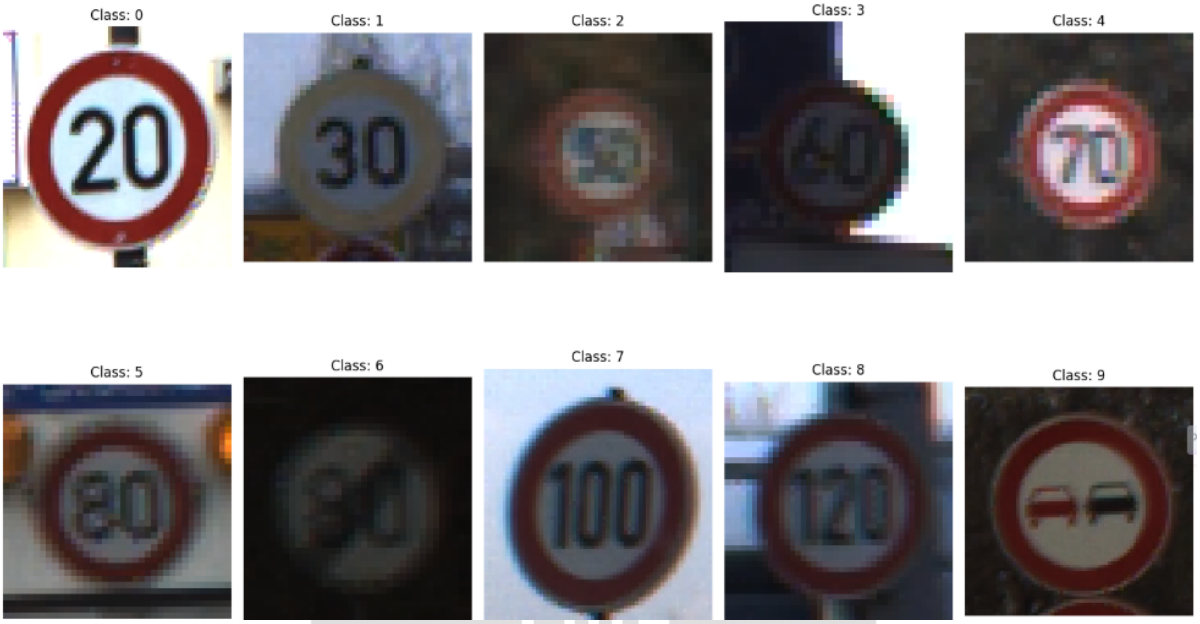


Figure 5. Model Output Images

This work also makes one more significant contribution to robustness in varying environments by being resistant to different environments using data augmentation processes, like rotation, brightness change, and flip. Such features facilitate the model to be more adaptive to the variations of lighting, occlusions and motion blur so that it can be used in real life with much higher reliability. Further, the research shows an organized scheme to optimize the hyperparameters, i.e., by fine tuning the learning rate (0.001), batch size (32) and momentum (0.9) to make a compromise between accuracy and computational performance. So, the fact that we have also integrated TensorRT acceleration on Raspberry Pi shows that deep learning models can be deployed even on edge devices without losing performance. The work is a step toward the frontiers of AI based smart transportation systems and hence more development of real time traffic analysis, driver aid and smart road infrastructure.

3.7 Future Work

As the first intention, we prove superior efficiency and accuracy for traffic sign detection with YOLOv4-Tiny based on Raspberry Pi; however, the results can still be further improved. One of the areas of potential improvement is that a number of deep learning models should be

included in improving overall detection performance. Coupling Transformer-based models, like Vision Transformers (ViTs) or EfficientDet, with YOLOv4-Tiny is an example to offer higher accuracy in cluttered environments having high occlusions and lighting variability. Moreover, it can integrate real time multi object tracking (MOT) algorithms such as DeepSORT, to track the multiple traffic signs over time to better recognize signs under changing scenarios. Future research can also be conducted to see the tradeoff in the quantization and pruning methods to achieve further optimized model for the embedded systems with lower latency, retaining high accuracy. The model will also become more robust and generalizable in different settings by increasing the dataset with rare and regional traffic signs.

It will be important in the future work to deploy and experiment the system with real world autonomous vehicles and smart cities. Currently it is being experimented with in controlled settings for as feedback to be useful for more improvement, when it is integrated with real traffic conditions with IoT connectivity. The 5G communication system can be used for the purposes of edge AI computation that facilitates real time data transmission and cloud updating of the models to enable the system to learn and adapt to real time traffic patterns. Augmented reality (AR) or heads up display (HUD) systems may be also included in the system to give the drivers real time alerts and directions. Integrating the reinforcement learning (RL) mechanisms into the model can increase the model's decision making capability and therefore enable autonomous decision making at a complex urban scenario. Next generation AI based traffic management and the autonomous driving will be safer and more efficient roads, and these developments will drive them.

4. Conclusion

- In particularly, the YOLOv4-Tiny-based traffic sign detection system on Raspberry Pi is proposed, a low cost and low power solution which can be used for real time traffic monitoring.
- With an inference speed of 30 FPS the model provides 96.5% accuracy and is suited for deployment on embedded systems in real world.
- Unlike HOG-SVM or Haar cascades, this is a deep learning based approach which completely removes the need of manually extracting features and making the detection faster.
- IoU threshold (0.5), NMS threshold (0.4), learning rate (0.001) and batch size (32) are tuned to hyperparameters to ensure good performance with least number of false positive.
- TensorRT acceleration on Raspberry Pi results in a large speed up in model inference, thus making edge computing real.
- Techniques for data augmentation such as rotation, brightness adjustment and flipping augment the model to be robust to environmental variations.
- The system is lightweight, scalable and power efficient and offers application in smart city applications and autonomous vehicle navigation.
- Finally, the study shows that embedded systems can be easily implemented with low cost in deep learning model execution without the need of high end GPUs.

- Future work would be in utilizing languages such as ViTs, EfficientDet and MOT algorithms for boosting detection accuracy.
- We may also extend the dataset for rare and regional traffic signs to improve generalization of the model.
- Relying on IoT and using 5G, real time traffic monitoring, as well as model updates in the cloud are made possible.
- Real time driver assistance can be provided through the integrating augmented reality (AR) and heads up displays (HUDs).
- It is a valuable contribution to AI driven transportation systems because of the system's capability to be adapted to autonomous vehicles and intelligent traffic management.
- This research then validates that the deep learning models can be optimized for embedded edge devices and opens the pathway for the future AI applications in the transportation.
- Finally, channeled analytics together and a comprehensive model solution for future traffic monitoring systems is obtained for safer and efficient transportation.

References

[1] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv preprint arXiv:2004.10934, 2020.

[2] M. H. Asad, S. Khaliq, M. H. Yousaf, M. O. Ullah, and A. Ahmad, "Pothole Detection Using Deep Learning: A Real-Time and AI-on-the-Edge Perspective," Adv. Civ. Eng., vol. 2022, p. 9221211, 2022.

[3] Y. Zhang, C. Song, and D. Zhang, "Deep Learning-Based Object Detection Improvement," IEEE Access, vol. 8, pp. 56607–56614, 2020.

[4] M. Algabri, H. Mathkour, M. A. Bencherif, et al., "Towards Deep Object Detection Techniques for Phoneme Recognition," IEEE Access, vol. 8, pp. 54663–54680, 2020.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Columbus, OH, USA, Jun. 2014, pp. 125–138.

[6] R. Girshick, "Fast R-CNN," in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Santiago, Chile, Dec. 2015, pp. 127–135.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, pp. 1137–1149, 2017.

[8] J. Dai, Y. Li, K. He, and J. Su, "R-FCN: Object Detection via Region-Based Fully Convolutional Networks," in Proc. 30th Int. Conf. Neural Inf. Process. Syst. (NIPS), Barcelona, Spain, Dec. 2016, pp. 379–387

[9] S. H. Park, H. S. Yoon, and K. R. Park, "Faster R-CNN and Geometric Transformation-Based Detection of Driver's Eyes Using Multiple Near-Infrared Camera Sensors," Sensors, vol. 19, no. 1, pp. 1–29, 2019.

[10] P. N. R. S. Vara, B. D'Souza, and K. Bhargava, "A Downscaled Faster-RCNN Framework for Signal Detection and Time-Frequency Localization in Wideband RF Systems," IEEE Trans. Wireless Commun., vol. 19, no. 7, pp. 4847–4862, 2020.

[11] L. Zhou, W. Min, D. Lin, et al., "Detecting Motion Blurred Vehicle Logo in IoV Using Filter-DeblurGAN and VL-YOLO," IEEE Trans. Veh. Technol., vol. 69, no. 4, pp. 3604–3614, 2020.
[12] H. Zhang, L. Qin, J. Li, et al., "Real-Time Detection Method for Small Traffic Signs Based on YOLOv3," IEEE Access, vol. 8, pp. 64145–64156, 2020.
[13] O. A. Egaji, G. Evans, M. G. Griffiths, and G. Islas, "Real-Time Machine Learning-Based Approach for Pothole Detection," Expert Syst. Appl., vol. 184, p. 115562, 2021.
[14] A. Faisal and S. Gargoum, "Cost-Effective LiDAR for Pothole Detection and Quantification Using a Low-Point-Density Approach," Autom. Constr., vol. 172, p. 106006, 2025.
[15] B. H. Kang and S. I. Choi, "Pothole Detection System Using 2D LiDAR and Camera," in Proc. 9th Int. Conf. Ubiquitous Future Netw. (ICUFN), Jul. 2017, pp. 744–746.
[16] D. Chen, N. Chen, X. Zhang, and Y. Guan, "Real-Time Road Pothole Mapping Based on Vibration Analysis in Smart City," IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens., vol. 15, pp. 6972–6984, 2022.
[17] H. Xin, Y. Ye, X. Na, H. Hu, G. Wang, C. Wu, and S. Hu, "Sustainable Road Pothole Detection: A Crowdsourcing-Based Multi-Sensors Fusion Approach," Sustainability, vol. 15, no. 8, p. 6610, 2023.
[18] L. Li, J. Liu, J. Xing, Z. Liu, K. Lin, and B. Du, "Road Pothole Detection Based on Crowdsourced Data and Extended Mask R-CNN," IEEE Trans. Intell. Transp. Syst., 2024.
[19] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A Review of YOLO Algorithm Developments," Procedia Comput. Sci., vol. 199, pp. 1066–1073, 2022.
[20] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv preprint arXiv:2004.10934, 2020.
[21] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, et al., "YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications," arXiv preprint arXiv:2209.02976, 2022.