Pertanika J. Sci. & Technol. 31 (1): 137 - 148 (2023)

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

Real-Time Traffic Sign Recognition Using Deep Learning

Ananya Belagodu Shivayogi[1], Nehal Chakravarthy Matasagara Dharmendra[1]*, Anala Maddur Ramakrishna[2] and Kolala Nagaraju Subramanya[3]

[1]*Department of Computer Science, R. V. College of Engineering, Bangalore, 560059 India*
[2]*Department of Information Science, R. V. College of Engineering, Bangalore, 560059 India*
[3]*R. V. College of Engineering, Bangalore, 560059 India*

ABSTRACT

Traffic Sign Recognition (TSR) is one of the most sought-after topics in computer vision, mostly due to the increasing scope and advancements in self-driving cars. In our study, we attempt to implement a TSR system that helps a driver stay alert during driving by providing information about the various traffic signs encountered. We will be looking at a working model that classifies the traffic signs and gives output in the form of an audio message. Our study will be focused on traffic sign detection and recognition on Indian roads. A dataset of Indian road traffic signs was created, based upon which our deep learning model will work. The developed model was deployed on NVIDIA Jetson Nano using YOLOv4 architecture, giving an accuracy in the range of 54.68–76.55% on YOLOv4 architecture. The YOLOv4-Tiny model with DeepStream implementation achieved an FPS of 32.5, which is on par with real-time detection requirements.

Keywords: DeepStream, Indian traffic sign dataset, NVIDIA Jetson Nano, traffic sign detection, YOLOv4

ARTICLE INFO

Article history:
Received: 26 February 2022
Accepted: 31 May 2022
Published: 19 August 2022

DOI: https://doi.org/10.47836/pjst.31.1.09

E-mail addresses:
ananyabs.cs18@rvce.edu.in (Ananya Belagodu Shivayogi)
nehalcmd.cs18@rvce.edu.in (Nehal Chakravarthy Matasagara Dharmendra)
analamr@rvce.edu.in (Anala Maddur Ramakrishna)
subramanyakn@rvce.edu.in (Kolala Nagaraju Subramanya)
* Corresponding author

INTRODUCTION

The Traffic Sign Recognition System is part of a bigger set of features called the Advanced Driver Assistance System (ADAS). ADAS plays a key role in the working of driverless or self-driving cars. With the help of data accumulated from various sensors and cameras, driverless vehicles make certain decisions that a typical driver takes. Besides self-driving cars, a TSR system can help drivers avoid road

ISSN: 0128-7680
e-ISSN: 2231-8526

accidents by minimising human errors. Traffic signs have been installed on the streets to give the necessary information that drivers have to follow, but often drivers tend to miss or ignore traffic signs while trying to focus on the road. A Traffic Sign Recognition System thus assists the driver by providing information about the traffic signs without the need for the driver to have a look at them.

Traffic sign detection and recognition in computer vision have recently been an active research topic. Initial research (Ellahyani et al., 2016) concentrated on the most important traffic signs, such as mandatory, prohibitory sign classification using colour and shape features. Before Convolutional Neural Networks (CNN) became popular, classical detection techniques for object detection like Histogram of Gradient (HOG) combined with Support Vector Machine (SVM) were widely used (Muthukumaresan et al., 2016; Do et al., 2017).

The recognition of a traffic sign from an image comprises two steps: localisation and classification. Object detectors such as the Faster-RCNN, Region-based Fully Convolutional Networks (R-FCN), and Feature Pyramid Networks (FPN) are region-based object detectors. On the other hand, one-stage detectors such as SSD-ResNet (Lu et al., 2019), MobileNet-SSD and YOLO perform object detection in a single step and are usually faster but at the cost of accuracy. YOLOv2 algorithm is used for Traffic Sign Recognition (Zhang et al., 2017), and high precision and low latency results were noted, despite the usage of a small number of traffic signs of different categories. The YOLO algorithm was analysed and considered a powerful object detection technique for real-time implementation (Oltean et al., 2019). In particular, YOLOv3 was used to balance between accuracy and speed properly. The implementation in this paper uses the YOLOv4 model (Bochkovskiy et al., 2020), as real-time implementation demands faster execution.

Many proposed systems are just limited to the classification of traffic sign instances captured by a camera (Tabernik & Skocaj, 2020; Luo et al., 2018). However, for real-time implementation, environmental parameters like cloudy and rainy conditions can influence the detection of traffic signs to a large extent (Tabernik & Skocaj, 2020; Sari & Cibooglu, 2018); hence an accurate object detection model using deep learning is required, which is not prone to scale changes (Hasegawa et al., 2019). Therefore, this paper proposes a system to recognise a traffic sign by adapting to various environmental conditions.

Through the course of this study, an Indian road traffic sign dataset is created, upon which the implementation of the TSR system will work. Most practical implementations convey the information by displaying an image of the sign on the dashboard. However, it further requires the driver to take his eyes off the road and look at the alert on the dashboard. Hence, an audio message would be a safer and much better way to convey the information. This paper describes a practical implementation of a TSR system on the NVIDIA Jetson Nano, which can detect Indian road traffic signs in real-time and give out audio alert messages as output. Table 1 consolidates the features of similar works conducted in this direction.

Table 1
Related work

Author(s) & Year	Feature extraction algorithm	Classifier	Dataset	Recognition rate	Comments
Lu et al (2019)	HOG and LSS	Random Forest classifier	Swedish Traffic Signs Data set includes more than 20,000 images	96%	Difficulty in recognising signs with motion blur.
Bochkovskiy et al. (2020)	HOG Feature Extraction	SVM classifier	Image database created by capturing in streets of Ho Chi Minh City	98%	Detection and recognition of speed limit traffic signs with high accuracy are seen even in conditions with complex backgrounds.
Luo et al. (2018)	Mask R-CNN	Fast R-CNN	A novel dataset containing thirteen thousand traffic signs and seven thousand high-resolution images	2-3% error rate	A deep learning method for traffic sign detection, with signs with intra-category variation in appearance, is proven to be adequate for real-time deployment.
Sari and Cibooglu (2018)	Maximally Stable Extremal Regions (MSERs)	Multi-task CNN	Street view images and synthesised traffic sign images from standard sign templates	87.3%	Recognition of traffic sign images with distortion, rotation and blur is good.
Hasegawa et al. (2019)	Edge Detection and Contour Detection	CNN	German Traffic Signs Recognition Benchmark (GTSRB)	87.36 %	Colour filtering method cannot be relied upon in lightning and other complex conditions in the test environment.
Zoph et al. (2019)	YOLOv2, which in turn utilises an FCN	YOLOv2	Novel dataset containing 16 traffic sign classes, including a total of 7160 images	66.39%	Training the model with images of varying sizes and data augmentation effectively handles scale and contrast changes.
Zhong et al. (2020)	An end-to-end convolutional network that YOLOv2 inspires	YOLOv2	CCTSDB - Changsha University of Science and Technology Chinese traffic sign detection benchmark	95.31%	Detection of small-sized traffic signs is not achieved and needs to be improved.
Islam (2019)	HSV colour space with masking	CNN based on the LeNet architecture	U.K. traffic signs image database, which has around 40000 images, including both positive and negative images	90%	Real-time detection and recognition remain a challenge and need to be implemented.
Koresh (2019)	Hough transform	Capsule neural network	Indian traffic sign data set	15% higher accuracy than CNN and RNN	Efficiency of CapsNet is more due to the ability to identify pose and spatial variance better than CNNs.
Hegadi (2011)	Shape and colour-based detection	Pre-trained CNN	German Traffic Signs Recognition Benchmark (GTSRB)	96%	Traffic sign recognition system for driver assistance is deployed on the NVIDIA Jetson TX1.

METHOD

The methodology can be divided into four phases: data collection, annotation, training, and implementation. The series of steps involved is shown in Figure 1.

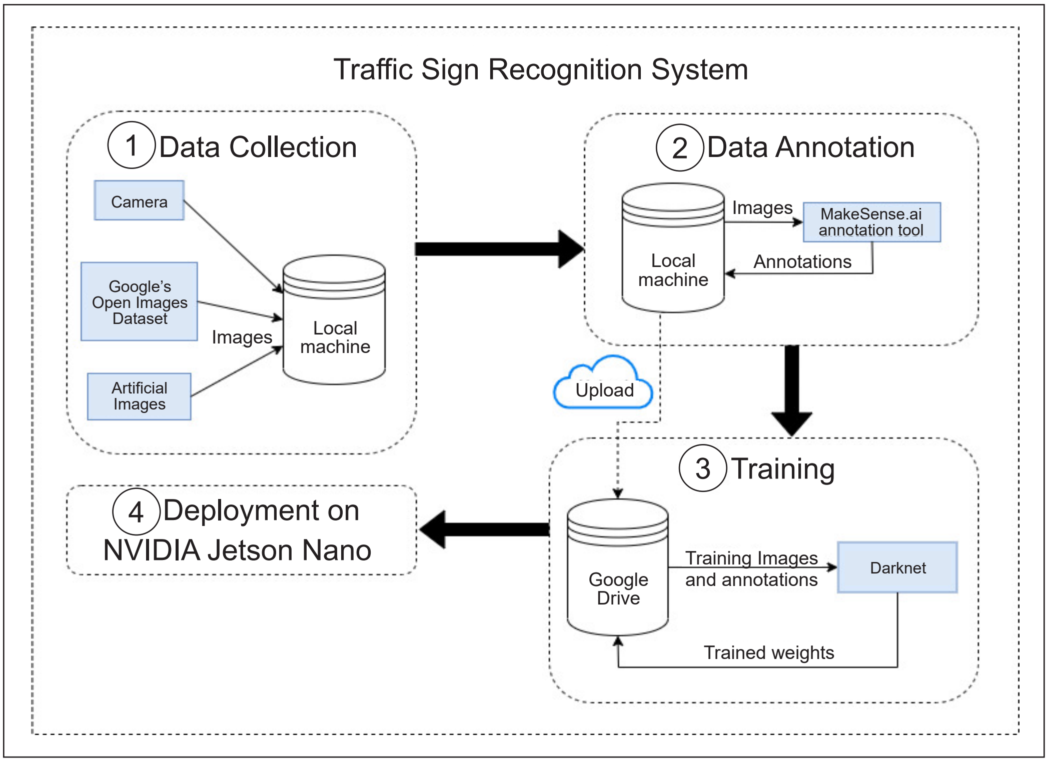


Figure 1. Traffic Sign Recognition System

Data Collection and Annotation

The study is focused on Indian road traffic signs. Due to the non-availability of a public dataset on Indian road traffic signs, a dataset of about 5800 images across 30 classes is created by capturing images and videos using mobile phones. Among the 30 classes, the 10 most commonly found classes with a sufficient number of images are considered for our study. The various classes considered for the study and the number of images are summarised in Table 2.

Training

The images are annotated manually according to the YOLO format and split at a ratio of 75:25 into training and validation sets. In addition, YOLOv4 performs data augmentation during the training process by varying image parameters such as saturation, exposure, and hue (known as colour jittering), as shown in Figure 2.

Table 2
Classes, their corresponding traffic sign image, and the number of images

Traffic Sign	Class Name	Number of images	Traffic Sign	Class Name	Number of images
	Stop	425		Speed Limit 20 km/h	285
	Left Curve	401		Speed Limit 30 km/h	332
	Right Curve	434		Speed Limit 40 km/h	305
	No Parking	376		Speed Limit 50 km/h	320
	No U-Turn	385		Speed Limit 60 km/h	325

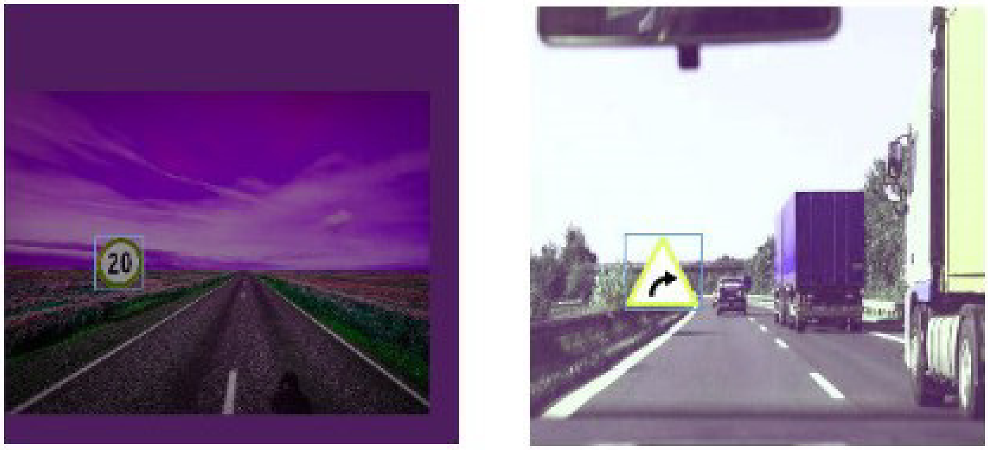


Figure 2. Data augmentation by colour jittering

Further, Mosaic data augmentation is performed, as shown in Figure 3, which combines four training images into one at certain ratios (Zoph et al., 2019). It encourages the model to learn how to detect traffic signs at different positions of the frame and on a smaller scale than usual.

Figure 3. Mosaic augmentation of training images in YOLOv4

System Design

The proposed design of the system is shown in Figure 4. First, a live video feed of the road is continuously captured by a camera fixed to the moving vehicle. Then, each frame that is captured by the camera is fed into the YOLOv4 detection module, deployed on the Jetson Nano. The detection module detects the traffic sign, if any, in the input frame. Next, the class name of the detected traffic sign is extracted and provided as input to the Text-to-Speech module, which generates an audio message that is given as output.

The YOLOv4 object detection model is trained on GPU, using the Darknet framework. The training is performed for 5000 epochs, and a mAP (mean Average Precision) of 87.22% is achieved on the validation set. The trained model is deployed on the NVIDIA Jetson Nano developer kit. YOLOv4 is computationally intensive and requires high-end GPUs for better performance. Even though YOLOv4-Tiny is lighter than YOLOv4, the obtained frame rate was far from real-time performance standards. Thus, Darknet implementation is not ideal for deployment on edge computing devices.

A TensorRT engine with FP16 precision is generated for the trained model for TensorRT acceleration. FP16 precision is chosen over FP32 since it would improve speed (TFLOPS), performance and reduce memory usage. With the help of TensorRT acceleration, the frame rate obtained is higher than that obtained using Darknet. However, it is insufficient for deployment in real-time use case. For example, TensorRT engines for YOLOv4-Tiny did not perform better than Darknet and showed a drop in the FPS values.

The model was deployed using the NVIDIA DeepStream 5.1 SDK to achieve real-time inference. DeepStream provides extensive support for object detection and segmentation

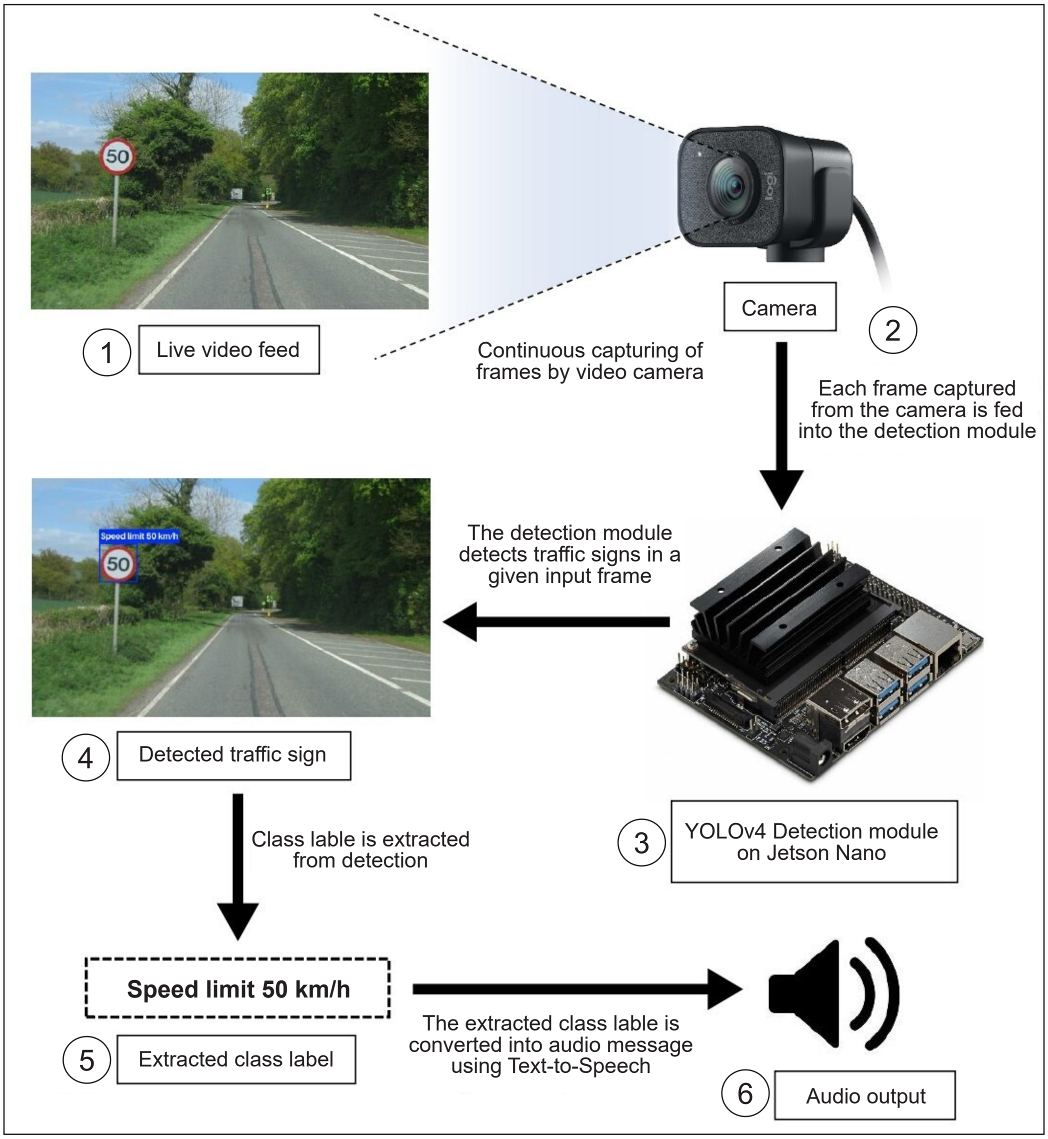


Figure 4. Proposed system design

models and offers exceptional throughput. Even though the frame rate of the YOLOv4 model is moderate, the YOLOv4-Tiny gave exceptional results. The frame rate obtained using YOLOv4-Tiny is on par with the real-time implementation requirements; hence DeepStream performs better than Darknet and TensorRT.

The next step is to translate the detected traffic signs into audio messages. The audio message corresponding to the detected traffic sign would be loaded from the secondary storage when a detection occurs and is given as output. It reduces the overhead of generating audio during execution, slowing down the processing of each frame.

RESULTS AND DISCUSSION

The model was tested with multiple network resolutions of 320 × 320, 416 × 416, 512 × 512 and 608 × 608. Mean Average Precision (mAP) is chosen as the performance evaluation metric, which is a precision averaged over all classes of traffic signs. The mAP values were obtained by testing the model on a test set of 297 images. Table 3 shows comparative results of mAP and frame rate for various network resolution sizes using Darknet, TensorRT and DeepStream 5.1.

Table 3
Performance comparison in terms of mAP and frame rate of different models

Model	Network Resolution	mAP^{50}	Frame rate (FPS)		
			Darknet	TensorRT	DeepStream
YOLOv4	320 × 320	67.83	2.4	4.9	7.5
YOLOv4	416 × 416	70.82	1.6	3.6	4.9
YOLOv4	512 × 512	72.65	1.3	3.0	4.0
YOLOv4	608 × 608	68.80	0.8	2.1	2.5
YOLOv4-Tiny	320 × 320	54.68	12.4	11.2	53.7
YOLOv4-Tiny	416 × 416	67.82	11.5	10.2	38.2
YOLOv4-Tiny	512 × 512	71.21	10.5	9.3	32.5
YOLOv4-Tiny	608 × 608	76.55	7.3	7.6	20.5

From analysing the results, it is seen that the mAP value of the model increases as the network resolution size increases. We observe a deviation from this trend for the highest resolution of YOLOv4 at 608 × 608 due to an increase in false positives because of duplicate bounding boxes. The highest mAP value achieved is 76.55% by YOLOv4-tiny at 608 × 608. The frame rate gradually decreases for both models as the network resolution increases. The reason behind this trend is that many features in the input image reduce the speed. However, the accuracy is better for higher input sizes.

An optimal architecture can be chosen depending on the practical TSR application requirements. For example, if faster processing is required, Tiny YOLOv4 architecture can be preferred over YOLOv4 architecture as it utilises less memory. However, if accuracy is an important criterion, YOLOv4 architecture can be preferred over Tiny YOLOv4. The memory consumption in YOLOv4 is quite high as it is computationally intensive but gives better accuracy. In this study, the focus is on building a real-time system. Thus, considering the importance of processing speed in this case, YOLOv4-Tiny architecture using DeepStream at 512 × 512 is chosen as the optimal architecture, which gives a mAP of 71.21% and 32.5 FPS, yielding about three times better frame rate than Darknet and TensorRT.

Even though the model is trained on limited data, it performs well in challenging conditions, as depicted in Figure 5. The sign in Figures 5(a) and 5(b) have been detected with good accuracy despite occlusion and deterioration of the traffic signs. The model also performs well at night as well as in rainy conditions, as shown in Figures 5(c) and 5(d), 5(e) and 5(f), respectively.

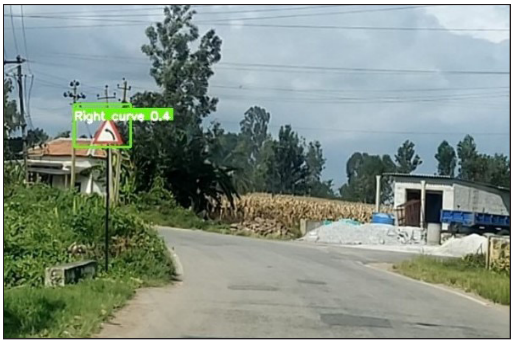
However, the model does not perform very well in detecting traffic signs that are small and far away, and the samples for the same condition are shown in Figure 6. The detector shows a moderate performance for the speed limit traffic signs in comparison with the other five classes, as seen in Figure 6(a). It is because classes of speed limits are very

Figure 5. Detection samples: (a) and (b) are in dim lighting conditions; (c) and (d) are during night-time; (e) and (f) are in rainy weather conditions

similar to each other. Thus, with training images as low as 300, the model has difficulty distinguishing the speed limit signs.

Due to the fairly small size and distant image of the traffic sign, detection is either missed, as in the case of Figure 6(c) or resulted in false detection, as seen in Figure 6(b). Deterioration of the traffic sign, as seen in Figure 6(d), makes it difficult for the detector to detect the sign; hence a missed detection is observed. Despite some failed detections, the model performs well in many complex conditions and is robust against scale changes. The attempt to provide output in the form of audio messages was also feasible. Thus, the model can detect 10 classes of urban traffic signs at a decent accuracy in real-time and provide audio alerts to the driver.

(a)



(b)

(c)

(d)

Figure 6. Detection samples: (a) and (b) are wrong class predictions; (c) and (d) are no class predictions

CONCLUSION

This paper discusses the implementation of a Traffic Sign Recognition System for Indian roads on the NVIDIA Jetson Nano using YOLOv4 architecture. The proposed solution is a good choice to be used as a Traffic Sign Recognition System as a part of ADAS, where it gives accuracy in the range of 54.68–76.55% for different input network resolutions. This module is integrated with the audio module to give the final output as an audio message.

The YOLOv4-Tiny model with DeepStream implementation was able to achieve an FPS of 32.5, which is on par with real-time detection requirements. However, despite its good overall performance, the ideal performance is yet to be achieved due to some missed and false detections. It can be improved by fine-tuning the neural network and expanding the training dataset. Furthermore, the designed system can be integrated with other subsystems like lane detection, navigation system, and obstacle detection with hardware assistance to form a complete ADAS.

ACKNOWLEDGEMENTS

The authors thank NVIDIA pvt Ltd. for donating the Jetson Nano Developer kit used to conduct this project.

REFERENCES

Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). *YOLOv4: Optimal speed and accuracy of object detection*. arXiv Preprint. http://arxiv.org/abs/2004.10934

Do, H. N, Vo, M. T., Luong, H. Q., Nguyen, A. H., Trang, K., & Vu, L. T. K. (2017). Speed limit traffic sign detection and recognition based on support vector machines. In *2017 International Conference on Advanced Technologies for Communications (ATC)* (pp. 274-278). IEEE Publishing. https://doi.org/10.1109/ATC.2017.8167633

Ellahyani, A., El Ansari, M., El Jaafari, I., & Charfi, S. (2016). Traffic sign detection and recognition using features combination and random forests. *International Journal of Advanced Computer Science and Applications, 7*(1), 686-693. https://doi.org/10.14569/IJACSA.2016.070193

Hasegawa, R., Iwamoto, Y., & Chen, Y. W. (2019). Robust detection and recognition of japanese traffic sign in the complex scenes based on deep learning. In *2019 IEEE 8th Global Conference on Consumer Electronics (GCCE)* (pp. 575-578). IEEE Publishing. https://doi.org/10.1109/GCCE46687.2019.9015419

Hegadi, R. S. (2011). Automatic traffic sign recognition. In *Proceedings of International Conference on Communication, Computation, Management & Nanotechnology* (pp. 1-5). REC Bhalki.

Islam, M. T. (2019). Traffic sign detection and recognition based on convolutional neural networks. In *2019 International Conference on Advances in Computing, Communication and Control (ICAC3)* (pp. 1-6). IEEE Publishing. https://doi.org/10.1109/ICAC347590.2019.9036784

Koresh, M. H. J. D. (2019). Computer vision based traffic sign sensing for smart transport. *Journal of Innovative Image Processing, 1*(1), 11-19. https://doi:10.36548/jiip.2019.1.002

Lu, X., Kang, X., Nishide, S., & Ren, F. (2019). Object detection based on SSD-ResNet. In *2019 IEEE 6th International Conference on Cloud Computing and Intelligence Systems (CCIS)* (pp. 89-92). IEEE Publishing. https://doi.org/10.1109/CCIS48116.2019.9073753

Luo, H., Yang, Y., Tong, B., Wu, F., & Fan, B. (2018). Traffic sign recognition using a multi-task convolutional neural network. *IEEE Transactions on Intelligent Transportation Systems, 19*(4), 1100-1111. https://doi.org/10.1109/TITS.2017.2714691

Muthukumaresan, T., Kirubakaran, B., Kumaresan, D., Satheesan, A., & Prakash, A. J. (2016). Recognition of traffic sign using support vector machine and fuzzy cluster. *International Journal of Science Technology & Engineering, 2*(10), 190-197.

Oltean, G., Florea, C., Orghidan, R., & Oltean, V. (2019). Towards real time vehicle counting using YOLO-Tiny and fast motion estimation. In *2019 IEEE 25th International Symposium for Design and Technology in Electronic Packaging (SIITME)* (pp. 240-243). IEEE Publishing. https://doi.org/10.1109/SIITME47687.2019.8990708

Sari, A., & Cibooglu, M. (2018). Traffic sign detection and recognition system for autonomous RC cars. In *2018 6th International Conference on Control Engineering & Information Technology (CEIT)* (pp. 1-5). IEEE Publishing. https://doi.org/10.1109/CEIT.2018.8751898

Tabernik, D., & Skocaj, D. (2020). Deep learning for large-scale traffic-sign detection and recognition. *IEEE Transactions on Intelligent Transportation Systems, 21*(4), 1427-1440. https://doi.org/10.1109/TITS.2019.2913588

Zhang, J., Huang, M., Jin, X., & Li, X. (2017). A real-time Chinese traffic sign detection algorithm based on modified YOLOv2. *Algorithms, 10*(4), Article 127. https://doi.org/10.3390/a10040127

Zhong, Z., Zheng, L., Kang, G., Li, S., & Yang, Y. (2020). Random erasing data augmentation. *Proceedings of the AAAI Conference on Artificial Intelligence, 34*(07), 13001-13008. https://doi.org/10.1609/aaai.v34i07.7000

Zoph, B., Cubuk, E. D., Ghiasi, G., Lin, T. Y., Shlens, J., & Le, Q. V. (2019). *Learning data augmentation strategies for object detection*. arXiv Preprint.