Advances in Artificial Intelligence Research (AAIR)
ISSN 2757-7422
www.dergipark.com/aair/

Vol. 3 (No. 2), pp. 66-75, 2023
doi: 10.54569/aair.1184569
Published online: Oct 23, 2023

Development of a Traffic Speed Limit Sign Detection System Based on Yolov4 Network

Semih Selçuk [1], Sefa Beker [1,*] Ömer Faruk Boyraz [1]

[1] Anadolu Isuzu Automotive Industry and Trade Inc., R&D Center, Kocaeli, Turkey

Abstract

Recent advancements in artificial intelligence (AI) technologies have hastened the shift towards intelligent systems within the automotive industry. These systems enable the prevention of driver-related errors and accidents, as well as the provision of crucial information to drivers by proactively detecting road conditions. The present study focuses on the development of an AI-based system designed to furnish drivers with information regarding speed limit signs on the road, thereby enhancing traffic safety. The YOLOv4 model was employed in this system to achieve exceptional speed and accuracy levels. Following model training, rigorous validation was conducted, resulting in a remarkable test accuracy of 98%.

Keywords: *Speed Limit Sign, Object Detection, Deep Learning, YoloV4, Jetson Nano.*

1. Introduction

Studies on driver assistance systems and smart driving have been on the rise in recent times. Among these systems, the Intelligent Speed Assistance (ISA) system holds significance. According to the ISA regulation outlined in the General Vehicle Safety Regulation (EU) 2019/2144 [1], motor vehicles falling under categories M and N (such as buses and trucks) must be equipped with intelligent speed assistance systems, which will be mandated across all EU member states.

The importance of mechanical, electronic, and software advancements in enhancing driving safety cannot be overstated. Manufacturers are actively working to safeguard the lives and properties of drivers, passengers, and pedestrians alike [2]. In this regard, international authorities play a crucial role in fostering a shared understanding of security among manufacturers by establishing the general framework for these systems through relevant regulations [3].

Recent advancements in AI technologies have accelerated the integration of smart systems into transportation systems. These systems are aimed at preventing driver-related errors and accidents while providing advanced road condition detection to inform the driver proactively [4]. Modern vehicles now incorporate sophisticated driver assistance systems such as blind-spot detection, forward collision warning, driver fatigue detection, and intelligent speed assistance systems. Among these, the intelligent speed assistance system (ISA) stands out as a crucial component in providing speed-related information to the driver, ultimately ensuring a safer journey [5].

In the past, traffic sign detection relied on traditional computer vision algorithms. These algorithms typically utilized traffic sign features like color and shape and extracted various attributes from the signs. HSV (Hue, Saturation, Value) color space was often preferred over RGB due to its resilience to lighting variations. Machine learning techniques like template matching, support vector machines, or artificial neural networks were used to classify the signs based on the extracted features. However, these methods proved to be less efficient and slower in detecting traffic signs successfully [6].

Today, deep learning algorithms, specifically convolutional neural networks (CNNs), have revolutionized image recognition and object detection tasks. These deep learning methods enable quicker and more accurate detection by automatically extracting essential features from the target images without the need for extensive preprocessing [7].

In general, traffic speed sign recognition systems consist of two main components: sign detection and sign classification. Detecting the traffic speed signs accurately and swiftly, especially the smaller signs on the road, is a critical aspect of such systems. Numerous methods have been proposed in the literature to address this challenge, with SSD [8], Faster R-CNN [9], and Yolo algorithms [10] being among the most popular ones. The Yolo algorithm, in particular, stands out as it strikes a balance between speed and accuracy, making it highly suitable for this task.

Researchers have conducted extensive studies on traffic sign detection, often utilizing open source datasets due to the time-consuming nature of collecting traffic speed sign data. Some of the most well-known datasets used in these studies include the German Traffic Sign Dataset [11], Tsinghua Tencent Dataset [12], and the

*Corresponding author
E-mail address: sefabeker@gmail.com

Received: 6/Oct/2022; Received in revised form: 31/Jul/2023; Accepted: 19/Sep/2023.

Chinese Traffic Sign Datasets [13].

In their study, Rajendran et al. employed the YoloV3 model along with the German Traffic Sign Detection Benchmark Dataset. They conducted model training and evaluation using the Keras and Tensorflow frameworks, leveraging an Nvidia GTX 1060 GPU with 6 GB memory. The YoloV3 model achieved a speed of 10 frames per second (FPS) and demonstrated a satisfactory performance rate of 92.2% [14].

In contrast, Zuo et al. utilized the Faster R-CNN model for traffic sign detection in their study. The Faster R-CNN model training resulted in a mean average precision (mAP) value of 0.34493. Although the obtained results were not deemed satisfactory, the researchers emphasized the need for substantial efforts to enhance the model's performance [15].

This study presents the development of a cost-effective system designed to provide real-time speed limit information to drivers within vehicles. The system is specifically tailored to operate on the low-cost Jetson Nano embedded computer, rendering it feasible for integration into portable and commercial vehicles. In order to enhance the dataset utilized in the system's development, we supplemented open-source datasets with our own collected data. Notably, images containing speed limit signs were identified and extracted from the road images collected by Anadolu Isuzu test vehicles under nighttime conditions, thereby enriching the training dataset. Consequently, the system's nighttime performance has been substantially improved, as nocturnal images are often inadequately represented in open-source datasets.

The model training in this study involved multiple datasets, including the German Traffic Sign Dataset, Tsinghua Tencent Dataset (TT100K Dataset), Chinese Traffic Sign Dataset, and a dataset created by Google Maps and Anadolu Isuzu test vehicles. By leveraging the YoloV4 model, the researchers achieved real-time detection of traffic speed signs ranging from 20 km/h to 120 km/h with high performance.

The article is structured into five main sections. The introduction offers a comprehensive overview of the study. Section 2 encompasses an exploration of related studies in the literature. Subsequently, Section 3 outlines the dataset creation process, as well as the hardware and software methods adopted in the study. In the fourth section, the results of the software method are presented, which showcase the performance and accuracy of the developed system. Finally, the conclusion effectively summarizes the key findings and underscores the contributions made by the study.

2. Related Works

In recent years, significant progress has been made in traffic sign detection and intelligent driving systems, leveraging the advancements in artificial intelligence technologies. Numerous studies have explored different algorithms and approaches to improve the accuracy and real-time performance of traffic sign recognition systems. In this section, we present a review of relevant works in the literature, and compare the YoloV4 model used in our study with Faster R-CNN and SSD algorithms, which are widely recognized in the field of object detection for traffic sign recognition:

Rajendran et al. [14]: Rajendran et al. utilized the YoloV3 model in their study, employing the German Traffic Sign Detection Benchmark Dataset. They achieved a remarkable model speed of 10 FPS and an accuracy of 92.2%. While their results were promising, we sought to further enhance the performance of traffic sign detection with the improved YoloV4 model on a diverse dataset, which includes challenging night conditions.

Zuo et al. [15]: In their research, Zuo et al. adopted the Faster R-CNN model for traffic sign detection. However, their Faster R-CNN model yielded an mAP value of 0.34493, indicating suboptimal accuracy. Our comparative analysis confirmed the limitations of the Faster R-CNN model in real-time applications, particularly for traffic sign recognition.

In their research, Gao et al. [16] investigated the recognition of traffic signs, which involves two critical stages: detection and classification. For this study, the SSD (Single Shot MultiBox Detector) object detection algorithm was adopted to effectively identify traffic signs. The SSD model, a convolutional neural network, efficiently utilizes multiple feature maps for object detection. To address the challenges posed by the relatively small size of traffic signs compared to the entire image, the SSD model was further enhanced. This improvement encompassed model simplification and adjustments to the prior box sizes, resulting in superior detection performance, particularly for small targets. Extensive experiments were conducted on the test dataset, specifically the German Traffic Sign Dataset, highlighting the remarkable proficiency of the proposed algorithm in handling single-target, multi-target, and diverse lighting conditions. Notably, the precision and recall achieved on the test dataset were 91.09% and 88.06%, respectively.

Jiang et al. [17] proposed an enhanced traffic sign recognition method based on YOLOv3. They employed depthwise separable convolution to reduce parameters and computational complexity while maintaining accuracy. By replacing MSE loss with GIoU loss and incorporating Focal loss, they improved optimization and addressed class imbalance. Experiments on the TT100K dataset showed significant performance gains, achieving 89% mAP with reduced parameters and increased FPS, effectively improving detection speed and accuracy.

Shan et al. [18] conducted a study to enhance traffic sign detection using the SSD model, specifically ssd_300, in the context of Chinese road conditions. Training the model on the Chinese Traffic Sign Dataset, they achieved an improved mAP of 0.85 on the test dataset, outperforming ssd_300 by 0.13, while maintaining real-time detection capability. The model exhibited proficiency in detecting three categories of Chinese traffic signs and demonstrated strong robustness against different disturbances.

In conclusion, the reviewed literature emphasizes the importance of selecting appropriate algorithms for traffic sign detection, where the YoloV4 model stands out as a powerful solution, offering an optimal balance between speed and accuracy. Our study aims to build upon the existing research and contribute to the advancement of intelligent driver assistance systems through our carefully chosen YoloV4 model for real-time traffic sign recognition.

3. Materials and Methods

3.1 Dataset

In the study, a comprehensive dataset consisting of a total of 22,000 traffic speed sign images was compiled for training and evaluation purposes. Among these, 4,400 images were reserved for testing, and the remaining 17,600 images were used for training the model. The dataset covered speed limit signs ranging from 20 km/h to 120 km/h and encompassed various background, weather, and lighting conditions to ensure robustness. To ensure diversity and realism in the dataset, the researchers collected data from multiple sources. Approximately 50% of the data was collected using Anadolu Isuzu test vehicles on urban and interurban roads under various conditions, including day and night, rainy, sunny, and snowy weather. Approximately 25% of the data was obtained through screen capture using the Google Maps platform. The remaining portion was collected from the German Traffic Sign Dataset, Tsinghua Tencent Dataset, and Chinese Traffic Sign Datasets, specifically focusing on the recognition of speed limit signs.

Figure 1 in the study presents some sample images from the prepared dataset, providing a visual representation of the different traffic speed signs collected under various conditions.The large and diverse dataset allowed the researchers to effectively train and evaluate the performance of the YoloV4 model in real-time traffic speed sign detection, covering a wide range of speed limits and environmental scenarios.

Figure 1. *Traffic speed sign sample images collected from different weather-environment conditions*

3.2. Materials

In the deep learning-based system, artificial intelligence model trainings were carried out on a computer equipped with Intel Core i7-vPRO, Nvidia GeForce RTX 2080 GPU, and 32GB RAM. Developed models have been tested in real time on both computer and embedded device. Nvidia Jetson Nano embedded device (**Figure 2**) used in the designed system has 4GB 64-bit LPDDR4 25.6GB/s memory, 4-core ARM A57 CPU and 128 CUDA core NVIDIA Maxwell™ GPU. With Nvidia Jetson Nano, real-time images from the camera are given as input to deep learning-based models. In addition CSI camera with Sony IMX219 sensor with 77 degree angle of view was used for creating traffic speed sign image dataset and for real-time speed limit sign detection.

Figure 2. *Nvidia Jetson Nano embedded device and CSI Camera [7->19]*

3.3. Method

After the images were collected, the traffic speed signs were labeled, and made ready for model training using the LabelImg toolbox.

The collected dataset was annotated using the open-source image labeling program, labelImg. Subsequently, data augmentation techniques were employed during the artificial intelligence training process of the developed system. These augmentation procedures encompassed angle changes, scaling, saturation, exposure, and hue adjustments to augment the dataset. While angle changes and scaling operations were integrated into the model training, the remaining parameters introduced random variations in terms of color saturation, brightness (exposure), and hue of the images, thereby enhancing the model's robustness and adaptability under diverse conditions. These enhancements contributed to the model's ability to operate with high performance, irrespective of environmental and lighting variations. The parameters, along with their corresponding values, were configured within the yolov4.cfg configuration file, as detailed in Table 1. To account for the relatively small area occupied by speed limit signs within the images, an image resolution of 608x608 was set. While higher resolutions in multiples of 32 could be employed, the resolution of 608x608 was deemed optimal in terms of both speed and accuracy, as higher resolutions may result in performance slowdowns for the model.

Table 1. *Data Augmentation Parameters for Network Training*

Parameter Name	Parameter Value
Width x Height (pixel)	608
Angle (degree)	5
Saturation	2
Exposure	2
Hue	0.1

The data prepared for the training were made ready as 20% test and 80% training set. To validate the performance impact on the created dataset, the system was compared using three different deep learning models, namely SSD and Faster RCNN, in addition to YOLOv4, as part of the study. By evaluating the speed (Frames Per Second - FPS) and accuracy (mean Average Precision - mAP) of these three distinct deep learning models, the system's performance was duly verified and confirmed.

3.3.1 SSD Algorithm

The SSD (Single Shot Multibox Detector) is a deep learning algorithm used for object detection in videos and images [8]. Similar to YOLO, SSD provides fast and accurate prediction performance. When compared to other algorithms such as Faster R-CNN and Mask R-CNN [20], SSD is preferred for its faster inference time and improved accuracy. The key feature that sets the YOLO algorithm apart from others is that it processes the entire image only once. The fundamental principle of the SSD algorithm is to use anchor boxes of various sizes and aspect ratios to detect objects in the image. These anchor boxes are fixed-size boxes that are employed to detect objects of different sizes and aspect ratios. The SSD architecture consists of the following steps: Input Image: SSD takes an input image for object detection. VGGNet and Feature Extractor: SSD is typically based on a pre-trained network like VGGNet, which is used to transform images into feature maps. The feature extractor network is utilized to obtain feature maps at different depths, which are essential for detecting objects. Feature Maps and Anchors: SSD makes predictions for object presence based on the anchor boxes present in the feature maps. Each anchor box predicts the class and location of potential objects using the features within it. Classification and Localization: SSD performs classification to determine the class of the objects and localization to accurately adjust their positions. Non-maximum Suppression (NMS): To handle the issue of multiple bounding boxes detecting the same object, SSD applies non-maximum suppression to remove redundant bounding boxes. In conclusion, the SSD algorithm accomplishes object detection efficiently by processing the image only once. Due to its real-time capabilities and high-performance nature, SSD is often employed in applications requiring fast object detection, including the detection of traffic speed signs. The structure of the SSD algorithm is shown in **Figure 3**.

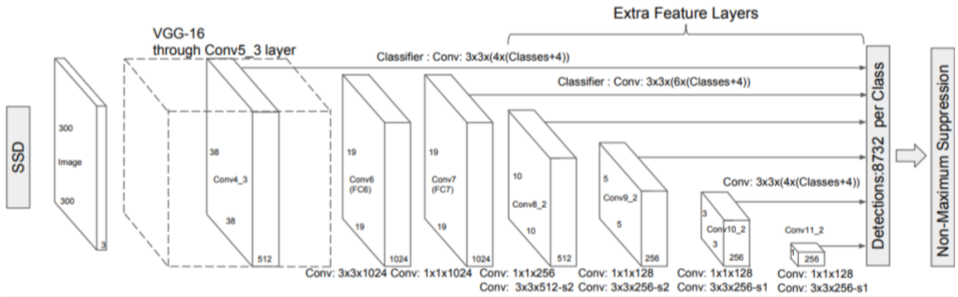

Figure 3. *Structure of the SSD algorithm [8]*

3.3.2 Faster RCNN

Faster R-CNN is a deep learning model developed for object detection tasks [9]. It offers higher detection accuracy compared to other algorithms like YOLO and SSD. It consists of two main components: a feature extractor network (backbone network) and a region proposal network (RPN). Feature Extractor Network (Backbone Network): Faster R-CNN utilizes a pre-trained convolutional neural network (CNN) such as VGGNet, ResNet, or other similar architectures as the feature extractor network. This network transforms the input image into feature maps, identifying important characteristics of objects. Region Proposal Network (RPN): Faster R-CNN employs a region proposal network (RPN) to suggest object candidate regions. RPN uses sliding windows on the feature maps to determine regions that potentially contain objects. These regions are later further examined in detail. The working principle of Faster R-CNN is as follows: Input Image: Faster R-CNN takes an input image for object detection. Feature Extractor: The input image is fed to the feature extractor network, generating feature maps. Region Proposals: RPN suggests object candidate regions using the feature maps. These regions represent areas that may contain objects and are further analyzed. Region-based CNN (RCNN): The proposed object regions are matched with the actual features in the feature maps and then provided to a Region-based CNN (RCNN). The RCNN is used to determine the class of each object region and refine the precise position of its bounding box. The structure of the Faster R-CNN algorithm is shown in **Figure 4**.

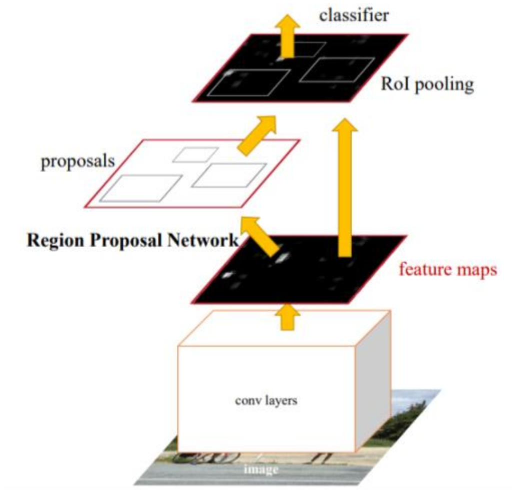

Figure 4. *Structure of the Faster R-CNN [9]*

Faster R-CNN provides high detection accuracy by effectively suggesting object candidate regions through the region proposal network and obtaining better features through the feature extractor network. As a result, it is a popular algorithm used successfully in more complex and detailed object detection tasks.

3.3.3 YoloV4 Algorithm

YOLO (You Look Only Once) is a deep learning algorithm that is built on convolutional neural networks and can detect objects from videos and images. Compared to algorithms such as Faster R-CNN and Mask R-CNN, the YoloV4 algorithm is preferred because it has faster and more accurate prediction performance. The biggest feature that distinguishes the Yolo algorithm from these algorithms is that it processes the image once. The structure of the YoloV4 algorithm, which consists of 106 layers, is shown in **Figure 5**. This structure consists of the input layer, feature extraction layers (Backbone-Neck and Dense Prediction), respectively.

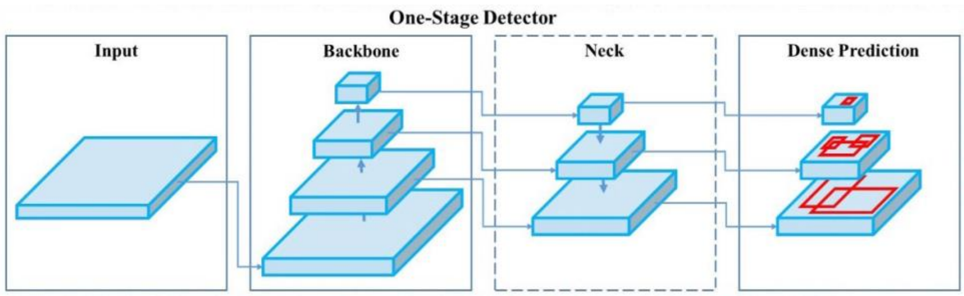

Figure 5. *Structure of the Yolov4 algorithm [21]*

Unlike the Yolov3 version, it includes features such as data augmentation, random image cropping, image blurring functions with CutMix and Mosaic functions. Thanks to these features, the variability of the input image can be increased so that the model is more robust against images obtained in different environments.

Yolov4 algorithm as feature extractor in backbone stage VGGNet [22] It uses the CSPDarknet-53 backbone shown in **Figure 6** to improve the learning capacity of convolutional neural networks similar to CNN architecture.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 6. *Darknet-53 structure [23]*

4. Results

The data generated for traffic speed sign detection are divided into test and training data sets. Then, Intel Core i7-vPRO was trained with the YoloV4 algorithm on a computer with Nvidia GeForce RTX 2080 GPU hardware. 300 images not included in the datasets were used to test the training process.

To thoroughly evaluate the detection performance on a consistent dataset, this study compares various models, including SSD and Faster R-CNN from the literature, using metrics such as mAP (mean Average Precision) and FPS (Frames Per Second). The number of recognized frames is employed as the evaluation metric for the detection efficacy. The experimental findings are presented in **Table 2**. According to the values seen in the table, Although the SSD model is faster, the accuracy of the model and the correct prediction of traffic signs are much more important. On the other hand, while the Faster R-CNN model achieved high accuracy rates, it performed quite poorly in terms of speed. As a result, the proposed Yolov4 algorithm exhibited a higher success rate and demonstrated a highly satisfactory Frames Per Second (FPS) in terms of speed.

Table 2. *Detection Performance Comparison of Various Mod*els

Model	mAP (%)	FPS (frame/second)
Faster R-CNN	93	7
SSD	68	**51**
YoloV4	**95**	42

The model was evaluated using various metrics, and the test results for the traffic speed sign dataset were analyzed with the aid of the confusion matrix presented in **Table 3**.

Table 3. Confusion Matrix

	Positive	Negative
Positive	True Positive (TP)	False Negative (FN)
Negative	False Positive (FP)	True Negative (TN)

The Confusion Matrix is a table used to evaluate the performance of the object detection model on test data with known true values. The adapted version of the matrix for this study is presented in **Table 4**.

Table 4. *Confusion Matrix Explanation.*

Terms	Results
True Positive (TP)	Ex: 50 km/h speed sign detected and it is correct
True Negative (TN)	Presumed to have no speed sign in the image and it's true
False Positive (FP)	Ex: 50 km/h speed sign detected and this is wrong
False Negative (FN)	Presumed to have no speed sign in the image and this is incorrect

Precision Rate is one of the evaluation metrics utilized in object detection tasks. It represents the ratio of correctly predicted objects to all predicted objects. The figure displays the sum of True Positives (TP) and False Positives (FP) for all positive cases as assessed by the model. The Precision is calculated as the division of the number of True Positive cases by the number of all positive cases, and it is represented by Eq. (1).

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{1}$$

The Recall Rate (Recall) is utilized to assess the model's capability to detect true instances within the test set. It is calculated as the sum of True Positives (TP) and False Negatives (FN) for all positive samples in the test set. The Recall Rate is expressed as the ratio of the number of true positive cases to all positive samples in the test set, as shown in Eq. (2).

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{2}$$

The F1 Score value represents the harmonic average of the Precision and Recall values. It is calculated according to Eq. (3).

$$F1\ Score = 2 * \frac{Precision * Recall}{\mathrm{Precision} + \mathrm{Recall}} \tag{3}$$

The higher the F1 value, the more effective the model's testing will be [24]. The mAP (mean average precision) value is a commonly used metric for accuracy assessment in object detection models, representing the average of the average precision (AP) for each object class [25]. The IoU (Intersection over Union) value is another metric used to evaluate object detection systems. Typically, a prediction is considered correct when the IoU is greater than or equal to 0.5. For this study, the IoU threshold was set to 0.5.

The Yolov4 algorithm has been customized based on the dataset. Accordingly, the resizing is set to

608x608. The filters parameter is determined as 3*(5+classes), and for 4 classes, it is set to 48. The number of iterations is set to 30000, and in each iteration, 64 images are utilized, with each image divided into 8 grids. The learning rate parameter is set as 0.0013.

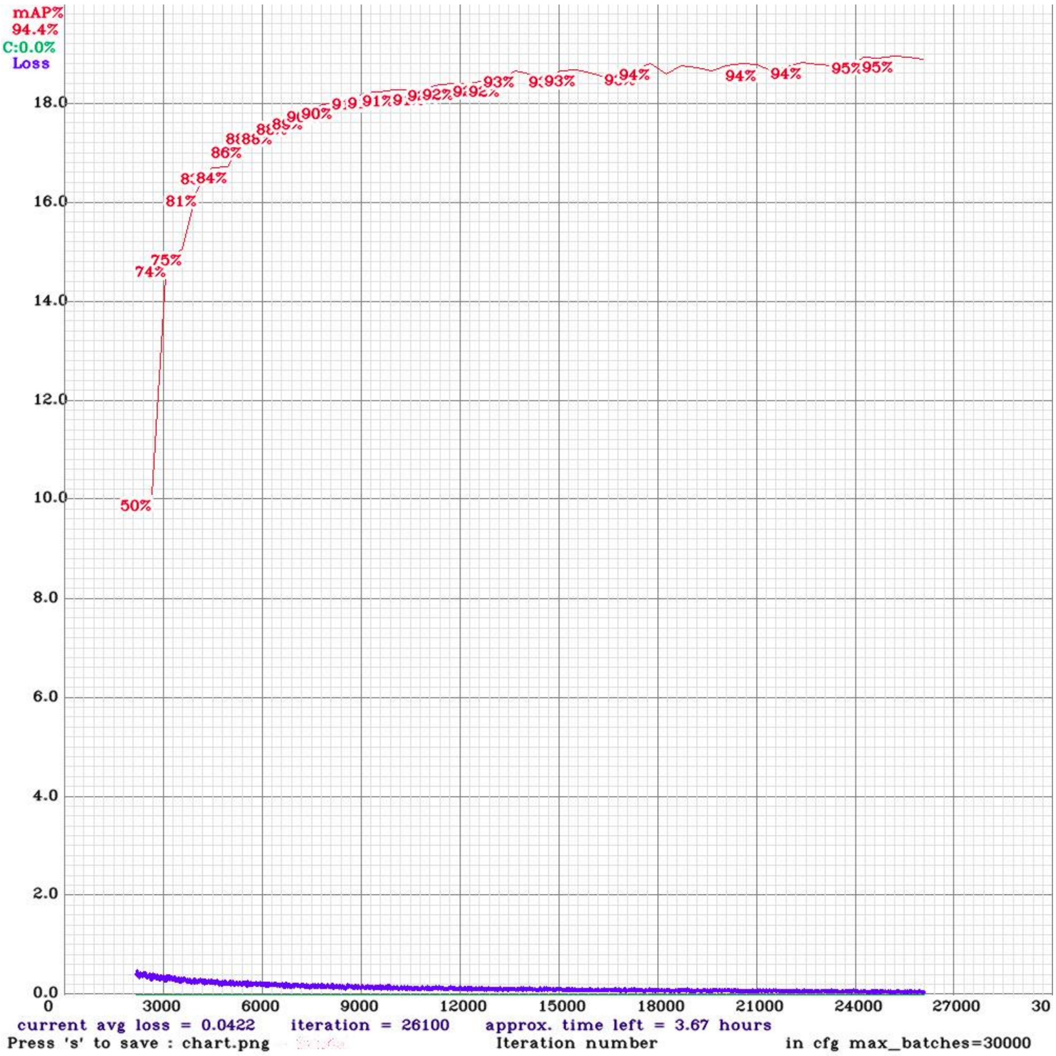

Figure 7. *Yolov4 Performance in a dataset with 11 Traffic Speed signs*

When examining the training result graph (**Figure 7**), it is observed that the error rate gradually decreases as the number of iterations progresses. Consequently, the average precision value, calculated after 3000 iterations, starts at 70% and progressively reaches 95% as the training advances.

Following the application steps, the model's IoU, True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) values were examined to calculate the precision (Precision) of the model. Subsequently, the mAP and IoU values were determined. The evaluation results of the developed model are presented in **Table 5**.

Table 5. *Evaluation Results*

Evaluation Factors	Results
TP	274
TN	20
FP	2
FN	4
mAP	%95
Average IoU	%82
Precision	%99
Recall	%98
F1-score	%98

When evaluating the model results, it is observed that the F1-score value is 98%, the IoU value is 82%, and the mAP value is 95%. Based on the mAP result, it is concluded that the average sensitivity for each object class is adequate.

5.Conclusion

Mechanical, electronic, and software developments have gained paramount importance in enhancing driving safety within the automotive industry. For this reason, manufacturers undertake efforts to safeguard the lives and property of drivers, passengers, and pedestrians.

In this research, an artificial intelligence-based system was devised to furnish drivers with real-time information about speed signs on the road, with the primary goal of supporting traffic safety. A dataset consisting of 22,000 images was amassed for object detection, where 80% of these images were allocated as training data, and the remaining 20% served as test data.

The LabelImg Image Labeling Program was employed for annotating the objects, and the training process was executed on a computer equipped with an Intel Core i7-vPRO, Nvidia GeForce RTX 2080 GPU, and 32GB RAM using the Darknet Neural Network Framework. Python Programming Language facilitated result determination, and the OpenCV library was utilized for Image Processing algorithms. Python scripts were developed to detect objects in the images using the YOLOv4 Algorithm. Upon the completion of the study, 300 images were subjected to system evaluation, with 294 predictions accurately detected, indicating a 98% correctness rate. The performance metrics of the study yielded an mAP value of 0.95 and an IoU value of 0.82.

To effectively assess the detection performance of the YoloV4 model employed in the system, we additionally deployed the Faster R-CNN and SSD models, which are commonly used in the literature, on the same dataset. While the Faster R-CNN model exhibited commendable overall accuracy, its real-time performance was noticeably limited. Conversely, the SSD model ran faster than the YoloV4 model but demonstrated a significantly lower accuracy. Following the study, it was evident that the Yolov4 model displayed the most favorable performance in terms of both speed and accuracy.

References

[1] Type-approval requirements to ensure the general safety of vehicles and the protection of vulnerable road users [Online]. https://eur-lex.europa.eu/eli/reg/2019/2144/oj, (accessed: 29.09.2022)

[2] Parekh, D., Poddar, N., Rajpurkar, A., Chahal, M., Kumar, N., Joshi, G. P., & Cho, W. (2022). A review on autonomous vehicles: Progress, methods and challenges. Electronics, 11(14), 2162.

[3] Doecke, S. D., Raftery, S. J., Elsegood, M. E., & Mackenzie, J. R. (2021). Intelligent Speed Adaptation (ISA): benefit analysis using EDR data from real world crashes. Age, 26(21), 32.

[4] Dilek, E., & Dener, M. (2023). Computer vision applications in intelligent transportation systems: a survey. Sensors, 23(6), 2938.

[5] González-Saavedra, J. F., Figueroa, M., Céspedes, S., & Montejo-Sánchez, S. (2022). Survey of cooperative advanced driver assistance systems: from a holistic and systemic vision. Sensors, 22(8), 3040.

[6] Liu, C., Li, S., Chang, F., & Wang, Y. (2019). Machine vision based traffic sign detection methods: review, analyses and perspectives. Ieee Access, 7, 86578-86596.

[7] Turan, S., Milani, B., & Temurtaş, F. (2021). Different application areas of object detection with deep learning. Akıllı Ulaşım Sistemleri ve Uygulamaları Dergisi, 4(2), 148-164.

[8] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14 (pp. 21-37). Springer International Publishing.

[9] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28.

[10] Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2022). A Review of Yolo algorithm developments. Procedia Computer Science, 199, 1066-1073.

[11] Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2011, July). The German traffic sign recognition benchmark: a multi-class classification competition. In The 2011 international joint conference on neural networks (pp. 1453-1460). IEEE.

[12] Dewi, C., Chen, R. C., Liu, Y. T., Liu, Y. S., & Jiang, L. Q. (2020, June). Taiwan stop sign recognition with customize anchor. In Proceedings of the 12th International Conference on Computer Modeling and Simulation (pp. 51-55).

[13] Zhu, Z., Liang, D., Zhang, S., Huang, X., Li, B., & Hu, S. (2016). Traffic-sign detection and classification in the wild. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2110-2118).

[14] Rajendran, S. P., Shine, L., Pradeep, R., & Vijayaraghavan, S. (2019, July). Real-time traffic sign recognition using YOLOv3 based detector. In 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-7). IEEE.

[15] Zuo, Z., Yu, K., Zhou, Q., Wang, X., & Li, T. (2017, June). Traffic signs detection based on faster r-cnn. In 2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW) (pp. 286-288). IEEE.

[16] Gao, B., Jiang, Z., & zhang, J. (2019, July). Traffic sign detection based on ssd. In Proceedings of the 2019 4th International Conference on Automation, Control and Robotics Engineering (pp. 1-6).

[17] Jiang, J., Bao, S., Shi, W., & Wei, Z. (2020). Improved traffic sign recognition algorithm based on YOLO v3 algorithm. Journal of Computer Applications, 40(8), 2472.

[18] Shan, H., & Zhu, W. (2019, October). A small traffic sign detection algorithm based on modified ssd. In IOP Conference Series: Materials Science and Engineering (Vol. 646, No. 1, p. 012006). IOP Publishing.

[19] Jetson Nano [Online]. Available: https://www.nvidia.com/tr-tr/autonomous-machines/embedded-systems/jetson-nano/product-development/, (accessed: 29.09.2022)

[20] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 2961-2969).

[21] Bochkovskiy A. 2020. Yolo v4, v3 and v2 for Windows and Linux. https://github.com/AlexeyAB/darknet, Date of access: 29.09.2022

[22] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[23] Wang, C. Y., Liao, H. Y. M., Wu, Y. H., Chen, P. Y., Hsieh, J. W., & Yeh, I. H. (2020). CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops (pp. 390-391).

[24] Yu, J., & Zhang, W. (2021). Face mask wearing detection algorithm based on improved YOLO-v4. Sensors, 21(9), 3263.

[25] Hu, X., Liu, Y., Zhao, Z., Liu, J., Yang, X., Sun, C., ... & Zhou, C. (2021). Real-time detection of uneaten feed pellets in underwater images for aquaculture using an improved YOLO-V4 network. Computers and Electronics in Agriculture, 185, 106135.