Karbala International Journal of Modern Science

Manuscript 3373

Road Signs Detection Using SSD MobileNetV2

Zahraa Salah Dhaif

Nidhal K. El Abbadi

Road Signs Detection Using SSD MobileNetV2

Abstract

One of the most critical challenges for self-driving vehicles is accurately identifying traffic signs, which are essential for self-navigation and decision-making. Systems for the detection and recognition of road signs play a crucial role in this process by providing vital information for the vehicle's decision-making. This study proposes an approach for road sign identification and recognition utilising the TensorFlow Object Detection API and the SSD MobileNet V2 FPN Lite model.

In this proposal, we combine the efficiency and accuracy of SSD with the lightweight architecture of MobileNet to achieve excellent performance in object detection benchmarks while maintaining a small model size and low processing power requirements. The model was trained using the German Traffic Sign Recognition Benchmark (GTSRB) dataset. The proposed methodology achieved a mean detection accuracy of 100% while requiring 0.317 seconds to detect and recognise each sign.

Keywords

Object detection; Obstacle detection; Deep Learning; Feature pyramid network (FPN); Tensor flow

RESEARCH PAPER

Road Signs Detection Using SSD MobileNetV2

Zahraa S. Dhaif [a,*], Nidhal K. El Abbadi [b]

[a] Informatics Institute for Postgraduate Studies, Iraqi Commission for Computers and Informatics, Baghdad, Iraq
[b] Al-Mustaqbal Centre for AI Applications, Al-Mustaqbal University, Babylon, Iraq

Abstract

One of the most critical challenges for self-driving vehicles is accurately identifying traffic signs, which are essential for self-navigation and decision-making. Systems for the detection and recognition of road signs play a crucial role in this process by providing vital information for the vehicle's decision-making. This study proposes an approach for road sign identification and recognition utilising the TensorFlow Object Detection API and the SSD MobileNet V2 FPN Lite model.

In this proposal, we combine the efficiency and accuracy of SSD with the lightweight architecture of MobileNet to achieve excellent performance in object detection benchmarks while maintaining a small model size and low processing power requirements. The model was trained using the German Traffic Sign Recognition Benchmark (GTSRB) dataset. The proposed methodology achieved a mean detection accuracy of 100% while requiring 0.317 s to detect and recognise each sign.

Keywords: Object detection, Obstacle detection, Deep learning, Feature pyramid network (FPN), TensorFlow

1. Introduction

Deep learning is proving its worth every year in many vision based applications, including vital ones like traffic monitoring and autonomous driving. In this field, semantic road identification and precise traffic sign recognition are becoming essential elements for enhancing computer vision safety [1]. The development of advanced driving assistance systems (ADAS) and self-driving cars depends on efficient object detection. Although many algorithms have been developed for reliable object detection using machine learning techniques, the challenges unique to traffic sign recognition are currently being studied [2].

Understanding traffic signals is crucial for all drivers before operating a vehicle on public roads, a fundamental skill tested in the theoretical driver's licence exam [3]. Traffic signs, the oldest and most widely utilised method of traffic control, are crucial tools for conveying directives, warnings, and reminders. Traffic signs are essential in conveying the information needed by the driver to avoid accidents in which driver errors such as lack of attention or misinterpretation of signals cause a substantial number of accidents [4]. These signs must be both identified and categorised: while classification tasks identify subclasses of objects, detection tasks aim to precisely determine their position and size [5]. In computer vision, traffic sign identification involves locating areas of images with bounding boxes around these signs, defined by their distinctive shapes, reliable colours, and spatial relationships with the other objects by the roadside [6].

Some research using deep learning frameworks like the TensorFlow Object Detection API has been limited to specific object detection categories, such as traffic light and mobile weapon detection. Similar results have been observed in studies using the TensorFlow Object Detection API to identify vehicles. However, applying deep learning frameworks for comprehensive road sign identification, such as the TensorFlow Object Detection API, offers the advantage of scalability, even though they have primarily been used for traffic signal and vehicle detection [7]. Google's TensorFlow Object Detection

Received 23 June 2024; revised 5 September 2024; accepted 11 September 2024.
Available online 18 October 2024

* Corresponding author.
E-mail addresses: phd202120679@iips.edu.iq (Z.S. Dhaif), nidhal.abass@fulbrightmail.org (N.K. El Abbadi).

https://doi.org/10.33640/2405-609X.3373

API provides a flexible platform for developing object recognition models, enabling the creation and training of object detection models with ease due to its user friendly interface. Additionally, it implements various models pre-trained for object detection, such as SSDs, R-FCN, and Faster R-CNN. Using TensorFlow's ability to train models with large datasets facilitates the development of increasingly complex object detection models [8].

This study addresses the significant object detection challenges at various scales, specifically small ones, by integrating the Feature Pyramid Network (FPN) into the detection process. FPN, designed with a feature pyramid concept, improves both accuracy and speed. To enhance the precision and effectiveness of object detection in the context of traffic sign identification for ADAS and self-driving cars, we propose using SSD MobileNetV2 in this study.

The structure of the remaining paper is as follows: Section Two delves into related studies. Section Three outlines the research problems. Section Four explains the main algorithm used in this proposal, the SSD MobileNetV2 Object Detector Architecture. The tools used for the evaluation of the model are introduced in Section Five. A detailed explanation of the proposed system and its implementation is introduced in Section Six. Results and discussion are presented in Section Seven. Finally, Section Eight concludes the findings and presents limitations and future work.

2. Related works

One of the hottest research subjects in the last ten years has been the creation of autonomous vehicles.

Researchers in Ref. [9] suggested a real-time method for detecting tiny traffic signs, using a small region proposal generator and combining Faster R-CNN with Online Hard Examples Mining (OHEM). The system is robust for small signs only, and the mean average precision increased by 12.1% compared to the first object detection algorithm, demonstrating satisfactory performance in various videos. The proposed approach is based on the updated Faster R-CNN architecture and has been tested on various videos. For quick and precise traffic sign recognition in photos, a Traffic Sign Recognition (TSR) system was developed by the authors in Ref. [10] using two carefully designed convolutional neural networks (CNNs). Regarding scale-invariant detection, the suggested CNN utilises a fully convolutional network incorporating two multi-scale designs. Modified “Online Hard Example Mining” is used by the training network to decrease false positives. For efficiency, the classification network combines multi-scale characteristics with an “Inception” module. Through extensive testing, the system outperformed state of the art techniques, achieving 99.88% precision and 96.61% recall on the Swedish Traffic Signs Dataset (STSD). Compared to existing deep learning networks for traffic sign identification, it is lighter and faster. Two stages were proposed by Ahmed Hechri and Abdellatif Mtibaa [11] to detect and identify traffic signs in real time. Traffic signs are detected and classified into circular and triangular forms in the first stage using histograms of oriented gradients (HOG) features and support vector machine (SVM). In the second step, traffic signs are categorised into their respective categories using a CNN. The performance is tested on the German traffic sign detection benchmark (GTSDB) and German traffic sign recognition benchmark (GTSRB) datasets, showing comparable detection and recognition rates with less complexity. Its suitability for real-time processing applications is demonstrated by the average processing time. An effective traffic sign detection and recognition (TSDR) system was developed by Safat B. Wali et al. [12], utilising an enhanced collection of traffic signs from Malaysia. The system has a short computing time and a low false positive rate, and it is invariant in terms of lighting, rotation, translation, and viewing angle. The three stages of the system are detection, recognition, and image preprocessing. The system's results included a processing time of 0.43 s, a low false positive rate of 0.9%, and an accuracy of 95.71%. To evaluate recognition performance, the area under the receiver operating characteristic (ROC) curves was introduced. A novel method for real-time traffic sign detection and recognition was proposed by Faming Shao et al. [13]. The method converts road scenes to grayscale images, filters them with simplified Gabor wavelets, strengthens traffic sign edges, extracts the region of interest, and classifies traffic signs using SVM. CNNs classify traffic signs into subclasses. The experimental results meet real-time processing demands with better processing efficiency and are equivalent to state-of-the-art performance approaches. Chang Sun et al. [14] proposed a technique for a deep learning based model called (Dense-RefineDet) uses a single shot object detection framework and a dense connection related transfer-connection block to optimise contextual information. It achieves competitive accuracy at high speed detection of small, medium, and large scale traffic signs, with a 54.03% miss rate compared to other methods. Zhe Zhu et al. [15] created a large traffic sign benchmark, Tsinghua-Tencent 100K, from 100,000 Tencent Street View panoramas, covering 30,000 traffic-sign instances. The benchmark is annotated with class labels, bounding boxes, and pixel masks. Unlike prior CNN methods that target large objects, the resilient end-to-end CNN is shown to identify and categorise traffic signals. Experimental results demonstrate the network's robustness and superiority to alternatives.

3. Research problem

Several impediments and difficulties are present, including fluctuations in light, shifting meteorological conditions, and objects that occlude. Among these problems and difficulties are the following:

1. In a real-time application, sight may be impeded by the shadows and headlights of approaching vehicles, fog, clouds, rain, and snow.
2. The weather might affect the sign's colour.
3. Various man-made objects that resemble traffic signs in terms of colour, form, and size could be shown in a scene.
4. More than one road sign may overlap or be in one frame.
5. It is not feasible to observe traffic signs consistently from the same perspective. Shapes and patterns on signs are typically warped.
6. The accuracy and speed of classification detection while considering hardware requirements are two of the most challenging and significant challenges in a real-time setting. The majority of systems require expensive GPUs with extremely high specifications.
7. Blurred visuals are typically caused by moving cars.

4. SSD MobileNetv2 object detector architecture

The SSD MobileNet model combines the MobileNet and SSD models. MobileNet, a CNN, is used as a feature extractor to generate high-level features, while SSD, a widely used method, serves as the object detector [16]. MobileNetV2, one of the lightest and most popular network topologies in recent deep learning techniques [17], was developed by integrating inverted residual and linear bottleneck modules into MobileNetV1 [18]. MobileNetV2 significantly reduces processing memory requirements through its inverted residuals with linear bottleneck layers [19]. Its core building block is a bottleneck depth-separable convolution with residuals, and ReLU6 is used as the nonlinearity due to its robustness in low-precision computations. Modern networks typically employ a 3 × 3 kernel size and use batch normalisation and dropout during training. MobileNetV2 is employed as the feature extractor in the SSD model [20]. SSD, a widely used single stage object detector, is known for its fast computation [21]. The SSD technique uses a feed forward convolutional network to produce a fixed-size set of bounding boxes, each with a score indicating the presence of objects. The final set of detections is refined through a non-maximum suppression step. The initial network layers are based on a well-established architecture for high quality image classification and are truncated before the classification layers [22]. Fig. 1 illustrates the SSD MobileNetV2 architecture.

5. Model evaluation

The effectiveness of the current proposal is assessed using several metrics, including mean average precision (MAP), which is a widely utilised metric to assess the precision of classifiers and object detectors like SSD Mobile Net. The average precision per class for any model is determined by finding the area beneath the provided precision vs. recall curve. This value is averaged across all classes. To determine whether the prediction is accurate or inaccurate, the predicted bounding box is compared to the ground truth bounding box using Intersection Over Union (IOU). The percentage of predicted bounding boxes that overlap with real reality is shown by the IOU, which is the other metric. The forecast is deemed a true positive (TP) if the IOU is greater than 40%; otherwise, it is deemed a false positive (FP) [24]. Precision and recall are determined by the confusion matrix shown in Fig. 2. Equations (1)–(3) are used to calculate IOU, precision, and recall, respectively.

$$\text{IOU} = (\text{Area of the intersection of bounding boxes}) / (\text{Area of the union of bounding boxes}) \tag{1}$$

$$\text{Precision} = \frac{TP}{(TP + FP)} \tag{2}$$

$$\text{Recall} = \frac{TP}{(TP + FN)} \tag{3}$$

Using various IOU thresholds, the metrics for common objects in context (COCO) detection specify a range of average precision and recall values that are computed over small, medium, and large object scales. Fig. 3 displays the metrics utilised to gauge the detector's performance.

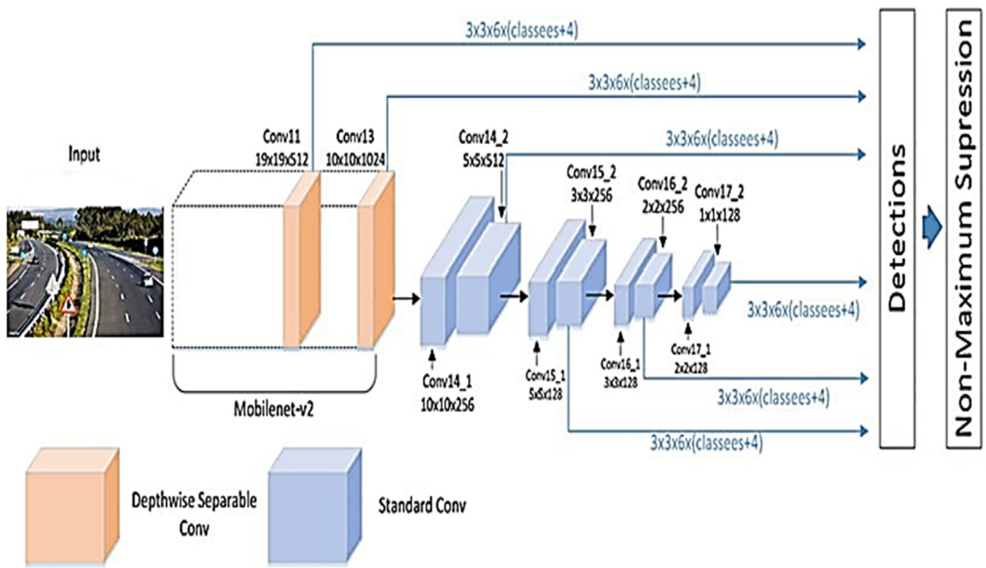

Fig. 1. SSD MobileNet v2 architecture [23].

6. Proposed system and implementation

6.1. Stages of proposed system

The suggested model includes two main parts: training and testing. For the training phase, we used images from the GTSRB dataset, which contains collected images with various backgrounds. The dataset is organised into 43 folders, each corresponding to a specific road sign and containing multiple images. The testing phase focuses on video data rather than still images. We utilised the pre-trained MobileNetV2 FPN Lite 320 × 320 SSD model, which was trained using TensorFlow Object Detection. The proposed system was trained with 40,000 images per epoch, achieving a learning loss of 0.0079. The architecture of the proposed model is shown in Fig. 4.

6.2. Datasets used in the training

In this proposal, we used the GTSRB dataset, which consists of 43,000 images for 43 explicit classes (1000 images of each class). The classes, for example, are Ahead Only, Beware of Ice or Snow, Bicycle Crossing, Children Crossing, Speed Limit (50 km), Stop, and Traffic Signals, with different backgrounds [26]. Fig. 5 presents an example of this dataset.

		Actual Class	
		Positive (P)	Negative (N)
Predicated Class	Positive (P)	True Positive (TP)	False Positive (FP)
	Negative (N)	False Negative (FN)	True Negative (TN)

Fig. 2. Confusion matrix.

Average Precision (AP):	
AP	% AP at IoU=.50:.05:.95 (primary challenge metric)
$AP^{IoU=.50}$	% AP at IoU=.50 (PASCAL VOC metric)
$AP^{IoU=.75}$	% AP at IoU=.75 (strict metric)
Average Recall (AR)	
$AP^{max=1}$	% AR given 1 detection per image
$AP^{max=10}$	% AR given 10 detections per image
$AP^{max=100}$	% AR given 100 detections per image
AP/ AR across Scales	
AP^{small}	% AP for small objects: area < 32^2
AP^{medium}	% AP for medium objects: 32^2 < area < 96^2
AP^{large}	% AP for large objects: area > 96^2

Fig. 3. COCO detection metrics [25].

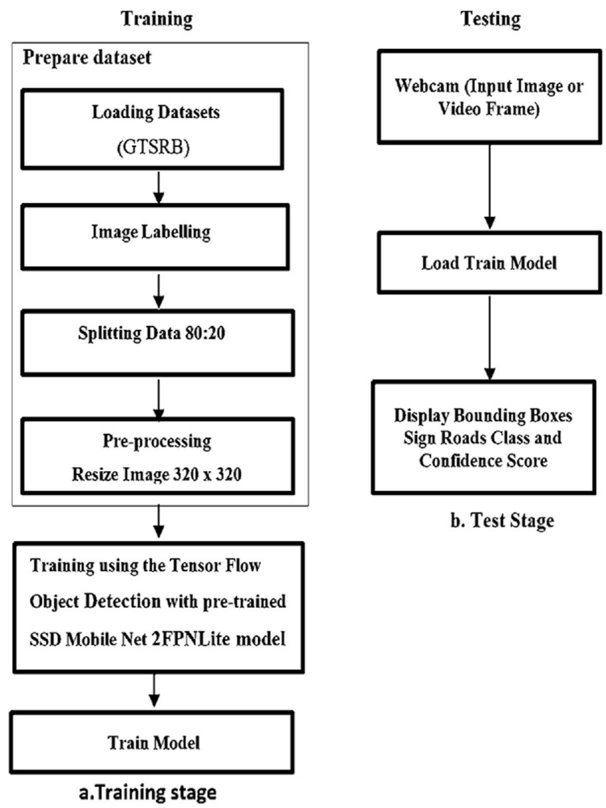

Fig. 4. Model planning: a. Training stage b. Test stage.

6.3. Preprocessing

The input image can be any type of image format; the image was resized according to the SSD MobileNetv2 FPN Lite, which is 320 × 320. To preprocess label images, the Roboflow application is used. The file format used by TensorFlow is built by Roboflow from images and a Pandas data frame. The data frame has the following columns:

- “filename”: path of the image file.
- “class”: Object class name.

Fig. 5. Road signs [27].

- “ymin”, “xmin”, “ymax”, “xmax”: normalised coordinates of the boundary detection box. Fig. 6 shows boundary boxes.

6.4. Training using TensorFlow API for object detection

The TensorFlow object detection framework version 2.8 and the pre-trained TensorFlow 2 ZOO model are now being utilised in our training and evaluation procedures. We enhance the Single Shot Detector (SSD) architecture for object recognition and the MobileNet v2 architecture for feature extraction by adding the Features Pyramid Network Lite (FPN Lite) module, resulting in the creation of the SSD MobileNetV2 FPN Lite 320 × 320 architecture.

The MobileNet v2 architecture is a lightweight CNN specifically designed for efficient data processing on mobile and embedded devices. By employing separable convolutions and residual connections, the network's performance is enhanced and the parameter count is reduced. To recognise objects in the input image, MobileNet is used for feature extraction, which involves identifying significant features.

The popular SSD object identification technique predicts the location and type of objects in the input image using a collection of pre-made bounding boxes known as anchor boxes. These fitting box predictions are combined with the non-maximum suppression (NMS) technique to obtain the final detection results. The characteristics gathered by the SSD are used to identify objects in the image and determine their location.

In summary, the FPN concept combines low-resolution information from deeper levels with high-resolution data from shallow layers to improve detection accuracy. The outputs of the SSD and MobileNet v2 layers are combined in SSD MobileNet V2 FPN Lite 320 × 320 models using the FPN Lite module. This increases the contextual data available for the object detection task, enhancing its accuracy by integrating data from various scales.

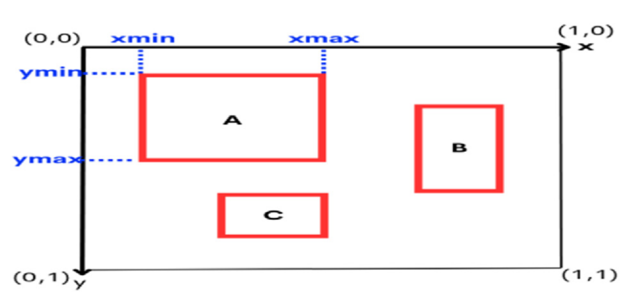

Fig. 6. Boundary boxes.

The SSD MobileNet V2 FPN Lite 320 × 320 is a practical object identification model for real-time applications due to its fast speed of 22 ms and MAP value of 22.2. It performs exceptionally well on object detection tasks. Table 1 explains the parameter values used by SSD MobileNet V2 FPN Lite 320 × 320.

6.5. Recognition

After the trained model is loaded, the webcam inputs the video. When using video data, the system continually analyses each frame of the video stream, identifying and categorising traffic signs instantly, at 30 frames per second. The model's output will display a bounding box for each road sign, along with a confidence score. Consequently, the system swiftly and accurately detects road signs.

7. Results and discussion

The measures used for evaluating the road sign detection model obtained in the final step (40k) are displayed in Table 2.

7.1. Losses

Losses indicate the efficiency of the model and determine the amount of error in it. Understanding error is crucial to gauge the extent to which the model benefits from training, as shown in Fig. 7. It consists of several forms:

Table 1. Training parameter values of SSD MobileNet V2 FPN Lite 320 × 320.

parameters	SSD MobileNetv2 FPN Lite 320 × 320
Batch/epoch	40000
Batch Size	16
Learning rate	0.0319994
Size image	320 × 320 px

Table 2. Metrics for evaluation of the road sign detection model.

Metric	Area	@ IOU	MaxDets	Value
Average Precision (AP)	all	0.50:0.95	100	1.000000
		0.5		1.000000
		0.75		1.000000
	small	0.50:0.95		1.000000
	medium	0.50:0.95		1.000000
	large	0.50:0.95		1.000000
Average Recall (AR)	all	0.50:0.95	1	1.000000
			10	1.000000
			100	1.000000
	small			1.000000
	medium			1.000000
	large			1.000000

- **Localisation_loss function:** This indicates the correctness of the model by providing the dimensions of the box around the object.

$$localization\ loss = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + \left(\sqrt{w_i} - \sqrt{\hat{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2 \right] \quad (4)$$

Where

- ◦ S^2 is the number of grid cells (typically, $S = 7$, so $S^2 = 49$).
- ◦ B is the number of bounding boxes predicted per grid cell.
- ◦ x_i, y_i are the coordinates of the centre of the bounding box.
- ◦ w_i, h_i are the width and height of the bounding box.
- ◦ $\hat{x}_i, \hat{y}_i, \hat{w}_i, \hat{h}_i$ are the predicted values.
- ◦ 1_{ij}^{obj} is an indicator function that is 1 if object j is in cell i and 0 otherwise.
- ◦ λcoord is a scaling factor that gives more weight to localisation errors.

- **Classification_loss function:** This indicates the correctness of the model, whether it performs the classification process correctly or not.

$$Classification\ Loss = \sum_{i=0}^{S^2} 1_i^{obj} \sum_{c\,\epsilon\,class} \left(p_i(c) - \hat{p}_i(c)\right)^2 \quad (5)$$

Where

- ◦ $p_i(c)$ is the ground truth probability of class c for grid cell i.
- ◦ $\hat{p}_i(c)$ is the predicted probability of class c for grid cell i.
- ◦ 1_i^{obj} is 1 if an object is present in grid cell i, otherwise 0.

- **Total_loss:** It is an average of the types of losses present.

The total loss function in SSD MobileNet V2 is the sum of the localisation, classification, and confidence losses. The confidence loss measures the error in predicting the objectness score, which indicates the likelihood of an object being present in the bounding box.

$$Total\ Loss = Localization\ Loss + Classification\ Loss + Confidence\ Loss \quad (6)$$

Where the **Confidence Loss** is given by the following:

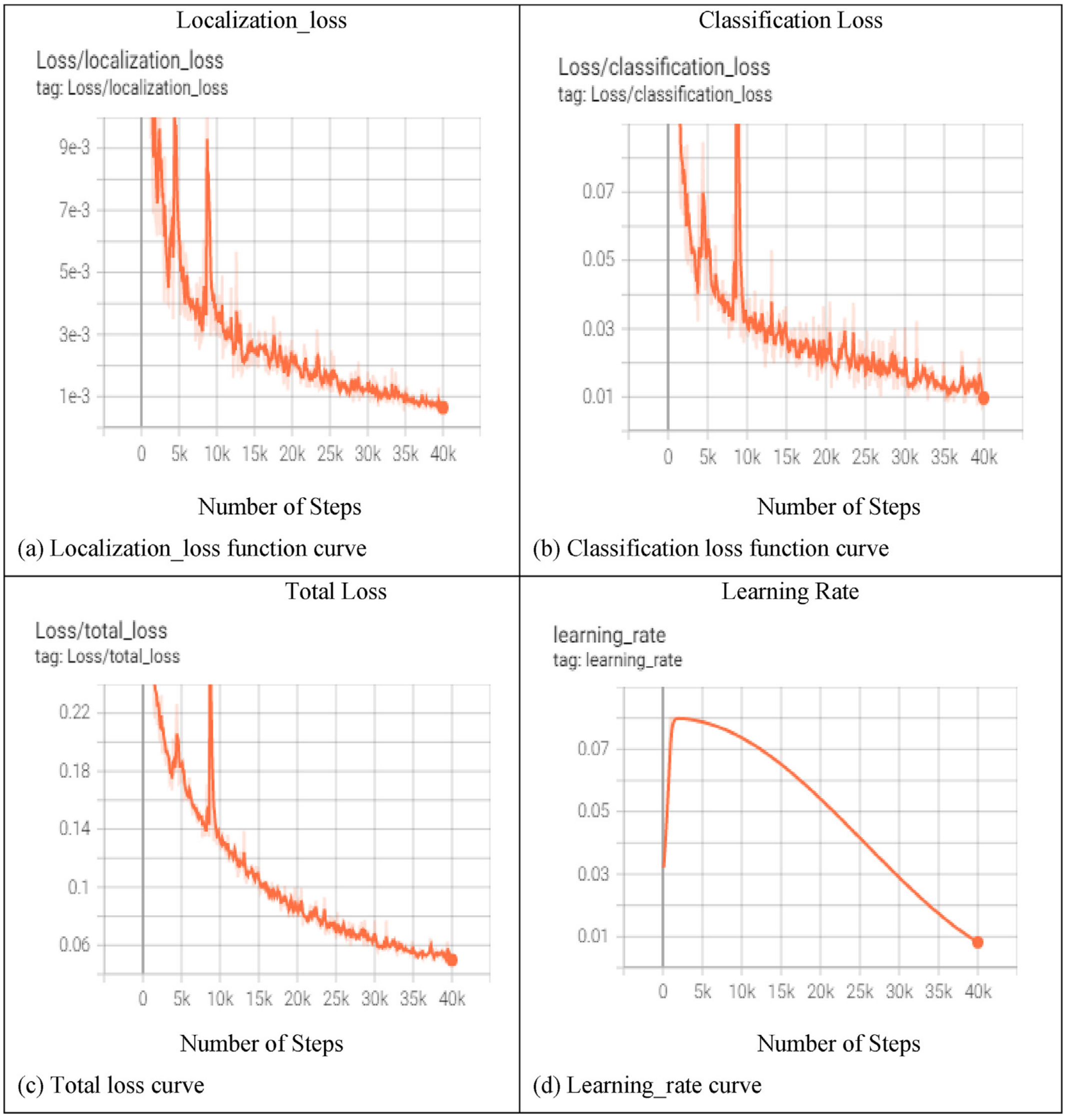

Fig. 7. The Losses num steps (a) Localization_loss (b) Classification loss (c) Total loss (d) Learning_rate.

$$Confidence\ loss = \sum_{i=0}^{S^2}\sum_{j=0}^{B} 1_{ij}^{obj}\left(\widehat{C}_i - C_i\right)^2 + \lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{B} 1_{ij}^{noobj}\left(\widehat{C}_i - C_i\right)^2 \tag{7}$$

Where

- C_i is the ground truth confidence score (usually 1 if the object is present, 0 otherwise).
- $\widehat{C}_i$ is the predicted confidence score.
- λ_{noobj} is a scaling factor to down-weigh the loss from predictions in which no object is present.
- 1_{ij}^{noobj} is 1 if no object is present in cell i, and 0 otherwise.

These equations represent the key components of the loss function used to train the SSD MobileNet V2 model to detect road signs effectively. The total loss is minimised during training to improve the accuracy of both object localisation and classification.

The proposed work minimises the losses by avoiding overfitting. Many strategies are used for that purpose: Adjusting learning rate, batch size, and epochs, using techniques like learning rate scheduling and early stopping to avoid overfitting. Using L2 regularisation and dropout to stabilise training and prevent overfitting. Employing different techniques to diversify training data (data augmentation). Ensuring adequate representation of all road sign types in the training data to minimise classification loss.

Table 3 presents some information on the losses per step of the road-sign detection model obtained in the number of epochs from 100 to 40,000. We notice that in the initial epoch, the values were very large for the localization and classification losses, but during the process of training, these values began to

Table 3. Information on the losses per step.

No. of Steps	Localization_loss	Classification loss	Total loss	Learning rate	per-step time
100	0.044876754	1.1215405	1.3196056	0.0319994	0.834s
200	0.035248633	0.75162685	0.93968153	0.0373328	0.316s
300	0.059617925	0.4227719	0.6348752	0.0426662	0.321s
………					
10000	0.0037299744	0.03044345	0.13079241	0.07352352	0.318s
………					
15000	0.0023271034	0.02128285	0.09838165	0.064939596	0.319s
………					
20000	0.0018106259	0.018629357	0.08078809	0.0538146	0.317s
………					
25000	0.0012641152	0.012266388	0.0644014	0.04128206	0.319s
………					
30000	0.0017922304	0.030422302	0.07704803	0.028618898	0.319s
………					
35000	0.0011810567	0.015617614	0.05800961	0.017115325	0.317s
………					
40000	0.0005427136	0.009527728	0.049387537	0.007943453	0.317s

Fig. 8. Shows predicted road signs and confidence values.

gradually decrease, even concerning the learning rate, which was initially equal to (0.0319994), but in the last step, it became (0.007943453), indicating the success of the training process.

7.2. Model testing

The test was conducted on 250 images, and all the signs in the images were detected correctly. Additionally, we tested the model under various challenging conditions, such as variations in lighting, images at night, occluded signs, unclear images due to shadows, and signs in various environments such as rain, fog, and snow. The results in all the above conditions showed high confidence values, as shown in Fig. 8.

7.3. Comparisons

Many algorithms have been suggested to detect road signs. We compared recent methods that focus

Table 4. A comparison of recent research on road sign detection.

Ref	Algorithms	Accuracy %
[28]	Faster R-CNN	80.86
[29]	TRD-YOLO (Trans-Decoupled YOLO)	86.3
[30]	Improved LeNet-5 CNN	99.75
[31]	YOLO v3 and YOLO v4-tiny algorithms and a customised CNN model	95.85
The proposed method in this paper	MobileNetV2 SSD	100

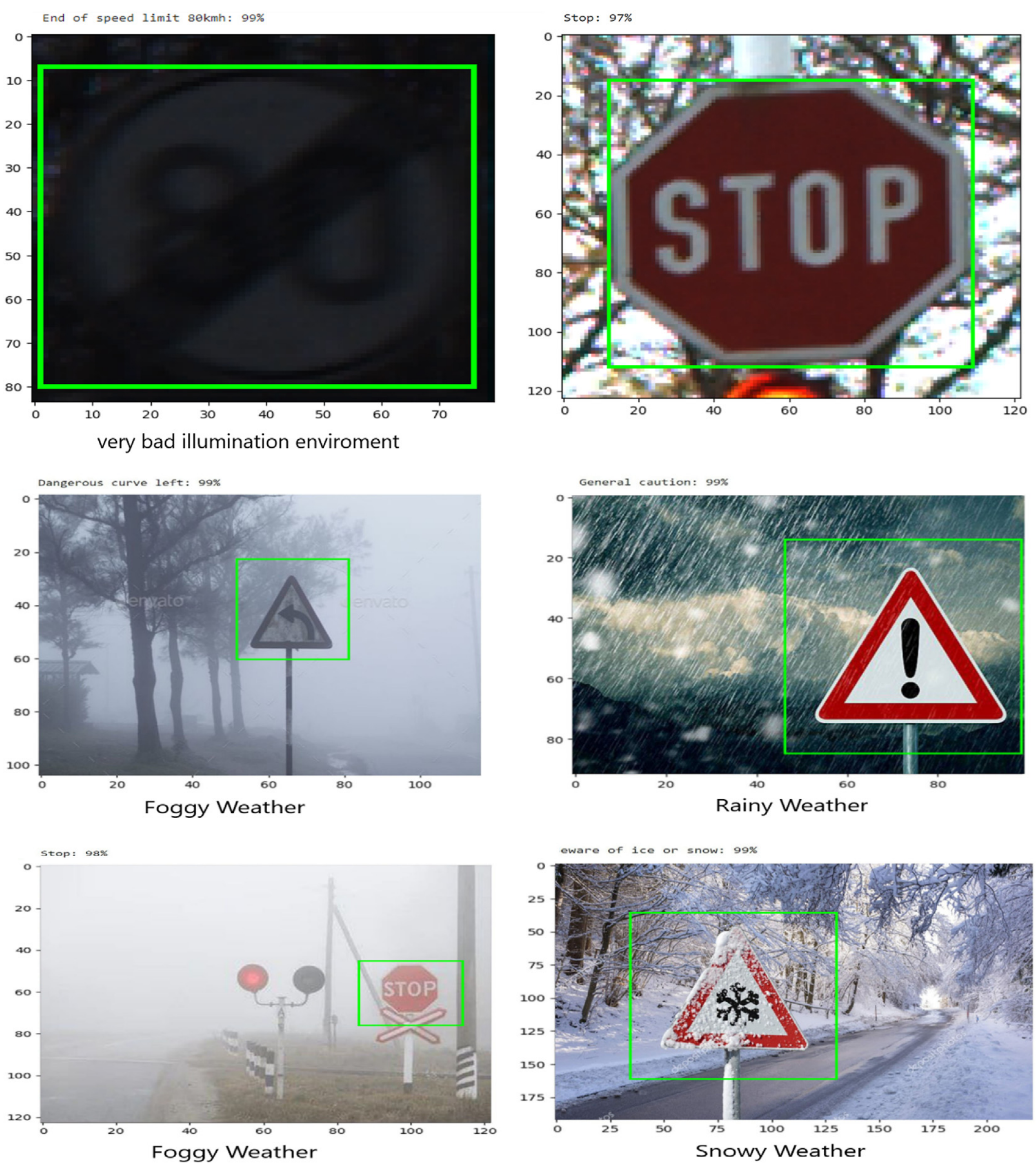

Fig. 8. (Continued).

on detecting road signs with our proposed model, as indicated in Table 4.

8. Conclusions

A second-generation MobileNet network, known as Mobile Net V2, serves as the foundational network model for the SSD detector in the Mobile Net SSD network design. In addition to maintaining the benefits of the original MobileNet-SSD's quick processing, the MobileNet-SSDv2 detector significantly increases detection accuracy. Being quicker than the alternative current networks for detecting objects, the MobileNet-SSD detector can function in real time. This advantage demonstrates that MobileNet-SSD v2 is appropriate for use in this research. I used [SSD MobileNet V2 Finite 320 × 320] to train my model's real-time platform for detecting road signs, based on SSD Mobile Net V2, which may help avoid accidents. I trained the proposed model for 40k epochs, achieving a learning loss of 0.0079. The platform exhibits a high level of precision and speed in determining the state. To enhance detection accuracy, we integrated the FPN and MobileNet V2 models to improve the input image's feature map. Our proposed system achieved 100% detection accuracy with a processing time of 0.317 s per step. Thus, our approach can be applied to the real-time detection and categorisation of traffic signs. We have overcome issues related to detection speed and classification accuracy, successfully implementing the proposed system in real time despite atmospheric influences that hindered the detection process.

Although the current approach works efficiently, there are still some limitations. While the 320 × 320 SSD MobileNet V2 FPN Lite offers benefits in terms of speed and efficiency in specific scenarios, its scalability and performance in various environmental conditions are limited. Due to the 320 × 320 resolution, it may struggle to detect objects reliably at longer distances or in crowded spaces where objects might be smaller or closer together, which may limit its scalability. Incorrect bounding boxes or missed detections may result from attempting to scale the model to larger image sizes. Operating the MobileNet V2 FPN Lite SSD can also be challenging due to its reliance on specific features or patterns that might not be apparent under certain conditions. Identifying objects in various environmental settings, including shifting clouds, changing light, and complex backgrounds, could reduce performance and reliability in real-world applications when environmental variables are continuously changing.

To enhance this study, we suggest expanding the dataset and using other popular datasets such as the Belgian Traffic Sign Dataset or the LISA Traffic Sign Dataset. For future work, we recommend studying the model's ability to generalise across different geographical locations and road sign standards, ensuring its applicability in various countries.

Ethics information

This study did not involve human participants or animals, and ethical approval was not required according to the institutional and national guidelines.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Acknowledgements

The authors would like to thank the Informatics Institute for Postgraduate Studies and the Iraqi Commission for Computers and Informatics for their support in the present work.

References

[1] K. Bayoudh, F. Hamdaoui, A. Mtibaa, Transfer learning based hybrid 2D-3D CNN for traffic sign recognition and semantic road detection applied in advanced driver assistance systems, Appl. Intell. 51 (2021) 124–142, https://doi.org/10.1007/s10489-020-01801-5.

[2] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587, https://doi.org/10.1109/CVPR.2014.81.

[3] D. Maulina, E.S. Siregar, T.A. Rachma, S.A. Nashria, D.Y. Irwanda, How effective is training for improving traffic sign comprehension? Examining the interaction between training and sign type among motorcyclists, IATSS Res. 46 (2022) 614–622, https://doi.org/10.1016/j.iatssr.2022.11.002.

[4] T. Ben-Bassat, D. Shinar, The effect of context and drivers' age on highway traffic signs comprehension, in: Transportation Research Part F: Traffic Psychology and Behaviour vol. 33, 2015, pp. 117–127, https://doi.org/10.1016/j.trf.2015.07.009.

[5] Z. Liu, D. Li, S.S. Ge, F. Tian, Small traffic sign detection from large image, Appl. Intell. 50 (2020) 1–13, https://doi.org/10.1007/s10489-019-01511-7.

[6] R. Qian, B. Zhang, Y. Yue, Z. Wang, F. Coenen, Robust Chinese traffic sign detection and recognition with a deep convolutional neural network, in: 2015 11th International Conference on Natural Computation (ICNC) vol. 11, 2015, pp. 791–796, https://doi.org/10.1109/ICNC.2015.7378092.

[7] D. Indra, H. Herman, F.S. Budi, Implementasi sistem penghitung kendaraan otomatis berbasis computer vision, Komputika: Jurnal Sistem Komputer 12 (2023) 53–62, https://doi.org/10.34010/komputika.v12i1.9082.

[8] J. Koçi, A.O. Topal, M. Ali, Threat object detection in x-ray images using SSD, r-FCN and faster r-CNN, in: 2020 International Conference on Computing, Networking, Telecommunications & Engineering Sciences Applications (CoNTESA), 2020, pp. 10–15, https://doi.org/10.1109/CoNTESA50436.2020.9302863.

[9] C. Han, G. Gao, Y. Zhang, Real-time small traffic sign detection with revised faster-RCNN, Multimed. Tool. Appl. 78 (2019) 13263–13278, https://doi.org/10.1007/s11042-018-6428-0.

[10] Y. Yang, S. Liu, W. Ma, Q. Wang, Z. Liu, Efficient traffic-sign recognition with scale-aware CNN, arXiv preprint arXiv:1805.12289. 31 (2018). https://doi.org/10.48550/arXiv.1805.12289.

[11] A. Hechri, A. Mtibaa, Two-stage traffic sign detection and recognition based on SVM and convolutional neural networks, IET Image Process. 14 (2020) 939–946, https://doi.org/10.1049/iet-ipr.2019.0634.

[12] S.B. Wali, M.A. Hannan, A. Hussain, S.A. Samad, An automatic traffic sign detection and recognition system based on colour segmentation, shape matching, and SVM, Math. Probl. Eng. 2015 (2015), https://doi.org/10.1155/2015/250461.

[13] F. Shao, X. Wang, F. Meng, T. Rui, D. Wang, J. Tang, Real-time traffic sign detection and recognition method based on simplified Gabor wavelets and CNNs, Sensors 18 (2018) 3192, https://doi.org/10.3390/s18103192.

[14] C. Sun, Y. Ai, S. Wang, W. Zhang, Dense-RefineDet for traffic sign detection and classification, Sensors 20 (2020) 6570, https://doi.org/10.3390/s20226570.

[15] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, S. Hu, Traffic-sign detection and classification in the wild, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2110–2118, https://doi.org/10.1109/CVPR.2016.232.

[16] E.M.T.A. Alsaadi, N.K. El Abbadi, An automated mammals detection based on SSD-mobile net, J Phys Conf 1879 (2021) 022086, https://doi.org/10.1088/1742-6596/1879/2/022086.

[17] F.I. Eyiokur, H.K. Ekenel, A. Waibel, Unconstrained face mask and face-hand interaction datasets: building a computer vision system to help prevent the transmission of COVID-19, Signal Image Video Process. 17 (2023) 1027–1034, https://doi.org/10.1007/s11760-022-02308-x.

[18] R. Indraswari, R. Rokhana, W. Herulambang, Melanoma image classification based on MobileNetV2 network, Procedia Comput. Sci. 197 (2022) 198–207, https://doi.org/10.1016/j.procs.2021.12.132.

[19] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: inverted residuals and linear bottlenecks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510–4520, https://doi.org/10.1109/CVPR.2018.00474.

[20] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: efficient convolutional neural networks for mobile vision applications, arXiv preprint arXiv:1704.04861. 17 (2017). https://doi.org/10.48550/arXiv.1704.04861.

[21] Y.-C. Chiu, C.-Y. Tsai, M.-D. Ruan, G.-Y. Shen, T.-T. Lee, Mobilenet-SSDv2: an improved object detection model for embedded systems, in: 2020 International Conference on System Science and Engineering (ICSSE), 2020, pp. 1–5, https://doi.org/10.1109/ICSSE50014.2020.9219319.

[22] A. Srivastava, A. Dalvi, C. Britto, H. Rai, K. Shelke, Explicit content detection using faster r-cnn and SSD mobile net v2, Int. Res. J. Eng. Technol. 7 (2020) 5572–5577.

[23] B. Ramalingam, R. Elara Mohan, S. Balakrishnan, K. Elangovan, B. Félix Gómez, T. Pathmakumar, M. Devarassu, M. Mohan Rayaguru, C. Baskar, Stetro-deep learning-powered staircase cleaning and maintenance reconfigurable robot, Sensors 21 (2021) 6279, https://doi.org/10.3390/s21186279.

[24] H. Mahdi, N. El Abbadi, Glaucoma diagnosis based on retinal fundus image: a review, Iraqi J. Sci. 63 (2022) 4022–4046, https://doi.org/10.24996/ijs.2022.63.9.32.

[25] M. Amo-Boateng, N.E.N. Sey, A.A. Amproche, M.K. Domfeh, Instance segmentation scheme for roofs in rural areas based on Mask R-CNN, Egypt. J. Remote Sens. Space Sci. 25 (2022) 569–577, https://doi.org/10.1016/j.ejrs.2022.03.017.

[26] J. Stallkamp, M. Schlipsing, J. Salmen, C. Igel, The German traffic sign recognition benchmark: a multi-class classification competition, in: The 2011 International Joint Conference on Neural Networks, 2011, pp. 1453–1460, https://doi.org/10.1109/IJCNN.2011.6033395.

[27] N. Youssouf, Traffic sign classification using CNN and detection using faster-RCNN and YOLOV4, Heliyon 8 (2022) e11792, https://doi.org/10.1016/j.heliyon.2022.e11792.

[28] Y.-C. Chiu, H.-Y. Lin, W.-L. Tai, A two-stage learning approach for traffic sign detection and recognition, VEHITS 7 (2021) 276–283, https://doi.org/10.5220/0010384002760283.

[29] J. Chu, C. Zhang, M. Yan, H. Zhang, T. Ge, TRD-YOLO: a real-time, high-performance small traffic sign detection algorithm, Sensors 23 (2023) 3871, https://doi.org/10.3390/s23083871.

[30] J. Cao, C. Song, S. Peng, F. Xiao, S. Song, Improved traffic sign detection and recognition algorithm for intelligent vehicles, Sensors 19 (2019) 4021, https://doi.org/10.3390/s19184021.

[31] K. Dhawan, S.P.R, N. R.K, Identification of traffic signs for advanced driving assistance systems in smart cities using deep learning, Multimed. Tool. Appl. 82 (2023) 26465–26480, https://doi.org/10.1007/s11042-023-14823-1.