Multimedia Tools and Applications (2023) 82:26465–26480
https://doi.org/10.1007/s11042-023-14823-1

Identification of traffic signs for advanced driving assistance systems in smart cities using deep learning

Kshitij Dhawan[1] · Srinivasa Perumal R.[2] · Nadesh R. K.[2]

Received: 11 January 2022 / Revised: 22 June 2022 / Accepted: 6 February 2023 /
Published online: 4 March 2023

Abstract

The ability of Advanced Driving Assistance Systems (ADAS) is to identify and understand all objects around the vehicle under varying driving conditions and environmental factors is critical. Today's vehicles are equipped with advanced driving assistance systems that make driving safer and more comfortable. A camera mounted on the car helps the system recognise and detect traffic signs and alerts the driver about various road conditions, like if construction work is ahead or if speed limits have changed. The goal is to identify the traffic sign and process the image in a minimal processing time. A custom convolutional neural network model is used to classify the traffic signs with higher accuracy than the existing models. Image augmentation techniques are used to expand the dataset artificially, and that allows one to learn how the image looks from different perspectives, such as when viewed from different angles or when it looks blurry due to poor weather conditions. The algorithms used to detect traffic signs are YOLO v3 and YOLO v4-tiny. The proposed solution for detecting a specific set of traffic signs performed well, with an accuracy rate of 95.85%.

Keywords ADAS · YOLO v3 · YOLO v4-tiny · Traffic signs recognition · Customised CNN · Image augmentation

Kshitij Dhawan, Srinivasa Perumal R. and Nadesh R. K. have contributed equally to this work.

✉ Nadesh R. K.
rknadesh@gmail.com

Kshitij Dhawan
kshitij.dhawan310@gmail.com

Srinivasa Perumal R.
r.srinivasaperumal@vit.ac.in

[1] School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, 632014, Tamilnadu, India

[2] School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, 632014, Tamilnadu, India

1 Introduction

Advancement in the automobile industry has provided many smart cars on the road in different traffic environment and assist them in automated parking. Technology expansion and ubiquitous devices assist drivers in locating themselves on the road and understanding the current state of road traffic, distance to be travelled, best route identification, and so on. The cruise control in the vehicle helps the drivers with smart driving but also requires a lot of attention. Today's vehicles are equipped with advanced driving assistance systems that make driving safer and more comfortable. A camera mounted on the car helps the system to recognise and detect traffic signs and alerts the driver about various road conditions, like if construction work is ahead or if speed limits have changed. In real-time, the identification of traffic signs has many limitations and complexity. A small mistake in identifying a traffic sign can lead to disastrous consequences, even resulting in death. In the current era, Traffic Sign Recognition (TSR) is critical in ADAS because it helps the driver drive the car safely and minimises the accident rate. This requires good computation power and responds to real-time traffic. TSR plays a vital role in identifying the traffic sign for ADAS [2]. ADAS has a vehicle-mounted camera to acquire traffic signs to recognize and understand traffic signs from the actual road to control the vehicles [7]. It has many challenges, such as complex backgrounds, daily natural conditions, ageing signs, and changes in light [23].

Many researchers address the above challenges based on inherent information using deep learning. The TSR will assist in detecting traffic signs on the roadside. TSR will identify the signboard's location, size, and orientation for classifying the signs by discriminative information. Deep learning algorithms based on artificial intelligence are critical in resolving traffic sign recognition challenges. The Deep Learning model improves the efficiency of the driving experience and ensures safe driving in ADAS. It has more cameras and sensors to acquire information about the driving environment and conditions. ADAS will inform the driving rules and understand the traffic signs for the users [3].

Several object detection algorithms in ADAS, such as SSD, Faster RCNN, RFCN, and YOLO, are used in mobile devices. YOLO has proven to be a viable competitor to CNN in real-time object detection. Related work shows that detecting smaller objects moving at more incredible speeds has always been a challenge. YOLO V3 and YOLO V4 tiny show significant developments in this capability [22]. Enhanced Local Binary pattern is used to detect the face in the images where images are split into sub images [20]. An better-quality of the random sampling mechanism called Supervised Relative Random Sampling is used to create a balanced dataset from a highly imbalanced [17]. Even though numerous applications are developed, supervised learning and pattern recognition are crucial areas of research in information retrieval, knowledge engineering, and image processing [16]. The MobileNet V2 model was verified to be competent with enhanced accuracy on lightweight computational devices, [21]. A well organised system is proposed to detect the traffic signs and then categorise them. The image augmentation method helps to generate more training data. The customized CNN model has more neurons in each layer than the pre-existing CNN model to classify road signs.

The latest vehicles have in-built technology and supported computing power to make the drive more comfortable. Smart driving has many advantages and also has its disadvantages. The goal is to identify the traffic sign and process the image in a minimal processing time. In such cases, the focus should be on using state-of-the-art methods and technologies. To provide a solution for detecting a traffic sign, a traffic dataset consisting of 43 different classes of traffic signs commonly used in most European countries has been taken from the German Traffic Sign Recognition Benchmark collection, which comprises 60,000 images separated into 43 categories. A custom Convolutional Neural Network model is used to classify the traffic signs with higher accuracy than the existing models. Image augmentation techniques are used to expand the dataset artificially, and that allows one to learn how the image looks from different perspectives, such as when viewed from different angles or when it looks blurry due to poor weather conditions. This will make the dataset more balanced. A Lucy-Richardson nonlinear filter is used to remove any noise from images without blurring the edges, and histogram equalisation is used for brightening the images. The progression of deep learning algorithms like YOLO v3 and YOLO v4-tiny is used to provide faster detection rates without sacrificing classification accuracy.

The following are the remaining sections: The literature review is summarised in Section 2; the technique and how the models function are explained in Section 3; the Experimental results are displayed in Section 4; and the conclusion and future scope are presented in Section 5.

2 Literature review

In a fast-moving life, sometimes people fail to recognize the road signs and consequently violate the rules, and in majority of the cases, the consequences are severe. In order to limit the number of fatalities, several studies have been undertaken in this area. Researchers employed several techniques, including HOG, SIFT, LBP, and SVM, to classify traffic signs and notify drivers. The recognition rate of these algorithms is not satisfactory. Recently, various CNN models have evolved for Traffic sign recognition [24]. Cire san et al. applied augmentation techniques to improve the performance of CNN and multi-layer perceptron's [5]. To distinguish small items from signboards, Yuan et al. [25] suggested a multi-resolution hybrid system, a sign detection architecture. A vertical spatial sequence attention module (VSSA) gathers additional relevant details. Mobile applications incorporate Augmented Reality and GPS-based tracking. As a result, users can use their smartphone's coordinates to find nearby resources in an accessible way based on the direction. The AlexNet structure of CNN had produced a 92.63% accuracy rate [9]. Working with massive data sets and a high number of parameters is made easier using Google Net architecture. On the other hand, large data causes network overfitting, lowering the performance to 80.5 percent [4].

VGG CNN outperformed all the other existing models by a wide margin. Their model improved the performance by lowering the hyperparameters. Additionally, the network's batch normalization and global average pooling layers help improve performance without adding too many parameters. The combined Faster-RCNN architecture and Online Hard Examples Mining makes the system efficient in detecting tiny symptoms of impairment by erasing the pool4 layers and employing compression for ResNet [8]. TSR method based on Lenet-5 network improved the overall performance. The classifier in the Lenet-5 network performs better than the CNN, SVM, and classic Gabor classifiers [26]. A CNN-based TSR implemented has the feature extraction was carried out using CNN; the classification was performed using MPPs. MPPs have greatly improved the accuracy of recognition [18]. An attempt to use the YOLO algorithm was made and compared with existing computer vision algorithms. Vehicles, pedestrians, traffic signs, and lights are among the objects detected. A pre-trained model trained using the COCO dataset was used to establish the model's weights. The model showed significantly better results than the pre-existing algorithms when applied in real-world scenarios [6, 13]. Yolo algorithm was also used to detect triple riding and speed violation in the traffic systems [14].

CNN has shown excellent results earlier [15] in traffic sign classification and is recently used in traffic sign detection. Yihui Wu et al. applied deep learning to reject non-traffic sign [10]. A fully convolutional network and deep CNN for classification are performed and extended to recognize traffic signs [1]. A combined CNN-LSTM network model was deeply studied to improve the detection rate of novel coronavirus using X-ray images. The authors proposed an automated system to detect from the X-ray images and can improve their diagnostics rate. Its overall performance was indicated with the ROC curve and their area under the curve is 99.8% [11]. An automated face mask detection was introduced using deep learning methods for a smart city environment and the proposed method using classification network provides 98.7% accuracy [19]. Accident falls are increasing day by day , to detect the accident falls using several deep learning techniques like convolutional neural network, Auto-encoded systems, Long Short Term memory and tried to improve the performance with the existing methods [12]. Table 1 presents a summary of the various findings with their drawbacks in the existing models.

Researchers face challenges in detecting and predicting images in poor weather conditions or blurry images; this article addresses those challenges. The dataset is expanded using the image augmentation technique to get the expected results even for poor-quality images. The Lucy-Richardson nonlinear filter is used to remove any noise from images without blurring the edges, and histogram equalization is used for brightening the images, thereby enabling the proposed model to detect even in poor weather conditions. The second issue addressed is the slower response time. The proposed model employs YOLO v3 and YOLO v4-tiny to provide faster detection rates without sacrificing classification accuracy.

Table 1 Summary of the literature survey

Author	Approach used	Dataset	Accuracy	Drawback
Cire san et al. [5]	MCDNN	GTSRB[a]	99.46%	Poor results for blurred images
Yuan et al. [25]	VSSA-net	VOC 2007	94.6%	Approach limited to small datasets
Shu-Chun Huang [9]	Alexnet	GTSDB[b]	92.63%	Cannot distinguish the background from the traffic sign
Zhongqin Bi [4]	Improved VGG	GTSRB	98.47%	Poor environmental conditions not considered
Cen Han [8]	Faster RCNN	Self	92.00%	Greater detection time
Chuanwei Zhang [26]	Lenet-5	GTSDB	94%	Slower response time
Zhao Wang [18]	CNN	GTSRB	95.0%	Poor environmental conditions not considered
Chengji Liu [13]	YOLO[c]	GTSDB	94%	Slower response time
NC Mallela [14]	YOLOv3	COCO	91.7%	Lacks real time application
Smit Mehta [15]	CNN	BTSD[d]	98.26%	Poor results for blurred images
Choy Ja Yeong [10]	Raspberry Pi	GTSRB	90.00%	Greater detection time

[a] GTSRB : German Traffic Sign Recognition Benchmark

[b] GTSDB : German Traffic Sign Detection Benchmark

[c] YOLO : You only look once

[d] BTSD : Belgium traffic sign dataset

3 Methodology

The proposed system, as shown in Fig. 1 has five stages: capturing the video, preprocessing the video, removing the noise from the extracted images, detecting the traffic signs, and identifying traffic signs in the video frames. HD cameras are mounted on the cars to capture video, which is then sampled according to a particular frame rate. The environmental conditions enhance the images. The detection stage must be quick and efficient for a real-time detection system to function correctly. The output of the detection stage is fed into the noise removal stage, where a median filter is applied to the images for smoothening. Then the images are deblurred using the Lucy Richardson filter. Later, histogram equalization is applied to the images to increase the brightness of the images. In the recognition stage, the proposed custom-CNN model is used for classification. The YOLO algorithm is used to detect signs, and the trained custom-CNN model is used to classify captured signs into their sub-classes.

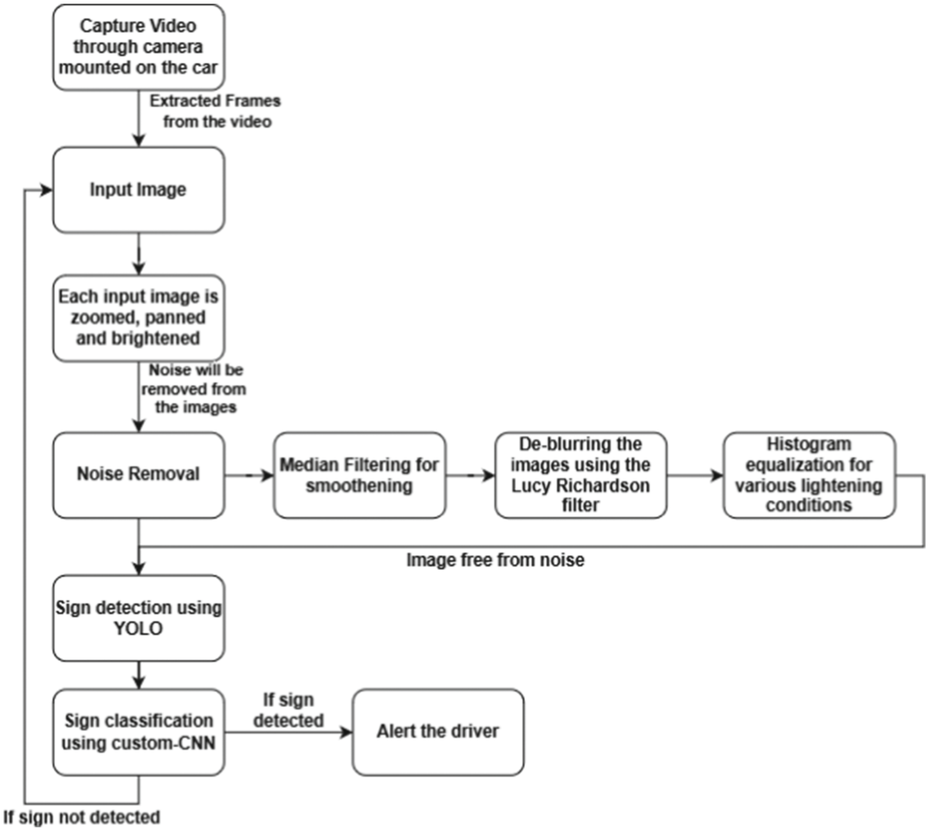

Capture Video through camera mounted on the car
Extracted Frames from the video
Input Image
Each input image is zoomed, panned and brightened
Noise will be removed from the images
Noise Removal
Median Filtering for smoothening
De-blurring the images using the Lucy Richardson filter
Histogram equalization for various lightening conditions
Image free from noise
Sign detection using YOLO
Sign classification using custom-CNN
If sign detected
Alert the driver
If sign not detected

Fig. 1 Proposed model

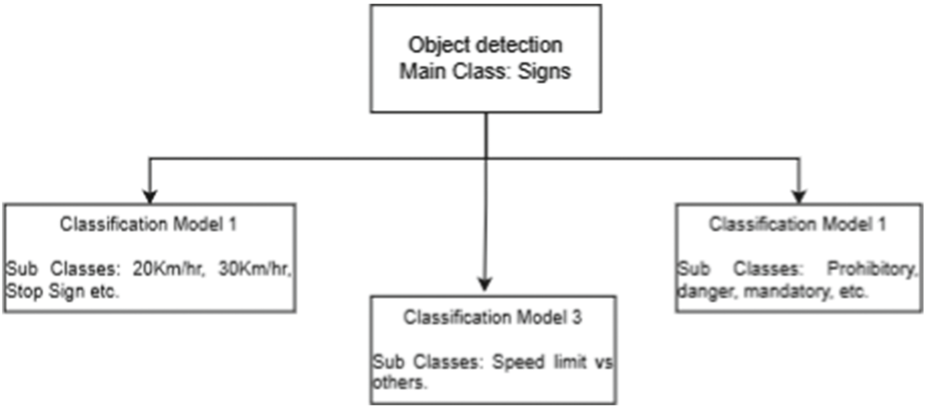

Object detection
Main Class: Signs
Classification Model 1
Sub Classes: 20Km/hr, 30Km/hr, Stop Sign etc.
Classification Model 3
Sub Classes: Speed limit vs others.
Classification Model 1
Sub Classes: Prohibitory, danger, mandatory, etc.

Fig. 2 Sign classification

1. Capture the video through the camera mounted on the moving vehicle
2. Extract the frames from the captured video clip
3. The gathered images are zoomed, panned and brightened for image augmentation
4. Noise is removed from the images by using the following 3 techniques:
   (a) Median filtering is performed for the smoothening of the images
   (b) De-blurring of images is performed using the Lucy Richardson filter
   (c) Histogram equalization for various lightening conditions.
5. From the noise free images traffic sign are detected using the YOLO v3 algorithm
6. The detected signs are classified using the proposed custom CNN model
7. The driver is alerted based on the results

Algorithm 1 Proposed approach.

The images are divided into three sub-classes, once they are collected using the object detection model, as illustrated in Fig. 2.

3.1 Dataset

The GTSRB Dataset is used to train the custom CNN model in the proposed system. Figure 3 shows the forty-three different traffic signs classes for training the model. The size of each image in this dataset is 32 x 32. It also shows that the signs in this dataset are divided into four main classes (prohibitory, danger, mandatory, and others).

3.2 Image augmentation

Due to the imbalance in the dataset, the predicted output of the model trained on this dataset would be incorrect. To ensure that the dataset is balanced, randomly remove images above the threshold from each class. The threshold is 250 images per bin. To balance the dataset, image augmentation is done. Each image is randomly zoomed, panned, and brightened, as shown in Fig. 4. In this way, the balanced dataset is expanded by adding these images.

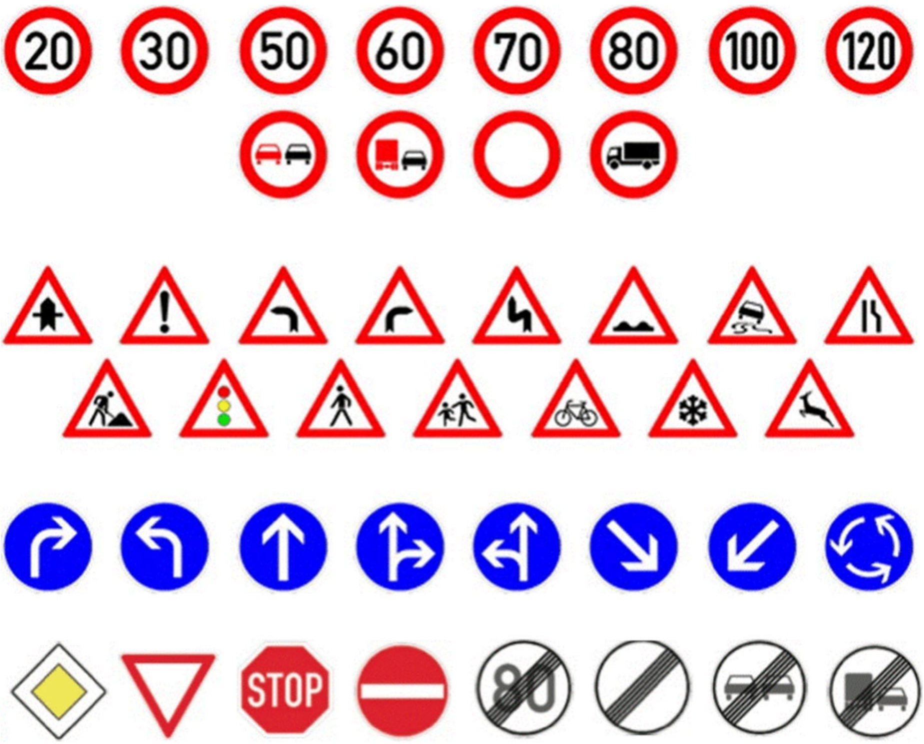


Fig. 3 43 classes of images divided into 4 categories

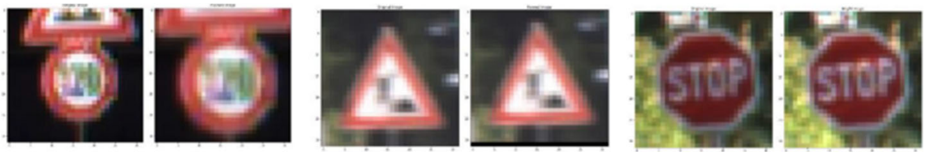
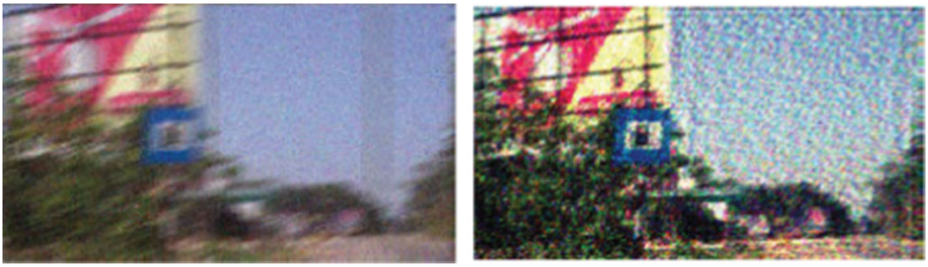


Fig. 4 Zoomed, panned, and brightened images



(a) Blurred image due to motion

(b) Pre-processed image

Fig. 5 5a. Blurred image , 5b. Pre processed image

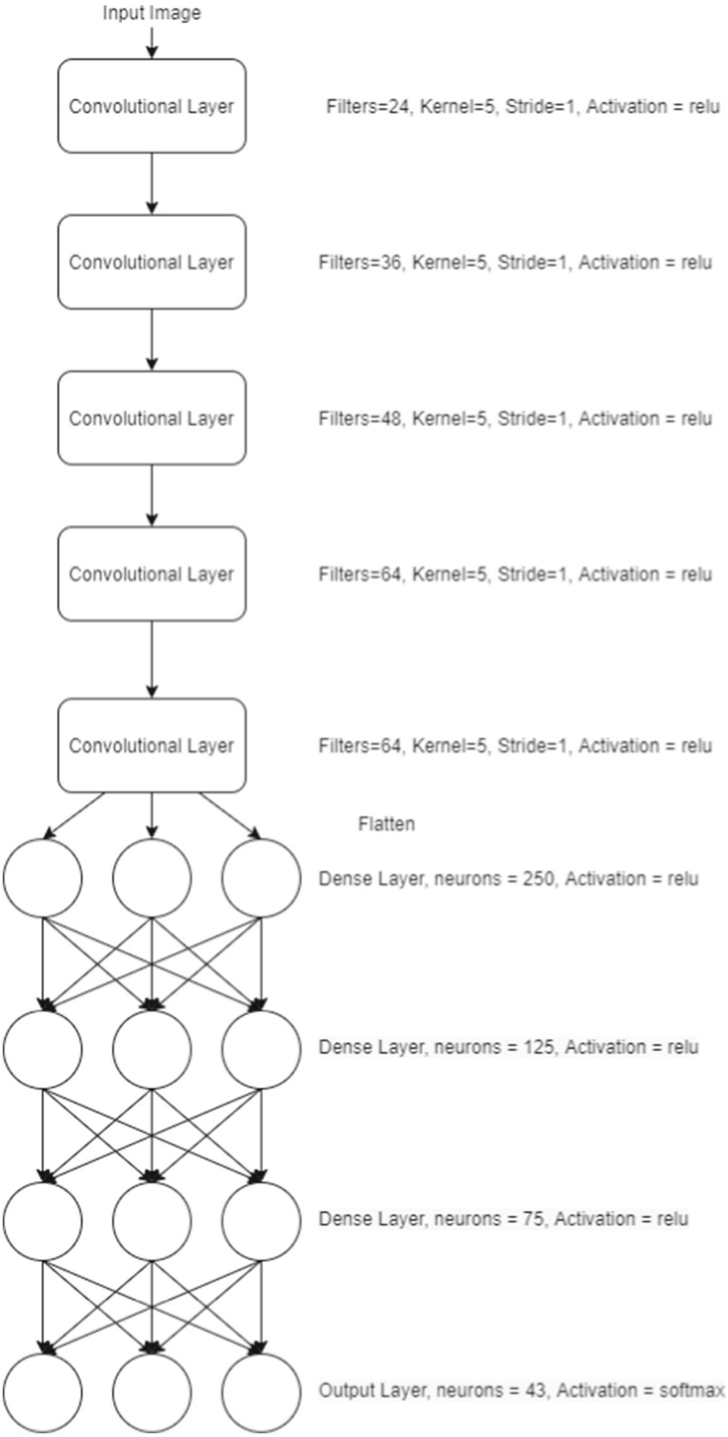

Input Image
Convolutional Layer
Filters=24, Kernel=5, Stride=1, Activation = relu
Convolutional Layer
Filters=36, Kernel=5, Stride=1, Activation = relu
Convolutional Layer
Filters=48, Kernel=5, Stride=1, Activation = relu
Convolutional Layer
Filters=64, Kernel=5, Stride=1, Activation = relu
Convolutional Layer
Filters=64, Kernel=5, Stride=1, Activation = relu
Flatten
Dense Layer, neurons = 250, Activation = relu
Dense Layer, neurons = 125, Activation = relu
Dense Layer, neurons = 75, Activation = relu
Output Layer, neurons = 43, Activation = softmax

Fig. 6 Customized convolutional neural network model

3.3 Image acquisition and pre-processing

Traffic sign video is captured with a HD camera installed on the vehicle. With its infrared low light capabilities, the device can also shoot high-quality photographs at night. The image reduction process samples the video at a certain sampling rate in order to obtain the image frames. After each image frame is pre-processed, it is sent for detection, classified into one of the four definite groups. The detected sign is then compared with the existing sign. The frequency of processed frames could be raised or lowered depending on the speed of the moving vehicle.

Images may appear hazy since videos are recorded with cameras mounted in speeding vehicles. Additionally, weather conditions, time of day, location, and brightness affect photographs' clarity, contrast, and brightness. Images may appear hazy since videos are recorded with cameras mounted in speeding vehicles. Due to this, detection will be challenging, and some pre-processing will be necessary to decrease their impact under various settings.

When captured by a moving vehicle, images will have a blurred effect, and edges will appear vague and washed out. Motion blur can be removed from images with the efficiency

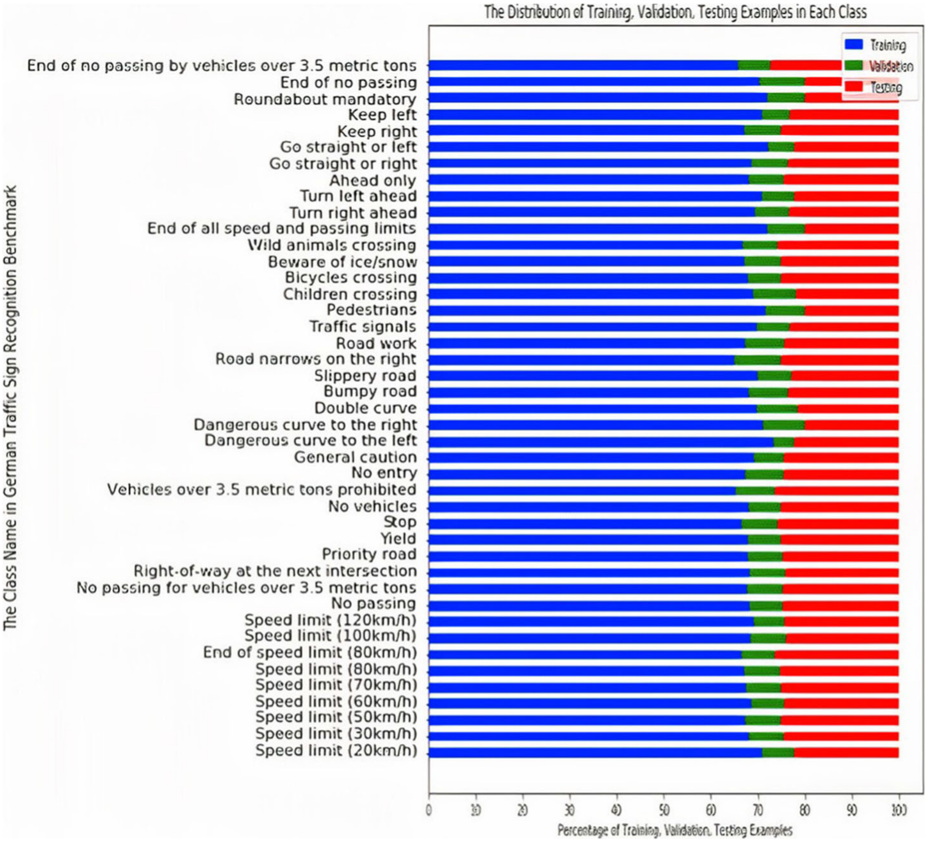


Fig. 7 Distribution of training, validation, and testing for each class

of frequency-domain techniques. Due to atmospheric turbulence, images can also be corrupted by white Gaussian noise. Motion blur and environmental disruption are shown in Fig. 5a. Figure 5b displays an image that has been median filtered, processed using the Lucy-Richardson filter, and then equalized by histogram for various lighting conditions.

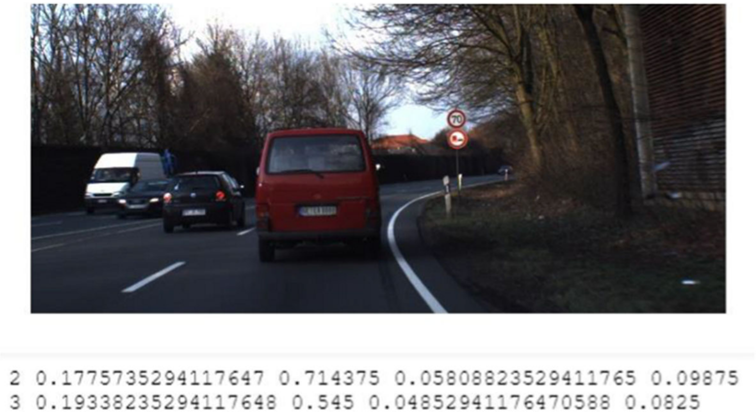
3.4 Building the custom CNN model

As shown in Fig. 6, a CNN model for classifying 43 sub-classes is trained using Keras and TensorFlow. The model is built using mini-batch stochastic gradient descent with the Adam optimizer. Figure 7 shows the division of the dataset into training, validation, and test sets. As this is a categorical classification, the labels are converted into one hot encoded array, as shown in Fig. 8. The first column represents the class ID where 0 represents Prohibitory, 1 represents danger, 2 is mandatory, and 3 is for others. The other column displays the actual bounding box for each class (x, y, w, h) within the image. The model is computed with a 0.0001 learning rate, a loss function of "categorical cross-entropy," and trained with a batch size of 32 with 10 epochs.

3.5 YOLO model for object detection

One of the best algorithm for object detection in real time is YOLO. The pre-trained YOLO can detect and classify 80 COCO objects (Common Objects in Context). Nevertheless, the YOLO architecture is modified to fit the dataset with a different number of classes in the proposed model. Two versions of YOLO are used, YOLOv3 and YOLOv4Tiny. Both versions incorporate the same neural network concept used in YOLOv1, which unifies object detection by dividing the input image into grids Every grid's centre determines its bounding box, and each box determines four coordinates, where 'tx' and 'ty' are the centre x and y coordinates, and 'tw' and 'th' are the coordinates of the box as shown in Fig. 9.

Furthermore, bounding boxes predict each object's label and confidence score based on its location. . There is a fixed dimension of the output tensor with S as size of the grid, B as the number of enclosing rectangles, and C as the number of named categories. The



```
2 0.1775735294117647 0.714375 0.05808823529411765 0.09875
3 0.19338235294117648 0.545 0.04852941176470588 0.0825
```

Fig. 8 Categorical classification

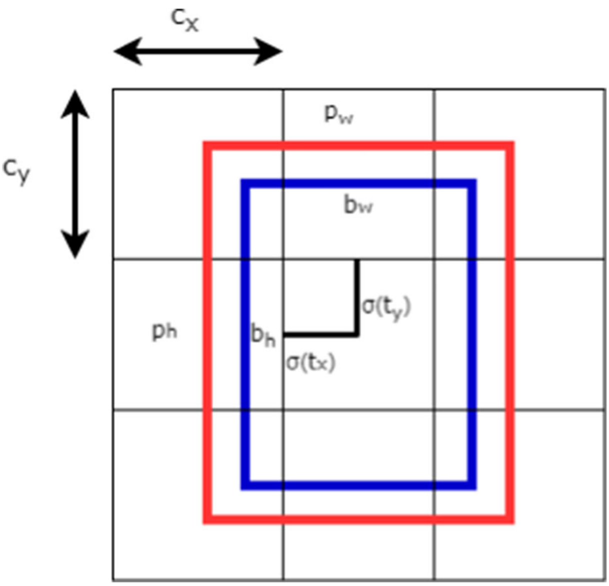


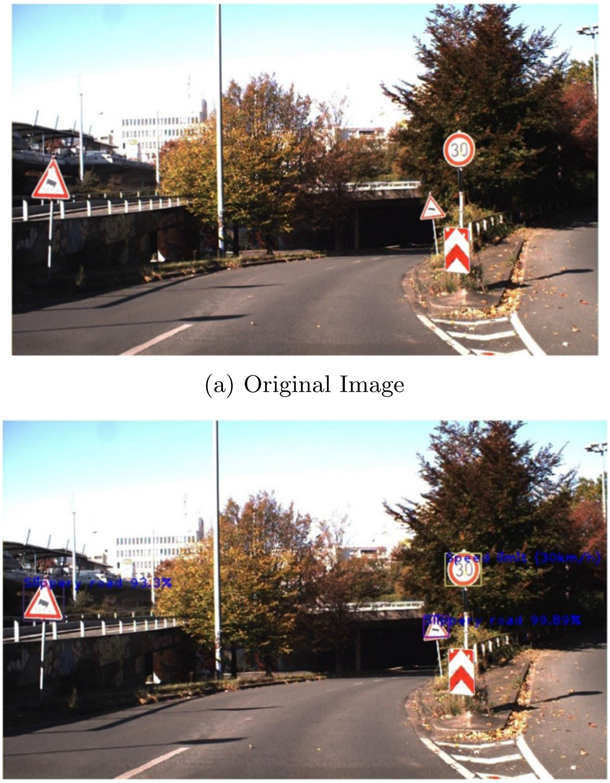
Fig. 9 Bounding boxes in YOLO v3

output from each bounding box is the label with the highest score. These bounding boxes are discarded if the output label confidence falls below a threshold specified by the user, which is 0.68. Yolo v3 uses binary cross-entropy loss to perform multi-label classification, which generates a probability for the observed object to belong to each label. Using the following relation, the output tensor size is calculated. S x S x [3 x [(4+1)+n]]

The number of classes is 4, i.e., prohibitory, danger, mandatory, and others, so there will be 27 filters in the S*S grid. The batch size and number of subdivisions were increased, the classes were lowered to 4, and the filters were lowered from 255 to 27. After these modifications, both YOLO v3 and YOLO v4-tiny are trained on the input images gathered by the vehicle's mounted camera.

3.6 Implementation

Once the model has been trained, it is saved and used for predictions. In the first section, the YOLO model is used, with the input being a video shot by the vehicle's camera. . Frames are extracted from that video and passed to the CNN model to classify the signs. After this the image will be classified as one among the 43 classes. The user will get a "No Sign Detected" prompt if a particular image does not contain a traffic sign. Using the "model predict" function, the model calculates a prediction based on the image's classification to determine which class it belongs to after returning a list of values. Multiple iterations revealed that, even when an image does not belong to any of the 43 classes, it still classifies it into one of the 43 classes, but the confidence score is relatively low. Therefore, the confidence level for separating actual traffic signs is 0.68. Classes are categorized from the confidence level of 0 to 1 in "model predict." As soon as an image has been classified, the sign text and its image are displayed to the driver on the vehicle's on-board display, as shown in Fig. 10a and b.



(a) Original Image

(b) Traffic Sign Recognition by the the proposed model

Fig. 10 10a. Original image, 10b. Output image

4 Experimental result analysis

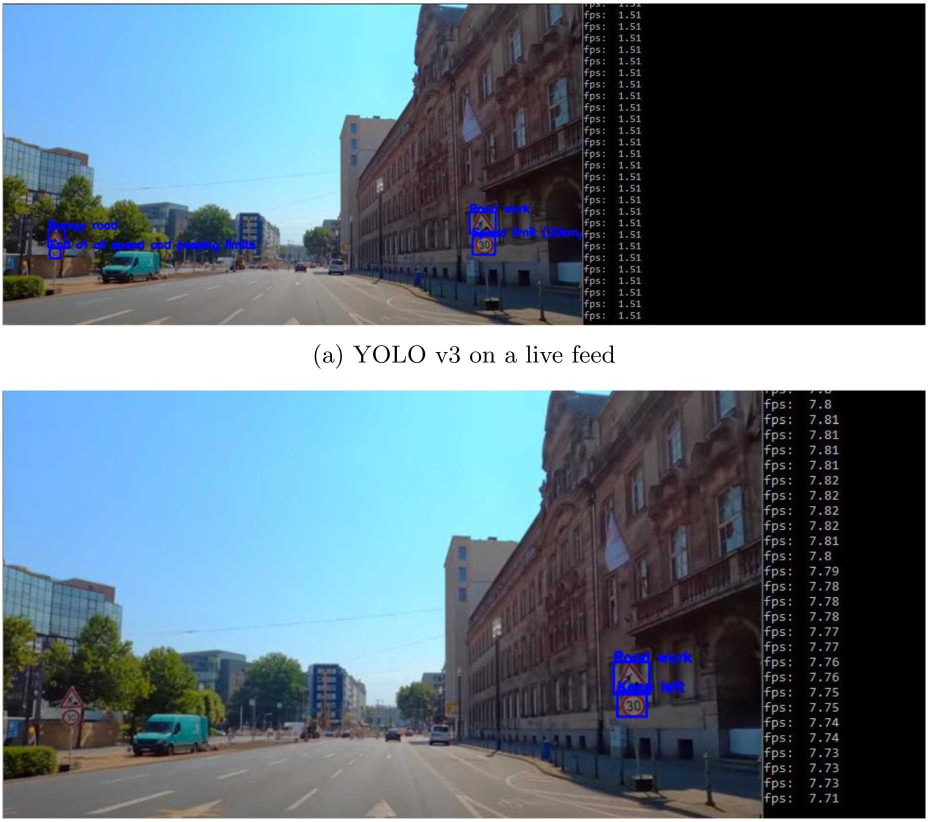
The execution was carried out on a computer with an INTEL CORE i5 processor, 8 GB of RAM, an AMD Radeon RX 5300 graphics card, and a 2TB hard drive. The traffic sign identification scripts are developed in Python 3.8.9 using OpenCV, NumPy, Pandas, PIL, and Darknet, among other libraries. The YOLO v3 model obtained a detection rate of 1.53 frames per second, whereas the YOLO v4-tiny model obtained a detection rate of 7.72 frames per second as shown in Fig. 11a and b.

The findings are based on 29330 training samples collected at random using a video camera mounted on a moving vehicle and on which the model conducted its detection procedure.

A total of 51900 images were present in the original dataset, but to balance the dataset, the image count was reduced to 30000. After image augmentation, the image count increased to 41900. 12570 images were used for testing and validation.

A comparative study was made between YOLO v3 and YOLO v4-Tiny based on the quantitative measures shown in Fig. 12. Training and Validation accuracy and loss for 10 epochs is shown in Fig. 13.

The proposed model's accuracy is 95.85%, the F1 score is 0.9528, and the precision value is 0.91. A comparative analysis is performed to compare the efficiency of the existing



```
fps:  1.51
fps:  1.51
fps:  1.51
fps:  1.51
fps:  1.51
fps:  1.51
fps:  1.51
fps:  1.51
fps:  1.51
fps:  1.51
fps:  1.51
fps:  1.51
fps:  1.51
fps:  1.51
fps:  1.51
fps:  1.51
fps:  1.51
fps:  1.51
fps:  1.51
fps:  1.51
fps:  1.51
fps:  1.51
fps:  1.51
fps:  1.51
fps:  1.51
fps:  1.51
fps:  1.51
```

(a) YOLO v3 on a live feed

```
fps:  7.8
fps:  7.81
fps:  7.81
fps:  7.81
fps:  7.81
fps:  7.82
fps:  7.82
fps:  7.82
fps:  7.82
fps:  7.81
fps:  7.8
fps:  7.79
fps:  7.78
fps:  7.78
fps:  7.78
fps:  7.77
fps:  7.77
fps:  7.76
fps:  7.76
fps:  7.75
fps:  7.75
fps:  7.74
fps:  7.74
fps:  7.73
fps:  7.73
fps:  7.73
fps:  7.71
```

(b) YOLO v4-tiny on a live feed

Fig. 11 Comparison between YOLO v3 and YOLO v4-tiny

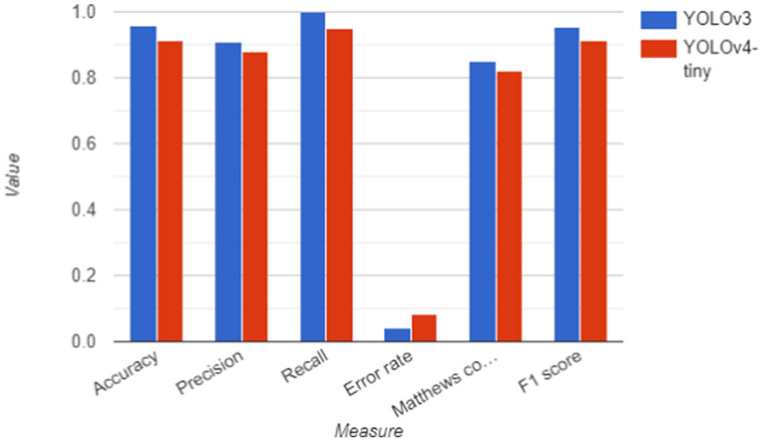


Fig. 12 Comparison between YOLOv3 and YOLOv4-tiny

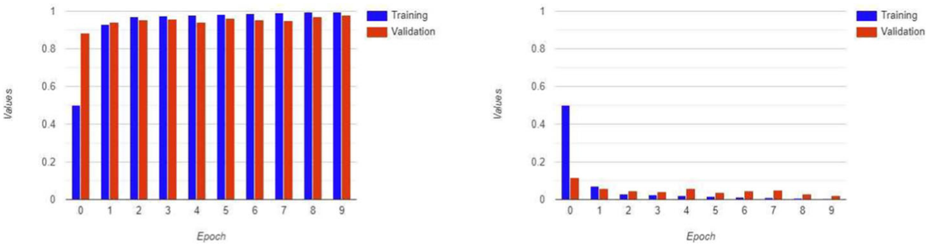


Fig. 13 Training and validation accuracy and loss for 10 epochs

systems with the suggested method. The current models fail to achieve higher accuracy when the images are captured in poor weather conditions or the captured images are blurry due to the fast motion of the vehicle.

Compared to the MobileNets and Inceptionv3 algorithms, the proposed model implementing YOLOv3 and YOLOv4-tiny has produced the highest accuracy, as shown in Fig. 14.

ROC curves are appropriate when the observations are balanced between each class. As the data is equally balanced between each class, an ROC curve is drawn for the proposed methodology to compare the overall performance. The German Traffic Sign Recognition Benchmark collection dataset was classified with good sensitivity, specificity, and F1-score (99.2%, 99.1%, and 98.85%). The value of sensitivity (99.2%) means that the sum of the false negatives is low, while the specificity value (99.1%) means that the sum of the true negatives is high.

Furthermore, the ROC curves are added between the true positive rate (TPR) and the false positive rate (FPR) to compare the overall performance as shown in Fig. 15. The area under the ROC curve (AUC) was calculated to be 99.3% for the proposed model, customised CNN. It was observed that the proposed network outperforms the best with good efficiency.

For ROC curve:

$$TruePositiveRate = \frac{TruePositives}{(TruePositives + FalseNegatives)}$$

$$FalsePositiveRate = \frac{FalsePositives}{(FalsePositives + TrueNegatives)}$$

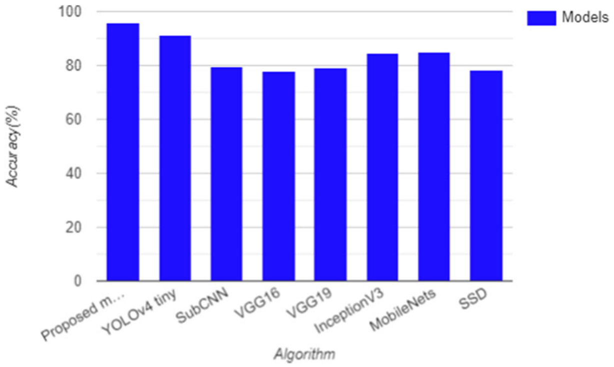


Fig. 14 Analysis of the proposed method in comparison to the existing algorithms

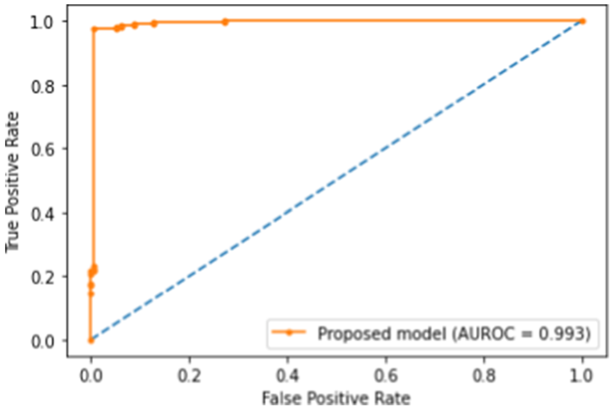


Fig. 15 ROC curve

5 Conclusion and future scope

The proposed approach automatically detects and classify the various traffic signs. The system uses YOLO v3 algorithm and a customized CNN model. Based on experimental results and quantitative measures assessing multiple scenarios, the proposed method is proven reliable for the proper implementation of the system. The use of image augmentation techniques and filters like Lucy-Richardson remove the noise from the images without blurring the edges, thus enabling the system to detect traffic signs in lousy weather conditions. This is a significant contribution to driving since it would make driving easier and safer without compromising safety.

Furthermore, this system can be integrated without the need for a lot of hardware, which expands its reach. System works better when the vehicle is moving in a straight-line path, but it can also be enhanced to give the same results on the turns. The framework can be enhanced to provide an embedded warning system with a camera in the vehicle's center. As an extension of this system, it can recognize traffic signals and inform the user of the time it will take to reach those signals and their status.

Funding There is no funding involved in this work.

Data Availability The datasets used for the current study are from https://www.kaggle.com/datasets/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign

Declarations

Conflict of Interests There is no conflict of interest.

References

1. Ali NM, Mustafah Y, Rashid NM (2013) Performance analysis of robust road sign identification. In: IOP Conference series: materials science and engineering, vol 53. IOP Publishing, p 012017
2. Arcos-Garcia A, Alvarez-Garcia JA, Soria-Morillo LM (2018) Evaluation of deep neural networks for traffic sign detection systems. Neurocomputing 316:332–344
3. Ayachi R, Afif M, Said Y, Atri M (2020) Traffic signs detection for real-world application of an advanced driving assisting system using deep learning. Neural Process Lett 51(1):837–851

4. Bi Z, Yu L, Gao H, Zhou P, Yao H (2020) Improved vgg model-based efficient traffic sign recognition for safe driving in 5g scenarios. Int J Mach Learn Cybern:1–12
5. Ciresan D, Meier U, Masci J, Schmidhuber J (2012) Multi-column deep neural network for traffic sign classification. Neural networks. In: The international joint conference on neural network, IDSIA-USI-SUPSI Galleria, vol 2
6. Ćorović A, Ilić V, Urić S, Marijan M, Pavković B (2018) The real-time detection of traffic participants using yolo algorithm. In: 2018 26th Telecommunications forum (TELFor). IEEE, pp 1–4
7. Gudigar A, Chokkadi S, Raghavendra U (2016) A review on automatic detection and recognition of traffic sign. Multimed Tools Appl 75(1):333–364
8. Han C, Gao G, Zhang Y (2019) Real-time small traffic sign detection with revised faster-rcnn. Multimed Tools Appl 78(10):13263–13278
9. Huang S-C, Lin H-Y, Chang C-C (2017) An in-car camera system for traffic sign detection and recognition. In: 2017 Joint 17th world congress of international fuzzy systems association and 9th international conference on soft computing and intelligent systems (IFSA-SCIS). IEEE, pp 1–6
10. Isa ISBM, Choy JY, Shaari NLABM (2022) Real-time traffic sign detection and recognition using raspberry pi. Int J Electr Comput Eng 12(1):331
11. Islam MZ, Islam MM, Asraf A (2020) A combined deep cnn-lstm network for the detection of novel coronavirus (covid-19) using x-ray images. Inform Med Unlocked 20:100412
12. Islam MM, Tayan O, Islam MR, Islam MS, Nooruddin S, Kabir MN, Islam MR (2020) Deep learning based systems developed for fall detection: a review. IEEE Access 8:166117–166137
13. Liu C, Tao Y, Liang J, Li K, Chen Y (2018) Object detection based on yolo network. In: 2018 IEEE 4th Information technology and mechatronics engineering conference (ITOEC). IEEE, pp 799–803
14. Mallela NC, Volety R, Nadesh R et al (2021) Detection of the triple riding and speed violation on two-wheelers using deep learning algorithms. Multimed Tools Appl 80(6):8175–8187
15. Mehta S, Paunwala C, Vaidya B (2019) Cnn based traffic sign classification using adam optimizer. In: 2019 International conference on intelligent computing and control systems (ICCS). IEEE, pp 1293–1298
16. Panigrahi R, Borah S, Bhoi AK, Ijaz MF, Pramanik M, Jhaveri RH, Chowdhary CL (2021) Performance assessment of supervised classifiers for designing intrusion detection systems: a comprehensive review and recommendations for future research. Mathematics 9(6):690
17. Panigrahi R, Borah S, Bhoi AK, Ijaz MF, Pramanik M, Kumar Y, Jhaveri RH (2021) A consolidated decision tree-based intrusion detection system for binary and multiclass imbalanced datasets. Mathematics 9(7):751
18. Qian R, Zhang B, Yue Y, Wang Z, Coenen F (2015) Robust chinese traffic sign detection and recognition with deep convolutional neural network. In: 2015 11th International conference on natural computation (ICNC). IEEE, pp 791–796
19. Rahman MM, Manik MMH, Islam MM, Mahmud S, Kim J-H (2020) An automated system to limit covid-19 using facial mask detection in smart city network. In: 2020 IEEE International IOT, electronics and mechatronics conference (IEMTRONICS). IEEE, pp 1–5
20. Ramalingam SP, Nadesh R, NC SK (2018) Robust face recognition using enhanced local binary pattern. Bull Electr Eng Inform 7(1):96–101
21. Srinivasu PN, SivaSai JG, Ijaz MF, Bhoi AK, Kim W, Kang JJ (2021) Classification of skin disease using deep learning neural networks with mobilenet v2 and lstm. Sensors 21(8):2852
22. Tai S-K, Dewi C, Chen R-C, Liu Y-T, Jiang X, Yu H (2020) Deep learning for traffic sign recognition based on spatial pyramid pooling with scale analysis. Appl Sci 10(19):6997
23. Wu Y, Li Z, Chen Y, Nai K, Yuan J (2020) Real-time traffic sign detection and classification towards real traffic scene. Multimed Tools Appl 79(25):18201–18219
24. Yadav S, Patwa A, Rane S, Narvekar C (2019) Indian traffic signboard recognition and driver alert system using machine learning. Int J Appl Sci Smart Technol 1(1):1–10
25. Yuan Y, Xiong Z, Wang Q (2019) Vssa-net: vertical spatial sequence attention network for traffic sign detection. IEEE Trans Image Process 28(7):3423–3434
26. Zhang C, Yue X, Wang R, Li N, Ding Y (2020) Study on traffic sign recognition by optimized lenet-5 algorithm. Int J Pattern Recog Artif Intell 34(01):2055003

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.