



Enhancing traffic sign recognition (TSR) by classifying deep learning models to promote road safety

Anju J. Prakash¹ · S. Sruthy²

Received: 6 December 2023 / Revised: 6 February 2024 / Accepted: 23 February 2024 / Published online: 30 March 2024
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2024

Abstract

Traffic sign recognition (TSR) systems are essential for strengthening road safety, enhancing traffic management, and promoting efficient driving due to the ever-increasing number of vehicles on the roads and the need for better transportation systems. Modern intelligent transportation systems rely on TSR systems to help with the detection, categorization, and interpretation of traffic signs. These technologies improve driving assistance and road safety by automatically recognizing and comprehending the information contained in traffic signs through the use of computer vision and AI. By automating the recognition process, these systems lessen the load on drivers, limit the possibility of human error, and help to keep the roads safe by reducing infractions and accidents. TSR systems are also necessary for the creation and application of advanced driver assistance systems, laying the foundation for a future that places a premium on safer and more effective transportation. An efficient deep learning (DL)-based model for TSR was presented in this paper. The suggested TSR system consists of two models: a traffic sign detection model that uses RetinaNet and a traffic sign classification model that uses DenseNet-121. The performance evaluation of detection and classification models relies on the utilization of the GTSDB and GTSRB datasets. The performance of the detector was found to surpass that of the existing traffic sign detectors. The traffic sign classifier achieved better classification performance with an accuracy of 98.32%. Using powerful TSR technology could revolutionize transportation and provide a better environment for all drivers.

Keywords Intelligent transportation systems · Road safety · Driver assistance · Traffic management · Image processing · Computer vision · Artificial Intelligence · Detection · Classification

1 Introduction

Images are gradually becoming a crucial use in day today life, along with their acquisition, storage, and subsequent processing. The method that captured photographs are converted to digital form and then shared through various online and mobile application platforms has been entirely improved by digital image processing (DIP) technologies. According to research, human errors have been identified as the primary cause of over 93% of road accidents resulting in fatalities and

injuries [1]. Due to this reality, it is now important for the research community to provide technological advances that will assist society and the government in developing regulations to lessen accidents. In addition to the 50 million injuries sustained each year, almost 25 million of those injuries result in permanent disability [2]. Each year, traffic accidents injure nearly 1.3 million people worldwide, with nearly 2% of those injuries occurring in EU nations. The WHO estimates that the cost of traffic accidents in most nations is roughly 3% of their GDP [3].

DIP involves utilizing computer technology to algorithmically manipulate digital images. The utilization of algorithms helps address issues related to noise and distortion that may arise during the processing stage. Here are some of the technological applications and techniques used in DIP. DIP encompasses a range of techniques such as image manipulation, image recovery, independent component analysis, and neural networks, among others. DIP applications encompass a wide range of tasks such as categorization, extracting key

✉ Anju J. Prakash
jpanju@gmail.com

S. Sruthy
sruthy78@gmail.com

¹ Department of Computer Science and Engineering, Sree Buddha College of Engineering, Pattoor, India

² Department of Computer Science and Engineering, St. Joseph's College of Engineering and Technology, Palai, India

attributes, analyzing signals at various scales, recognizing patterns, and projecting data. DIP offers a comprehensive set of industry-standard algorithms and workflow tools for tasks such as image processing, analysis, visualization, and the creation of new algorithms [4].

The terms “object detection” and “object recognition” are frequently used to describe computer vision tasks that involve locating things in digital environments or digital images. The acquired digital images undergo a series of preprocessing steps, segmentation, and subsequent classification for further analysis and interpretation. Image classification involves the process of accurately determining the correct class or category to which an object belongs. Following image segmentation for ROI detection, object localization finds the presence of the target object [5]. A few examples of images in which traffic signs are not clearly visible are shown in Fig. 1. Additionally, traffic indicators are less likely to catch drivers’ attention during these times.

These conditions make driving challenging and increase the number of traffic accidents. Traffic signs control traffic or notify drivers of information connected to the signs, keeping them safe and comfortable while driving. Figure 2 displays a few types of traffic sign images.

Traffic signs provide drivers and pedestrians with visual cues to help them navigate their way and control traffic by informing them of the conditions and restrictions of the road [7]. The primary causes of many accidents are hazardous roads and excessive accelerations by other road users. To mitigate these issues, the route is equipped with variable speed limits that are adjusted based on the road’s condition, traffic density, and visibility, ensuring a proactive approach to avoid potential problems. The design of traffic signs in different countries is governed by their respective legal frameworks, ensuring adherence to specific regulations and standards [8]. Traffic signs are categorized according to their color, shape, and texture. For example, information signs have blue signals, warning signs have triangles, and prohibition signs have red rims. Some of the categories of Indian traffic signs are depicted in Fig. 3. The interpretation of traffic signs is a significant challenge for the areas of computer vision and intelligent systems. As they are made to alert drivers to prospective hazards and existing road conditions, traffic signs efficiently support drivers and help them drive more safely. These signs typically have bold colors and stiff, simple forms like circles, triangles, and regular polygons [9]. As a result, TSR is gaining significance in the context of autonomous vehicles, highway upkeep, and systems that aid drivers. The suggested work in this study comprises object detection and classification which falls under the broad category of computer vision and image processing.

A specific traffic sign needs to have clear information about it. There are distinctive design styles used. Differences are based on size, shape, and color. Road signs offer ideal

solutions to ensure safe driving. The majority of collisions are caused by drivers who either fail to see a stop sign or fail to pay attention at crucial moments. Additionally, poor lighting makes it difficult for drivers to see. The headlights of the vehicles coming from opposite directions throughout the night, especially in severe weather conditions like rain, fog, and snow, may distract or even blind the drivers. A good computer vision system is necessary for the majority of tasks, and as these tools advance, they may even be able to replace people. By utilizing TSR technology, vehicles have the capability to interpret and comprehend road signs such as “narrow bridge” or “hump ahead” that are posted on the roadside. One of the crucial systems for managing driver safety is traffic management, which directs drivers in the proper direction at all times. In order to create an efficient intelligent road transport system, a well-defined automatic intelligent TSR system has been designed. Figure 4 depicts the overall structure of a TSR system.

When driving on busy roads, traffic sign detection might help drivers pay attention to the posted speed limits and other road signs on the console screen. The improvement of general road safety depends heavily on the accurate identification of traffic signs. Real-time warnings and notifications can be sent to drivers to encourage them to obey traffic laws and prevent accidents when traffic signs are accurately detected and recognized. This technology can drastically lower the danger of collisions and increase overall road safety by delivering accurate information on speed limits, stop signs, no-entry zones, and other traffic indicators. So, a useful DL-based model for TSR was proposed in this paper.

2 Literature review

This section provided a summary of the current studies and methodologies employed in the area of TSR.D. Tabernik et al. [10]. This approach is used to detect the 200 traffic sign categories contained in our new dataset. The findings are presented for extremely difficult traffic sign categories that have not before been studied. It presents a comprehensive analysis of the deep learning method for detecting traffic signs with large intra-category appearance variation and show that the proposed approach achieves error rates of less than 3%, which is sufficient for deployment in practical applications of traffic sign inventory management.

The traffic sign classification and recognition studies done by Cao et al. [11] are based on the German Traffic Sign Recognition Benchmark. The continual training and testing of the network model result in positive prediction and accurate recognition of traffic signs. According to the experimental data, the accurate recognition rate of traffic signs is 99.75%, and the average processing time per frame is 5.4 ms.



Fig. 1 Non-visible conditions of traffic sign: **a** unfavorable light condition, **b** partial occlusion, **c** cluttered background, **d** small size [6]



Fig. 2 Sample traffic sign categories

In 2019, Sun et al. [12] the training and validation datasets were created using image augmentation. The first classifier was trained using 40,000 photographs, including 28,000 positive images (images that contain traffic signs) and 12,000 negative images (images that do not contain any traffic signs). With 2400 positive photographs and 1200 negative images, 3600 images have been employed to train the second classifier. The photographs are processed to identify the region of interest, which is subsequently classified by two CNN classifiers.

In 2019, Vennelakanti et al. [13] describe a method for traffic sign detection and identification that uses image processing for sign detection and an ensemble of convolutional neural networks (CNN) for sign identification in this paper. CNNs have a high recognition rate, making them appealing for use in a variety of computer vision tasks. TensorFlow is utilized in the CNN implementation. On the Belgian and German data sets, we achieved recognition accuracies of more than 99% for circular signs.

In 2019, William et al. [14] solved the traffic sign detection problem using cutting-edge multi-object detection systems such as faster recurrent convolutional neural networks

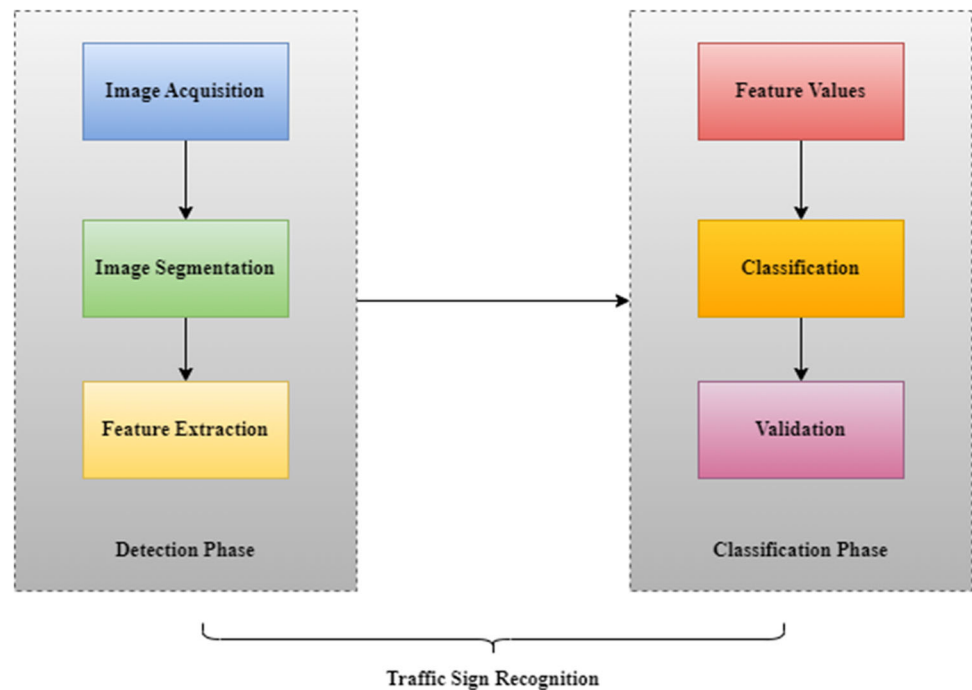
(F-RCNN) and single shot multi-box detector (SSD) in conjunction with various feature extractors such as MobileNet v1 and Inception v2, as well as Tiny-YOLOv2. However, because they produced the best results, F-RCNN Inception v2 and Tiny YOLO v2 will be the focus of this paper. The above-mentioned algorithms were refined using data from the German Traffic Signs Detection Benchmark (GTSDB).

In 2019, Kamal et al. [15] objective recommended network outperforms the current state-of-the-art object detection networks, such as the Faster RCNN inception ResNet V2 and R-FCN ResNet 101, by a large margin, with precision and recall of 94.60 and 80.21%, respectively, on this part of the dataset. Furthermore, the network is tested on the German Traffic Sign Detection Benchmark (GTSDB) dataset, achieving precision and recall of 95.29% and 89.01%, respectively.

According to 2020, Liang et al. [16], the detection and recognition of targets by networking with various feature scales has greatly improved, with recall and accuracy of 95.32% and 93.13%, correspondingly. Finally, the algorithm for traffic sign identification and authentication is tested on the NVIDIA Jetson Tx2 platform and offers great results at 28 frames per second.



Fig. 3 Traffic signs in India

Fig. 4 General workflow of TSR system

In 2019, Alghmgham et al. [17] an autonomous traffic and road sign (ATRS) detection and recognition system is built using a deep convolutional neural network (CNN). The suggested system detects and recognizes traffic sign images in real time. This study also includes a recently created database of 24 different traffic signs gathered from random road sides in Saudi Arabia. The photographs were obtained from various viewpoints and included various elements and circumstances. A total of 2718 photographs were collected to create the Saudi Arabian Traffic and Road Signs (SA-TRS-2018) dataset.

In 2019, Rajendran et al. [18] the traffic sign detection network is formed using YOLOv3, and the traffic sign class recognizer is formed by a CNN-based classifier. The German Traffic Sign Detection Benchmark (GTSDB) [5] dataset is used for network training and evaluation, and the classifier performance is validated using the German Traffic Sign Recognition Benchmark (GTSRB) [6] dataset.

In 2019, Yuan et al. [19] first, traffic signs are often small-sized items, making them more difficult to detect than large ones; second, without context information, it is difficult to distinguish false targets that resemble real traffic signs in complex street scenes. To address these issues, we offer a revolutionary end-to-end deep learning strategy for detecting traffic signs in complicated situations.

In 2019, Xu et al. [20] the upgraded Faster RCNN achieves 29.8 frames per second and a mean average precision of 99.5%, outperforming state-of-the-art approaches and making it more appropriate for traffic sign detection.

Furthermore, the suggested model is applied to the Tsinghua-Tencent 100 K (TT100K) dataset, yielding a competitive detection performance.

According to 2021 Ahmed et al. [21], this approach achieves an overall precision and recall of 91.1% and 70.71%, correspondingly, which is a 7.58 and 35.90% improvement in precision and recall over the current benchmark. Furthermore, they compare this method to various CNN-based TSDR approaches and show that it beats them by a wide margin.

In 2019, Han et al. [22] in actuality, traffic indicators, such as traffic signals or distant road signs, generally cover less than 5% of the total image in the camera's view. As a result, we devote great effort in this study to offer a real-time tiny traffic sign identification solution based on improved Faster RCNN.

In 2020, Liu et al. [23] it suggests an a two-stage adaptable classification loss function for regional proposal networks (RPN) and fully interconnected neural networks within DR-CNN to enhance training effectiveness and distinguish challenging negative samples from easy positive ones. Finally, we test our suggested technique on the novel and difficult Tsinghua-Tencent 100 K dataset.

In 2020, Jin et al. [24] in this research, it is suggested MF-SSD, an enhanced (single shot detector) SSD algorithm for traffic sign recognition based on multi-feature fusion and augmentation. First, low-level features are combined with high-level features to increase the detection of small objects in the SSD. The features in separate channels are then enhanced to detect the target by augmenting effective

channels characteristics and minimizing invalid channel features.

In 2022, Yi Shi and Young Chun Ko [25] using Chinese English grammar as the research object, three groups of phonetic data were chosen to serve as experimental auxiliary data, based on the convolutional neural network, through the preset reset of the model's pronunciation the identification system, the speech system's sampling and recognition extraction, wrong speech detection, and feature assessment of the multi-level data stream tandem, while the tests are carried out with CU-CHLOE language learning database.

In 2021, Ojha et al. [26] the study provides an effective step-by-step description of the procedure's flow in recognizing autos in various real-world scenarios. The suggested model achieves 94.66% accuracy and 95.13% precision after being trained on a combination of two benchmark datasets.

In 2021, Dewangan et al. [27] in these phases, the operational mechanism is primarily based on vision sensors, which allow these vehicles to comprehend heterogeneous and changing surroundings and make appropriate decisions. This study identifies numerous cutting-edge approaches and phase-wise datasets that have been used in the literature. It emphasizes the progress in various phases, problems, and scopes for the creation and production of intelligent vehicle systems. Table 1 shows the deep learning approaches to traffic sign recognition.

Despite tremendous improvements in TSR, there is still limitations in tackling the difficulties of accurate detection in challenging environments. Certain constraints in the current research on TSR have been discussed here. In the area of health care, such as disease diagnosis, previous research in digital imaging employing CNN, YOLO, and SVM algorithms has produced trustworthy results. However, there has been a lack of studies producing good results in the specific case of traffic sign detection. Additionally, studies that considered the usage of HSV or HSI color spaces, which are thought to be more beneficial for human eye perception with regard to sunlight and color intensity. The majority of study employed RGB color spaces. Previous research in related fields predominantly focused on grayscale images, leaving a gap in understanding the impact and effectiveness of utilizing color images, which are more pertinent in the context of traffic signs. Principal component analysis (PCA) techniques were found to be a component of dimensionality reduction techniques, which aid in further feature optimization. The chi-square, RRelief, and MRR approaches are some of these methods. The traffic sign data sets have not before been examined using these techniques. For performance improvement measurements, the authors choose to take these approaches into account on various image datasets. It has been discovered that the chi-square selection feature is one of the efficient filter-based selection techniques that can aid in the removal of superfluous features for optimizing overall processing time

and raising accuracy levels. Additionally, there was a gap in the absence of research on chi-square selection for traffic signs. So, to overcome the identified limitations, an effective DL-based TSR model was proposed in this paper.

3 Materials and methods

The suggested model contains two phases. The initial stage involves identifying traffic signs, followed by categorizing them based on their characteristics. The main objective of a traffic sign detection system is to precisely recognize and ascertain the exact location of traffic signs present in an image. Bounding boxes are utilized to enclose objects, capturing their spatial extent or location. The algorithm categorizes these identified labels into designated groups throughout the subsequent stages of TSR. The proposed TSR model comprises of a traffic sign detection module based on RetinaNet and a traffic sign classification module based on DenseNet-121. Figure 5 displays the block diagram of the suggested TSR model.

3.1 Dataset description

Two distinct datasets were utilized to assess the effectiveness of the proposed models in accurately recognizing and classifying traffic signs. The GTSDDB dataset [25] and the GTSRB dataset [26] have gained extensive usage in TSR tasks. Video sequences are utilized to create the GTSDDB dataset. The task is quite challenging as it involves a diverse range of images captured in different weather conditions and lighting situations, encompassing urban, rural, and highway scenes, making it a comprehensive and representative dataset.

The dataset has a total of 900 images, where 600 images are used for training purposes and the remaining 300 images are reserved for testing.

Each image has dimensions of 1360×800 pixels and is in raw PPM format. The test images have 360 traffic signs, while the training ones have 846. The images show traffic signs with sizes ranging from 16 to 128 pixels. Figure 6 displays sample images from the GTSDDB collection.

There are around 50,000 traffic sign images in the GTSRB dataset, and they are classified into 43 distinct categories. The classifier model, trained on a dataset called GTSRB, utilizes 39,209 training images to learn and is then evaluated on 12,630 test images to assess its performance. Sample images from the GTSRB dataset are depicted in Fig. 7.

3.2 Image preprocessing

Image preprocessing techniques encompass a range of operations that are applied to images prior to conducting subsequent analysis or tasks. These methods seek to improve

Table 1 Summary of deep learning approaches to traffic sign recognition

References	Algorithm	Dataset	Advantages	Disadvantages	Accuracy (%)	Research gap
Liu et al. [23]	DR-CNN	Tsinghua-Tencent 100 K	Traffic in all directions	Increases total delay, especially off-peak	98	Detecting and recognizing traffic signs on roads is the low quality of images owing to variable urban environments
Jin et al. [24]	MF-SSD	GTSRB	Traffic in a signal system corridor	Increase fuel consumption		To model complex urban road surroundings, which is a similar shortcoming inherited from features-based methods
Vincent et al. [40]	CNN	GTSRB	Provisions for side street vehicles to enter the traffic stream	Time delay and less appropriate route	96.15	To solve the traffic sign recognition problem using a CNN-based object detection framework
Haque et al. [41]	DeepThin	GTSRB BTSC	Less time delay	Not properly locate	88.49	Object tracking is an algorithm that assigns a unique ID to a detected object and maintains the ID value unchanged even as frames flow
Fang et al. [42]	MicroNet-BF	GTSRB	Possible improvements in capacity	Time variation	98.11	Since urban road scenes can have different aspects from benchmark images, it is crucial to investigate deep learning models for real-world application
Shi et al.[25]	–	CU-CHLOE language learning database	Traffic moments in all directions	Bad timing and road accidents	98.69	Utilizing features from the color and shape of a given image have been proposed for traffic sign recognition tasks

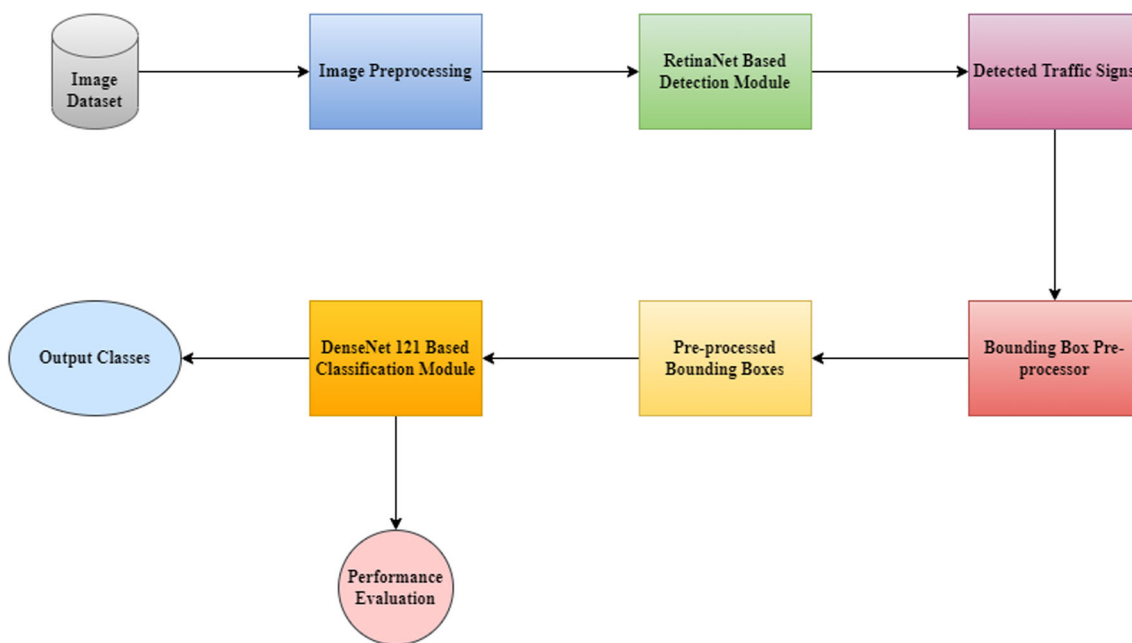


Fig. 5 Block diagram of proposed TSR model



Fig. 6 Sample images from GTSDb dataset

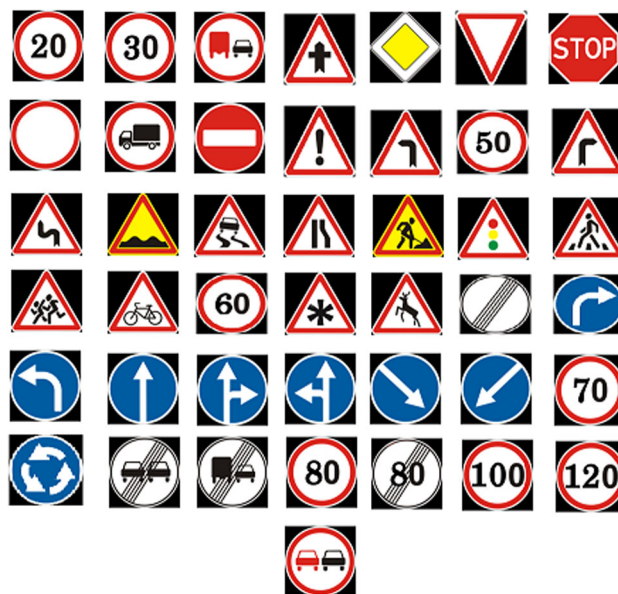


Fig. 7 Sample images from GTSRB dataset

image quality by lowering noise, repairing distortions, and extracting relevant characteristics. In computer vision and image processing tasks, image preprocessing is essential because it increases the precision, dependability, and effectiveness of executing techniques. The most often used image preprocessing techniques include image scaling and resizing, denoising, contrast boosting, normalization, rotation, and cropping, among others. Preprocessing approaches minimize image noise, which can impair algorithm performance and accuracy. Noise reduction enhances image quality and aids in the extraction of important information. Preprocessing approaches boost the performance and effectiveness of

algorithms by preparing images for upcoming tasks. Better results and interpretations are produced by enhancing image quality and feature representation.

3.3 Traffic sign detection module based on RetinaNet

The RetinaNet model is employed as a detector for identifying traffic signs. RetinaNet is an object detection model

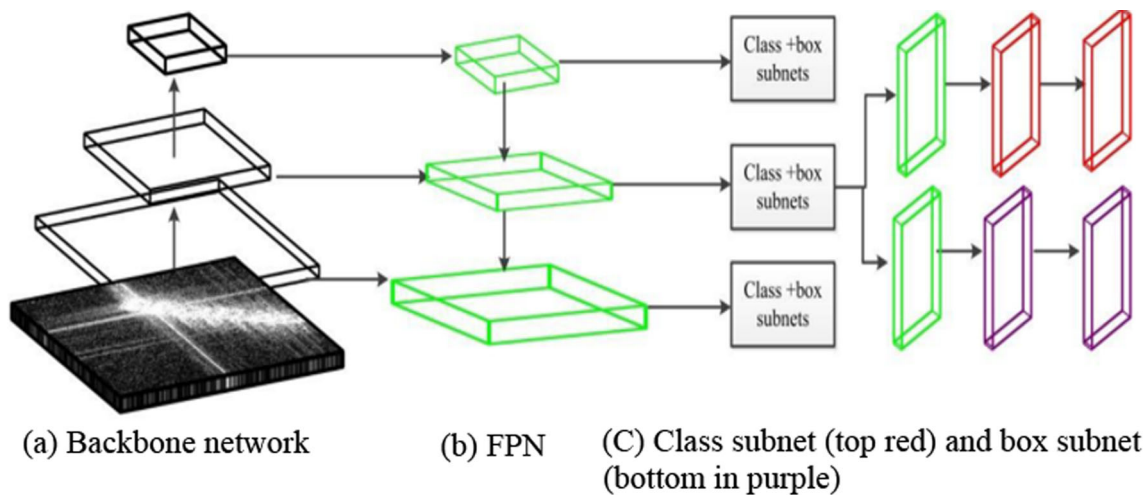


Fig. 8 Basic architecture of RetinaNet detector

that combines a Feature Pyramid Network (FPN) and uses a unique loss function called Focal Loss. The RetinaNet architecture consists of three main components: a backbone network, a FPN, and two subnetworks dedicated to classification and regression tasks. The backbone network is assigned to taking the features out of the input image. ResNet50, which consists of four feature maps with various resolutions, serves as the backbone network for RetinaNet. By utilizing the FPN, the feature maps are merged together to create a layered structure of feature maps at various scales, facilitating the identification of objects with varying sizes. The regression subnetwork refines the bounding box coordinates of the objects, while the classification subnetwork categorizes the objects into different classes by utilizing the feature maps produced by the FPN. RetinaNet incorporates anchor boxes, which are pre-defined bounding boxes that come in different sizes and aspect ratios, allowing for the detection of objects at various scales and positions. A crucial part of the architecture of RetinaNet is the Focal Loss function. Figure 8 displays the fundamental design of the RetinaNet detector.

3.3.1 Feature pyramid network (FPN)

An essential part of RetinaNet's architecture is the FPN. Through the creation of a feature pyramid with various scales and resolutions, it addresses the problem of objects of varied scales. FPN functions by initially examining the feature maps generated by the underlying network using both a top-down and bottom-up strategy. Once the feature maps from both the lower-level and higher-level pathways are upsampled to a higher resolution, they are merged or fused together. To create a feature pyramid with several tiers, this technique is repeatedly carried out. The model is capable of detecting objects of different sizes because the feature maps at each

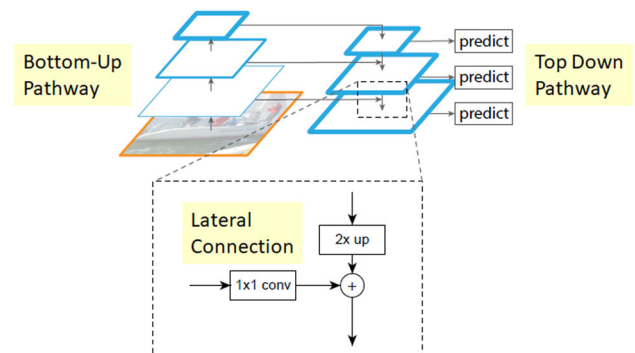


Fig. 9 FPN architecture

level of the pyramid possess diverse scales and resolutions. The higher-level feature maps in the pyramid have finer detail and are specifically designed to identify smaller elements, while the lower-level feature maps have reduced resolution and are optimized for detecting larger objects. Figure 9 displays the FPN framework.

Bottom-Up pathway The input image is initially processed through a convolutional neural network (CNN) to extract features. The CNN typically consists of multiple layers, and as you move deeper into the network, the spatial resolution decreases, while the semantic information increases.

Top-down pathway FPN introduces a top-down pathway to the network. It starts by adding a lateral connection from a higher-resolution feature map to a lower-resolution one. This top-down pathway involves upsampling the spatially coarser feature maps to match the spatial resolution of the higher-level feature maps.

The goal is to reintroduce fine-grained spatial information at higher resolutions.

Combining features At each level of the feature pyramid, the combined features are obtained by adding the upsampled higher-level features to the corresponding lower-level features. This process is repeated iteratively through the pyramid, creating a series of feature maps with both high-level semantic information and fine-grained spatial details.

Feature pyramids The result is a feature pyramid where each level corresponds to a different scale. Higher levels contain more abstract and semantic information, while lower levels retain more spatial details. The feature pyramid enables the network to effectively handle objects of different sizes, as features at multiple scales are available for analysis.

By incorporating the top-down pathway and lateral connections, FPN allows the network to maintain a rich representation of spatial information at different scales. This is crucial for tasks like object detection, where objects may vary in size and appearance, and capturing both global context and fine details is essential for accurate predictions.

3.3.2 Classification subnet

The object classification subnetwork is linked to each level of the feature pyramid through a fully convolutional network for effective feature mapping. Each subnetwork consists of four layers of 3×3 convolutional operations with 256 filters, which are activated using the ReLU function. This is succeeded by another layer of 3×3 convolutional operations with a sigmoid activation function. The dimensions of the output are influenced by the dimensions of the input feature map. The dimensions of the output are scaled in proportion to the original map, whereas the depth of the output is determined by the number of classes and anchor boxes used.

3.3.3 Regression subnet

For every level of the FPN, there is a corresponding bounding box regression subnet. The only difference in structure between the subnetwork and the classification network is the number of filters in the final convolutional layer. In this case, the final convolutional layer has a filter count of 4A, representing four bounding box coordinates for each anchor box.

3.3.4 Focal loss function

The architecture of RetinaNet also includes the Focal Loss function. To address imbalanced data, the strategy involves assigning higher weights to challenging examples, which are instances that the model struggles to accurately detect or classify. The targeted loss function works by reducing the impact of loss on correctly classified examples and increasing the impact on incorrectly classified examples. This is accomplished by incorporating a regulating element called

the focal factor, which reduces the loss assigned to instances that are correctly classified and amplifies the loss attributed to examples that are classified incorrectly. The anticipated probability of the object is a function of the focal factor. The loss is given less importance when the anticipated probability is high but the focal factor is low. When the loss is up-weighted, the focal factor is high, and the anticipated probability is low. The use of this approach is superior to the traditional cross-entropy loss, as it considers the varying difficulty of examples, assigning appropriate weights accordingly. The multi-task loss function used in RetinaNet brings together the classification loss and the regression loss, effectively combining them into a single loss function. The multi-task loss function in RetinaNet is determined by combining two separate losses in a weighted manner [28–31].

Let's denote the ground truth class label for an anchor box as y_i and the predicted class probabilities as p_i . Similarly, the ground truth bounding box regression targets are represented as t_i , and the predicted bounding box regression offsets are v_i . The multi-task loss function in RetinaNet can be defined as follows:

$$\text{Loss} = \text{Classification loss} + \lambda * \text{Regression loss} \quad (1)$$

The weighting factor in the multi-task loss function integrates the classification loss and regression loss to produce a single objective. The parameter, λ , regulates the weighting of the two losses and can be changed to give preference to one over the other. It determines the relative importance of classification and regression during training. The classification loss measures the difference between the predicted class probabilities, p_i and the ground truth class labels y_i . It is commonly computed using the cross-entropy loss. The classification loss for a single anchor box is given by

$$\text{Classification loss} = -y_i * \log(p_i) - (1 - y_i) * \log(1 - p_i) \quad (2)$$

Here, y_i is a binary indicator that equals 1 if the anchor box is positive (contains an object) and 0 otherwise. p_i depicts the predicted probability of the positive class.

The regression loss captures the discrepancy between the predicted bounding box regression offsets y_i and the ground truth bounding box regression targets t_i . The smooth L_1 loss is commonly used for this purpose. The regression loss for a single anchor box is calculated as:

$$\text{Regression loss} = \text{Smooth } L_1(v_i - t_i) \quad (3)$$

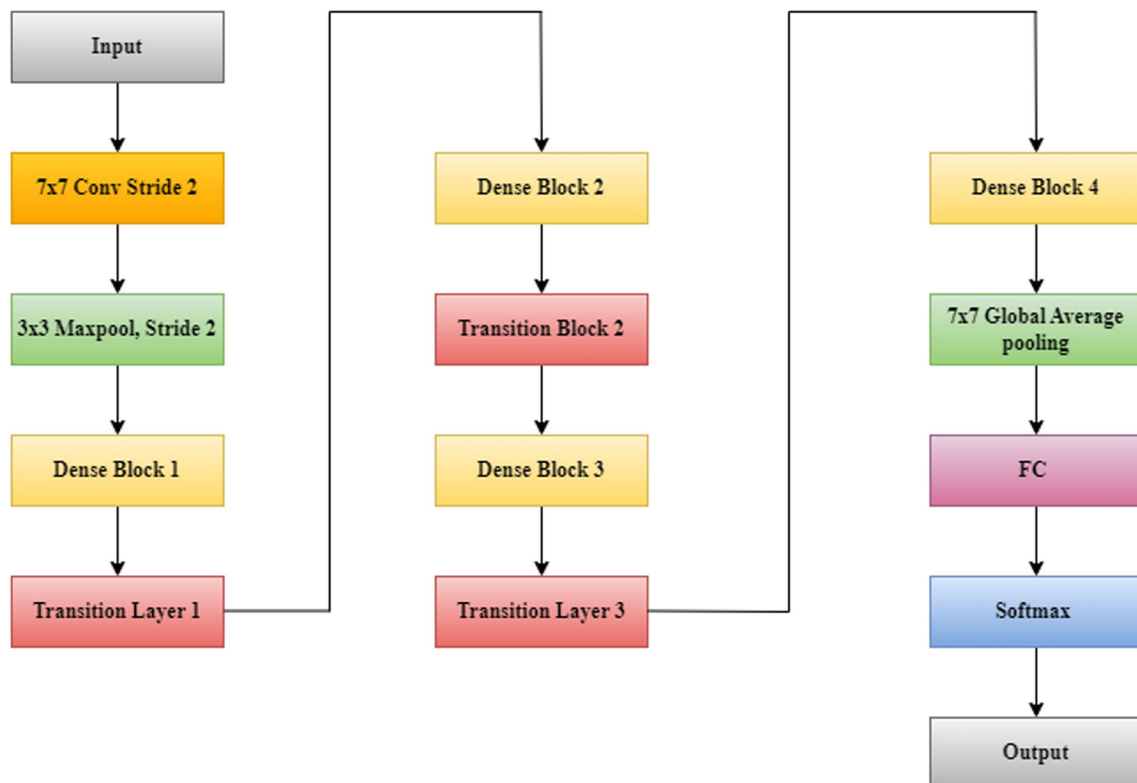


Fig. 10 Basic architecture of DenseNet-121

3.4 Bounding box preprocessor

The initial processing step for bounding boxes involves taking the identified candidate traffic sign boxes and preparing them for classification. This preparation includes extracting relevant information and making any required adjustments to ensure accurate classification [32]. The traffic sign is placed in the region's exact center, which is then increased by 25% to account for any regression mistakes and ensure that the area is completely encircled by the bounding box. In order for the classifier network to evaluate the class of a traffic sign, the enlarged boxes are cropped and adjusted to a uniform size of 48×48 pixels.

The GTSDDB dataset, which contains 600 training images, was used to train the detector. The RetinaNet was loaded with the Resnet50 model, which had been pre-trained on the COCO dataset. The subnets are trained using minibatch training with a batch size of thirty-two over a duration of 100 epochs. The loss function optimizer employed was the Adam algorithm.

3.5 Traffic sign classification module based on DenseNet-121

CNNs, a kind of deep neural networks, are employed to analyze images [33]. CNN has the important advantage of

requiring no human effort or prior knowledge for feature design. A basic CNN is made up of layers, and each layer of a CNN employs a differentiable function to convert one volume of activations to another. The DenseNet-121, a deep CNN architecture, has gained popularity in numerous computer vision tasks due to its effectiveness in image classification [27]. It works particularly well for tasks like classifying traffic signs, where precise sign identification is essential for guaranteeing traffic safety. The family of DenseNets, which are renowned for their densely connected layers and effective feature reuse, includes DenseNet-121. DenseNet adds dense connections, which enable each layer to receive direct inputs from all preceding layers, in contrast to conventional CNN architectures that stack layers one after the other. The network can efficiently learn and reuse features at various depths due to this extensive connectivity, which encourages information flow across layers [34–36]. Each dense block in DenseNet-121 has a number of densely connected convolutional layers. Each dense block has a rich collection of features that are created by concatenating the feature maps from earlier levels, which are then passed on to the following layers. By preventing the vanishing gradient issue, this dense connectivity makes the network easier to train.

A DenseNet is a type of fully CNN that establishes dense connections by directly linking all of its layers together using Dense Blocks. This design creates a dense interconnection

Table 2 Parameter setup

Parameter	Value
The number of dense blocks	3
The depth of the convolutional block	15
Batch size	52
Epoch	52
Growth rate	10

among the layers. Every layer in the network receives additional inputs from all the layer above it and passes its own set of feature mappings to the layer below it in order to maintain the forward flow of information. DenseNet was developed primarily to improve high-level NN accuracy due to the vanishing gradient, which happens when data vanishes after arriving at its destination due to the enormous distance between input and output layers [37]. Each layer uses concatenated feature maps as inputs rather than summarizing the feature maps of all prior levels. As a result, DenseNet uses fewer variables than a standard CNN, enabling feature reuse by tossing out redundant feature maps. Figure 10 depicts the basic framework of the DenseNet-121 model. In addition to the basic convolutional and pooling layers, DenseNet also includes two crucial elements. They consist of dense blocks and transition layers. A simple convolution and pooling layer comprise the first layer of DenseNet. The structure of the model includes a series of dense blocks, with each block being accompanied by a transition layer. This sequence leads to a concluding dense block, followed by a classification layer. The growth rate, denoted by k , is a hyperparameter that determines how many new features maps each layer contributes to the network. If $k = 12$; for example, each layer adds 12 new feature maps to the input it receives. Table 2 shows the parameter setup for DenseNet.

DenseNet-121 consists of 121 layers, including convolutional layers, transition layers, a global average pooling layer, and a FC layer. The majority of the layers in the model are made up of dense blocks, each of which has a particular number of convolutional layers. The feature maps are down sampled using the transition layers, and the spatial dimensions are shrunk utilizing the global average pooling layer. Based on the total number of target classes, the fully connected layer generates the final classification probabilities. A classifier was trained using 39,209 training images from the GTSRB dataset. During the training process, a batch size of 32 was employed for minibatch training. The optimization procedure was facilitated by the Adam optimizer. The classifier underwent 100 epochs of training.

4 Results and discussion

4.1 Hardware and software setup

The proposed method was implemented once the dataset had been collected. Keras with a TensorFlow backend is utilized to implement the traffic sign detector and classifier. The GTSDb dataset is utilized for training and evaluating the detector, while the GTSRB dataset is employed for training and testing the classifier. The suggested detector and classification model are trained and evaluated on the Google Colaboratory platform.

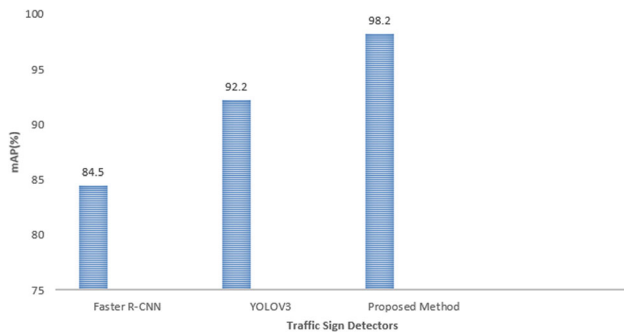
4.2 Performance evaluation

4.2.1 Detector performance

The effectiveness of the suggested traffic sign detector was evaluated by measuring two key factors, mean average precision (mAP) and speed of detection per image. The calculation of the mAP score considers the average precision for each class as well as the intersection over union (IoU) thresholds, which vary depending on the specific detection challenges being addressed. The overlap between the actual and predicted bounding boxes is determined by IoU. It is commonly employed to ascertain whether a detection is an accurate positive finding or an erroneous positive result. The IoU between the predicted and actual bounding boxes is evaluated by dividing the combined area of the two boxes by the area of overlap. A positive proposal is one where the IoU between the anticipated bounding box and the ground truth box is more than 0.5. The following procedures are commonly taken to calculate the mAP for object detection. The first step entails figuring out the IoU for each expected bounding box and its matching ground truth bounding box. The degree of precision in object localization is determined by the IoU value, which quantifies the extent of overlap between the predicted bounding boxes and the actual ground truth boxes. The anticipated bounding boxes are then ranked depending on the confidence levels associated with each one. At various confidence score thresholds, precision and recall values are calculated for each class. A precision–recall curve is produced for each class by altering the threshold. The precision–recall curve is then integrated to determine the average precision for each class. A higher value indicates greater performance for the average precision metric, which has a range of 0 to 1. It measures the effectiveness of a detector in accurately identifying and classifying a specific class. The weighted average of the average precision values over all object classes is then determined, considering the proportion of ground truth instances for each class. By addressing class imbalance, this weighting makes it possible to fairly assess the detector’s effectiveness

Table 3 Comparison of detection performance

Detector	mAP%	Speed of detection per image (milliseconds)
YOLOV3	92.2	101
Faster RCNN	84.5	261
Proposed model	98.2	193

**Fig. 11** Performance comparison of traffic sign detectors

across all classes. An overall evaluation of the object detector's precision, localization, and performance is provided by the generated mAP. It is a widely used statistic in object detection tasks for contrasting and assessing various models. The time it takes an object detection model to process a single image and produce predictions for the items inside it is referred to as the speed of detection per image for object detection. It evaluates the detector's performance and computational effectiveness by examining a single image. The speed of detection per image can be influenced by various factors including model architecture, hardware acceleration, input image size, batch processing, optimization techniques, and framework and implementation. Table 3 compares the performance of the suggested detection approach with YOLOV3 [38] and F-RCNN [39].

The experimental findings revealed that the RetinaNet-based traffic sign detection model surpassed the existing detection model in terms of mAP. The mAP value for the suggested RetinaNet-based traffic sign detection model was 98.2%. Figure 11 illustrates a visual depiction of the performance evaluation, contrasting the proposed detection model with existing detection models. Figure 12 displays a few of the traffic sign detection results.

4.2.2 Classifier performance

Various performance metrics are generally employed to assess the effectiveness of a classification model in correctly identifying instances. Classification performance relies on several key metrics such as accuracy, precision, recall, and

**Fig. 12** Traffic sign detection results**Table 4** Performance metrics

Performance metrics	Equation
Accuracy	$\frac{(TP+FP)}{(TP+FP+TN+FN)}$
Precision	$\frac{(TP)}{(TP+FP)}$
Recall	$\frac{(TP)}{(TP+FN)}$
F1-score	$2 \times \frac{(Precision \times Recall)}{(Precision+Recall)}$

where TP = True positive, TN = True Negative, FP = False positive, FN = False negative

Table 5 Performance of traffic sign classification model

Performance Metrics	Obtained results (%)
Accuracy	98.32
Precision	99.16
Recall	97.92
F1-score	98.53

F1-score, which are crucial in assessing the quality of the classification model's results. Table 4 lists the mathematical formulas for the performance parameters.

Table 5 provides the performance metrics for the proposed traffic sign classification model. Table 3 demonstrates that the suggested classification model produced enhanced classification findings. The accuracy rate of the suggested traffic sign classification model was 98.32%. The efficiency of the suggested traffic sign classification model is depicted in Fig. 13. Figure 14 displays some of the classification outcomes for traffic signs.

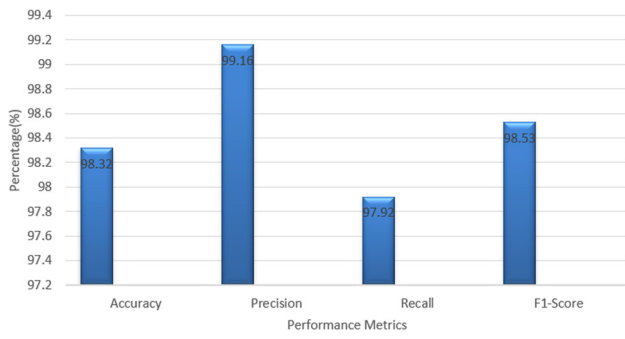


Fig. 13 Performance evaluation of proposed traffic sign classification model



Fig. 14 Traffic sign labeled with classification results

4.2.3 Traffic sign classification module based on DenseNet-121

The training speed and convergence of DenseNet-121, with existing deep learning architecture, depend on various factors such as the dataset, hardware, optimization techniques, and specific hyperparameters used during training. DenseNet-121 is a specific variant of the DenseNet architecture with 121 layers, and it has demonstrated several advantages in terms of training dynamics. It is essential to benchmark the performance of DenseNet-121 against other architectures on a specific task and dataset to draw conclusions about its relative efficiency. Additionally, advancements in hardware (e.g., GPUs and TPUs) and software optimization techniques can influence the training speed of deep learning models, making it necessary to consider the latest tools and technologies when assessing model performance. In Fig. 15, the training and test accuracy with running time of three algorithms are evaluated. Traffic Sign Classification Module Based on DenseNet-121 about 52 epochs, DenseNet-121, achieves 98% the highest training speed when compared with existing algorithm such as Faster RCNN.

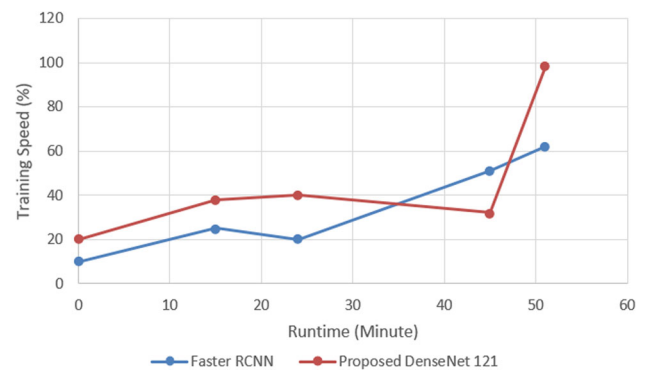


Fig. 15 Training speed with running time for DenseNet-121

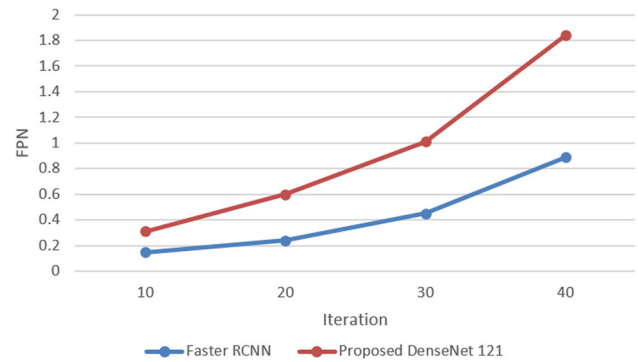


Fig. 16 Convergence speed

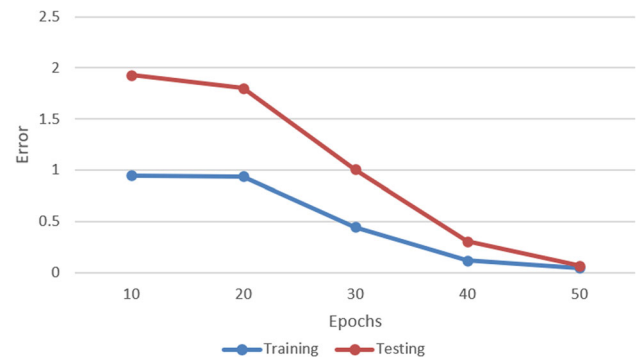


Fig. 17 Focal loss function

Figure 16 illustrates the FPN of various iterations in order to learn the optimal policy of the reality mining dataset. The operating speed of the proposed work is analyzed for 220 iterations. For the FPN process, the convergence speed is analyzed with 5 10 iterations. This shows that our proposed algorithm has a higher convergence speed than the Faster RCNN process. The proposal algorithm’s equivalent state value helps reduce the high dimensionality for learning the process efficiently.

Table 6 Comparison of FPN performance with pre-trained model DenseNet-121

Models	Accuracy	Precision	Recall	F1_score
YOLOV3	92.34	91.41	92.45	94.18
Faster RCNN	95.48	96.23	97.16	88.16
VGG-16	86.26	94.12	93.22	89.15
ResNet-18 Model	91.45	94.25	93.11	80.16
Proposed model	98.12	98.16	97.26	92.54

4.2.4 Focal loss function

In Fig. 17, focal loss is a loss function designed to address the class imbalance problem, particularly in object detection scenarios where there is a large number of background examples compared to positive (foreground) examples. The class imbalance problem arises when the number of instances of one class significantly outweighs the instances of another class, leading the model to be biased toward the majority class. In object detection, most of the image regions are often background, making the problem more pronounced.

4.2.5 Comparison of FPN with pre-trained models

Transfer learning involves leveraging knowledge gained from pre-training a model on one task to improve performance on a different but related task. When integrating pre-trained models with feature pyramid network (FPN) for improved performance, the general approach is to use a pre-trained backbone network and build the FPN on top of it. By integrating a pre-trained backbone with FPN, you can benefit from the generalization capabilities of the pre-trained model while simultaneously leveraging the multi-scale feature representation provided by FPN. This combination is particularly powerful for tasks like object detection, where capturing both high-level semantic information and fine-grained spatial details is essential for accurate predictions. The integration of pre-trained models with FPN helps in achieving better performance on a wide range of computer vision tasks with limited annotated data. According to Table 6, FPN contributes to improving the accuracy of object detection models, especially in scenarios with a large number of background examples, by addressing the class imbalance problem. It allows the model to focus more on challenging instances, leading to better performance in terms of accuracy, precision, recall, and F1-score, particularly for the minority class (foreground) in object detection tasks.

5 Conclusion

Road safety and traffic control are significantly impacted by the creation and implementation of a TSR system. TSR

systems offer several benefits. They improve the driver's understanding of their surroundings by delivering precise and timely details regarding speed limits, traffic rules, and the state of the roads. This promotes safe driving practices and lessens the possibility of accidents brought on by carelessness or disregard for traffic signs. These technologies can also help to reduce infractions and increase adherence to traffic laws, which will make the roads safer for all road users in the long run. The development and use of autonomous vehicles are further supported by TSR technologies. These technologies enable autonomous vehicles to navigate and interact with their environment successfully by correctly recognizing and interpreting traffic signs. By ensuring that vehicles can react appropriately to traffic signs, this technology improves the safety and dependability of autonomous driving, contributing to increased traffic flow efficiency and road safety. The accuracy and dependability of TSR systems are improving due to ongoing developments in computer vision and AI. This paper introduces a deep learning system designed for efficient TSR. The RetinaNet-based traffic sign detection model and the DenseNet-121-based traffic sign classification model constitute the suggested TSR model. The detection and classification models are evaluated using the GTSDDB and GTSRB datasets, each serving their purpose in assessing the models. The results indicated that the suggested TSR system has enhanced its ability to detect and classify traffic signs.

Acknowledgements The author would like to express his heartfelt gratitude to the supervisor for his guidance and unwavering support during this research for his guidance and support.

Author contributions The authors confirm contribution to the paper as follows: study conception and design: Anju J Prakash and Sruthy. S; data collection: Anju J Prakash; analysis and interpretation of results: Sruthy. S; and draft manuscript preparation: Anju J Prakash, and all authors reviewed the results and approved the final version of the manuscript.

Funding Authors did not receive any funding.

Data availability Data sharing is not applicable to this article as no new data were created or analyzed in this research.

Declarations

Conflict of interest Authors do not have any conflicts.

Ethical approval My research guide reviewed and ethically approved this manuscript for publishing in this Journal.

Human and animal rights This article does not contain any studies with human or animal subjects performed by any of the authors.

Informed consent I certify that I have explained the nature and purpose of this study to the above-named individual, and I have discussed the potential benefits of this study participation. The questions the individual had about this study have been answered, and we will always be available to address future questions.

References

- Salmon, P.M., Lenné, M.G., Stanton, N.A., Jenkins, D.P., Walker, G.H.: Managing error on the open road: the contribution of human error models and methods. *Saf. Sci.* **48**(10), 1225–1235 (2010). <https://doi.org/10.1016/j.ssci.2010.04.004>
- Nepelski, M.: The impact of means of transport on the mortality of road users in the European Union. *Nowa Polityka Wschodnia* (2021). <https://doi.org/10.15804/npw20213106>
- Hua, J., Li, L., Ning, P., Schwebel, D.C., He, J., Rao, Z., Hu, G.: Road traffic death coding quality in the WHO Mortality Database. *Bull. World Health Organ.* **101**(10), 637 (2023). <https://doi.org/10.2471/BLT.23.289683>
- Gonzalez, R.C.: *Digital Image Processing*. Pearson Education India, Bengaluru (2009)
- Amit, Y.: *2D Object Detection and Recognition: Models, Algorithms, and Networks*. MIT Press, Cambridge (2002)
- Zhang, J., Huang, M., Jin, X., Li, X.: A real-time Chinese traffic sign detection algorithm based on modified YOLOv2. *Algorithms* **10**(4), 127 (2017). <https://doi.org/10.3390/a10040127>
- Stallkamp, J., Schlipsing, M., Salmen, J., Igel, C.: Man vs. computer: benchmarking machine learning algorithms for traffic sign recognition. *Neural Netw.* **32**, 323–332 (2012). <https://doi.org/10.1016/j.neunet.2012.02.016>
- Mogelmoose, A., Trivedi, M.M., Moeslund, T.B.: Vision-based traffic sign detection and analysis for intelligent driver assistance systems: perspectives and survey. *IEEE Trans. Intell. Transp. Syst.* **13**(4), 1484–1497 (2012). <https://doi.org/10.1109/TITS.2012.2209421>
- Zhu, Z., Liang, D., Zhang, S., Huang, X., Li, B., Hu, S.: Traffic-sign detection and classification in the wild. In: *IEEE conference on computer vision and pattern recognition* (2016).
- Tabernik, D., Skočaj, D.: Deep learning for large-scale traffic-sign detection and recognition. *IEEE Trans. Intell. Transp. Syst.* **21**(4), 1427–1440 (2019). <https://doi.org/10.1109/TITS.2019.2913588>
- Cao, J., Song, C., Peng, S., Xiao, F., Song, S.: Improved traffic sign detection and recognition algorithm for intelligent vehicles. *Sensors* **19**(18), 4021 (2019). <https://doi.org/10.3390/s19184021>
- Sun, Y., Ge, P., Liu, D.: Traffic sign detection and recognition based on convolutional neural network. In *2019 Chinese automation congress (CAC)*. pp 2851–2854 (2019). <https://doi.org/10.1109/CAC48633.2019.8997240>
- Vennelakanti, A., Shreya, S., Rajendran, R., Sarkar, D., Muddegowda, D., Hanagal, P.: Traffic sign detection and recognition using a CNN ensemble. In: *2019 IEEE international conference on consumer electronics (ICCE)*. 1–4 (2019). DOI: <https://doi.org/10.1109/ICCE.2019.8662019>
- William, M. M., Zaki, P. S., Soliman, B. K., Alexsan, K. G., Mansour, M., El-Moursy, M., Khalil, K.: Traffic signs detection and recognition system using deep learning. In: *2019 Ninth international conference on intelligent computing and information systems (ICICIS)* 160–166 (2019). DOI: <https://doi.org/10.1109/ICICIS46948.2019.9014763>
- Kamal, U., Tonmoy, T.I., Das, S., Hasan, M.K.: Automatic traffic sign detection and recognition using SegU-Net and a modified Tversky loss function with L1-constraint. *IEEE Trans. Intell. Transp. Syst.* **21**(4), 1467–1479 (2019). <https://doi.org/10.1109/TITS.2019.2911727>
- Liang, Z., Shao, J., Zhang, D., Gao, L.: Traffic sign detection and recognition based on pyramid convolutional networks. *Neural Comput. Appl.* **32**, 6533–6543 (2020). <https://doi.org/10.1007/s00521-019-04086-z>
- Alghmgham, D.A., Latif, G., Alghazo, J., Alzubaidi, L.: Autonomous traffic sign (ATSR) detection and recognition using deep CNN. *Proc. Comput. Sci.* **163**, 266–274 (2019). <https://doi.org/10.1016/j.procs.2019.12.108>
- Rajendran, S. P., Shine, L., Pradeep, R., Vijayaraghavan, S.: Real-time traffic sign recognition using YOLOv3 based detector. In: *2019 10th international conference on computing, communication and networking technologies (ICCCNT)* 1–7 (2019). <https://doi.org/10.1109/ICCCNT45670.2019.8944890>
- Yuan, Y., Xiong, Z., Wang, Q.: VSSA-NET: vertical spatial sequence attention network for traffic sign detection. *IEEE Trans. Image Process.* **28**(7), 3423–3434 (2019). <https://doi.org/10.1109/TIP.2019.2896952>
- Xu, X., Jin, J., Zhang, S., Zhang, L., Pu, S., Chen, Z.: Smart data driven traffic sign detection method based on adaptive color threshold and shape symmetry. *Future Gener. Comput. Syst.* **94**, 381–391 (2019). <https://doi.org/10.1016/j.future.2018.11.027>
- Ahmed, S., Kamal, U., Hasan, M.K.: DFR-TSD: a deep learning-based framework for robust traffic sign detection under challenging weather conditions. *IEEE Trans. Intell. Transp. Syst.* **23**(6), 5150–5162 (2021). <https://doi.org/10.1109/TITS.2020.3048878>
- Han, C., Gao, G., Zhang, Y.: Real-time small traffic sign detection with revised faster-RCNN. *Multimed. Tools Appl.* **78**, 13263–13278 (2019). <https://doi.org/10.1007/s11042-018-6428-0>
- Liu, Z., Li, D., Ge, S.S., Tian, F.: Small traffic sign detection from large image. *Appl. Intell.* **50**, 1–13 (2020). <https://doi.org/10.1007/s10489-019-01511-7>
- Jin, Y., Fu, Y., Wang, W., Guo, J., Ren, C., Xiang, X.: Multi-feature fusion and enhancement single shot detector for traffic sign recognition. *IEEE Access* **8**, 38931–38940 (2020). <https://doi.org/10.1109/ACCESS.2020.2975828>
- Shi, Y., Ko, Y.C.: Construction of English pronunciation judgment and detection model based on deep learning neural networks data stream fusion. *Int. J. Pattern Recogn. Artif. Intell.* **36**(06), 2252011 (2022). <https://doi.org/10.1142/S0218001422520115>
- Ojha, A., Sahu, S.P., Dewangan, D.K. (2022). VNet: vehicle detection network using computer vision and deep learning mechanism for intelligent vehicle system. In: Noor, A., Sen, A., Trivedi, G. (eds) *Proceedings of Emerging Trends and Technologies on Intelligent Systems and Computing*. ETTIS 2021. Advances in Intelligent Systems and Computing, vol 1371. Springer, Singapore. https://doi.org/10.1007/978-981-16-3097-2_9
- Dewangan, D.K., Sahu, S.P.: Lane detection for intelligent vehicle system using image processing techniques. In: Verma, G.K., Soni, B., Bourennane, S., Ramos, A.C.B. (eds.) *Data Science. Transactions on Computer Systems and Networks*. Springer, Singapore (2021)
- Huang, G., Liu, Z., Pleiss, G., Van Der Maaten, L., Weinberger, K.Q.: Convolutional networks with dense connectivity. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**(12), 8704–8716 (2019). <https://doi.org/10.1109/TPAMI.2019.2918284>
- Vallabhajosyula, S., Sistla, V., Kolli, V.K.K.: Transfer learning-based deep ensemble neural network for plant leaf disease detection. *J. Plant Dis. Prot.* (2021). <https://doi.org/10.1007/s41348-021-00465-8>

30. Redmon, J., Farhadi, A.: Yolov3: an incremental improvement. arXiv preprint [arXiv:1804.02767](https://arxiv.org/abs/1804.02767) (2018). <https://doi.org/10.48550/arXiv.1804.02767>
31. Li, J., Wang, Z.: Real-time traffic sign recognition based on efficient CNNs in the wild. *IEEE Trans. Intell. Transp. Syst.* **20**(3), 975–984 (2018). <https://doi.org/10.1109/TITS.2018.2843815>
32. Appathurai, A., Manogaran, G., Chilamkurti, N.: Trusted FPGA-based transport traffic injects, impersonate (I2) attacks beaconing in the Internet of Vehicles. *IET Netw.* **8**(3), 169–178 (2019)
33. Dhipa, M., Anitha, D.: Detection of violence in football stadium through big data framework and deep learning approach. *Int. J. Data Sci. Artif. Intell.* **01**(02), 21–31 (2023)
34. Prabhu, M., Revathy, G., Raja Kumar, R.: Deep learning based authentication secure data storing in cloud computing. *Int. J. Comput. Eng. Optim.* **01**(01), 10–14 (2023)
35. Josephin Shermila, P., Ahilan, A., Jasmine Gnana Malar, A., Jothin, R. MDEEPFIC: Food item classification with calorie calculation using modified dragon fly deep learning network. *J. Intell. Fuzzy Syst.* pp 1–12
36. Kumar, B.M., Guduru, R.K.R., Srinivas, A., Ana, F., Ramudu, K., Dhiman, G.: Wavelength assignment in optical fiber with intelligent optimization and assignment scheme for static and dynamic traffic intensity based Photonic networks. *Opt. Quant. Electron.* **54**(8), 526 (2022)
37. Dewangan, D.K., Sahu, S.P.: Towards the design of vision-based intelligent vehicle system: methodologies and challenges. *Evol. Intel.* **16**, 759–800 (2023). <https://doi.org/10.1007/s12065-022-00713-2>
38. https://www.kaggle.com/datasets/safabouguezzi/german-traffic-sign-detectionbench_mark-gtsdb
39. https://www.kaggle.com/datasets/meowmeowmeowmeowmeow/gtsrb-german-traffic_sign.
40. Vincent, M.A., Vidya, K.R., Mathew, S.P.: Traffic sign classification using deep neural network. In: 2020 IEEE Recent Advances in Intelligent Computational Systems (RAICS), 13–17 (2020). IEEE. <https://doi.org/10.1109/RAICSS51191.2020.9332474>
41. Haque, W.A., Arefin, S., Shihavuddin, A.S.M., Hasan, M.A.: DeepThin: A novel lightweight CNN architecture for traffic sign recognition without GPU requirements. *Expert Syst. Appl.* **168**, 114481 (2021). <https://doi.org/10.1016/j.eswa.2020.114481>
42. Fang, W., Zhuo, W., Yan, J., Song, Y., Jiang, D., Zhou, T.: Attention meets long short-term memory: A deep learning network for traffic flow forecasting. *Phys. A Stat. Mech. Appl.* **587**, 126485 (2022). <https://doi.org/10.1016/j.physa.2021.126485>

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.