Traffic Sign Detection and Recognition using Deep Learning

Shaik Shameem [1] and **S. Hanis**[2]

[1]Software Engineer, Mindsprint, Chennai.
Email: shaik.shameem021@gmail.com

[2]Department of Electronics and Communication Engineering,
Sri Sivasubramaniya Nadar College of Engineering,
Chennai.
Email: haniss@ssn.edu.in

ABSTRACT

The development of intelligent vehicles, which directly impacts the adoption of driving behaviours, depends on the detection and recognition of traffic signs. Accurately identifying traffic signs has become increasingly challenging under normal environmental circumstances because of the varying light, complex backgrounds, and ageing of the signs. Creating a Deep Learning algorithm to recognize traffic signals and highlight the signs is the primary goal of this paper. Here, we used an image annotation tool to separate the traffic signs from the images before putting them through the modified efficient net training algorithm to create the model files. The traffic or road sign is identified after the training process, and the sign's name is displayed. As a result, this work effectively guides autonomous vehicles, highlights traffic signs, and predicts traffic signals.

Keywords: Traffic sign detection, Traffic sign classification, Efficient Net algorithm, Deep learning

Mathematics Subject Classification: 11Y16

Computing Classification System: I.2.10

1. INTRODUCTION

Road and traffic sign recognition is a topic of study that can help with the creation of an inventory system (for which real-time recognition is not necessary) or a system for in-car advice (for which real-time recognition is required). Since traffic signs [11] are involved, both road sign inventory and road sign recognition use automatic detection and recognition. Traffic sign recognition [1] is a new technology that allows a car to perceive signs like "speed limit" or "turn ahead" that have been placed on the road. Many contemporary cars and trucks have front-facing cameras that can be used to analyse traffic signs. By giving orders, warnings, and occasionally by seizing control of the car itself, they compel the driver to obey. A convolutional neural network [6, 12, 13,14,15] (CNN, or Conv Net) is a particular kind of multi-layer neural network created to recognize visual patterns directly from pixel images with little to no preprocessing. For use in the development of visual object recognition software, the ImageNet is a sizable visual database. The ImageNet holds an annual software competition called the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), in which computer programs compete to correctly identify and classify objects.

The following are, in brief, the main contributions of this paper: We trained the images by modifying a deep learning algorithm called efficient Net, which produced the output model file. In this work, a convolutional neural network architecture called Efficient Net is used, along with a scaling technique that uses a compound coefficient to scale all depth, width, and resolution dimensions uniformly. Convolutional head, Batch Normalization, Dropout, and fully connected layers receive output weights from efficient net models during the training process. The developed algorithm performs well in both day and night conditions, with an average accuracy of roughly 98%. Additionally, the suggested algorithm's model size is significantly smaller than that of the current systems, making it suitable for usage in systems with limited resources.

The rest of the article is structured as follows: The related works are introduced in Section II, the proposed detection method is described in Section III, our experimental findings are examined in Section IV, and the conclusions are presented in Section V.

2. RELATED WORKS

This section introduces a few representative works of deep convolutional neural network-based traffic sign detection techniques. One of the key research areas is now traffic sign detection [7-9] and recognition. The specifics of the approaches used show that there are numerous alternatives and many ideas as to how better solutions, better robustness, or a better classification rate [2-3] can be achieved from that point onwards. An improved light-weight traffic sign recognition algorithm based on YOLOv4-Tiny [4] was proposed to address the issues of low detection accuracy and inaccurate positioning accuracy of light-weight networks [19] in the task of recognizing traffic signs. The traffic sign data set's [11, 16] anchor is generated using an improved version of the K-means clustering algorithm to increase target positioning precision and detection recall rate. The large-scale feature map optimization approach is suggested, which strengthens the representation of the small target's feature information, increases the detection precision of the long-range small target, and enriches the feature level of the network by using the low-level information. The paper proposed an improved NMS (non- maximum suppression) algorithm to screen the prediction box, avoid deleting the prediction results of various targets, and further improve the detection accuracy and recall rate of the target considering the issue of missed detection of high overlapping targets in the post-processing stage of the model. According to experimental findings, the mean average precision and recall of the improved algorithm in the traffic sign recognition task based on the TT100K dataset are improved by 5.73% and 7.29%, respectively, and the FPS (frame per second) value is maintained at about 87f/s, which satisfies the accuracy and real-time requirements of the task. An enhanced NMS algorithm was used to screen the prediction box, prevent deleting the predictions of various targets, and increase the detection accuracy. However, the accuracy is hampered by the low IoU (Intersection over union) value.

In metropolitan transportation networks, traffic signal control [5] under oversaturated conditions poses a significant challenge. Back-pressure techniques have been shown to maximize network throughput and ensure network stability in earlier works. But most of these techniques are used in an adaptive manner. Fixed phase sequences are still frequently used in traffic signal control systems currently. A novel back-pressure-based signal optimization technique that combines fixed phase sequences with spatial model predictive control has been put forth by the authors here in [10]. To analyse the movement of vehicles between a central intersection and four Peri-urban intersections, a spatial prediction model for traffic flow was first built. After that, a multi-objective traffic signal timing optimization model was created with the aim of lowering the risk of spillover and balancing the distribution of vehicles throughout the entire network. The Pareto frontier and optimal solution were then reached using a technique based on the Multi-Objective Particle Swarm Optimization algorithm and Technique for Order Preference by Similarity to an Ideal Solution principle. Finally, Paramics traffic simulations were run to evaluate how well the suggested approach worked. The simulation results indicate that the above-mentioned method performs better than cycle-based back-pressure schemes and fixed-time control under oversaturated conditions. The Pareto frontier and the ideal solution were reached using the algorithm and technique for order preference by similarity to an ideal solution. The traffic sign detection was also performed using attention network [17] and visual odometry [18] which consumes more time. Consequently, additional research is necessary for real-time implementation.

3. PROPOSED WORK

This section introduces the convolutional neural network (CNN) first, followed by techniques for image annotation and an effective net algorithm. It then talks about network training before introducing the specifics of the implementation. Figure 1 displays the proposed model's overall architecture for detecting traffic signs.

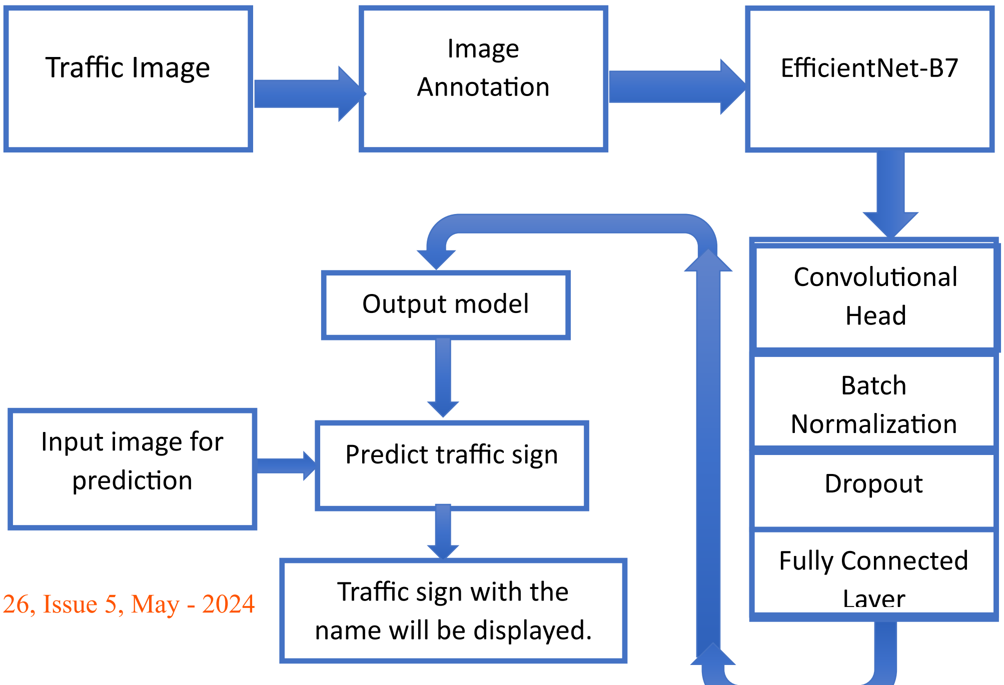


Figure 1. Overall Architecture of the Proposed Method

3.1. Convolutional Neural Networks

The efficient net algorithm follows the CNN architecture. A multi-layered neural network called CNN uses a special architecture to extract increasingly complex features from the data at each layer to produce the desired result. For tasks requiring perception, CNNs are excellent. Most of the time, CNN is utilized when attempting to extract information from an unstructured data set such as an image database. A description of CNN is provided in Figure 2 below.

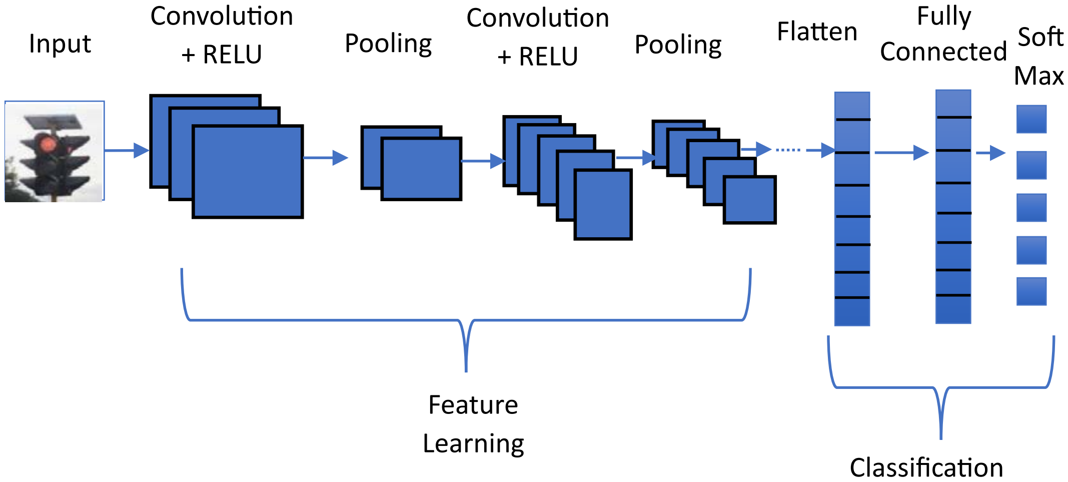


Figure 2. CNN Architecture

3.2. Image Annotation

The process of adding labels to an image is known as image annotation, and it typically involves human labour with some computer assistance. A machine learning engineer predetermines labels and selects them to provide the computer vision model with details about what is shown in the image. Labelling images also enables machine learning specialists to focus on key elements that affect the general precision and accuracy of their model. Possible naming and categorization issues, how to represent obscured objects, how to handle unrecognizable portions of the image are a few examples of considerations.

Classification, object detection, and image segmentation are the various types of image annotation. The objective of whole-image classification is to simply identify the presence of various objects and other properties in an image. You can take it a step further with image object detection and determine the location (bounding boxes) of specific objects. The purpose of image segmentation is to identify and comprehend the pixel-level content of the image. As opposed to object detection, where the bounding boxes of objects can overlap, every pixel in an image belongs to at least one class.

The first step in the annotation process of an image is to identify and provide annotators with instructions. Since every company will have different requirements, annotation professionals need to be well-versed in the specifications and guidelines of each annotation project. Following their training in data annotation, the annotators will start annotating thousands or even hundreds of images on a training data platform specifically for image annotation. Software with multiple tools to outline complex shapes for image annotation is called a training data platform. It is made to have all the tools required for the desired type of annotation.

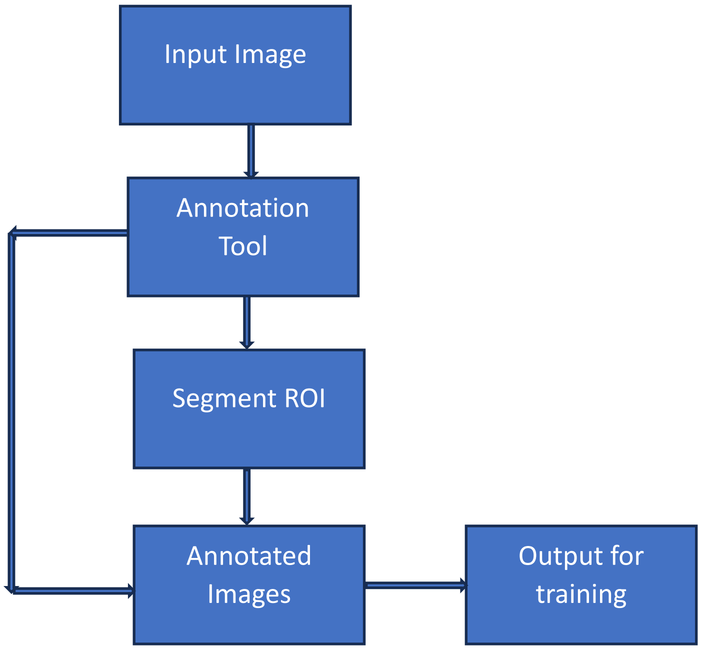


Figure 3. Image Annotation

3.3. Efficient Net Algorithm

Efficient Net is a convolutional neural network architecture and scaling method that uniformly scales depth, width, and resolution using a compound coefficient. Unlike conventional practice, which scales these factors arbitrarily, the Efficient Net scaling approach uniformly scales network width, depth, and resolution using a set of fixed scaling coefficients.

The scalability of models is significantly influenced by the baseline network. Therefore, to further improve performance, the resulting architecture uses mobile inverted bottleneck convolution (MBConv), which is similar to MobileNetV2 and MnasNet but slightly larger due to an increased FLOP budget. After that, the baseline network is expanded to create the Efficient Nets model family. In general, the Efficient Net models perform more accurately and efficiently than the current CNNs because they drastically reduce FLOPS and parameter size. Although Efficient Nets perform admirably on ImageNet, in order to be truly valuable, they should also perform well on other datasets.

The compound scaling technique can be applied to modern CNN architectures such as ResNet and Mobile Net. Choosing a robust baseline network is crucial to achieving the best results because the compound scaling approach only increases the networks' predictive capacity by replicating the convolutional operations and network architecture of the base network. The efficient net baseline model is shown below in Figure 4.

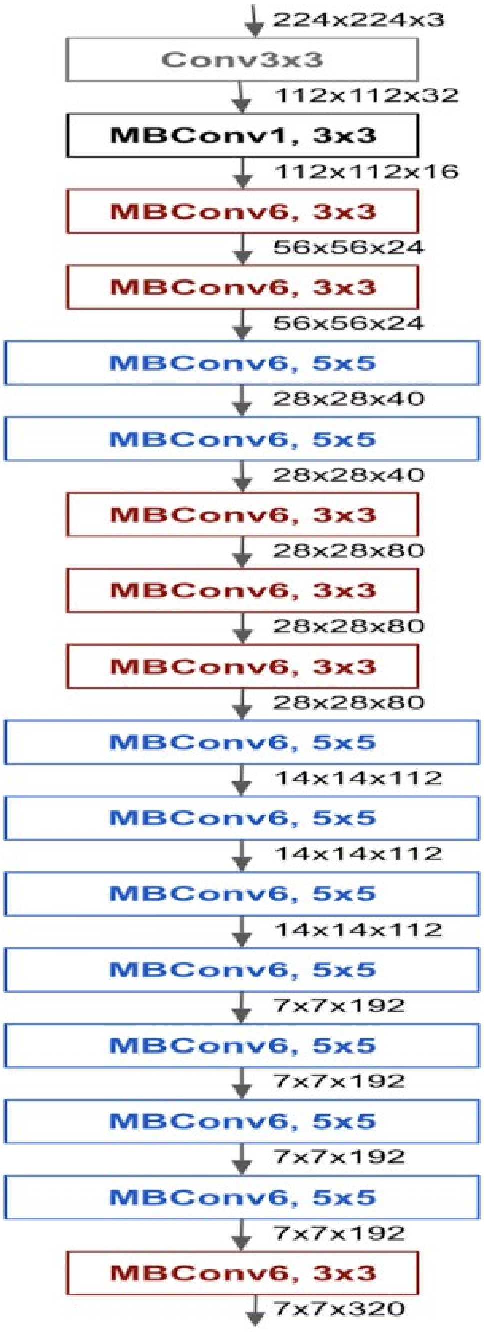


Figure 4. Efficient net baseline model

The output weights of the efficient net model are fed into the convolutional head, batch normalization, dropout, and fully connected layers during training, as shown in Figure 1. We propose a four-layer Efficient Net model in our framework. Four layers have been added to improve performance: convolutional, batch normalisation, dropout, and fully connected. The convolutional layers of a deep CNN are where the filters are applied to the source image or other feature maps. This is where most user-specified parameters for the network are found. The two factors that matter most are the size and quantity of kernels. A layer called batch normalisation enables more independent learning at each layer of the network. It's employed to standardise the output from the earlier layers. It is an algorithmic technique that speeds up and stabilises Deep Neural Network (DNN) training. A technique known as dropout is used to prevent overfitting in a model. The Dropout layer randomly sets the input units to 0 at a rate per step during training. Fully connected layers in a neural network are those in which each activation unit is linked to each input layer. Most deep learning models have a few fully connected layers that combine the data extracted by earlier layers to produce the final output. Once the training process is finished and the input image is supplied for prediction, the traffic or road sign will be recognised, and its name will be displayed. Consequently, the traffic signs have been successfully located and highlighted by the proposed work.

4. EXPERIMENTAL RESULTS

This section first goes over how the dataset was gathered and divided into training and testing models. It also goes over how to apply the data augmentation method to increase the dataset's diversity without gathering new data and how to resize images using preprocessing techniques. Rob flow is then used to annotate these datasets. Ultimately, the task's implementation produced the useful outcomes.

4.1. Data set and Data Augmentation

The dataset was gathered for the proposed project, and figure 5 below displays a sample of the data that was gathered.

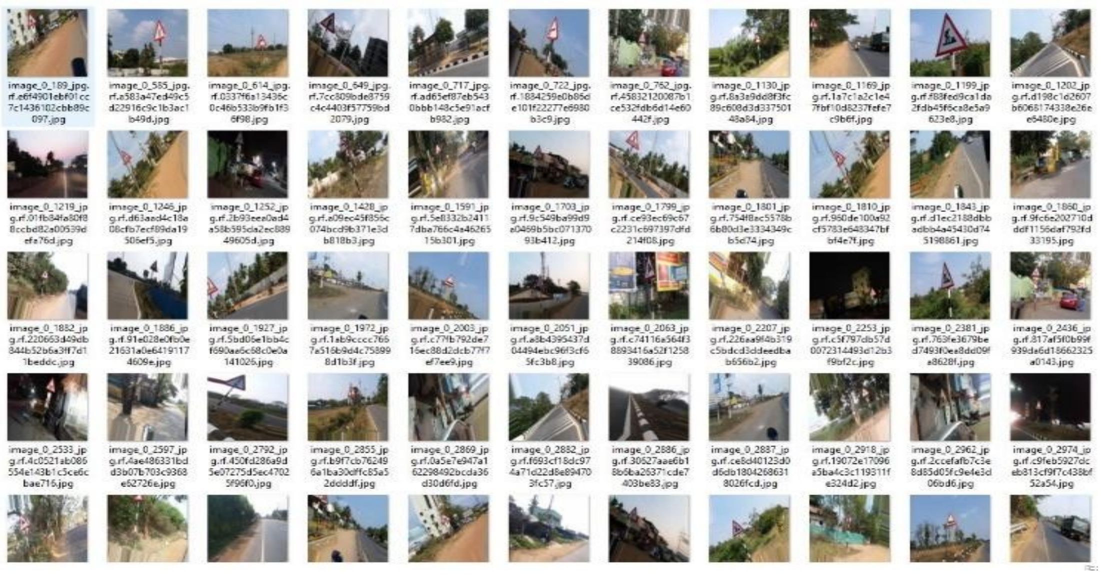


Figure 5. Dataset

The process of data augmentation enhances the data set even more. Data augmentation is the process of slightly changing pre-existing data to increase its diversity without collecting new data. This technique is used to expand a dataset's size. Common techniques for data augmentation include rotation, cropping, shearing, and flipping the data both horizontally and vertically. One can prevent a neural network from learning irrelevant features by augmenting the data. As a result, the model performs better.

To improve our data, we have employed the "image data generator" class in Keras. This is due to the fact that it offers a rapid and simple method of image augmentation. After that, these datasets are expanded to make them larger, as shown in Figure 6 below:

Figure 6. Dataset augmentation

Three pre-processing methods are applied when resizing images. Resizing an image to the desired size (height, width) is known as simple preprocessing. All of the images in the gathered dataset have various sizes (height, width). All images in deep learning should have the same dimensions. Aspect-Aware Preprocessing technique is comparable to basic preprocessing, but it preserves an image's aspect ratio during optimisation. Image to Array Preprocessing: Images in the [Height, Width, Channel] format, are used by Python in deep learning. Faster convergence can be achieved by normalising the image array so that its values are scaled down between 0 and 1 from 0 to 255 for a similar data distribution. The following figure 7 shows the outcomes of various pre-processing methods.

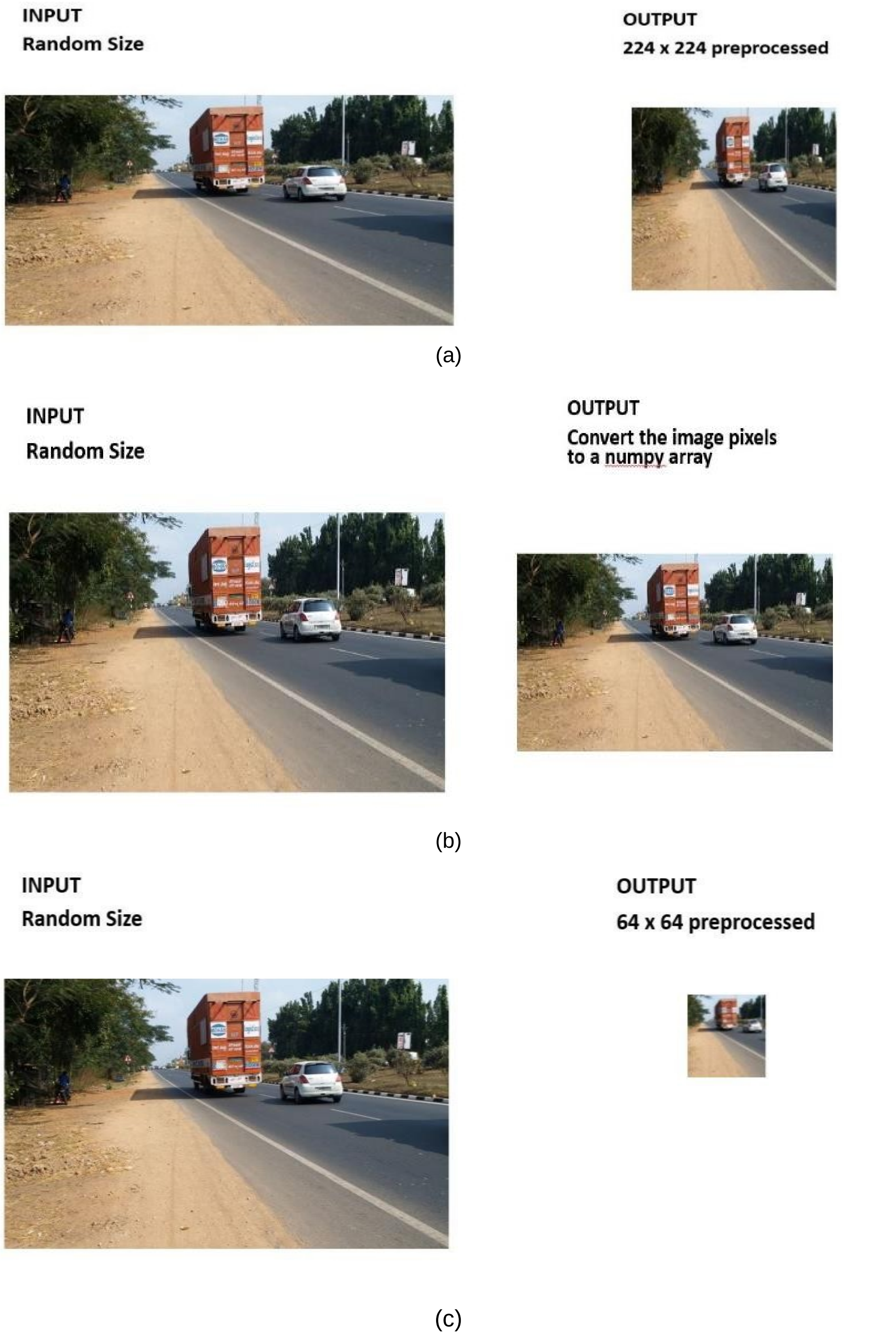


Figure 7. (a) simple preprocessing, (b) Image to Array Pre-processing, (c) Aspect Aware Pre-processing.

During training, one of the most popular plots for debugging neural networks is the "Loss curve." It provides us with a quick overview of the training procedure and the learning trajectory of the network. Every data item in an epoch is used to calculate the loss function, which is guaranteed to yield the quantitative loss measure at that epoch. Throughout the training epoch, our model's loss is getting smaller. Every epoch sees a decrease in loss as shown in figure 8. Our model's efficiency is demonstrated by the loss gradually decreasing.

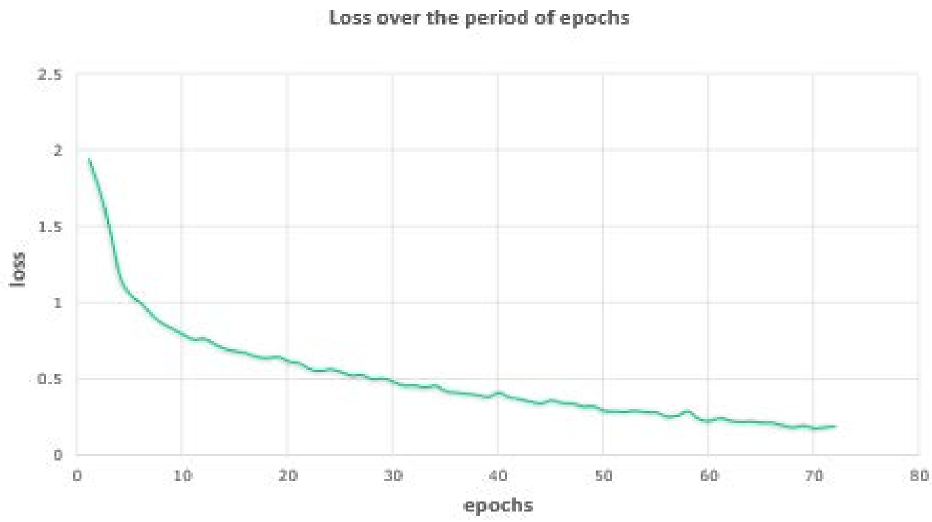
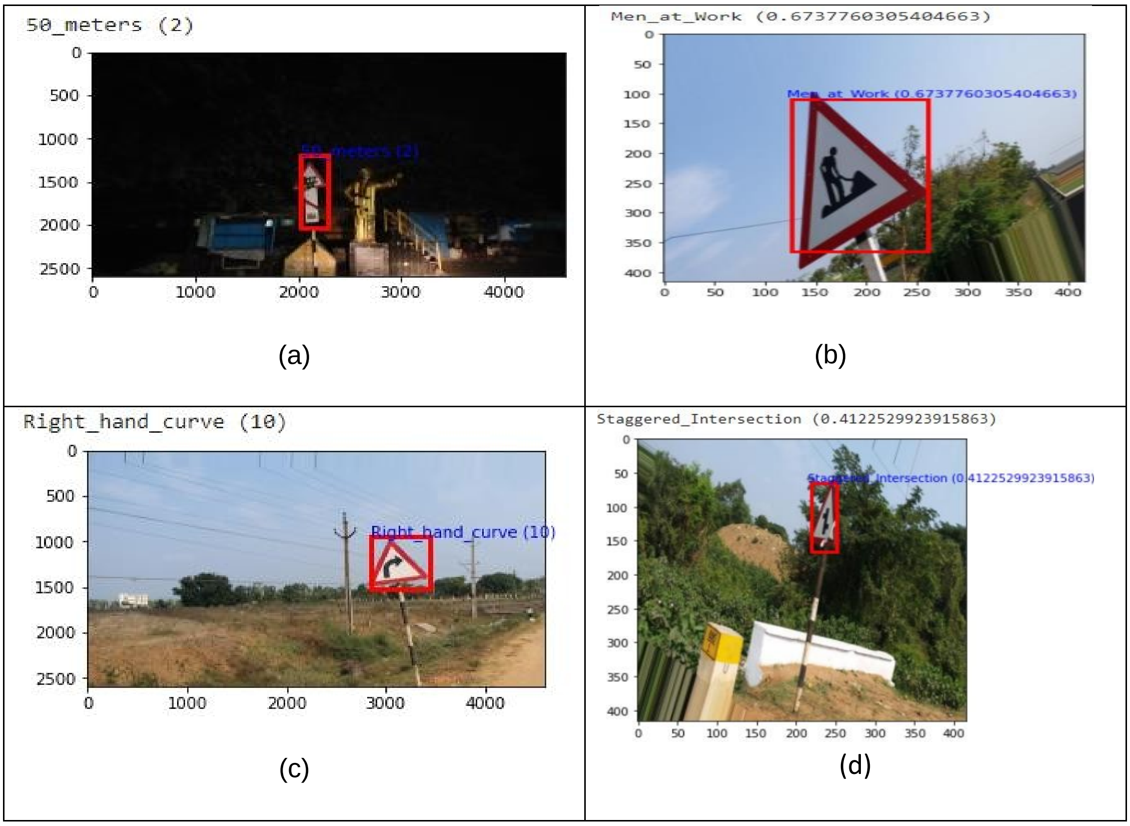


Figure 8. Training loss curve

In Figure 9, traffic sign detection samples produced by the efficient net model are displayed.



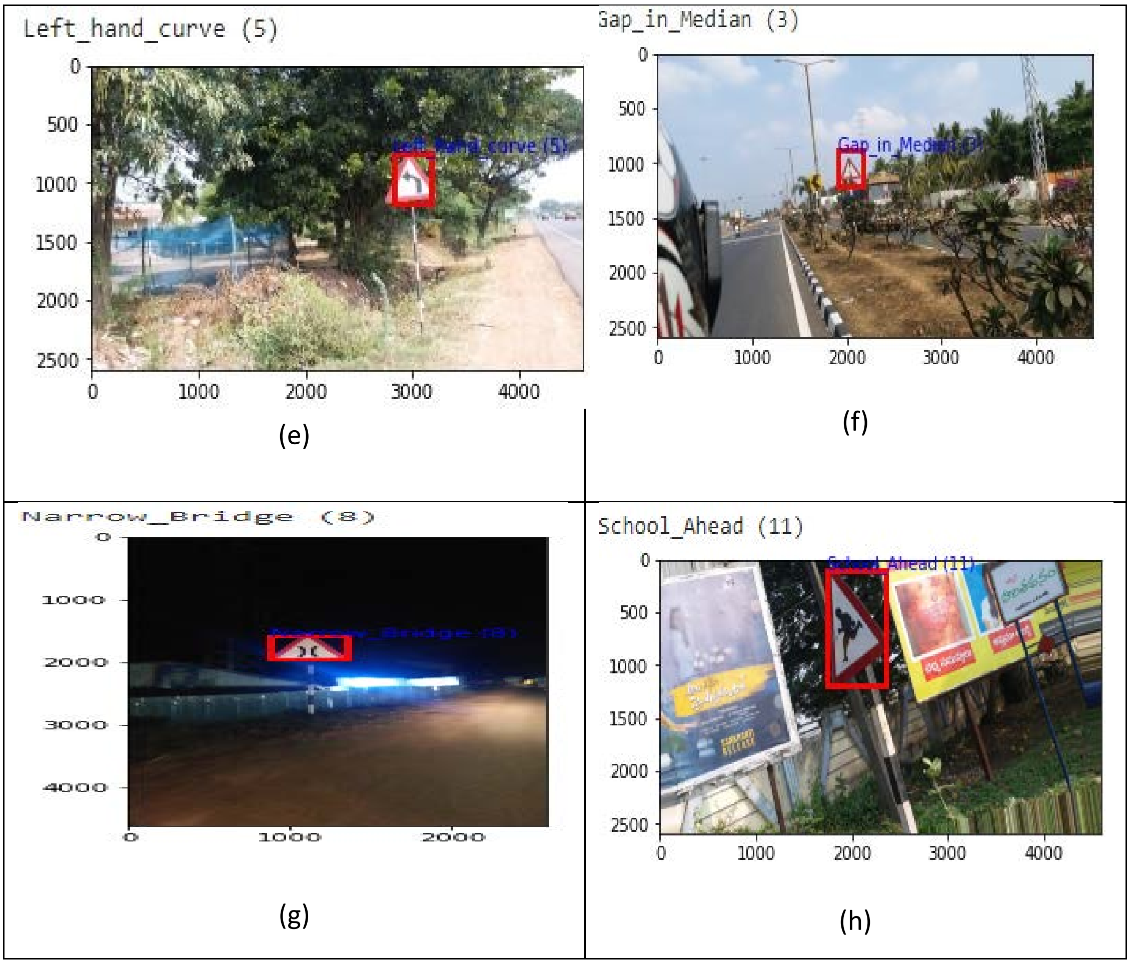


Figure 9. (a) 50_meters, (b) men_at_work (c) straggered_intersection, (d)right_hand_curve. (e) left_hand_curve, (f) gap_in_median (g) Narrow bridge (h) School ahead

5. PERFORMANCE ANALYSIS

The proposed algorithm's test time, intersection of union, and accuracy are assessed. The efficiency of the algorithm is then illustrated by comparing the developed model with the existing approaches. In the end, the algorithm is implemented on a Raspberry Pi, and its performance is evaluated as well.

The ratio of correctly classified images to the total image can be used to calculate the accuracy. The accuracy is calculated using equation 1.

$$Accuracy = \frac{Truly\ identified\ images}{Total\ image} \quad (1)$$

A performance metric called Intersection over Union (IOU) is used to assess how accurate object detection, segmentation, and annotation algorithms are. It measures the amount of overlap between a dataset's ground truth bounding box or annotated region and the predicted bounding box or segmented region. The IOU is calculated using equation 2.

$$IOU = \frac{Area\ of\ Intersection}{Area\ of\ Union} \tag{2}$$

Table 1: Performance analysis of the proposed model with normal and random images

Normal Images				
Images captured	Number of images	IOU	Accuracy %	Test Time (sec)
Day Time	199	0.9	97.4	2.1
Nighttime	160	0.9	98.75	2.1
Random Images				
Day Time	103	0.9	99.02	2.1
Nighttime	70	0.9	98.57	2.1

The performance analysis of the suggested system using random and normal images is displayed in Table 1 above. The average accuracy rate for identifying the correct traffic signal is 98.44%. and the time required to perform the test is approximately 2.1 seconds.

Algorithm	Day time accuracy (%)	Nighttime accuracy (%)	Training time (sec)	Test Time (sec)	IOU	Model size/MB
Faster RCNN	67.8	65.2	1hr 10min	1.6s	0.5	10.2
CNN with adaptive classifier	96.7	96.7	1hr30min	2.5s	0.5	18.9
Efficient Net	62.3	60.2	1hr12min	1.9s	0.9	20.9
Proposed Model	98.2	98.1	1hr25 min	2.1s	0.9	16.9

Table 2: Comparison of the proposed model with that of the existing techniques

The performance analysis of the suggested model using various current algorithms is displayed in Table 2 above. To test each algorithm, we employed the ICTC dataset. Our algorithm is appropriate for real-time deployment because it has a high proposed model accuracy, an improved IOU value, and a smaller model size than other current systems.

6. CONCLUSION

The proposed approach of predicting traffic signals and emphasising traffic signs has been effectively put into practice. Compared to other methods currently in use, the designed algorithm is the best because it guarantees accuracy in the output produced, has the best speculation, and can be relied upon to identify more conventional traffic signs. The model's smaller size than existing approaches makes it suitable for use in Internet of Things based systems. The technology still has a lot of room for improvement, and autonomous cars can identify traffic signs faster.

7. REFERENCES

[1] S.G. Gámez and Y. Ruichek, “Traffic Signs Detection and Classification for European Urban Environments”, *IEEE Trans. Intell. Transp. Syst*., vol. 21, (2020), pp.4388-4399.

[2] S. Yassmina and A. Behloul, “An overview of traffic sign detection and classification methods”, *Int. J. Multimed. Info. Retr*., vol.6, (2017), pp.193-210.

[3] F. Zaklouta and B. Stanciulescu, “Real-time traffic sign recognition in three stages”, *Robot. Auto. Syst.* Vol. 62, (2014), pp.16–24.

[4] L. Wang, K. Zhou, A. Chu, G. Wang and L. Wang, “An Improved Light-Weight Traffic Sign Recognition Algorithm Based on YOLOv4-Tiny”, *IEEE Access*, vol. **9**, (2021), pp.124963-124971.

[5] C. B. Rafter, B. Anvari, S. Box and T. Cherrett, “Augmenting Traffic Signal Control Systems for Urban Road Networks with Connected Vehicles," *IEEE Trans. Intell. Transp. Syst*, vol. 21, (2020), pp.1728-1740.

[6] H.H. Aghdam, E.J. Heravi and D. Puig, “A practical and highly optimized convolutional neural network for classifying traffic signs in real-time”, *Int. J. Comput. Vis*., vol. 122**,** no.2, (2017), pp. 246–269.

[7] Z. Li Chen, Q.M.J. Wu, and C. Liu, “Deep Saliency with Channel-Wise Hierarchical Feature Responses for Traffic Sign Detection”, *IEEE Trans. Intell. Transp. Syst*, vol. 20, (2019), pp.2497-2509.

[8] D. Temel, M. -H. Chen and G. AlRegib, “Traffic Sign Detection under Challenging Conditions: A Deeper Look into Performance Variations and Spectral Characteristics”, *IEEE Trans. Intell. Transp. Syst*. vol. 21, (2020), pp. 3663-3673.

[9] D. Tabernik and D. Skočaj, “Deep Learning for Large-Scale Traffic-Sign Detection and Recognition”, *IEEE Trans. Intell. Transp. Syst.* vol. 21, (2020), pp. 1427-1440.

[10] D. Ma, J. Xiao, X. Song, X. Ma, S. Jin, “A Back-Pressure-Based Model with Fixed Phase Sequences for Traffic Signal Optimization Under Oversaturated Networks”, *IEEE Trans. Intell. Transp. Syst.* vol. 22, (2020), pp. 5577-5588.

[11] F. Almutairy, T. Alshaabi, J. Nelson, S. Wshah, “ARTS: Automotive Repository of Traffic Signs for the United States”, *IEEE Trans. Intell. Transp. Syst. vol.* 22, (2021), pp. 457-465.

[12] H.S. Lee, K. Kim, “Simultaneous Traffic Sign Detection and Boundary Estimation Using Convolutional Neural Network”, *IEEE Trans. Intell. Transp. Syst*, vol. 19, (2018), pp.1652-1663.

[13] K. M. Rajesh, K. Thanigundala, S.R. Musani, H. Nidamanuru and L. Gadde, “Indian traffic sign detection and recognition using deep learning”, *Int. J. Transp. Sci. Technol*., vol. 12, (2023), pp. 683-699.

[14] J. Li, and Z. Wang, “Real-Time Traffic Sign Recognition Based on Efficient CNNs in the Wild”, *IEEE Trans. Intell. Transp. Syst*, vol. 20, (2019), pp. 975-984.

[15] K. Muhammad, A. Ullah, J. Lloret, J.D. Ser and V.H.C.de Albuquerque, “Deep Learning for Safe Autonomous Driving: Current Challenges and Future Directions”, *IEEE Trans. Intell. Transp. Syst*, vol. 22, (2021), pp. 4316-4336.

[16] M. Akallouch, K.S. Boujemaa, A. Bouhoute, K. Fardousse and L. Berrada, “A Dataset for Arabic-Latin Scene Text Localization in Highway Traffic Panels”, *IEEE Trans. Intell. Transp. Syst*, vol. 9, (2021), pp. 93937-93947.

[17] Y. Tian, J. Gelernter, X. Wang, J. Li and Y. Yu, “Traffic Sign Detection Using a Multi-Scale Recurrent Attention Network”, *IEEE Trans. Intell. Transp. Syst*, vol. 20, (2019), pp. 4466-4475.

[18] Y. Zhang, H. Zhang, G. Wang, J. Yang and J. Hwang, “Bundle Adjustment for Monocular Visual Odometry Based on Detections of Traffic Signs”, *IEEE Trans. Intell. Transp. Syst*, vol. 69, (2020), pp. 151-162.

[19] Z. Wang, J. Wang, Y. Li and S. Wang, “Traffic Sign Recognition with Lightweight Two-Stage Model in Complex Scenes”, *IEEE Trans. Intell. Transp. Syst*, vol. 23, (2020), pp. 1-11.