Proceedings of the 2023 International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/30/20230062

Indian traffic sign detection and recognition using deep learning

T. Kumaravel[1], Sathishkumar V. E.[2,4], P. Natesan[1], S. Dharanesh[1], G. Hariharan[1] and N. Krishnamoorthy[3]

[1] Department of Computer Science andEngineering, Kongu Engineering College, Erode, Tamil Nadu, India
[2]Department of Software Engineering, Jeonbuk National University, Jeonju-si, Republic of Kroea
[3]Department of Software Systems and Engineering, Vellore Institute of Technology, Vellore campus, Vellore, Tamil Nadu, India

[4]sathish@jbnu.ac.kr

Abstract. Traffic signs are a fundamental piece of transportation infrastructure and play a vital role in regulating traffic flow, enforcing proper driving behavior, and reducing the risk of accidents, injuries, and fatalities. An Intelligent Transportation System (ITS) must have the ability to automatically detect the sign and then recognise traffic signs which is to be effective. Automatic traffic sign detection is necessary. and is growing in significance with the advent of self-driving cars. This study introduces a brand-new deep learning-based method for identifying traffic signs in India. The proposed system utilizes a region-based convolutional neural network (CNN) to achieve automatic identification and recognition of traffic signs. The authors describe various architectural and data augmentation enhancements to the CNN model and take into account unique and challenging Indian traffic sign types that have not been previously discussed in literature. The system is trained and evaluated using a database of real-time images captured on Indian highways. The deep learning approach is utilized to work on the accuracy and precision of the system, determined to make automated driving automobiles.

Keywords. convolutional neural network (CNN), traffic sign image dataset, data augmentation, pre-processing.

1. Introduction

Traffic-sign recognition (TSR) technology allows cars to read and interpret roadside signs such as speed limits, warning of children in the area, and upcoming turns. Many car manufacturers are currently working on developing this technology. Image processing techniques are used to locate traffic signs and forward-facing cameras on contemporary cars, trucks, and other vehicles can be used to analyze them. Speed limits are one of the primary applications for TSR systems. Nonetheless, extra speed limit signs can likewise be used to accumulate information and show it on a GPS gadget's dashboard. This data can be utilized to make drivers aware of traffic signs and speed limits. The framework is prepared to perceive different traffic signs utilizing strategies like picture handling and PC vision, empowering it to identify new signs ceaselessly.

DOI: 10.54254/2755-2721/30/20230062

2. Related Work

This section provides a review of previous research on the detection of traffic signs in various regions of the world. Lue et al. (2018) [1] proposed a 3-stage information-driven framework for identifying image-oriented and text-oriented signs, using a camera mounted on a vehicle. The three stages include return-on-investment (ROI) extraction, refinement-description of ROI, and post-processing. However, their proposal had a significant drawback in the extensive post-processing stage. Mammeri et al. (2013) [2] addressed issues with the TSDR structure, a crucial component of ADAS, but their system only operated within a limited frequency range and had difficulty recognizing traffic signs with a lower-resolution camera, as well as camera vibrations and movements. Lee and Kim (2018) [3] developed a remarkable CNN traffic-sign identification system that simultaneously computes the precise location and boundary of traffic signs. While the accuracy was excellent, Lee's group was limited by high-resolution photos. Hu et al. (2016) [4] focused on three classes of objects: traffic signs, cars, and cycles. Their proposal detected all the three classes, by a single learning-based detection framework. In their model, the traffic sign detector needed the least amount of time as they used a smaller number of sub-detectors. When additional features were added for detection of other objects, the runtimes for the detection noticeably increased. Greenhaigh J and Mirmhdi (2015) [5] used the scene structure to pinpoint search regions within the image that have a higher probability of containing a traffic sign. Maximally stable extremal regions (MSERs) and hue, saturation, and value colour thresholding are used to locate a large number of candidates, which are then reduced by applying constraints based on temporal and structural information. Individual lines are scanned first as Maximally stable extremal regions and then they are grouped into to line. However, the false up-sides resulted in significant losses due to frustrated primary data, and the processing rate decreased from 14 frames per second to 6 frames per second.

Materials and methods:

2.1. Datasets

During the initial stages of the project, data collection is a crucial aspect. Images of various traffic signs were obtained from Kaggle, and have been pre-processed. This dataset includes a variety of classes and diverse sets of images. Using existing training data, image augmentation techniques were applied to create unique and individualized training images. These techniques include horizontal flipping, padding, cropping and rotation, which help to prevent over-fitting of the model. After augmentation, the dataset consisted of around 40,000 images. The collected data was then consolidated to generate a unique dataset, which was made public on Kaggle under the name "Traffic sign dataset". The dataset was then divided into a training, testing, and validation set, with 15% for testing, 15% for validation, and 70% for training the model, which will be used for classification. This division ensures that there is ample data available for training, resulting in a more accurate model. In total, 1000 images were used for each class, with 100 images each for testing and validation, and 800 images for training.

2.2. Convolutional Neural Networks

The convolutional neural network (CNN) is a popular deep learning model that is widely recognized for its effectiveness. It is a type of algorithm that is frequently employed to tackle challenges related to classification of image, and gaining popularity of various fields like music and health. The CNN model is composed of several layers that work together, including fully-connected layers, convolutional layers, and pooling layers. These layers are used to learn the data topologies dynamically through the backpropagation technique.

2.2.1. The Convolutional layer

The convolution layer is one of the most crucial elements of a CNN. It is accomplished by applying a channel to an info, which decreases the size of the picture while solidifying all field information into the solitary pixels. Figure 2 which shows a picture complexity that identifies the edges of an item.

DOI: 10.54254/2755-2721/30/20230062

2.2.2. The Pooling-Layer

A Pooling layer is used for diminish-size of a picture utilized as contribution for the convolutional layer. As a result, fewer features must be learned, and the network needs to perform less computation. Maximum pooling layers, global pooling layers, and other varieties are available, and average pooling layers. In this study, maximum pooling is employed. As shown in Figure 1, the maximum pooling layer reduces the dimensions of the image.

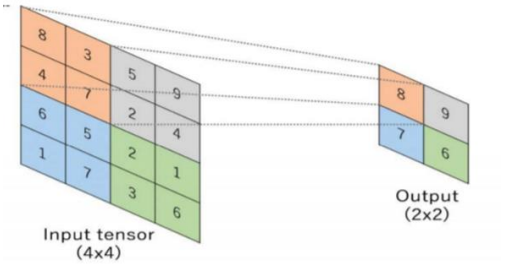

Figure 1. Max pooling layer.

2.2.3. Activation Function

Before passing on to the layer of neurons below, a non-linear adjustment known as an activation function is used to determine whether a neuron is stimulated. The activation function used in this case is Rectified Linear Unit (ReLU).

3. Proposed Model

The proposed work consists of several modules. First, input images are provided. Then, the shape detection on the color probability of the convolutional model is calculated as the region of interest in the proposed CNN model. This is based on the layers of the convolutional neural networks. After that, data augmentation and object detection are implemented. This is the step-by-step procedure that is planned to be followed in the project. (Fig 2)

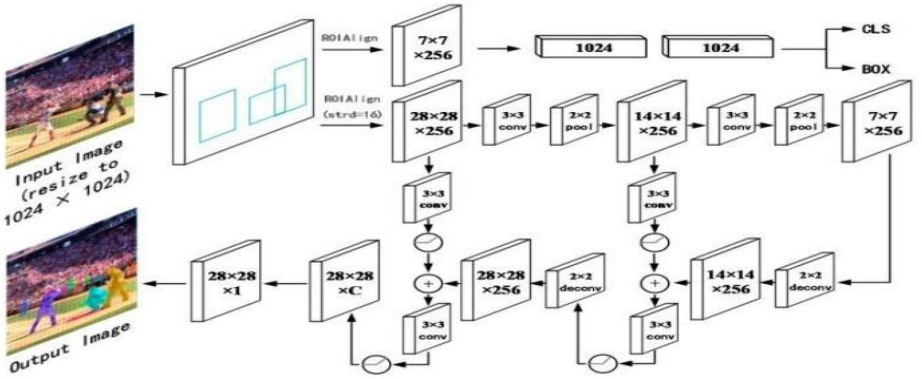

Figure 2. CNN framework.

3.1. Resnet50

The ResNet50 model consists of one MaxPool layer, one Normal Pool layer, and 48 Convolution layers. It comprises of (3.8 x 10^9) floating point operation. It was the typical variation of the ResNet model. ResNet 50 model achieves a top-1 error rate of 20.45 percent and also top5 error -rate of 5.26 percent. That is stated for the single model of 50 layers and not a group of them. These design could be used for the computer vision tasks, including image classification, object detection, and object recognition. It can also be used for non-computer vision tasks to benefit from its depth and save on computation. Figure 3 illustrates the design of the ResNet50.

DOI: 10.54254/2755-2721/30/20230062

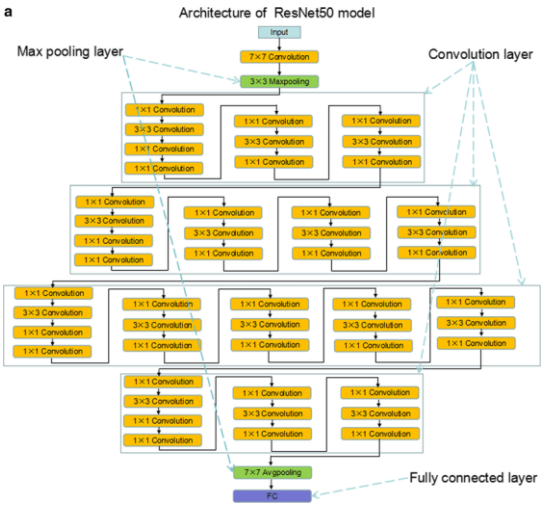

Figure 3. Resnet50 architecture.

3.2. InceptionV3

The image identifition model Inception V3 has demonstrated to achieve the accuracy of 78.1 percent on ImageNet datasets. Inception V3 recognition of image model was found to achieve more than 78.2 % accuracy on Image-Net dataset. Inception V3 model has 42 layer, that has few more V1 and V2 model. Despite this, effectiveness of a model is remarkable. Figure 4 illustrates the architecture of the Inception V3 model.

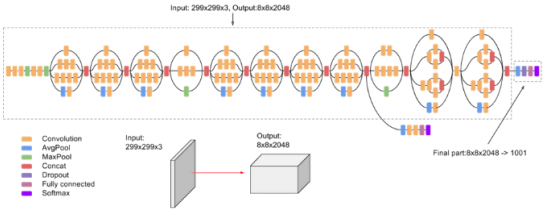

Figure 4. InceptionV3 architecture.

4. Result and discussion

The accuracy and loss for the VGG19 Model's training and validation datasets are shown in Table 1. With a learning rate of 0.01 across 25 epochs, this model was trained. Table 1 displays the VGG19 Model's accuracy and loss for the training and validation datasets. With a learning rate of roughly 0.01, this model performed 25 iterations of training, yielding a training accuracy of 95.58%. The table demonstrates that when training accuracy increases, the validation loss will rise relative to the first epoch. The model's accuracy was tested using the test dataset, and it came out at 88.36%. With 25 epochs and with a learning rate for 25 epochs of 0.01, the RESNET50 model fared better in contrast, achieving a training accuracy of 98.63%. Table 2 presents these findings. On the test dataset, the test accuracy for the model was 87.82%. Table 3 displays the InceptionV3 model's performance. It has a training accuracy of 98.20% and was trained for 25 epochs with a learning rate for respective epochs is 0.01. The model's testing precision was 88.93%. The results for the EfficientNetB4 model, which outperformed the other models with a test accuracy of 91.95 percent, are shown in Table 4.

DOI: 10.54254/2755-2721/30/20230062

Table 1. Performance of VGG19 model.

Epoch	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
1	1.7765	0.3320	1.1643	0.4817
5	0.6367	0.7346	0.6324	0.7429
10	0.3749	0.8389	0.4420	0.8138
15	0.2817	0.8745	0.3673	0.8586
20	0.1895	0.9291	0.3698	0.8663
25	0.1325	0.9518	0.3077	0.8912

Table 2. Resnet50 performance.

Epoch	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
1	1.872	0.6809	0.6390	0.7569
5	0.1981	0.9185	0.3474	0.8652
10	0.0995	0.9710	0.3795	0.8853
15	0.0396	0.9851	0.3471	0.9366
20	0.0348	0.9863	0.4201	0.8888

Table 3. Inceptionv3- performance.

Epoch	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
1	0.986	0.633	0.510	0.784
5	0.246	0.900	0.301	0.888
10	0.096	0.961	0.284	0.904
15	0.041	0.986	0.358	0.908
20	0.048	0.982	0.337	0.903

Table 4. Performance of efficientnetb4 model.

Epoch	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
1	0.9329	0.6421	0.5024	0.7950
5	0.2698	0.8986	0.2610	0.8829
10	0.1259	0.9512	0.2282	0.9147
14	0.0646	0.9732	0.2420	0.9250
15	0.0569	0.9790	0.2557	0.9299

5. Conclusion

For identification and the recognition of road traffic sign image, this effort put forth a workable deep learning technique that performed well under a range of situations, including changes in scale, direction, and illumination. The project introduces CNN, which incorporates advancements in parametrical values, data augmentation, and architecture. In real time, a cutting-edge tailored dataset was acquired for the submitted CNN model's training and validation. In addition to data augmentation, ResNet-50 is used for efficient output, it is convolutional brain network that has 50 layers. It stack the pretrained form of organization which prepared excess of 1,000,000 picture from ImageNet data set. The organization can order picture into the 1000 item classification. The images are sent to the InceptionV3 model. This Commencement V3 model has 42-layers, which was a couple of higher than V1 and the V2-models .The decrease of miss-rate and also the false the positive were a clear indicator that the reported performance gains were true.

DOI: 10.54254/2755-2721/30/20230062

References

[1] Luo, H., Yang, Y., Tong, B., Wu, F., Fan, B., 2018. Traffic sign recognition using a multi-task convolutional neural network. IEEE Trans. Intell. Transp. Syst. 19 (4), 1100–1111. https://doi.org/10.1109/TITS.2017.2714691.

[2] Mammeri, A., Boukerche, A., Almulla, M., 2013. Design of traffic sign detection, recognition, and transmission systems for smart vehicles. IEEE Wireless Commun. 20 (6), 36–43

[3] Lee, H.S., Kim, K., 2018. Simultaneous traffic sign detection and boundary estimation using convolutional neural network. IEEE Trans. Intell. Transp. Syst. 19 (5), 1652–1663. https://doi.org/10.1109/TITS.2018.2801560.

[4] Hu, Q., Paisitkriangkrai, S., Shen, C., van den Hengel, A., Porikli, F., 2016. Fast detection of multiple objects in traffic scenes with a common detection framework. IEEE Trans. Intell. Transp. Syst. 17 (4), 1002–1014. https://doi.org/10.1109/TITS.2015.2496795.

[5] Greenhalgh, J., Mirmehdi, M., 2015. Recognizing text-based traffic signs. IEEE Trans. Intell. Transp. Syst. 16 (3), 1360–1369. https://doi.org/10.1109/ TITS.2014.2363167.