Classification of Traffic Sign for Smart Vehicle using Ensemble Learning and Deep Learning

Tushar Malpani
Student
Amity School of Engineering and Technology
Sector 125, Noida
tusharmalpani200@gmail.com

Yuvraj Singh Chauhan
Student
Amity School of Engineering and Technology
Sector125, Noida
yuvraj.chauhan1990@gmail.com

Gauri Gupta
Student
Amity School of Engineering and Technology
Sector 125, Noida
gaurigupta707@gmail.com

Nirbhay Kashyap
Assistant Professor
Amity School of Engineering and Technology
Sector 125, Noida
nkashpay@amity.edu

***Abstract*— Traffic signs are crucial for directing traffic, enforcing safe driving practices, and lowering the number of collisions, injuries, and fatalities. Automated traffic sign recognition and identification is a need for all Intelligent Transportation Systems (ITS). It cannot be overstated how important automatic traffic sign detection and identification is in the era of self-driving cars [1]. This will help the driver's eyesight and maneuvering abilities, which can't be relied upon to always be accurate for all weather-related issues. Road safety is being compromised by the human eye's limited ability to read a traffic sign. This study presents a deep-learning-based autonomous traffic sign cognizance technique for India. AlexNet, VGG-16, ResNet-50, and GoogleNet are the four designs of the Convolutional Neural Network (CNN), which is the foundation for autonomous traffic sign detection and identification. The proposed idea was evaluated using a novel dataset made up of 13971 photos divided into 59 categories. As a pre-trained transfer learning model ResNet-50 has greater accuracy and operational speed to produce a best fit model, we will be incorporating these images to it[2].**

Keywords—Alexnet, VGG16, Resnet, Google net, Convolution , architecture, ensemble learning, deep learning, Lenet, CNN, traffic sign, etc

I. Introduction

A. Motivation

Smart vehicles have a critical responsibility of ensuring safe and efficient driving, and traffic sign recognition is a key component of this. The complexity of the driving environment brought on by the growing number of vehicles on the road makes it difficult for human drivers to stay on top of all the traffic signs. As a result, it is crucial to deploy smart vehicles that are outfitted with cutting-edge technology to increase traffic safety, such as computer vision and deep learning.[3] The main objective of this initiative was to create methods for reducing traffic accidents. We are driven to reduce the amount of accidents that occur. Road accidents in India not only claim lives, but also have a significant impact on their families. While there are several ways to lower the likelihood of collisions, using Advanced Driver Assistance Systems (ADAS), which include traffic sign recognition, is the most effective. The greatest solution for this issue is ADAS.[4]

B. Research Objective

We used a cutting-edge deep learning technique known as transfer learning to achieve the research goal of developing a better deep learning model for image classification.High precision is a benefit of this strategy and speed in image classification and can utilize a pre-existing dataset for training, eliminating the need for a large dataset. We compiled a dataset consisting of an image for each class, sourced from a government website, and performed image scaling. The images were then pre-processed, converted to grayscale, and subjected to standardized lighting to simulate real-world challenges. The project results will provide insights into the model's limitations and accuracy levels.

C. Difficulties

Similar challenges to object identification in real situations arise when extracting or detecting a traffic sign for later recognition.

1) Lighting conditions might fluctuate and are not within our control. The lighting changes depending on the time of day, season, amount of cloud cover, and other meteorological factors, etc[5].

2) The presence of additional items. Traffic signs are frequently surrounded by other items, with the simple exception of motorways. Partial occlusions, shadows, etc. result from this[6].

3) Due to the large number of degrees of freedom, it is not practical to create off-line models of every potential variation of the sign's appearance. The size of the item relies on its distance from the camera; if the camera optic axis is not perpendicular to the sign, the scale for each axis varies, changing the aspect of the sign. In addition, the physical state of a sign changes with age, accidents, etc[7]

To begin our research, we utilized the German Traffic Sign Dataset (GTSR) available on Kaggle for traffic sign detection using deep learning techniques. Through this initial experimentation, we were able to gain insights into the challenges involved in traffic sign detection and observe the variations in accuracy among different models. We applied four different CNN models, namely AlexNet, VGG-166, GoogleNet, and ResNet-50, to our dataset and compared their performances to determine the best model based on accuracy. Among the four models, the ResNet-50 architecture provided the highest accuracy, and we chose it as our pre-trained model for transfer learning[8].

For our next phase, we collected data from a government website and created the Indian Traffic Sign Dataset for our model. To prepare the images for deep learning, we performed pre-processing techniques, such as converting the images to grayscale and subjecting them to standardized lighting to simulate real-world challenges. With this data, we applied transfer learning using our pre-trained ResNet-50 model, which resulted in improved accuracy and performance for our Indian Traffic Sign Dataset[9].

Furthermore, we also incorporated object detection using OpenCV to enable real-time analysis and detection of traffic signs in captured video footage. We developed a graphical user interface (GUI) and a fully-fledged website to display our results, making them easily accessible and user-friendly. This comprehensive approach enabled us to evaluate the performance of our model, identify areas of improvement, and develop a practical and user-friendly solution for traffic sign recognition[10].

II. LITERATURE REVIEW

Evaluation & synthesis of the existed research paper relevant to our area, it consist of the conclusion of different research papers to explain and validate the basis of our research.

1) A vision-based vehicle guidance system for road vehicles may do the following three essential tasks, according to De La Escalera et al. in the article Road traffic sign detection and classification: Road detection, obstacle detection, and sign identification are the first three steps. He employs an algorithm with two primary components that makes use of the features. The first one uses shape analysis to identify the signs and colour thresholding to divide the image into sections. The second one classifies data using a neural network. Results from certain natural scenes are shown.

2) In the article "Detection of traffic signs in real-world images," Houben, S. et al. He also presents a web-interface for comparing methods, carefully selected assessment measures, baseline findings, and a set of real-world benchmark data for traffic sign detection. Real-time traffic sign identification is the process of identifying a traffic sign's location in photos taken in the wild, according to the German Traffic Sign identification Benchmark.

3) Yang et al. discuss the use of traffic sign identification in their work with autonomous cars with intelligence and driver aid technologies. towards the recognition and classification of traffic signs in real-time. Real-time traffic sign recognition is the focus. They propose a detection module that is 20 times quicker than the best detection module currently on the market to achieve this.

4) In the article An overview of traffic sign detection and classification approaches, Saadna et al. provide a summary of a few cutting-edge and effective techniques for traffic sign detection and classification. In fact, locating areas of interest that contain traffic signs is the primary objective of detection techniques. For easy reference, the various datasets detection and classification methods are provided in tables alongside the methodologies.

5) Article by Bahlmann The article outlines a real-time, dependable traffic sign detection, tracking, and identification system based on computer vision. A system that uses colour, shape, and motion data to identify, track, and recognise traffic signs. The recommended approach is divided into two sections. To identify signs, we first employ a collection of Haar wavelet characteristics obtained via AdaBoost training. Second, classification is carried out using Bayesian generative modelling. Using the tracking data, the hypotheses are fused across a number of frames.

6) In their paper titled "Indian traffic sign detection and recognition using deep learning," Rajesh Kannan et al. describe an autonomous deep-learning-based technique for cognizance of traffic signs in India." Convolutional Neural Network (CNN) and Refined Mask R-CNN (RM R-CNN) end-to-end learning provided the inspiration for automatic traffic sign identification and recognition.

7) Object recognition in outdoor settings is discussed by A. de la Escalera*, J. Ma Armingol, and M. Mata in their work Traffic sign recognition and analysis for autonomous vehicles. In these circumstances, the lighting is unpredictable, objects may be partially obscured, and it is unknown in advance where they are going to be. The employment of a genetic algorithm during the detection phase allows for an adaptation to changes in location, size, rotation, weather, partial occlusion, and the presence of similar-colored objects.

III. METHODOLOGY

The classification of image can be of two types Supervised and unsupervised, in our case the classification of traffic sign is supervised as the dataset is available online and can be downloaded as well. We are using dataset German Traffic Sign Recognition Benchmark[11].

A) *Description of Dataset*

The GSTRB dataset in available on Kaggle, the dataset includes more than 50 thousand images that are spread over 43 different classes, the dataset provided is actually imbalance (the different classes have different number of images) the dataset is then divided into Training set (31367), Validation set (7842) and Testing set (12631) images as shown in Fig [1].

Fig 1. Overview of Different Signs

B) Flow of the Model

C)

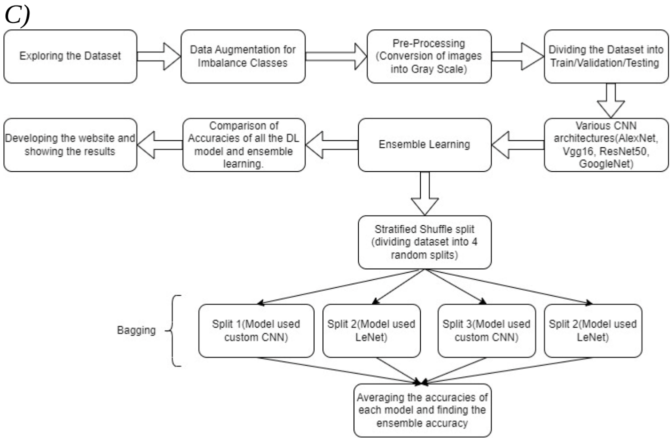


Fig 2. Flow of the Model

The flow of our model will work as initially we are taking the dataset and exploring, majority of the images in the dataset are of size (200x200x3). After that using Data augmentation on different classes in order to maintain uniformity of images and model does not get biased for particular classes and for that we are using techniques like shifting, rotation and flipping to increase dataset. Then we are converting the colored images into gray scale for model input purpose as shown in figure [12]. In the next we are normalizing the dataset by converting the dataset into binary form using python function (to_categorical) after that we divide the dataset into training, validation and testing set after that we used various CNN architectures like AlexNet, Vgg16, ResNet50 and Google Net and then comparing the accuracies and developing a interface and showing results on it. After that we are making 2 other CNN models 1st one is our own custom CNN and other one is LeNet with some modification we are using stratified shuffle split to reduce the overfitting and splitting the dataset randomly into 4 parts and applying our 2 models on these split and the ensemble the training of 4 models into one and testing it on the dataset. At last we doing the comparison between individual models and ensemble method and showing the result on website[13].

D) Classification using various CNN architectures

*1) Customized Alex Net:*The architecture we used consist of 5 Convolution, 3 Max Pooling and 3 fully connected layers. The standard input size of an image to this architecture is i.e. (227x227). The AlexNet architecture that we build uses dropout after each dense layer so that overfitting of the model can be reduced to great extent[14]. The model also uses batch normalization to speed up training process and making learning process convenient. The loss function we used is spare categorical cross entropy as we have multiple classes and optimizer we used is stochastic gradient descent and the learning rate we used is fixed (0.001). After summarizing the model, we found that the total trainable parameters are around 58 million[15].

*2) Modified Vgg16 from Scratch :*The Vgg16 architecture as the name says it has 16 layers, 13 convolution layers and 3 fully connected layers the standard input size to this architecture is (224x224x3). The Vgg16 contains multiple non-linear hidden layers due to which the networks depth is increased and finer pixels are extracted. The kernel was initialized using he_normal(stddev = sqrt(2 / fan_in) initializer and relu activation function is used to solve the problem of vanishing gradient descent and at the last layer Softmax activation function is used so that the overall probability lies between 0 and 1. The loss function we used is spare categorical cross entropy as we have multiple classes and optimizer we used is Adam. After summarizing the model, we found that the total parameters are around 70 million and total trainable parameters are around 56 million because of dropout[16].

3) Residual Neural Network with modification: Residual Neural network are mainly used when we want heavy batch normalization and it mainly uses “skip connections”, there are different types of residual networks like ResNet-34, ResNet-50, ResNet-101, ResNet-152. We are using architecture with 50 layers, the input is resized to (50x50x3) to reduce the processing time as it contains lots of layers and is complex. We are also applying global average pooling which averages out the channel value across the feature map and it also reduces the spatial dimension. Before the last dense layer, we are using dropout and after summarizing it we get 23 million parameters. Since we are running it for we 50 epochs, we are using Model Checkpoint and Early Stopping to get the best training accuracy and stop the process if validation accuracy after every 5 epoch start to decrease[17].

4) Google Net (22 Layers): GoogLeNet architecture comprises of 22 Layers and it was the winner of the ILSVRC-2014. This architecture is mainly uses the concept of 1x1 convolutions to reduce the dimensionality and in order to reduce the feature maps it uses global average pooling. In our model we are passing the input size of (224x224). We are also using learning rat scheduler in order to adjust the learning rate after few epochs as the loss increases initially and then gradually decreases. We are using the inception module, inception module as a tool that helps a computer "see" and understand images better by breaking them down into different pieces and analyzing them separately, before combining the results to make a final prediction. It's like using different magnifying glasses to look at different parts of a picture, and then putting all the information together to understand what the picture is showing. After applying dropout, we get total 10 million parameters[18].

E) Ensemble Method of Learning

1) Stratified shuffle split

One of the cross validation technique used to split dataset into training and testing sets while maintaining the same class distribution in each set. This method shuffles the data randomly before splitting it into the desired number of folds. The class distribution is ensured by assigning an equal number of samples from each class to each fold. This technique is useful when dealing with imbalanced datasets, where one class may have significantly more samples than others. By stratifying the split, the model can learn from each class equally, improving its ability to predict on unseen data. Overall, this technique helps to minimize bias in the model's performance and improve its generalization ability. In our case we have split the dataset 4 times into 80:20 ratio of training and testing dataset [19].

2) Model configuration for ensemble learning

a) The customized CNN contains 4 convolution layer and 2 max pooling layer and 3 dense layer. We have used categorical class entropy as the loss function and Adam optimizer which has been applied to two different random splits of the dataset.

b) The LeNet is a shallow CNN which contains 2 Convolution layer, 2 Average pooling layer and 3 dense layers it uses categorical cross entropy and Adam optimizer which is also being applied to the two different random split of dataset.

After fitting all four models and running them on 20 epochs each we get the all the accuracies and after that we combine the all four models(Bagging) and find their combines accuracies as the ensemble accuracy.

IV RESULTS AND CONCLUSION

Our results of various modified and custom CNN models on the German traffic sign dataset are as follow:

1) On the Customized AlexNet we get the training acc = 96.06% and the validation accuracy=91% and when applying on the testing dataset we get the accuracy= 83.8% and the accuracy and the loss graph is shown below in Fig 3.

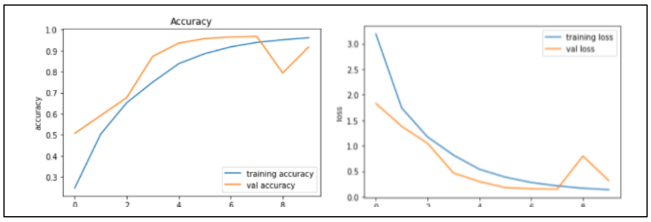


Fig 3 Accuracy and Loss graph of Alexnet model.

2) On the modified Vgg16 we get the training accuracy =98% and the validation accuracy=98.9% and when applying on the testing dataset we get the accuracy= 89.688% and the loss graph and the accuracy graph is shown below in Fig 4.

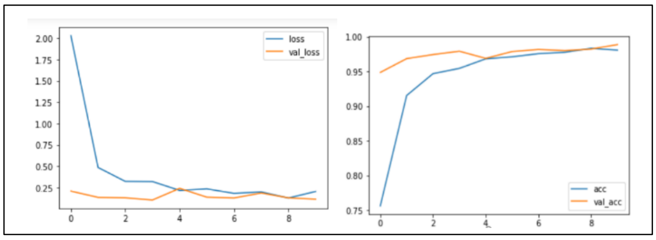


Fig 4 Loss and Accuracy graph of Vgg16 model

3) On the modified Resnet we get the training accuracy =98% and the validation accuracy=97.33% and when applying on the testing dataset we get the accuracy= 93.96% and the accuracy and the loss graph is shown below in Fig 5.

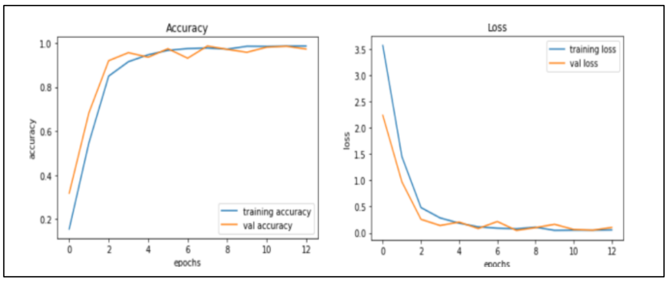
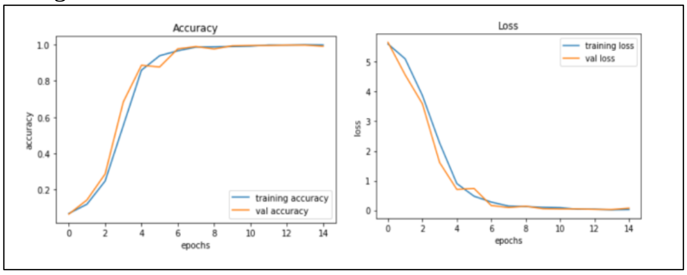


Fig 5 Accuracy and Loss Graph of Resnet50

4) On the Customized GoogleNet we get the training accuracy = 99.8% and the validation accuracy=99.1% and when applying on the testing dataset we get the accuracy= 95.92% and the accuracy and the loss graph is shown below in Fig 6.



5) When we applied various models like custom CNN and Lenet on the different split of dataset the following accuracies as shown in Table 1

TABLE 1 Accuracies on various splits of Dataset

CNN architecture on 4 splits of dataset	Training accuracy	Validation accuracy	Testing accuracy
Custom CNN (split 1)	95.59%	98.64%	95.24%
LeNet (split 2)	97.9%	97.45%	95.34%
Custom CNN (split 3)	95.58	98.83%	95.24%
LeNet(split 4)	99.04%	97.33%	95.36%

After combining all the above 4 models we get the ensemble accuracy on testing dataset = 96.31% which is higher than individual model accuracy therefore it is better to use ensemble technique after applying various model in order to increase the efficiency of the overall model.

The traffic sign classification and detection can be achieved the highest accuracy using the ensemble method of learning by combining the various individual models as the testing accuracy of our project is higher on ensemble learning.

Overall, the conclusion of the project would be that traffic sign detection and classification is a promising area of research that has the potential to make a significant impact on road safety and traffic management[20].

REFERENCES

[1] De La Escalera, A., Moreno, L. E., Salichs, M. A., & Armingol, J. M. (1997). Road traffic sign detection and classification. *IEEE transactions on industrial electronics*, *44*(6), 848-859.

[2] Houben, S., Stallkamp, J., Salmen, J., Schlipsing, M., & Igel, C. (2013, August). Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In *The 2013 international joint conference on neural networks (IJCNN)* (pp. 1-8). Ieee.

[3] Yang, Y., Luo, H., Xu, H., & Wu, F. (2015). Towards real-time traffic sign detection and classification. *IEEE Transactions on Intelligent transportation systems*, *17*(7), 2022-2031.

[4] Saadna, Y., & Behloul, A. (2017). An overview of traffic sign detection and classification methods. *International journal of multimedia information retrieval*, *6*(3), 193-210.

[5] Bahlmann, C., Zhu, Y., Ramesh, V., Pellkofer, M., & Koehler, T. (2005, June). A system for traffic sign detection, tracking, and recognition using color, shape, and motion information. In *IEEE Proceedings. Intelligent Vehicles Symposium, 2005.* (pp. 255-260). IEEE.

[6] Bi, Z., Yu, L., Gao, H., Zhou, P., & Yao, H. (2021). Improved VGG model-based efficient traffic sign recognition for safe driving in 5G scenarios. International Journal of Machine Learning and Cybernetics, 12(11), 3069-3080.

[7] Yu, J., Ye, X., & Tu, Q. (2022). Traffic Sign Detection and Recognition in Multiimages Using a Fusion Model With YOLO and VGG Network. IEEE Transactions on Intelligent Transportation Systems.

[8] Guennouni, S., Ahaitouf, A., & Mansouri, A. (2014, October). Multiple object detection using OpenCV on an embedded platform. In 2014 Third IEEE International Colloquium in Information Science and Technology (CIST) (pp. 374-377). IEEE.

[9] Chandan, G., Jain, A., & Jain, H. (2018, July). Real time object detection and tracking using Deep Learning and OpenCV. In 2018 International Conference on inventive research in computing applications (ICIRCA) (pp. 1305-1308). IEEE.

[10] Huang, S. C., Lin, H. Y., & Chang, C. C. (2017, June). An in-car camera system for traffic sign detection and recognition. In 2017 Joint 17th World Congress of International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems (IFSA-SCIS) (pp. 1-6). IEEE.

[11] Wan, H., Gao, L., Su, M., You, Q., Qu, H., & Sun, Q. (2021). A novel neural network model for traffic sign detection and recognition under extreme conditions. Journal of Sensors, 2021, 1-16.

[12] Sütő, J. (2022). An improved image enhancement method for traffic sign detection. Electronics, 11(6), 871.

[13] Rajendran, S. P., Shine, L., Pradeep, R., & Vijayaraghavan, S. (2019, July). Real-time traffic sign recognition using YOLOv3 based detector. In 2019 10th international conference on computing, communication and networking technologies (ICCCNT) (pp. 1-7). IEEE.

[14] Dhar, P., Abedin, M. Z., Biswas, T., & Datta, A. (2017, December). Traffic sign detection—A new approach and recognition using convolution neural network. In 2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC) (pp. 416-419). IEEE.

[15] Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2011, July). The German traffic sign recognition benchmark: a multi-class classification competition. In The 2011 international joint conference on neural networks (pp. 1453-1460). IEEE.

[16] He, Z., Xiao, Z., & Yan, Z. (2020, November). Traffic sign recognition based on convolutional neural network model. In 2020 Chinese automation congress (CAC) (pp. 155-158). IEEE.

[17] Shahud, M., Bajracharya, J., Praneetpolgrang, P., & Petcharee, S. (2018, November). Thai traffic sign detection and recognition using convolutional neural networks. In 2018 22nd International Computer Science and Engineering Conference (ICSEC) (pp. 1-5). IEEE.

[18] Jao, J., Song, C., Peng, S., Xiao, F., & Song, S. (2019). Improved traffic sign detection and recognition algorithm for intelligent vehicles. Sensors, 19(18), 4021.

[19] Prakash, A. S., Vigneshwaran, D., Ayyalu, R. S., & Sree, S. J. (2021, April). Traffic Sign Recognition using Deeplearning for Autonomous Driverless Vehicles. In 2021 5th International Conference on Computing Methodologies and Communication (ICCMC) (pp. 1569-1572). IEEE.

[20] Peng, X., Li, Y., Wei, X., Luo, J., & Murphey, Y. L. (2017, November). Traffic sign recognition with transfer learning. In 2017 IEEE Symposium Series on Computational Intelligence (SSCI) (pp. 1-7). IEEE.