Traffic Sign Recognition Using Transfer Learning with Pre-trained CNNs and Bayesian Hyperparameter Optimization

Arpana Chaturvedi[1], Praveen Malik[2], Deepak Jugran[3]
[1]*Associate Professor, Dept of IT,AI,ML,New Delhi Institute of Management, Delhi,*
[2]*Assistant Professor,Dept of IT and DA,New Delhi Institute of Management,*
[3]*PGDM, Student, New Delhi Institute of Management*

***Abstract*—The visual appearance of traffic signs varies greatly in real-world settings, which is one of their defining characteristics. The perception of road signs is impacted, for instance, by variations in lighting, changing weather, and partial occlusions. In real-world applications, a huge variety of distinct sign classes must be accurately detected. Traffic signs are made to be simple to read by humans, who are particularly good at this task. However, identifying traffic signs still appears to be a difficult pattern recognition task for computer systems. Transfer learning's major objective is to transfer pertinent knowledge from the source domain while enhancing learning in the target domain. This research concentrates on transfer learning using deep convolutional neural network (CNN) and its designs, including VGG16, ResNet101 and EfficientNetB2, to handle such challenges. Then by applying Bayesian Optimization, we tuned the hyperparameters. In the study, these pre-trained CNN classifiers without their TOP layers are trained and tested on three different datasets as German Traffic Sign Recognition Benchmark (GTSRB), the Chinese Traffic Sign Recognition Benchmark (CHTSRB) and Indian Traffic Sign Dataset (ITSD). According to experimental findings, the suggested approach performed well in terms of metrics for evaluating the identification of traffic signs. The ResNet101 model performs better than all other implemented models on all three datasets, it gives the accuracy of 97.6537% on GTSRB, 96.7677% on CHTSD and 90.9547% on ITSD.**

***Index Terms*—Bayesian Optimization, Convolutional Neural Network, Deep Learning, Pre-trained Models, Traffic Sign Recognition, Transfer Learning.**

I. INTRODUCTION

In most economic sectors today, automobiles are one of the most frequently used forms of transportation. Drivers may be threatened by a multitude of risks when driving, which leads to an increase in accidents and injuries over time. The World Health Organization has conducted a study in 2022 showing that road accidents cause the death about 1.3 million people each year because of road traffic crashes [1] . Additionally, one of the top causes of death worldwide each year is traffic accidents.

Intelligent systems are consequently at the heart of road traffic safety in automobiles to decrease the frequency of traffic deaths. These systems include the advanced driving assistance system (ADAS), which enables the autonomous car to recognise threats associated with nearby objects using onboard sensors and receive real-time support in specific traffic circumstances. Autonomous driving has advanced quickly thanks to ADAS technology. Based on visual information obtained from complex sensors like cameras, a TSR system can recognise one or more traffic signs. A better awareness of the environment, which relates to the driving space for cars on the side of the road topography, is also influenced by a comprehension of road scenes. Their employment in the process of managing and monitoring road traffic has been made possible by significant advancements in the creation of novel sophisticated technologies and the widespread deployment of fixed and mobile sensors, such as image sensors.

To lessen the impact of human lives lost in traffic accidents and rising congestion, intelligent transportation systems (ITS) have undergone a significant revolution because of advancements in computer vision research. Additionally, the emergence of deep neural networks (DNNs), the development of effective computers with graphics processors, such as graphics processing units (GPUs), and the rapid evolution of machine learning algorithms, particularly with the enormous growth in traffic data volumes (big data), have all contributed to significant advancements in the computer vision domain. But it should be remembered that some vision-based applications, such real-time embedded systems, need a lot of memory and fast computation. Indeed, one of the trickiest problems in computer vision, which entails investigating and interpreting the environment around the vehicle, is segmentation-based road detection.

Deep learning models are taught gradually with a vast volume of data, automating the extraction and training of hierarchical feature representations, in contrast to conventional methods, which are based on learning hand-crafted features like edges and corners. Convolutional neural networks (CNNs) are feedforward neural networks that constitute the foundation for the most popular computer vision applications, including object recognition, semantic segmentation, visual tracking, etc. CNNs are composed of a set of trainable and non-trainable layers. In general, the prediction step of the tasks of object identification and semantic segmentation is very different. One aspect of object identification is determining the class labels and bounding boxes of the identified items, while another aspect of semantic segmentation entails labelling every pixel in the image and dividing it into dense sections.

In this study, we'll focus on the problem of object classification, specifically, traffic signs recognition which faces numerous challenges. Bad weather visibility, occlusion and many other factors poses a serious challenge for a recognition system to recognize traffic signs accurately. Further unavailability of appropriate data to train CNN models and long training times reduces the efficiency and accuracy of these model to perform well on real world scenarios. We'll make use of transfer learning approach to reduce the overall training time and computational requirements. An optimization algorithm called Bayesian Optimization will be used to optimize the hyperparameter of the model to increase the overall accuracy of the model.

II. RELATED WORK

Numerous methods for recognizing traffic signs have been published. In [2], a hybrid dataset made up of a German traffic sign identification benchmark dataset from Kaggle and a self-created Indian traffic sign dataset is used to recognize traffic signs using convolutional neural networks. the model achieved an accuracy of 95.45% for hybrid datasets, 91.08% for the Indian dataset alone, and 99.85% for the German dataset alone.

In [3], proposed a CNN based recognition system which made use of VGG architecture, overfitting of the model was reduces by adding batch normalization and dropouts. To take care of class imbalance, image augmentation was used. The model achieved 99.33% accuracy on German dataset. Another work [4], proposed a similar VGG-16 architecture and dropout regularization-based approach. Moreover, other data processing techniques like shuffling, normalization and gray scaling are applied resulting in 98.44% accuracy on German dataset.

In [5], an architecture based on ResNet50 used as a feature extractor and a classifier to determine the final class label of the traffic sign board. The images are pre-processed by clipping of edges, image enhancement and size normalization. The resulting model gave the accuracy of 98.84%. Further, [6] presented an approach which made use of CNN and LSTM for traffic sign recognition. The model recorded an efficiency of 96.76% on German benchmark dataset and 71.51% on IRSDBv1.0.

A non-CNN approach is mentioned in [7]. It makes use of vision transformer for classification of traffic sign images. Even though transformers are able to give good results on many computer vision tasks, they underperform highly in comparison to CNN by 12.81%, 2.01% and 4.37% on German, Indian and Chinese traffic sign datasets.

III. METHODOLOGY

A. Framework

The proposed model is a comprehensive pipeline that consists of an image pre-processing module, a pre-trained state-ofthe-art Convolutional Neural Network (CNN), and a DenseNN classifier. This pipeline ensures that images in the dataset undergo various operations such as resizing, augmentation, normalization, and other pre-processing techniques before being fed into the pre-trained CNN, which acts as a feature extractor for the Dense-NN classifier.

In the first step, default hyperparameters for the Dense-NN classifier, as listed in table 1, are used to train and evaluate the proposed model. This approach is expected to lead to a slight drop in performance and accuracy due to the presence of a pre-trained CNN. In the second step, hyperparameters of the Dense-NN classifier are optimized and tuned using the Bayesian Optimization Algorithm to increase the accuracy of the model.

Table 1: Default Architecture of Dense-NN Classifier

Layer	Neurons	Activation
Dense	128	ReLU
Dropout %	25	
Dense	128	ReLU
Dropout %	25	
Dense	128	ReLU
Prediction Layer	Acc. to # Classes	Softmax

The Hyperparameter Optimization step is crucial as it significantly improves the accuracy of the model by finding the optimal hyperparameters. This approach replaces the need to train a new SOTA CNN architecture from scratch, which is both computationally expensive and time-consuming. Overall, the proposed model with its pre-processing pipeline and hyperparameter optimization technique is a robust and efficient approach to achieve state-of-the-art performance on traffic sign datasets.

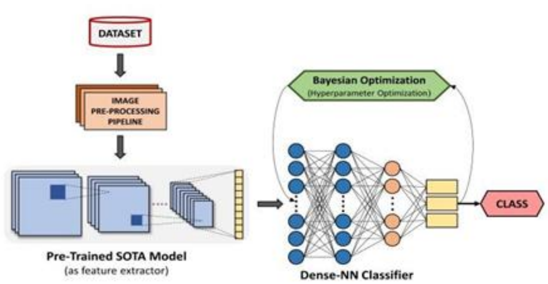

Fig 1. Architectural Overview

B. Image Preprocessing Pipeline

1) Before loading the datasets into a variable, the images are resized to (64*64*3) matrix to remove any inconsistency in image sizes and to avoid the high computation requirements to process larger images. Image Augmentation is used to resolve any class imbalance present in the dataset. Zero Centering is applied with respect to the ImageNet dataset before normalization to get the optimal results from the pretrained models and to avoid any gradient saturation. As, the pixel values for the images in given datasets lies between [0, 255]. Normalization is done using the below formula to scale the pixel values to the range of [0, 1]:

$$Normalized\ value = \frac{(Pixel\ Value).astype('float32')}{255}$$

Fig 2. Image Preprocessing Pipeline

2) Image Augmentation: One of the most popular regularization techniques that tries to increase the generalizability of deep models is data augmentation. By producing fresh data samples artificially, it is possible to enhance the amount of data that is available for training weights. Figure depicts a major imbalance in the distribution of data across classes in the GTSRB dataset (class ID). It could be possible to solve this problem by artificially increasing the data volume. By applying augmentation operation such as random rotations and cropping, new image data is generated. Similar technique is used on Indian Traffic Sign Dataset as well.

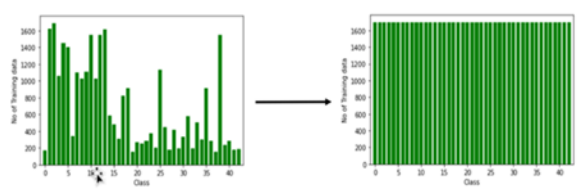

Fig 3. Fixing class imbalance using Image Augmentation

C. Transfer Learning with Pre-Trained CNNs

Transfer Learning is a technique where an already trained model is reused as a starting point for a model on a second task. We can say that it is a technique refers to the situation where what has been learned in one setting will be exploited to improve the performance of another model. Transfer learning drastically reduces the training time of the new model as one can use the previously learned weights and biases to solve new problems.

The proposed architecture will take advantage of pretrained SOTA models which are loaded with ImageNet weights and will be without their classification layer by giving 'Include Top = False' argument to the TensorFlow API. The pretrained model will act as a feature extractor for the Dense-NN layer. The training time and requirement of large amounts of data is expected to reduce drastically while using a pretrained model. Hyperparameter tuning with the help of Bayesian Optimization Algorithm will help in increasing the overall accuracy by searching for the optimal hyperparameters for training the proposed model.

In this paper, the comparison of ResNet101 [8], VGG16 [9] & EfficientNetB2 [10] is used on the three given datasets GTSRBD [11], CHTSD and ITSD after and before Bayesian Optimization.

D. Bayesian Optimization

To lead a search of a global optimization issue efficiently and effectively, Bayesian Optimization offers a systematic method based on the Bayes Theorem. It functions by creating a surrogate function, a probabilistic model of the objective function, which is then effectively searched with an acquisition function before candidate samples are chosen for evaluation on the actual objective function. In applied machine learning, Bayesian optimization is frequently used to fine-tune the hyperparameters of a given well-performing model on a validation dataset.

In our proposed model, an objective function is formulated which returns the final accuracy of model after training. A new hyperparameter is applied after each iteration..

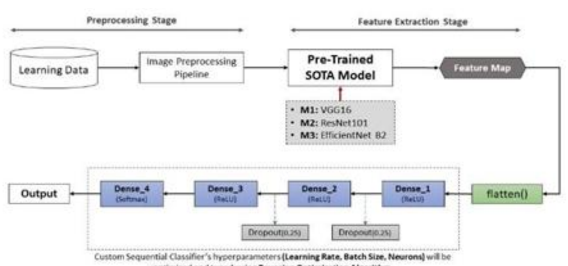

Fig 4. Proposed Model

Bayesian Optimization Algorithm is initiated and continued for 30 iterations to search for the optimal hyperparameters for the Dense-NN Classifier. After receiving the results, the new hyperparameters are applied to the models. For results, the performance of each model before and after optimization is compared

E. Evaluation Parameters

In this paper, the suggested deep learning approach-based traffic sign identification is evaluated using four performance measures:

1) Accuracy: The number of accurately predicted data points made by the model across all sorts of predictions made is the most significant deep learning indicator and known as accuracy of the model.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

2) Precision: Is used to measure the number of forecasts for the positive class that are accurate.

$$Precision = \frac{TP}{TP + FP}$$

3) Recall: Is used to measure the number of accurate positive predictions made from all the dataset's positive examples.

$$Recall = \frac{TP}{TP + FN}$$

4) F1-Score: Is an indicator of the correctness of the model, and it is equal to the multiplication of the recall and precision measures.

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

IV. EXPERIMENTAL RESULTS

A. Datasets

- The German Traffic Sign Recognition Benchmark (GTSRB) dataset is a multi-class, single-image dataset that poses a challenge in traffic sign classification tasks. The dataset consists of 51,839 samples, ranging in size from 15 × 15 to 250 × 250, and not all of them are square. This dataset has 43 categories, each of which comprises 100~1000 images including prohibitory signs, danger signs, and mandatory signs. The training set contains 39,209 images; the remaining 12,630 images are selected as the testing set.
- The CHTSD dataset is used to assess how well the suggested model recognized items. There are 58 sign categories and 6164 traffic sign images in the TSRD. The photos are separated into training database and testing database sub-databases. While the testing database has 1994 photos, the training database has 4170.
- Indian Traffic Sign Dataset includes 13971 Images which are divided into 59 classes bases on the type of traffic signs. Images are 32 x 32 in size. The dataset is split into 10,478 training images and 3493 testing images.

B. Results

The model under consideration has been trained on the GTSRDB dataset, utilizing three different state-of-the-art models as feature extractors. Specifically, the first model (M1) employs the VGG16 architecture, while the second model (M2) utilizes ResNet101, and the third model (M3) leverages the EfficientNetB2 architecture as the feature extractor.

Table 2: Results on GTSRBD (Default Setting)

Model	Accuracy	Precision	Recall	F1-score
M1	88.7485	0.8944	0.8875	0.8892
M2	84.5190	0.8673	0.8452	0.8499
M3	90.8377	0.9139	0.9084	0.9094

Upon application of the Bayesian Optimization Algorithm for tuning and optimizing hyperparameters of the Dense Neural Network (Dense-NN), the models were retrained using the newly optimized hyperparameters. As a result of this process, the models achieved notable improvements in performance, as evaluated on the GTSRDB dataset.

Table 3: Results on GTSRBD (Optimized)

Model	Accuracy	Precision	Recall	F1-score
O-M1	97.1056	0.9715	0.9711	0.9711
O-M2	97.6537	0.9769	0.9765	0.9765
O-M3	97.0288	0.9706	0.9703	0.9703

The experimental results shown in table 2 reveal that, when trained with default hyperparameters, the third model (M3) utilizing EfficientNetB2 as its transfer learning model achieved the highest accuracy on the German traffic sign dataset. However, the models trained with default hyperparameters failed to generalize well on the testing data, as demonstrated by the learning curve in fig 6, and took a significantly longer time to generalize. On the other hand, all three models (O-M1, O-M2, O-M3) trained with fine-tuned hyperparameters obtained through Bayesian Optimization exhibited excellent generalization capability on the testing data, with low time taken to generalize. These optimized models achieved good accuracy scores, with O-M2, which used ResNet-101 as its transfer learning model, slightly outperforming its counterpart. Remarkably, after optimization, a notable 15% increase in accuracy score was observed in the O-M2 model.

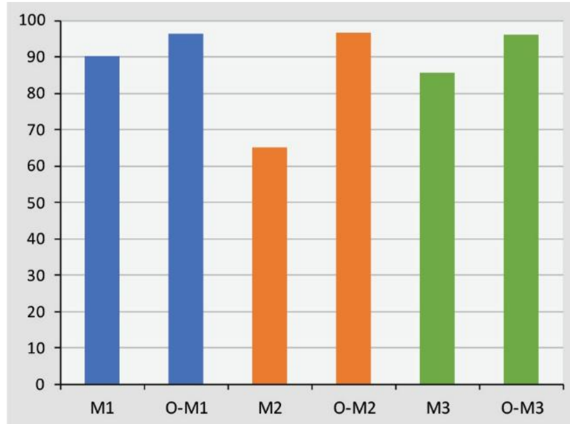

Fig 5. Accuracy (%) on German Traffic Signs

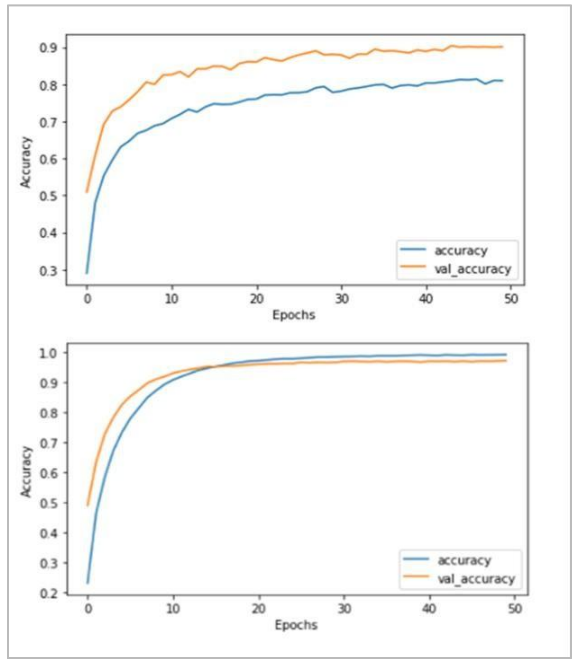

Fig 6. Learning curve of M2 (top) and O-M2 (bottom) on GTSRBD

Similarly, Results for before and after optimization for ITSD and CHTSD dataset are given in table 3 and table 4 respectively.

Table 4: Results on ITSD

Model	Accuracy	Precision	Recall	F1-score
M1	69.0438	0.7785	0.6904	0.7064
M2	68.8843	0.8008	0.6888	0.7095
M3	80.6714	0.8805	0.8067	0.8239
O-M1	88.1234	0.8964	0.8812	0.8811
O-M2	90.9547	0.9133	0.9095	0.9068
O-M3	88.5851	0.8969	0.8859	0.8846

When trained with default hyperparameters, the first model (M1), utilizing EfficientNet B2 as its transfer learning model, achieved the highest accuracy score among the three models, although it still lagged significantly behind the optimized models. However, all three models (O-M1, O-M2, O-M3) trained with hyperparameters obtained through Bayesian Optimization demonstrated good generalization performance on the testing data, yielding higher accuracy scores compared to their default counterparts. Notably, the model utilizing ResNet-101 as its transfer learning model (O-M2) slightly outperformed its counterpart. After optimization, a remarkable 32% increase in accuracy score was observed in the O-M2 model. Despite this, the limited availability of quality datasets and images in ITSD resulted in a relatively lower accuracy score for the models compared to other datasets.

Table 5: Results on CHTSD

Model	Accuracy	Precision	Recall	F1-score
M1	90.3030	0.9217	0.9030	0.9077
M2	65.1010	0.8404	0.6510	0.6906
M3	85.8586	0.9055	0.8586	0.8710
O-M1	96.5657	0.9701	0.9657	0.9668
O-M2	96.7677	0.9734	0.9677	0.9690
O-M3	96.2626	0.9685	0.9626	0.9639

The experimental results in table 4 demonstrate that, when trained with default hyperparameters, the first model (M1) utilizing VGG16 as its transfer learning model achieved the highest accuracy among the three models when evaluated on the Chinese traffic sign dataset, although it still fell far behind the optimized models. However, all three models (O-M1, OM2, O-M3) trained with hyperparameters obtained through Bayesian Optimization demonstrated excellent generalization capability on the testing data and produced impressive accuracy scores. Notably, the model leveraging ResNet-101 as its transfer learning model (O-M2) performed slightly better compared to its counterparts, achieving the highest accuracy score. The optimization process led to a significant increase of 31% in the accuracy score of the O-M2 model.

Table 6: Accuracy (%) Summary Table

	GTSRBD	CHTSD	ITSD
M1	88.75	90.30	69.04
M2	84.52	65.10	68.88
M3	90.84	85.86	80.67
O-M1	97.11	96.57	88.12
O-M2	97.65	96.77	90.95
O-M3	97.03	96.26	88.59

Table 5 provides a comprehensive summary of the experimental results in terms of accuracy percentage achieved by the models on each of the datasets. It presents a comparative analysis of the performance of the three models trained with default hyperparameters against the corresponding optimized models. The table shows that, for all datasets, the optimized models performed significantly better than the non-optimized models, with considerable increases in accuracy percentage. Specifically, the highest increase in accuracy percentage was observed in the optimized ResNet101-based model (O-M2) trained on the Chinese traffic sign dataset, with a remarkable 31% increase in accuracy percentage. These results provide strong evidence that finetuning hyperparameters through Bayesian Optimization can significantly enhance the performance of models on traffic sign recognition tasks.

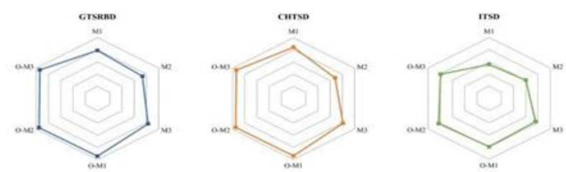

Fig 7. Precision (%) of Each Model on All Datasets

The radar visualization in Fig 7 presents the precision values for each model across the different datasets. The graph clearly demonstrates that the optimized models significantly outperformed their non-optimized counterparts in terms of precision percentage.

V. CONCLUSION

In this work, we have proposed a robust hybrid pipeline based on transfer learning and Bayesian Optimization Algorithm for traffic sign recognition. Signs from two different databases mainly, German Traffic Signs, and Indian Traffic Signs were used. Even though Pre-trained ResNet101 outperforms other models by a small margin after optimization on every benchmark dataset. The accuracy achieved by our transfer leaning models were close, and decent when compared to other SOTA models.

On benchmark datasets, system performance can be increased by employing regularisation techniques like data augmentation and dropout. The use of these strategies shortens training time, prevents overfitting issues, and improves the generalisation capability of models. Hyperparameter optimization in conjecture with transfer learning can lead to some great results in very short period with minimal computation power.

In future, transfer learning model can be further fine-tuned by unfreezing some of the blocks and other hyper parameters such as depth of neural networks can be considered while applying Bayesian optimization, which might extract even better performance in term of generalization from our model.

REFERENCES

[1] W. H. Organization, "Road traffic injuries," World Health Organization, 20 06 2022. [Online]. Available: https://www.who.int/news-room/factsheets/detail/road-traffic-injuries.

[2] N. a. L. P. a. L. V. B. Bhatt, "A Real-Time Traffic Sign Detection and Recognition System on Hybrid Dataset using CNN," *2022 7th International Conference on Communication and Electronics Systems (ICCES),* pp. 1354-1358, 2022.

[3] A. a. G. M. a. M. M. Loukmane, "A Model for Classification of Traffic Signs Using Improved Convolutional Neural Network and Image Enhancement," *2020 Fourth International Conference on Intelligent Computing in Data Sciences (ICDS),* pp. 1-8, 2020.

[4] S. K. S. S. K. K. A. Inderpreet Singh, "Dropout-VGG Based Convolutional Neural Network for Traffic Sign Categorization," *Congress on Intelligent Systems, Springer Nature Singapore,* pp. 247-261, 2022.

[5] C. N. S. S. Kiruthika Devi, "A Deep Learning-Based Residual Network Model for Traffic Sign Detection and Classification," *Ubiquitous Intelligent Systems, Springer Singapore,* pp. 71-83, 2022.

[6] B. a. K. M. R. a. D. R. Sanyal, "Traffic sign recognition on Indian database using wavelet descriptors and convolutional neural network ensemble," *Concurrency and Computation: Practice and Experience,* vol. 34, no. 10, p. e6827, 2022.

[7] Y. a. J. W. Zheng, "Evaluation of Vision Transformers for Traffic Sign Classification," *Wireless Communications and Mobile Computing,* vol. 2022, pp. 1-14, 2022.

[8] X. Z. S. R. J. S. Kaiming He, "Deep Residual Learning for Image Recognition," *CoRR,* vol. abs/1512.03385, p. 1512.03385, 2015.

[9] K. a. Z. A. Simonyan, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556,* 2014.

[10] Q. V. L. Mingxing Tan, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," *CoRR,* vol. abs/1905.11946, p. 1905.11946, 2019.

[11] "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural Networks,* 2012.