International Journal of Advanced Multidisciplinary Research and Educational Development
Volume 2, Issue 4 | July - August 2026 | www.ijamred.com
ISSN: **3107-6513**

Comparison of VGG-16 and InceptionV3 for Traffic Sign Detection and Classification

Fatemah AbuAleenain, Hams Rashedi, Hanan Elazhary

IST Department, College of Computer Science & Engineering,
University of Jeddah, Jeddah, Saudi Arabia

Abstract

In the future, autonomous vehicles need to detect and identify traffic signs, and figure out their locations. Traffic sign recognition systems may play a crucial role in self-driving cars, artificial driver assistance, traffic surveillance as well as traffic safety. Accordingly, traffic sign recognition is an important area of research that triggered many research studies. Nowadays, more and more object recognition tasks are solved using Convolutional Neural Networks (CNN). Due to their high recognition rate and fast execution, CNNs have enhanced most computer vision tasks, both those that are already in place and those that are new. Many researchers exploited CNNs (among other machine learning and deep learning models) for traffic sign recognition and experimented with numerous datasets. Nevertheless, this is still an open research area in which many other models and datasets can be explored. In this paper, we compare VGG-16 and InceptionV3 CNN models for this purpose and experiment using two different datasets. Experimental results showed that VGG-16 outperforms InceptionV3 with 97% accuracy on one dataset, but suffer from overfitting on the other. This will be handled in future work.

Keywords—Autonomous Vehicles; CNN; InceptionV3; Self-driving Cars; Traffic Sign Recognition; VGG-16

I. Introduction

The use of technology has become more prevalent among humans. In order to automate vehicles, multinational companies such as Uber, Ford, Audi, Toyota, Google, Tesla, Mercedes-Benz, and many others are enhancing their technology. They are trying to make more accurate autonomous or driverless self-driving vehicles, where the vehicle itself behaves like a driver and does not need human guidance to run on the road. One of the prominent components of these vehicles is Traffic-Sign Recognition (TSR) by which a vehicle is able to recognize the traffic signs on the road, e.g., "speed limit," "children," or "turn ahead." Traffic sign recognition is an important task for autonomous driving systems, as it enables the vehicle to understand and follow the rules of the road. It also plays a salient role in safety technology. Traffic signs can be analyzed using forward-facing cameras in many modern cars, vehicles, and trucks. Speed limits, stop signs, and "do not enter" signs can be often identified by TSR systems. However, traffic signs can vary in shape, color, size, orientation, illumination, and occlusion, making them challenging to recognize accurately.

Several researchers addressed this problem. They used many datasets such as Belgium, India, China and Japan traffic sign datasets, and experimented with several models such as AlexNet, VGG-16, GoogleNet and ResNet110. Nevertheless, this is still an open area of research in which more models and datasets can be explored. In this paper, we aim to compare the performance of two popular deep learning models for traffic sign recognition, namely VGG-16 and InceptionV3 using two datasets: GTSRB - German Traffic Sign Recognition Benchmark and Traffic Sign Dataset - Classification, both from Kaggle, a well-known and trusted platform. The combination of these models with these datasets has not been considered before to the best of our knowledge.

The paper is organized as follows: Section II discusses surveyed related work. Section III presents the dataset and discusses data preprocessing tasks. Section IV presents the models and the experimental setting. Section V provides and discusses the results, and finally, Section VI presents the conclusion and future work.

II. Related Work

Several researchers studied the detection and classification of traffic signs from images using machine learning and deep learning techniques. For example, Zhu and Yan [1] used two models, namely YOLOv5 and SSD to process a dataset formed of 2,182 traffic signs images classified into 8 classes. To evaluate the models, they used precision, recall, and mAP@0.5. In their experiments, YOLOv5 showed the best performance with 78.32% precision, 90% recall, and 97.7% mAP@0.5. Hasan et al. [2] used two models to classify traffic signs, namely SVM and CNN. The used dataset was built by collecting some random videos and cropping traffic sign areas to build a real dataset with a total 1200 images. To evaluate the models, they used accuracy in their experiments. CNN showed the best performance with 99.56% training accuracy and 96.40% validation accuracy. Mehta et al. [3] used two models to classify traffic signs namely CNN and SGD. The dataset they used was BTSD that contains 4575 training images and 2520 test images. They used accuracy and epochs to compare between the models. CNN showed the best performance with accuracy of 96.61% at 10 epochs. Koresh [4] used more than four models to classify traffic signs, namely SURF, SIFT, BRISK, CNN and Caps Net. The dataset was the Indian traffic sign dataset. They compared the models using accuracy. The best performance was that of Caps Net with accuracy of 98%.

ISSN: 3107-6513

TABLE I. RELATED WORK SUMMARY

Ref NO.	Dataset	Models	Metrics	Best Model
[1]	a real dataset of 2,182 images	YOLOv5, SSD	precision, recall, mAP@0.5	YOLOv5 with precision of 78.32%, recall of 90% and mAP@0.5 of 97.70%
[2]	a real dataset of 1200 images	SVM, CNN	Accuracy	CNN with accuracy of 96.40%
[3]	BTSD	CNN (Adam), SGD	accuracy, epochs	CNN with accuracy of 96.61% and epochs at 10.
[4]	The Indian traffic sign	SURF, SIFT, BRISK, CNN, Caps Net	Accuracy	Caps Net with accuracy of 98%
[5]	GTSRB and GTSDB	SVN, Modified GHT, ConvNet, Viola-Jones, HOG	accuracy, FPS	CNN with accuracy of 99.94% and FPS 50
[6]	GTSRB and BTSC	GoogleNet, ResNet10, VGG-16, AlexNet	Accuracy	ResNet10 with accuracy of 99.61% VGG-16 with accuracy of 92.64%
[7]	GTSRB, ITS and CTS	ViT, RealFormer, Sinkhorn Transformer, Nyströmformer, Transformer in Transformer, CNN	accuracy, epochs	CNN with accuracy of 97.73% and epochs at 20
[8]	GTSRB	YOLOv5, strongSORT	precision, recall, mAP@0.5, F1-measure	YOLOv5 with precision of 89%, recall of 91%, mAP@0.5 of 91% and F1-measure of 89%
[9]	GTSRB	Semi-supervised classification, Semi-supervised self-learning, CNN	accuracy, recall, F1-measure	Semi-supervised self-learning with accuracy of 99.46٪
[10]	GTSRB and GTSDB	Traditional CNN, Improved CNN	accuracy, caffe model size (KB), forward operation time (ms)	Improved CNN with accuracy of 98.1%, KB of 884.6 and ms of 692.34

One of the commonly used datasets is GTSRB, which is a large-scale dataset that contains 43 classes of traffic signs with more than 50,000 images in total. For example, Shustanov and Yakimov [5] used more than four models to classify traffic signs, namely SVN, Modified GHT, ConvNet, Viola-Jones and HOG. The datasets were GTSRB and GTSDB. They compared the models used accuracy and FPS. The best performing model was CNN with 99.94% of accuracy and 50 FPS. Zhang et al. [6] used four models to classify traffic signs, namely AlexNet, VGG16, GoogleNet, and ResNet10. The datasets were GTSRB and BTSC. They compared the models used accuracy. The best performing model was ResNet10 with accuracy of 99.61%, followed by VGG16 with accuracy of 92.64%. Zheng and Jiang [7] used more than five models to classify traffic signs, namely Vision Transformer model (ViT), RealFormer, Sinkhorn Transformer, Nyströmformer, Transformer in Transformer and CNN. They used GTSRB, the Indian Traffic signs (ITS), and Chinese Traffic Signs (CTS) datasets and compared the models used accuracy and the number of epochs. The best performing model was CNN, with accuracy of 97.73% at as high as 20 epochs.

Kim et al. [8] used two models to classify traffic signs, namely YOLOv5 and strongSORT. The dataset they used was GTSRB. They compared the models using precision, recall, mAp@.5 and F1-measure. The YOLOv5 model resulted in better performance than other models, with precision of 89%, recall of 91%, mAP@0.5 of 91% and F1-measure of 89%. He et al. [9] used three models to classify traffic signs, namely Semi-supervised classification, Semi-supervised self-learning, and CNN. The dataset was GTSRB. They compared the models used accuracy, recall and F1-measure. The best performing model was Semi-supervised self-learning model with accuracy of 99.46٪. Li et al. [10] used traditional CNN and improved CNN. The datasets were GTSRB and GTSDB. They compared the models using accuracy, caffe model size (KB) and forward operation time (ms). In both datasets the best performing model was the improved CNN with accuracy of 98.1%, KB of 884.6 and ms of 692.34.

Fig. 1. Sample of GTSRB Dataset

The above research studies are summarized in Table I. As shown in the table, many research studies used the GTSRB dataset, so we decided to use it in our research. We also decided to contribute by selecting a dataset that has not been used before for this task, to the best of our knowledge. Based on the results in the table, we decided to compare the performance of two popular deep learning models, namely VGG-16 and InceptionV3 using two datasets: GTSRB and Traffic Sign Dataset – Classification since the combination of

 ISSN: **3107-6513**

these models with these datasets for traffic sign recognition has not been considered before to the best of our knowledge.

Fig. 2. Sample of Traffic Sign Dataset – Classification Dataset

III. Datasets and Data Prepossessing

As previously noted, in this paper, we use two different traffic sign datasets, namely GTSRB [11] and Traffic Sign Dataset - Classification [12], both from Kaggle. These datasets have different characteristics and challenges that make them suitable for comparing and analyzing different models for traffic sign recognition and classification.

GTSRB is a large-scale dataset that contains 43 classes of traffic signs with more than 50,000 images in total. The images are taken from different angles, distances, lighting conditions and occlusions. The images are also cropped and resized to 32x32 pixels. The dataset is divided into a training set with 39,209 images and a test set with 12,630 images. It also provides additional information such as the bounding boxes, the region of interest and the coordinates of the traffic signs in the images. On the other hand the Traffic Sign Dataset - Classification is a smaller dataset that contains 11 classes of traffic signs with about 11,000 images in total. The images are taken from different countries and regions, such as Belgium, India, China, and Japan. The images are also cropped and resized to 64x64 pixels. The dataset is divided into a training set with 8,800 images and a test set with 2,200 images.

To preprocess the datasets and make them suitable as inputs to the models, we used the ImageDataGenerator class from the Keras library to apply data augmentation to images from both datasets. Data augmentation is a technique that applies random transformations to the images, such as rotation, scaling, shifting, shearing, flipping, and zooming to increase the diversity and size of the training data and reduce overfitting. Specifically, the pixel values of the images were rescaled from the range of 0-255 to the range of 0-1, and the images were rotated by up to 40 degrees, shifted horizontally and vertically by up to 20% of their width and height, sheared by up to 20%, zoomed by up to 20%, flipped horizontally, and their missing pixels were filled with the values of the nearest ones.

The data was also split into training and testing sets with a ratio of 80:20. We used a batch size of 32 for both datasets. Additionally, we resized the images to 299x299 pixels, which is the default input size for InceptionV3 and VGG-16 models. The number of channels was set to 3, which corresponds to the RGB color model. Categorical class mode was specified, which means that each image belongs to one of the classes in the dataset and is represented by a one-hot encoded vector.

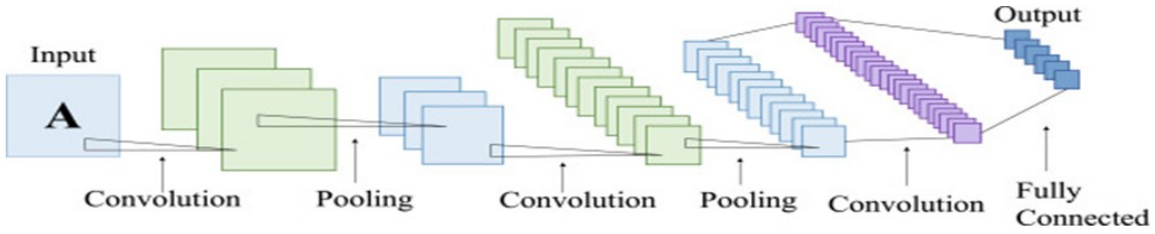

Fig. 3. InceptionV3 model architecture [14].

IV. Models and Experimental Details

In this section, we provide more details of the used models and the experimental setting.

A. InceptionV3 Model

The InceptionV3 architecture [13, 14] is a deep CNN that is designed for image classification tasks. As shown in Figure 3, this architecture consists of a series of convolutional layers, pooling layers, and fully-connected layers. It is based on the concept of "Inception modules," which are blocks of layers that perform parallel convolutions with different filter sizes and then concatenate their outputs. This allows the model to capture features at multiple scales and reduces the number of parameters compared to using only one filter size.

The InceptionV3 architecture consists of a total of 48 layers, including 3 Stem layers, 10 Inception modules, and 1 final fully-connected layer. A Stem layers consists of a 3x3 convolutional layer followed by a 2x2 pooling layer, while an Inception module consists of a combination of 1x1, 3x3, and 5x5 convolutional filters.

B. VGG-16 Model

As the name implies, as shown in Figure 4, VGG-16 [15] consists of 16 layers of CNNs. VGG-16 model uses CNNs for image recognition. Instead of using a large number of hyperparameters, it has 16 layers with weights. VGG-16 improves on AlexNet and replaces large filters with smaller 3×3 filters. AlexNet kernel size is 11 for the first convolutional layer and 5 for the second layer. VGG-16 has a simple and

ISSN: **3107-6513**

uniform architecture with multiple convolutional layers followed by max pooling layers and fully connected layers. It uses small filters (3x3) and large receptive fields (224x224) in order to be able to capture more features from the images.

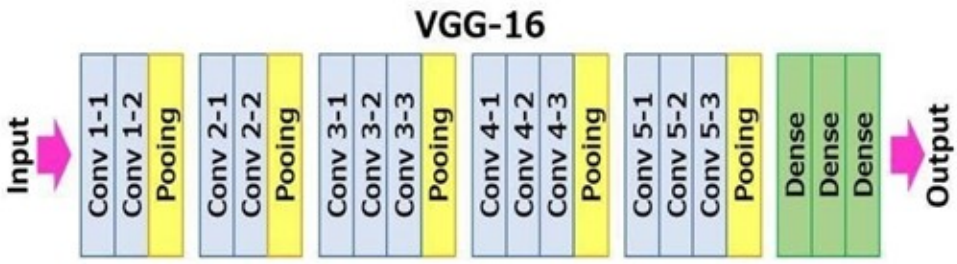

Fig. 4. VGG-16 model architecture [15].

C. *Experimental Setting*

We used the InceptionV3 and VGG-16 models from the Keras library as our base models for traffic sign recognition. Though we used the weights of these models as initial values for our models, we excluded the top layers, which are specific to the ImageNet classes. We then added our own custom layers on top of the base models to adapt them to our datasets. We also used a max pooling layer to reduce the dimensionality of the output of the base models, a batch normalization layer to normalize the activations and improve the training speed and stability, a dense layer with 1024 units and ReLU activation function to introduce nonlinearity and learn complex features.

To train the models, we froze the weights of the base models and only trained the custom layers, as we assume that the base models have already learned general features from the ImageNet dataset that are useful for our task. We used the Adam optimizer with a learning rate of 0.001 to update the weights of the custom layers and used the default batch size of 32. We also used the categorical cross entropy loss function to measure the difference between the predicted probabilities and the true labels. Finally, to evaluate the performance of the models, we used accuracy, precision and recall as metrics. Accuracy is the ratio of correctly predicted images to the total number of images. Precision is the ratio of correctly predicted images of a class to the total number of images predicted as that class, while recall is the ratio of correctly predicted images of a class to the total number of images belonging to that class.

We trained each model for 30 epochs using the *fit_generator* method, which feeds batches of augmented and preprocessed images and their labels from the *train_data* generator to the model. We also used the *val_data* generator to provide batches of images and labels for validation purposes and set the *steps_per_epoch* parameter to be equal to the number of training images divided by the batch size, and did the same for the *validation_steps* parameter. Finally, we set the verbose parameter to 1 to display the progress of each epoch.

It is worth noting that the hyperparameters of the models, such as learning rate, batch size, optimizer and regularization, can also affect the performance of the models. Thus, we used the same hyperparameters for all models to ensure a fair comparison. We trained each model for a fixed number of epochs (30 for InceptionV3 and VGG-16), but it is possible that some models could achieve better performance with more or less epochs depending on their learning curve and convergence speed, which we monitored.

V. Results and Discussion

In this section, we provide and discuss the result of experimenting with the two aforementioned datasets using InceptionV3 and VGG-16 CNN models.

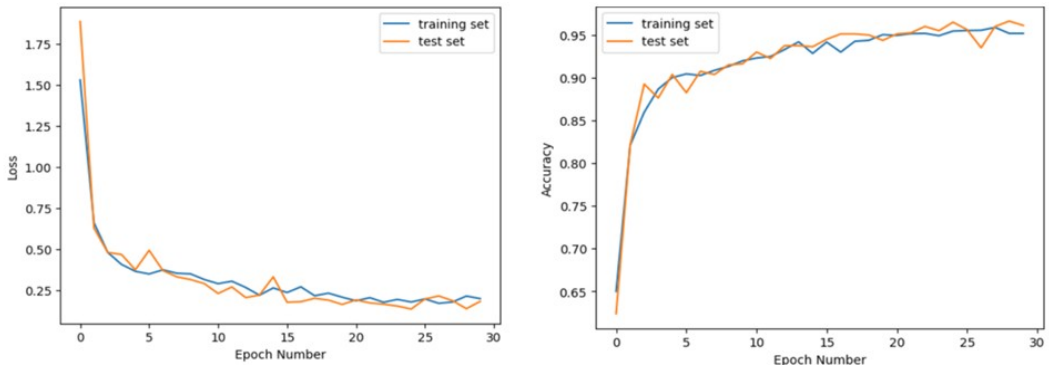

Fig. 5. InceptionV3 model Loss & Accuracy applied to Traffic Sign dataset - Classification.

ISSN: **3107-6513**

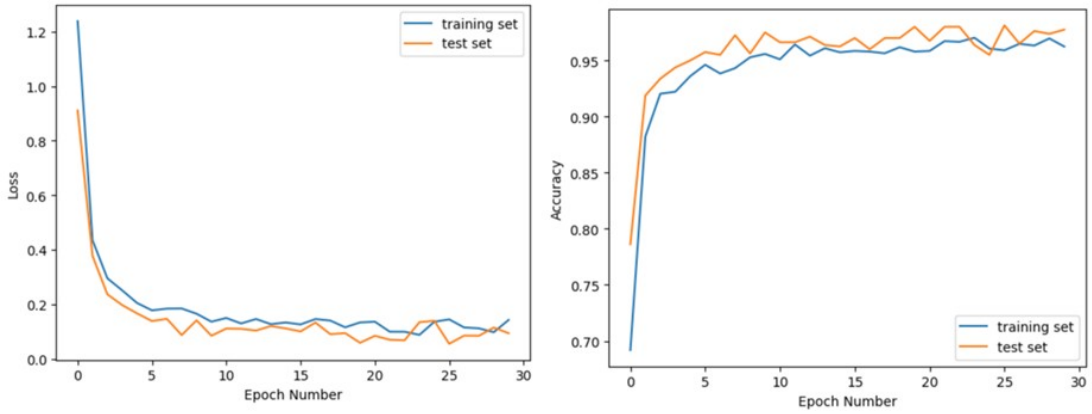

Fig. 6. VGG-16 model Loss & Accuracy Applied to Traffic Sign Dataset.

TABLE II. TRAFFIC SIGN DATASET - CLASSIFICATION RESULTS

Model	Epoch	Loss	Accuracy	Precision	Recall	F1- measure
InceptionV3	30	0.17	96	96	96	96
VGG16	30	0.08	97	98	97	97

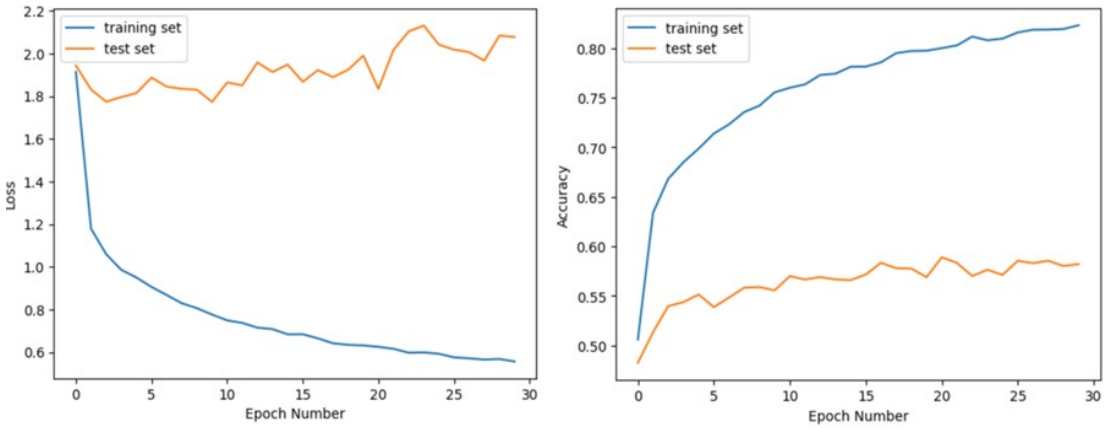

Fig. 7. InceptionV3 model Loss & Accuracy applied to GTSRB dataset

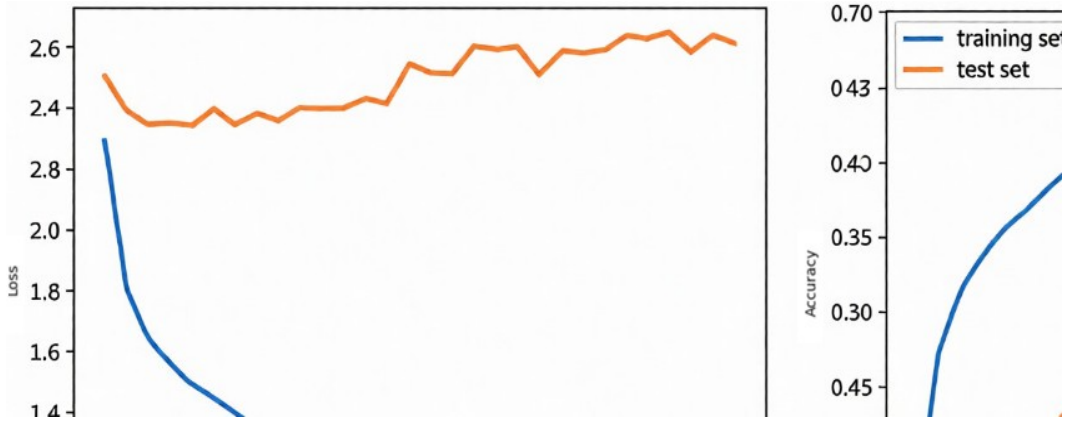

Fig. 8. VGG-16 model Loss & Accuracy applied to GTSRB dataset

ISSN: **3107-6513**

TABLE III. GTSRB - GERMAN TRAFFIC SIGN RECOGNITION BENCHMARK RESULTS

Model	Epoch	Loss	Accuracy	Precision	Recall	F1-measure
InceptionV3	30	2.11	57	64	54	58
VGG-16	30	2.63	44	59	35	44

A. Results of Traffic Sign Dataset - Classification

Fig. 5 shows the loss and accuracy learning curves of the InceptionV3 model on the Traffic Sign Dataset - Classification. As shown in the figure, after 30 epochs, the accuracy reaches nearly 96% and the loss reaches 0.17. On the other hand, Fig. 6, shows the loss and accuracy learning curves of the VGG-16 model on the same dataset. We can observe from the figure that at 30 epochs, the accuracy reaches nearly 97% and the loss reaches 0.08. This implies that VGG-16 performed better than InceptionV3 when applied to the Traffic Sign Dataset - Classification. The results of all metrics are summarized in Table II. It is clear that VGG-16 also performed better than InceptionV3 across all the additional computed metrics, namely precision, recall, and F-measure.

B. Results of GTSRB dataset

Fig. 7 shows the loss and accuracy learning curves of the InceptionV3 model on the GTSRB dataset. As the figure indicates, at 30 epochs, the accuracy reaches nearly 57% and the loss reaches 2.11. On the other hand, Fig. 8, shows the loss and accuracy learning curves of the VGG-16 model on the same dataset. As the figure indicates, at 30 epochs, the accuracy reaches only about 44% and the loss reaches 2.63.

This implies that InceptionV3 performed better than VGG-16 when applied to the GTSRB dataset. The results of all metrics are summarized in Table III. It is clear that InceptionV3 performed better across all metrics. Nevertheless, in both cases, the performance is unacceptable. It is clear that they both suffer from overfitting and so the results are unreliable and inconclusive. This is left for future work.

TABLE IV. HYPERPARAMETERS

Hyperparameter	**Value**
Epochs	30
Learning rate	0.001
Batch size	32
Optimizer	Adam

C. Discussion

Table IV shows the values of the hyperparameters used for both datasets on both models. Using those hyperparameters, VGG-16 performed slightly better than InceptionV3 on the Traffic Sign Dataset - Classification. On the other hand, when processing the GTSRB dataset, Inception V3 outperformed VGG16, though the results for both models were unacceptable due to overfitting as shown by examining the learning curves. Such a problem arises when the model is more complex than the dataset, so we need a simpler model. We can reduce the complexity of the model by using fewer neurons or layers, or we can use regularization techniques such as L1, L2, and dropout. This step will be performed in future work.

VI. CONCLUSION AND FUTURE WORK

In this paper, we compared the performance of two popular deep learning models for traffic sign recognition: InceptionV3 and VGG-16. We applied these models to two different datasets of traffic sign images: GTSRB - German Traffic Sign Recognition Benchmark and Traffic Sign Dataset - Classification, both from Kaggle. The models were evaluated based on their accuracy, precision, recall and F1-measure. Experimental results showed that VGG-16 outperformed InceptionV3 on the latter dataset with 97% accuracy. Nevertheless, when experimenting with GTSRB, the models suffered from overfitting and will be handled in future work.

REFERENCES

[1] Y. Zhu and W. Yan, “Traffic sign recognition based on deep learning,” *Multimedia Tools and Applications*, vol. 81, 2022, pp. 17779–17791. https://doi.org/10.1007/s11042-022-12163-0

[2] N. Hasan, T. Anzum, and N. Jahan, “Traffic Sign Recognition System (TSRS): SVM and convolutional neural network,” *Inventive Communication and Computational Technologies, Lecture Notes in Networks and System,* 2020, pp. 69–79. https://link.springer.com/chapter/10.1007/978-981-15-7345-3_6.

[3] S. Mehta, C. Paunwala, and B.Vaidya, “CNN-based traffic sign classification using Adam optimizer,” *International Conference on Intelligent Computing and Control Systems (ICCS)*, Madurai, India, 2019, pp. 1293-1298. https://ieeexplore.ieee.org/document/9065537.

[4] H. Koresh, "Computer vision based traffic sign sensing for smart transport", *Journal of Innovate Image Processing*, vol. 1, no. 1, 2019. https://irojournals.com/iroiip/article/view/597.

[5] A. Shustanov and P. Yakimov, “CNN design for real-time traffic sign recognition,” *Procedia Engineering*, vol. 201, 2017, pp. 718–725. https://www.sciencedirect.com/science/article/pii/S1877705817341231.

[6] J. Zhang, W. Wang, C. Lu, J. Wang, and A. Sangaiah, "Lightweight deep network for traffic sign classification," *Annals of telecommunication*, 2020, pp. 369–379. https://link.springer.com/article/10.1007/s12243-019-00731-9

[7] Y. Zheng and W. Jiang, “Evaluation of vision transformers for traffic sign classification,” *Wireless Communications and Mobile Computing*, 2022. https://doi.org/10.1155/2022/3041117.

[8] C. Kim, J. Park, Y. Park, W. Jung, and Y. Lim, “Deep learning-based real-time traffic sign recognition system for urban environments,” vol. 8, no. 2, 2023, p. 20. https://doi.org/10.3390/infrastructures8020020.

[9] Z. He et al., “Traffic sign recognition by combining global and local features based on semi-supervised classification,” *IET Intelligent Transport Systems*, 2019. https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/iet-its.2019.0409.

ISSN: **3107-6513**

[10] W. Li, D. Li, and S. Zeng, "Traffic sign recognition with a small convolutional neural network," *IOP Conference Series: Materials Science and Engineering*, vol. 688, 2019, https://doi.org/10.1088/1757-899x/688/4/044034.

[11] J. Stallkamp et al., "The German traffic sign recognition benchmark: A multi-class classification competition," *Proceedings of the IEEE International Joint Conference on Neural Networks*, pp. 1453-1460, 2011. https://ieeexplore.ieee.org/document/6033395

[12] A. Hemateja, "Traffic sign dataset - Classification," *Kaggle*, https://www.kaggle.com/datasets/ahemateja19bec1025/traffic-sign-dataset-classification.

.[13] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens. and Z. Wojna, "Rethinking the Inception architecture for computer vision," *arXiv.org*, 2015. https://arxiv.org/abs/1512.00567.

[14] N. Dong, L. Zhao, C. Wu, and J. Chang, "Inception V3 based cervical cell classification combined with artificially extracted features," *Applied Soft Computing*, vol. 93, 2020. https://doi.org/10.1016/j.asoc.2020.106311.

[15] Great Learning, "Everything you need to know about VGG16," 2021. https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918
.