Traffic Sign Detection Using Lightweight Deep Learning Model

ABRAR AHBABUL HAQUE, Brac University, Bangladesh
SAKIB UL HAQUE, Brac University, Bangladesh
NIBIR NEELIM, Brac University, Bangladesh
ADIB REZA, Brac University, Bangladesh

In this time of autonomous driving systems, accurate traffic light detection is essential for ensuring safety. This paper will investigate the performance of some of the leading lightweight deep learning architectures named MobileNet V3 and ResNet152 that have been customized for traffic light recognition contributing to the enhancement of autonomous driving systems. The aim of this research is to analyze the trade-offs between accuracy and computational complexity of these models to make it deployable on platforms that are resource constrained. These models were trained on a dataset containing a diverse range of traffic sign images under different lighting conditions. The models were fine tuned to optimize their performance in traffic sign detection while keeping the processing time to minimal. Our findings reveal that MobileNet V3 has outstanding performance despite simple architecture while ResNet152 performed poorly. This analysis serves as a benchmark and highlights the potential for using these lightweight models in practical applications ensuring both performance and efficiency.

ACM Reference Format:
Abrar Ahbabul Haque, Sakib Ul Haque, Nibir Neelim, and Adib Reza. 2018. Traffic Sign Detection Using Lightweight Deep Learning Model. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX).* ACM, New York, NY, USA, 5 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

The ever expanding field of autonomous navigation systems constantly demand increasing safety levels for smart systems that are employed to detect traffic signs as the safety of people's lives may depend on it. Here, this paper tries to bring attention on two leading lightweight deep learning models, namely MobileNet V3 and ResNet152, which have been adapted specifically to recognize traffic signs. There is a very big gap in literature when it comes to finding a balance between accuracy and processing demands in resource-constrained environments of self driving vehicles. Our research addresses this by comparing the two models under varying real world conditions. Our aim is to learn exactly how different models perform in terms of accuracy and computational complexity and discovering insights into improvements for autonomous navigation systems in the real world. This paper presents detailed experimental results and analyses of lightweight models in real life scenarios and use-cases, thus shedding light on their potential and their limitations.

Authors' Contact Information: Abrar Ahbabul Haque, Brac University, Dhaka, Bangladesh; Sakib Ul Haque, Brac University, Dhaka, Bangladesh; Nibir Neelim, Brac University, Dhaka, Bangladesh; Adib Reza, Brac University, Dhaka, Bangladesh.

Manuscript submitted to ACM

2 BACKGROUND

In today's societies, electric vehicles have taken the markets by storm, creating interest in all sorts of computerized autonomous driving systems. To accomplish this however, many sorts of techniques have been proposed and used over the years, all to identify traffic lights and their states in different lighting conditions and distances. Firstly, for the longest time, researchers have been trying out computer vision based approaches to solving this massive problem. Haltakov et al. used such a method by semantically separating day and night TLs but it had low robustness [1]. Du et al. used a system based on prior knowledge of the TLs, but it had the problem of feasibility, scalability and costs needed for improving its capabilities [2].

Then researchers looked into machine learning based methods to solve the problem. Vitas et al. brought forth a machine learning approach with dataset image augmentation [3] while Weber et al. presented his hierarchical traffic light detection algorithm [4]. Then Saini et al. presented a computer-vision based TL structure detection algorithm that worked in tandem with the convolutional neural network (CNN)-based state information extraction system [5]. Then Vitas et al. brought forth a method that used adaptive thresholding and deep learning for the tasks of region proposal and traffic light localization respectively [6]. Kulkarni et al. used a deep learning approach using transfer learning to determine the traffic light recognition and detection in Indian traffic scenarios [7]. Wang et al. gives a novel LL4PH - Net framework and enhances images using dark channel prior knowledge to propose an accurate method for traffic objects in low light . It outperforms existing methods and has potential applications in traffic monitoring and autonomous driving [8]. Thus researchers were leaning into using a combination of different techniques instead of relying on any one of them for better accuracy in real world scenarios.

3 METHOD

3.1 Dataset

The dataset used in this paper consists of preprocessed traffic sign data in pickle format. The training dataset was balanced in all of its 43 classes of traffic signs through augmentation that transformed the original images with the adjustment of brightness and rotation. This would enable the model to be generalized across diverse traffic signs with training samples of up to 86,989. German Traffic Sign Recognition Benchmark (GTSRB) was the source of the initial data of this dataset. Each image in the dataset is a 32x32 pixel with 3 color channels (RGB). The preprocessing enabled the images in the dataset to be available for robust training and testing.

3.2 Preprocessing

Both models had a batch size of 64 which means it processed 64 images during each iteration where it computed loss and updated the parameters for enhancing generalization of the model. Each image in both models is then resized to 224 pixels by 224 pixels to ensure consistent data with the same dimension is received in the input layer. The model will use the training data to learn and adjust its weight accordingly. Validation data is then used to analyze the performance of the model during the training so that the hyperparameters can be tuned to avoid overfitting where the model has good performance on the training data but performs badly on the unseen data. Finally, the test data will determine the overall final performance of the model for its effectiveness in real life scenarios. For each of the red, green and blue channels, the pixel values of all the images ranging from 0 to 255 were rescaled in the range between 0 and 1 to speed up the learning process. The order of the images are shuffled so that the model does learn any order-dependent patterns from the training data.

3.3 MobileNet V3

MobileNet v3 Large model architecture is used that is highly optimized for its performance in mobile devices for its reduced size and complexity. The model was also initialized with a parameter that uses the pre-trained weights on the ImageNet dataset so that model is given knowledge about the weights that have already been trained in a large and diverse dataset. The MobileNet model is customized by adding several layers that would transform its feature map from output into the predictions of the 43 classes of traffic signs. This output is then passed through a fully connected Dense layer with 512 units to learn non-linear transformation of the features where the RELU activation function was used. The inputs were standardized within each mini-batch through Batch Normalization to help reduce the number of training epochs required for training. It also normalizes the output from each previous layer by the subtraction of batchmean and division of batch standard deviation. Lastly, another Dense layer is added with 43 units equaling the number of classes as the output layer which uses softmax activation function so that the output can be represented as probabilities. Adam optimizer is used in this model that maintains distinct learning rates based on the weights updated. Categorical cross entropy is used as the loss function as the model has to deal with categorical classification tasks of 43 classes where the labels are one-hot encoded. This loss function measures the difference between the predictions and true labels which will guide the network to make more accurate predictions. The learning rate is set at 0.001 for epochs less than 5 and it is decreased to 0.0005 for epochs between 5 and 10 and it is further reduced to 0.0001 for epochs greater than 10. This adjustment of learning rates enables the model to converge faster in the initial phases with higher learning rate and fine-tune better in the later stages with lower learning rate. Another model is used where only the weights of batch normalization layers will not be updated during training to maintain the statistics learned from the original training data.

3.4 ResNet-152

Resnet-152 is a deep convoluted neural network used in the classification of images. In this model we used the weights pre-trained on the ImageNet-1K dataset version 2 in which the NVIDIA GeForce GTX 1070 is used as the GPU for faster deep learning tasks. Each channel of the image is normalized by using the same mean and standard deviation values as pre-trained on ImageNet images to adjust the input data having zero mean and unit variance. Data loaders are used in training,validation and test data for efficient and faster loading of data for feeding into the neural network. The use of multiple workers enables the data loader to parallelize data loading which results in much faster processing time during training. However it also increases memory there for the number of workers is set to 4 for optimal solution. The parameters are freezed which ensures that they do not get updated by optimizer during training so that only additional layers are trained for fixed feature extraction. It also helps in significantly reducing the computational time as the entire network is not trained. The ResNet model is customized by having additional fully connected layers with 512 output features and a rectified linear unit (ReLU) activation function. The final output layer has 512 input features and 43 output features where log softmax activation is used to provide log probabilities for the classification. Batch normalization is applied to accelerate the convergence. After all these modifications, the model is moved to the appropriate device. The cross entropy loss function and Adam optimizer is used with a learning rate of 0.001 to update the parameters of the fully connected layers. Finally, the model was trained for 20 epochs where on each epoch the entire dataset is trained once. The training part was divided into two phases. In the first phase, only the custom final classification layers are trained from epochs 1 to 10 where the parameters of the base model are frozen to prevent them

from being updated during backpropagation as the initial layers have already learned useful representations. In the second phase after epoch 10, the layers of the base model are unfreezed except the batch normalization layers.

4 RESULTS

4.1 MobileNet V3

The two different models of MobileNet V3 were trained where the training accuracy of the first model is 0.9994 by epoch 10. However, the validation accuracy is significantly lower and very unstable, varying dramatically across epochs which suggests a severe overfitting issue. The validation loss in the first model shows extreme values of 1466.7863 in epoch 7 which indicates that the model performance on validation data is very poor. As for the second model, both the training and validation accuracies are very high and more stable compared to the first model. The training accuracy reaches 0.9999 by Epoch 10, while the validation accuracy peaks at 0.9968 in the last epoch. The validation loss is also much more stable and remains relatively low throughout the training, indicating that this model is generalizing much better than the first one. Finally, the model has performed quite well on the test set with a test accuracy of approximately 98.53% which suggests that the model is effectively generalizing to unseen data. From the classification report, it is observed that the model was very effective at classifying a wide range of classes while some classes had low performance in which its features are not captured well by the model. The macro average value was 0.97 which treats all classes equally, averaging the metric for each class without considering the number of instances in each class. The weighted average is 0.99 which considers the number of true instances for each class, indicating that the model's performance is generally excellent across classes that have more instances.

4.2 ResNet-152

From our experiment, it is observed that the training loss of the ResNet-152 model decreased from 1.0815 in the first epoch to 0.1642 in the twentieth epoch which shows that the model is increasingly learning from the training data at each epoch. The training accuracy is observed to have increased from 67.30% in the first epoch to 94.45% in the twentieth which shows that the model is performing better at each epoch in the correct classification of the training data. However, the validation accuracy does not increase significantly like training accuracy and almost stays flat where it starts at 60.79% and ends at 61.95%. This accuracy is calculated using the validation dataset where the model has not seen those data during training. Finally, after evaluating on the test set, the model achieved an accuracy of 61.11% which indicates poor performance of the model in predicting unseen data.

5 DISCUSSION

The MobileNetv3 model showed exceptional training behavior with good generalization on the validation set and high test accuracy which makes it ready for deployment on real-world data. Its performance should be monitored to ensure it maintains high accuracy outside of the test environment. The model should be tested on more challenging scenarios that are outside of the test set which would enable us to understand the limits of the model's capabilities. Besides, early stopping can be applied to prevent unnecessary training if the model consistently performs well on the validation data before the last epoch.

As for the ResNet-152 model, the validation accuracy is almost similar to the accuracy on the test dataset which suggests that the model's performance on unseen data is consistent but the low accuracy on both validation and test set compared to the high accuracy on the training dataset suggests that the model is overfitting. It occurs when the model

learns the training data so well that it even learns the noise and cannot generalize to unseen data. To address this issue of overfitting, some strategies can be followed to enhance the model's ability to not just fit on the training data but also perform well on the new and unseen data:

(1) Regularization techniques can be used such as dropouts and L2 regularization
(2) Increasing the diversity of the training data through data augmentation
(3) Simplifying the model in case if is too complex for the training data through the tuning of hyperparameters.

6 CONCLUSION

This research has provided a detailed view into the strengths and weaknesses of both the models MobileNet V3 and ResNet152, in terms of their ability to detect traffic signs for self-driving vehicle systems. On one hand, MobileNet V3 showed superior performance and remarkable efficiency, while on the other, ResNet152 showed limitations in generalizing unseen data. These results demonstrate why choosing the correct model based on specific operational and resource availability conditions is crucial in the real world. In the future, more could be explored regarding the optimization of these models, potentially incorporating real-time adaptive learning algorithms which will then improve accuracy and robustness under varied operating conditions. Moreover, including more challenging and diversified scenarios in the dataset may help make models that are more resilient to environmental changes and more reliable in many different use-cases in real life. This research paves the way for the development of safer and more reliable autonomous vehicles based on targeted and efficient usage of deep learning in this field.

REFERENCES

[1] Haltakov, V.; Mayr, J.; Unger, C.; Ilic, S. Semantic Segmentation Based Traffic Light Detection at Day and at Night. In Proceedings of the Pattern Recognition, Lecture Notes in Computer Science, Vol. 9358, Springer, 2015; pp. 446–457.
[2] Du, X.-P.; Xiong, H.; Li, X.-F. Traffic Light Recognition Based on Prior Knowledge and Optimized Threshold Segmentation. *J. Comput.* **2017**, *28*(2), 197–205.
[3] Vitas, D.; Tomic, M.; Burul, M. Image Augmentation Techniques for Cascade Model Training. In Proceedings of the Zooming Innovations in Consumer Technologies Conference, May 2018; pp. 78–83.
[4] Weber, M.; Huber, M.; Zollner, J.M. HDTLR: A CNN Based Hierarchical Detector for Traffic Lights. In Proceedings of the 21st International Conference on Intelligent Transportation Systems, 2018; pp. 255–260.
[5] Saini, S.; Nikhil, S.; Konda, K.R.; Bharadwaj, H.S.; Ganeshan, N. An Efficient Vision-Based Traffic Light Detection and State Recognition for Autonomous Vehicles. In Proceedings of the IEEE Intelligent Vehicle Symposium, June 2017; pp. 606–611.
[6] Vitas, D.; Tomic, M.; Burul, M. Traffic Light Detection in Autonomous Driving Systems. *IEEE Consumer Electronics Magazine* **2020**, *9*(4), 90–96, doi: 10.1109/MCE.2020.2969156.
[7] Kulkarni, R.; Dhavalikar, S.; Bangar, S. Traffic Light Detection and Recognition for Self Driving Cars Using Deep Learning. 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 2018; pp. 1–4, doi: 10.1109/ICCUBEA.2018.8697819.
[8] Wang, X.; Wang, D.; Li, S.; Li, S.; Zeng, P.; Liang, X. Low-light Traffic Objects Detection for Automated Vehicles. 2022 6th CAA International Conference on Vehicular Control and Intelligence (CVCI), Nanjing, China, 2022; pp. 1–5, doi: 10.1109/CVCI56766.2022.9964586.