Title of Your Project

Md Mahbubur Rahman Tusher (200101100)*, Susmita Roy Rinky (200101081)*,

Muhiminul Islam (200101074)*

Course Code: CSE 4132 Course Title: Artificial Neural Networks and Fuzzy Systems Sessional
Semester: Winter 2023
*Department of Computer Science and Engineering, Bangladesh Army University of Science and Technology (BAUST)

Abstract

As they play a significant part in autonomous driving and traffic safety, traffic sign identification and recognition have recently become one of the most significant areas of computer vision and image processing. Early studies in this field offered several deep-learning based methods for classifying distinct traffic signs using various standard datasets. However, not many researchers focused on creating a dataset of traffic signs in Bangladesh and applying deep learning techniques to recognize them. We compare and contrast several deep learning models for recognizing traffic signs from the perspective of Bangladesh. We construct a novel dataset with over 2000 images representing thirteen distinct kinds of typical traffic signs in Bangladesh. Using data augmentation, about 8386 images are generated from the original dataset. subsequently, transfer learning and fine-tuning approaches are applied to nine different deep-learning models using this dataset, and the outcomes are compared. Results indicate that ViT had a validation accuracy of 99.91% for fine-tuning, while DenseNet201 had a validation accuracy of 99.86% for transfer learning. Almost all models attained excellent training and validation accuracy levels, showing that they were able to successfully learn the dataset's characteristics.

1. Introduction

In computer vision and intelligent systems, traffic sign analysis is a crucial subject [1, 2]. Traffic signs provide essential information to drivers and pedestrians, such as speed limits, traffic rules, warnings about road hazards, and so on. The purpose of these signs is to warn drivers of the current condition of the road and other essential information. They have unbending, straightforward forms with vibrant colors that are simple to interpret. Nonetheless, when vehicles fail to see a traffic sign in time, accidents may still happen. Designing an automatic system that can recognize and understand traffic signs is therefore crucial. Traffic sign recognition has a wide range of applications, from improving driver assistance systems and autonomous vehicles to enhancing road safety and traffic management. It is a rapidly evolving field, with ongoing research and development focused on improving the accuracy, speed, and efficiency of traffic sign recognition systems.

2. Literature Review

Deep-learning models trained on large-scale datasets have recently shown promising results in improving the accuracy of computer vision tasks. Several pre-trained models, such as Xception [3], Vision Transformer (ViT) [4], ResNet101 [5], MobileNetV2 [6], VGG19 [7], InceptionV3 [8], DenseNet201 [9], EfficientNetB2 [10], NASNetLarge [11] and others, were used in such tasks. Most of these models rely on quality datasets to reach high levels of accuracy.

A substantial number of works in the field of traffic sign recognition have been published in recent years. Many research studies [12–15] offer clarification based on the most widely used datasets, including German Traffic Sign Detection Benchmark (GTSDB) and GTSRB. While many other research employed other datasets, including the Swedish Traffic Sign Dataset (STSD) [16], the Belgium Traffic Signs(BTS) [17], the Laboratory for Intelligent and the Safe Automobiles (LISA) [18], the Tsinghua-Tencent 100K dataset [19–21], so on. Many of them used deep learning-based techniques on the aforementioned datasets for the identification of traffic signs.

Studies on transfer learning-based traffic sign recognition have recently been published with promising results. To emphasize the features of traffic signs and increase the accuracy of traffic sign recognition, [22] presented a cascaded RCNN to acquire the multiscale features in pyramids and a multiscale attention approach to obtain the weighted multiscale features by dot product and softmax. Two novel lightweight networks were suggested by the authors in [23], which can achieve improved recognition precision using GTSRB and BTSC datasets while keeping fewer trainable model parameters. They employed a method called knowledge distillation, in which the knowledge from a larger model known as the teacher network was transferred to a smaller model known as the student network. The authors of [24] used the most advanced Mask R-CNN-based deep learning technology to solve the problem of identifying and recognizing a large number of traffic-sign types suited for automating traffic-sign inventory management. They made a number of suggestions for improvement, which were tested on a brand-new dataset with 200 kinds of traffic signs and led to better performance all around. The amount of training data and computational costs were greatly reduced by the researchers in [25] who employed a transfer learning based strategy for classifying traffic signs using the Inception-v3 model. Using a transfer learning-based method, the model was repeatedly retrained on the BTS dataset with fine-tuning parameters at various learning rates, and this resulted in a high-level recognition performance for traffic sign identification. In order to boost the system's ability to find the region of small traffic signs, the authors of [26] integrated the revised Faster-RCNN architecture with Online Hard Examples Mining (OHEM) after extracting the features of small traffic signs using a small region proposal generator. The experimental results increased by mAP of 12.1% over the initial object detection system after thorough testing and empirical assessments on a variety of video materials. Support Vector Machine and a Histogram of Oriented Gradient were used in the method described in [27] to present a novel automated methodology for the identification and recognition of Bangladeshi traffic signs from the video frames. After being applied to 78 videos of Bangladeshi traffic signs, which include six distinct types, this system obtained 96.15% accuracy using image processing techniques including binarization, contour identification, and determining similarity to circle, among others.

In this work, we used our owned dataset which represents the Bangladesh traffic signs, and apply deep learning models including Xception, DenseNet201, VGG19, InceptionV3, MobileNetV2, EfficientNetB2, Vision Transformer (ViT), ResNet101, NASNetLarge, etc to recognize them accurately. The aim of this research work can be outlined as follows: 1) Build a unique and diverse Bangladesh traffic sign dataset that comprises 2000 images of 13 different categories, 2) Conduct a comparative performance analysis of different deep learning models including Xception, DenseNet201, VGG19, InceptionV3, MobileNetV2, EfficientNetB2, Vision Transformer (ViT), ResNet101 and NASNetLarge on this dataset to find which model performs better.

3. Dataset Description

We gathered a collection of over 2000 images [28] representing 13 different types of roadside traffic signs in Bangladesh including College in front, Pedestrian crossing, Crossroad, Left turn, Right turn, Market in front, Mosque in front, Rail crossing, School in front, Speed breaker, Speed limit, Side road left and Side road right for the purpose of this research. Three separate smartphones OnePlus Nord10, Foco X4 Pro, and Realme X2 were used to take the pictures to increase the dataset diversity and ensure the robustness of our models. We build 3 datasets using some different data augmentation techniques. To enhance dataset-1 size of 8386 images, dataset-2 size of 8785 images, and dataset-3 size of 9212 images a variety of data augmentation techniques such as scaling, rotating, converting to grayscale, canny, zoom, and blurring are performed on the obtained dataset. Fig. 1 presents a few sample images of our dataset. Table 1 gives a summary of our dataset.

Table 1. Overview of Bangladesh Traffic Sign dataset.

Dataset-1		Dataset-2		Dataset-3	
Class Name	No. of images	Class Name	No. of images	Class Name	No. of images
College in front	657	College in front	691	College in front	720
Pedestrian crossing	654	Pedestrian crossing	687	Pedestrian crossing	720
Crossroad	658	Crossroad	682	Crossroad	715
Left turn	658	Left turn	694	Left turn	700
Right turn	636	Right turn	692	Right turn	715
Market in front	658	Market in front	693	Market in front	710
Mosque in front	653	Mosque in front	699	Mosque in front	715
Rail crossing	647	Rail crossing	690	Rail crossing	720
School in front	644	School in front	683	School in front	720
Speed breaker	653	Speed breaker	689	Speed breaker	710
Speed limit	652	Speed limit	696	Speed limit	700
Side road left	656	Side road left	682	Side road left	705
Side road right	660	Side road right	692	Side road right	700

Fig. 1. A few sample images from Bangladesh Traffic Sign dataset.

4. Methodology

Describe Deep learning methods for image classification applications frequently employ transfer learning and fine-tuning. Transfer learning is the process of freezing layers from a previously trained model on a big dataset such as ImageNet, adding new trainable layers on top of the frozen levels, then training the new layers on a smaller dataset like Bangladesh Traffic Sign dataset. On the other hand, fine-tuning entails unfreezing the entire model built through pre-training on a large dataset like ImageNet (or a portion of it), adding new trainable layers on top, and then retraining the entire model on a new dataset like our dataset in order to make significant improvements by gradually adapting the pre-trained features to the new data. Baseline training refers to the process of training a machine learning model from scratch, without using any pre-trained weights or transfer learning. In other words, the model is initialized with random weights and is trained on a specific task or dataset from scratch. In this work, we applied the base line training, transfer learning and fine-tuning method to exploit the characteristics of the pre-trained models from the ImageNet dataset for our classification job of Bangladesh Traffic Signs.

All experiments are conducted using TensorFlow version 2.11.0. The entire code is written and executed on the Google Colab platform. The hardware specification for the experiments is as follows: Intel(R) Core(TM) i5-8265U CPU running at 1.60 GHz to 1.80 GHz, 8.00 GB of RAM, and GPU Google Colab (12 GB of GDDR5 memory and NVIDIA Tesla K80 GPU). We complete our work in two step.

Step-1:

In this step, we aimed to apply three pre-trained models - Xception, DenseNet201, and InceptionV3 - using three different training techniques: baseline training, fine-tuning, and transfer learning. To begin with, we divided the dataset into three sets: training and validation and testing. In contrast to the training set, which contains 70% of the entire number of images, the validation set comprises 20% of the total number of images and the test dataset contains 10% of the total number of image. Then we remove the top layer of the pre-trained models and replace it with an MLP which consists of a global average pooling layer, a dropout layer, a fully connected layer, and a softmax layer. The softmax layer is made up of thirteen neurons representing our thirteen output classes. For transfer learning, we freeze all the weights of a pre-trained base model while the weights of newly added MLP are left unfrozen for updating. However, for fine-tuning, the weights of the whole network are left unfrozen for training. In Baseline training the model is initialized with random weights and is trained on the dataset from scratch. Figure 2 depicts the baseline training process of a pre-trained model during the experimentation of our work.

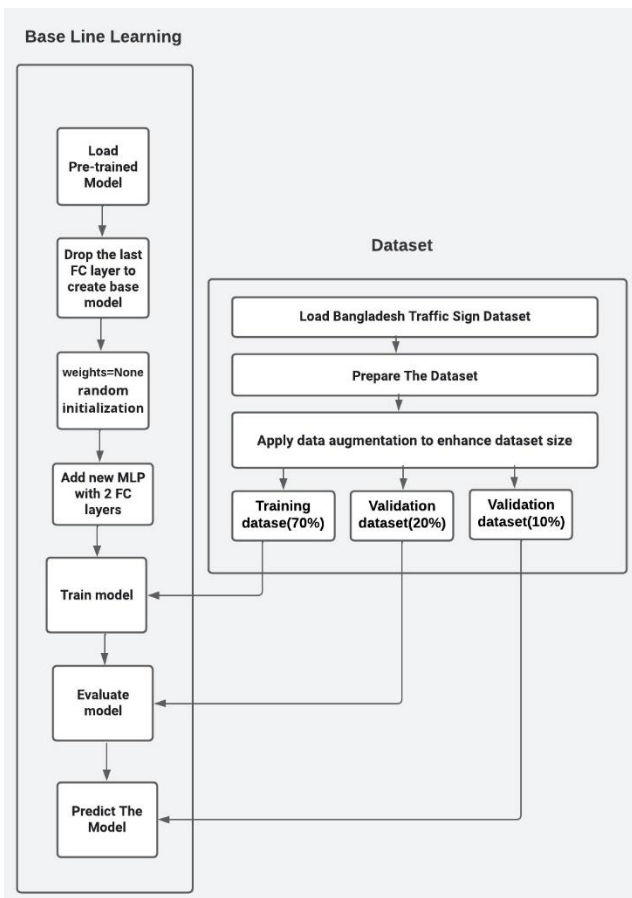

Fig. 2. Training and validation process of a pre-trained model of Bangladesh Traffic sign recognition system with baseline training.

We utilize an image size of (224, 224) and a batch size of 32. We implement random flip, random rotation, random translation, random zoom, random contrast and random brightness while using the Sequential model from Keras to enhance the data.

Additionally, we include a rescaling layer since the pre-trained weights require the input to be scaled from a range of (0, 255) to a range of (-1, 1). Sparse categorical cross-entropy is used as the loss function during training, and the Adam optimizer is used as the optimizer. The learning rate for transfer learning is set to 1e-3. In the result analysis section we analysis the result.

Step-2:

All nine pre-trained models, including Xception, DenseNet201, VGG19, InceptionV3, MobileNetV2, EfficientNetB2, Vision Transformer (ViT), ResNet101, and NASNetLarge, were trained using both transfer learning and fine-tuning strategies on our own dataset-1 described in the preceding section. To begin with, we divided the dataset into two sets: training and validation. In contrast to the training set, which contains 80% of the entire number of images, the validation set comprises 20% of the total number of images. Then we remove the top layer of the pre-trained models and replace it with an MLP which consists of a global average pooling layer, a dropout layer, a fully connected layer, and a softmax layer. The softmax layer is made up of thirteen neurons representing our thirteen output classes. For transfer learning, we freeze all the weights of a pre-trained base model while the weights of newly added MLP are left unfrozen for updating. However, for fine-tuning, the weights of the whole network are left unfrozen for training. Figure 3 depicts the entire training, validation, and testing process of a pre-trained model during the experimentation of our work.

Dataset-1

Dataset-2

Dataset-3

Fig. 4. Sample images after applying data augmentation.

We utilize an image size of (224, 224) and a batch size of 32. We implement random flip and random rotation while using the Sequential model from Keras to enhance the data.

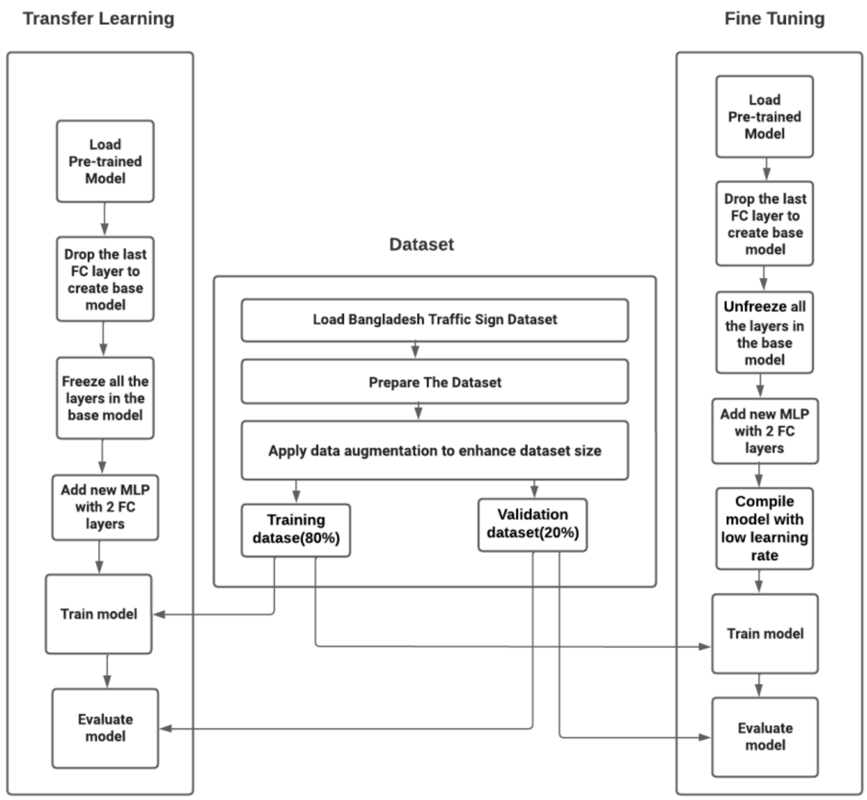

Fig. 3. Training and validation process of a pre-trained model of Bangladesh Traffic sign recognition system with transfer learning and fine-tuning.

Additionally, we include a rescaling layer since the pre-trained weights require the input to be scaled from a range of (0, 255) to a range of (-1, 1). Sparse categorical cross-entropy is used as the loss function during training, and the Adam optimizer is used as the optimizer. The learning rate for transfer learning is set to 1e-3, while the learning rate for fine-tuning is set at 1e-5. For the first moment, the exponential decay rate is set to 0.9, and for the second moment, it is set to 0.999. No weight decay is applied. To determine the final outcomes, the five-fold cross-validation method is employed, and the results are averaged. We apply the early stopping strategy to halt training if the validation loss measure has not improved after five epochs.

5. Result and Analysis

Result analysis for step-1:

In this step, we evaluated the performance of three pre-trained models - Xception, DenseNet201, and InceptionV3 - using three different training techniques: baseline training, fine-tuning, and transfer learning.

The results showed that the fine-tuning technique performed the best for all three models, with higher accuracy scores on the validation and test sets compared to the other two techniques. For example, the Xception model achieved a test accuracy of 99.03% using fine-tuning, compared to 98.85% using transfer learning and 96.45% using baseline training. Figure 6 depicts the comparative analysis of the training, validation and test performance of all three models using transfer learning, base line training and fine tuning in terms of accuracy.

Fig. 6. Comparison among training, validation and testing accuracy of all nine models using transfer learning, base line training and fine tuning.

Furthermore, the DenseNet201 model achieved the highest validation accuracy score of 98.04% using fine-tuning, compared to 97.51% using baseline training and 90.99% using transfer learning. Our findings for baseline training, transfer learning and fine-tuning are depicted in Table 2, Table 3 and Table 4 respectively.

Table 2. Comparison among training, validation and tasting accuracy of all three models using baseline training

Model name	Training accuracy	Validation accuracy	Test accuracy
Xception	0.9598	0.9471	0.9645
DenseNet201	0.9641	0.9751	0.9726
InceptionV3	0.9529	0.9512	0.9632

Interestingly, the InceptionV3 model achieved the highest training accuracy score of 99.88%, but the fine-tuning technique still outperformed the other two techniques on the validation and test sets.

Table 3. Comparison among training, validation and tasting accuracy of all three models using transfer learning.

Model name	Training accuracy	Validation accuracy	Test accuracy
Xception	0.8925	0.8649	0.8634
DenseNet201	0.9099	0.9099	0.9142
InceptionV3	0.9529	0.9512	0.9632

Overall, these results suggest that fine-tuning is a more effective technique for improving the performance of pre-trained models on image classification tasks, compared to baseline training and transfer learning. The results also highlight the importance of selecting the appropriate pre-trained model and training technique for a specific task to achieve the best performance.

Table 4. Comparison among training, validation and tasting accuracy of all three models using fine tuning

Model name	Training accuracy	Validation accuracy	Test accuracy
Xception	0.9814	0.9884	0.9903
DenseNet201	0.9685	0.9804	0.9785
InceptionV3	0.9988	0.9964	0.99953

Result analysis for step-2:

During the experiment, each of the nine pre-trained models was run on our build dataset separately using the transfer learning and fine-tuning techniques, and the training and validation outcomes were meticulously documented. Our findings for transfer learning and fine-tuning are depicted in Table 5 and Table 6 respectively.

Table 5. Comparison among training and validation accuracy of all nine models using transfer learning.

Model name	Training accuracy	Validation accuracy
ViT	0.9754	0.9792
VGG19	0.9337	0.9706
NASNetLarge	0.9622	0.9457
Xception	0.961	0.9567
DenseNet201	0.9747	0.9876
InceptionV3	0.9368	0.9455
EfficientNetB2	0.4473	0.4935
ResNet101	0.7705	0.7838
MobileNetV2	0.9737	0.9675

In the case of transfer learning, we discovered that DenseNet201 was the most accurate model, with a validation accuracy of 98.76%. The ViT model worked admirably as well, with a validation accuracy of 97.92%. The generalization performance of the VGG19 and MobileNetV2 was quite comparable to that of ViT. On the other hand, InceptionV3, Xception, and NasNetLarge exhibited a generalization performance of around 95% in terms of accuracy. However, EfficientNetB2 did extremely poorly, with the lowest validation accuracy of 49.35%, whereas ResNet101 had the second lowest validation accuracy of 78.38%. Figure 7 depicts the comparative analysis of the training and validation performance of all nine models using transfer learning in terms of accuracy.

All nine of the models generalized really well and were fairly similar to one another when it came to fine-tuning. The generalization performance of the Xception DenseNet201, VGG19, MobileNetV2, NASNetLarge, EfficientNetB2, InceptionV3, and ViT was above 99% in terms of accuracy. Additionally, ViT had the best generalization performance with a score of 99.91%, while ResNet101

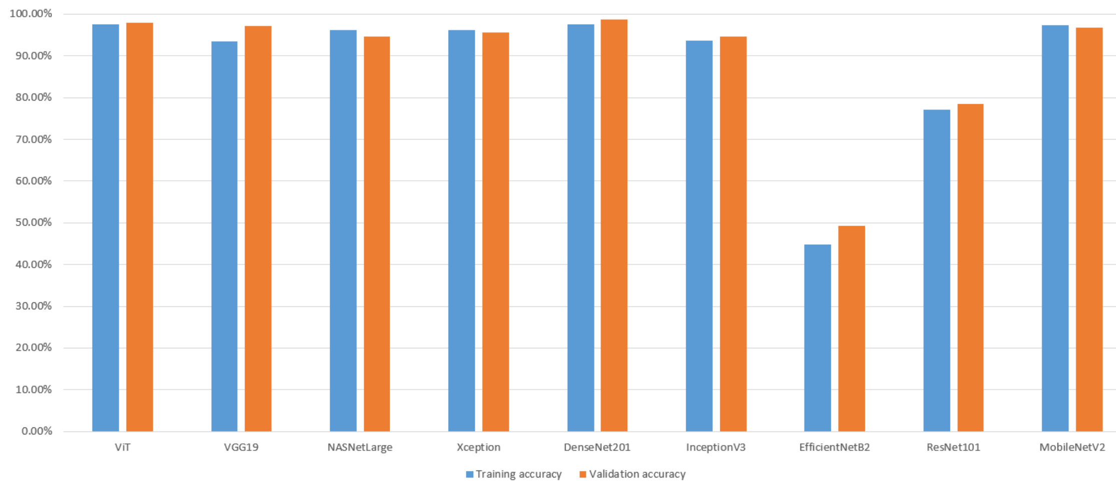

Fig. 7. Comparison among training and validation accuracy of all nine models using transfer learning.

demonstrated the lowest generalization performance with a score of 98.79%. Figure 8 depicts a comparison of the training, validation, and testing accuracy of all nine models for fine-tuning approach.

Table 6. Comparison among training and validation accuracy of all nine models using fine-tuning.

Model name	Training accuracy	Validation accuracy
ViT	0.9992	0.9991
VGG19	0.997	0.998
NASNetLarge	0.9994	0.9983
Xception	0.9983	0.9973
DenseNet201	0.9983	0.9986
InceptionV3	0.9982	0.997
EfficientNetB2	0.993	0.993
ResNet101	0.9892	0.9879
MobileNetV2	0.998	0.9933

In the end, the DenseNet201 model had the greatest validation accuracy of 98.76%, making it the top performer for transfer learning. The ViT model obtained the best validation accuracy of 99.91% for fine-tuning. In contrast, the ResNet101 model had the lowest accuracy for fine-tuning, while the EfficientNetB2 model had the lowest accuracy for transfer learning, with a validation accuracy of 49.35%.

Overall, our results indicate that constructing precise models for traffic sign identification in Bangladesh may be accomplished using transfer learning and fine-tuning strategies. In addition, given their excellent accuracy, DenseNet201 and ViT models would be the best options for this task. However, the choice of the model may depend on various factors, such as the size of the dataset, computational resources, and other application-specific requirements.

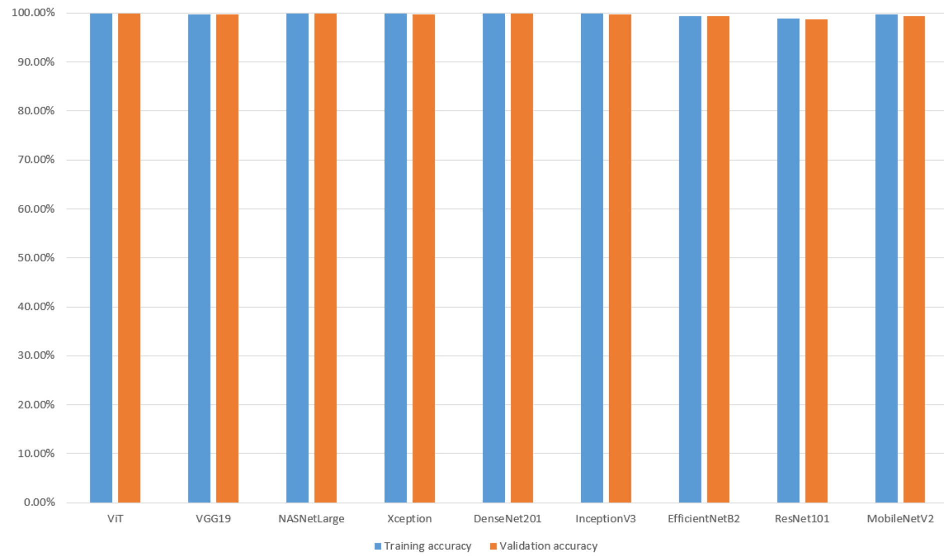

Fig. 8. Comparison among training and validation accuracy of all nine models using fine-tuning.

6. Stakeholders

List Identifying stakeholders is an important aspect of any project, as it helps to ensure that the project meets the needs and expectations of those who will be impacted by its outcomes. In the case of our traffic sign recognition project, stakeholders may include:

1. **Drivers:** Drivers are the primary users of traffic signs, and are therefore the most directly impacted by the accuracy and reliability of traffic sign recognition systems. They may benefit from improved road safety and reduced risk of accidents through the use of such systems.

2. **Pedestrians:** Pedestrians also rely on traffic signs to navigate roads and intersections safely. Traffic sign recognition systems may help to improve their safety by reducing the risk of accidents involving pedestrians and vehicles.

3. **Government agencies:** Government agencies responsible for transportation and road safety may be stakeholders in our project, as they may be interested in using traffic sign recognition systems to improve road safety and traffic management.

4. **Technology companies**: Technology companies that develop computer vision and machine learning systems may be interested in our project, as it may provide opportunities for developing new products or services related to traffic sign recognition.

5. **Insurance companies:** Insurance companies may also have a stake in our project, as improved road safety may lead to fewer accidents and claims, potentially reducing their costs.

6. **General public:** The general public may also be stakeholders in our project, as improved road safety may benefit society as a whole by reducing the number of accidents and injuries on the road.

Identifying and understanding the needs and expectations of these stakeholders is important for ensuring that our project meets its objectives and has a positive impact on society.

7. Issues Encountered

List During the course of our traffic sign recognition project, we encountered several issues that had to be addressed. Some of the major issues we faced include:

Limited dataset: We initially had a limited dataset of traffic sign images, which could have affected the accuracy and robustness of our models. To address this, we employed data preprocessing and augmentation techniques to create a more diverse and representative dataset.

Hardware limitations: Our hardware resources, particularly CPU and GPU, were limited, which could have affected the speed and performance of our models. To address this, we utilized Google Colab, a cloud-based platform that provided access to powerful hardware resources.

Model selection and evaluation: We found that the performance of different pre-trained models varied significantly, with some models achieving very low accuracy even after transfer learning and fine-tuning. To address this, we carefully selected and evaluated different models, taking into account factors such as performance, complexity, and computational requirements.

By addressing these issues, we were able to develop a more accurate and robust traffic sign recognition system that has the potential to improve road safety and traffic management.

8. Conclusion, Limitations and Future Recommendations

Traffic sign identification and recognition have lately emerged as one of the most important fields of computer vision and image processing due to their importance in autonomous driving and traffic safety. Early studies employed a variety of standard datasets, but few researchers concentrated on developing a database of traffic signs from the perspective of Bangladesh. We collected approximately 2000 photographs representing 13 distinct types of Bangladesh traffic signs using four independent cameras and then used a variety of data augmentation techniques to increase the dataset size to 8386 images. Additionally, we used transfer learning and fine-tuning approaches to train nine deep-learning models on our dataset, and we then compared the outcomes. In our analysis, we observed that the ViT model achieved the highest validation accuracy of 99.91% for fine-tuning while the DenseNet201 model earned the best validation accuracy of 98.76%, making it the top performer for transfer learning. Overall, the findings demonstrate that fine-tuning pre-trained models is an effective technique for recognizing traffic signs in Bangladesh. These models can be used to improve road safety and decrease accident rates in real-world situations.

However, there are some limitations to our project that should be acknowledged. Firstly, our dataset was limited in size and may not be fully representative of all traffic signs. Secondly, we did not explore the integration of our system with camera-based systems in real-time, which could be an important area for future work. Finally, our hardware resources were limited, which may have affected the speed and performance of our models.

In the future, we hope to expand our database with more images captured in diverse real-world scenarios and other types of Bangladeshi traffic signs. Additionally, we would like to experiment with more deep-learning models and develop a traffic sign recognition app that can assist drivers in averting serious accidents in real-time.

References

[1] Zhaoxiang Zhang, Tieniu Tan, Kaiqi Huang, and Yunhong Wang. Practical camera calibration from moving objects for traffic scene surveillance. IEEE transactions on circuits and systems for video technology, 23(3):518–533, 2012.

[2] Yingjie Xia, Weiwei Xu, Luming Zhang, Xingmin Shi, and Kuang Mao. Integrating 3d structure into traffic scene understanding with rgb-d data. Neurocomputing, 151:700–709, 2015.

[3] François Chollet. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1251–1258, 2017.

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[6] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and LiangChieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4510–4520, 2018.

[7] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[8] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and ZbigniewWojna. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2818– 2826, 2016.

[9] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4700–4708, 2017.

[10] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning, pages 6105–6114. PMLR, 2019.

[11] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 8697–8710, 2018.
[12] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. Detection of traffic signs in real-world images: The german traffic sign detection benchmark. In The 2013 international joint conference on neural networks (IJCNN), pages 1–8. Ieee, 2013.
[13] Shehan P Rajendran, Linu Shine, R Pradeep, and Sajith Vijayaraghavan. Realtime traffic sign recognition using yolov3 based detector. In 2019 10th internationalDeep-Learning-Based Bangladesh Traffic Signs Recognition 11 conference on computing, communication and networking technologies (ICCCNT), pages 1–7. IEEE, 2019.
[14] Ahmed Hechri and Abdellatif Mtibaa. Two-stage traffic sign detection and recognition based on svm and convolutional neural networks. IET Image Processing, 14(5):939–946, 2020.
[15] R Karthika and Latha Parameswaran. A novel convolutional neural network based architecture for object detection and recognition with an application to traffic sign recognition from road scenes. Pattern Recognition and Image Analysis, 32(2):351– 362, 2022.
[16] Yingying Zhu, Chengquan Zhang, Duoyou Zhou, Xinggang Wang, Xiang Bai, and Wenyu Liu. Traffic sign detection and recognition using fully convolutional network guided proposals. Neurocomputing, 214:758–766, 2016.
[17] Genevieve Sapijaszko, Taif Alobaidi, and Wasfy B Mikhael. Traffic sign recognition based on multilayer perceptron using dwt and dct. In 2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS), pages 440–443. IEEE, 2019.
[18] Shouhui He, Lei Chen, Shaoyun Zhang, Zhuangxian Guo, Pengjie Sun, Hong Liu, and Hongda Liu. Automatic recognition of traffic signs based on visual inspection. IEEE Access, 9:43253–43261, 2021.
[19] Chang Sun, Yibo Ai, Sheng Wang, and Weidong Zhang. Dense-refinedet for traffic sign detection and classification. Sensors, 20(22):6570, 2020.
[20] Bei Bei Fan and He Yang. Multi-scale traffic sign detection model with attention. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 235(2-3):708–720, 2021.
[21] Zhigang Liu, Dongyu Li, Shuzhi Sam Ge, and Feng Tian. Small traffic sign detection from large image. Applied Intelligence, 50:1–13, 2020.
[22] Jianming Zhang, Zhipeng Xie, Juan Sun, Xin Zou, and Jin Wang. A cascaded r-cnn with multiscale attention and imbalanced samples for traffic sign detection. IEEE access, 8:29742–29754, 2020.
[23] Jianming Zhang, Wei Wang, Chaoquan Lu, Jin Wang, and Arun Kumar Sangaiah. Lightweight deep network for traffic sign classification. Annals of Telecommunications, 75:369–379, 2020.
[24] Domen Tabernik and Danijel Skočaj. Deep learning for large-scale traffic-sign detection and recognition. IEEE transactions on intelligent transportation systems, 21(4):1427–1440, 2019.
[25] Chunmian Lin, Lin Li, Wenting Luo, Kelvin CP Wang, and Jiangang Guo. Transfer learning based traffic sign recognition using inception-v3 model. Periodica Polytechnica Transportation Engineering, 47(3):242–250, 2019.
[26] Cen Han, Guangyu Gao, and Yu Zhang. Real-time small traffic sign detection with revised faster-rcnn. Multimedia Tools and Applications, 78:13263–13278, 2019.
[27] Nabil Ahmed, Sifat Rabbi, Tazmilur Rahman, Rubel Mia, and Masudur Rahman. Traffic sign detection and recognition model using support vector machine and histogram of oriented gradient. International Journal of Information Technology and Computer Science, 13(3):61–73, 2021.
[28] tusher7575. Traffic sign in bangladesh, 2023. https://www.kaggle.com/datasets/tusher7575/traffic-sign-in-bangladesh [Accessed: (Apr 2023.)].

Appendix

Attainment of Complex Engineering Problem (CP)

S.L.	CP No.	Attainment	Remarks
1.	P1: Depth of Knowledge Required	Yes	K3 (Engineering Fundamentals): python describe in section 4. Statistics describe in section 5.
			K4 (Engineering Specialization): knowledge about Deep learning describe in section 4.
			K5 (Design): Methodology flow chart describe in section 4.
			K6 (Technology): knowledge about tensorflow describe in section 4.
			K8 (Research): Related work describe in section 2
2.	P2: Range of Conflicting Requirements	Yes	Learn about Bangladeshi traffic sign describe in section 3 and knowledge about deep learning describe in section 4.
3.	P3: Depth of Analysis Required	Yes	Build new dataset about traffic sign describe in section 3.
4.	P4: Familiarity of Issues	Yes	Study about traffic sign describe in section 1 and section 3
5.	P5: Extent of Applicable Codes	Yes	Follow standards methodology describe on section 4
6.	P6: Extent of Stakeholder Involvement and Conflicting Requirements	Yes	Stakeholder are Drivers, Pedestrians, 4. Technology companies etc. describe in section 6
7.	P7: Interdependence	Yes	Collecting roadside traffic signs describe in section 3. Apply different augmentations also describe in section 3.

Mapping of Complex Engineering Activities (CA)

S.L.	CA No.	Attainment	Remarks
1.	A1: Range of resources	Yes	Drivers, Pedestrians, Technology companies etc. describe in section 6.
2.	A2: Level of interaction	Yes	There are several issues come when we work this project describe in section 7.

3.	A3: Innovation	Yes	Build new dataset and build model for detect Bangladeshi traffic signs describe in section 3 and section 4.
4.	A4: Consequences for Society and the Environment	Yes	This project helps Drivers, Pedestrians, Technology companies etc. Describe in section 1 and section 6.
5.	A5: Familiarity	Yes	Build a new Bangladeshi traffic sign detection model describe in section 4.