A Transfer Learning Approach to Traffic Sign Recognition

Mohammed Khaja Khan[1], Mohammed Abdullah[2], Shaik Mohammed Suhaib[3]

[1,2,3]*Department of Information Technology, Muffakham Jah College of Engineering and Technology, Hyderabad, India*

Abstract: *Traffic sign recognition is a crucial task for many modern applications and intelligent systems like autonomous vehicles, advanced driver assistance systems and real-world computer vision problems. Deep learning techniques require great effort and long durations of training to even start being usable. In this study, traffic sign recognition and classification is implemented using transfer learning concept. We focussed on smaller CNN architectures compared to state-of-the-art CNN architectures like VGG, AlexNet and some ResNet models. We experimented with three pre-trained models-InceptionV3, Resnet50 and Xception. The results from using each of these models are compared in efficiency and accuracy. The transfer learning models are trained using the German Traffic Sign Recognition Benchmark (GTSRB) dataset. Of these three, the highest accuracy of 97.15% was achieved using InceptionV3 architecture.*

Key Words:- Traffic Sign Classification, CNN, Transfer Learning, InceptionV3, Xception.

1. INTRODUCTION

Accurate and efficient traffic sign recognition is becoming an increasingly important task in automation systems like traffic assistance, automatic driving systems etc. In 1968 the Vienna Convention on Road Signs and Signals was signed which standardized traffic signs across various countries. In accordance to it, road signs are broadly classified into seven categories: A, B, C, D, E, F, G and H. Due to this standardization, it has become feasible to develop traffic sign recognition systems that can be used widely. Traffic-sign recognition commercially emerged first in 2008 as a form of speed limit sign recognition in cars. Following that, many models were introduced to detect other signs such as overtaking restrictions, school zones etc. Systems like these are expected to be mandatory in new vehicles sold in the European Union May 2022 onwards. With autonomous vehicles on the rise, now more than ever traffic sign recognition research is being conducted intensively.

In recent years, research in traffic sign recognition has grown rapidly due to increasing needs of systems using it. Performance required by such systems includes high recognition rates, real time implementation, recognition of many different signs in various weather conditions and in night mode. Many algorithms have hence been developed to address these problems. Image Processing and Computer vision is used to facilitate carrying out the training. In the real world, it is difficult for machines to interpret traffic signs due to various factors such as occlusion, exposure, fading, weather conditions etc. In order to improve traffic sign recognition, many studies have been performed. Mainly, automated traffic sign recognition methods are categorised into two groups: feature based and deep learning-based techniques.

In 2010 Flayeh *et al*. [1] introduced an eigen-based traffic sign recognition method to classify unknown traffic signs using Principal Component Analysis (PCA) algorithm. The method was implemented on two different datasets of traffic signs and speed limit signs of 1259 and 1927 images respectively. In 2014 Biswas *et al*. [2] presented a traffic sign recognition system based on extracting speed limit sign from the traffic scene using Circular Hough Transform (CHT), digits of the speed limit were then classified using a trained Support Vector Machine (SVM) classifier. The SVM classifier was trained on a small dataset of 270 images collected in different light conditions. Several other researchers have proposed SVM classifier based solutions to solve this problem [3-5]. Ellahani *et al.* [6] presented a traffic sign detection and recognition using random forests and SVMs which was experimented on the German Traffic Sign Detection and Recognition Benchmark and the Swedish Traffic Signs Data sets with robust accuracy. Zaklouta *et al.* [7] used K-d trees and Random Forests using different sized Histogram of Oriented Gradients (HOGs) and Distance Transforms. Using the German Traffic Sign Benchmark dataset containing 43 classes and more than 50000 images. These classifiers were then combined with HOGs descriptors and Distance Transform to get robust accuracies. Other researchers [8] have also used Random Forests for the same problem. Feature based methods like SVMs or Random Forests produce satisfactory results, however hand-engineering features need very specific knowledge and skills, which demands much of both human expertise and labour; and hand-engineering features fail to capture the overall features of traffic signs hence resulting in unsatisfactory results when implemented in the real world.

With the recent surge in deep learning research, high volumes of research have been carried out in traffic sign recognition using deep hierarchical networks. With the German Traffic Sign Recognition Benchmark (GTSRB) being held by the International Joint Conference Neural

Network (IJCNN) and IEEE Computational Intelligence Society (CIS), many deep learning models are being created having great accuracies. Ciresan *et al.* [9] won the German traffic sign recognition benchmark (GTSRB) by achieving a recognition rate better than humans: 99.46%. This was done by combining many Deep Neural Networks (DNNs) trained on separately pre-processed data into a Multi-Column Deep Neural Network (MCDNN). Many other researchers have used Deep Learning methods to solve Traffic Sign Recognition which have produced great results in term of recognition accuracy [10-12]. However, these amazing accuracy levels come with a cost; deep learning models are designed to have an iterative trial and error process requiring great amounts of labelled data for training. Moreover, the huge number of connections in the networks make sure it is computationally very expensive to train them.

Transfer learning is a machine learning strategy in which a pre-trained model for a certain problem is transferred to solve a similar problem by fine tuning parameters across all the layers and learning new data. Various studies have been carried out on image recognition using transfer learning [13-15]. VGG architectures [16] were explored for traffic sign recognition tasks by researchers [17,18], and have shown promising results, but the main drawback of these architectures are that they are computationally expensive. For example, the VGG16 and VGG19 architectures have parameter count of approximately 138 million and 143 million respectively.

In this study, we use transfer learning strategy for traffic sign recognition and using three CNN architectures: Xception, Inception-v3 and resnet50. The results using each model are compared.

2. METHODOLOGY

We have used an open dataset [19] consisting of 50,000 images across 43 classes of different traffic signs. The images were resized to 224 x 224. Data or image augmentation technique was used on the training images to further increase the size of the dataset and to improve the performance of the model.

Transfer learning concept was used to implement the Convolutional Neural Networks (CNNs). Transfer learning is a concept of using a pre-trained model trained for a particular task for another similar task. In other words, a model trained for a particular task is reused to perform a similar task. In our work, we implemented three pre-trained CNN architectures – InceptionV3, Xception and ResNet50. These architectures were trained in the imagenet dataset [24] for classifying 1000 classes of images. These architectures were then finetuned and re-trained for classifying 43 classes of traffic signs. The images were divided into batch sizes of 32. The optimizer used for implementing these architectures was ‘adam’ and ‘categorical_crossentropy’ was used as loss function.

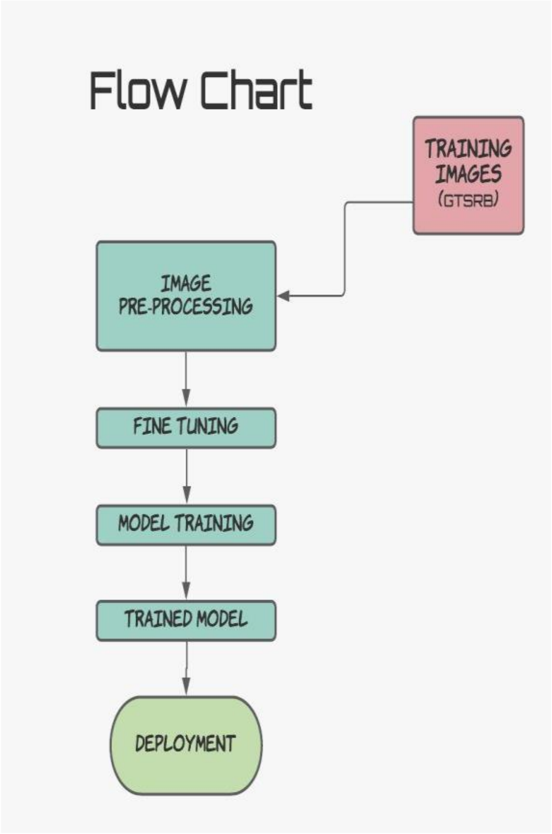

Fig -1: Flowchart of the proposed approach

2.1 InceptionV3

InceptionV3 proposed by [20] is a Convolutional Neural Network (CNN) architecture from the Inception family [21] that produces many enhancements together with Label Smoothing, factorized convolutions, and the use of an auxiliary classifier to propagate label data lower down the neural network and with the employment of batch standardization for layers within the side head. Inception-V3 is a 48-layer deep neural network with 23,851,784 parameters.

The architecture of the Inception v3 network is built step-by-step, as explained:

1. Factorized Convolutions: This helps to cut back the process efficiency because it reduces the quantity of parameters concerned during a network. Larger convolutions are substituted with smaller convolutions. For example, a 5x5 filter has 25 parameters, two 3x3 filters substituting a 5x5 convolutions has solely 18 (3*3 + 3*3) parameters instead. It conjointly keeps a check on

the network efficiency. Overall, applying factorized convolutions reduces the parameters by 28%.

2. Asymmetric convolutions: A 3 x 3 convolution could be replaced by a 1 x 3 convolution followed by a 3 x 1 convolution. Using asymmetric convolutions, the number of parameters is further reduced by 33%.

3. Auxiliary classifier: An auxiliary classifier is a compact CNN inserted between layers throughout training. The auxiliary classifier is used to improve the convergence of the network during training by pushing useful gradients to the lower layers, making them immediately useful, and combating the vanishing gradient descent problem.

4. Grid-size reduction: Grid size reduction is done by using pooling after a convolution operation.

2.2 Xception

The Xception model, proposed by [22] uses the concept of depth-wise separable convolutions. Xception was inspired by inceptionV3 architecture where the inception modules were replaced by depth-wise separable convolution layers. It has a depth of 126, including 36 convolutional layers to extract features. A global average pooling layer is used to replace the fully connected layer to reduce the number of parameters, the softmax function is used to output the prediction. The 36 convolutional layers are structured into 14 modules, all of which have linear skip connections around them, except for the first and last modules. The 36 convolutional layers are divided into 3 components: entry flow; middle flow and exit flow. The input data format of 299x299 RGB images, first goes through the entry flow, then through the middle flow which is repeated eight times, and then finally through the exit flow. The entry flow consists of 8 convolutional layers, the middle flow consists of 24 (8*3) convolutional layers and the exit flow consists of 4 convolutional layers. The Xception model applies depth-wise separable convolution, which can significantly reduce the convolution operation cost.

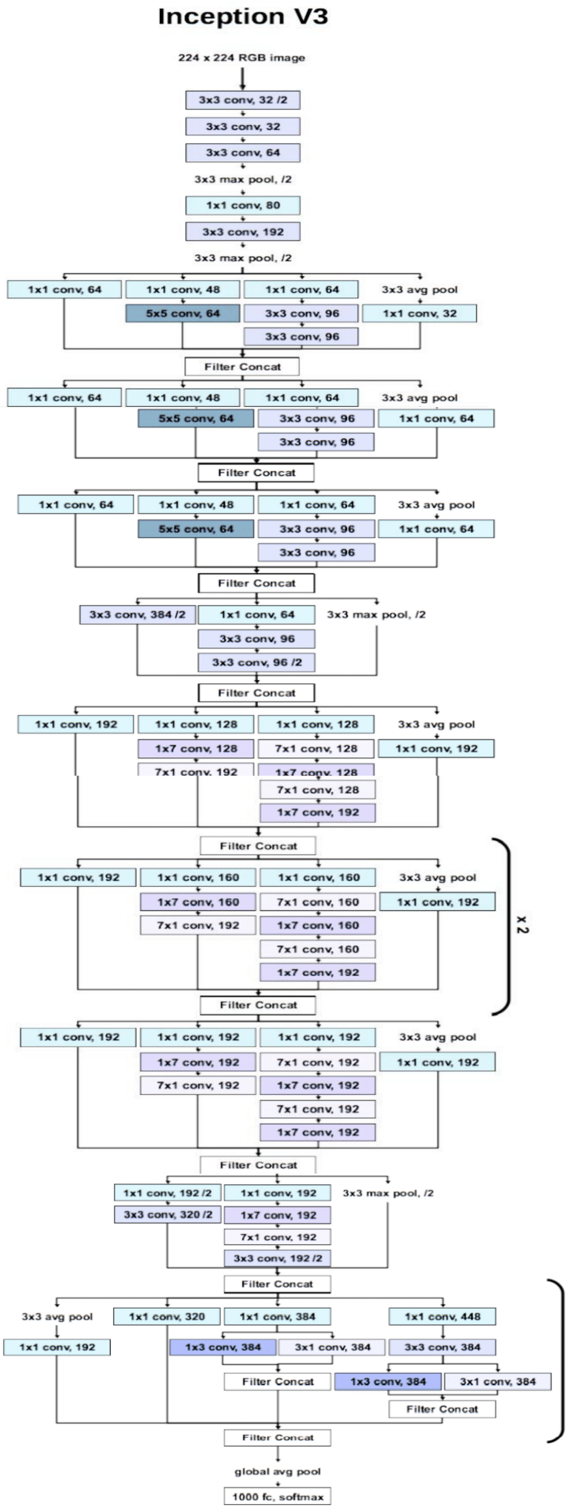

Fig -2: Architectures for InceptionV3

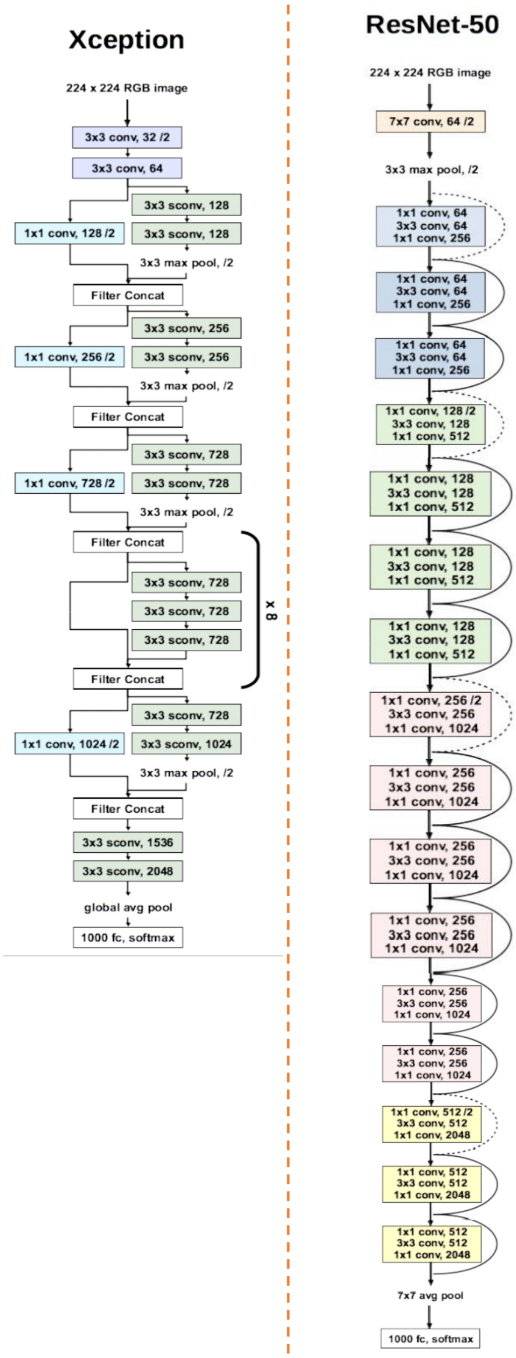

Fig -3: Architectures for Xception and ResNet50

2.3 ResNet50

Residual Networks (ResNets) [23] are deep convolutional networks where the basic idea is to skip blocks of convolutional layers by using shortcut connections. The basic blocks named "bottleneck" blocks follow two simple design rules:

(i) for the same output feature map size, the layers have the same number of filters; and

(ii) if the feature map size is halved, the number of filters is doubled.

The down-sampling is performed directly by convolutional layers that have a stride of 2 and batch normalization is performed right after each convolution and before ReLU activation. When the input and output are of the same dimensions, the identity shortcut is used.

The down-sampling is performed directly by convolutional layers that have a stride of 2 and batch normalization is performed right after each convolution and before ReLU activation. When the input and output are of the same dimensions, the identity shortcut is used. When the dimensions increase, the projection shortcut is used to match dimensions through 1 × 1 convolutions. In both cases, when the shortcuts go across feature maps of two sizes, they are performed with a stride of 2. The network ends with a 1000 fully connected layers with SoftMax activation. The total number of weighted layers is 50, with 23,534,592 trainable parameters.

The ResNet50 structure has different groups of identical layers as indicated by different colours as shown in the figure. The curved lines represent the identified blocks that are used to indicate the use of previous layers in the following layers. It is the key difference in ResNet50 that counterfeit problem of vanishing or exploding gradients, degradation problem (accuracy first saturates and then degrades) in training very deep networks. In the figure, the first layer has 64 filters with a kernel size of 7×7, which is followed by a max-pooling layer of size 3×3. The first group of layers (as indicated by grey colour) consists of three identical blocks. In the same way group two, group three, and group four have 4 identical blocks, 4 identical blocks, and 3 identical blocks respectively. In between some groups, the curves marked with blue colour represent the identity block that connects two layers of different sizes. After all these blocks, there is a total of 38 fully connected layers responsible for the classification task.

3. RESULTS

The traffic sign recognition and classification systems were developed using the transfer learning approach. Three pre-trained CNN architectures – InceptionV3, Xception and ResNet50, were used in this study to develop traffic sign classification models. These architectures were originally trained to classify 1000 categories of objects. So they were fine-tuned based on our dataset i.e. to classify 43 categories of traffic signs. The models developed were trained using the training parameters as shown in table 1. The models were trained on a laptop with Intel i7 processor, 16GB RAM and Nvidia GeForce mx450 GPU. It took around 9 to 10 hours to train each model, as these models are of similar size and have around the same number of parameters.

Table 1: Training Parameters

Parameter	**Value**
Batch Size	32
Epoch	20
Optimizer	adam
Loss Function	categorical_crossentropy

Of the three CNN architectures used, InceptionV3 and Xception achieved high results with accuracies greater than 96%. The InceptionV3 model slightly outperformed the Xception model in terms of accuracy. This was contrary to the study done by M.A.Moid *et al*. [25], where they had experimented with InceptionV3 and Xception architectures for the plant disease classification and here the Xception architecture showed better results than the InceptionV3.

In our experiments, the Xception model achieved an accuracy of 96.79% and loss of 0.701. The accuracy for the InceptionV3 based model was 97.15% and its loss was 0.852. On the other hand, the model developed using ResNet50 architecture showed unsatisfactory results of 60.69% accuracy and loss of 2.531.

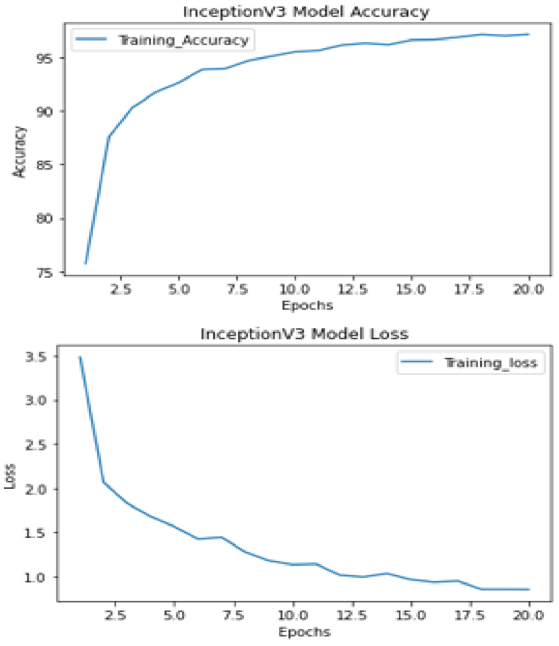

Fig -4: InceptionV3 model accuracy and loss

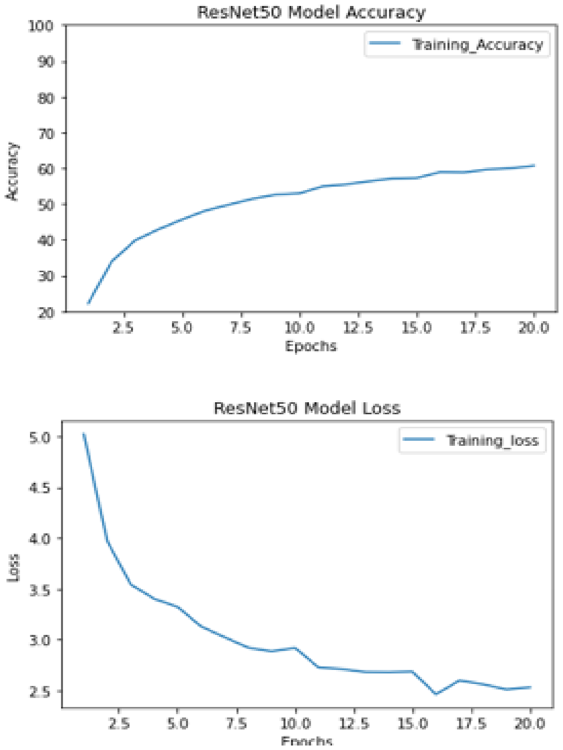

Fig -5: Xception model accuracy and loss

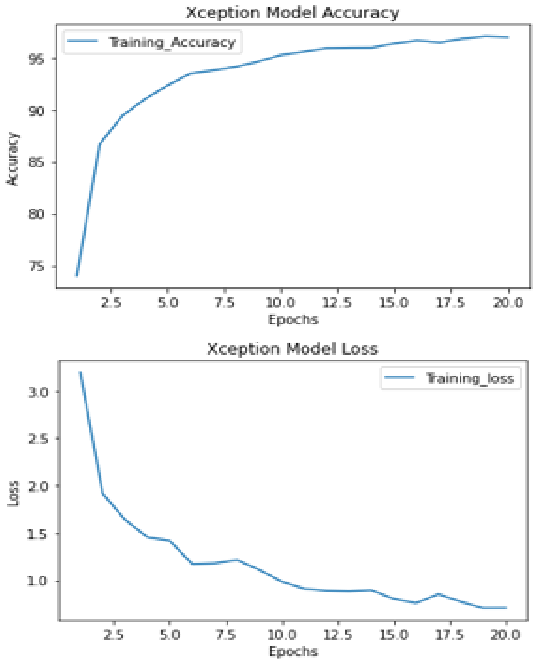

Fig -6: ResNet50 model accuracy and loss

4. CONCLUSIONS

In this paper, we presented transfer learning-based solutions for recognizing and classifying traffic signs. The expected real world applications of traffic sign recognition systems are in autonomous vehicles and driver assistance systems, and the models trained to detect and classify traffic signs need to be deployed on edge devices which have limited computational requirements. Hence, we implemented only those architectures with smaller model sizes and lower number of parameters. We have experimented with three pre-trained CNN architectures: InceptionV3, Xception and ResNet50. These architectures have relatively smaller model sizes and parameter counts, compared to other state of the art CNN architectures like VGG, AlexNet and other ResNet architectures.

Both InceptionV3 and Xception models achieved high results; however the former slightly outperformed the latter model in terms of accuracy. The model developed through Xception architecture achieved an accuracy of 96.79 per cent and loss of 0.701. The InceptionV3 model achieved an accuracy of 97.15 per cent with a loss of 0.85. These two architectures are quite small compared ResNet50. The ResNet50 model achieved low results of around 60% accuracy and loss of 2.5.

In this study, we have found that transfer learning approach for traffic sign classification produced satisfactory results using inceptionV3 and Xception architectures.

REFERENCES

[1] Fleyeh Hasan & Davami, Erfan. (2011). Eigen-based traffic sign recognition. Intelligent Transport Systems, IET. 5. 190 - 196. 10.1049/iet-its.2010.0159.

[2] Biswas, R. & Fleyeh, Hasan & Mostakim, Moin. (2014). Detection and classification of speed limit traffic signs. 2014 World Congress on Computer Applications and Information Systems, WCCAIS 2014. 10.1109/WCCAIS.2014.6916605.

[3] G. Wang, G. Ren, Z. Wu, Y. Zhao and L. Jiang, "A hierarchical method for traffic sign classification with support vector machines," The 2013 International Joint Conference on Neural Networks (IJCNN), 2013, pp. 1-6, doi: 10.1109/IJCNN.2013.6706803.

[4] C. G. Kiran, L. V. Prabhu, A. R. V. and K. Rajeev, "Traffic Sign Detection and Pattern Recognition Using Support Vector Machine," 2009 Seventh International Conference on Advances in Pattern Recognition, 2009, pp. 87-90, doi: 10.1109/ICAPR.2009.58.

[5] H. Fleyeh and M. Dougherty, "Traffic sign classification using invariant features and Support Vector Machines," 2008 IEEE Intelligent Vehicles Symposium, 2008, pp. 530-535, doi: 10.1109/IVS.2008.4621132.

[6] Ayoub Ellahyani, Mohamed El Ansari, Ilyas El Jaafari, Traffic sign detection and recognition based on random forests, Applied Soft Computing, Volume 46, 2016, Pages 805-815, ISSN 1568-4946, https://doi.org/10.1016/j.asoc.2015.12.041.

[7] F. Zaklouta, B. Stanciulescu and O. Hamdoun, "Traffic sign classification using K-d trees and Random Forests," The 2011 International Joint Conference on Neural Networks, 2011, pp. 2151-2155, doi: 10.1109/IJCNN.2011.6033494.

[8] J. Greenhalgh and M. Mirmehdi, "Traffic sign recognition using MSER and Random Forests," 2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO), 2012, pp. 1935-1939.

[9] Dan Cireşan, Ueli Meier, Jonathan Masci, Jürgen Schmidhuber, Multi-column deep neural network for traffic sign classification, Neural Networks, Volume 32, 2012, Pages 333-338, ISSN 0893-6080, https://doi.org/10.1016/j.neunet.2012.02.023.

[10] Amara, Dinesh. (2018). Novel Deep Learning Model for Traffic Sign Detection Using Capsule Networks. International Journal of Pure and Applied Mathematics Volume 118 ISSN: 1314-3395, arXiv:1805.04424

[11] S. Mehta, C. Paunwala and B. Vaidya, "CNN based Traffic Sign Classification using Adam Optimizer," 2019 International Conference on Intelligent Computing and Control Systems (ICCS), 2019, pp. 1293-1298, doi: 10.1109/ICCS45141.2019.9065537.

[12] M. A. Vincent, V. K. R and S. P. Mathew, "Traffic Sign Classification Using Deep Neural Network," 2020 IEEE Recent Advances in Intelligent Computational Systems (RAICS), 2020, pp. 13-17, doi: 10.1109/RAICS51191.2020.9332474.

[13] Raina, R., Battle, A., Lee, H., et al. (2007) Self-Taught Learning: Transfer Learning from Unlabelled Data. Proceedings of International Conference on Machine Learning. Corvallis, OR, USA. DOI: 10.1145/1273496.1273592

[14] Esteva, A., Kuprel, B., Novoa, R. (2017) Dermatologist-Level Classification of Skin Cancer with Deep Neural Networks. Nature. 542, pp. 115-118. DOI: 10.1038/nature21056

[15] Devikar, P. (2016) Transfer Learning for Image Classification Various Fog Breed. International Journal of Advanced Research in Computer Engineering & Technology. 5(12), pp. 2278-2323.

[16] S. Liu and W. Deng, "Very deep convolutional neural network-based image classification using small training sample size," 2015 3rd IAPR Asian Conference

on Pattern Recognition (ACPR), 2015, pp. 730-734, DOI: 10.1109/ACPR.2015.7486599.

[17] Z. Lin, M. Yih, J. M. Ota, J. D. Owens and P. Muyan-Özçelik, "Benchmarking Deep Learning Frameworks and Investigating FPGA Deployment for Traffic Sign Classification and Detection," in IEEE Transactions on Intelligent Vehicles, vol. 4, no. 3, pp. 385-395, Sept. 2019, doi: 10.1109/TIV.2019.2919458.

[18] Bi, Z., Yu, L., Gao, H. *et al.* Improved VGG model-based efficient traffic sign recognition for safe driving in 5G scenarios. *Int. J. Mach. Learn. & Cyber.* 12, 3069–3080 (2021). https://doi.org/10.1007/s13042-020-01185-5

[19] J. Stallkamp, M. Schlipsing, J. Salmen, C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition", Neural Networks, Volume 32, 2012, Pages 323-332, ISSN 0893-6080, https://doi.org/10.1016/j.neunet.2012.02.016.

[20] C. Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke and Andrew Rabinovich, "Going deeper with convolutions," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1-9, DOI: 10.1109/CVPR.2015.7298594.

[21] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2818-2826, DOI: 10.1109/CVPR.2016.308.

[22] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1800-1807, DOI: 10.1109/CVPR.2017.195.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, "Deep Residual Learning for Image Recognition", 2015, https://arxiv.org/abs/1512.03385

[24] J. Deng, W. Dong, R. Socher, L. Li, Kai Li and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248-255, DOI: 10.1109/CVPR.2009.5206848.

[25] M. A. Moid and M. Ajay Chaurasia, "Transfer Learning-based Plant Disease Detection and Diagnosis System using Xception," 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2021, pp. 1-5, doi: 10.1109/I-SMAC52330.2021.9640694.