2021 6th International Conference for Convergence in Technology (I2CT)
Pune, India. Apr 02-04, 2021

Plant Disease Detection Using Deep Learning

Ebrahim Hirani, Varun Magotra, Jainam Jain, Pramod Bide
Department of Computer Engineering,
Sardar Patel Institute of Technology
Mumbai-400058, India
Email: ebrahim.hirani@spit.ac.in, varun.magotra@spit.ac.in jainam.jain@spit.ac.in, pramod_bide@spit.ac.in

Abstract—**In recent years, use of Convolutional neural networks has been explored in a wide range of applications whether its image classification, feature extraction or image segmentation. One of those applications is plant disease detection, since plant disease is one of the most significant factors that leads to poor yield in the agricultural sector. Over the period of time various deep learning approaches have been used to solve this problem of identification and classification of plant disease. But then to there are some limitation to these approaches. The recent application of transformer networks in computer vision tasks has shown great promise. This paper compares these approaches with traditional CNN approaches in the task of plant disease detection.The best validation accuracy that our transformer model achieves is 97.98%.**

Index Terms—**CNN, Transfer Learning, InceptionV3, Transformer**

I. Introduction

Agriculture contributes to around 17% of India's GDP and provides 60% of the total employment in the country. Climate change is already affecting the yield in many parts of the country, and it is a factor that cannot be controlled. On the other hand, yield reduction due to plant diseases is a factor that can be managed. Agricultural yield is affected by various Plant diseases. Plant diseases are affecting crops to a great extent causing not only economical damage but also affecting the food quality and its supply.

Various methods and processes are being initiated to diagnose these plant diseases. Though there are some farmers who are fortunate enough to afford labs and expert supervision. Most of them cannot afford experts and equipment to help them recognize plant diseases in their early stages and prevent their spread. Also, to identify the disease of the plant farmers have to take the sample to the designated labs. The amount of money and resources spent to set up these labs is not economically viable, and those resources could be better used to make other investments to improve the yield. So, to overcome the cost of lab and expert personnel in that field, we must adopt digitized ways to diagnose the plant diseases, in order to make the yield fruitful for consumers as well as farmers.

There have been previous attempts at this problem using deep learning methodologies[1,2,3], we plan to explore them further and use the current state of the art algorithms to maximize accuracy and minimize the hardware requirements. So that the process of identifying diseases can be eased and the cost would be minimized.

In this study, we review the newly introduced deep learning methods and state of the art models to identify plant diseases which can be helpful to people in the agriculture sector to overcome the existing problem in this area. Rest of the paper is structured as follows: In Section II, we briefly discuss related works. Section III describes Material and Methods. Section IV, Result and Discussion. Section V, Conclusion.

II. Related Studies

This paper presents the various methods that can be used for identifying the plant leaf diseases. Over the period of time, various image processing approaches followed by machine learning models have been adopted for image classification. Recently, new deep learning architectures and approaches have been proposed which are the current state of the art in most computer vision problems. Introduction of CNNs[5] has revolutionized deep learning involvement in computer vision applications. CNNs make use of kernels of fixed dimensions that act as filters to capture spatial and temporal information of the object through feature maps. These feature maps are then fed to a feedforward neural network that acts as a classifier. Introduction of CNNs has made image processing and other tasks easier and more efficient as kernels take much less space compared to feedforward layers to process an image of the same size.

Multiple CNN architectures are used with variation in the count of occurrences and sequence of convolutional layers and pooling layers along with different parameters for stride and padding. The more recent architectures include residual connections that allow these models to have increased depth without the negative effects of vanishing gradients[6]. These models have millions of parameters and can hold huge amounts of information. Information captured through training on one set of data can be used to solve problems on another set of data through simple fine tuning without having to recapture information for initial layers of the network.

In the last two- three years, the concept of transfer learning has gotten popular in almost every computer vision application. The recent gain in popularity is because of the improved performance and accuracy in doing the tasks especially the identification and classification tasks[7]. Using transfer learning for Plant disease identification has proven to be useful and more accurate than training an state of the art model from the scratch[8]. Sharada P. Mohanty et al., in his paper has demonstrated how two state of the art models AlexNet[9] and GoogleNet[10] when trained using two different training approaches varies in result. On training the Alexnet from the scratch on the Plant Disease dataset, yielded less accurate results as compared to the pre trained GoogleNet on which the transfer learning is applied and the weights that are used in the GoogLeNet are from that of Imagenet dataset[11].

More recently, the introduction of attention layers through the inception of transformer neural networks have proved to solve the long dependency issue in a lot of natural language processing problems[12] . Initially, transformers were designed to handle sequential data which occur in Natural Language Processing tasks like Text generation, abstractive summary generation etc. But In the last couple of years, the transformer networks have proved to be applicable in computer vision tasks and have crushed a lot of benchmarks set by all the CNN architectures. These networks create attention maps which can be used to find dependencies between different parts of the image and then be fed to classification layers for image classification[13].

All of these methods have been proven to be successful in solving image classification problems. In our paper, we are exploring each of these methods for plant disease identification and classification and reviewing how efficient each of these methods are and what are the limitations of each classification method.

III. Material and Methods

A. Dataset Description

- The dataset used in this study is an augmented version of the PlantVillage dataset containing 87.9k images derived from the 54k images of the original dataset. This dataset contains 38 classes of plant-disease pairs and is divided into 80 percent for training and 20 percent for validation (more precisely 70295 images are for training and 17572 images are for validation). This dataset contains the images of the healthy plant leaves with their respective diseased leaves also. The resolution of the each image is 256*256 pixels.

B. Image processing and Data Augmentation

Data we are using in our study is already augmented. So, we only used the rescale parameter to normalize the images and the data split parameter of the ImageDataGeneator to split the data into a ratio of 80:20 where 80% of the data is for training and 20% is for validation. No other transformation is applied to the data.

C. Approach

In this study, Three approaches are used for the purpose for Plant disease Identification:

- **CNN** : A customized Convolutional neural network is created with 3 convolutional layers followed by relu activation and 3 MaxPooling2D layers(a max pooling layer succeeding each convolutional layer).The input image size that is feeded to the network is same as that of the original image(256*256). Different filter sizes are used

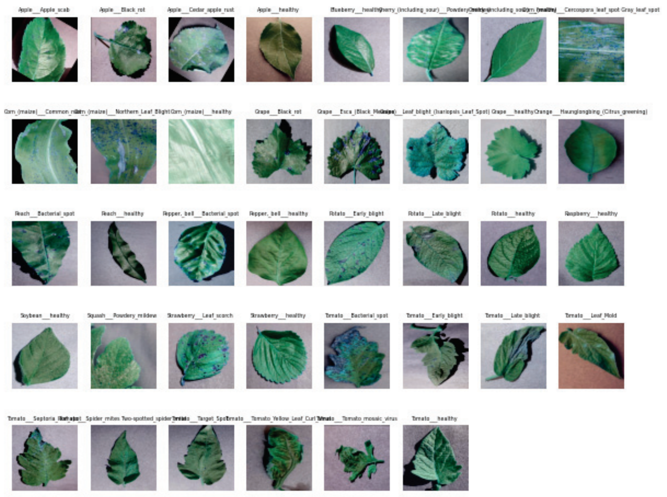

Fig. 1. Example of each class from the plant disease dataset

for each convolutional layer. One fully connected layer with 'relu' as activation function is used. BatchNormalization is also used to stabilize the learning process of the model and also to reduce the time to train the model. Dropout function is also used to reduce the overfitting of the model while training and to make the model more robust on the validation data. In the last layer 38 neurons are used to categorize the data and 'softmax' activation function is used for this layer. The total parameters of the model is 28,903,164 in which the Trainable params: 28,903,100 and Non-trainable params: 64
- **Transfer Learning**: For transfer learning purposes, we used one of the state of the art models i.e INCEPTIONv3[14] and used the weights of the model when it is trained on the IMAGENET dataset[11]. We made the top layers of the model untrainable and added a few layers to the INCEPTIONv3 net according to the need of the dataset. The layers we added for our purpose is the flatten layer to flatten the output layer of the INCEPTIONv3, then two dense layers are added with different neurons but with the same activation function 'relu'. The last layer is same as that we used for our custom CNN with the same parameter and activation function.This model has only 21,802,784 parameters and the target size of our model is (150,150).
- **Visual Transformers**: In this approach, the image is divided into grids of 16*16. Each component is fed into a feedforward network to get its embedding and added with the embedding for its positions. These embeddings are used as tokens and then fed to another feedforward layer to get 3 tokens for query, key and value. These tokens are used to calculate attention. The output is then fed to a feedforward layer which could then be used as tokens for the next transformer layer. These repeated layers could capture semantic information without the need of a lot of parameters. Each feedforward and attention layer is

preceded by a normalization layer and every repeated block of attention followed by feedforward has a residual connection with the previous one.

2 different sizes for transformer models were used:

Small Transformer Network(STN) : Model with token embedding representations having 256 dimensions, and feedforward layers also outputting 256 dimensions following each attention block. There are 8 attention blocks and input for each block is fed with 8 heads. This model has 3,499,046 parameters which is very less compared to the other models.

Large Transformer Network(LTN) : This model has token embedding representation of 128 dimensions, with feedforward layers following the attention blocks outputting 128 dimensions. It has 4 attention blocks and each block is fed 4 heads. This model has only 549,926 parameters which is the least of any model that we have tested in this study.

D. Training

After initializing these models, we trained them on the same ratio of data that is ever model is trained on the 80% of the original data and 20% of the original data is used for the validation. We also used various callback methods to make sure the model doesn't go on training without having any increase in its performance. We used 'ModelCheckpoint', 'ReduceLROnPlateau', 'EarlyStopping' for this purpose. ModelCheckpoint class allows us to define where to checkpoint the model weights, and what would be the circumstance under which the checkpoint is created whether we want to save the model when the validation accuracy is high or the validation loss is low. It provides us all these facilities so that after training we get the model with the best weights and accuracy. ReduceLROnPlateau assists us to reduce the learning rate when the metric has stopped improving. Models often improve from reducing the learning rate by a factor of 2-10 once learning ceases developing. This callback monitors a quantity and if no efficiency is detected for a patience number of epochs, then it reduces the learning rate. EarlyStopping class allows us to halt the training process once our model has stopped improving because if the model is left for training over a long period of time without having any increase in its performance, then it can lead to overfitting of the model.

For Custom Convolutional Neural Network, the optimizer used is "Adam"with default learning rate (lr=0.001) and the loss function used for the model is "Categorical Cross Entropy". The batch size used is 100 images.

For INCEPTIONv3 model, the optimizer used is "RMSprop" with the learning rate of "0.0001" and the loss function used for the model is "Categorical Cross Entropy". The batch size used is 32 images.

For the transformer networks, a learning rate of 0.001 is found to give the best results. The optimizer used for these networks is "Adam" and the loss that is used is categorical cross entropy loss. The learning rate is reduced at a standard rate of once per 3 epochs to 70% of that of the previous iteration. The batch size used is 128 images.

IV. Result and Discussion

After training the model on the dataset, each model got different validation accuracy after training for a different amount of time and for different epochs.

TABLE I
Comparison Table

Table Serial No.	**Comparison Table**		
	Model Name	***Training Accuracy***	***Validation Accuracy***
1	Custom CNN	99.34	95.566
2	INCEPTIONv3	97.98	97.14
3	STN	98.82	97.66
4	LTN	99.31	97.98

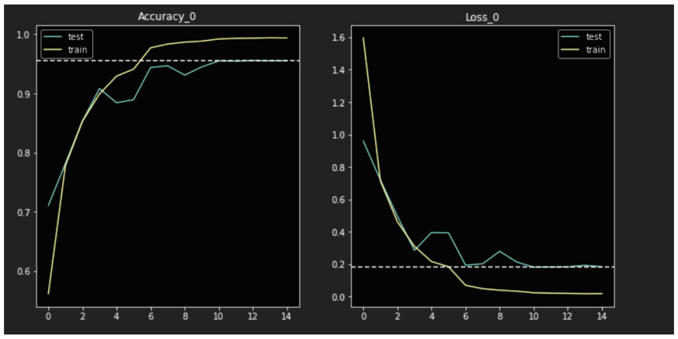

Fig. 2. Accuracy and Loss of Custom CNN

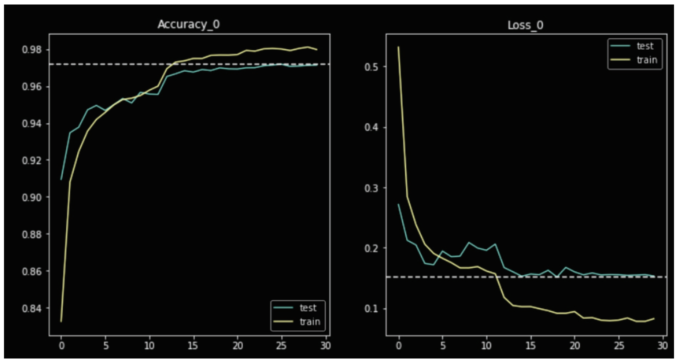

Fig. 3. Accuracy and Loss of INCEPTIONv3

The custom CNN gives the highest training accuracy of all the models that we trained but got the lowest validation accuracy of all. The difference in the training and validation accuracy of the custom CNN (as shown in the fig.2) may suggest that the model may be slightly overfitting to the data. This model is trained for only 15 epochs with all the callbacks applied .

The inceptionv3 model that is trained using transfer learning is trained for 30 epochs. The InceptionV3 got the least training accuracy(as shown in fig.3) as compared to the other architectures in this study but the validation accuracy is better than

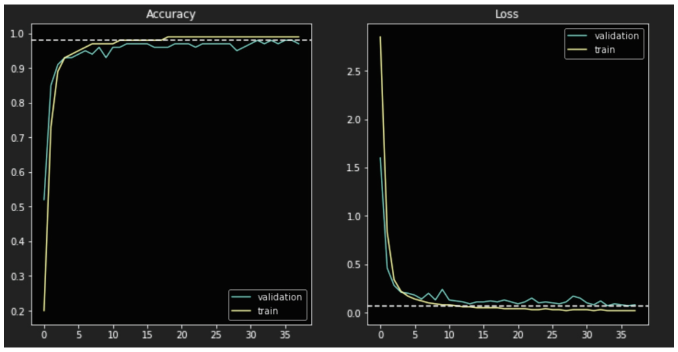

Fig. 4. Accuracy and Loss of Large Transformer Network

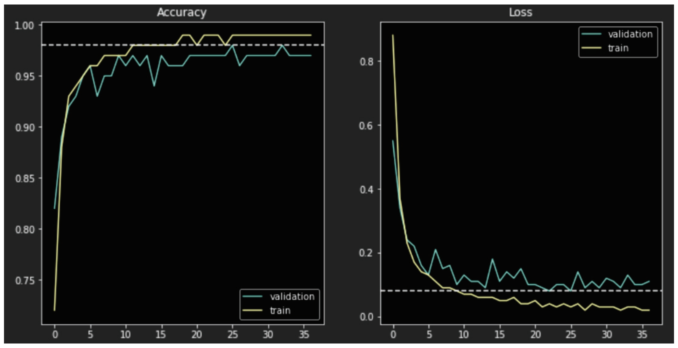

Fig. 5. Accuracy and Loss of Small Transformer Network

that of the Custom CNN and also the training accuracy and validation accuracy of INCEPTIONv3 are very close to each other which rules out the possibility of model overfitting.The Large Transformer Network(LTN)is trained for 38 epochs(as shown in fig.4), is able to achieve the highest validation accuracy of all the models that are tested in this study. Seeing that we could achieve such a high accuracy with LTN which already had only a fraction of the number of parameters of that of the other models(3,499,046 compared to 28,903,100 of the custom CNN model), we created STN with only 549,926 parameters to see if we could produce any comparable results. The model was trained for 36 epochs(as shown in fig.5). It is able to achieve a validation accuracy comparable to that of the LTN. STN outperforms the traditional CNN approaches in our study based on the validation accuracy metric.

V. Conclusion

Convolutional Networks have played a major role in the past for all tasks related to image processing whether the task is related to classification, augmentation, description. But The results of our study show us how visual transformers are the next best thing in computer vision.Though transformers were introduced for accommodating the task of Natural language processing, recent studies have shown their potential application in computer vision tasks. For our application which requires the model to be available on minimal resources, the small transformer network(STN) seems to be the best fit. The large transformer network is also a great fit though it dwarfs the STN model when taking size into consideration, it gives the highest accuracy in our study. The proper application of transformers in computer vision tasks needs more research as this architecture is in its early days.

References

[1] Sue Han Lee, Hervé Goëau, Pierre Bonnet, Alexis Joly, New perspectives on plant disease characterization based on deep learning, Computers and Electronics in Agriculture, Volume 170, 2020, 105220, ISSN 0168-1699,

[2] Srdjan Sladojevic, Marko Arsenovic, Andras Anderla, Dubravko Culibrk, Darko Stefanovic, ”Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification”, Computational Intelligence and Neuroscience,2016

[3] Trivedi J., Shamnani Y., Gajjar R. (2020) Plant Leaf Disease Detection Using Machine Learning. In: Gupta S., Sarvaiya J. (eds) Emerging Technology Trends in Electronics, Communication and Networking. ET2ECN 2020. Communications in Computer and Information Science, vol 1214. Springer, Singapore

[4] Jayme G.A. Barbedo, Factors influencing the use of deep learning for plant disease recognition, Biosystems Engineering,Volume 172,2018,Pages 84-91,ISSN 1537-5110,

[5] Bengio, Y. Lecun, Yann. (1997). Convolutional Networks for Images, Speech, and Time-Series.

[6] Kaiming He and Xiangyu Zhang and Shaoqing Ren and Jian Sun 2015 Deep Residual Learning for Image Recognition http://arxiv.org/abs/1512.03385

[7] Hussain M., Bird J.J., Faria D.R. (2019) A Study on CNN Transfer Learning for Image Classification. In: Lotfi A., Bouchachia H., Gegov A., Langensiepen C., McGinnity M. (eds) Advances in Computational Intelligence Systems. UKCI 2018. Advances in Intelligent Systems and Computing, vol 840. Springer, Cham.

[8] Mohanty Sharada P., Hughes David P., Salathé Marcel,”Using Deep Learning for Image-Based Plant Disease Detection”,Frontiers in Plant Science,vol.7, 2016

[9] AlexNet ,Krizhevsky, Alex, Ilya Sutskever, andGeoffrey E. Hinton,”ImageNet Classification with Deep ConvolutionalNeural Networks”, Advances in Neural Information Processing Systems 25 (NIPS 2012)

[10] Christian Szegedy and Wei Liu and Yangqing Jia and Pierre Sermanet and Scott Reed and Dragomir Anguelov and Dumitru Erhan and Vincent Vanhoucke and Andrew Rabinovich,”Going Deeper with Convolutions”, 2014,https://arxiv.org/abs/1409.4842

[11] 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, Miami, FL, 2009, pp. 248-255, doi: 10.1109/CVPR.2009.5206848.

[12] Ashish Vaswani and Noam Shazeer and Niki Parmar and Jakob Uszkoreit and Llion Jones and Aidan N. Gomez and Lukasz Kaiser and Illia Polosukhin, ”Attention Is All You Need”, 2017,https://arxiv.org/abs/1706.03762

[13] Bichen Wu and Chenfeng Xu and Xiaoliang Dai and Alvin Wan and Peizhao Zhang and Zhicheng Yan and Masayoshi Tomizuka and Joseph Gonzalez and Kurt Keutzer and Peter Vajda,”Visual Transformers: Token-based Image Representation and Processing for Computer Vision”,2020

[14] Christian Szegedy and Vincent Vanhoucke and Sergey Ioffe and Jonathon Shlens and Zbigniew Wojna,”Rethinking the Inception Architecture for Computer Vision”, 2015, https://arxiv.org/abs/1512.00567

[15] Konstantinos P. Ferentinos, ”Deep learning models for plant disease detection and diagnosis”,Computers and Electronics in Agriculture,Volume 145, 2018,Pages 311-318, ISSN 0168-1699,