Classification of Maize leaf diseases from healthy leaves using Deep Forest

Jatin Arora[1], Utkarsh Agrawal[2],Prerna Sharma[3],*

[1]Maharaja Agrasen Institute of Technology, Delhi, India
Email: jatin51997@gmail.com
[2]Maharaja Agrasen Institute of Technology, Delhi, India
Email: utkrsh.agrawal@gmail.com
[3]Maharaja Agrasen Institute of Technology, Delhi, India
Email: prernasharma@mait.ac.in
*Corresponding Author: Prerna Sharma, Email: prernasharma@mait.ac.in

Abstract

Apart from being relied upon for feeding the entire world, the agricultural sector is also responsible for a third of the global Gross-Domestic-Product (GDP). Additionally, a majority of developing nations depend on their agricultural produce as it provides employment opportunities for a significant fraction of the poor. This calls for methods to ensure the accurate and efficient diagnosis of plant disease, to minimize any adverse effects on the produce. This paper proposes the recognition and classification of maize plant leaf diseases by application of the Deep Forest algorithm. The Automated novel approach and accurate classification using the Deep Forest technique are a significant step-up from the existing manual classification and other techniques with less accuracy. The proposed approach has outperformed Deep Neural models and other traditional machine learning algorithms in terms of accuracy. It justifies its low dependency on extensive Hyper-parameter tuning and the size of the dataset as against other Deep Learning Models based on neural networks.

Keywords

Deep Forest, Maize leaf, Disease classification, Agriculture, Image classification, gcForest

1. Introduction

A nation's economy depends upon many factors; one of the imperative factors is the success of farming. Diseases are common in plants and their successful detection and timely diagnosis is essential to support growth in the agriculture sector. Agroindustry focuses a lot on the diagnosis of plant diseases, and large sums of money have been spent to deal with such problems.

Maize belongs to the Poaceae grass family and is the third most important cereal crop in the world. Though maize is not eaten directly, it is used to make several products such as corn starch, corn syrup and ethanol. Maize plant leaves suffer from a variety of infections, and the three most prevalent maize leaf diseases are the northern corn leaf blight disease, common rust disease & grey leaf spot disease.

The fungus Exserohilium turcicum is responsible for the northern corn leaf blight disease. The primary striking symptom for this disease is the large grey cigar-shaped lesions that form on the leaf's surface. Moderate to cool temperatures, and a relatively high humidity level act as a catalyst for this disease. The sample of such disease-stricken leaves is in **Table 1**.

Table 1. Northern leaf blight disease


| Early Stage | Advance Stage |
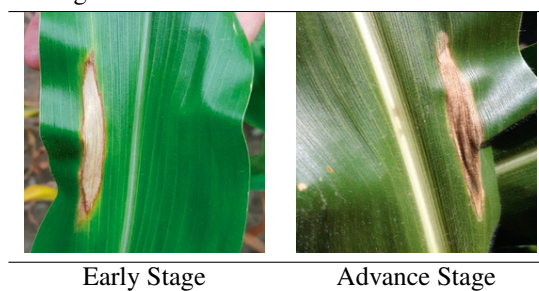
Common rust disease is yet another maize disease favoured by high humidity levels and cold temperatures. In this disease, a number of small tan spots develop on both the surfaces of the leaf and as a result, the photosynthesis of the leaf reduces drastically. The same has been shown in **Table 2**.

Table 2. Common rust disease.

| Early Stage | Advance Stage |
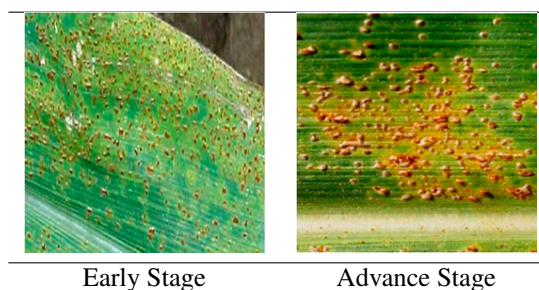
The Gray leaf spot (GLS), caused by Cercospora zeae-maydis fungal pathogen, is one of the most significant yield-limiting foliar diseases found in the maize plant. Ever since being first reported in the 1970s, it has posed a serious threat to maize production worldwide, with a significant impact in large areas of Africa and the U.S. Corn Belt. The symptoms of this disease characterized by linear (and rectangular) lesions on the lower surface of the leaf in the early stage, which later turns into rust spots as depicted in **Table 3**.

Table 3. Gray leaf spot disease.

| Early Stage | Advance Stage |
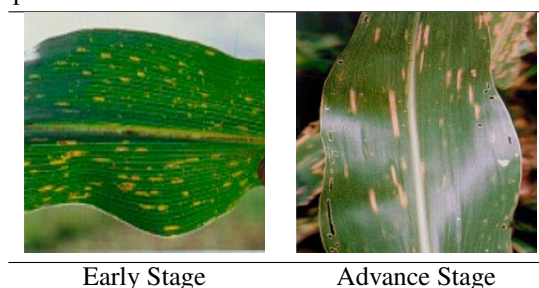
Healthy maize leaves have been depicted in **Table 4**. Four different samples have been picked for demonstration.

Table 4. Healthy maize leaves
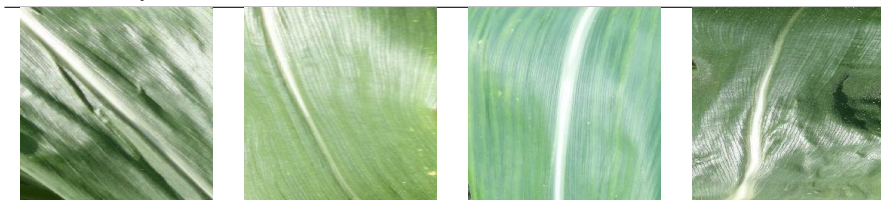
The key points of the presented research work are:

- This paper proposes the recognition and classification of maize plant leaf diseases by application of the Deep Forest algorithm.
- The approach has been tested by varying the hyperparameters of DeepForest namely, Number of trees, Number of forests and Number of grains.
- The proposed model has been compared with traditional machine learning models such as SVM, random forest, Logistic regression, KNN and decision tree as well as deep models such as CNN and leNet5 to show its superiority.
- An accuracy of 96.25% was obtained by deep forest at the following values of hyperparameters, 1000 trees, 4 forests and 3 number of grains.

The sectional breakdown of the paper is as follows, following the Introduction section, related work and recent literature works have been included in Section 2. Section 3 contains the background information required to understand the working of the proposed approach. Methodology has been included in Section 4. Finally, Section 5 contains the results followed by conclusion and future works in Section 6. The references are mentioned under section 7.

2. Related work

Neural Networks and a wide array of Deep Learning models have been previously implemented for the purpose of classifying the leaf as healthy or diseased. In their paper, Mohanty et al.[1] trained a deep convolutional neural network over a large dataset of 54306 plant leaf images for classifying 26 diseases among 14 crops. The trained model achieved a high accuracy of 99.35%, owing to the large dataset. Khirade et al. [2] also trained an Artificial Neural Network for plant disease classification. However, instead of training the model directly on the image dataset, image pre-processing was done, followed by segmentation, and finally extracted morphological and textural features and trained the model on the extracted features. Another image processing approach was developed by Vijai Singh et al. [3], who proposed a novel method of using a genetic algorithm for image segmentation. In the paper, they also discussed and reviewed various approaches for plant leaf disease classification, focussing primarily on the shortcomings for each one. A successful average accuracy of 95.71% was obtained using the SVM classifier. Kamlesh Golhani et al. [4] reviewed plant disease detection aided with the help from neural networks on hyperspectral image files.

Polke et al. [5] implemented an approach using machine learning and image handling to identify leaf infections from the pictures of leaves. They extracted quantitative features such as mean, standard deviation, skewness, and wavelet transforms from the image and finally implemented the minimum distance classifier for the classification process. It proposes a large-scale approach for robotic plant sickness diagnosis. David et al. [6] compared four models, two rule-based models and two neural based models for disease classification of plant leaves .

The deep forest is a recently researched algorithm, developed as an alternative to deep learning models centred on neural networks, by developing deep learning models based on non-differentiable modules like decision trees [7]. This model was developed by Zhi-Hua

Zhou et al., and takes inspiration from the three main characteristics of NN: sufficient model complexity, layer-by-layer processing, and in-model feature transformation. In its layer-by-layer structure, each layer consists of an ensemble of decision tree forests. Owing to its fewer hyperparameters, the performance of this model is noteworthy, as observed by the authors across a wide array of datasets across different domains. Tianchi Zhou et al. [8] exploited the property that Deep forest generate new features through multi-grained scanning, and implemented this for predicting defects in software. In another research work, Lingling Zhao et al. [9] extended the application of Deep Forest for the prediction of Protein Subcellular Localization. Xianghai Cao et al. [10] worked on the basic model to develop a rotation-based deep forest (RBDF) for the classification of hyperspectral images. The authors, again, in this case, exploited the characteristic property of Deep forest to generate new features from the input features while processing from one layer to another.

Zhang et al. [11] developed a distributed version of Deep Forest for large scale processing of data. They tested out their proposed algorithm on the problem of cash-out fraud detection, which has more than 100 million samples. They demonstrated promising results with high accuracy and little parameter tuning. Shao et al. [12] proposed and implemented a novel way to apply deep forest for classification of ADHD. They extracted two types of features, namely, 1-D functional connectivity & low-frequency fluctuations' 3-D amplitude. The results depicted that their approach performed better than previously reported accuracies. Utkin et al. [13] developed a Siamese deep forest as a gcForest modification and also as an alternative to Siamese networks and neural networks which are prone to overfitting.

Another application was given by Deepak Gupta et al. [14] on an artificial plant optimization algorithm to detect infected leaves using machine learning and in [15] where he presented a novel approach on seasonal Crops Disease Prediction and Classification using Deep Convolutional Encoder Network. Neural network and it applications are vast as demonstrated by Gupta et al in [16], [17], and [18]. Finding optimal parameters is essential to a problem, one such approach is the use of Evolutionary algorithm which has been used in a wide range of applications such as for usability feature selection and evaluation of SDLC models produce quality products [19, 20] and reduce development time, for selecting cluster heads in the Mobile Ad hoc Network (MANET) [21], for prediction of protein structure based on the chemical and physical properties of each structure [22]. Detecting disease is another common and an extremely important practical aspect of employing deep neural models as demonstrated in [23], [24] and [25].

In this paper we aim to extend these studies and introduce deep forest algorithm for the application of classifying maize leaf diseases from a set of images containing four different kind of leaf diseases and healthy leaves.

3. Background

3.1. Deep Forest

The Deep Forest is an ensemble-based decision tree approach which emphases on building deep models using modules which are non-differentiable. It is built around 3 major principles which are considered to be the reasons behind the rich accomplishments of deep models. The reasons are as follows:

- Layer by Layer processing: It is considered one of the major factors since, no matter how complex the flat model becomes, the characteristics of layer by layer processing cannot be achieved.
- In-model feature transformation: Basic machine learning models work on the original set of features. However, new features are generated during the learning process of a deep model.
- Appropriate model complexity: The fact that large datasets need complex models, basic machine learning models are limited in terms of complexity, however, it is not

the case with deep models.

The overall structural working of the deep forest is divided under two broad parts Cascade Forest Structure & Multi-Grained Scanning. Cascade forest structure is employed to ensure the layer by layer processing while Multi-grained scanning allows the model to achieve sufficient complexity.

3.1.1. Cascade Forest Structure

A cascade structure is employed to represent the layer-by-layer processing of raw features. Each layer in the cascade takes input (processed information) from the previous layer and feeds it into the next layer. A layer in the structure can be defined as an ensemble of decision tree forests. It is ensured that diversity is maintained while creating ensembles by including different kinds of forests.

The working in cascading stage proceeds as follows, for a given case, an approximate of class distribution will be generated by each forest. This is done by taking into consideration the training examples and percentage of different classes at the terminal or leaf node where the particular instance falls followed by averaging across all the trees in the same forest. This has also been depicted in **Figure** 1. The approximated class distribution so obtained forms a vector of classes with the help of k-fold cross-validation, the vector is then concatenated with the original set of features. The result is then forwarded to the next cascading layer. K-fold cross-validation helps in reducing the risk of overfitting. The number of levels is determined automatically based on the performance on the validation set.
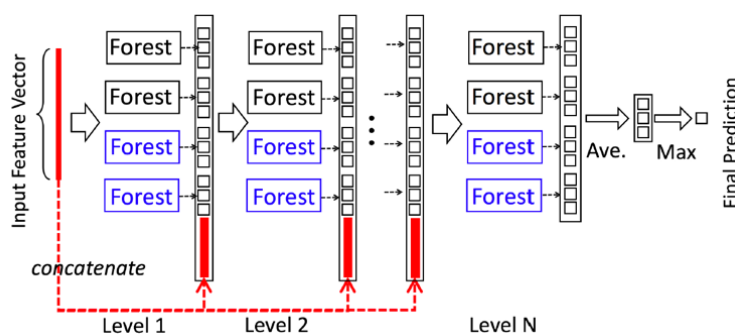

Figure 1. Cascade forest structure

A striking difference in the working of deep forest and other deep models is the ability to adaptively change the model complexity by terminating the amount of training data when tolerable. This provides a considerable advantage when working with datasets of varying sizes.

3.1.2. Multi-Grained Scanning

The cascading forest procedure is enriched with the procedure of multi-grained scanning. The inspiration behind the inclusion of the multi grained scanning procedure was that deep models are generally well suited and also good at handling feature relationships. The whole procedure has been depicted in **Figure** 2. Raw features are scanned by the sliding windows and feature vectors are produced. The feature vectors are regarded as either negative or positive instances based on the extraction from the training sample; they are then used to produce class vectors. A completely random tree forest and a random forest are trained using the instances extracted from windows having the same size. Transformed features are obtained by the concatenation of generated class vectors.

The actual label of the training sample is used to assign the instances that are extracted from the windows. Though these assignments can be incorrect, they can be attributed to

the flipping output method. Also, feature sampling can be performed if transformed feature vectors are too long. The sliding windows size is varied to obtain grained features vectors that are different.
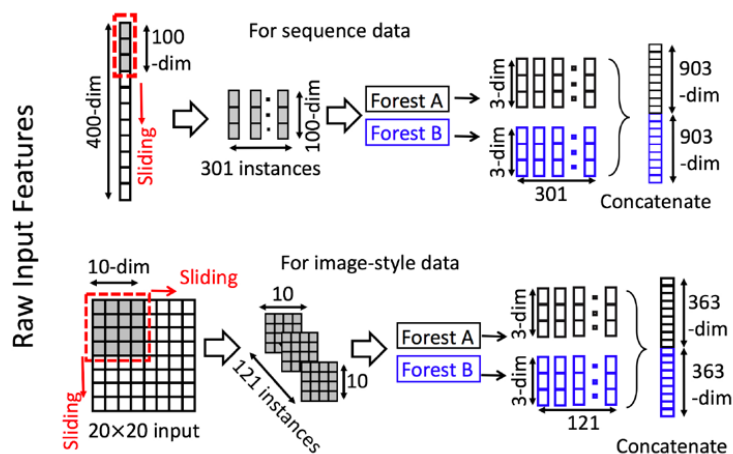

Figure 2. Multi-Grained scanning

The Deep forest has shown a lot of promise and its success can be attributed to the following factors:
- Fewer hyper-parameters
- Data-dependent tuning of model's complexity
- Less dependence on GPU

4. Methodology

During the implementation of multiple models applied on the given dataset, the RandomSearch and GridSearch python functions were used to find the optimal configuration of each model. These inbuilt functions iteratively calculate the performance for all permutations of the possible values for each parameter and return the optimum configuration. The configurations thus obtained have been listed down in detail in Table 4.

In the cascade structure of the Deep Forest, the multi-grained scanning consisted of 2 forests with 20 trees each and 4 forests with 1000 trees each in cascading. The model used three-fold cross-validation while determining the cascade levels on its own. In the implementation part, the training set splits into a growing set and an estimating set in the ratio 80:20. While the growing set is used to grow the cascade, the latter set estimates the performance. In the first part, the multi-grained scanning, feature window of size(s) d/16, d/8, d/4 are used, as mentioned in Table 4.

4.1. Dataset

The original dataset contained a total of 12332 images of maize plant leaves, each having a dimension of 512 x 512. As mentioned before, the dataset is divided into 4 categories, namely: northern corn leaf blight disease (Exserohilum), common rust disease (Puccinia sorghi) and gray leaf spot disease (Cercospora) and healthy leaves, the same have also been depicted in **Table** 5.. This dataset was reduced to 100 images per category, having the dimensionality of 64 x 64 for the following reasons:
- To deal with the high memory requirements of Deep Forest.
- To demonstrate major performance improvement of Deep Forest in comparison with other standard deep learning models for small-sized datasets.

Table 5. Sample processed dataset images

4.2. Parameters

The optimal parameters for each traditional machine learning model has been found out using Grid Search and Random Search methods. Firstly random search was employed for each model to narrow down the number of available options as there are plenty of hyper parameters and even the search space is extremely large. After employing random search, Grid search was performed to select the optimal value for the already selected parameters. However, for LeNet5 the default configuration was used which is depicted in **Figure** 3. In the case of CNN, the tuning of the network for performed with the help of Auto-sklearn which is a machine learning toolkit that can be used to auto tune a deep neural model. The overall parameters are as depicted in **Table** 6.
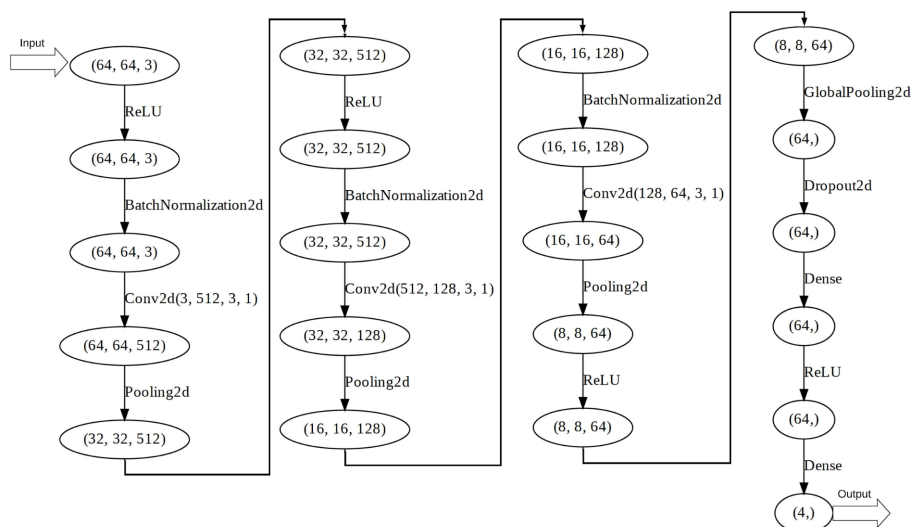

Figure 3. CNN architecture

Table 6. Optimal Parameters

S.no	Model	Hyper parameter	Value
1	DeepForest	Forest in multi-grained scanning	
		No of forest	2
		No of Trees in each forest	30
		Tree growth	Pure leaf or ≤ 20 instances
		Sliding window size	d/16, d/8, d/4
		Forest in Cascade	
		No of forest	4
		No of Trees in each forest	1000
		Tree growth	Pure leaf or ≤ 10 instances
2	CNN	Model depicted in **Figure** 3	
3	LeNet5	Default configuration	
4	SVM	kernel	rbf
		C	10
		gamma	0.001
5	RandomForest	criterion	gini
		max depth	8
		max features	log 2
		n estimators	2000
6	LogisticRegression	C	0.1
		solver	lbfgs
7	KNN	n neighbors	16
		weights	distance
8	DecisionTree	criterion	entropy
		max depth	15

4.3. Implementation Setup

4.3.1. Hardware Setup

The proposed algorithm has been tested on the system HP 348-G3 running Ubuntu 18.04.2 LTS with the following configuration:

- CPU with processor Intel® Core™ i5-6200U CPU @ 2.30GHz × 4
- 8 Gb RAM.

Additionally, NVIDIA Tesla K80 GPU available on the Google Colab environment was used for performing the proposed method.

4.3.2. Software Setup

Following Python libraries were imported and used for the same:

- NumPy: A general-purpose package for array-processing. It has multidimensional array objects and tools for working and provides high performance. NumPy if one of the most common scientific computing packages in Python.
- Scikit-learn: It supports and provides several supervised and unsupervised learning algorithms along with the support for hyperparameters. The major focus is modelling of data.
- Itertools: The library is used for handling iterators and provides tools for the same. The tools can be used in combination or, individually be themselves. It makes it

possible to construct algebra and perform efficiently when dealing with iterators algebra.

5. Results

Table 4 contains the cumulative results for various machine learning algorithms applied to the abovementioned Maize leaves image dataset, along with the optimal parameter configuration for each model. Judging by the accuracy as a performance index, it is evident that gcForest outperforms some well-established Deep Neural Networks (CNN, LeNet5) and other traditional machine learning algorithms including SVM, RandomForest, LogisticRegression, kNN and Decision Tree for image classification. It justifies its low dependency on extensive Hyper-parameter tuning and the size of the dataset as against other Deep Learning Models.

The accuracy and F1 score comparison has also been depicted in **Figure** 4 and **Table** 7, where DeepForest achieves the maximum accuracy of 96.25% and maximum F1 score of 0.9624 among all other models.

Table 7. Performance comparison

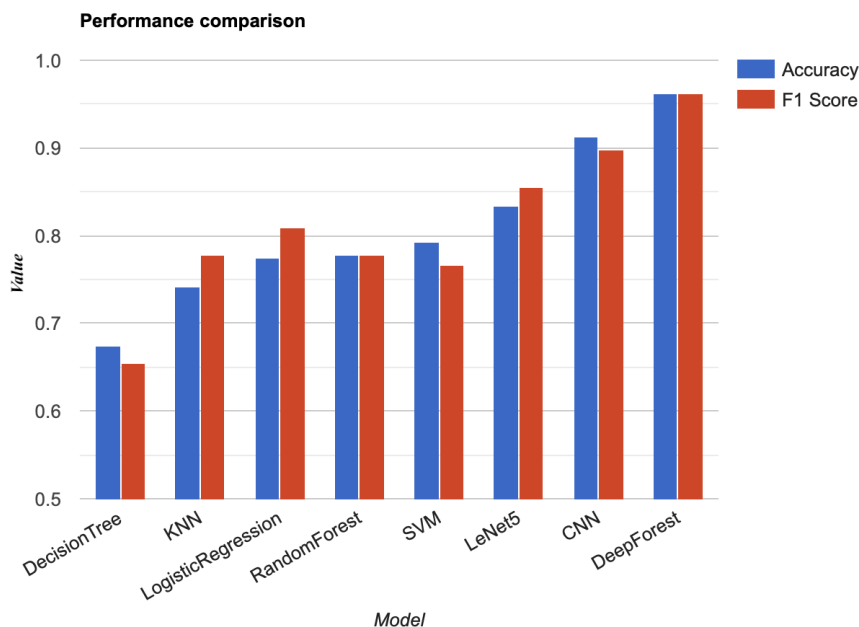| S.No	Model	Accuracy	F1 score
1	DeepForest	0.9625	0.9624
2	CNN	0.9125	0.8981
3	LeNet5	0.8346	0.8551
4	SVM	0.7925	0.7662
5	RandomForest	0.7775	0.7775
6	LogisticRegression	0.7750	0.8090
7	KNN	0.7425	0.7777
8	DecisionTree	0.6750	0.6554

Figure 4. Performance comparison

Figures 5, 6, 7 and **Tables** 8,9, 10 graphically depict the variation in performance of gcForest when the following parameters are altered:

Number of grains: The number of grains were increased gradually from 1 to 5. The best accuracy of 96.25% was obtained when number of grains is equal to 5.

Table 8. Accuracy variation with no of grains

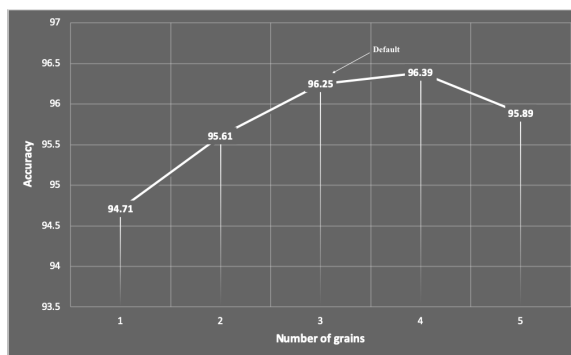| No of Grains	Accuracy
1	94.71%
2	95.61%
3	96.25%
4	96.39%
5	95.89%

Figure 5. Effect of increase in number of grains

Number of forests: The number of forests were varied from 2 to 10 in steps of 2. The maximum accuracy of 96.25% was obtained when the number of forests were 4 as compared to the default accuracy of 95.67% at 8 forest.

Table 9. Accuracy variation with no of forests

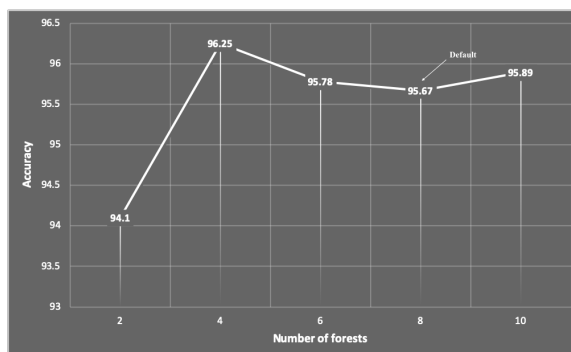| No of Forests	Accuracy
2	94.1%
4	96.25%
6	95.78%
8	95.67%
10	95.89%

Figure 6. Effect of increase in number of forests

Number of trees: The number of trees were increased 50 to 1000. The maximum accuracy of 96.25% was obtained when the number of trees were 1000 as compared to the default accuracy of 95.87% at 500 trees.

Table 10. Accuracy variation with no of trees

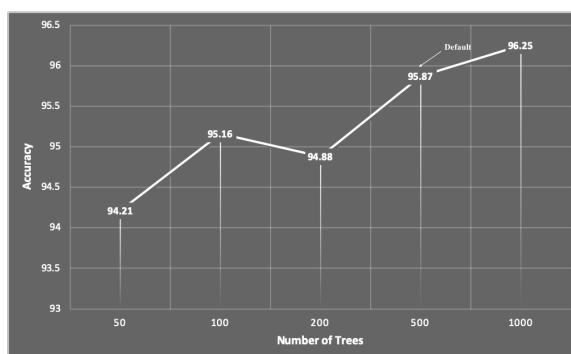| No of Trees	Accuracy
50	94.21%
100	95.16%
200	94.88%
500	95.87%
1000	96.25%

Figure 7. Effect of increase in number of trees

6. Conclusion and Future Works

The hypothesis of gcForest outperforming other standard Deep learning models centred on neural networks has been strengthened by results obtained on the Maize leaf diseases dataset. The dataset consisted of 100 images per class (for given 3 diseased leaf classes and one healthy leaf class). The efficient detection and classification of plant disease are incredibly vital for sustainable agriculture. Achieving such high accuracy of 96.25% on a mere 400 images dataset justifies how powerful gcForest model is, as it combines the architecture of neural networks and the robustness of ensembled decision trees.

The recently developed Deep Forest model proves to be highly competitive to traditional NN-based Deep learning models, thus offering an excellent alternative to Deep Learning for image-based applications. The application of gcForest on the Maize leaves' dataset opens up a plethora of other applications in the field of agriculture, where there is a never-ending opportunity for improving efficiency and saving precious time and resources.

References

[1] Mohanty, Sharada & Hughes, David & Salathe, Marcel. (2016). Using Deep Learning for Image-Based Plant Disease Detection. Frontiers in Plant Science. 7. 10.3389/fpls.2016.01419.

[2] S. D. Khirade and A. B. Patil, "Plant Disease Detection Using Image Processing," 2015 International Conference on Computing Communication Control and Automation, Pune, 2015, pp. 768-771. doi: 10.1109/ICCUBEA.2015.153

[3] Singh, Vijai & Misra, A.K. (2016). Detection of Plant Leaf Diseases Using Image Segmentation and Soft Computing Techniques. Information Processing in Agriculture. 4. 10.1016/j.inpa.2016.10.005.

[4] Golhani, Kamlesh & Balasundram, Siva & Vadamalai, Ganesan & Pradhan, Biswajeet. (2018). A Review of Neural Networks in Plant Disease Detection using Hyperspectral Data. Information Processing in Agriculture. 10.1016/j.inpa.2018.05.002.

[5] Polke, Anish & Joshi, Kavita & Gouda, Pramod. (2019). Leaf Disease Detection Based on Machine Learning. 10.1007/978-3-030-00665-5_172.

[6] Nettleton, David & Katsantonis, Dimitrios & Kalaitzidis, Argyrios & Sarafijanovic-Djukic, Natasa & Puigdollers, Pau & Confalonieri, Roberto. (2019). Predicting rice blast disease: machine learning versus process-based models. BMC Bioinformatics. 20. 514. 10.1186/s12859-019-3065-1.

[7] Zhou, Zhi-Hua & Feng, Ji. (2019). Deep forest. National Science Review. 6. 74-86. 10.1093/nsr/nwy108.

[8] Zhou, Tianchi & Sun, Xiaobing & Xia, Xin & Bin, Li & Chen, Xiang. (2019). Improving Defect Prediction with Deep Forest. Information and Software Technology. 114. 10.1016/j.infsof.2019.07.003.

[9] Zhao, Lingling & Wang, Junjie & Nabil, Mahieddine & Zhang, Jun. (2018). Deep Forest-based Prediction of Protein Subcellular Localization. Current Gene Therapy. 18. 10.2174/1566523218666180913110949.

[10] Cao, Xianghai & Wen, Li & Ge, Yiming & Zhao, Jing & Jiao, Licheng. (2019). Rotation-Based Deep Forest for Hyperspectral Imagery Classification. IEEE Geoscience and Remote Sensing Letters. PP. 1-5. 10.1109/LGRS.2019.2892117.

[11] Zhang, Ya-Lin & Li, Xiaolong & Qi, Yuan & Zhou, Zhi-Hua & Zhou, Jun & Zheng, Wenhao & Feng, Ji & Li, Longfei & Liu, Ziqi & Li, Ming & Zhang, Zhiqiang & Chen, Chaochao. (2019). Distributed Deep Forest and its Application to Automatic Detection of Cash-Out Fraud. ACM Transactions on Intelligent Systems and Technology. 10. 1-19. 10.1145/3342241.

[12] Shao, Lizhen & Zhang, Donghui & Du, Haipeng & Fu, Dongmei. (2019). Deep forest in ADHD data classification. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2941515.

[13] Utkin, Lev & Ryabinin, Mikhail. (2017). A Siamese Deep Forest. Knowledge-Based Systems. 10.1016/j.knosys.2017.10.006.

[14] Deepak Gupta, Prerna Sharma, Krishna Choudhary, Kshitij Gupta, Rahul Chawla, Ashish Khanna, Victor Hugo C. de Albuquerque, "Artificial plant optimization algorithm to detect infected leaves using machine learning", Expert Systems (Wiley) K

[15] hamparia, Aditya & Saini, Gurinder & Gupta, Deepak & Khanna, Ashish & Tiwari, Shrasti & Albuquerque, V.H.C.. (2019). Seasonal Crops Disease Prediction and Classification Using Deep Convolutional Encoder Network. Circuits, Systems, and Signal Processing. 10.1007/s00034-019-01041-0.

[16] Gupta D., Rodrigues J.J.P.C., Sundaram S., Khanna A., Korotaev V., Albuquerque V.H.C., Usability Feature Extraction Using Modified Crow Search Algorithm: A Novel Approach, Neural Computing and Applications, pp. 1-11, 2018.

[17] Jain R., Gupta D., Khanna A., Usability Feature Optimization using MWOA, In proceedings of International Conference on Innovative Computing and

Communications (ICICC), Lecture Notes in Networks and Systems, Springer, Singapore, vol. 56, pp. 453-462, 2018.

[18] Gupta D., Khanna A., SK L., Shankar K., Furtado V., Rodrigues J.J.P.C., Efficient Artificial Fish Swarm Based Clustering Approach on Mobility Aware Energy-Efficient For MANET, Transactions on Emerging Telecommunications Technologies, 2018.

[19] Gupta D., Ahlawat A., Sagar K. (2017). "Usability Prediction and Ranking of SDLC models using Fuzzy Hierarchical Usability Model", Open Engineering (Central European Journal of Engineering), ESCI, SCOPUS. Volume 7, No. 1.

[20] Gupta D., Khanna A. (2017). "Software Usability Datasets", International Journal of Pure and Applied Mathematics, SCOPUS. Volume 117, No. 15, 1001-1014.

[21] Gupta D., Sagar K. (2010) "Remote file synchronization single-round algorithm", International Journal of Computer Applications, 4(1), 32-36.

[22] Patnaik A., Gupta D. (2010). "Unique identification system", International Journal of Computer Applications, 7(5).

[23] Deepak Gupta, Jatin Arora, Utkarsh Agrawal, Ashish Khanna, Victor Hugo C. de Albuquerque, Optimized Binary Bat algorithm for classification of white blood cells, Measurement, Volume 143, 2019, Pages 180-190, ISSN 0263-2241, DOI: 10.1016/j.measurement.2019.01.002.

[24] Jatin Arora, Utkarsh Agrawal, Prayag Tiwari, Deepak Gupta, Ashish Khanna, Ensemble Feature Selection Method based on recently developed Nature-inspired algorithms, Proceedings of ICICC 2019, DOI: 10.1007/978-981-15-1286-5_39

[25] K. Shankar, S.K. Lakshmanaprabu, Deepak Gupta, Andino Maseleno, Victor Hugo C. de Albuquerque, "Optimal features-based multi kernel SVM approach for thyroid disease classification", Journal of Supercomputing (Springer), July 2018, https://doi.org/10.1007/s11227-018-2469-4