Ann. For. Res. 66(1): 4320-4332, 2023
ISSN: 18448135, 20652445

ANNALS OF FOREST RESEARCH
www.e-afr.org

TOMATO LEAF DISEASE DETECTION USING CUTTING-EDGE DEEP LEARNING ARCHITECTURES

Alampally Sreedevi,
Research scholar, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Aziznagar, Moinabad Road, Hyderabad, Telangana, India-500075
Email id: sreedevi.a@klh.edu.in

Dr. Chiranjeevi Manike
Associate Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Aziznagar, Moinabad Road, Hyderabad, Telangana, India-500075
Email id: chiranjeevi@klh.edu.in

Abstract— Agriculture is vital to India's economic progress. The major goal of agriculture the cultivation of a wide variety of valuable and necessary crops. Food safety could be adversely affected by plant diseases and generate substantial losses in production of agricultural products. For agriculture's long-term viability, disease diagnosis on the leaf is critical. Due to time constraints and the complexity of disease, it's difficult to make sense of plant diseases by hand. In the field of agricultural inputs, automatic classification of crop disease is widely required. Instance disease detection in plants is essential so it reduces the work that takes a long time of monitoring big farms and detects diseases at an early stage, limiting further plant degradation. In precision agriculture, deep learning has significantly contributed to classification and detection tasks. However, widespread adaption of these approaches and methodologies via low-cost limited devices for use in agricultural crops on a regular schedule is critical. Using the plant village dataset, which contains 87867 images divided into 38 classes to train and evaluate three deep neural network approaches for leaf disease classification. The detection accuracy of the 5-layer CNN, ResNet152 and EfficientNet-B3 architectures is about 88.32%, 93% and 97% respectively.
Keywords—Neural networks, CNN, deep learning, ResNet152, EfficientNet-B3

Introduction

This A plant disease is a major concern in agriculture as it reduces crop quality and production by causing a variety of symptoms, symptoms range from minor to severe, resulting in the complete destruction of entire crop fields, which is extremely costly [1]. Various diagnostic methods are available to prevent major losses. To precisely identify the agents that cause disease, Molecular biology and immunology methods are used. United Nations food and agriculture organization, the vast majority of the world's farms are small. However, many farmers do not have access to these methods, which involve special purpose information and expertise to implement. Food is produced by these families for a vast part of the population, but they often live in poverty and have limited access to both markets and services [2]. A great deal of studies has been focused to ensure that all these methods are accurate and accessible to the growers. By using efficient image recognition technology, image recognition can be improved, costs can be reduced and recognition accuracy can be increased. With deep learning, crop disease identification can be greatly simplified and shortened. The biggest characteristics of deep learning are its complex network structure and large amount of data samples. It is now possible to perform image recognition with deep learning technology. Due to advances in modern cameras and computer vision, morphological characteristics, green houses, and other innovations have quickly become able to compete with automated disease diagnosis models. This work aims to compare deep learning classification models such as 5-layer CNN, ResNet-152, EfficientNet-B3 for automatic diagnosis of plant leaf diseases using publicly available plant village dataset [3].

Related work

In several studies, image-based evaluation methods are more accurate and measurable than human visual analysis. Plant disease diagnosis requires a significant amount of complexity, which is accomplished by monitoring illnesses on plant leaves with the naked eye. Because of the complication, as well as the huge proportion of animals and plants which have been cultivated, even experienced pathologists frequently fail to diagnose specific phytopathological problems due to their existing phytopathological problems. As a result, they attain false conclusions and potential treatments. The detection and diagnosis of plant diseases would enhance the development of computer-assisted computing system for the agriculturist who is supposed to perform such diagnoses. Mohanty et al., 2016[4]; Yad and Gup, 2017[5] used optical investigation of diseased plant leaves was used. Michal Jasinski applied shallow VGG with XGboost on corn, potato and tomato plants. These images were obtained from plant village dataset. The accuracy obtained was 94.22%, 97.36% and 93.14% respectively. [6] A compressed version of UNet has been created using Differential Evolution to detect diseased regions on leaf images by Mohit Agarwal's work. Plant village dataset has been used to evaluate the compressed model on images of potato late blight leaves. As a result of compression, the compressed network architecture uses only 6.8% of the original space, and disease categorization inference durations are twice as fast as the original UNet architecture. Without compromising quality in the man intersection over union metric (IOU). [7] By incorporating different attention modules into a lightweight convolutional neural network, Anil Bhujel improved the performance of the models using tomato leaf images constrained in the public datasets provided by plantvillage.[8]. For the detection of infection in tomato leaves, Hariharan proposed two deep architectures were presented in his study, with the first architecture, significant features are learned through residual learning. The second architecture builds on residual deep networks by applying attention mechanisms. He conducted experiments Early blight, late blight, and leaf mold were analyzed using a plant village dataset. Using the five-fold cross-validation method, the proposed work exploited the features learned by the CNN at various processing hierarchies, and it achieved an overall accuracy of 98 percent.[9] Atila made a proposal with an F1-score of 93.05 percent, [10] the NASNet Large good classifier based on the proposed attention mechanism achieves the highest classification effect.[11]

State-of-the-art Deep Learning architectures on Plant Village Dataset

Deep Convolutional Neural Network is a multi-layer neural network. To locate the most suitable totally A convolutional framework is utilised to solve the problem of fine-grained disease categorization. We look at two designs for creating transfer learning models using photographs from the Plant village dataset: one recommends generating a shallow network, and the other offers finetuning the top layers of a pre-trained deep network.

The shallow networks comprise connected layers, softmax normalization, and convolutional layers with a few filters per layer. There are five convolutional layers, and each layer has 32 3x3 filters, a Rectified Linear Units (ReLU) activation, and the same padding. All layers are followed by a 3X3 max-pooling layer, except for the final convolutional layer, which has 512 filters. Only the dense layer, which includes 1568 filters, has a 50% dropout rate—sixty-four filters with ReLU activation and no dropout in the first ultimately connecter layer. The final fully connected layer has 38 outputs corresponding to 38 classes, which feed into the softmax layer, using the Adam optimizer with a learning rate=0.0001 to generate the probability output.

Transfer learning When it comes to transferring learning, it's vital to realise that the number of images from which we may learn is fairly restricted. By fine-tuning the parameters of a pre-trained network trained on a large dataset such as ImageNet, transfer learning is a wonderful technique to develop a powerful classification network with little data. The bottom layers of transfer learning communicate simple features used in various computer vision tasks.

For transfer learning we compare ResNet152[12] and EfficientNetB3[13] architectures. ResNet is composed of residual building blocks that are stacked one on top of the other. Each block is made up of multiple convolutional layers that are linked by a skip. Each stacked layer can now fit a residual mapping, while identity mapping is handled by skip connections. Optimizing the residual mapping is easier than optimizing the original mapping. Convolution and pooling layers make up these units. This architecture employs three VGG16 filters and accepts 300X300 pixel input images. Batch normalization was introduced between two consecutive conv2D layers, followed by the ReLU activation layer. Before moving on to the next stage, the stage result is combined with the original image and sent through a ReLU activation layer [12]. The architecture addresses the problem of degeneration: as more layers are stacked on top of each other, the validity becomes saturated and then rapidly degrades. The 152-layer version of the network is known as ResNet152. Since 2012, the models used in the ImageNet dataset have become increasingly complex, even though effectiveness has improved, many of them are ineffective in terms of compute load. The EfficientNet model is a set of CNN models that is among the best in the ImageNet classification problem, with 84.4 percent accuracy and 66 million parameters.

Materials and Experiment setup

Dataset Plant Village is an available database has over 86,000 images of diseased and healthy plant leavesorganised into 38 different classes. For our experiment, we consider images of healthy and damaged crops from all classes.

Implementation The current research project was developed fully in pyton(3.7.6), tensorflow GPU(2.5.0), Jupyter Notebook will be used as the integrated development environment (IDE).This is due to the wide number of deep learning packages available on python, such as tensor flow and keras, which makes it simple and straightforward to put into practise. Packages that are required are Pandas, NumPy, scikit and others were installed using the pip command in the course of implementation. The results were finally analysed using Matplotlib and other tools. This study's models were all compiled using NVIDIA GEFORCE GTX 1650 with 16 GB RAM.

Image pre-processing The Plant village datasets were randomly separated among training and test sets 70298 and 17572. The test set was used to evaluate the model's prediction performance on samples it had never seen before, while the training set was used to train and fit the model.

fig-1 single batch images

Model training

Convolutional neural networks have a basic architecture that includes numerous Convolutional layers, The three types of layers are pooled layers, fully connected layers, and fully connected layers. Figure 2 depicts the basic architecture of a convolutional neural network. It computes M as the window size and N as the number of input channels for an x of the ith convolutional layer's input. Deep convolutional neural networks using ReLU train multiple times quicker than those with saturating nonlinearities, since ReLU(x) = max (0, x) is employed as the activation function in our models.

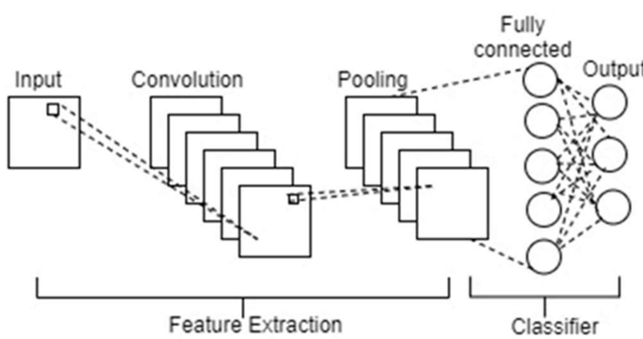

fig-2 Basic Convolutional Neural Network

Fully linked layers are built on top of the final convolutional layer. Each fully connected layer calculated ReLU(WfcX), where X is the input and Wfc is the weight matrix of the fully connected layer.

$x_{ic=ReLU(W_i * x)}$	(1)

The loss function calculates the gap between the expected result and the input label. The sum of cross entropy is defined as:

Entopy(Wgt) $= -\frac{1}{n}\sum_{x_i}^{n}\sum_{l=1}^{l}[y_{ik}\log P(x_i = l) + (1\text{-}y_{ik})\log(1\text{-}P(x_i = l))]$	(2)

Wgt denotes the weight matrices of n the number of training samples, I the training sample index, and k the class index in the above equation. P(xi=l) is the model's predicted probability of input xi belonging to the lth class, which is a function of parameters Wgt. If the ith sample belongs to the kth class, yil=1; otherwise, yil=0. As a result, W is used as a loss function parameter. The goal of network training is to determine what value of Wgt minimizes the Entropy loss function. When we use the gradient descent method, Wgt is changed iteratively.

The goal of network training is to figure out what value of Wgt minimizes the loss function Entropy. Wgt is iteratively changed when we utilize the gradient descent approach.

$Wgt_{l=}W_{l-1} - \propto \frac{\partial E(Wgt)}{\partial Wgt}$	(3)

Where is the learning rate, which is a critical factor in determining the learning step size. Learning rate's worth should be properly assessed. We flatten the output, add a dense layer, and it can recognize features that are significantly linked to output class in the final convolution layer. The pooling layer results are flattened to produce a one-dimensional vector. prior to applying the softmax layer, the final layer of the network that helps classify individual input images into one of several classes based on the network's learned properties. Dropout of 0.5 percent is used to reduce model overfitting by removing a random set of neurons from that layer. With learning-rate =0.0001, the adam optimizer is used. There are 76,092,966 trainable parameters.

ResNet-152

In ResNet152 model last 15 layers are fine-tuned by using Global Average Pooling then added a dense layer next deducted 3% of features using dropout, again one more dense layer is added for this output again 3% dropout applied and at last to get the 38 classes prediction we used Softmax activation function. In this experiment also Adam optimizer is used with the learning rate of 0.001. In fig-3 the residual block is shown[16]. When the network begins to overfit the data, we use early stopping as a training stop strategy to halt training. The network's performance is evaluated using the test set at the end of each epoch. The network will stop training if the loss value of the test sets does not improve. When the network starts to overfit the data, we use early stopping as a training stop strategy to halt training. The network's evaluation is performed using the test set at the end of each epoch. The connectivity will stop training if the loss value of the test set does not improve.

There are 6,850,598 trainable parameters and 52,812,288 non-trainable parameters in this experiment.

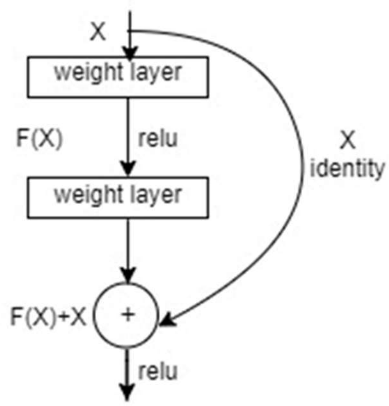

fig-3: Residual block from original paper [16]

EfficientNet-B3

We present an efficient method based on the Efficient-B3 CNN model, this particular variant of the Efficient family was chosen because it offers a good balance of computational resources and accuracy. In fig-4EfficientNet model scaling is shown [17]. Using transfer learning EfficientNetB3 model is implemented with the input resolution is 300X300, the model output parameters are 10,841,941 in this trainable parameter are 10,754,638 and the non-trainable parameters are 87,303.

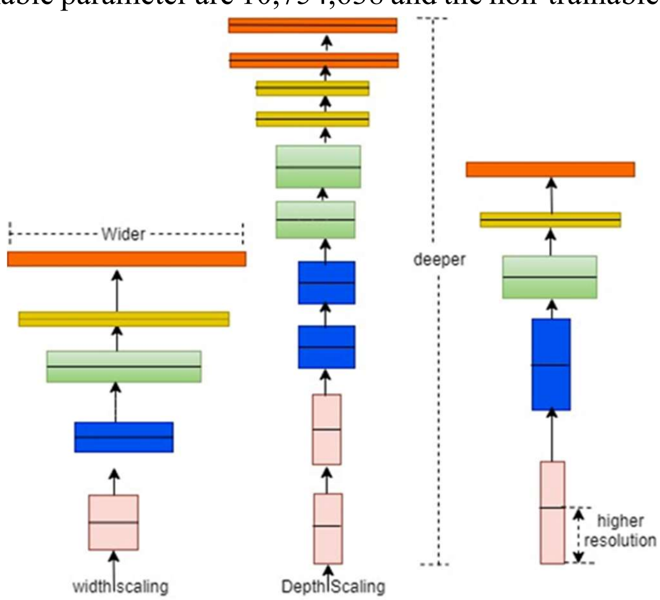

fig-4 Model scaling original paper [17]

Evaluation

Since this study is based on classification, accuracy is one of the metrics used to assess model performance. f1-score, precision and recall to determine the best model, CNN and transfer learning models are compared using these metrics. A brief explanation of each of these metrics in the context of the current study is made below.

Accuracy1: It is an accepted standard for evaluating the efficacy of classification models, and it shows how many accurate predictions the model made out of the total number of samples. It's a figure that shows what percentage of test samples the model correctly classified.

$$Accura = \frac{True_{positive} + True_{negative}}{True_{positive} + True_{negative} + False_{positive} + False_{negative}} \quad (4)$$

Precision2: It is measured mathematically as the ratio of the number of contexts in which the model was able to accurately predict the class of test data over the sum of true and false predictions. The results were calculated by the following equation below and fig-8 shows the Precision graph of the three CNN models.

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \quad (5)$$

Recall3: Another metric for analysis is the percentage of cases where the model was correct in predicting, as shown in the equation below, over the sum of correctly predicted and incorrectly classified cases, fig-9 shows the Recall graph of all the three models.

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \quad (6)$$

f1-score4: As this study is multiclass classification problem, for each class in the data, precision and recall are calculated. The harmonic mean of both of these values is used to calculate the models F1-score, which is a reliable way to compare different models. It is calculated mathematically as shown in the below equation and fig-10 shows the f1-score graph of the 3CNN models.

$$f1 - score = 2 * \frac{1}{\frac{1}{prescision} + \frac{1}{recall}} \quad (7)$$

Confusion matrix5: It shows a classification problem actual and predicted label, fig 5-7 shows the confusion matrix of 3 CNN models.

Model wise metrics:

S.No	Name of the leaf/DL Model-Accuracy	Dataset size	Image classification								
			CNN			ResNet152			EfficientNetB3		
			Precis	Recall	f1-sc	Precisi	Recall	f1-sco	Precis	Recal	f1-score
1	Apple_Apple_scab	504	0.89	0.8	0.84	0.95	0.94	0.95	1	0.99	0.99
2	Apple black rot	497	0.9	0.89	0.89	0.97	0.96	0.96	0.99	0.99	0.99
3	Apple ceder rust leaf	440	0.87	0.95	0.91	0.93	0.98	0.95	0.97	0.99	0.98
4	Apple healthy	502	0.88	0.84	0.86	0.95	0.93	0.94	0.96	1	0.98
5	Blueberry leaf-healthy	454	0.86	0.83	0.85	0.98	0.97	0.98	0.99	0.97	0.98
6	Cherry leaf-healthy	456	0.91	0.95	0.93	0.97	0.98	0.98	0.98	1	0.99
7	cherry -powdery milde	421	0.93	0.97	0.98	0.97	0.97	0.97	0.99	0.99	0.99
8	Corn Gray leaf spot	410	0.87	0.89	0.88	0.89	0.88	0.88	0.88	0.96	0.92
9	corn-common rust(Mai	477	1	0.97	0.98	1	0.98	0.99	0.98	0.99	0.99
10	Corn healthy	465	0.9	0.9	0.9	0.99	0.99	0.99	1	1	1
11	corn Northern leaf Blig	477	0.98	0.99	0.99	0.88	0.92	0.9	0.98	0.89	0.93
12	grape leaf black rot	472	0.88	0.99	0.93	0.88	0.99	0.93	1	0.9	0.95
13	Grape-Esca_Black_mea	480	0.98	0.92	0.95	0.98	0.9	0.94	0.93	1	0.96
14	grape leaf healthy	423	0.98	0.97	0.98	1	0.99	0.99	1	0.99	1
15	Grape_leaf_blight	430	0.92	0.96	0.94	0.98	0.98	0.98	0.98	0.99	0.98
16	Orange_Haunglongbin	503	0.96	0.98	0.97	0.99	0.99	0.99	1	0.99	0.99
17	Peach_bacterial_spot	459	0.91	0.88	0.89	0.98	0.93	0.96	0.96	0.98	0.97
18	Peach leaf healthy	432	0.93	0.93	0.93	0.96	0.98	0.97	0.98	0.97	0.98
19	pepper, bell bacterial	478	0.78	0.9	0.84	0.95	0.95	0.95	0.99	0.99	0.98
20	pepper, bell_healthy	497	0.86	0.72	0.78	0.96	0.94	0.95	0.99	1	1
21	Potato leaf early bligh	485	0.95	0.93	0.94	0.96	0.97	0.96	0.99	1	1
22	Potato leaf healthy	456	0.75	0.88	0.81	0.96	0.95	0.95	0.95	0.98	0.96
23	Potato leaf late blight	485	0.88	0.87	0.88	0.87	0.96	0.91	0.99	0.95	0.97
24	Raspberry healthy	445	0.92	0.94	0.93	0.99	0.99	0.99	0.83	1	0.91
25	Soyabean healthy	505	0.95	0.95	0.95	0.96	0.97	0.97	0.99	0.95	0.97
26	Squash Powdery milde	434	0.97	0.97	0.97	0.96	1	0.98	1	0.98	0.99
27	Strawberry leaf health	456	0.99	0.96	0.98	0.99	0.97	0.98	1	0.87	0.93
28	Strawberry leaf_scorch	444	0.99	0.96	0.98	0.98	0.99	0.98	0.99	1	0.99
29	Tomato_Bacterial_spo	425	0.83	0.94	0.88	0.79	0.94	0.86	0.94	0.97	0.96
30	Tomato_Early blight	480	0.6	0.78	0.68	0.82	0.8	0.81	0.96	0.92	0.94
31	Tomato_healthy	481	0.76	0.74	0.75	0.83	0.99	0.9	0.97	1	0.98
32	Tomato leaf late bligh	463	0.79	0.89	0.84	0.92	0.73	0.81	0.92	0.97	0.94
33	Tomato leaf_leaf_Mol	470	0.79	0.57	0.66	0.93	0.84	0.88	0.99	0.98	0.99
34	Tomato_Septoria_leaf	436	0.88	0.64	0.74	0.77	0.79	0.78	0.96	0.94	0.95
35	spider mites leaf	435	0.67	0.63	0.65	0.87	0.81	0.84	0.84	1	0.91
36	Tomato_Target_spot	457	0.92	0.96	0.94	0.81	0.72	0.76	1	0.73	0.84
37	aic_virus	448	0.89	0.98	0.93	0.96	0.96	0.96	0.99	1	0.99
38	ow_leaf_curl_virus	486	0.97	0.85	0.91	0.96	0.93	0.95	0.98	0.98	0.98
		17568									

	CNN			ResNet152			EfficientNetB3		
	Precis	recall	f1-sco	Precisi	recall	f1-score	Precis	recall	f1-score
Accuracy			0.88			0.93			0.97
macro avg	0.89	0.88	0.88	0.93	0.93	0.93	0.97	0.97	0.97
weighted avg	0.89	0.88	0.88	0.93	0.93	0.93	0.97	0.97	0.97

table-1: Performance of 3 CNN models

Experimental Analyses:

We explored the applicability of two state-of-the-art CNN models to categorize sick and healthy leaves of 38 spices using the plant village dataset for training and testing. To compare the performance of CNN models the accuracy, precision, sensitivity, and F1-score values were analyzed. CNN models allow for the classification of plant disease using digital imagery.

Setup for a plant village Training (70295) and testing (70296) images are randomly divided. In addition, three CNN models, a 5-layer CNN and two state-of-the-art CNN architectures, ResNet152 and EfficientNetB3, were trained to classify 38 plant leaf classes. The performance results of the CNN models were computed using precision, recall, and f1-score. Table 1 shows the test results for the three CNN models for categorising 38 classes.

The CNN model test results reveal that for the classification of the 38 classes in the Plantvillage dataset, all trained models achieved considerable accuracy, precision, recall and f1-score values. The results demonstrate that the EfficientNetB3 attained the highest precision, recall and f1-score classification values for 37 classes except Tomato target spot disease which is showing accurate precision results but recall is 73% and f1-score is 84% only, remaining all classes attained above 91% of all metrics. It was also observed that the corn healthy leaves, pepper healthy and potato leaf early blight leaves was accurately detected by 100%.

The classification rate of the EfficientNet-B3 model is comparable with the well-known ResNet152 classification model. This model achieved considerable metrics for 36 classes except corn gray leaf spot, Tomato-Early blight, Tomato target spot. All the classes attained above 90% of all metrics except these three plant leaves.

The proposed 5-layer CNN model also performed well for 33 classes, Corn gray leaf spot, Tomato early blight and Tomato-septoria-leaf-spot classes are not classifying well. Very less performance for Tomota_early_blight i.e., only 60%

The 5-layer CNN model overall accuracy is 88% and ResNet152 is 93% and EfficientNet-B3 is 97%. Overall EfficientNet-B3 model achieved better results comparatively 5-layer CNN and ResNet152. Architectures, proving to be more suitable for disease classification in real-time.

The overall accuracy of the 5-layer CNN model is 88%, ResNEt152 is 93% and EfficientNet-B3 is 97%. In general, the EfficientNet-B3 model outperformed the 5-layer CNN and ResNet152 models. All the models proven to be more suitable for real-time disease categorization.

conclusion and Future work

Deep learning architectures for image data processing and target recognition in real-time applications have rapidly grown in popularity. With three trained CNN models this study focuses on identifying the leaf diseases which are acquired from plantvillage dataset. The 5-layer CNN, ResNet142, EfficientNet-B3 architectures were developed and evaluated for the detection of leaf diseases. It is concluded that ResNet152 and EfficientNet both state-of-the-art algorithms are accurately detecting leaf diseases except tomato-target-leaf spot all 37 classes are classified with above 90% of precision, even this tomato-target-leafspot also giving 100% precision bit recall is only 73%.

Deep learning architectures which are state-of-the-art will make a big contribution to better agriculture production. In this investigation, I Identified that the tomato disease spots are too small for the models to recognize. So, I plan to build a CNN classification model and quantifying the disease severity.

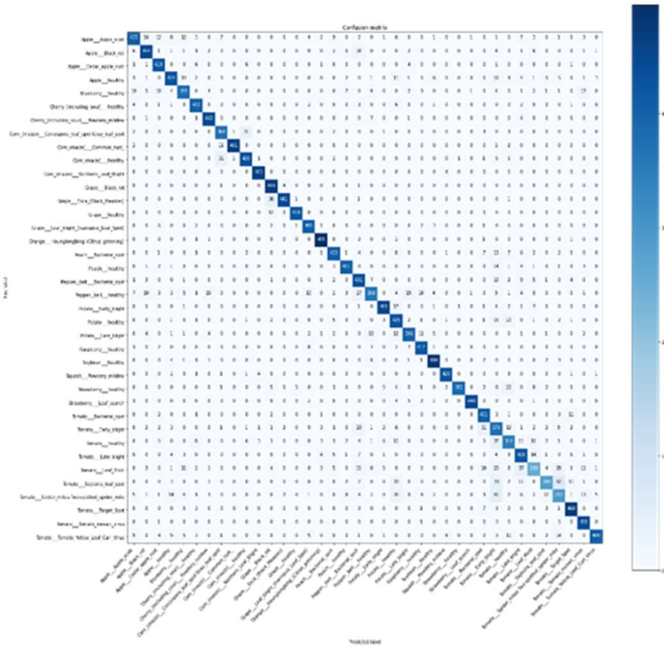

fig-5: Confusion Matrix for CNN

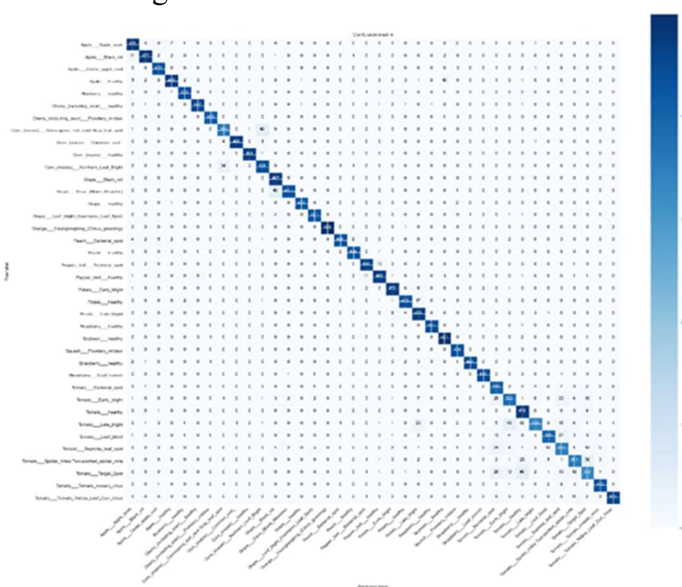

fig-6: Confusion Matrix for ResNet152

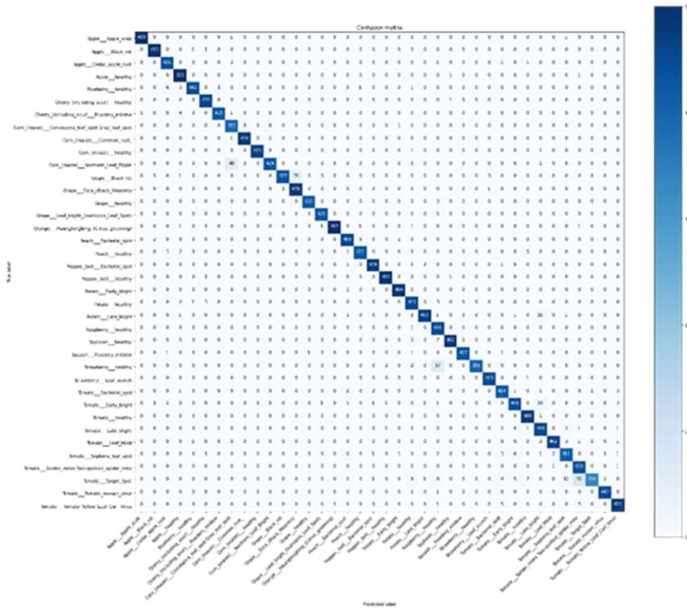

fig-7: Confusion Matrix for Efficient-B3

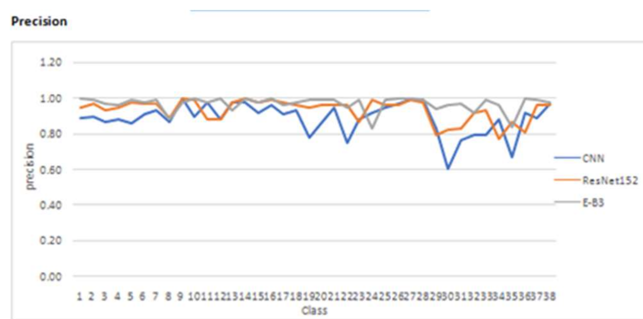

fig-8: Precision of all 3 CNN Models

fig-9: Recall of all 3 CNN Models

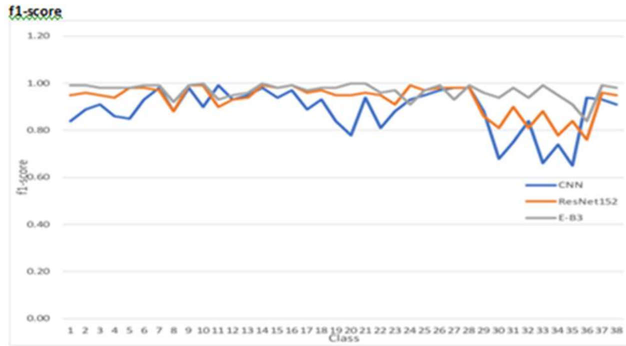

fig-10: f1-score of all 3 CNN Models

Acknowledgment
I acknowledge "KLEF –2012030001" Full Time fellowship awarded for the successful completion of the work.

References

[1] Savary, S.; Ficke, A.; Aubertot, J.-N.; Hollier, C. Crop losses due to diseases and their implications for global food production losses and food security. Food Sec. 2012, 4, 519–537. [CrossRef]
[2] Small family farmers, Family Farming Knowledge Platform, Food and Agriculture Organization of the United Nations. Available online: http://www.fao.org/family-farming/themes/small-family-farmers/en/ (accessed on 22 June 2019).
[3] Mohanty, S.P.; Hughes, D.P.; Salathé, M. Using deep learning for image-based plant disease detection. Front. Plant Sci., 7, 1419. [CrossRef]
[4] Mohanty, S.P., Hughes, D.P., Salathé, M., 2016. Using deep learning for image-based plant disease detection. Front. Plant Sci. 7http://dx.doi.org/10.3389/fpls.2016. 01419. Article: 1419.
[5] Yang, X., Guo, T., 2017. Machine learning in plant disease research. Europ. J. BioMed. Res. 6–9. http://dx.doi.org/10.18088/ejbmr.3.1.2016.pp6-9.
[6] Hassan, Sk&Jasiński, Michał&Leonowicz, Zbigniew &Jasińska, Elżbieta& Maji, Arnab. (2021). Plant Disease Identification Using Shallow Convolutional Neural Network. Agronomy. 11. 2388. 10.3390/agronomy11122388.
[7]Gupta M., Ramakrishnan G. (eds) Trends and Applications in Knowledge Dis- covery and Data Mining. PAKDD 2021. Lecture Notes.
[8] Citation: Bhujel, A.; Kim, N.-E.; Arulmozhi, E.; Basak, J.K.; Kim, H.-T. A Lightweight Attention-Based Convolutional Neural Networks for Tomato Leaf Disease Classification. Agriculture 2022, 12, 228. https://doi.org/10.3390/ agriculture12020228.
[9]R., K., M., H., Anand, S., Mathikshara, P., Johnson, A., & R., M. (2019). Attention embedded residual CNN for disease detection in tomato leaves. Applied Soft Computing, 105933. doi:10.1016/j.asoc.2019.105933.

[10] Atila, Ü., Uçar, M., Akyol, K., &Uçar, E. (2020). Plant leaf disease classification using EfficientNet deep learning model. Ecological Informatics, 101182. doi:10.1016/j.ecoinf.2020.101182.

[11] Yang Guofeng,He Yong, Yang Yong, Xu BeibeiFine-Grained Image Classification for Crop Disease Based on Attention Mechanism (2020) Frontiers in plant science Volume=11, URL=https://www.frontiersin.org/article/10.3389/fpls.2020.600854, DOI=10.3389/fpls.2020.600854, ISSN=1664-462X.

[12] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[13] Tan, Mingxing, and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." International conference on machine learning. PMLR, 2019.

[14] Li, M., Zhang, T., Chen, Y., Smola, A.J., 2014. Efficient mini-batch training for stochastic optimization. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD'14. ACM Press, New York, New York, USA, pp. 661–670. https://doi.org/10.1145/2623330.2623612.

[15] https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c

[16] Tan, M. and Le, Q., 2019, May. Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning (pp. 6105-6114). PMLR.

[17] He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).