

Original Research Paper

Farmer-Friendly Mobile Application for Automated Leaf Disease Detection of Real-Time Augmented Data Set using Convolution Neural Networks

Rishiikeshwer, B.S., T. Aswin Shriram, J. Sanjay Raju, M. Hari, B. Santhi and G.R. Brindha

SASTRA Deemed University, Thanjavur India

Article history

Received: 07-07-2019

Revised: 05-08-2019

Accepted: 21-08-2019

Corresponding Author:

Brindha, G.R

SASTRA Deemed University,

Thanjavur India

Email: brindha.gr@ict.sastra.edu

Abstract: In farming, crops are prone to a wide variety of diseases. The impact of sudden climatic change has adverse effects on their growth, providing incubation to harmful viruses and bacteria. Diseases to crops imply a significant negative impact on health, economy and livelihood of the human population. According to the data from the Food and Agricultural Organization (FAO), an average of 1.3 billion tonnes of food crops succumb to such diseases annually. This paper presents an approach to prevent such diseases from propagating, by early diagnosis of such abnormalities in leaves using state of the art deep learning techniques. Convolution Neural Networks based model with additional pre-processing techniques are developed to classify the leaves into affected and healthy category. Various Deep Learning architectures and hyperparameter tuning were carried out and the resulting model produces an accuracy of up to 95% with 400 actual leaf images and up to 98% with 3600 augmented datasets. The models are trained on real-life leaf images of crops, captured from an actual agricultural field. A user-intuitive IoT Web Application is developed to capture, process and display the predicted result (disease status) from the model.

Keywords: CNN, Deep Learning, Artificial Intelligence, Plant Disease Detection, Image Processing, Web Application, Internet of Things

Introduction

Thousands of farmers in India, present in remote locations, are solely dependent on the crop's yield. Plant diseases are quite prevalent amongst crops in these regions, causing damage to the whole batch of crops. Even though farmers have experience in identifying these diseases, it becomes difficult to diagnose them at a primary stage early enough to stop them from spreading and causing sufficient damage. Various researches were focused on early disease identification systems using Machine learning techniques, in an attempt to mitigate this issue of crop damage.

Study about the impact of climatic factors on major crops was carried out in Bhopal district of MP state in (Veenadhari, 2007). Machine learning techniques were applied to predict the soil characteristics, along with GPS based technologies (Verheyen *et al.*, 2001). ML

approaches were used to generate crop protection strategies and recommendation systems with a solution to diseases (Van Evert *et al.*, 2017). Sugiura *et al.* (2016), the results imply that Machine learning algorithms can replace the visual estimation of late blight affected region. CNN based model obtained an MAE of 11.72% with a relatively small variance.

More suitable state of the art deep learning architectures, such as Faster Region-based Convolutional Neural Network (Faster R-CNN), Region-based Fully Convolutional Network (R-FCN) and Single Shot Multibox Detector (SSD) were enriched with well-built feature extractors like VGG Net and Residual Net for faster disease detection in plants to develop an early treatment technique to minimize economic losses (Fuentes *et al.*, 2017). Different image pre-processing techniques, segmentation, feature extraction, feature selection and

classification methods were tried out on citrus plant leaves. This study briefs its strengths and limitations and discusses further issues (Iqbal *et al.*, 2018). Another study proposed a methodology that involved an ontology-based approach to plant diseases and discussed the approach by analyzing rice disease pattern. This method helps in identifying plant diseases from their symptoms (Jearanaiwongkul *et al.*, 2018). Early detection of plant diseases using diverse image processing techniques and the use of ANN to distinguish healthy and diseased leaves with an accuracy of 91% (Kulkarni and Patil, 2012).

Statistical inference methods and candidate hot-spot detection were proposed to improve disease identification for endemic wheat diseases such as Septoria and detection of rust and tan spot with the help of 7 mobile devices. The results revealed AUC (Area under the Receiver Operating Characteristic–ROC–Curve) to be higher than 0.80 for all the analyzed diseases on the pilot tests under real conditions (Johannes *et al.*, 2017). Mobile app for plant disease detection, built for detection of the disease type, is conveyed as several rules concerning the color, the shape of the spots, historical weather data and other dependent factors as features (Petrellis, 2017).

Moreover, until now a farmer-friendly intelligent device has not been designed in any studies, but this proposal paves a way to design smart IoT based Mobile App for the farmers to detect plant diseases at an early phase and increase the crop yields. Few people are reluctant to get into farming or even maintain a backyard garden because of difficulty involved in treating plants in the event of a disease.

So, this work concentrates on detecting diseased plants among data of plant leaves collected in real-time and it uses the concept of Convolutional Neural Network (CNN) for image processing, which is a class of deep learning algorithm. The images underwent some basic pre-processing like rotation and calibration of image brightness. Any leaf can be diagnosed as to whether it is healthy or diseased, irrespective of the species. The work is extended to leverage the concept of the Internet of Things (IoT) by building a web application, that can take images of crops captured via a mobile device and generate the output diagnosis. This output from the application can be used to take appropriate measures on the crops.

Data and Methods

With the advancement in the IoT, that combines deep learning and elementary statistics, the areas of its usage are multitudinous. By incorporating the Deep learning methodology, Convolutional Neural Network (CNN), the

model is built by training on images of a variety of plants' leaves, given the type of disease they possess (if any). Using this trained model, we can predict the class the images belong to, that is, if they are affected or healthy.

Deep Learning is a great tool that has provided with proven real-time solutions for problems that were otherwise considered challenging to be solved by traditional methods. It has been applied in diverse domains successfully and has got the extensive scope in Image Processing. In this application, images obtained by farmers are processed to obtain useful insights. Extraction of essential features is done automatically by deep learning algorithms, from the raw data. The advantage of a Deep learning algorithm is that it can perform the feature extraction mechanism on its own and it is not crucial to have an exclusive feature extractor.

A Convolutional Neural Network (Conv Net/CNN) is a Deep Learning algorithm which can take in an input image, analyze various features and be able to classify the image into its category. It was difficult before ConvNet, as algorithms are specially engineered to do filtering. The pre-processing effort required in a ConvNet is much lower. In the Fig. 1, we have an HSV (High Saturation Value) image. Using this in the model helps in reliable predictions (distinguishing capacity between the predictor classes) and are resistant to the influence of external light intensity fluctuations on the model performance. There are many such color spaces in which images can exist Grayscale, RGB, HSV.

Using this trained CNN model, we can predict the class of a leaf image to which it belongs (infected or healthy). The accuracy of the model was improved by tuning the hyperparameters like the number of layers, type of activation function, type of gradient descent optimizer and its parameters. The whole process is given in phases, as shown in Fig. 2.

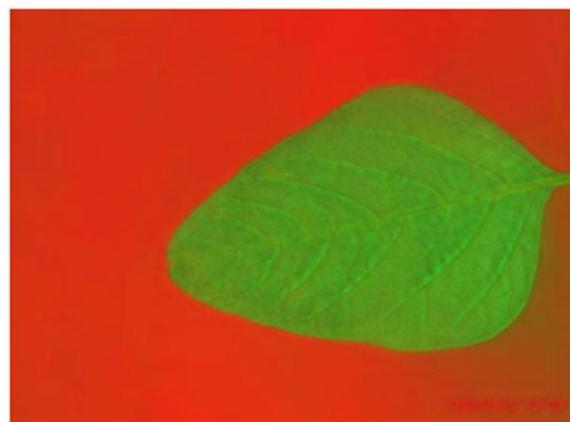


Fig. 1: HSV Image

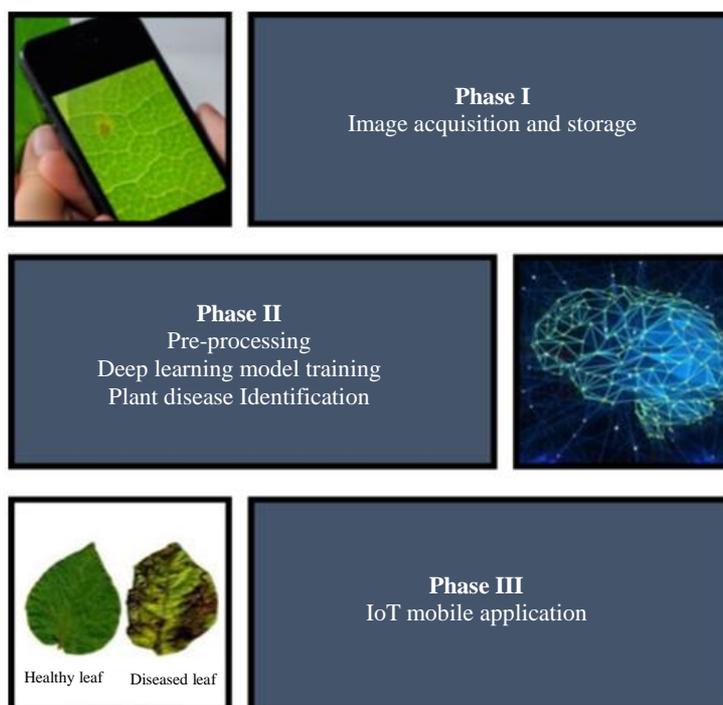


Fig. 2: Process flow of the proposed work

Phase-1

The beginning phase in the application pipeline constitutes of acquisition of data for training the classification model. The images obtained are stored in a database for further processing and can be reused to train and explore various other models:

- Distance of camera from the image – 1Foot
- Background – white sheet
- Camera hardware – Oneplus5T 16mp f/1.7 (generally any Smartphone with a camera can be used)
- No of images acquired – 100 (healthy-50, disease-50) even distribution.

Phase-2

Pre-processing of the raw data: This step serves as an essential part which contributes towards the accuracy and reliability of the model. For a Deep learning paradigm to work well, increased availability of input data is vital. Data Augmentation is done in this step:

- Through Image rotation, the data set is increased to 400 images

- Through the brightness alteration process, further, the data set is augmented to 3600 images

- Modelling of the problem with the acquired data:* The architecture of the Convolution Neural Network is designed for the classification of healthy and diseased plants. More effort is spent on this step to develop the best model suited for this problem. The detailed explanation is provided in further sections. The classifier identifies the disease level if the plants are unhealthy.
- Classification of images in real-time & model validation:* CNN performance is evaluated through the validation data set. The model is improvised by tuning the Activation functions and Optimizers, resulting in a minimum loss.

Phase-3

Design of smart, farmer-friendly IoT Mobile Application.

Web Application

The image captured from the mobile device is sent to the server to be reshaped and inverted in the pre-processing stage. The features from the captured image

will then be extracted, followed by the classification of the image by the model we developed and is translated into a well-defined output format for the end-users to utilize. Fig. 3 depicts the application layout where the GUI will request the user to capture an image of a leaf. The captured image will be processed and fed into the trained model hosted in the server. The model will then perform the prediction and generates the result. A function retrieves this result and presents it on display for the end-user to view the predicted label, whether this leaf is healthy or not. If not, what type of disease it possesses will be shown.

The architecture of the proposed model is of client/server type, wherein the mobile acts as a client and the local host acts as a server. In the current testing phase, the localhost server is used to simulate the actual cloud server environment. Initially, the connection between Client and Server is hosted locally.

Windows web browser acts as a client in this architecture, which gives options to browse and view the available models and analyzers. Once the user clicks the predict button, the image is then uploaded to the uploads folder present in the server through a POST request. Once the image is uploaded, the server runs the python script to load the required packages & trained models. The script also runs the model on this uploaded image data and returns the predicted result to the web browser again. If the user intends to analyze the acquired results, the analyzer button in the application's GUI will enable the user to view the accuracy and other statistics of the model. Since the HTTP request and reply is the communication protocol for this application, all the properties of the HTTP model are inherent in the application. Scalability of the Architecture is feasible since the application is made with several independent modules, that is, the UI can be redesigned without perturbing the Backend (Python) and vice-versa, the backend scripts can be modified for additional functionality and enhancement, without interrupting the front-end.

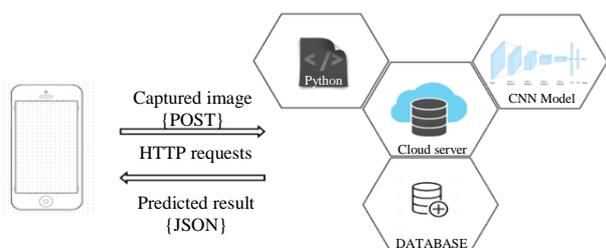


Fig. 3: Web App Process

Results and Discussion

The proposed system to develop an application for the detection of disease is created as a web application with the feature of disease detection through Deep learning. This application will be useful for farmers and people willing to get into farming or gardening, to aid in early detection of diseases in leaves and to take necessary precautions to prevent it from spreading and achieving the expected yield. The best performance of the model is estimated by training using a combination of different optimizers and activation functions on a different number of pre-processed datasets. The model is then saved as the final optimum model with the tuned hyperparameters. This model is used in the prediction of diseases in leaves of unseen samples.

Activation Function

An Activation Function (AF) is an essential element in the context of Deep learning, as it is responsible for a neuron to be fired (activated) or not in a Neural Network layer. It controls the amount of information to be passed through the networks. Mathematically, they are nonlinear transformations of an input signal that helps to learn the complex mapping between the input and the output. It is what that makes any Deep learning algorithm, a generic function approximator. They are applied to every neuron in a network. Different AFs are available to produce various complex relationships, in that we have used ReLU, tanh, Linear, Sigmoid, Softsign while training the model. The choice of AFs and their combinations are subjected to the problem statement and its requirement.

ReLU

A Rectified Linear Unit activation function, is a popularly used non-linear transformation function with an output range of $[0, \infty)$. It only activates a few neurons and computation is quicker if used.

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

Tanh

A non-linear function which is symmetric over the origin with an output range of $(-1, 1)$. It is continuous and differentiable at all points and easy while performing backpropagation. It is approximately linear when x is very nearer to the origin.

$$f(x) = \frac{2}{(1 + e^{-2x})} - 1$$

Linear

Also known as the Identity activation function, is a linear transformation function having the output directly proportional to the input. It helps to activate multiple neurons at once and can be used to predict continuous values in the output range of $(-\infty, \infty)$.

$$f(x) = x$$

Sigmoid

A widely used non-linear function, symmetric about the line $y = 0.5$ with an output range of $(0, 1)$. It is also continuous and differentiable at all points and easy while performing backpropagation. When x is closer to 0, small changes in x translates to considerable changes in y , resulting in pushing values to the extreme ends. This property is beneficial in a classification problem.

$$f(x) = \frac{1}{(1 + e^{-x})}$$

Softsign

A less popular non-linear function, with an output range of $(-1, 1)$. It is similar to tanh except it converges in a polynomial style and the gradient descent will not suffer vanishing or exploding gradient problem.

$$f(x) = \frac{x}{(1 + |x|)}$$

After experimenting with different AFs and their combinations, tanh fetched us relatively better performance and so it was considered in the final version of the model.

Optimizers

Optimizers or Optimization Algorithms are used to minimize a cost function (A convex error function of the learnable parameters- $J(\theta)$) while learning the parameters of the model. They are responsible for faster convergence of the solution to the optimization (minimization in our case) problem. In a deep learning model, Weights (W) and Bias (B) are the learnable parameters and a suitable optimizer will help determine the global minima value that would result in the lowest cost.

Let a cost function be $J(\theta)$, where θ is the parameter.

In Gradient descent, a parameter that was randomly initialized will be updated after every iteration until the minimum of the cost function is obtained.

$$\theta := \theta - \alpha \cdot \frac{\partial J(\theta)}{\partial \theta}$$

α - learning rate

SGD

In Stochastic Gradient descent, the parameter gets updated for every training sample in the dataset instead of every iteration.

Momentum:

$$\Delta\theta = \gamma \cdot \Delta\theta - \alpha \cdot \frac{\partial J(\theta)}{\partial \theta}$$

$$\theta := \theta + \Delta\theta$$

$\Delta\theta$ - gradient update

γ - coefficient of momentum

It is an augmented version of SGD, by accelerating the gradient descent in the relevant direction and diminishing the oscillations of parameters. The new gradient is the weighted average of past gradients, resulting in better control of the optimization.

Adagrad:

$$\theta := \theta - \frac{\eta}{\sqrt{\varepsilon + G_{i,i}}} \frac{\partial J(\theta)}{\partial \theta}$$

$$G_{i,i} = \sum_{r=1}^r g_r g_r^T$$

g_t = Gradient at iteration t

$G_{i,i}$ = Diagonal matrix of gradient vectors

ε = Constant to avoid divide by 0

η = Initial learning rate

The Adaptive gradient algorithm modifies each of the parameters with different learning rates. This varying learning speed will enable faster convergence.

Adam

$$\theta := \theta - \eta \frac{\hat{m}_\theta}{\sqrt{\hat{v}_\theta + \varepsilon}}$$

$$\hat{m}_\theta = \frac{m_\theta}{1 - \beta_1}$$

$$\hat{v}_\theta = \frac{v_\theta}{1 - \beta_2}$$

$$m_\theta := \beta_1 m_\theta + (1 - \beta_1) \frac{\partial J(\theta)}{\partial \theta}$$

$$v_\theta := \beta_2 v_\theta + (1 - \beta_2) \left(\frac{\partial J(\theta)}{\partial \theta} \right)^2$$

β_1, β_2 = Bias correction constants
 m_θ = First moment (mean) of the gradient
 v_θ = Second moment (variance) of the gradient

The Adaptive Moment estimation algorithm, like Adagrad, adjusts different learning rates for every parameter. It has an additional term for moving average of second moments along with the average of the gradient, giving it a robust learning experience and rapid convergence.

The results obtained from various studies by applying different combinations of optimizers and activation functions are provided in Fig. 4 and Table 1.

Here, Adam optimizer gives an overall impression of better performance on the initial dataset containing 400 images. Because, it maintains an adaptive learning rate strategy for every parameter and improves performance based on the weighted average of recent magnitudes of the gradients, whereas momentum optimizer shows significant improvement in accuracy result, on the augmented dataset containing 3600 images. The momentum algorithm is an enhanced version of SGD (Stochastic Gradient Descent), boosting its performance by accelerating the speed of learning and dampening the oscillations around the minima.

Figure 4, by using Adam optimizer and Tanh activation function, we are getting 98.08% accuracy, which is the highest of all possible combinations for 400 images. The same Adam optimizer using ReLU activation function takes the next position with an approximate 8% difference in the percentage of accuracy. While using Linear activation function, momentum optimizer performed well compared to Adam, stacking up around 84% of

accuracy. When using Softsign, Adam performed better with a magnitude of 10% in comparison with Momentum.

The images are augmented further and the data set size increased to 3600 images. The increase of samples changed the resulting scenario, in which, Momentum wins over Adam while using Tanh activation and Adagrad outperformed Adam while ReLU activation was used.

The samples are augmented by rotation. When the samples have rotated values of same images in features, Adam is struggling to identify an effective learning rate. At the same time, Momentum can handle the rotated images through stochastic gradient descent. Hence, data size plays a significant role and the reason is: Adam selects a different learning rate for each parameter and considers the gradient of the previous iterations to decide the next step. Whereas, Momentum updates only relevant parameters at every example which prevents them from getting stuck at local minima. When the sample size increases, Adam is unable to learn the parameters efficiently while it benefits Momentum for its learning style of updating parameters after every example. Hence for less than 1000 samples use Adam-Tanh and for above 1000 samples use Momentum-Tanh.

Thus, we are implementing the CNN architecture with Tanh activation function, learned using Adam optimizer as the core model for our web app for accurate categorization of healthy and diseased leaves. The GU interface of the architecture is given as a web application. The page for uploading the leaf images (Fig. 5a) and the predicted result screen (Fig. 5b) are depicted.

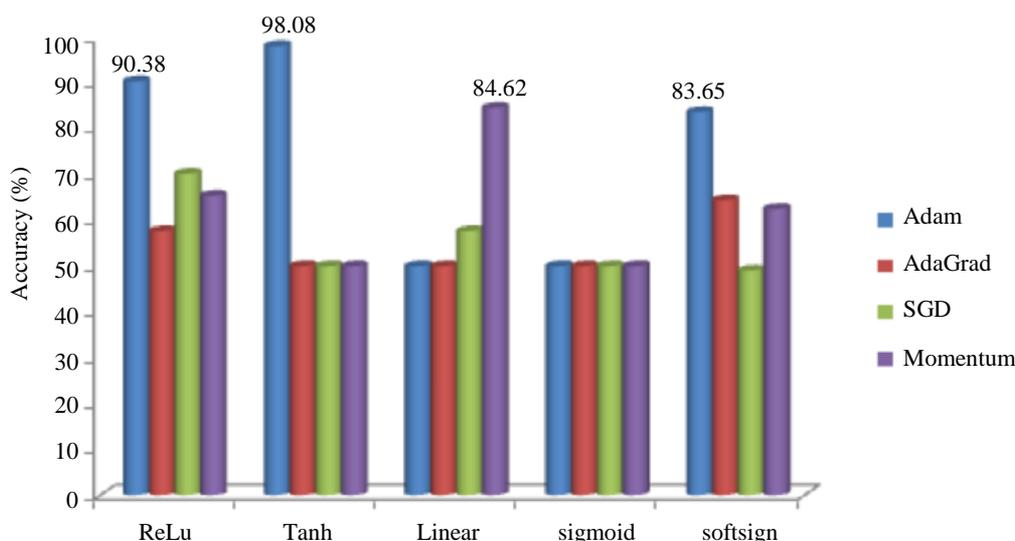


Fig. 4: Comparison – Optimizer vs. Activation Function for 400 images as dataset

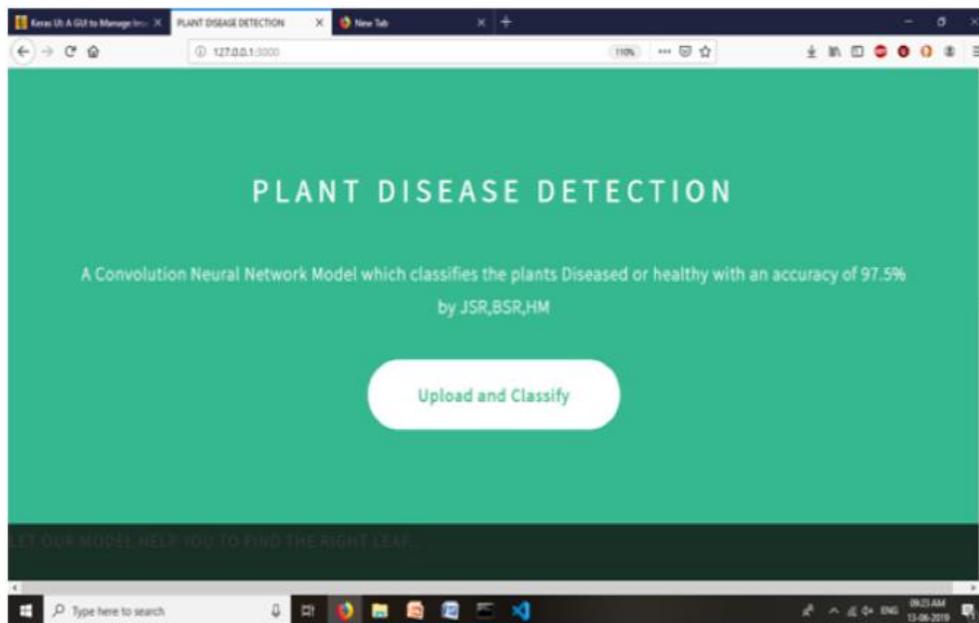


Fig. 5a: Web Application Interface image loading screen

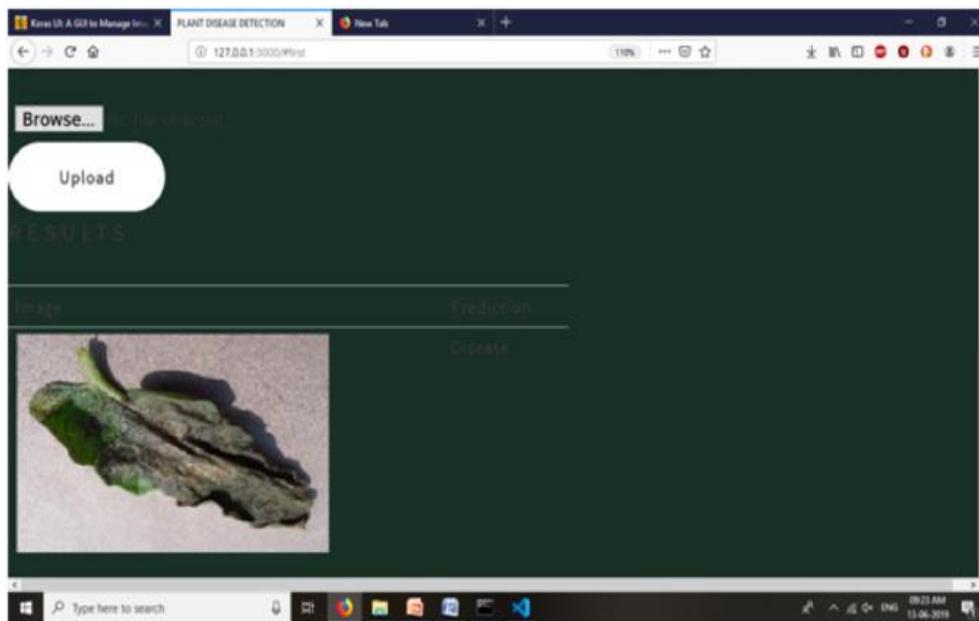


Fig. 5b: Web Application Interface image predicted result screen

Table1: Optimizer vs. Activation Function for 3600 images dataset

	ReLU	Tanh	Linear	Sigmoid	Softsign
Adam	89.4	85	93	50.8	91.4
Adagrad	91.2	78.4	89.4	50.2	49.2
SGD	90.4	49	89	50.4	55.6
Momentum	85.8	94.8	78.4	50.2	89.8

Conclusion

This paper explores the potential usage of Deep Learning in early diagnosis of leaf diseases. The outcome of this paper is a smart IoT software with Artificial Intelligence capabilities, that will monitor and detect plant diseases at an early stage. Since the developed Mobile App is intuitive (farmer-friendly), it will provide proper guidance to identify damaged and affected crops in prior, giving enough time to plan preventive measures. It also helps in reduced usage of pesticides, thus alleviating the harmful side-effects caused by them. It leads to a healthier environment and indirectly addresses the global warming issues. The impact will be followed by a sustained ecosystem, apparently improving the global economy. It can be understood from this journal that various advanced technologies are elemental in making farming more efficient and rewarding.

In the future, the same framework can be scaled and enhanced to address multiple shortcomings in farming. These technologies will become quintessential in the future of farming and agriculture. The quality and quantity of crop yields will be taken care of with less human/manual supervision. Hence, resulting in a healthier society and thereby promoting the development of a nation.

Acknowledgement

This work was completed as a summer internship project (May 2019 to Jun 2019) at FIRST-TBI of SASTRA Deemed University. We are thankful for the opportunities, facilities and the high-end systems provided to us, which helped in completing the project efficiently and on time. We thank the Food and Agricultural department of Tamil Nadu for testing our Web application in a real agricultural field and providing with motivating feedback.

Funding Information

Funding was not necessary for this work since the systems and other facilities are provided by FIRST-TBI of SASTRA Deemed University.

Author's Contributions

Brindha, G.R.: Ideas and modules of project, Structural design of paper.

Rishiikeshwer, B.S: Data set collection, Interface design, manuscript content writing.

Hari, M: Coding of CNN and grammar checking

Sanjay Raju, J.: Pre-processing of images, web application code, formatting.

Aswin Shriram, T.: Fine tuning CNN code, Full revision based on reviewers' comment, addition of references, images.

Santhi, B.: Algorithm selection, web app design and review ideas.

Ethics

There are no ethical issues in publishing this manuscript.

References

- Veenadhari, S., 2007. Crop productivity mapping based on decision tree and Bayesian classification. Unpublished M. Tech Thesis submitted to Makhanlal Chaturvedi National University of Journalism and Communication, Bhopal.
- Verheyen, K., D. Adriaens, M. Hermyand and S. Deckers, 2001. High-resolution continuous soil classification using morphological soil profile descriptions. *Geoderma*, 101: 31-48.
DOI: 10.1016/S0016-7061(00)00088-4.
- Van Evert, F.K., S. Fountas, D. Jakovetic, V. Crnojevic and I. Travelsand *et al.*, 2017. Big data for weed control and crop protection. *Weed Res.*, 57: 218-233. DOI: 10.1111/wre.12255
- Sugiura, R., S. Tsuda, S. Tamiya and A. Itoh, 2016. Science direct field phenotyping system for the assessment of potato late blight resistance using RGB imagery from an unmanned aerial vehicle. *Biosyst. Eng.*, 148: 1-10.
DOI: 10.1016/j.biosystemseng.2016.04.010
- Fuentes, A., S. Yoon, S.C. Kimand and D.S. Park, 2017. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors*, 17: 20-22.
DOI: 10.3390/s17092022
- Iqbal, Z., M.A. Khan, M. Sharif, J.H Shah and M.H. urRehmanand, 2018. An automated detection and classification of citrus plant diseases using image processing techniques: A review. *Comput. Electronics Agric.*, 153: 12-32.
DOI: 10.1016/j.compag.2018.07.032

- Jearanaiwongkul, W., C. Anutariyaand and F. Andres, 2018. An Ontology-based Approach to Plant Disease Identification System. Proceedings of the 10th International Conference Advances Information Technology, Dec. 10-13, Bangkok, Thailand, DOI: 10.1145/3291280.3291786
- Kulkarni, A.H. and A. Patil, 2012. Applying image processing technique to detect plant diseases. Int. Modern Eng. Res., 2: 3661-3664.
- Johannes, A., A. Picon, A. Alvarez-Gila, J. Echazarra and S. Rodriguez-Vaamonde *et al.*, 2017. Automatic plant disease diagnosis using mobile capture devices, applied on a wheat use case. Comput. Electronics Agric., 138: 200-209. DOI: 10.1016/j.compag.2017.04.013
- Petrellis, N., 2017. Mobile application for plant disease classification based on symptom signatures. Proceedings of the 21st Pan-Hellenic Conference Informatics, Sep. 28-30, Larissa, Greece, pp: 1-6. DOI: 10.1145/3139367.3139368