dCrop: A Deep-Learning based Framework for Accurate Prediction of Diseases of Crops in Smart Agriculture

Vishal Pallagani
Computer Science and Engineering
IIIT Naya Raipur, India.
Email: vishal16100@iiitnr.edu.in

Vedant Khandelwal
Computer Science and Engineering
IIIT Naya Raipur, India.
Email: vedant16100@iiitnr.edu.in

Bharath Chandra
Electronics and Communication Engineering
IIIT Naya Raipur, India.
Email: muppasani16101@iiitnr.edu.in

Venkanna Udutalapally
Computer Science and Engineering
IIIT Naya Raipur, India.
Email: venkannau@iiitnr.edu.in

Debanjan Das
Electronics and Communication Engineering
IIIT Naya Raipur, India.
Email: debanjan@iiitnr.edu.in

Saraju P. Mohanty
Computer Science and Engineering
University of North Texas, USA.
Email: saraju.mohanty@unt.edu

***Abstract*—The human population has been increasing exponentially and is estimated that the total population will be at 8.6 billion by mid-2030. This tremendous rise in population will demand for an increase in the production as well as consumption of food. The major menace for security of food is the crop diseases that attack the agricultural produce during their growth. The rapid identification of the diseases that affect the crops remains difficult in different parts of the world due to the limited availability of infrastructure. To help in agriculture, this research uses computer vision technology and deep learning methods for assisting prediction of diseases of crops. A deep convolutional neural network (DNN) is trained on a public dataset of 54,306 images consisting of both diseased as well as healthy plant leaves.The model is trained on the train set and is validated using the validation split of data. The trained model has achieved an accuracy of 99.24% and can identify 14 crop species and 26 diseases. As a concrete deliverable of this research dCrop is made available as a smartphone app which is built using the trained disease prediction model. The farmer can capture the crop images using the app and analyze the presence or absence of diseases, thereby demonstrating the feasibility of the solution.**

***Index Terms*—Smart Agriculture, Deep Learning, Disease Prediction, Computer Vision**

I. Introduction

In the recent years, there has been an explosive growth of Information and Communication Technologies (ICTs), due to which a majority of the cities are turning smart [1]. The World Population has increased significantly and so has the need for quality of living. It is predicted that around 70% of the world population will live in urban smart cities by the year 2050. This will result in a demand for increased sustenance, which, in turn paved its way for smart agriculture. Crop protection [2] plays a crucial role in catering to the surging demands for food quality and quantity. However, the protection of agricultural crops remains threatened by a varied number of factors including climate change, excessive weed growth, attack of pests, plant diseases and others. Around 35% of the crops are damaged every year due to crop diseases. This raises a serious concern and demands for developing an efficient solution to solve the pressing issue.

Smartphones have become a basic commodity in every household. The smartphones are now clocked with high computing power and boast of a wide range of sensors. The camera sensors being used in the phones are capable of producing high quality HDR+ photos, which can be used for vision analysis. The number of smartphone users worldwide is assumed to reach a mark of 3.8 billion [3]. This penetration is not just limited to the youth of the country, but also the elderly (55+ of age) population, making smartphones ubiquitous [4] in the future. In this paper, the demonstration of deploying a production grade deep learning disease prediction model for use in the smartphones is discussed. The deployed model is embedded into an app, dCrop for use by the farmer.

Machine Vision and Object Classifiers have undergone immense advancements in the recent times. The PASCAL VOC Challenge [5] and the Large Scale Visual Recognition Challenge [6] based on the ImageNet dataset have been extensively used as the baseline for tackling various vision related problems. In the past few years, CNNs have achieved the status of being able to produce state of the art results for developing image classifiers. The neural networks using back propagation faced the problem of running into vanishing gradient when more layers are added to make it deeper. The weights in the network are updated on the basis of partial derivatives obtained from the error function of the previous layers. These partial derivatives can sometimes tend to become very small, thereby nullifying the role of updates in the weights. A solution to tackle this problem is making use of Residual Neural Networks [7]. ResNets use skip connections in layers, thus, connecting two non adjacent layers. ResNets have been a natural go-to choice of network architecture due to its state-of-the-art performance on image recognition. Using the architecture of

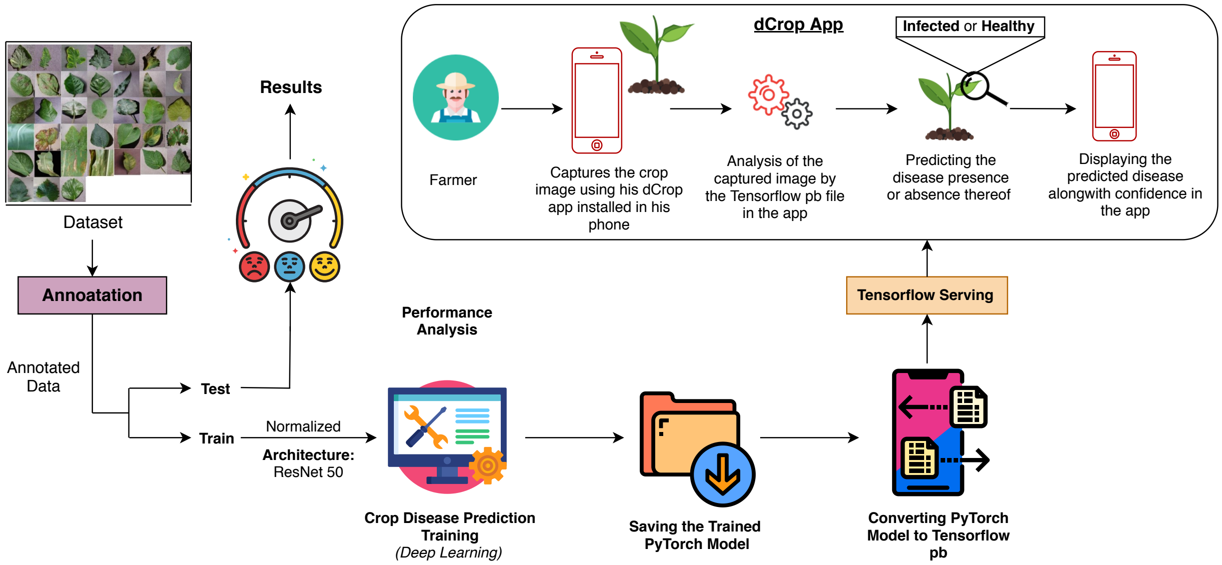


Fig. 1: Overview of the proposed system

ResNet 50 in this paper, a deep learning based Crop disease prediction system is proposed using a library of fast.ai, which is built on PyTorch.

The major contributions of the current paper are as follows:

- Proposed a robust deep learning model for crop disease prediction
- Further, a trained PyTorch model converted into tensorflow .pb file, is loaded into an android app for live prediction of disease.
- Finally, an dCrop mobile application is developed which can be used by the farmers to detect diseases in the captured crop's leaf image.

The remaining of the paper is organized in the following manner. A detailed review of the traditional methods used for the detection of crop diseases in presented in Section II. An overview of the proposed system and deployment of production grade app for the same is detailed in Section III. Section IV gives an overview of the performance analysis of the developed deep learning app. Finally, Section V concludes the paper.

II. Related Works

The topic of Agriculture Security has been a quite trending research area in the last decennial. Researchers have explored various solutions for disease prediction from crop data. Some of those works proposed in the last 5 years is presented in this section.

A method for the recognition of diseases has been proposed based on backpropagation networks in [8]. Four diseases in wheat and grape crops is predicted by the system. To predict the diseases the model is trained using Backpropagation method. The problem of overfitting araised, even when the achieved accuracy is 100%.

The Data Mining Techniques have been used for prediction in [9]. To predict the diseases the solution uses Random Forest Method and Decision Tree Method , out of which a better accuracy is found with the performance of Random Forest Method as compared to Decision Tree Method.

A Prediction Model for Automated Leaf Disease Detection has been proposed in [10]. On a total of 535972 leaf disease images, system has implemented KMeans algorithm. An accuracy above 50% is achieved by the solution.

A leaf disease detection and classification system for soyabeans has been proposed in [11]. A dataset of size more than 4000 has been considered. To classify the images an Support Vector Machine classifier is used. However, for each problem to attain higher accuracy in Support Vector Machine, the large parameters requirements needs to be changed for every specific problem.

A Automated Irrigation System with disease prediction has been proposed in [12]. To predict the diseases Hidden Markov Model is used. Results can be viewed and irrigation can be handled remotely using the smartphone. Yet HMM was unable to capture the high level correlation features.

III. The Proposed Novel System of dCrop

dCrop, provides the farmer with a mobile app which can be used to predict the crop diseases by exploiting the high computing power and the camera sensors of the smartphones. This forms the next wave of advancement in smart agriculture. The overview of the proposed system is shown in Fig.1, which is explained in detail in the following sub-sections.

1) Dataset: Dataset plays a major role in determining the performance of the trained deep learning model. Until recently, there were not many public datasets consisting of crop diseases. This problem is addressed by the Plant Village project which has led to the collection of thousands of healthy as well as diseased crop images. The Plant Village dataset [13] which consists of 54,306 images of leaves and can identify 38 different diseases has been used to construct the deep learning model.

2) Annotation of the Dataset: All the 54,306 images are classified into folders consisting of crop images of a particular disease. The folders are then named same as the disease. The reason for doing this is because ImageDataBunch class present in fast.ai offers a simple mechanism to extract the class labels from the names of the folders. The data is then split for training and testing the model.

3) Splitting the Dataset: ImageDataBunch class of fastai simplifies the process of slicing the dataset into training and validation sets. The training set consists of examples from the dataset which are used for learning to fit the parameters for training the image classifier model. The training set is used to find the optimal weights using the back-propagation rule. The validation set is used the fine tune the parameter of the classifier being trained. It helps in finding the end-point for the back propagation algorithm. For testing the trained model, we use image sets of both healthy and diseased crop diseases and analyze the performance of the trained model. The next phase involves using the train dataset to develop the crop disease prediction model.

4) Crop Disease Prediction Training: The aim of training is to fit a model that best suits our dataset. The Learner class has a function `create_cnn` to create the model. The advantages of using CNN is the capability of capturing and learning relevant features from the image, which is calculated by the following expression:

$$G[m,n] = (f * h)[m,n] \tag{1}$$

$$= \sum_j \sum_k h[j,k]\, f[m-j, n-k] \tag{2}$$

In the above expressions, input image is denoted by f and the kernel or filter (small matrix of numbers) is denoted by h. The indexes of rows and columns of the result matrix are denoted by m and n.

A pre-trained network architecture needs to be used which acts as a building block for training the crop disease prediction model. There are various pre-thained architectures available, such as, ResNet 50, AlexNet, etc. The advantage of training the model for a specific application on top of an existing pre-trained model is adding to the existing knowledge. This is known as Transfer Learning and makes the model more intelligent. The image classifier being developed is trained on top of ResNet 50, AlexNet and RestNet 34 to find the better architecture, which will be explained in detail in Section IV. Out of the three, ResNet 50 has achieved the maximum accuracy of 98.73% and the model has been trained for 4 epochs using the method `fit_one_cycle` of fastai. ResNet 50 performs the best because:

- It can go deeper without degradation in accuracy and error rate. This is achieved through injecting identity mappings.
- It has the ability to learn the residuals so that the predictions are close to the actuals.

The network can be represented using the following expression:

$$H(x) = F(x) + x \tag{3}$$

$$F(x) = W2 * relu(W1 * x + b1) + b2 \tag{4}$$

In Eqn. (3), $H(x)$ is a mapping function, $F(x)$ and x simultaneously represent the stacked non-linear layers and the identity function. $W2$ and $W1$ represent the weight matrices in the Eqn. (4). Also, $b1$ and $b2$ are the bias terms. During training period, the ResNet learns the weights of its layers during which $F(x)$ learns to adjust the predictions to match the actuals. Once the model is created, the weights need to be saved for later use.

5) Model Saving and Conversion: The weights generated from the trained model are saved using .save function present in the learner class. Since, fastai is a wrapper class of PyTorch, the saved weights will have the extension of .pth. To develop a production grade app, the best approach is to convert the .pth weights file to a Tensorflow model .pb. This conversion is done using ONNX. The obtained Tensorflow model can be embedded into an industry-grade app by the use of Tensorflow Serving.

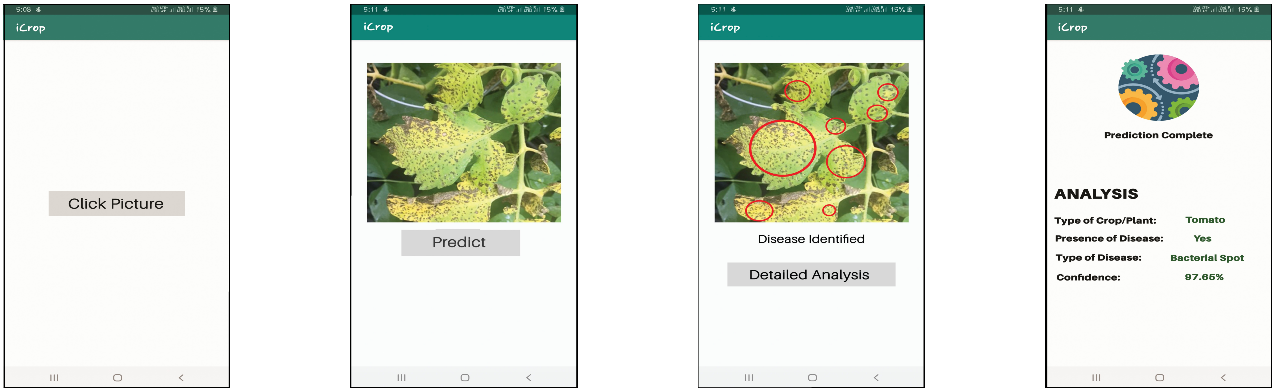
6) Tensorflow Serving: The key challenge in moving from experimenting with a deep learning model to producing it for usage is the ability to quickly serve machine learning models. Serving refers to the process of using the trained model to meet the prediction requests of the users. Tensorflow established itself as one of the best platforms to serve machine learning models as it provides a reproducible, isolated, and secure environment. The Tensorflow .pb weight file is embedded with the app for use by the farmer to predict diseases.

7) dCrop App: The dCrop app has the disease prediction classifier embedded. This ensures that the app does not need Internet connectivity to predict the diseases. The functionality available to the farmer is presented in Fig. 1. The farmer can use the app to capture the images of the crops to predict the diseases (or absence thereof). The captured images are processed by Tensorflow .pb file and outputs the predicted disease along with the confidence of the prediction. The farmer can further use this knowledge to tackle the disease before its widespread and save the crop.

IV. Experimental Results

dCrop app is developed using JAVA language in Android Studio. The app is then tested with various leaf images captured from agricultural fields. The app is useful in notifying the farmer about the type of disease and also draws a bounding box around the diseased portion of the plant. Sample analysis for one such test case along with the functionality of each app activity screen is shown in Fig. 2.

This section briefs about the performance measures of the trained image classifier and the working of the dCrop app. The deep learning model is trained using three different architectures to compare and select the best performing architecture. The accuracy produced by using each of the three architectures i.e. ResNet 50, ResNet 34 and AlexNet is shown in Table I.



(a) The User Interface shown provides with a button which opens camera and enables user to capture the image of the crop

(b) The captured image of the crop is shown and a button is provided which starts the prediction process on the captured image

(c) After the prediction is completed, the infected areas (if any) are bounded and presence of disease is notified

(d) Finally it shows the complete analysis (Type of crop, presence of disease, type of disease and confidence percentage) of the captured crop image

Fig. 2: User Interface of dCrop

TABLE I: Comparison of accuracy's of model architectures

Architecture	Accuracy for epochs = 4
ResNet 50	99.24%
ResNet 34	94.97%
AlexNet	90.63%

As ResNet 50 is offering the best accuracy, it has been chosen as the pre-trained model architecture for production. Since, the accuracy achieved is taxing, there arises a demand for performing a check on overfitting of data. To perform this analysis, the dataset is split into varying percentages of training and validation which is presented in Table II.

TABLE II: Test for overfitting of data

Train Split (%)	Validation Split (%)	Accuracry for epochs = 4
80%	20%	99.24%
60%	40%	96.19%
40%	60%	95.27%
20%	80%	93.37%

As the accuracy is consistently above 90% even when the train split is just 20%, it can be safely assumed that there has been no over-fitting.

To evaluate the performance of the trained model on the test data, a confusion matrix is plotted as shown in Fig. 3. It is calculated between the actual values and predicted values that are different class labels. The value at a cell `[i, i]` in confusion matrix states the number of correctly classified disease labels. The value at this cell will be highest for that column and row of confusion matrix of a well trained model. If the value at a cell `[i, j]` in confusion matrix is greater than one, it indicates that the `i` class is wrongly classified as `j` class, where, `i` is the actual label and `j` is the predicted label. The number of mis-classifications in predicting the

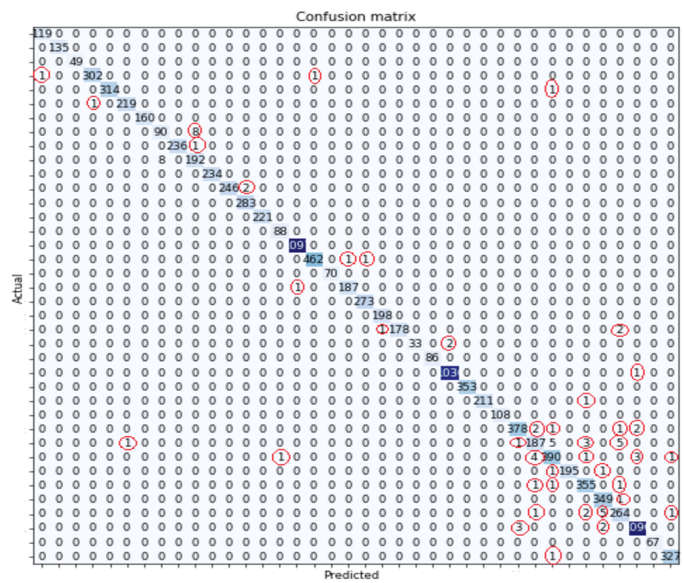


Fig. 3: Confusion Matrix for the trained model

actual class label for various diseased images can be seen circled in red in Fig. 3. From the figure, it can be seen that a total of 72 images among the entire dataset have been misclassified. On the contrary, a total of 7580 images tested have been correctly classified. Thus, with a good accuracy rate and dependable confusion matrix, the disease prediction model is well ready for production and is embedded with the app. The trained disease prediction model is a multi-class classifier which classifies a total of 38 different disease of 14 different crops. The testing conditions and the minimum requirements for live data collection from dCrop app is given in Table IV.

With the testing conditions specified in Table IV we have tested a total of 200 images on the developed app. Five images of 28 diseases and 6 images of rest 10 diseases were taken to test for a total of 14 different types of crops. Out of 38 unique

TABLE III: Comparison of Existing Solutions with dCrop

Name of the method	Algorithm Used	User Interface	Accuracy	Over fitting	Internet Connectivity	Crops/Diseases Supported
Image Recognition of Plant Disease [8]	PCA with Backpropagation	No	100%	Yes	NIL*	2/4
Sugarcane Disease Prediction [9]	Random Forest and Decision Tree	No	NIL*	NIL*	NIL*	1/4
Automated leaf Disease detection [10]	K-Means	Yes	60%	No	Required	1/4
Smart Disease Prediction [12]	Hidden Markov Model	Yes	NIL*	NIL*	Required	NIL*
dCrop	CNN	Yes	98.24%	No	Not Required	14/26

* No information available

TABLE IV: Requirements for the Proposed System

Parameters	**Testing Conditions**	**Minimum Requirement**
Camera	12.2 MP	8 MP
RAM	4GB	2GB+
Lighting	Daylight	Daylight
Memory Free	8GB	150 MB
Android Version	9.0	6.0+

diseases 36 diseases were identified correctly. And out of 200 images of the diseases 197 were identified correctly with the developed mobile app as presented in Table V.

TABLE V: Results with Real Datasets

	Correctly Classified	**Misclassified**
200 images	197	3
38 diseases	36	2

Comparisons of dCrop solution with existing solutions is shown in Table III. From this Table, it can be seen that the proposed dCrop differs from existing solutions in terms of algorithm used, user interface, accuracy of the trained model, requirement of internet connectivity and the number of crops as well as diseases it can detect.

Also, from Table IV and Table V, it can be seen that the proposed dCrop makes use of minimal resources with no additional costs and delivers a robust and reliable system for disease prediction.

V. Conclusion

In this paper, a smartphone assisted crop disease prediction model is proposed using deep convolutional neural networks. The trained deep learning model as well as the app are put to a series of tests to ensure the use of the proposed system for production.The app has been able to predict 38 different diseases with the maximum confidence. dCrop app can be used by the farmers all around the globe, without even internet connectivity. The app empowers any individual with a smartphone to help protect their crops.In the future, the app can be made to work in different regional languages for ease of use by the farmer. Also, for the identified disease, the pesticide or fertilizer to use can be predicted without internet connectivity.

References

[1] S. P. Mohanty, U. Choppali, and E. Kougianos, "Everything you wanted to know about smart cities: The internet of things is the backbone," *IEEE Consumer Electronics Magazine*, vol. 5, no. 3, pp. 60–70, Jul. 2016.

[2] S. Savary, A. Ficke, J.-N. Aubertot, and C. Hollier, "Crop losses due to diseases and their implications for global food production losses and food security," *Food Security*, vol. 4, Dec. 2012.

[3] A. Holst, "Number of smartphone users worldwide from 2014 to 2020 (in billions)," accessed: 2019-09-01. [Online]. Available: https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/

[4] A. Berenguer, J. Goncalves, S. Hosio, D. Ferreira, T. Anagnostopoulos, and V. Kostakos, "Are smartphones ubiquitous?: An in-depth survey of smartphone adoption by seniors," *IEEE Consumer Electronics Magazine*, vol. 6, no. 1, pp. 104–110, Jan. 2016.

[5] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[6] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Jun. 2016, pp. 770–778.

[8] H. Wang, G. Li, Z. Ma, and X. Li, "Image recognition of plant diseases based on backpropagation networks," in *Proc. 5th International Congress on Image and Signal Processing*, May 2012, pp. 894–900.

[9] R. Beulah and M. Punithavalli, "Prediction of sugarcane diseases using data mining techniques," in *Proc. IEEE International Conference on Advances in Computer Applications (ICACA)*, Oct. 2016, pp. 393–396.

[10] N. Goel, D. Jain, and A. Sinha, "Prediction model for automated leaf disease detection & analysis," in *Proc. IEEE 8th International Advance Computing Conference (IACC)*, Dec. 2018, pp. 360–365.

[11] S. Kaur, S. Pandey, and S. Goel, "Semi-automatic leaf disease detection and classification system for soybean culture," *IET Image Processing*, vol. 12, no. 6, pp. 1038–1048, Feb. 2018.

[12] L. Yashaswini, H. Vani, H. Sinchana, and N. Kumar, "Smart automated irrigation system with disease prediction," in *Proc. IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, Nov. 2017, pp. 422–427.

[13] S. Mohanty, *Plant Village Dataset*. [Online]. Available: https://github.com/spMohanty/PlantVillage-Dataset