An Artificial Intelligence and Cloud Based Collaborative Platform for Plant Disease Identification, Tracking and Forecasting for Farmers

Kaushik Kunal Singh
Inventure Academy
Bangalore, India
kunalsingh747@gmail.com

***Abstract* - Plant diseases are a major threat to farmers, consumers, environment and the global economy. In India alone, 35% of field crops are lost to pathogens and pests causing losses to farmers. Indiscriminate use of pesticides is also a serious health concern as many are toxic and biomagnified. These adverse effects can be avoided by early disease detection, crop surveillance and targeted treatments. Most diseases are diagnosed by agricultural experts by examining external symptoms. However, farmers have limited access to experts. Our project is the first integrated and collaborative platform for automated disease diagnosis, tracking and forecasting. Farmers can instantly and accurately identify diseases and get solutions with a mobile app by photographing affected plant parts. Real-time diagnosis is enabled using the latest Artificial Intelligence (AI) algorithms for Cloud-based image processing. The AI model continuously learns from user uploaded images and expert suggestions to enhance its accuracy. Farmers can also interact with local experts through the platform. For preventive measures, disease density maps with spread forecasting are rendered from a Cloud based repository of geo-tagged images and micro-climactic factors. A web interface allows experts to perform disease analytics with geographical visualizations. In our experiments, the AI model (CNN) was trained with large disease datasets, created with plant images self-collected from many farms over 7 months. Test images were diagnosed using the automated CNN model and the results were validated by plant pathologists. Over 95% disease identification accuracy was achieved. Our solution is a novel, scalable and accessible tool for disease management of diverse agricultural crop plants and can be deployed as a Cloud based service for farmers and experts for ecologically sustainable crop production.**

Keywords - Crop Diseases, Agriculture, Artificial Intelligence, Cloud, CNN, Mobile, Plant Pathology, Neural Networks

I. Introduction

Agriculture is fundamental to human survival. For populated developing countries like India, it is even more imperative to increase the productivity of crops, fruits and vegetables. Not only productivity, the quality of produce needs to stay high for better public health. However, both productivity and quality of food gets hampered by factors such as spread of diseases that could have been prevented with early diagnosis. Many of these diseases are infectious leading to total loss of crop yield. Given the vast geographical spread of agricultural lands, low education levels of farmers coupled with limited awareness and lack of access to plant pathologists, human assisted disease diagnosis is not effective and cannot keep up with the exorbitant requirements.

To overcome the shortfall of human assisted disease diagnosis, it is imperative to build automation around crop disease diagnosis with technology and introduce low cost and accurate machine assisted diagnosis easily accessible to farmers. Some strides have been made in applying technologies such as robotics and computer vision systems to solve myriad problems in the agricultural domain. The potential of image processing has been explored to assist with precision agriculture practices, weed and herbicide technologies, monitoring plant growth and plant nutrition management [1][2]. However, progress on automating plant disease diagnosis is still rudimentary in spite of the fact that many plant diseases can be identified by plant pathologists by visual inspection of physical symptoms such as detectable change in color, wilting, appearance of spots and lesions etc. along with soil and climatic conditions. Overall, the commercial level of investment in bridging agriculture and technology remains lower as compared to investments done in more lucrative fields such as human health and education. Promising research efforts have not been able to productize due to challenges such as access and linkage for farmers to plant pathologists, high cost of deployment and scalability of solution.

Recent developments in the fields of Mobile technology, Cloud computing and Artificial Intelligence (AI) create a perfect opportunity for creating a scalable low-cost solution for crop diseases that can be widely deployed. In developing countries such as India, mobile phones with internet connectivity have become ubiquitous. Camera and GPS enabled low cost mobile phones are widely available that can be leveraged by individuals to upload images with geo-location. Over widely available mobile networks, they can communicate with more sophisticated Cloud based backend services which can perform the compute heavy tasks, maintain a centralized database, and perform data analytics. Another leap of technology in recent years is AI based image analysis which has surpassed human eye capabilities and can accurately identify and classify images. The underlying AI algorithms use Neural Networks (NN) which have layers of neurons with a connectivity pattern inspired by the visual cortex. These networks get "trained" on a large set of pre-classified "labeled" images to achieve high accuracy of image classification on new unseen images. Since 2012 with "AlexNet" winning the ImageNet competition, deep Convolutional Neural Networks (CNNs) have consistently been the winning architecture for computer vision and image analysis [3]. The breakthrough in the capabilities of CNNs have come with a combination of improved compute capabilities, large data sets of images available and improved NN algorithms. Besides accuracy, AI

978-1-5386-9441-1/18/$31.00 ©2018 IEEE
DOI 10.1109/CCEM.2018.00016

has evolved and become more affordable and accessible with open source platforms such as TensorFlow [4].

Prior art related to our project includes initiatives to gather healthy and diseased crop images [5], image analysis using feature extraction [6], RGB images [7], spectral patterns [8] and fluorescence imaging spectroscopy [9]. Neural Networks have been used in the past for plant disease identification but the approach was to identify texture features. Our proposal takes advantage of the evolution of Mobile, Cloud and AI to develop an end-to-end crop diagnosis solution that simulates the expertise ("intelligence") of plant pathologists and brings it to farmers. It also enables a collaborative approach towards continually increasing the disease database and seeking expert advice when needed for improved NN classification accuracy and tracking for outbreaks.

II. AN END-TO- END SOLUTION FOR CROP DIAGNOSIS

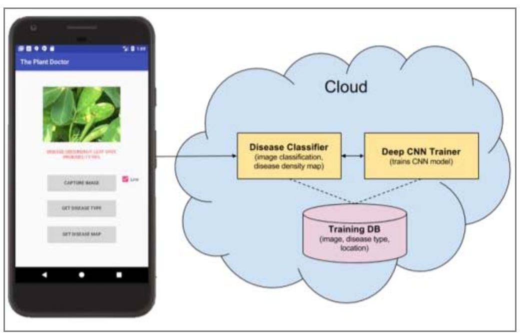

Fig. 1. System architecture with Cloud and Mobile components

Our proposed solution brings plant disease diagnostics to farmers through a Cloud based scalable collaborative platform. The platform is accessible through a mobile app that enables users to upload images of multiple parts of their plant and get the plant disease automatically diagnosed in real-time. They can also view "disease-density" map for their neighborhood showing geographical spread of diseases. The uploaded image gets classified by our AI engine into the appropriate category of disease for which a previously identified best-known-method solution is provided to the individual. Simultaneously, the geo-location of the image and a time stamp is used to tag the presence of the particular disease in that location. A collective density of diseases stored in a Cloud database is displayed on a map to show its location relative to the user. This allows the user to take preventive measures based on diseases in their neighborhood and serves as an alert for any spreading epidemic. The major components in the end-to-end system architecture of the proposed solution is shown in Fig. 1 and the description of the components is provided below.

- **Mobile App** - The mobile app contains a simplified frontend for the farmer that is easy to use and hides the complexity of the backend. It enables the user to take images of the plant (*live* mode) or choose existing images from the gallery (*offline* mode) and upload them to the Cloud backend for analysis. It allows them to get the disease type of the uploaded images with a score reflecting the probability or accuracy of classification. It also enables the user to view a disease density map of the local area (if location service is enabled on the phone). Overall, the mobile app has 8 screens (sign-in with mobile number, main page with options, capture new image, load existing image, get disease type, get disease maps, history and expert connect). Android Studio 3.1.3 was used to develop the mobile app in Java with usage of Google Camera API and Maps API. The mobile app communicates with the Cloud backend running on Amazon Web Services (AWS) over the cellular network using AWS Mobile SDK for Android.
- **Disease Classifier** – The Classifier is a standalone application running in the Cloud platform that receives the images uploaded via the mobile app and uses a trained deep Convolutional Neural Network (CNN) model to classify the disease type. The CNN model is computed by the Deep CNN Trainer and is used by the Classifier to automatically classify the uploaded images into the correct disease type. The Classifier also performs post-processing such as making a decision on whether the uploaded images should be added to the Training Database based on the classification score or sent to an agricultural expert registered on the platform for further analysis. When the classification score is greater than a preconfigured threshold, the images along with their metadata such as disease type and location of the images get added to the Training Database. In case of low classification score, the system forwards the case and seeks assistance from agricultural expert teams for manual classification which are then sent to the farmer and stored in the Training Database. Low accuracy typically occurs if the user uploads an image with an underlying disease that is so far not known to the trained CNN model, or the image quality is poor. Expert intervention in case of low classification score allows addition of new disease types which can be stored for future training runs. After the Training Database has sufficiently large number of images of the new disease category and a high classification accuracy is achieved, the Classifier can start recognizing the new disease automatically. Over time as more farmers collaborate and contribute images, it enables us to improve the accuracy for automated response to covered diseases, while using the limited expert resources to expand coverage for new diseases.
- **Deep CNN Trainer** - This Cloud application is responsible for the more intensive work of training the neural network and builds the deep CNN model that is used by the Classifier to classify images into the correct disease types. This application is run asynchronously (without any interference to the Classifier) whenever

the number of new images added to the Training Database goes beyond a pre-configured threshold. Each subsequent run of this training application works on a larger training dataset, and hence continually improves the deep CNN model used by the Classifier for more accurate disease classification. AWS was used to build the entire Cloud platform. The Disease Classifier and the Deep CNN Trainer are applications developed in Python. To make these Python applications accessible over mobile internet, they were developed using a web framework called FLASK and deployed behind an Apache Web Server running on an AWS EC2 machine (Ubuntu 16.04.2 LTS, 2 GiB memory, 8 GiB EBS volume). Disease Classifier and Deep CNN Trainer are built with TensorFlow [4], which is an open source library for Artificial Intelligence by Google.

- **Training Database -** This is a Cloud based database that stores all images used to train the deep CNN model. In addition to the images, it stores the metadata such as disease type, location of the images and time stamps. This database grows with wider use of the mobile app and as farmers upload more images taken from their fields. Growth of the Training Database allows continual retraining of the deep CNN model with larger datasets. Data in this database is also used to compute disease density relative to the user's location from collective metadata, such as disease types and image geo locations, and the generated disease density maps are rendered in the mobile app. AWS S3 was used to implement the image database and MySQL running on AWS EC2 was used to store disease metadata such as classification, treatment and location.
- **Expert Interface –** A web based expert interface has

Fig. 2. Expert dashboard with disease data visualizations

been developed that allows agricultural experts to manually classify images that get low classification score. After the expert manually classifies the image, an SMS alert is sent to the user to check the mobile app history to receive the updated classification and remedial suggestions. Another feature of this interface is that it leverages the disease metadata stored by the Cloud platform to allow the experts to render time-based and geographical visualizations of disease data as shown in Fig. 2 for analytics and monitoring purposes. Fig. 3 depicts the process flow with the sequence of steps performed by the constituent components of the platform as well as the interactions between them.

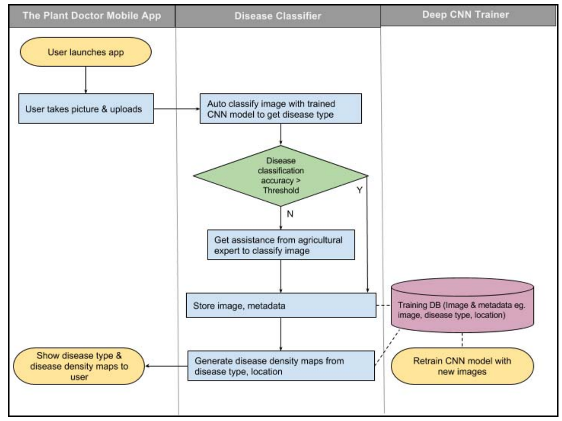

Fig. 3. Process flow of the components

III. EXPERIMENTS, RESULTS AND OBSERVATIONS

Multiple levels of experimentation were conducted to adequately simulate lab based and field based scenarios for image analysis, which forms the core aspect of this proposal. Experiments done can be broadly categorized into 3 types: Experiment 1 was conducted with training images retrieved from Google search to establish feasibility of the proposal; Experiment 2 was conducted with a large open source public dataset with images taken under controlled conditions to prove that the proposal has a high degree of accuracy even with many disease categories; and Experiments 3 and 4 were conducted with self-collected, high fidelity, high quality images from an agricultural farm to simulate real life with images of common crops such as groundnuts, tomatoes and grapes taken under natural conditions. Seasonality of crops, easy access to them, severity of diseases and their prevalence at the time of experimentation were factors in our choice of crops and our decision to perform aggressive data collection during the season.

A. Experiment 1

As a first level of experimentation, we trained a deep CNN model to build a Disease Classifier using the latest Inception-v3 architecture [10] and Python using the TensorFlow framework [4]. The goal of this experiment was to perform image recognition of different diseases of mangoes as a baseline experiment to demonstrate feasibility of the approach before progressing to a wider data set or to real data collected from fields. Images of mangoes with diseases were bulk downloaded using Google Search. Out of many mango diseases [11], four common diseases were included in this experiment as they had distinct symptoms and their images

TABLE 1. EXPERIMENT 1 TRAINING AND CLASSIFICATION RESULTS

	Training						Classification Probability Score			
Expt #	Training Steps	# training Images bacterial	# training Images mildew	# training Images phoma	# training Images red rust	test image	bacterial	Mildew	Phoma	Red Rust
1.1	250	67	70	22	37	bacterial	0.749	0.051	0.142	0.058
1.1						mildew	0.102	0.509	0.247	0.141
1.1						phoma	0.328	0.02	0.641	0.011
1.1						red rust	0.111	0.033	0.047	0.809
1.2	500	67	70	22	37	bacterial	0.863	0.036	0.076	0.025
1.2						mildew	0.075	0.697	0.166	0.061
1.2						phoma	0.322	0.013	0.658	0.005
1.2						red rust	0.068	0.024	0.023	0.883
1.3	1000	67	70	22	37	bacterial	0.927	0.018	0.042	0.013
1.3						mildew	0.051	0.824	0.093	0.031
1.3						phoma	0.26	0.007	0.731	0.002
1.3						red rust	0.031	0.011	0.01	0.946
1.4	500	65	68	20	35	bacterial	0.885	0.031	0.059	0.027
1.4						mildew	0.115	0.604	0.218	0.063
1.4						phoma	0.48	0.02	0.495	0.005
1.4						red rust	0.379	0.181	0.174	0.265
1.5	500	65	68	20	35	bacterial	0.852	0.048	0.074	0.026
1.5						mildew	0.1	0.599	0.233	0.067
1.5						phoma	0.432	0.017	0.545	0.006
1.5						red rust	0.048	0.035	0.089	0.827

were more easily available. These diseases were Bacterial Canker, Mildew Mango, Phoma Blight and
Red Rust (Fig. 4).

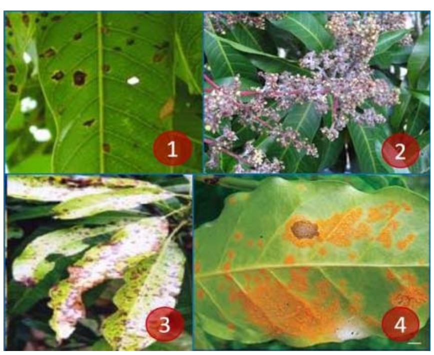

Fig. 4. Symptoms of mango diseases (source: Google)

The downloaded images were used as the training data set and the Inception based CNN model was trained (transfer learning) on this data set to teach the network to recognize the four types of mango diseases. Our model converts input images to 299X299 RGB for training and classification. Table 1 depicts the details of training and testing of image classification under Experiment 1.

67 images of bacterial canker, 70 images of mildew, 22 images of phoma blight and 37 images of red rust were used as a training data set for the CNN in Experiments 1.1 to 1.3. Different number of training steps were used (250, 500, 1000) to evaluate if test accuracy changes with the training steps. The trained model was then used to classify test images. The output of classification for each test image is an array of 4 probability scores that sum to 1. Each score contains the probability that the test image belongs to one of the four categories.

Overfitting is a potential problem with neural networks where a model may just be memorizing irrelevant details of the training images to come up with the right answers, and it may give good results on the images it's seen during training but fails on new images. To avoid the problem of overfitting, some of images were removed from the training set so that the model can't memorize them and the process of training and testing were executed and tabulated under Experiments 1.4 and 1.5 in Table 1, keeping the number of training steps constant at 500.

Table 1 demonstrates that even with a very low number of low fidelity images used for training, the classification accuracy is satisfactory. For e.g., Experiment 1.1 shows that a bacterial canker test image was classified in the right category with a probability score of 0.749. Keeping the number of training images constant, training is repeated by varying the number of training steps in 1.1, 1.2 and 1.3. Results show that as the number of training steps are increased from 250 to 500 to 1000, the score of classification improves (e.g. the scores of classification of the same bacterial canker image increases from 0.749 to 0.863 to 0.927 with number of training steps 250, 500 and 1000 respectively).

When the test images were removed from training data, the accuracy of classification of images that the trained model had not seen before was still fairly high as demonstrated by Table 1 Experiment 1.4. Classification of red rust image was incorrect (the model classified the red rust image as bacterial canker with a score of 0.379, marked in red). The experiment was rerun (Experiment 1.5) after swapping the test image of red rust with a better image from the training set (with clearer spots), and the classification score improves, with the model classifying the image as red rust with a score of 0.827. This proves that the accuracy of classification improved with the quality of test data.

Training neural networks can be computationally and time intensive (10-60 minutes for our runs), however, the trained CNN models can be used to classify images very quickly (1-3 seconds) which makes the application of neural networks in smartphone apps possible. Results captured in Experiment 1 prove that CNNs can be adopted for image classification in our use case of plant disease diagnosis and motivates further experimentation with wider and high fidelity test data.

B. *Experiment 2*

Fig. 5. PlantVillage Sample Images [5]

In the second level of experimentation, a large public dataset was used that includes images of diseased and healthy plants collected under controlled conditions by agricultural experts. This was to prove the applicability of the solution on a larger scale with more number of disease categories. PlantVillage is an open-source platform [5] for crop health and has released a public dataset of over 50,000 plant images to enable development of computer vision approaches to help solve the problem of loss of crop yields due to infectious diseases. This dataset includes curated images on healthy and infected leaves of crops. It has images of 26 diseases in 14 crops, leading to 38 possible crop-disease pairs (classes/categories labelled as c0 to c37). Fig. 5 is a collage created from samples of images taken from different categories of PlantVillage dataset to give an idea on the type of images used for experimentation.

For our experiment, 8 categories of PlantVillage images were randomly chosen for training and testing. From each category, 5 images were removed from the training set to serve as the test data. Remainder training data set from the 8 categories was used to train our Inception based CNN model, followed by the classification of test images by the trained model. Table 2 shows the statistics for training data set that was used to produce the trained CNN model.

TABLE 2. EXPERIMENT 2 TRAINING DATASET

	# Training Images for each category							
# Training Steps	c3	c4	c5	c20	c21	c22	c24	c29
1000	708	581	505	407	375	59	1912	400

Table 3 captures the output of classification of the five test images for each category using the trained CNN model.

TABLE 3. EXPERIMENT 2 CLASSIFICATION RESULTS

	Classification Probability Score							
Test image	c3	c4	c5	c20	c21	c22	c24	c29
c3_test	**0.766**	0.108	0.024	0.014	0.033	0.011	0.034	0.009
c3_test2	**0.571**	0.003	0.003	0.012	0.141	0.138	0.087	0.044
c3_test3	**0.813**	0.012	0.021	0	0.005	0.002	0.012	0.133
c3_test4	**0.988**	0.004	0.005	0	0	0	0.001	0.002
c3_test5	**0.366**	0.082	0.024	0.013	0.076	0.079	0.316	0.043
c4_test	0.016	**0.967**	0.004	0.001	0.003	0.005	0.001	0.001
c4_test2	0.008	**0.972**	0.004	0.001	0.005	0.002	0.005	0.001
c4_test3	0.028	**0.688**	0.17	0.011	0.011	0.02	0.069	0.002
c4_test4	0.006	**0.978**	0.001	0	0.002	0.002	0.01	0
c4_test5	0.009	**0.933**	0.016	0	0.001	0.007	0.031	0
c5_test	0.039	0.002	**0.868**	0.005	0.036	0	0.035	0.013
c5_test2	0.129	0.006	**0.827**	0.002	0.011	0.001	0.011	0.01
c5_test3	0.015	0.058	**0.893**	0	0.004	0.002	0.023	0.005
c5_test4	0.001	0.01	**0.965**	0.008	0.013	0	0	0.002
c5_test5	0.035	0.009	**0.935**	0.004	0.005	0.001	0.006	0.002
c20_test	0.001	0.008	0.01	**0.925**	0.045	0.008	0.001	0.001
c20_test2	0.005	0.001	0.049	**0.881**	0.051	0	0.005	0.007
c20_test3	0	0	0.001	**0.921**	0.07	0.002	0.001	0.003
c20_test4	0.001	0.001	0.004	**0.976**	0.009	0.001	0.002	0.005
c20_test5	0	0	0	**0.996**	0.002	0	0	0.002
c21_test	0.001	0.002	0	0	**0.977**	0.009	0.003	0.007
c21_test2	0.004	0	0.006	0.005	0.03	0	0	**0.954**
c21_test3	0.005	0.006	0.001	0.017	**0.918**	0.022	0	0.029
c21_test4	0.001	0.001	0.003	0.062	**0.908**	0	0.004	0.02
c21_test5	0.112	0.104	0.01	0.017	0.254	0.072	**0.398**	0.032
c22_test	0.001	0.004	0	0.003	0.026	**0.946**	0.019	0.001
c22_test2	0.027	0.032	0.001	0.005	0.019	**0.842**	0.068	0.006
c22_test3	0.001	0.001	0	0	0.005	**0.987**	0.005	0.001
c22_test4	0.032	0.035	0.011	0.004	0.284	**0.589**	0.027	0.017
c22_test5	0.001	0.001	0	0.012	0.035	**0.907**	0.042	0.001
c24_test	0.12	0.022	0.03	0.002	0.006	0.002	**0.797**	0.018
c24_test2	0.006	0.003	0	0	0.006	0.039	**0.941**	0.001
c24_test3	0.194	0.078	0.056	0.004	0.048	0.05	**0.549**	0.02
c24_test4	0.006	0.003	0	0.004	0.014	0.11	**0.859**	0.001
c24_test5	0.01	0.014	0.015	0	0.005	0.005	**0.946**	0.004
c29_test	0.134	0.007	0.002	0.003	0.148	0.01	0.016	**0.679**
c29_test2	0.005	0.001	0.003	0.037	0.014	0	0	**0.939**
c29_test3	0.003	0.003	0.002	0.004	0.083	0.001	0.001	**0.902**
c29_test4	0.007	0.001	0.043	0.015	0.3	0	0	**0.633**
c29_test5	0.001	0.002	0.005	**0.895**	0.023	0.001	0.001	0.067

Following significant observations can be drawn from this experiment with data collected under controlled conditions :

- Classification is correct for 37 out of 40 images, hence 92.5% images were classified correctly, proving that the solution will work even with a large dataset with more disease categories. 3 incorrect cases are marked in red in the table.
- The incorrect classifications are potentially due to reasons such as few categories being visually very similar (e.g. c21 and c24) and also poor quality of the test image as shown in Fig. 6 due to which the classifier fails to identify it correctly.

Fig. 6. c21_test2 that failed to classify correctly

C. Experiment 3

Groundnut was chosen as the main case study for field research and experiment in order to verify end user experience. The goal of this experiment was to simulate real life scenario with images taken in the field by users under natural conditions. Groundnut, also known as peanut, is widely grown and consumed all over the world and has significant economic importance being a rich source of edible oil and protein. We chose groundnut as a case study for field work due to the fact that 80% of the world groundnut crop is produced in developing countries where yields are usually very low and diseases have become a major obstacle to the groundnut output throughout the world [12]. China, India and US are the top three producers of groundnut globally. Although several diseases inflict groundnut crops [12], for purposes of this experiment, two major diseases of groundnut i.e. leaf spot or 'tikka' and bud necrosis, were selected for collection and analysis of field data due to their severity, large scale impact on production and widespread occurrence in India.

As part of field work, hundreds of images of healthy and diseased groundnut plants were collected from the experimental farms of the Punjab Agricultural University, Ludhiana and University of Agricultural Sciences, GKVK, Bangalore. For data collection and experimentation, three types of groundnut plants - healthy plants, plants suffering from leaf spot and plants suffering from bud necrosis - were selected from the farms. The collected images were used as training data for the CNN model to identify the three categories under experimentation - healthy, diseased with leaf spots and diseased with bud necrosis. Table 4 shows the statistics for training the CNN model for groundnuts. Total of 811 images were collected for training, out of which 243 belonged to healthy plants, 358 belonged to plants infected with leaf spot or 'tikka' disease and 210 belonged to plants infected with bud necrosis. It took 14 minutes to run the entire process of training the network to produce a trained CNN model. The trained CNN model was then used to classify a set of test images that were not part of training data.

TABLE 4. EXPERIMENT 3 TRAINING DATASET

	# Training Images for each category		
Training Steps	**Healthy**	**Leaf Spot or Tikka Disease**	**Peanut Bud Necrosis**
1000	243	358	210

For the purpose of testing with the trained CNN model, a total of 15 test images of groundnut were classified using the model, out of which 5 were healthy, 5 had symptoms of leaf spot and 5 had symptoms of bud necrosis. Fig. 7, 8 and 9 show two images each of healthy, leaf spot and bud necrosis that were used in testing the CNN model.

Fig. 7. Test images of healthy groundnut

Fig.8. Test images of Leaf Spot groundnut

Fig. 9. Test images of Bud Necrosis groundnut

Table 5 shows the results of classification of the 15 field test images with the trained CNN model. To classify each image with the trained CNN model, it took approximately 1.4 seconds.

TABLE 5. EXPERIMENT 3 CLASSIFICATION RESULTS

	Classification Probability Score		
Test image	**Healthy**	**Leaf Spot or Tikka**	**Peanut Bud Necrosis**
healthy_test1	0.974	0	0.025
healthy_test2	0.963	0.007	0.028
healthy_test3	0.975	0.012	0.012
healthy_test4	0.828	0.028	0.143
healthy_test5	0.799	0.031	0.168
leaf_spot_test1	0	0.988	0.01
leaf_spot_test2	0.001	0.99	0.007
leaf_spot_test3	0	0.983	0.016
leaf_spot_test4	0.003	0.995	0
leaf_spot_test5	0.001	0.995	0.003
bud_necrosis_test1	0	0.192	0.98
bud_necrosis_test2	0.046	0.027	0.925
bud_necrosis_test3	0.071	0.078	0.85
bud_necrosis_test4	0	0.053	0.945
bud_necrosis_test5	0.22	0.353	0.74436

Following observations can be made from the results of our main case study of Experiment 3 captured in Table 5:

- Correct classification was achieved in 100% of test scenarios, with high accuracy across all 3 categories (healthy, leaf spot, bud necrosis) proving that the accuracy will be high with correctly categorized high fidelity training data set.
- Experiment 3 results were better than Experiment 1 (downloaded Google images) as well as Experiment 2 (open source data set from controlled conditions), even though the number of training images in Experiment 2 were more in number. This could indicate that images taken in natural conditions improve training and the resulting classification.
- Within Experiment 3 categories, the classification accuracy of leaf spot or 'tikka' disease is the highest. Leaf spot category had the highest number of training images under Experiment 3 which proves that higher number of verified training images will lead to higher accuracy if all other factors are kept constant.
- Experiment 3 produced a high degree of accuracy with a comparatively smaller set of training data. We can extrapolate this to claim that even with large scale production deployment, the rate of error or ambiguity in diseases classification can be maintained low as larger scale deployment would also mean larger training datasets with user added images.
- Although the training of the CNN model takes ~14 minutes, classification of a test image using the trained CNN model is very fast (average ~1.4 seconds). This proves that the efficiency of underlying deep CNN makes it possible to achieve the complex task of image classification via a consumer facing mobile app. The training process runs independently in the Cloud and does not interfere with the run time of the image classification. The mobile user application has a simple frontend interface, and the heavy lifting of training and classification is done by the powerful AI algorithms running in the Cloud to return classification results to the user application in real time.

D. *Experiment 4*

Our goal was to expand coverage of the model to other types of local crops. Experiment 3 was extended to carry out training and testing of real-life images of grapes and tomato diseases. The goal of the grape experiment was to demonstrate that the disease classification works correctly even for very early symptoms of diseases. Many image samples of healthy grapes and with early symptoms of downy

Fig. 10. Healthy and downy mildew grape leaves

mildew were collected from agricultural farms for CNN training. Fig. 10 shows sample images collected for healthy grape leaves and that of early downy mildew.

The goal of the tomato experiment was to demonstrate capability of the CNN model to differentiate between diseases of similar nature. Many image samples with symptoms of fungal diseases early blight and late blight of tomato were collected for the training of CNN model as shown in Fig. 11.

Fig. 11. Healthy and Blight of tomato

In test runs with previously unseen images of healthy and diseased grapes and tomato, 100% success was achieved in the correct classification of the diseases, proving that the algorithm is powerful enough to identify even early symptoms and also differentiate between the same family of diseases.

IV. FUTURE WORK AND EXTENSIONS

Future work involves expanding the model to include more parameters which can improve the correlation to the disease. We can augment the image database with supporting inputs from the farmer on soil, past fertilizer and pesticide treatment along with publicly available environmental factors such as temperature, humidity and rainfall to improve our model accuracy and enable disease forecasting. We also wish to increase the number of crop diseases covered and reduce the need for expert intervention except for new types of diseases. For automatic acceptance of user uploaded images into the Training Database for better classification accuracy and least possible human intervention, a simple technique of computing the threshold based on a mean of all classification scores can be used.

Further application of this work could be to support automated time-based monitoring of the disease density maps that can be used to track the progress of a disease and trigger alarms. Predictive analytics can be used to send alerts to the users on the possibility of disease outbreaks near their location.

V. CONCLUSION

This paper presents an automated, low cost and easy to use end-to-end solution to one of the biggest challenges in the agricultural domain for farmers – precise, instant and early diagnosis of crop diseases and knowledge of disease outbreaks - which would be helpful in quick decision making for measures to be adopted for disease control. This proposal innovates on known prior art with the application of deep Convolutional Neural Networks (CNNs) for disease classification, introduction of social collaborative platform for progressively improved accuracy, usage of geocoded images for disease density maps and expert interface for analytics. High performing deep CNN model "Inception" enables real time classification of diseases in the Cloud platform via a user facing mobile app. Collaborative model enables continuous improvement in disease classification accuracy by automatically growing the Cloud based training dataset with user added images for retraining the CNN model. User added images in the Cloud repository also enable rendering of disease density maps based on collective disease classification data and availability of geolocation information within the images. Overall, the results of our experiments demonstrate that the proposal has significant potential for practical deployment due to multiple dimensions – the Cloud based infrastructure is highly scalable and the underlying algorithm works accurately even with large number of disease categories, performs better with high fidelity real-life training data, improves accuracy with increase in the training dataset, is capable of detecting early symptoms of diseases and is able to successfully differentiate between diseases of the same family.

ACKNOWLEDGMENT

Our sincere thanks to Dr. Pranav K. Singh, Assistant Professor, GADVASU, Ludhiana and Dr. M. K. Prasanna Kumar, Associate Professor of Plant Pathology, University of Agricultural Sciences, GKVK, Bangalore for consultation and providing access to the farms at both the universities for data collection. Guidance by Mr. Ashutosh Utpal in Android programming is also greatly appreciated.

REFERENCES

[1] L. Saxena and L. Armstrong, "A survey of image processing techniques for agriculture," in *Proceedings of Asian Federation for Information Technology in Agriculture,* 2014, pp. 401-413.

[2] E. L. Stewart and B. A. McDonald, "Measuring quantitative virulence in the wheat pathogen Zymoseptoria tritici using high-throughput automated image analysis," in *Phytopathology 104 9*, 2014, pp. 985–992.

[3] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012.

[4] TensorFlow.[Online].Available: https://www.tensorflow.org/

[5] D. P. Hughes and and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics through machine learning and crowdsourcing," in *CoRR abs/1511.08060*, 2015.

[6] S. Raza, G. Prince, J. P. Clarkson and N. M. Rajpoot, "Automatic detection of diseased tomato plants using thermal and stereo visible light images," in *PLoS ONE*, 2015.

[7] D. L. Hernández-Rabadán, F. Ramos-Quintana and J. Guerrero Juk, "Integrating soms and a bayesian classifier for segmenting diseased plants in uncontrolled environments," 2014, in *the Scientific World Journal*, 2014.

[8] S. Sankaran, A. Mishra, J. M. Maja and R. Ehsani, "Visible-near infrared spectroscopy for detection of huanglongbing in citrus orchards," in *Computers and Electronics in. Agriculture 77*, 2011, pp. 127–134.

[9] C. B. Wetterich, R. Kumar, S. Sankaran, J. B. Junior, R. Ehsani and L. G. Marcassa, "A comparative study on application of computer vision and fluorescence imaging spectroscopy for detection of huanglongbing citrus disease in the USA and Brazil," in *Journal of Spectroscopy*, 2013.

[10] C. Szegedy,"Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818-2826.

[11] Mango Diseases and Symptoms. [Online].Available: http://vikaspedia.in/agriculture/crop-production/integrated-pest-managment/ipm-for-fruit-crops/ipm-strategies-for-mango/mango-diseases-and-symptoms

[12] P. Subrahmanyam, S. Wongkaew, D. V. R. Reddy, J. W. Demski, D. McDonald, S. B. Sharma and D. H. Smith, "Field Diagnosis of Groundnut Diseases". *Monograph. International Crops Research Institute for the Semi-Arid Tropics*, 1992.