International Journal of Engineering Innovations in Advanced Technology
ISSN: 2582-1431 (Online), Volume-4 Issue-2, June 2022

Plant Disease Identification, Tracking and Forecasting for Farmers

K Bharath, K Yashwanth, R Manasa, P Akanksha
UG Students, Department Of CSE, Sphoorthy Engineering College, Hyderabad, India.
Email: kodavatikantibharath@gmail.com

A Venu Gopal Rao
Assistant Professor, Department Of CSE, Sphoorthy Engineering College, Hyderabad, India.
Email: venugopal299@gmail.com

Abstract **- Plant diseases threaten farmers, consumers, the environment, and the economy. Pathogens and pests cause 35% of crop losses in India. Many pesticides are toxic and biomagnified, thus indiscriminate usage is dangerous. Early disease detection, crop monitoring, and targeted treatments may avert these effects. Most agricultural ailments are identified by external indicators. Farmers need expert help. Our platform automatically diagnoses, tracks, and forecasts illnesses. Farmers may use a smartphone app to rapidly and accurately diagnose plant diseases and discover solutions. Using AI algorithms for cloud image processing, real-time diagnosis is conceivable. The AI model learns from user-uploaded photos and expert advice. Farmers may contact regional experts online. A cloud-based library of geo-tagged pictures and micro-climatic information is utilised to build disease density maps with spread predictions. Experts may analyse diseases using web-based spatial visualisations. In our investigations, the AI model (CNN) was trained using disease datasets created from plant photographs taken over 7 months from several farms. Plant pathologists accepted the CNN model's picture diagnoses. Diseases were identified with 95% accuracy. Our technique offers a novel, adaptive disease management tool. Farmers and industry experts may utilise this cloud-based service to produce ecologically friendly crops.**

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

I.INTRODUCTION

Human survival relies on agriculture. In rapidly developing countries with a high population density like India, it is of utmost importance to increase agricultural, fruit, and vegetable yields. Both the quality and production of the products must be kept at a high level in order to enhance public health. However, issues such as the spread of illnesses that may have been contained by earlier diagnosis are a barrier to both production and the quality of the food produced. The fact that many of these diseases may be spread from person to person results in a total loss of agricultural yield. Because of the widespread geographic dispersion of agricultural fields, the low education levels of farmers, the restricted expertise, and the lack of access to plant pathologists, human-assisted disease diagnosis is inadequate and unable to keep up with the excessive demand.

It is vital to automate crop disease diagnosis using technology and build low-cost, reliable machine assisted diagnostic that is widely accessible in order to solve the drawbacks of human helped disease detection. This may be accomplished by developing machine aided diagnostics. The agricultural industry is facing a wide range of challenges, but recent developments in computer vision and robotics have made it possible to tackle some of these problems. The use of image processing as a tool to enhance precision agriculture methods, weed and pesticide technologies, monitoring plant growth, and the regulation of plant nutrition has been the subject of research. [1][2]. Discovered by plant pathologists through the visual examination of physical signs such as observable colour changes, wilting, the emergence of spots and lesions, etc., as well as soil and environmental factors. Despite the fact that many plant diseases can be automatically diagnosed, advancement in this area is still considered to be primitive. The commercial level of investment in combining agriculture and technology is still at a lower overall level compared to investments made in more profitable areas such as human health and education. Promising research efforts have not been able to materialise because of hurdles such as access and connectivity for farmers to plant pathologists, high implementation costs, and the scalability of solutions.

Because of recent developments in mobile technology, cloud computing, and artificial intelligence, it is now feasible to create a solution to agricultural illnesses that is both scalable and inexpensive, as well as one that can be utilised by a large number of people (AI). It is becoming usual in developing countries like India to find people using mobile devices that have internet connection. People are able to submit images with geolocation information by using commonly available low-cost mobile phones equipped with cameras and GPS. They have the capability to connect with more modern cloud-based backend

Manuscript received June 15, 2022; Revised June 25, 2022; Accepted June 30, 2022

services, which can carry out compute-intensive tasks, maintain a centralised database, and conduct out data analytics over widely available mobile networks. An further development in technology is that in recent years, the capacity of AI-based image analysis to properly recognise and categorise photographs has exceeded that of the human eye. This is a significant step forward in the field. The underlying artificial intelligence algorithms make use of something called neural networks (NN), which are comprised of many layers of neurons and a connection structure that is based on the visual cortex.

These networks are "trained" using a large number of previously recognised, or "labelled," photos in order to obtain a high level of accuracy in image classification when applied to new, previously unseen photographs. Since 2012, when "AlexNet" was declared the winner of the ImageNet competition [3,] deep convolutional neural networks, often known as CNNs, have been widely regarded as the most effective architecture for computer vision and image processing.

The breakthrough in CNN capabilities may be attributed to many factors, including advances in processing power, the accessibility of large-scale data sets, and the creation of more effective NN algorithms. The field of artificial intelligence (AI) has grown and improved because to open source platforms such as TensorFlow, in addition to being more affordable and accessible [4]. Examples of earlier works that are pertinent to our investigation include spectrum patterns, RGB pictures, spectral patterns, fluorescence imaging spectroscopy, and efforts to gather photographs of healthy and diseased crops [5, 6, 7, 8, 9].

Historically, neural networks have been used to the problem of diagnosing plant diseases, with the caveat that the process included identifying certain textural qualities. Our strategy takes use of the advancement of mobile technology, cloud computing, and artificial intelligence in order to provide an end-to-end crop diagnostic system that duplicates the knowledge ("intelligence") of plant pathologists and places it at the forefront of the process. It also provides a collaborative method to continually increasing the sickness database and obtaining expert advice when it is essential in order to improve the NN classification accuracy and keep track of epidemics.

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

Plant disease diagnostics are provided to farmers through our proposed solution, which takes the form of a scalable and collaborative platform hosted in the cloud. Users may take photographs of a variety of plant parts, and the website will immediately and automatically diagnose the ailment that is affecting the plant.

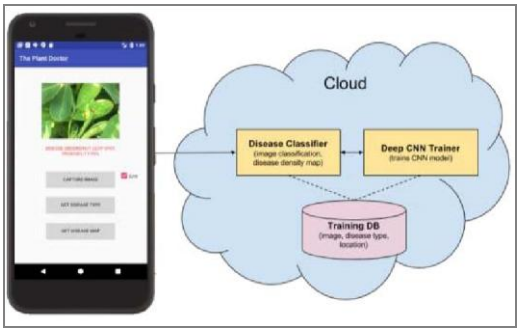

Fig. 1. System architecture with Cloud and Mobile components

The platform may be accessed via a downloadable mobile application. In addition to this, they might look at a map that illustrates the "disease density" of their area and how diseases are distributed geographically. Our artificial intelligence system classifies the provided picture according to the appropriate illness category and then offers the user a treatment that has been recognised in the past as being the most effective one. The time stamp and geolocation of the snapshot are used in conjunction with one another to concurrently tag the existence of the particular sickness in that area. On a map, the overall density of diseases that are kept in a cloud database is shown, together with their location in relation to the user.

A map depicting the prevalence of illnesses in the area may also be seen by the user (if location service is enabled on the phone). The mobile application is comprised of a total of eight 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). Java, Android Studio 3.1.3, and the Google Camera and Maps APIs were used in the development of the mobile application. The mobile application communicates with the cloud backend that is hosted by Amazon Web Services (AWS) via the use of cellular networks by using the AWS Mobile SDK for Android.x

Disease Classifier –When photos are submitted via the mobile app, the Classifier, a stand-alone programme operating in the cloud, receives them and applies a trained deep Convolutional Neural Network (CNN) model to identify the kind of disease. The Deep CNN Trainer computes the CNN model, which is then utilised by the Classifier to categorise the submitted photos into the appropriate illness types. The Classifier also carries out post-processing, such as choosing whether the submitted photos should be delivered to an agricultural expert registered on the site or added to the Training Database based on the classification score. for more research. The photos and their associated metadata, such as the illness kind and location of the images, are uploaded to the Training Database when the classification score exceeds a predetermined threshold. If the system receives a low

classification score, it sends the case to agricultural expert teams for manual classification. The manual classifications are then provided to the farmer and kept in the training database. When a user submits a picture with an illness that the trained CNN model is not yet aware of, it often results in low accuracy.or the image is of poor quality. When the classification score is poor, expert assistance enables the insertion of new illness categories that may be saved for further training sessions. The Classifier can begin automatically diagnosing the new disease after the Training Database contains a significant number of photos representing the new disease category and a high classification accuracy is attained. We are able to increase the accuracy of automated response to covered illnesses over time as more farmers interact and provide photographs, while also expanding coverage for new diseases with the limited expert resources.

x **Deep CNN Trainer** –

This web-based application is responsible for taking care of the more laborious tasks involved in training the neural network and developing the deep CNN model. These are the tools that the classifier uses in order to place photographs into the relevant sickness categories. This programme begins running in the background asynchronously once the number of newly uploaded pictures to the Training Database reaches a certain threshold (without interfering with the Classifier). The deep CNN model that is used by the Classifier to categorise illnesses more correctly is continually improved by subsequent runs of this training programme, which use a larger training dataset. This allows for an ever-increasing level of precision in disease classification. AWS was used throughout the construction of the whole cloud platform. Python has been used to develop a number of useful applications, such as the Disease Classifier and the Deep CNN Trainer. These Python programmes have been altered in order to make them accessible through mobile internet.

These were developed with the assistance of the FLASK web framework, and they were placed in front of an Apache Web Server that was running on a server that was hosted by Amazon EC2 (Ubuntu 16.04.2 LTS, 2 GiB memory, 8 GiB EBS volume). Both the Disease Classifier and the Deep CNN Trainer were constructed with the help of TensorFlow [4, an open source artificial intelligence framework created by Google].

x **Training Database –**

This cloud-based database houses all of the images that were used in the process of training the deep CNN model. In addition to the images, the metadata, which includes information such as the kind of disease, the location of the picture, and time stamps, is recorded. This database becomes larger as the smartphone app is used more often and as more images taken in farmers' fields are submitted to the website. As a result of the expansion of the training database, the deep CNN model may undergo ongoing retraining using ever more extensive data sets. The data in this database is also used to compute disease density in relation to the user's location. The calculation is based on collective information such as the types of illnesses and the geolocations of picture files. and the mobile app presents the disease prevalence maps that were generated as a consequence. AWS S3 was used to build the photo database, while MySQL running on AWS EC2 was used to store the sickness information, which included category, treatment, and location.

x **Expert Interface**

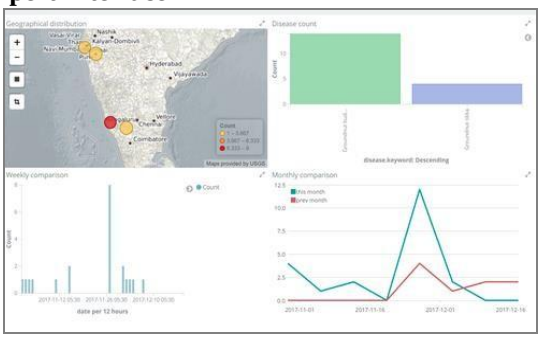

Fig. 2. Expert dashboard with disease data visualizations

Agricultural professionals now have access to a web-based expert interface that allows them to manually identify photographs that have received poor categorization scores. When an expert manually classifies a picture, an SMS notification is sent to the user, urging them to review the history of the mobile app for the most recent classification and any recommendations to make. Another feature of this interface is that it takes use of the disease data that is stored on the cloud platform. This gives professionals the ability to generate the time- and location-based sickness data visualisations shown in Figure 2 for the purposes of analytics and monitoring. Figure 3 depicts the flow of the process as well as the operations that are carried out by the platform's separate components and the interactions that occur between them.

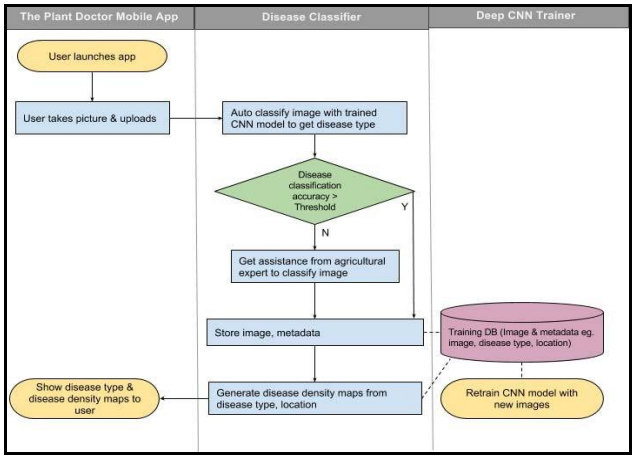

Fig. 3. Process flow of the components

III. EXPERIMENTS, RESULTS AND OBSERVATIONS

Numerous phases of testing were carried out in order to properly imitate lab-based and field-based situations for image analysis, which is the foundation of this idea. In general, there are three distinct types of experiments that may be distinguished: Experiment 1 used training images discovered through a Google search to test the viability of the proposal; Experiment 2 demonstrated the proposal's high degree of accuracy even when used with a wide range of disease categories by using images taken under controlled conditions from a large open source public dataset; Experiments 3 and 4 were carried out using photographs that were self-collected, high fidelity, and high resolution, and were taken on an agricultural farm to replicate real life usage; Experiment 1 tested the viability of the proposal The seasonality of crops, their accessibility, the severity of diseases, and their prevalence at the time of the experiment all had a role in our choice to undertake active data gathering throughout the season. Our decision was impacted by all of these factors.

A. Experiment 1

In order to develop a Disease Classifier as a first level of exploration, we trained a deep CNN model using the most recent Inceptionv3 architecture [10] and Python's TensorFlow framework [4]. The objective of this experiment was to do picture identification of several mango illnesses as a baseline experiment to show that the method was feasible before moving on to a larger data set or to actual data gathered from fields. Using Google Search, several images of mangoes with illnesses were retrieved. Four prevalent mango illnesses [11] were chosen for this experiment because they each had different symptoms and visual representations.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 Inception-based CNN model was trained (transfer learned) using the downloaded photos in order to educate the network to distinguish the four different mango disease kinds. For training and classification, our model transforms input photos to 299X299 RGB. The information on picture categorization training and testing for Experiment 1 is shown in Table 1.

Table 1. Experiment 1 Training And Classification Results

	Training						Classification Probability Score			
Experiment	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	0.589	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	0.885
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	0.948
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

In Experiments 1.1 through 1.3, a training data set for the CNN was created using 67 photos of bacterial canker, 70 photographs of mildew, 22 images of phoma blight, and 37 images of red rust. To determine if test accuracy varies with the training steps, several training step counts (250, 500, and 1000) were employed. The classification of test photos was done using the trained model. An array of four probability scores that add up to one is the result of categorization for each test picture. The likelihood that a test picture falls into one of the four categories is represented by each score.

With neural networks, overfitting is a potential issue where a model may simply be memorising unimportant aspects of the training photos to get the correct responses, giving strong results on the images it has seen during training but failing on new images. The process of training and testing were carried out and tabulated under Experiments 1.4 and 1.5 in Table 1, keeping the number of training steps constant at 500, in order to avoid the issue of overfitting. Some of the images from the training set were removed in order to prevent the model from memorising them.

Table 1 shows that the classification accuracy is acceptable even with a relatively small number of poor quality training photos. Experiment 1.1, for instance, demonstrates that a test picture for bacterial canker was correctly categorised with a probability score of 0.749. Training is repeated by altering the number of training steps in 1.1, 1.2, and 1.3 while maintaining the same number of training pictures. Results demonstrate that the classification score increases as the number of training steps grows from 250 to 500 to 1000. (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).Experiment 1.4 in Table 1 shows that even after the test photos were deleted from the training data, the trained model was still able to classify images it had never seen before with a decent amount of accuracy. Red rust image's classification was wrong (the model classified the red rust image as bacterial canker with a score of 0.379, marked in red). After replacing the test picture of red rust with a better image from the training set (with sharper spots), the experiment was redone (Experiment 1.5), and the classification score increased, with the model correctly categorising the image as red rust with a score of 0.827. This demonstrates how the quality of the test data improves categorization accuracy.

However, trained CNN models may be used to identify photos relatively fast (1-3 seconds), making the usage of neural networks in smartphone apps conceivable. Training neural networks can be computationally and time-intensive (10–60 minutes for our runs). Experiment 1's findings demonstrate that CNNs can be used for picture classification in our use case of diagnosing plant diseases, and they encourage additional research using more extensive and high-fidelity test data.

Experiment 2

Fig. 5. PlantVillage Sample Images [5]

A sizable public dataset containing pictures of both healthy and ill plants that was gathered under controlled circumstances by agricultural professionals was utilised in the second stage of research. This was done to demonstrate how the treatment may be used to treat additional types of diseases on a wider scale. In order to facilitate the development of computer vision methods to assist address the issue of agricultural production loss owing to infectious illnesses, PlantVillage, an open-source platform [5] for crop health, has made a public collection of more than 50,000 plant photos available. This collection contains edited photos of both healthy and diseased plant leaves. 38 potential crop-disease pairings (classes/categories denoted as c0 to c37) are generated from photos of 26 illnesses in 14 crops.

Eight categories of PlantVillage photos were randomly selected for training and assessment in our experiment. Five photos from each category were taken out of the training set to use as the test data. Our Inception-based CNN model was trained using the remaining training data from the 8 categories, and test pictures were then categorised using the learned model. The statistics for the training data set that was utilised to create the trained CNN model are displayed in Table 2.

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 :
x 37 out of 40 photos had the proper classification, meaning that 92.5 percent of the images were correctly identified, demonstrating that the answer will hold true even for a sizable dataset with many different illness categories. The table has 3 erroneous cases highlighted in red.

X The inaccurate classifications may be the result of factors like the test image's poor quality, which prevents the classifier from accurately identifying it, and a few categories that have a lot of visual similarities (such c21 and c24).

Fig. 6. c21_test2 that failed to classify correctly

C. Experiment 3

To confirm the end user experience, groundnut was selected as the primary case study for field research and experiment. The purpose of this experiment was to recreate real-world situations using pictures that users had shot outside in the open air. Due to its widespread cultivation and use across the globe and its great economic value as a rich source of protein and edible oil, groundnut is also known as peanut. We decided to use groundnut as our case study since 80 percent of the world's groundnut crop is grown in underdeveloped nations, where yields are frequently quite low and illnesses have become a significant barrier to groundnut production globally [12The top three countries that produce groundnuts worldwide are China, India, and the US. Although there are several diseases that affect groundnut crops [12], two prominent ones—leaf spot, also known as "tikka," and bud necrosis—were chosen for this experiment because of their severity, significant production impacts, and widespread incidence in India.

On the experimental fields of the Punjab Agricultural University in Ludhiana and the University of Agricultural Sciences, GKVK in Bangalore, hundreds of pictures of healthy and sick groundnut plants were gathered as part of the fieldwork. Three different varieties of groundnut plants—healthy plants, plants with leaf spot, and plants with bud necrosis—were chosen from the farms for data gathering and experiments. The three categories under experimentation—healthy, sick with leaf spots, and diseased with bud necrosis—were identified using the gathered photos as training data for the CNN model. The data for training the CNN model for groundnuts are shown in Table 4.A total of 811 photos were gathered for training, of which 243 were of healthy plants, 358 were of plants with the illness known as "tikka," or leaf spot, and 210 were of plants with bud necrosis. The network's training procedure was completed in 14 minutes, yielding a trained CNN model. The classification of a set of test photos that weren't included in the training data was then done using the trained CNN model.

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:

- x It was demonstrated that the accuracy would be high with accurately categorised high fidelity training data set by the fact that correct classification was accomplished in 100% of test situations with high accuracy across all 3 categories (healthy, leaf spot, and bud necrosis).
- x The findings of Experiment 3 were superior to those of Experiment 1 (downloaded Google photos) and Experiment 2 (open source data set from controlled settings), despite the fact that Experiment 2 included more training images. This would suggest that using photos captured in their natural environment helps with training and categorization. x The leaf spot or "tikka" illness had the best classification accuracy among the categories in Experiment 3. Experiment 3's Leaf Spot category had the most training photographs, demonstrating that, provided all other variables are held constant, a larger number of verified training images will result in a higher accuracy. With a relatively smaller collection of training data, x Experiment 3 generated a high level of accuracy. This leads us to infer that even with large-scale production deployment, the rate of error or ambiguity in illnesses categorization can

be kept low because bigger-scale deployment also entails larger training datasets with user-added photos.

- The categorization of a test picture using the trained CNN model is extremely quick (on average 1.4 seconds), despite the 14 minute training process. This demonstrates how the effectiveness of the underlying deep CNN enables the challenging work of picture categorization to be accomplished via a user-facing mobile app. The training process is separate and operates in the Cloud without interfering with the runtime of a picture. classification. The user application for mobile devices has a straightforward frontend interface, while the complex training and classification tasks are handled by the robust AI algorithms operating in the cloud and providing real-time classification results to the user application.

D. Experiment 4

Our intention was to extend the model's coverage to additional local crop varieties. Experiment 3 was expanded to include training and testing with actual photographs of tomato and grape illnesses. The grape experiment was designed to show that illness categorization is accurate even for disease symptoms that appear relatively early. For CNN training, several picture examples of healthy grapes and grapes showing early signs of downy mildew were gathered from agricultural fields. Sample photos of healthy grape leaves and early downy mildew are shown in Fig. 10.

Fig. 10. Healthy and downy mildew grape leaves

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 tests using never-before-seen photos of healthy and sick grapes and tomatoes, 100% success was attained in correctly classifying the illnesses, demonstrating that the algorithm is capable of recognising even the earliest signs and differentiating across diseases in the same family.

IV. FUTURE WORK AND EXTENSIONS

Expanding the model to include other factors that might strengthen the illness association is a task for future study. We may add supporting data from the farmer on the soil, past fertiliser and pesticide treatment, as well as publicly available climatic elements like temperature, humidity, and rainfall to the picture database to improve the model's accuracy and allow disease forecasting. In addition to lowering the overall need for professional assistance and raising the incidence of agricultural diseasesOur objectives span unique disease types. A straightforward method of calculating the threshold based on the average of all classification scores may be used to automatically admit user-uploaded photos into the Training Database for greater classification accuracy and little human interaction.

This research may also be utilised to assist time-based automated monitoring of illness density maps, which may be used to follow a disease's progression and sound alerts. Users may receive notifications using predictive analytics about potential illness outbreaks in their area.

V CONCLUSION

The accurate, prompt, and early diagnosis of crop diseases as well as knowledge of disease outbreaks, which would be helpful in making decisions regarding the actions to be taken for disease control, are two of the greatest challenges that farmers face in the agricultural industry. Knowing how to effectively combat diseases also presents a significant obstacle. In this study, we provide a fully automated, low-cost, and straightforward end-to-end solution to the challenges described above. This proposal improves upon the known prior art by making use of deep Convolutional Neural Networks (CNNs) for disease classification, introducing a social collaborative platform for steadily increasing accuracy, making use of geocoded images for disease density maps, and making use of an expert interface for analytics. The high-performing deep CNN model "Inception" enables real-time sickness classification in the Cloud platform, and it does so via a mobile app that the end user interacts with.

The collaborative model makes it possible for the accuracy of sickness classification to be continuously improved. This is accomplished by automatically increasing the cloud-based training dataset with user-added photographs for the purpose of retraining the CNN model. User-added images in the Cloud repository allow for the generation of disease density maps, which are based on the collective sickness classification data as well as the availability of geolocation information within the photos themselves. In general, the findings of our experiments indicate that the proposal has significant potential

for practical application. This is the case for a number of reasons, including the following: the infrastructure that is based in the cloud is highly scalable; the underlying algorithm works accurately even with a large number of disease categories; it performs better with high-fidelity real-life training data; it improves accuracy with increase in the training dataset; it is able to detect early symptoms of diseases.

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 Zymoseptoriatritici 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-pestmanagment/ipm-for-fruit-crops/ipm-strategies-for-mango/mangodiseases-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.