SURVEY PAPER ON CROP DISEASE NOTIFICATION SYSTEM

Vaishnavi Nighot[1], Amit Dhage[1], Prajwal Wandhare[1], Sayali Dhole[1], Prof. Rashmi Kale[2]

[1]Student, Department of Computer Science Engineering
[2]Assistant Professor, Department of Computer Science Engineering
Smt. Kashibai Navale College of Engineering, Pune, India

---***---

Abstract - *Crop sicknesses are a primary danger to food security; however, their fast identification remains tough in lots of elements of the sector because of the shortage of the essential infrastructure. The aggregate of growing worldwide smartphone penetration and current advances in computer imaginative and prescient made viable with the aid of deep getting to know has paved the way for cellphone-assisted sickness prognosis. The use of a public dataset of 20,306 photos of diseased and healthy plant leaves accumulated underneath controlled situations, we educate a deep convolutional neural network to pick out 15 crop species and 29 diseases (or absence thereof). The educated model achieves an accuracy of 85.35% on a held-out check set, demonstrating the feasibility of this approach. Universal, the method of schooling deep learning models on an increasing number of huge and publicly to be had image datasets gives a clean route in the direction of phone-assisted crop ailment diagnosis on a massive worldwide scale. Notifications offer a completely unique mechanism for increasing the effectiveness of actual-time facts transport systems. However, notifications that demand farmers' attention at inopportune moments are more likely to have destructive effects and may become a motive of capability disruption in place of proving beneficial to farmers. In order to address these demanding situations a spread of notification mechanism based on tracking and gaining knowledge of crop disease behavior were proposed. The goal of such mechanism is maximizing farmers receptiveness to the added records by means of routinely inferring the proper crop and the proper fertilizers, for assuring accurate yield of crops.*

***Key Words*: Smartphone, Notifications, Crop Disease, Deep Convolutional Neural Network.**

1.INTRODUCTION

Plant illnesses[1], pest infestation, wind pressure, and nutrient deficiencies are a number of the grand demanding situations for any agricultural manufacturer, at any vicinity and for anything Commodities or size of the operation is dealing every day. It's far critical that farmers would know the existence of such challenges of their operations on a well-timed basis. Nevertheless, it could be pretty helpful to agricultural manufacturers to have access to with no trouble to be had Generation to coach them on how to cope with each of those threats for Agricultural Production to beautify crop manufacturing and operation profitability.

For example, within the India, plant ailment reasons losses of between 50 and 60 percentage of the agricultural crop manufacturing annually. Consequently, farmers ought to directly diagnose the specific varieties of plant illnesses to stop their spread within their agricultural fields. Traditionally, underserved farmers attempt to diagnose plant diseases through optical commentary of plant leaves signs, which includes a drastically excessive diploma of complexity. Any misdiagnosis of crop decreases will cause the use of the wrong fertilizers that could stress the plants and result in nutrient deficiencies within the Agricultural area.

Machine Learning (ML) coupled with laptop vision have already enabled sport-changing precision agriculture abilties by means of providing the capability to optimize farm returns, hold herbal resources, lessen unnecessary use of fertilizers, and discover sickness in vegetation and animals from remotely sensed imagery.

Consider a clever Mobile-based totally system that farmers can use to perceive the unique forms of plant diseases with excessive accuracy[2]. Such structures might assist both small- and huge-scale farmers to make the right choices on which fertilizers to use to confront plant sicknesses in their plants.

This paper offers a cell-based machine for detecting plant leaf diseases[3] the use of Deep Learning (DL) in realtime. In particular, we developed a distributed gadget that is prepared with components executing on centralized servers at the cloud and locally at the user's cellular gadgets. We created a dataset that includes more than 20 k pictures for the most common place 38 plant ailment categories in 15 crop species, together with tomato scab, tomato, grape leaf[3] blight, rice[4] and lots of others[5].

On the cloud side, we created a Convolutional Neural network (CNN) model which can feed pixS without delay from farmers' mobile devices. The version then performs item detection[6] and semantic segmentation, and displays the ailment class in conjunction with the self-assurance percent and class time have taken to process the photo. We Evolved an Android mobile app to permit constrained-assets farmers to seize a image of the diseased plant leaves. The cellular app runs on pinnacle of the CNN model on the user side. Also, the application shows the self-assurance percentage and class time taken to technique the picture.

The contributions of this paper are threefold. First, we advocate a dispensed ML powered Platform that is organized with parts executing on the cell consumer gadgets at the agricultural discipline and excessive-overall performance servers hosted inside the cloud. Second, the proposed Machine is capable of taking pictures, processing, and visualizing massive imagery agrarian datasets. Third, we evolved a person-pleasant interface on pinnacle of the CNN model to allow Farmers to interact with the disorder detector quite simply on the cellular aspect. Fourth, Mobile notifications are presented in a unified fashion by almost all mobile operating systems.

In order to ensure real-time awareness of users about the delivered information, mobile operating systems rely on notifications that steer users' attention towards the delivered information through audio, visual and haptic signals. This is indeed in contrast with the traditional paradigm of pull-based information retrieval and delivery in which the user has to initiate a request for the transmission of information.

2. SYSTEM DESIGN

The distributed run-time system for the plant disease detector with Notification is organized with parts executing on mobile devices at the user side, as well as on centralized servers at the cloud side. Module 1 describes the farmer registration at client side, which includes Log In and Sign In Dashboard. Module 2 depicts the crop prediction depending on soil, temperature[7] and rainfall attributes on the preferred location[8]. Module 3 recognizes the disease of the crop which formulates the device to send Notification for the users.

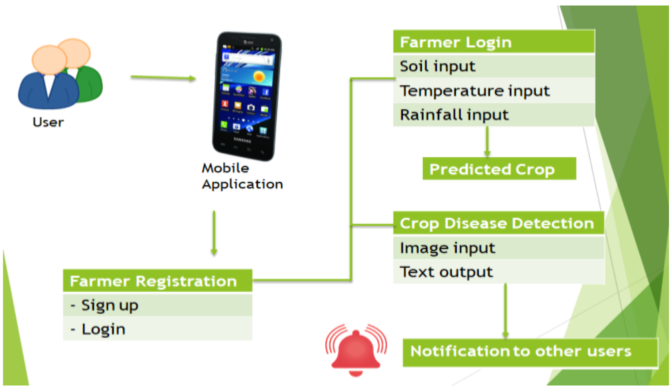


Fig -1: System Architecture

2.1 CNN Structure

We trained a CNN model[9] with 2 convolutional layers, one input layer and one output layer. $I = [i_1, i_2, \dots, i_r]$ and $O = [o_1, o_2, \dots, o_h]$ represent the input and output vectors,

respectively, where r represents the number of elements in the input feature set and h is the number of classes. The main objective of the network is to learn a compressed representation of the dataset. In other words, it tries to approximately learns the identity function F, which is defined as:

$$F_{W,B}(I) \neq I \qquad (1)$$

where W and B are the whole network weights and biases vectors.

A log sigmoid function is selected as the activation function *f* in the hidden and output neurons. The log sigmoid function *s* is a special case of the logistic function in the *t* space, which is defined by the formula:

$$s(t) = \frac{1}{1+e^{-t}} \qquad (2)$$

The weights of the CNN network create the decision boundaries in the feature space, and the resulting discriminating surfaces can classify complex boundaries. During the training process, these weights are adapted for each new training image. In general, feeding the CNN model with more images[10] can recognize the plant diseases more accurately. We used the back-propagation algorithm, which has a linear time computational complexity,

for training the CNN model.

The input value ϴ going into a node i in the network is calculated by the weighted sum of outputs from all nodes connected to it, as follows:

$$\Theta_i = \sum\left(\omega_{i,j} * Y_j\right) + \mu_i \qquad (3)$$

where $\omega_{i,j}$ is the weight on the connections between neuron *j* to *i*; Y_j is the output value of neuron *j*; and μ_i is a threshold value for neuron *i*, which represents a baseline input to neuron *i* in the absence of any other inputs. If the value of $\omega_{i,j}$ is negative, it is tagged as inhibitory value and excluded because it decreases net input.

The training algorithm involves two phases: forward and backward phases. During the forward phase, the network's weights are kept fixed, and the input data is propagated through the network layer by layer. The forward phase is concluded when the error signal

e_i computations converge as follows:

$e_i = (d_i - o_i)$ (4)

where d_i and o_i are the desired (target) and actual outputs of ith training image, respectively.

In the backward phase, the error signal e_i is propagated through the network in the backward direction. During this phase, error adjustments are applied to the CNN network's weights for minimizing e_i.

We used the gradient descent first-order iterative optimization algorithm to calculate the change of each neuron weight $\Delta\omega_{i,j}$, which is defined as follows:

$$\Delta\omega_{i,j} = -\eta \frac{\delta\epsilon(n)}{\delta e_j(n)} y_i(n) \qquad (5)$$

where $y_i(n)$ is the intermediate output of the previous neuron n, η is the learning rate, and $\varepsilon(n)$ is the error signal in the entire output. $\varepsilon(n)$ is calculated as follows:

$$\varepsilon(n) = \frac{1}{2}\sum_j e_j^2(n) \qquad (6)$$

The CNN network has two types of layers: convolution and pooling. Each layer has a group of specialized neurons that perform one of these operations. The convolution operation means detecting the visual features of objects in the input image such as edges, lines, color drops, etc. The pooling process helps the CNN network to avoid learning irrelevant features of objects by focusing only on learning the essential ones. The pooling operation is applied to the output of the convolutional layers to downsampling the generated feature maps by summarizing the features into patches. Two common pooling methods are used:

average-pooling and max-pooling. In this paper, we used the max-pooling method, which calculates the maximum value for each patch of the feature map as the dominant feature.

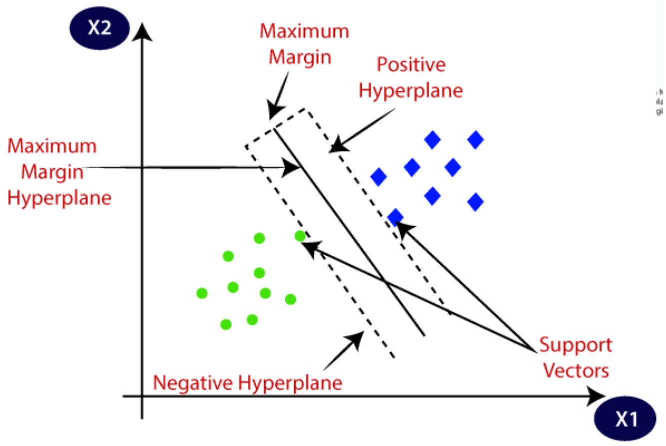
Before moving the trained CNN model to the mobile device, we converted it into an optimized IR model based on the trained network topology, weights, and biases values. We used the Intel OpenVINO[18] toolkit to generate the IR model, which is the only format that the inference engine on the Android platform accepts and understands. The conversion process involved removing the convolution and pooling layers that are not relevant to the mobile device's inference engine. In particular, OpenVINO splits the trained model into two types of files: XML and Bin extension. The XML files contain the network topology, while the BIN files contain the weights and biases binary data.

2.2 SVM

An essential aspect of medical research is the prediction[11] for a health outcome and the scientific identification of important factors. As a result, numerous methods were developed for model selections in recent years. In the era of big data, machine learning has been broadly adopted for data analysis. In particular, the Support Vector Machine (SVM)[12] has an excellent performance in classifications and predictions with the high-dimensional data. In this research, a novel model selection strategy is carried out, named as the Stepwise Support Vector Machine (StepSVM). The new strategy is based on the SVM to conduct a modified stepwise selection, where the tuning parameter could be determined by 10-fold cross-validation that minimizes the mean squared error. Two popular methods, the conventional stepwise logistic regression model and the SVM Recursive Feature Elimination (SVM-RFE), were compared to the StepSVM. The Stability and accuracy of the three strategies were evaluated by simulation studies with a complex hierarchical structure. Up to five variables were selected to predict the dichotomous cancer remission of a lung cancer patient. Regarding the stepwise logistic regression, the mean of the C-statistic was 69.19%. The overall accuracy of the SVM-RFE was estimated at 70.62%. In contrast, the StepSVM provided the highest prediction accuracy of 80.57%. Although the StepSVM is more time consuming, it is more consistent and outperforms the other two methods.

There are two types of machine learning, the supervised machine learning with a specific outcome variable and the unsupervised machine learning that only examines the associations between a set of predictors. Regression and classification are two primary applications for supervised learning, such as the generalized linear model (GLM), the logistic regression model, and the Support Vector Machine (SVM). For unsupervised learning, clustering is the leading interest and the most popular method is the Principal Components Analysis (PCA). The SVM is a machine learning tool dealing with classification problems. With an increasing amount of variables collected, the high dimensional data draw more attention in image processing[13] and the SVM is considered a powerful classification method. Chang et al. concluded that the SVM is useful in the imaging diagnosis of breast cancer and its classification ability is nearly equal to a neural network model. In particular, when a non-linear structure exists, the SVM demonstrates its superior ability to find the optimal separating hyperplane by kernel tricks into a higher dimensional feature space.

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:



Hyperplane: There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM. The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features (as shown in image), then hyperplane will be a straight line. And if there are 3 features, then hyperplane will be a 2-dimension plane. We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.

Example: SVM can be understood with the example that we have used in the KNN classifier[14]. Suppose we see a strange cat that also has some features of dogs, so if we want a model that can accurately identify whether it is a cat or dog, so such a model can be created by using the SVM algorithm. We will first train our model with lots of images of cats and dogs so that it can learn about different features of cats and dogs, and then we test it with this strange creature. So as support vector creates a decision boundary between these two data (cat and dog) and choose extreme cases (support vectors), it will see the extreme case of cat and dog. On the basis of the support vectors, it will classify it as a cat. SVM algorithm can be used for Face detection, image classification, text categorization, etc.

Types of SVM

SVM can be of two types:

1]Linear SVM: Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

2]Non-linear SVM: Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

Although the SVM has been widely extended in a variety of concepts, such as the parameter tuning or kernel choices, this research is focusing on statistical methodologies for stepwise selection models. The backward eliminations implemented by the SVM-RFE consider variables that are top-ranked to be eliminated last are not necessarily the factors that are individually most relevant. In particular, these predictors are the most relevant conditional on the specific ranked subset in the model. In order to avoid incorrectly eliminated factors, we propose a novel forward algorithm with stepwise considerations based on the SVM. The name of the new strategy is the Stepwise Support Vector Machine (StepSVM).

2.3 Dataset

Although standard object detection datasets (e.g., Microsoft COCO) exhibit volume and variety of examples, they are not suitable for plant disease detection as they annotate a set of object categories not include plant diseases. Therefore, we collected more than labelled 20k images of healthy and infected plant leaves for training the CNN model from different sources such as Kaggle[15], Plant Village and Google Web Scraper[16]. Many images in our dataset are in their natural environments because object detection is highly dependent on contextual information.

Our dataset is divided into three parts: training, validation and testing. **Table 1** shows the number of images used in the three phases across the 38 disease classes in 14 crop

species. The number of images in each phase is determined based on the fine-tuned hyperparameters and structure of the CNN model.

We conducted a set of controlled experiments to estimate the hyperparameters to improve the prediction accuracy and performance. In particular, we progressively tested random combinations of hyperparameter values until we achieved satisfactory results. Cross-validation optimizers were also used to find the best set of hyperparameters.

To increase the training accuracy and minimize training loss of the CNN model, we applied a series of image pre-processing transformations to the training dataset. Particularly, we altered the contrast of image colors, added Gaussian noise, and used image desaturation, which makes pixel colors more muted by adding more black and white colors. The primary purpose of these transformations is to weaken the influence of the background factor during the training process. This had a better effect on learning the 38 disease classes more effectively and increased our CNN model's stability.

We had to normalize the range of pixel intensity values of leaf images in the dataset before training the CNN model. This step was necessary because all dimensions of feature

vectors extracted from input images should be in the same intensity range. This made the convergence of our CNN model faster during the training phase. Image normalization was implemented by subtracting the input image's mean value μ from each pixel's value $I(i, j)$, and then dividing the result by the standard deviation σ of the input image. The distribution of the output pixel intensity values would resemble a Gaussian curve centered at zero. We used the following formula to normalize each image in our training set:

$$O(i,j) = \frac{I(i,j) - \mu}{\sigma}$$

where I and O are the input and output images, respectively; and i and j are the current

pixel indices to be normalized.

Table 1. The Number of Images used in the Training, Validation, and Testing Phases Across The

Disease Classes.

Class #	Plant Disease Classes	Training	Validation	Testing	Total
1	Apple scab	2016	504	209	2819
2	Apple Black rot	1987	497	246	2730
3	Apple Cedar apple rust	1760	440	220	2420
4	Apple healthy	2008	502	187	2697
5	Blueberry healthy	1816	454	232	2502
6	Cherry healthy	1826	456	192	2282
7	Cherry Powdery mildew	1683	421	209	2214
8	Corn Cercospora Gray leaf spot	1642	410	162	2214
9	Corn Common rust	1907	477	234	2618
10	Corn healthy	1859	465	233	2557
11	Corn Northern Leaf Blight	1908	477	209	2594
12	Grape Black rot	1888	472	231	2591
13	Grape Esca Black Measles	1920	480	220	2620
14	Grape healthy	1692	423	198	2313
15	Grape blight Isariopsis	1722	430	220	2372
16	Orange Citrus greening	2010	503	253	2766
17	Peach Bacterial spot	1838	459	220	2517
18	Peach healthy	1728	432	231	2391
19	Pepper bell Bacterial spot	1913	478	220	2611
20	Pepper bell healthy	1988	497	242	2727
21	Potato Early blight	1939	485	231	2655
22	Potato healthy	1824	456	231	2511
23	Potato Late blight	1939	485	231	2655
24	Raspberry healthy	1781	445	209	2435
25	Soybean healthy	2022	505	253	2780
26	Squash Powdery mildew	1736	434	209	2379
27	Strawberry healthy	1824	456	242	2522
28	Strawberry Leaf scorch	1774	444	209	2427
29	Tomato Bacterial spot	1702	425	209	2336
30	Tomato Early blight	1920	480	242	2642
31	Tomato healthy	1926	481	231	2638
32	Tomato Late blight	1851	463	220	2534
33	Tomato Leaf Mold	1882	470	242	2594
34	Tomato Septoria leaf spot	1745	436	220	2401
35	Tomato Two-spotted spider mite	1741	435	143	2319
36	Tomato Target Spot	1827	457	220	2504
37	Tomato mosaic virus	1790	448	209	2447
38	Tomato Yellow Leaf Curl Virus	1961	490	220	2671
	Total	70.295	17.572	8339	96.206

3. CONCLUSIONS

Faced with growing demands, shrinking of natural resources, and more stringent regulations, the agriculture sector worldwide found refuge in AI through the use of smart and innovative IoT technologies to optimize production and minimize losses. Crop diseases are one of the critical factors behind the crop production losses in the India. Therefore, correct disease diagnosis and awareness about the diseases is one of the most important aspects of modern agriculture. Without proper identification of the disease, disease control measures can waste money and lead to further plant losses.

This paper presented the design of an ML-powered plant disease detector that enables farmers to diagnosis the most common 38 diseases in 14 species. We trained a CNN model using an imagery dataset consisting of 96,206 photos of healthy and diseased plant leaves, where crowded backgrounds, low contrast, and diverse illumination

condition images are taken into consideration. To increase the system usability, we developed a mobile app[17] that would create a better opportunity for limited-resources farmers to detect plant diseases in their early stages and eliminate the use of incorrect fertilizers that can hurt the health of both the plants and soil.

REFERENCES

[1]. K. Jagan Mohan, M. Balasubramanian, S. Palanivel " Detection and Recognition From Paddy Plant Leaf Images "

[2]. Sakshi Kasat , Dhawnit Thadhani , Manali Kolekar, Shreya Gudsurkar," eshetkari: an interface for farmers", In *International Education and Research Journal,Volume:2,Issue: 3,March 2016.*

[3]. Udgave, Monika Kasture, " Grape Leaf Disease Detection using K-means Clustering ", *IRJET,2016*

[4]. Dhanashri Nemishte, Rupali Patil, Supriya More, Sayali Udgave, Monika Kasture " Proposed Decision Support System for Indian Rice crop Yield Prediction", *2016 IEEE International conference for agriculture and rural Development*

[5]. Niketa Gandhi, Leisa J. Armstrong, Owaiz Petkar " Proposed Decision Support System for Indian Rice crop Yield Prediction", 2016 *IEEE International conference for agriculture and rural Development*

[6]. Ms. P.R. Patil, Prof. N.B. Bhawarkar," Crop Diseases Detection with preventive measures Using Image Processing ", In *International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622, NCERT – November 2015.*

[7]. M Arya, K Anjali, D Unni Detection of unhealthy plant leaves Using image processing and genetic algorithm with Arduino. *IEEE International Conference on Power, Signals, Control and Computation (2018), pp. 1-5*

[8]. Ahmed, A.A. A privacy-preserving mobile location-based advertising system for small businesses. *Eng. Rep.* ***2021****, 1–15.*

[9]. S Abed, A Esmaeel A novel approach to classify and detect bean diseases based on image processing. *IEEE Symposium on Computer Applications & Industrial Electronics (2018), pp. 297-302.*

[10]. Ahmed, A.A.; Echi, M. Hawk-eye: An ai-powered threat detector for intelligent surveillance cameras. *IEEE Access* ***2021****, 9,63283–63293.*

[11]. S. Nagini, Dr. T. V. Rajini Kanth, B. V. Kiranmayee," Agricultural Yield Prediction Using Predictive Analytic Techniques" 2016 *2nd International Conference on Contemporary Computing and Informatics, IEEE, 2016*

[12]. S. Nema, A. Dixit Wheat Leaf Detection and Prevention Using Support Vector Machine. *International Conference on Circuits and Systems in Digital Enterprise Technology (2018), pp. 1-5*

[13]. Suraksha, Sushmita, Sushma, Sushmita Keshav" Disease Prediction of Paddy Crops Using data Mining and Image Processing Techniques", *IJAREEIE-2016*

[14]. Kalavathi K, Nimitha Safar P.V.," Performance Comparison between Naïve Bayes,Decision Tree and kNearest Neighbor ",In *International Journal of Emerging Research in management and Technology,ISSN:2278-9359,Volume 4, Issue 6,2015*

[15]. Kaggle: Machine Learning and Data Science Community. 2021. Available online: https://www.kaggle.com/ *(accessed on 25 January 2023).*

[16]. GoogleWeb Scraper. 2021. Available online: https://chrome.google.com/webstore/detail/web-scraper/jnhgnonknehpejjnehehllkliplmbmhn?hl=en (accessed on 25 January 2023).

[17]. Kotlin: A Multiplatform Mobile Framework. 2021. Available online: https://kotlinlang.org/lp/mobile/ (*accessed on 25 January 2023).*

[18]. Openvino: A Toolkit for Optimizing Deep Learning Models. 2021. Available online: https://software.intel.com/content/www/

us/en/develop/tools/openvino-toolkit.html (*accessed on 25 January 2023).*