JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

AgroSmart: Enhancing Agricultural Productivity through Data-Driven Crop Recommendation

P. Nokesh
ECE, Sreenidhi Institute of Science and Technology Hyderabad, India
nokesh743@gmail.com

D. Prasanna
ECE, Sreenidhi Institute of Science and Technology Hyderabad, India
dharavathsaiprasanna16@gmail.com

V. Srikanth
ECE, Sreenidhi Institute of Science and Technology Hyderabad, India
varakalasrikanth966@gamil.com

Dr. Syed Jahangir Badashah
ECE, Sreenidhi Institute of Science and Technology Hyderabad, India
jahangir.s@sreenidhi.edu.in

Dr. Vikram Palodiya
ECE, Sreenidhi Institute of Science and Technology Hyderabad, India
vikram.p@sreenidhi.edu.in

***Abstract*—The problem of choosing the proper crops to put in the fields in order to maximize agricultural productivity across many regions of India is a complex one because weather patterns are erratic and soil quality is uneven. Modern technological tools that could support farmers in selecting the most appropriate crops for their land are often often not available to them or not available in a consistent form. And other knowledge related to agricultural science being limited, unaware many do not adopt innovations in the agricultural science, and they have to rely on traditional practices which are outdated. This reduces crop yields and, in some cases, whole harvests fail. The causes for these problems are typically due to improper fertilizer use, or unexpected rainfall. Therefore, the most optimal solution would be to suggest crops complementary with prevailing soil conditions and rainfall expected during the sowing period. This challenge is confronted with the design of a software recommendation system called the Soil-Based Crop Profiling Framework. This tool takes into account soil characteristics, like nitrogen, phosphorus, potassium levels and pH as well as rainfall predictions, to determine what crop and fertilizer is most suitable to grow. The implementation of the framework is using a graphical desktop application that helps to provide a practical support to minimize crop failure and decision making while farming.**

***Keywords*— Deep learning, crop prediction models, plant disease detection.**

I. INTRODUCTION

India's agriculture is the basis of livelihood of millions of farmers, but most of them experience struggle in majoring up their crop productivity. While farmers continue to be the backbone of Indian economy, there are numerous problems faced by them while understanding soil composition, using fertilizers effectively, adapting to changes in rainfall pattern and choosing the best crop for their land. These are further complicated decisions however, given that soil and climate conditions can vary from one field to another even in the same place.

For the most part, agricultural decisions are inherently complex. That means it will have to evaluate a number of variables including the soil pH, levels of nutrients, rainfall seasons and soil health overall. Sadly, though, farmers in many rural areas have no convenient access to the kind of expert analysis or digital tools to help guide those decisions. Since then they usually do according to guesswork and traditions, which typically never work out for the best – taking the wrong fertilizers or sowing at the wrong time can ruin an entire crop.

In order to help farmers tackle these challenges, we created a desktop based application that provides scientifically based crop and fertilizer recommendations. The system then identifies the most suitable crops, the fertilizer use for enhancing soil fertility to achieve consistent yields using real time soil and environmental inputs.

Solution: We came with a simple desktop software which lets the user provide input values for soil test input fields and depending on the values, also display suitable crops for cultivation. The inputs are used by the system to analyze whether the given soil fertility would be favorable to grow those crops. The benefit to this tool is especially helpful for farmers who do not know what nutrients are on their land, and are unsure which crop harvest will be the best.

Small scale farmers in India rarely have frequent access to soil testing facilities and not even have the knowledge regarding the condition of their soil. We trained our model using publicly available datasets linking different crops to the nutrient requirements that those crops require. This data was analyzed with an implemented Support Vector Machine (SVM) algorithm. The model will compare users' soil values within the training dataset and generate a list of suitable crops. Moreover it recommends the fertilizers for each growing season that would help farmers make more informed and productive decision.

II. LITERATURE SURVEY

A high degree of technological exploration in the agricultural industry has been for the purpose of the solution of long standing farming challenges as well as increasing productivity. While the traditional Indian farming is quietly inefficient, researchers are actively looking to identify the gaps and are also proposing the digital and encouraging the evidence-based tools for the cultivators to improve their proceedings. This scholarly work is devoted to a large part to finding out that which crop will fit in which preference in soil composition and climate conditions, references [1], and are among this part of the work. Precision farming from the ground up is another research in stream that takes the farmer as a beginner and slowly creates a prediction system to assist farmers in making their crop selection and environmental and soil based variables[2].

In order to support theses predictions, different range of algorithms such as SVM, KNN, decision trees and incorporated learners are tested. According to the results from studies, processes based on ensemble strategy have been shown to yield more precise and consistent results [3]. Multiple machine learning techniques are used in the same paper, "Improved Segmentation Approach for Plant Disease Detection" [4], to suggest the fertilizer based on the crop. Factors which are considered as input variables are like nitrogen, phosphorus, soil acidity and rainfall whereas methods being evaluated are the accuracy and efficiency of each of these methods.

A second important area for interest would be in detecting plant diseases on the analysis of leaf conditions. The paper titled "Image processing techniques can be used to detect diseases of different plant leaves: A Survey" [5] discuss the use of multiple technological ways to improve the health and yield of plant. Visual indicators of infections on leaves are used to analyze through deep learning architectures such as CNNs. These models offer robust solution to predict the diseases based on the image data.

Relevant other paper about "Smart Agriculture Forecasting using Machine Learning" [6] focuses on building an effective predictive farming assistant. It starts off by introducing conceptual components and later provides a system that can make dynamic suggestions for highly fragmented farmland. The system adopts the framework of Precision Agriculture and allows for hyper local decisions as low at the plot level. Since the tools serve to provide advice using basic communication channels (e.g. SMS, email) in areas underserved by digital communication, farmers will be able to gain from using the model's advice.

Rich datasets and global challenges have spurred innovation in the field of computer vision with technological advancements. It wasn't long after the ImageNet dataset released and the ILSVRC were released, that it got hugely developed with AI image recognition. This progress has, to a great extent, helped to solve the problems of images applied to agriculture mostly when data availability is limited. Over 80,000 WordNet categories are provided with image representation in ImageNet, and moreover, it has between 500 and 1,000 labelled images for a class. The existing CNN models were then proposed over time, in order to continue to improve the classification capabilities.

They include AlexNet [7] with five convolutional layers, VGG with 19 layers, ResNet with addressing the gradient loss issue by employing the shortcut pathways, MobileNet [8] designed to run on mobile devices, and EfficientNet which uniformly alters scale parameters to achieve the depth, width and resolution balance. They have been put to practical use on agricultural diagnostics and disease identification networks[9].

Agricultural AI became a prominent branch in identifying plant infections. The use of CNN based systems is gaining popularity in the innumerous papers which present advanced

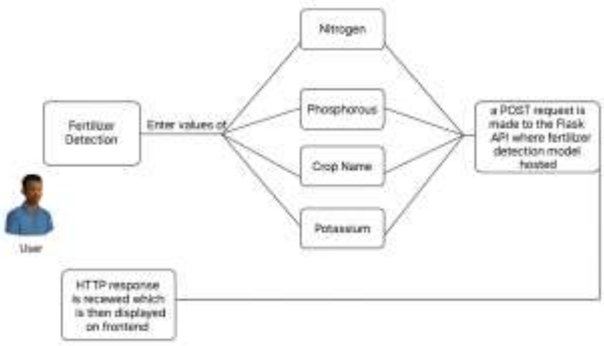

Figure.1.fertilizer detection

strategies. The one study [10] includes preprocessing techniques such as noise filtering, cropping, detecting edges and using Otsu's method for image segmentation. Next, it feeds these essential visual patterns into an ANN to extract those things like shape, color, and texture.

The PlantVillage dataset is used to evaluate the CNN frameworks such as AlexNet, and GoogleNet, both with and without transfer learning, and up to 99.35% accuracy are achieved [11]. The visualization techniques were also applied on new collected image data from online platforms such as Bing and Google, and this data was tested on these models as well. Finally, there is a wide-ranging [12] analysis of over 100 studies of deep learning in disease detection, finding vast majority of these use PlantVillage dataset and generally use off the shelf pretrained architectures like VGG, ResNet, DenseNet, or Inception.

This review covers additional tools such as feature maps, saliency maps, and heat maps that explain how the AI models come to a conclusion. Next, in another experiment [13], we used 87,000 images in this dataset for testing the use for these networks (VGG, Resnet, Inception V3) and VGG found to be the best. According to the findings of [14] on recent research pertaining to pest and disease detection (2014 – 2020), it is partly based on practicing multiple improvements in locating vital image areas.

A single comparison [15] exists between traditional ML approaches and deep learning to detect plant diseases. There are however some tools for mobile and web that do not tackle real world conditions with natural leaf images. The main limitation is the lack of generalization and the high image complexity. A previous study [16] compares other CNN architectures such as VGG 16, ResNet 50, Inception V4 and

DenseNet 121 with PlantVillage data. In [17], work is done on environmental and soil based crop recommendations using models like SVM, Random Forest, ANN. Random Forest was the strongest performer in that most supported a GPS enabled mobile app to predict yield and determine crop selection. Similarly, [18] uses a voting based ensemble of models such as CHAID, KNN, Naive Bayes, Random Trees for the crop suggestion. Additional works [19]–[22] determine the suitability of different fertilizers under other conditions, e.g., NPK levels, soil pH, depth and weather, and region. Usually, it comes with rule based logic or uses K Mean and Random forest based algorithms.

For interpretability, its readability is improved by techniques such as LIME[23] and Grad-CAM[24]. Both Grad-CAM and LIME locate influential points by dotting boundary layer end points with gradient direction, LIME using the direction to affect prediction variation, while Grad-CAM finding the best direction to color the coating layers. Understanding and trust are on the scene with these critical tools to use in novel agriculture. A few of these include their implementation of enhanced ability to design intelligent platforms that will assist farmers in designing the best trees, disease management, and soil solutions, with equipment capable of artificial intelligence and viewing ability.

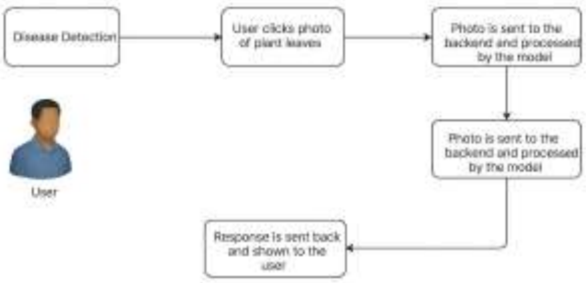

Figure-2.disease detection

III. METHODOLOGY

The following part speaks about the system structure as a whole, i.e.used AI models, testing process and training process. Given that, we begin our start by presenting the design of the said platform with graphical representations like structural charts and interface flowcharts. We then describe intelligent model modules with the used specific algorithms for the functionalities and mechanisms of testing. Elements for fertilizer advice, plant disease identification and crop choice are sub sections. Another collection of sections explains how LIME can be used to explain the model under the machine learning category, and how live farm news can be integrated into the interface features.

Figure-3.fertilizer

A. Application Overview

1) Fertilizer Recommendation Engine :

Users give the soil nutrients in forms of nitrogen, phosphorus and potassium and the crop planted to get customized fertilizer recommendations. Then through a POST call, it sends these to the backend flask service which has the fertilizer prediction model and the prediction is made. The response to the interface will include fertilizer recommendations based on the soil condition after processing.

2) Plant illness Detection

Figure-4.crop prediction and causes

It is from the users who, either, can upload a picture of dama ged plant or he/she can take it within the app. The server ana lyzes the picture using a pre-trained convolutional neural network. It returns diagnosed c ondition along with the potential remedies as an HTTP respo nse. A respective workflow diagram (see Fig. 2) of this whol e process is shown

3) Crop recommendation engine :

POST request with the information on nutrient levels (N, P, K) is sent to the backend Flask server. A trained machine lea rning model takes in inputs and provides a prediction over th e best crop to plant to maximize yield. This is explained in t he system diagram (see Fig. 3)

4) Crop Diseases Knowledge Base:

The platform also comprises a knowledge portal that informs users about different crop diseases, their visible symptoms and remedies. It serves as a digital assistant, guiding users in their choices to promote the health of their plants.

5) Explainable AI (LIME Support) :

After a user uploads an image, it is sent to a hosted server on DigitalOcean, where the LIME framework is applied to identify which areas in the image had the most contribution to the AI's decision making. These results are as hyperlinks created as a web link formatted visually within the front-end of the app for the user to inspect and reference.

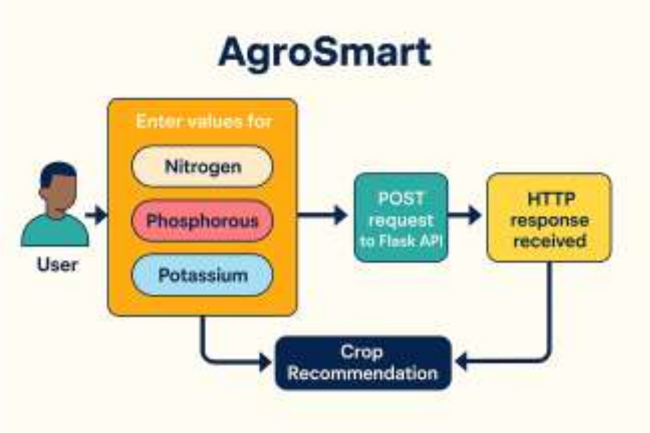

Figure-5.Agrosmart

B.Crop Recommendation Dataset Analysis and Approach

Dataset Details :

For the purpose of crop prediction, we used a public dataset from Kaggle[1]. It was made simple but efficient by targeting only essential environmental and soil factors. The dataset consists of seven fundamental features: Nitrogen, Phosphorus, Potassium levels in the soil, temperature in degrees Celsius, humidity as a percentage, pH of the soil, and rainfall in mm. The aim is to associate these features with the most appropriate crop for an environment.

There are 2,200 records, which are uniformly distributed across 22 crops, namely rice, maize, muskmelon, and coffee --there are exactly 100 entries in each class of crop. With such uniform distribution, the dataset is well-balanced, and complexities of working with skewed class representation are minimized.

Model Development Process:

In order to make sure that our findings were consistent and reliable, we employed a five-fold cross-validation method while training. We tried out six various machine learning models:

A Decision Tree model, set to use entropy as the splitting criterion and a maximum depth of 5

A Gaussian Naive Bayes model

A Support Vector Machine (SVM) with a polynomial kernel of degree 3, L2 regularization (C=3), and inputs scaled from 0 to 1

A K-Nearest Neighbors (KNN) model

A Random Forest model

An XGBoost implementation from its specialized library

All models except for XGBoost were implemented with the Scikit-learn package. Those parameters not individually tuned were left on their default values. The model that proved to have the best performance as measured by test accuracy and capacity to generalize was chosen for production.

Plant Disease Identification – Dataset and Technique

Dataset Summary:

For identifying plant diseases, we referred to an improved variant of the popular PlantVillage dataset[2], which is also available on Kaggle. It has approximately 87,000 high-resolution color images of both healthy and diseased plant leaves. These are distributed across 38 classes—14 of crops and 26 of diseases. On average, each class contains about 1,850 images with a standard deviation of around 104.

We split the dataset into two subsets: 80% of the images were trained on, and the other 20% was reserved for validation. The task of the system was to classify both the plant species and the particular disease based solely on the leaf image.

All images were normalized (divided pixel values by 255) and resized to 224x224 pixels to standardize them and align with the input requirements of the deep learning model. A graphical representation of the processed dataset batch is shown in Figure X.

Training Strategy

To construct a strong plant disease detection model, we utilized three top-performing deep learning architectures—VGG-16, ResNet-50, and EfficientNetB0. These architectures are widely used in the computer vision community and were initially trained on the gigantic ImageNet dataset. Instead of beginning from scratch, we fine-tuned these pre-trained networks with our own particular dataset, as this tends to provide superior outcomes, particularly when dealing with small agricultural datasets.

For training, we used the Adam optimizer with well-chosen hyperparameters: a learning rate of 0.00002, $\beta1 = 0.9$, $\beta2 = 0.999$, and an epsilon value of 1e-08. The training configuration was for a batch size of 32 and 25 epochs. Categorical cross-entropy was the loss function we used. To avoid overfitting and retain the highest--performing model, methods such as early stopping and model checkpoints wereused--performance-based on the validation set. Accuracy per epoch was monitored to monitor training behavior.

Hardware & LIME-Based Interpretability

Given the computationally intensive task of image classification, we used available free GPU resources on platforms such as Kaggle and Google Colab for training and model inference. To provide increased transparency of model choices, we included LIME (Local Interpretable Model-Agnostic Explanations) from the lime Python package.

During the evaluation, LIME computed 1,000 perturbed samples of an input image to mark the ten most impactful regions that had affected the model's prediction. The marked areas were presented to users through the application interface in order to facilitate trust and understanding. At runtime, to optimize efficient performance under hardware limits, we lowered the number of analyzed samples to 249 per prediction.

Fertilizer Recommendation – Dataset and Approach

Dataset Insights :

To give fertilizer advice, we designed a special dataset related to significant inputs for soil health evaluation. It is comprised of five features: name of the crop, nitrogen, phosphorus, potassium, pH, and content of soil moisture. There are 22 common crops in the dataset, e.g., maize, rice, coffee. Each record has the best possible nutrient values needed for proper growth of the respective crop. The purpose of this dataset is to inform fertilizer application based on nutrient deficiencies in the soil that have been detected.

Solution Strategy :

We applied a simple yet consistent rule-based approach to design the fertilizer advisory system. The logic-based system employs a set of IF-THEN rules to test if the given NPK values are less than or greater than the optimal requirement of the crop. On the basis of these tests, it suggests one among six pre-defined fertilizers that are customized to rebalance the imbalance.

This effective and simple rules engine, integrated in the user interface, is so simple that even farmers and field experts can easily understand it. As it is integrated in the user interface itself, real-time suggestions are offered on the basis of the input made by the user. In this way, farmers are offered actionable recommendations in real time, which helps make better and quick decisions regarding maintaining the nutrient level in their fields.

IV. RESULTS AND DISCUSSION

A. Crop Recommendation Module

Our crop suggestion system's output is graphically illustrated in the following figure, with main performance metrics detailed in Table I. For further ease of performance comparison, the bar chart is also presented in Figure 5. From test results, the XGBoost model ranked top with the best accuracy, followed by Random Forest and Gaussian Naive Bayes. Generally, ensemble approaches—like Random Forest (bagging) and XGBoost (boosting)—consistently outperformed single algorithms.

Figure-6.crop prediction

Using intensive cross-validation, we found that the Random Forest model performed extremely well with a 99.5% accuracy. Due to its stable predictions and capacity for explaining how it makes decisions, we chose to use it as our main model for our crop recommendation system. One of the strengths of Random Forest is transparency—it provides us with the means to understand the effect of every input through feature importance analysis.

The role of individual characteristics, as predicted by this model, is depicted in Figure 6. Out of all variables, rainfall was the most significant factor in deciding the optimal crop to be grown—clearly reflecting the role of water availability. Humidity was the second most important variable. Other important contributors were potassium, phosphorus, and others. This ranking not only improves prediction accuracy but also provides farmers with a better sense of which soil and climatic parameters directly determine crop choice—enabling wiser agricultural choices.

B.Plant Disease Detection Module

In our comparison of various deep learning models for plant disease classification, the EfficientNet model always produced the highest accuracy values—beating established models such as VGG-16 and ResNet-50. This is clear in Figure 7, where we compare the validation results of the models.

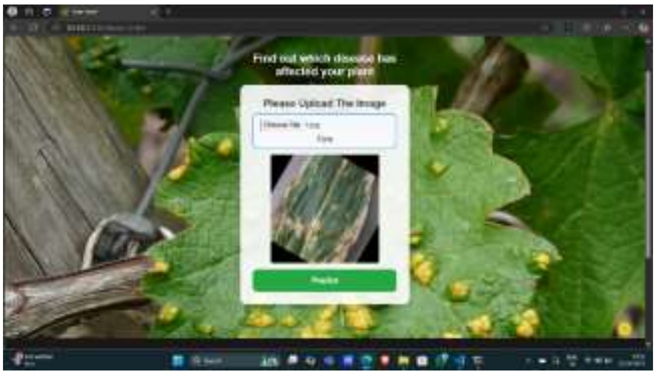

Figure-7-plant disease detection

Validation accuracy and loss behaviors, presented in following figures, indicate that EfficientNet converges faster and achieves greater accuracy stabilization compared to the others. EfficientNet is a new class of CNNs trained with compound scaling, enabling the model to expand in depth, width, and resolution in balanced proportions.

Due to its stable performance and effective utilization of computational resources, EfficientNet was finally selected to drive the plant disease detection function in our app. Its precision in detecting diseases from leaf images makes it well-suited for real-time and mass deployment.

C. LIME-Based Model Interpretability

To keep our system explainable and transparent, we added the LIME framework (Local Interpretable Model-Agnostic Explanations) to the disease detection pipeline. This approach graphically indicates the precise regions of an input image that the model relies upon in order to produce its predictions—basically allowing users to have a peek into the "thought process" of the neural network.

As shown in Figures 9 and 10, the salient regions of the image marked by LIME overlap with the true infected regions of the leaves highly. This overlap ensures the model is actually concentrating on the right visual features when producing its disease prediction. The explanation mask boundary regions that contribute significantly to determining the model's output offer reliability as well as user confidence.

While our present configuration takes advantage of a generous number of explanation samples, the follow-ups of ours will incorporate more of the same so that more precise analysis is available. Optimizations are available by utilizing high-resolution input images as well as finer segmentations that could further shorten the model's focal length. It could be possible so while at the same time keeping interpretability intact and optimizing inaccuracy again.

In short, adding interpretability to our pipeline not only instills faith in the system output but also provides valuable insights into the dataset itself—information that is good for continued model refinement.

V. CONCLUSION AND FUTURE WORK

Farmer's Assistant, this paper presents a machine learning-powered web-based application that assists farmers in making sound agricultural decisions. The app provides a user-friendly and interactive interface and features crop disease diagnosis through an EfficientNet-based image classifier, fertilizer recommendation by rule-based logic, and crop recommendation through a Random Forest classifier.

With easy web forms, users can simply enter data and get near-instant feedback and insights. For the purpose of transparency and establishing user trust, the system employs LIME (Local Interpretable Model-Agnostic Explanations) for the disease detection module. This enables users to visually observe which regions of a leaf image affected the model's prediction, allowing for better model behavior understanding, error area identification, and dataset improvement.

Though the existing version of the app works steadily, a few upgrades are in the offing for the future. One of the big upgrades could be adding product suggestions for fertilizers and seeds from online shopping websites. This would allow users to buy the suggested farm inputs directly, and the system would be not only advisory but also actionable.

The fertilizer module currently provides recommendations on six generic types of fertilizers based on imbalances in the nutrients nitrogen, phosphorus, and potassium. This could be expanded in subsequent versions to include brand-specific and more precise suggestions by including online retailer data through web scraping or APIs. This would give farmers specific, context-dependent advice.

A limitation known to exist in the current disease classification model is that it relies on the dataset at hand. The model is tuned specifically for the conditions and crops on which it has been trained, and its accuracy may suffer when it meets unseen or uncommon conditions. To better address this, the system would be augmented by using larger and more variegated datasets, either available publicly or produced by data augmentation methods. This would make the model more adaptable and precise when operating in real-world applications.

Besides that, we also want to build a user driven image annotation platform inside the app, where farmers will post leaf images and give annotations or labels to the diseases identified by them. The more this crowd sourced information comes in, the richer and constantly evolving the dataset will be, the accuracy of the model itself will improve as well as the model will always stay relevant to local farming challenges.

The current form of LIME can help explain individual predictions at a point of interest, but does not introduce global interpretability, handling of errors, or broader use within a model. In future, global interpretability techniques such as Grad-CAM or Integrated Gradients may be integrated to give a better understanding of the decisions made by the model. The visual explanation could be further improved using more advanced version of LIME or other hybrid explainer which are more stable and clearer.

The system is limited in that it does not yet provide the fine segmentation of infected area within a plant leaf. The reason for this largely lies in the absence of pixel level labelled data. Later on, we want to create a manual labelling feature which will enable users to mark the infected regions by themselves. This user input combined with unsupervised segmentation throughput would substantially improve precision and visualization of disease detection.

Finally, We summarize the results of 'Farmer's Assistant' by stating that it is well poised as an intelligent agricultural support tool. The platform can then evolve into a more comprehensive, smarter, and collaborative solution for farmers of various regions through the incorporation of real time product recommendations using broader data sets, user input, enhance interpretability methods and pixel level disease segmentation.

REFERENCES

[1] S. D. Khirade and A. B. Patil, "Plant disease detection using image processing," in *Proc. Int. Conf. Comput. Commun. Control Autom. (ICCUBEA)*, Pune, India, 2015, pp. 768–771, doi: 10.1109/ICCUBEA.2015.153.

[2] A. V. Panchal, S. C. Patel, K. Bagya Lakshmi, P. Kumar, I. R. Khan, and M. Soni, "Image-based plant diseases detection using deep learning," *Mater. Today: Proc.*, vol. 45, pp. 5434–5439, 2021, doi: 10.1016/j.matpr.2021.07.281.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Miami, FL, USA, 2009, pp. 248–255, doi: 10.1109/CVPR.2009.5206848.

[4] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015, doi: 10.1007/s11263-015-0816-y.

[5] G. A. Miller, "WordNet: A lexical database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Adv. Neural Inf. Process. Syst.*, vol. 25, 2012.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv:1409.1556, 2014. [Online]. Available: https://arxiv.org/abs/1409.1556

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[9] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," arXiv:1704.04861, 2017. [Online]. Available: https://arxiv.org/abs/1704.04861

[10] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.

[11] D. P. Hughes and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," arXiv:1511.08060, 2015. [Online]. Available: https://arxiv.org/abs/1511.08060

[12] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection,"

Front. Plant Sci., vol. 7, p. 1419, 2016, doi: 10.3389/fpls.2016.01419.

[13] Y. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Comput. Electron. Agric.*, vol. 145, pp. 311–318, 2018, doi: 10.1016/j.compag.2018.01.009.

[14] J. Liu and X. Wang, "Plant diseases and pests detection based on deep learning: A review," *Plant Methods*, vol. 17, p. 22, 2021, doi: 10.1186/s13007-021-00722-9.

[15] R. U. Khan, K. Khan, W. Albattah, and A. M. Qamar, "Image-based detection of plant diseases: From classical machine learning to deep learning journey," *Wireless Commun. Mobile Comput.*, vol. 2021, Art. ID 5541859, 13 pages, 2021, doi: 10.1155/2021/5541859.

[16] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," *Comput. Electron. Agric.*, vol. 161, pp. 272–279, 2019, doi: 10.1016/j.compag.2018.03.032.

[17] S. M. Pande et al., "Crop recommender system using machine learning approach," in *Proc. Int. Conf. Comput. Methodol. Commun. (ICCMC)*, 2021, pp. 1066–1071, doi: 10.1109/ICCMC51019.2021.9418351.

[18] S. Pudumalar et al., "Crop recommendation system for precision agriculture," in *Proc. Int. Conf. Adv. Comput. (ICoAC)*, 2017, pp. 32–36, doi: 10.1109/ICoAC.2017.7951740.

[19] K. S. Subramanian, "Design and implementation of fertilizer recommendation system for farmers," 2020. [Online]. Available: https://example.com

[20] A. Pratap et al., "Soil fertility examination and manure suggestion framework," in *Proc. Int. Conf. Adv. Comput. Manag. (ICACM)*, 2019. [Online]. Available: https://ssrn.com/abstract=3446609

[21] A. Palaniraj, A. S. Balamurugan, R. D. Prasad, and P. Pradeep, "Crop and fertilizer recommendation system using machine learning," *Int. Res. J. Eng. Technol. (IRJET)*, vol. 8, no. 4, 2021.

[22] G. Preethi et al., "Agro-based crop and fertilizer recommendation system using machine learning," *Eur. J. Mol. Clin. Med.*, vol. 7, no. 4, pp. 2043–2051, 2020.

[23] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, San Francisco, CA, USA, 2016, pp. 1135–1144, doi: 10.1145/2939672.2939778.

[24] R. R. Selvaraju et al., "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 618–626, doi: 10.1109/ICCV.2017.74.