Poster No 34

AI-Powered Agriculture Advisor: Crop Recommendation, Yield Prediction and Disease Detection Using Soil, Weather, and Image Data

Presenter Name: Muhammad Mubeen

Professor Name: Prof. Xiaohong (Sharon) Gao, Department of Computer Science, Middlesex University, London, NW4 4BT.
Professor Email: x.gao@mdx.ac.uk

Middlesex University

Introduction

Agriculture is the backbone of food security, yet farmers often struggle with three major challenges: choosing the right crop for their soil, protecting crops from diseases, and predicting yield under uncertain weather conditions. Traditional decision-making relies on experience and guesswork, which can lead to significant losses.
To address this, we developed an **AI-Driven Agriculture Advisor** that combines **machine learning and deep learning** to support farmers with actionable insights. Our project is built around three components.

Crop Recommendation: Using soil nutrients (N, P, K), pH, temperature, humidity, and rainfall, a Random Forest Classifier suggests the most suitable crop.

Plant Disease Detection: Using leaf images, both an EfficientNetB0 model (transfer learning) and a custom CNN (22 layers, built from scratch) classify 15 plant diseases and healthy cases.

Yield Prediction: Leveraging 19 years of farm weather records with crop yield statistics, a Random Forest Regressor estimates future yield performance, linking soil and climate to crop outcomes.

By integrating these three components, the system helps farmers make smarter decisions before planting, during crop monitoring, and at harvest time. The outcome is a pipeline that transforms raw soil, weather, and image data into **practical recommendations**, offering a step towards sustainable and data-driven farming.

Implementation and Approach

The implementation and approach of this project is consist of three components which are as follows
1. **Crop Recommendation**
2. **Yield Prediction**
3. **Plant Disease Detection**

1. Crop Recommendation Using Random Forest Classifier

So in this component we have 2 different type of datasets which consist of multiple soil features. Our first dataset which we called **Basic Dataset** has 7 key features of soil include (N,P,K pH, Temperature, Humidity, Rainfall). And Second Dataset which we called **Seasonal Dataset** has 28 features 7 are same as basic one and other includes (soil colors, maxtemp, min temp and other feature). Both datasets went through a round of preprocessing. This included standardizing the feature names, removing duplicate entries, encoding categorical fields like soil color, filtering out features that were almost constant or too highly correlated, and finally applying stratified train–test splits to keep the data balanced. We have applied a random forest classifier where we selected 300 trees to make a decision in both the dataset and our target was crops. Since Crops is the categorical field we used Random Forest Classifier. Random Forest Classifier was chosen for modelling due to its robustness with tabular agricultural data, ability to capture non-linear patterns, and resilience to noise.

1.1. Results Of Crop Recommendation Model

After applying Random Forest in both the datasets, we find out that basic dataset is providing very strong result in terms of precision, accuracy while the other dataset providing the accuracy around 50% with f1 score Below 0.25. We have done more preprocessing by reducing feature which has below 1 feature Score but after again preprocessing the dataset the accuracy and F1 score got much worsen. So we have decided to go with the basic dataset.

- ***Figure 1*** *Shows the Result of Basic Dataset*
- ***Figure 2*** *Shows the Result of Seasonal Data Set*
- ***Figure 3*** *Shows the imbalance of the features in seasonal dataset*

Basic Dataset	
Accuracy	0.993
Precision Macro	0.993
Recall Macro	0.993
F1 Macro	0.993

Figure 1

Seasonal Dataset	
Accuracy	0.51
Precision Macro	0.27
Recall Macro	0.23
F1 Macro	0.23

Figure 2

Feature Score Of Seasonal Dataset	
K, N, Zn	0.14, 0.12, 0.12
S, P	0.12, 0.11
pH	0.12
All Other Features	Below Zero

Figure 3

2. Yield and Weather Prediction Using Random Forest Regressor

We built the yield and weather module by bringing together two separate datasets. The first one was a global crop yield dataset (1990–2015), which lists yearly yields for different crops in various countries, along with broad indicators like rainfall, pesticide use, and temperature. The second dataset was a local weather record (2006–2024) that included daily readings of temperature, humidity, wind, and rainfall. Since the two datasets covered different time periods, we had to align them. To do this, we kept the yield data from 1994–2013 and shifted it forward like a data manipulation we manipulated the year field of it into 2006–2024 so it lined up with the 2006–2024 range in the weather dataset. Any earlier years that didn't match were dropped. We then averaged the daily weather values into yearly figures, standardized the crop and column names, and removed entries that couldn't be paired. After these steps, we ended up with a clean, merged dataset by doing inner join from year column that could be used to train a regression model for predicting yield per hectare based on seasonal weather patterns.

2.1. Results Of Yield & Weather Prediction

So after applying the Random Forest Regressor we achieved an R^2 score of 0.982, indicating that the model explains approximately 98.2% of the variance in crop yield. The Mean Absolute Error (MAE) was 4,174 hg/ha (≈417 kg/ha), meaning that on average, predictions deviate from the actual yield by about 417 kilograms per hectare. And For weather we used weather data in a way that it will combined the previous 10 years of weather data and give the predicted as per the average weather for the upcoming year for the user to use in soil recommendation model. Below are some figures to justify the result of this model

- ***Figure 4*** *The scatter plot shows that predictions align closely with actual yields, with most points clustered along the 45° line, indicating strong model fit.*
- ***Figure 5*** *Prediction errors are centred around zero and mostly small in magnitude, showing that the model is unbiased and consistently accurate*
- ***Figure 6*** *For Wheat, predicted yields closely follow actual year-to-year variations, confirming that the model generalizes across time and does not simply capture mean values*
- ***Figure 7*** *The model identifies crop type (e.g., Potatoes, Cassava) as the strongest predictors of yield, with contextual features such as pesticide usage, average temperature, and regional factors also contributing significantly*

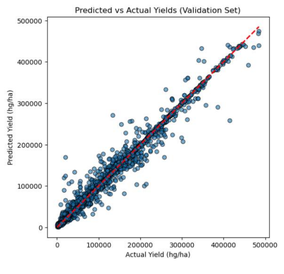

Figure 4

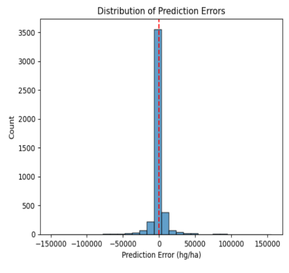

Figure 5

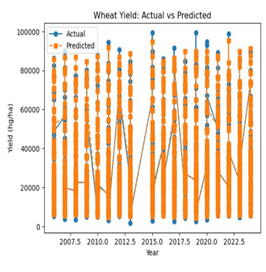

Figure 6

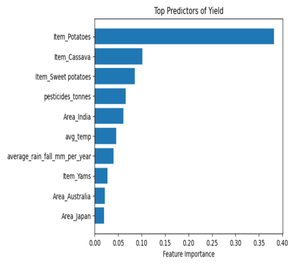

Figure 7

3. Plant Disease Detection Using CNN:

In this component our goal is to detect user upload images and give them the result if the plants are in healthy condition or have any disease if it has any disease what Is it. We are using plant village dataset in this model consist of 20000 images, the datsaset consit of 15 classes. Mainly consist of 3 plants in a way that 12 classes are distributed into different diseases of these 3 plants and 3 classes consist of 3 healthy plants images of each crop. first we Implement this model using 2 method in CNN.

1. CNN Using EfficientNetB0
2. Own CNN from Scratch

For this component, both the EfficientNet model and the custom CNN shared the same preprocessing pipeline. The dataset was first split into training, validation, and testing sets across 15 crop disease and healthy classes. All images were resized to 224×224 pixels, normalized to a [0,1] scale, and labels were one-hot encoded for multi-class classification. To improve generalization and reduce overfitting, basic data augmentation such as random flips and rotations was applied. This consistent preprocessing ensured fairness in comparison between the two approaches, with differences in performance driven only by the architectures rather than the input data.

In this task, we implemented two complementary approaches. First, we applied **transfer learning with EfficientNetB0**, freezing its pretrained layers to leverage general image features, then unfreezing the last 40 layers to fine-tune disease-specific patterns and improve accuracy. In parallel, we designed a **custom CNN from scratch**, consisting of stacked convolutional, pooling, and dense layers to learn directly from the dataset. This dual approach allowed us to compare the benefits of a lightweight, tailored CNN versus a deep pretrained architecture for the same problem. The CNN from scratch consist of 4 convolutional Layers. Below three images are the three classes of dataset.

- ***Figure 8*** *Early Blight of Potato Plant*
- ***Figure 9*** *Late Blight of Potato Plant*
- ***Figure 10*** *Healthy Plant Of Potato*

Figure 8

Figure 9

Figure 10

3.1 Results Plant Disease Detection Using Both CNN:

After training the dataset using both the techniques we achieved a quite strong result. Below are the figures to describe the result of both the models.

- ***Figure 11*** *Shows the Result Of CNN from Scratch*
- ***Figure 12*** *shows the Result of CNN from EfficientNetB0*

CNN From Scratch	
Accuracy	0.919
Precision Macro	0.910
Recall Macro	0.9237
F1 Macro	0.9133

Figure 11

CNN From EfficientNetB0	
Accuracy	0.939
Precision Macro	0.930
Recall Macro	0.943
F1 Macro	0.9333

Figure 12

Overall Project Work Flow from User Perspective:

We have created a basic UI to see how will the whole model work combined. From Farmer perspective it should work like this when farmer inputs the values of N, P, K, pH, Country Name, Year of farming. It will work in a way that it will give the top 3 crop suggestion from the soil recommendation model the weather it will get from the merged yield dataset with the average of last 10 years according to the country farmer gave and when it will provide the 3 crop farmer could check the predicted yield according to the dataset of merged yield prediction for these 3 crops. And other part of the Project is user upload the image it will provide the output of whether the palnt is healthy or have any disease.

Figure 13 *shows the workflow of the project*

Figure 13

Future Directions

In Future what we can do is to enhance this whole project we can add some function to make it a complete Farming solution. Functions that should be implement in future are as follows:

Remedies Against the detected disease: **We can add the webscrapping model to train it against the classes of each disease so it can provide possible remedies for the farmers**
Adding more plant classes: **Adding more plant images into the trained models**
Monitoring of farming: **Monitoring of farming by training the every week plant images during farming**

References:

[1] Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, 7, 1419.
[2] Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, 147, 70–90.