Project Report - AI-Powered Early Detection of Crop Diseases in Kenyan Smallholder Farms

Table of Contents:

1. Introduction

1.1 Challenge Background

In 2022, the agriculture sector accounted for 20% of the GDP of Kenya, and it employs 40% of its population. Kenya's 7.5 million smallholder farmers contribute up to 80% of agricultural produce. However, smallholder farmers in Kenya face substantial challenges in maintaining crop health due to a lack of access to timely and accurate crop disease diagnosis. This gap often results in significant crop losses, which not only undermine the livelihoods of these farmers but also contribute to broader issues of reduced food security in the region. Early detection and treatment of crop diseases are critical to minimizing these losses, as timely intervention can prevent the spread of infections and enhance yields. However, achieving this is a formidable task, as many smallholder farmers operate in remote areas with limited access to agricultural extension services, diagnostic tools, and expert guidance. These challenges highlight the urgent need for innovative solutions to bridge the knowledge and resource gap, ensuring farmers can protect their crops and sustain their communities.

1.2 Motivation

The project aims to empower smallholder farmers in Kenya by providing a highly accurate mobile application for crop disease detection, designed with offline capabilities and rapid performance. By increasing disease diagnosis accuracy, reducing crop losses, and improving yields, the initiative seeks to enhance farmers' livelihoods, boost food security, and foster strong partnerships with agricultural extension services.

1.3 Problem Statement

To minimize significant crop losses and ensure food security, the project aims to design a machine-learning model that recognizes the major commonly grown crops grown in Kenya and the most common diseases they face. The model must maintain high accuracy in order to reflect reliable results in real-time detection and diagnosis. Another core problem is making the detection tool scalable, easy to use, and accessible to bridge the technological gap for farmers.

2. Collaborators

Tushar Aggarwal(Community Head)
Haron Hama-Product Owner
Kunal Sood -Product Owner
Barbara Obayi -Product Owner
Sammer Ali
Riddhi Rai
George Odero
Dinesh Kumar Raju
Youssef Elkilany
Sanjiban Santra
Dhavni
Luis Gallego
Chris Coble
Samyuktha
Jo Alex
Steven Christolis
John Paul (JP)
Stephen Munge
Saumya Gupta
Annie Ndungu
Nicky Chemos
Ajibola Oyedeji
Rosephine Maideyi
Adrian Chavarro
RJ Aca-ac
Jonh Vincent Alvaro
Oliver Brown
Peter Murimi
Wallace Fang
Peter Ibikunle
R Reshma
Rishabh Singh
Kelvin Ngumo
Savadogo Abdoul
Faten Ragab
Dinesh Kumar
Luis Okech
Stephen Gathiru
Hannah Kariuki
Maheshika
Mofajjel Hossain
Yamkela
Shivam Sourav
Bhavani Priya
Akreti Sharma

3. Literature Review

Crop production is vital to Kenya's economy, providing income, employment, and food security while supporting 10 million households through local and export markets. Despite contributing significantly to agricultural output, the sector faces challenges, including a 40% loss in potential yield due to pests, diseases, and weeds. Consequently, there is an inability to meet domestic demand for staples and industrial crops. To address these issues and enhance productivity, modern agricultural research focuses on biotechnological advances, integrated pest management, and soil value addition to boost health and sustainability.

Based on the mission's goal, literature review, and group consensus from the global team's research, the team decided to model only four crops: beans, cassava, maize, and tomatoes. These crops are vital to nutrition in Kenya. Protecting their growth and harvest is crucial, as farmers currently face a 40% loss in yield. This loss directly impacts caloric sustenance for Kenyan people, increases demand for imports, and limits participation in global markets. Technology, such as an artificial intelligence-based crop disease detector, creates opportunities to enhance the crop harvest and economic power in Kenya.

Other research groups have attempted to use machine learning and visual information to encode and predict diseases within crops. One approach leverages spectral vegetation indices to develop disease detection models for maize, using hyperspectral remote sensing to analyze spectral reflectance from healthy and diseased leaves across growth stages. Spectral indices like NDVI (Normalized difference vegetation index) and CRI (Carotenoid Reflective Index) proved effective in identifying disease-related spectral variations, particularly during the vegetative stage, highlighting the potential of spectral information for precise and timely disease detection in maize.

Current research highlights previous applications of computer vision (CV) and machine learning (ML) in disease detection for crops like maize and tomatoes, demonstrating their potential to revolutionize agriculture. In one paper, a MobileNetV2-based model powers a mobile application that enables farmers to diagnose crop diseases in real-time. Despite challenges like data limitations and accessibility, the study emphasizes the transformative role of CV and ML in improving food security, reducing crop losses, and advancing agriculture. It is the goal of the project to emulate this application, specifically aimed at detecting disease in not only maize and tomatoes, but also beans and cassava, expanding the utility of the app.

The literature sources compiled in this project demonstrate the potential of using big data and modeling to create an invaluable resource for farmers in Kenya: a crop disease detector.

Developing this crop disease detector will protect crops, strengthen the economy, and support the well-being of everyone dependent on Kenyan agriculture.

4. Methodology

4.1 Data Collection

With the creation of an application for detecting crop diseases, data collection was the first step. The team scavenged the internet, searching for images of different diseases for the following crops: cassava, beans, tomato, and maize. The table below indicates which diseases had photographic datasets available online:

Crop:	**Beans**	**Cassava**	**Maize**	**Tomato**
Diseases:	1. Angular Leaf Spot 2. Bean Rust	1. Bacterial Blight 2. Brown Streak Disease 3. Green Mottle 4. Mosaic Disease	1. Lethal Necrosis 2. Gray Leaf Spot 3. Common Rust 4. Northern Leaf Spot 5. Maize Streak Virus	1. Bacterial Spot 2. Early Blight 3. Late Blight 4. Leaf Mold 5. Black Mold 6. Mosaic Virus 7. Septoria Leaf Spot 8. Target Spot 9. Yellow Leaf Curl Virus

In addition a healthy class was also included for all crops listed above in order to provide the model with a true representation of the variance within the crops' appearances.

4.2 Data Preprocessing

Our approach to preprocessing was comprehensive and aimed at enhancing the quality and readiness of our dataset for model training. The steps are detailed in the following sections: image and metadata review, image processing techniques, and metadata validation.

4.2.1 Image and Metadata Review

Each image was cross-checked with its corresponding metadata to confirm the accuracy of the disease label. This step was crucial in maintaining the integrity of our dataset, as incorrect labels could lead to incorrect training and poor model performance.

Any discrepancies found between the image content and the metadata were addressed by updating the metadata. This iterative process ensured that our dataset was as accurate as possible, which is essential for the reliability of our AI-powered disease detection system.

4.2.2 Image Processing Techniques

The images were resized to a consistent resolution to standardize the input for our model. This step helped in reducing the computational load and ensured that the model was not biased towards images of certain sizes.

To enhance the robustness of our model, we employed data augmentation techniques such as rotation, flipping, and zooming. These techniques artificially expanded our dataset by creating variations of the existing images, which helped the model generalize better and reduced the risk of overfitting.

The pixel values of the images were normalized to scale them appropriately. This step is important as it ensures that the model is not biased towards certain pixel intensity ranges and helps achieve faster convergence during training.

4.2.3 Metadata Validation

Alongside image processing, we performed rigorous quality checks on the metadata. This included verifying the completeness and consistency of the data and ensuring that each image had the necessary accompanying information for accurate disease classification.

4.3 Model Development

The team integrated the data into three preexisting convolutional neural network architectures (YOLO, ResNet, and ViT), evaluated their performance, and compared the results across the models.

YOLO (You Only Look Once) is a real-time object detection architecture that frames object detection as a single regression problem, directly predicting class probabilities and bounding boxes from an input image in one evaluation of a neural network. It is known for its speed and efficiency compared to other detection methods.

ResNet (Residual Network) is a deep neural network architecture designed to address the vanishing gradient problem and improve the training of very deep networks. Introduced by Microsoft Research in 2015, ResNet achieved groundbreaking results in image classification, including winning the ImageNet competition.

ViT (Vision Transformer) is a neural network architecture that applies the principles of transformers, originally developed for natural language processing (NLP), to image classification tasks. It was introduced by Google Research in 2020 and has since become a significant innovation in computer vision.

5. Model Deployment

The bulk of creating a disease detecting application in crops comes from the modeling. The modeling task group was divided into three separate approaches using three preexisting CNN architectures: YOLO, ResNet, and ViT. The groups moved the preprocessed data into the preexisting CNN models and tested the performance of the validation/test datasets. Due to time constraints and efficiency measures, the ResNet and ViT models were not fully explored with all crops. The YOLO model instead was used on all four crops for testing. The performances were ascertained with accuracy, the fraction of the correct predictions to the total prediction made (a score of one being completely accurate). The modeling performances are as follows:

CNN Architecture	Maize Performance (Accuracy)	Tomato Performance (Accuracy)	Cassava Performance (Accuracy)	Bean Performance (Accuracy)
YOLO	.996	.991	.887	1.00
ResNet	.983	N/A	N/A	N/A
ViT	N/A	.995	N/A	.960

Based on performance and ease of integration, the YOLO model was used for the deployment of the app.

The crop disease detection application is an Android-based platform designed to facilitate the identification of crop diseases. Users can capture images of one of four predefined crops, with the application first classifying the crop type and subsequently diagnosing its disease state. The application is purpose-built for disease detection, featuring a user-friendly interface with clearly labeled instructions to ensure accessibility. This development highlights the potential for scaling the platform to include additional crops and disease profiles, thereby offering significant utility to Kenyan farmers and enhancing agricultural productivity.

6. Discussion

6.1 Data Collection

The data collection process played a vital role in the success of the crop disease detector because accurate modeling relies on data that reflects the true disease conditions in crops. The team compiled datasets of images of different diseases found online and incorporated them into the final dataset. However, limitations arose regarding which diseases were included. The datasets may lack completeness for diseases affecting crops, specifically in the geographical region of Kenya, or contrastingly they may include diseases that are not significant in the region. Predicting future crop pathologies and compiling the necessary data for identification remains challenging. For instance, the dataset included only two disease classes for beans, four for cassava, five for maize, and nine for tomatoes. Also, to improve the utility of the app, more crops can be included in future iterations of this application. Spending more time researching Kenyan crops and their most common diseases, as well as gathering input from Kenyan farmers and botanists, could provide deeper insights into agricultural challenges and better guide future data collections.

6.2 Exploratory Data Analysis

In the exploratory data analysis, we identified several key qualities of the data. First, we noticed that the image paths were not uniform; some ended in 'JPEG,' others in 'jpeg,' while the desired format was 'jpg.' We standardized these paths using pandas string manipulation. We checked for duplication and confirmed that none existed in the dataset. There were no missing values in the data frame also. Next, we analyzed the disease classes and visualized their distribution with a bar graph. The analysis revealed a clear data imbalance in the tomato dataset, with around 5,000 images for the yellow leaf virus compared to only about 2,000 images in the next largest class. The theme of class imbalance was present in the beans, cassava, and maize as well. This class imbalance creates challenges for modeling, as it may cause neural networks to overfit and fail to generalize. However, by recognizing this issue, we can implement corrective measures like data augmentation and class weight adjustments to mitigate it.

A class imbalance exists in the disease states, which may cause the model to overfit to the more prevalent classes, resulting in poor model generalization. This issue could lead to real-world problems if farmers using the app misclassify their crops' diseases and apply incorrect treatments. To mitigate the overfitting risk, researchers can adjust class weights and apply data augmentation. Adjusting class weights emphasizes the significance of images in smaller classes, theoretically increasing their likelihood of accurate recognition in the final model. Data augmentation expands the smaller class sizes by making slight edits, such as rotations and flips, to create new images that retain key disease and crop features while offering a fresh perspective. Although these methods address the class imbalance, real-life acquisition of the less represented classes would be most beneficial for strengthening the model's robustness in detection.

6.3 Data Preprocessing

During the data preprocessing step, we faced several limitations that impacted our workflow. Initially, we chose Dagshub for its data science workflow features, but its slow performance and inability to efficiently access both images and metadata simultaneously hindered productivity and model preparation. Additionally, Dagshub's status as a newer platform presented a steep learning curve, compounded by the lack of comprehensive tutorials and documentation, which diverted significant time and resources. We also explored Kaggle for its popularity in data science projects, but storage limitations on the platform proved unsuitable for handling the large dataset required for preprocessing and model training. These challenges forced us to reconsider our platform choices and strategies.

To address the challenges encountered with other platforms, we transitioned our preprocessed data to GitHub, a decision motivated by several advantages. GitHub offered stable and seamless data accessibility, ensuring that all team members could quickly and reliably access the required files, an essential feature for managing a high-volume dataset under strict deadlines. Additionally, GitHub's large and active community provided abundant resources and support, helping us resolve issues encountered during the preprocessing phase. Its robust version control system also enhanced collaboration by allowing us to track changes efficiently and maintain an organized workflow.

6.4 Model Development

One of the primary limitations to model development was access to data. Due to storage constraints, we could only upload the metadata CSV to GitHub rather than the images themselves, which introduced some complexity to the process as far as image organization and labelling. Additionally, data quality significantly influenced the accuracy of our models. Although we achieved very high accuracy scores during testing, concerns remain about the model's performance in real-world applications due to potential data quality issues. Chiefly, a persistent challenge was class imbalance, which affected the reliability of the model and required careful handling during the development process.

Future efforts should prioritize obtaining additional data to enhance model training. Collecting more images of underrepresented crops would improve the robustness of the models, while increasing the diversity of diseases in the dataset would provide greater utility to the farming community. Furthermore, extending the application to include crops such as sorghum, rice, and wheat would significantly increase its overall value and impact.

The current development utilized only three pre established architectures, presenting an opportunity to explore a broader range of alternatives. Architectures such as AlexNet, VGGNet, or MobileNet could offer promising avenues for modeling and prediction using the existing data. These approaches may lead to more generalizable models that are less susceptible to overfitting. Additionally, exploring new contexts or platforms for modeling could further expand opportunities for success.

6.5 Model Deployment

There is significant potential for further development within this application. Built using the Kivy framework, it provides a robust platform for developing Android applications and was utilized to upload pre-trained models for maize, beans, tomatoes, and cassava. However, due to limited financial resources, the application was not extended to iOS. Developing an iOS version in the future would enhance its accessibility and reach.

The application has a unique opportunity to expand its training data by leveraging submissions from users. By enabling users to upload images to a cloud storage system, this data could be used to improve existing models and contribute to academic research in agricultural technology.

Improving the application's aesthetics could make it more user-friendly and appealing. Branding the app specifically for Kenyan farmers, adding multilingual capabilities, incorporating a colorful layout, and using visually pleasing fonts would enhance the user experience, encouraging broader adoption among its target audience.

Adding more resources and features would further boost the application's utility and popularity. For example, including a glossary with detailed descriptions of various crop diseases could provide farmers with valuable information. This feature could explain the disease's characteristics, causes, and potential treatments, offering actionable advice to support better farming practices.

7. Conclusion

Smallholder farms play a crucial role in Kenya's economy and food supply, but crop diseases threaten their productivity by reducing crop yields. Early detection of these diseases enables timely intervention to mitigate losses. Leveraging the power of artificial intelligence, this Omdena chapter developed an application using a convolutional neural network architecture called YOLO to predict diseases in beans, cassava, maize, and tomatoes. While challenges remain with the accuracy of the models implemented in the application, significant opportunities exist for expansion to maximize its utility for farmers. By incorporating artificial intelligence, smallholder farmers can enhance their ability to prevent crop diseases and safeguard their livelihoods.

8. References

Literature Resources

1. https://www.omdena.com/chapter-challenges/ai-powered-early-detection-of-crop-diseases-in-kenyan-smallholder-farms
2. https://link.springer.com/article/10.1007/s12571-012-0203-2 (abstract about enhancing food security in Kenya)
3. https://www.kalro.org/divisions/crops/ (Kenya Livestock and Agricultural Research Organization page)
4. https://www.greenlife.co.ke/focus/diseases/ (Common diseases in crops in Africa)
5. https://journals.mmupress.com/index.php/jiwe/article/view/807
6. https://www.researchsquare.com/article/rs-20784/v2 (pesticides used in Kenya for farming)
7. https://scholarmedia.africa/agribusiness/ai-in-agriculture-how-kenyan-farmers-benefit-from-plantvillage-nuru-app/ (report on an app, PlantVillage Nuru, similar to the goal application of this project)
8. https://www.sciencedirect.com/science/article/pii/S1110982324000577 (using spectral patterns found from key biomarkers in maize to detect disease)
9. https://journals.unesco.go.ke/index.php/jknatcom/article/view/46 (deep learning application with detection of pest and bacterial diseases in maize and tomatoes)
10. https://ieeexplore.ieee.org/document/10192722 (paper on tomato leaf disease detection)
11. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0267650 (Maize leaf disease detection using WG-MARNet)
12. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0257203 (Maize and common beans' response to intercropping practices)
13. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0257008 (using spectral readings to indicate disease in soybeans)
14. https://www.frontiersin.org/journals/plant-science/articles/10.3389/fpls.2023.1079366/full (Creation of an app used to detect disease within Maize)
15. https://link.springer.com/article/10.1007/s11042-023-17459-3 (Deep learning attention model used for cassava disease detection)
16. https://dl.acm.org/doi/10.1016/j.compag.2021.106279 (Using CGAN to detect diseases within tomatoes)
17. https://medium.com/@navarai/unveiling-the-diversity-a-comprehensive-guide-to-types-of-cnn-architectures-9d70da0b4521 (Article on CNN architectures)

Packages Used

Pandas, NumPy, Matplotlib, PyTorch, Scikit-learn, TensorFlow, Kivy
Github Notebooks

Exploratory Data Analysis

- EDA and Preprocessing for Tomatoes
  https://drive.google.com/file/d/1hrLvMKeDHGSxQ_RXHtQwlHGbrCQP6cds/view?usp=sharing

Preprocessing

- Tomato Preprocessing
  (https://github.com/OmdenaAI/KenyaChapter_EarlyDetectionofCropDiseases/blob/improve-react-app/model_development/YOLO/data_preprocessing_tomato.ipynb)

Model Development

- ResNet18 pre-trained model used on Maize
  (https://www.kaggle.com/code/oliveredwardbrown/maize-cnn)
- ViT pre-trained model used on Tomato
  (https://github.com/OmdenaAI/KenyaChapter_EarlyDetectionofCropDiseases/blob/main/VITmodel.ipynb)
- ViT pre-trained model used on Beans
  (https://drive.google.com/file/d/1aPftn7RZU3xvekIQcKpl2R0MB5xevsDv/view?usp=sharing)
- YOLO modeling for tomatoes
  (https://github.com/WENDGOUNDI/crop_diseases_detection/tree/main/togithub/yolov8n_tomato)
- YOLO modeling notebook used in the android application
  (https://github.com/OmdenaAI/KenyaChapter_EarlyDetectionofCropDiseases/blob/main/notebooks/YOLO-model.ipynb)
- YOLO testing results on Maize dataset:
  (https://github.com/WENDGOUNDI/crop_diseases_detection/blob/main/togithub/yolov8n_maize/training_performance.png)
- YOLO testing results on Tomato dataset:
  (https://github.com/WENDGOUNDI/crop_diseases_detection/blob/main/togithub/yolov8n_tomato/model_test.ipynb)
- YOLO testing results on Beans dataset:
  https://github.com/WENDGOUNDI/crop_diseases_detection/blob/main/togithub/yolov8_beans/val_end_training_performance.png

- YOLO testing results on Cassava dataset: https://github.com/WENDGOUNDI/crop_diseases_detection/blob/main/togithub/yolov8n_cassava/test_end_training_performance.png

Model Deployment:

- Proof of Concept Streamlit Application: (https://github.com/WENDGOUNDI/crop_diseases_detection/blob/main/app_crop_deploy.py)
- Proof of Concept Kivy Application: (https://github.com/WENDGOUNDI/crop_diseases_detection/blob/main/get_started_kivy_app.py)
- Final App Link: (https://github.com/OmdenaAI/KenyaChapter_EarlyDetectionofCropDiseases/tree/improve-react-app/app/kivy)