2021

A Computer vision-based model for crop yield prediction using remote sensing data.

Kiragu, Daniel Mburu
School of Computing and Engineering Science
Strathmore University

A Computer Vision-based Model for Crop Yield Prediction using Remote Sensing Data

By

Daniel Mburu Kiragu

122596

A Thesis Submitted to the School of Computing and Engineering Sciences in Partial Fulfilment for the Requirement of the Degree of Master of Science in Information Technology of Strathmore University

School of Computing and Engineering Sciences

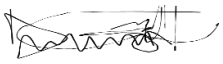
Strathmore University

December 2021

Declaration

I declare that this work has not been previously submitted and approved for the award of a degree by this or any other University. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

Kiragu Daniel Mburu



8th October 2021

Approval

The thesis of Kiragu Daniel Mburu was reviewed and approved by the following:

Dr. Julius Butime,

Dean, School of Computing and Engineering Sciences,

Strathmore University.

Dr. Bernard Shibwabo,

Director of Graduate Studies,

Strathmore University.

Abstract

Arguably, crop yield data forms the most important measure of crop productivity in agriculture. With adequate crop yield data, local and international bodies can develop effective agricultural policy leading up to sustainable food supplies and elevated food security. However, timely acquisition of crop yield data can be a cumbersome task as existing crop yield prediction approaches face numerous challenges. In this study, these challenges are identified as high cost and high dimensionality of data required for the prediction activities as well as limited scaling of the resultant prediction models. In efforts of overcoming these challenges, this study leveraged an alternative source of data to design and develop a cheap, accurate and scalable deep learning model using convolutional neural networks. Satellite imagery datasets were used as the primary and only source of data for training the model. This benefited the study in two major ways. Firstly off, the approach automatically took care of the high dimensionality problem as demonstrated in the GEMS data. Second, satellite imagery data is readily available globally, a factor that greatly reduced the costs needed to collect real-time data for the study. Validation of the developed model was done using 10% of the overall dataset acquired. Reliability of the model in performing crop yield predictions was captured using an MSE loss function for each epoch trained. Cumulatively, the model achieved an MSE loss score of 3.6.

Keywords:

Deep Learning, Convolutional Neural Networks, Crop Yield Prediction, Agriculture.

Dedication

This research work is dedicated to my parents and siblings whose unyielding love, support, and encouragement have enriched my soul and inspired me to pursue and complete the master's program.

Acknowledgements

First and foremost, praises and thanks to the Almighty God for His mercies, guidance, knowledge, and the wisdom He asserted in me throughout my research work.

I would like to express my deep and sincere gratitude to my research supervisor Dr. Bernard Shibwabo for his relentless efforts, encouragement, and support to see that this research was completed successfully. Special thanks to Dr. Vincent Omwenga for his assistance during the research. His dynamism, vision, sincerity, and motivation have deeply inspired me. Sincere thanks to my family for the prayers, support, and encouragement accorded to me.

Finally, my thanks go to all the people who have supported me to complete the research work directly or indirectly.

Table of Contents

List of Figures

List of Tables

List of Equations

Abbreviations and Acronyms

AI: Artificial Intelligence

ANN Artificial Neural Network

CNN: Convolutional Neural Network

EVI: Enhanced Vegetation Index

GEMS: Genetics, Environment, Management, and Socioeconomic factors

MA: Management Areas

ML: Machine Learning

NDVI: Normalized Difference Vegetation Index

SAVI: Soil Adjusted Vegetation Index

UMAP: Uniform Manifold Approximation and Projection

Definition of Terms

Crop Production A common agricultural practice adopted by farmers globally in the growth and production of crops to be used as fibre or food. It entails all feeds sources such as soil preparation, seeds sawing, irrigation, harvesting, storage and produce preservation among others that are required to maintain and produce crops (Committee on Science Breakthroughs 2030: A Strategy for Food and Agricultural Research et al., 2019).

Deep Learning: A group of machine learning methods that use artificial neural networks to perform representation learning and are capable of learning data that is unstructured or unsupervised without supervision (Schmidhuber, 2015).

Neural Network: This consists of thousands or even millions of simple processing nodes that are densely interconnected. They are a means of machine learning in which a computer learns to perform a task by analyzing specific examples (Schmidhuber, 2015).

Spectra Data: Includes data collected via specific wavelengths of radiation and is related to the phenomenon being observed.

Yield: The mass of harvested crop product in a specific area and is influenced by several factors (Ngoune Liliane & Shelton Charles, 2020).

Chapter 1: Introduction

1.1 Background of the Study

Crop yield prediction is crucial in the fight against food insecurity as it guides farmers to make informed choices to boost crop produce (Khaki & Wang, 2019). Crop yield prediction models help farmers in deciding which crop varieties to plant as well as how to manage them for maximum produce. They also help farmers to determine the amount of resources required from planting to the time of harvesting (Dodds & Bartram, 2016). Government agencies use crop yield forecasting to guide export and import decision making of foods to promote food security (United Nations National Assembly, 2015).

According to Khaki and Wang (2019), crop yield prediction is a cumbersome process due to the high dimensionality of data that must be taken into consideration. For any crop yield prediction task to succeed, numerous features that directly or indirectly affect crop yielding capability must be observed and accounted for. You et al. (2017) state that some of the aspects that impact crop yield include genotype, rainfall, winds, temperature, soil moisture content, soil type, depth, and available minerals and nutrients. Farm management practices also have a direct impact on crop yield.

In its 2019 publication on Science Breakthroughs to Advance Food and Agricultural Research by 2030, The National Academies of Sciences, Engineering, and Medicine (2019) acknowledge that data dimensionality and heterogeneity pose great challenges to the efforts aimed at applying modern technologies to improve efficiency in food and agricultural production. The Academies extends and further summarizes the factors affecting crop yield and crop yield prediction by the acronym GEMS. GEMS is a modelled function that asserts that crop production is a result of complex interactions between genetic (G), environment (E), management (M), and socioeconomic (S) factors (National Academies of Sciences, Engineering, and Medicine, 2019). Further discussions on factors influencing crop yielding and crop yield prediction are made in sections 2.3 and 2.4 of the study respectively.

With the current advancements in computer technology, the scope of crop yield prediction has shifted, making it possible to factor in the wide array of aspects affecting crop performance. Further, the new era of information technology has made it possible for farmers to collect data on all these elements and store it safely for reference in the future.

Remote sensing technology has also helped in creating alternative sources of data that can be applied in prediction of crop yield on a global scale. Data analysts can obtain Nominal Difference Vegetation Index (NDVI), Enhanced Vegetation Index (EVI) and Soil Adjusted Vegetation Index (SAVI) among other information from spectra images obtained from satellites orbiting the earth (Franz et al., 2020). Other data points that can be obtained from spectra data and that are useful to crop yield prediction include land slope, vegetation cover, evapotranspiration, climate water deficit, runoff, soil moisture, downward surface shortwave radiation, temperature, windspeed, and drought severity index among others. A time-series of remotely sensed data can be used to track seasonal changes in climate and vegetation cover throughout the year, enabling real-time crop yield prediction (Franz et al., 2020).

Machine learning algorithms can be trained using remote sensing data and reinforced with existing crop-cut and locally generated farm records to predict crop yield (Zindi, 2020). In this process, data analysts input available data about crop yield factors and crop performance into the machine learning models for training. The model is then tested with data used in training to determine if it will generate accurate yield predictions (Anitescu et al., 2019).

Computer vision-based models are machine learning algorithms capable of performing both supervised and unsupervised learning from visuals like images and videos to produce inference, and hence predictions. Since remotely sensed data is readily available globally, training computer vision-based models with remotely sensed data can help in creating cheap and scalable alternatives for crop yield prediction (You et al., 2017).

1.2 Problem Statement

Despite their utility, leading techniques in crop yield prediction are associated with high costs, tedious to curate and are extremely difficult to scale. These techniques utilize crop-cut and locally sensed data, such as soil, temperature, humidity and rainfall measurements and farmer surveys' data acquired from physical field visits. Both crop-cut and locally sensed data provide detailed information leading up to yield estimation, but are costly to collect, highly prone to noise, and are difficult to scale. Further, few crop-cut yield estimate datasets exist, and even fewer are regularly sampled every season.

Alternatively, remotely sensed data, such as satellite imagery, is cheap to collect and widely accessible. Coupled with modern machine learning technologies, the data could potentially provide a cheap and equally effective alternative (You et al., 2018; Wang et al., 2018). Indeed, the global Committee on Science Breakthroughs 2030 emphasize on the need to utilize modern technologies including artificial intelligence to transform the food and agriculture system (Committee on Science Breakthroughs 2030: A Strategy for Food and Agricultural Research et al., 2019).

The outcome of this research can provide an alternative, inexpensive, accurate and scalable technique for crop yield prediction. Accurate prediction of crop yields is arguably the most effective measure of agricultural production and productivity. As such, the research would greatly promote formulation of effective agricultural policies, improve food security and guide the development and implementation of sustainable agricultural practices for both developed and developing nations.

1.3 1.3 Objectives

1.3.1 General Objective

The aim of this study is to develop a computer vision-based model for predicting yielding capacity of crops using remote sensed data. This is expected to promote discovery, production, and sustainability in agriculture through precision farming.

1.3.2 Specific Objectives

i. To investigate factors influencing crop yielding in Agriculture.

ii. To examine existing techniques and approaches that have been used to predict crop yielding in Agriculture.

iii. To design and develop a computer vision-based model for predicting crop yield in Agriculture using remotely sensed data.

iv. To test the reliability of the algorithm in predicting crop yield in Agriculture.

1.4 Research Questions

The central question of this research is, how can computer vision models be used to predict crop yielding in agriculture?

i. How have factors influenced crop yielding in Agriculture?

ii. How have techniques and approaches been used to predict crop yielding in Agriculture?

iii. How can a computer vision-based model for predicting crop yield in Agriculture using remotely sensed data be developed?

iv. How can the reliability of the developed model be tested in predicting crop yield in Agriculture?

1.5 Justification

In this study, the developed model uses remotely sensed data to forecast crop yields. Using remotely sensed data, farmers will be able to obtain real-time crop forecasts and make timely decisions about what to grow and what management actions to take to ensure maximum yield of their crops. The developed model can be applied at scale and can perform independent of locally sensed data. This makes it highly suitable for adoption and application in developing countries where locally sensed datasets are sparse.

The developed model aims at helping policy makers promote national and global food security by determining, in advance, how the planted crops will perform and prepare to either import or export foods. Similarly, policy makers can pass informed decisions regarding the endorsement or ban of certain crop genomes depending on their projected performance. The model also helps seed companies to assess the performance of different seed hybrids in different places. This allows them to determine which crop varieties perform best in different locations.

The study contributes to the academic field by attempting to identify an optimally performing model for crop yield prediction using deep learning methods. It contributes to the general body of knowledge that can be used by other academic professionals and researchers in their work.

1.6 Scope and Limitation

The focus of this research is designing and developing a model that can be used to perform real-time crop yield prediction at a scalable and global level. The model uses Convolutional Neural Networks, a deep learning technique, to learn from a time-series of satellite imagery and draw inferences on crop yielding capabilities based on the discovered spectra data. The resultant trained model can, thereafter, be used to perform crop yield prediction in real-time.

The developed model is trained using data acquired from crop-cut maize yield estimates from fields in East Africa but can be applied at scale to other regions and to other crop domains. The research is conducted within the technological space of agriculture. The research can be used as a point of reference for various consumers and academic practitioners whose intention is to optimize crop production efficiency and reduce environmental impacts of farming practices through Precision Agriculture.

Chapter 2: Literature Review

2.1 Introduction

This chapter evaluates the theoretical frameworks that are fundamental in the realization of the importance of this study. The frameworks dictate the key principles that govern how the study was executed and the benefits gained from designing and developing the model.

The chapter also reviews and performs a critical analysis on related literature to create a clear understanding of the research problem and find optimal methods for developing the model. A general overview of computer vision-based prediction systems is investigated and explained here. Algorithms of interest for this research and their application in prediction of crop yielding are also described below with their objectives, methods, and weaknesses and how they relate to the objectives of this research.

2.2 Theoretical Framework

2.2.1 Information Theory

First formalized by Claude Shannon in 1948 (cited in Stone, 2015), the information theory examines the transmission, processing, extraction and use of information. Theoretically, information can be described as the resolution of uncertainty. This can be illustrated using Figure 2.1 where information is seen as a basic element in the creation of knowledge.

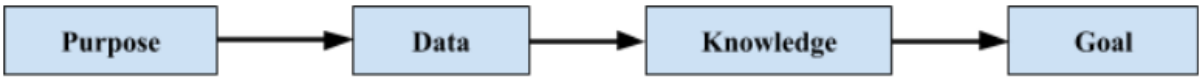


Figure 2.1 Basic Information Processing Mechanism (Adapted from Selçuk (2019))

In his book on "Information Theory", Selçuk (2019) devised a framework to depict the basic information processing mechanism. The mechanism designed demonstrated the key elements of the information theory including data, information, knowledge, and goal (Selçuk, 2019). Data represents the raw information expressed in natural languages while information represents the

specific set of rules defining how the various datasets relate and in what order. Knowledge represents the ability to understand and execute the defined rules while the goal represents the final products, the ultimate purpose for which the data was created.

Information theory carries a deep and broad mathematical theory basis, all with equally deep and broad applications (Stone, 2015). These characteristics of the theory have created its close association with a collection of both pure and applied disciplines throughout the world. Over the past half century, the applications of information theory have been reduced to engineering practice under a variety of rubrics such as machine learning, adaptive systems, cybernetics, informatics and artificial intelligence among others (Stone, 2015).

Extended applications of information theory have spearheaded scientific discoveries across all these disciplines. In his work, and as demonstrated in Figure 2.2, Selçuk (2019) also developed a model illustrating the information processing mechanism as applied in scientific discovery. Bearing its inception from just an initiative, the information processing mechanism models the transition of mere data through to new scientific discoveries. A vital characteristic of information is its relevance or value, which is highly subjective and naturally difficult to quantify.

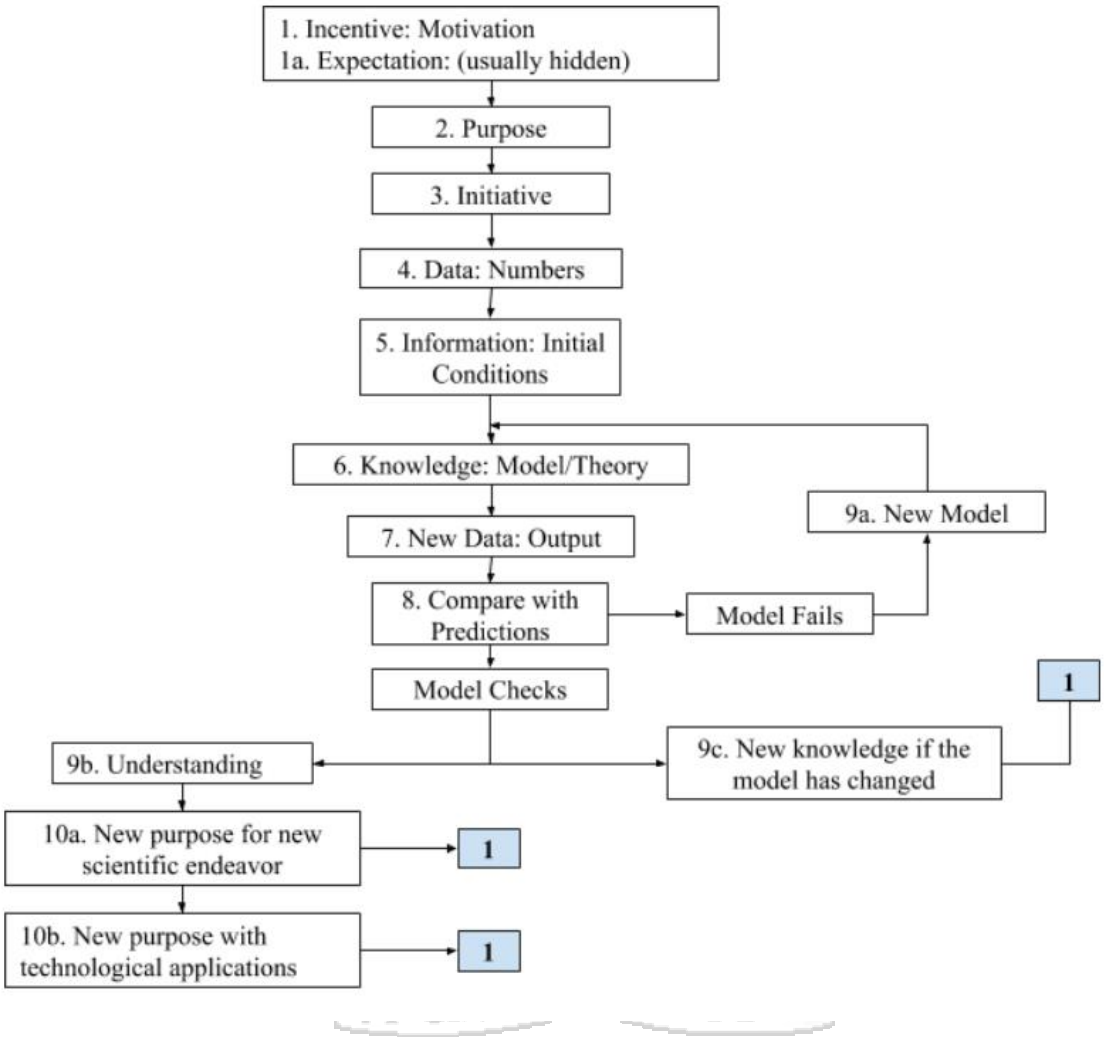


Figure 2.2: Basic Information Processing Mechanism for Scientific Discovery (Adapted from Selçuk (2019))

2.2.2 Applied Optimization Theory

Information theory clearly demonstrates that humanity has arrived at an era where information forms the basis of all advancements. The rise of big data has further complimented this fact by readily availing information for simulation and modelling of new scientific discoveries and advancements. As such, it has become progressively crucial to advance large-scale data mining

and optimization models in an effort to solve emerging data driven problems in science and engineering (Rao, 2019).

Optimization implies the application of mathematics to determine the optimal choice given varying situations. To develop a functional optimization model, all control variables must be treated as candidates of slow changing dimension. With this consideration, an objective function is created and fed with constraints where it works to identify a feasible solution that satisfies all given constraints.

Figure 2.3 illustrates the path taken when working out an optimization problem as described by Chaovalitwongse, Chou, Liang and Wang (2017). The model begins by defining an empirical problem that depicts a complex process. The process is then simplified systematically, and delimiters are defined by dictating the most relevant and sorting out the unwanted features. At this stage, a model consisting of rules, control variables and an objective function is designed and developed. Optimization models are often developed for both linear and non-linear problems in a case where non-linear problems comprise at least one non-linear function.

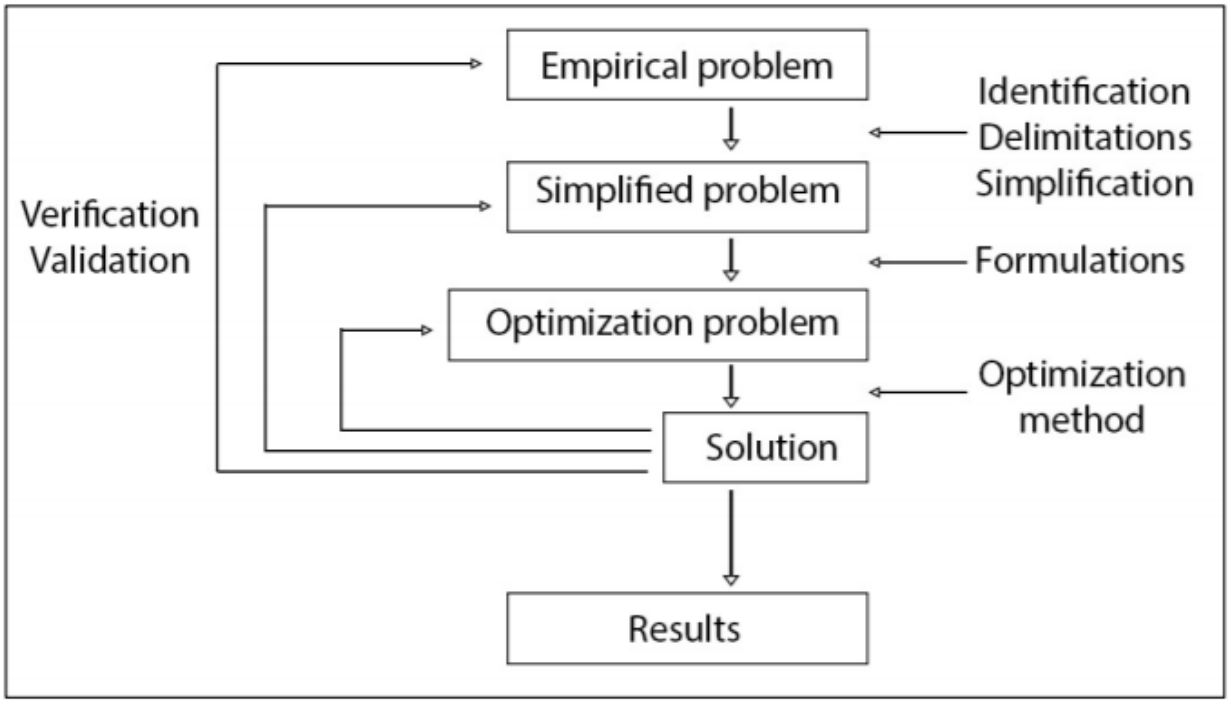


Figure 2.3: Work path to solving an optimization problem (Adapted from Chaovalitwongse et al. (2017))

2.3 Factors Influencing Crop Yielding in Agriculture

A comprehensive understanding of the ingredients leading up to food production is vital for the enhancement of national and global food security. Several research works have attributed poverty reduction with an increase in crop yields (Ngoune Liliane & Shelton Charles, 2020). Yield is defined as the mass of harvested crop product in a specific area.

Everyday crop production activities comprise complex decision making, characterized with high levels of uncertainty and multiple courses of action (Schemberger, Fontana, Johann, & Souza, 2017, p. 189). Before making a decision at every stage in crop production, a farmer needs to consider a variety of factors such as rainfall, location, soil and air humidity, soil type, dew point, temperature and other site-specific factors such as irrigation, pesticides and application of fertilizer that help optimize plant treatment.

According to the National Academies of Sciences, Engineering, and Medicine (2019) these factors can be summarized by the acronym GEMS. GEMS is a modelled function that asserts that crop production is as a result of complex interactions between genetic (G), environment (E), management (M), and socioeconomic (S) factors (Committee on Science Breakthroughs 2030: A Strategy for Food and Agricultural Research et al., 2019). Collectively, the variation between these factors account for worldwide yield differences, region by region.

2.3.1 Genetic Factors

Genetic factors consist of the DNA modifications of the particular plant which dictate their capabilities to grow and thrive within a given set of environmental parameters (Khaki & Wang 2019). For example, some crops are genetically modified to grow in drier conditions than typical plants would. The genetic structure of plants also dictate how they respond to pests and diseases. Disease and pest resistant crops grown in precision agriculture help farmers save money they would have spent on pesticides and disease management chemicals.

2.3.2 Environmental Factors

As explained by Filippi and colleagues (2017), environmental factors such as soil, altitude, and weather directly affect crop yield. Soil conditions such alkalinity and the presence of various minerals boost or inhibit crop growth and productivity. The height of the farm above sea level also affects crop growth and production. For example, tea thrives in high altitude areas while most millet varieties thrive in medium to low altitude zones. Similarly, the weather impacts crop productivity with some crops being drought-resistant while others do well in areas with substantial rainfall.

2.3.3 Management Factors

The management approach in the fields also has an effect on crop yield. Farm management processes embedded on advanced technologies lead to increased monitoring and management efficiency and increased yields (Shafi et al., 2019). These include the efficient use of fertilizers, keeping electric records, and using drones for monitoring the farm. Poor management practices

including lack of proper record-taking create opportunities for errors and prevent farmers from making the right decisions.

2.3.4 Social Economic Factors

Crop production is also affected by socioeconomic factors like farmers' education, income, market forces and access to credit. Educated farmers with substantial income are more likely to adopt precision agriculture practices to enhance crop output as compared to uneducated people with little income (Ngoune Liliane & Shelton Charles, 2020). On the other hand, the market prices of crop yields influence farmer attitudes and thus the production rate of particular crop products. Availability of credit facilities enables farmers to take loans to increase inputs and the yielding rate.

2.4 Crop Yield Prediction

Crop yield prediction is vital in the food production industry. According to Abbas et al. (2020), farmers rely on yield prediction data to manage their fields and make informed financial decisions. Seed companies modify seed genotypes to improve their productivity in different environmental conditions. Despite the importance of crop yield projection, there is an insufficient supply of reliable tools for approximating future crop yields. The main challenge in predicting crop produce is the need to take into consideration the various factors influencing yield capacity.

2.4.1 Approaches Used in Crop Yield Prediction

The farmer recall's is a common method whereby farmers predict the quantity of yield they expect to get from their farms at the end of the season based on past experiences (Franz et al., 2020). Their predictions are based on how the current crops are performing in comparison to the performances of their predecessors. According to Sapkota et al. (2016), this approach is applicable at the stage where the crops have attained full maturity. At this stage, farmers can tell how much produce they will get by merely noting the state of the fully mature crops. The quantity obtained after harvest is then compared with the pre-harvest estimate to assess the farmer's accuracy.

Crop-cuts methods can also be used in crop yield prediction. In this method, a single or many subplots are cut within the main field and the produce obtained from these subplots divided with their area to get the yield per unit area. Sapkota et al. (2016) note that the number of subplots used usually depend on the available resources in the farm as well as the degree of accuracy desired. Using several plots is especially necessary when the field has variable productivity. The smallest size of sampling plot for estimating maize yield using the crop cut method should be $1m^2$.

In the whole plot harvest technique, all produce from the entire field of a trial plot is collected and weighed. Produce from crops growing along boundary lines is excluded as it may not precisely show the conditions tested on the plot. According to Sapkota et al. (2016), the method is appropriate for experimental plots as well as for small scale farmers. This is the most accurate method and is often used to test the accuracy of other crop yield prediction methods. Farmers use the yield per unit area from these trial plots to predict the performance of their maize crops.

Other techniques involve deep learning with remotely sensed data and sampling harvest units. However, these methods can become unreliable as they do not take into consideration other factors like soil conditions which affect crop yield (Sapkota et al., 2016). Table 2.1 shows a comparison of some of methods used to predict crop yield.

Table 2.1: Comparison of common maize crop yield prediction techniques (Sapkota et al., 2016)

Comparison of common maize crop yield prediction techniques

Method	Cost effective	Scale	Accuracy in prediction
Farmer recall	Cheap and time saving	Farm to landscape	Fairly accurate although it requires strict supervision. Farmers may purposely overestimate or underestimate yield
Crop cuts	Labour and time intensive	Farm to landscape	High likelihood for overestimation
Whole plot harvest	Cost and labour intensive	Plot level, farm level, and experimental plots	High precision with the least or zero error
Sampling harvest unit	Cost effective	Farm to landscape	Errors are possible when farmers harvest once from multiple points and not practical for crops with staggering maturity rates
Remote sensing	Cost effective	Landscape	Chances of error in case different crops have the same signature

2.4.2 Impacts of Crop Yield factors on Yield Prediction

Numerous factors affect crop production, making yield prediction an extremely difficult task. Regardless, all these factors should be considered to come up with precise yield approximations. To reduce the high dimensionality of data, researchers might make certain assumptions about some of the factors. Consider maize production for example, assuming the same genotype is grown constantly, the only factors affecting the crop's yield would be mainly environmental and managerial. These include temperature, soil moisture content variations, availability of soil nutrients, and diseases and pest control among others.

It is important to note that each of these aspects affects productivity independently. In essence, there exist optimal ranges of these parameters whose effects combined generate maximum yields. Any slight changes in each of these parameters could have an adverse impact on the crops yield. For example, the temperature and soil nutrients could be optimal while the moisture is not within the optimal range. This will ultimately lead to significant decrease in crop produce.

2.5 Machine Learning Techniques used in Crop Yield Prediction

Machine learning algorithms offer the computer algorithm the ability to automatically process data and learn without explicitly being programmed (Gaster, 2012). Such algorithms include Random Forests (RF), Support Vector Machine (SVM), Artificial Neural Networks (ANN), Uniform Manifold Approximation and Projection (UMAP), Singular Value Decomposition (SVD), and convolution neural networks (CNN). This research is focused on CNN as the algorithm to be used.

Crop yield prediction through remote sensing method is achieved through the analysis of spectra data as captured in satellite images. The image processing techniques involve these key basic steps as shown in Figure 2.4 that help to arrive to the detection.

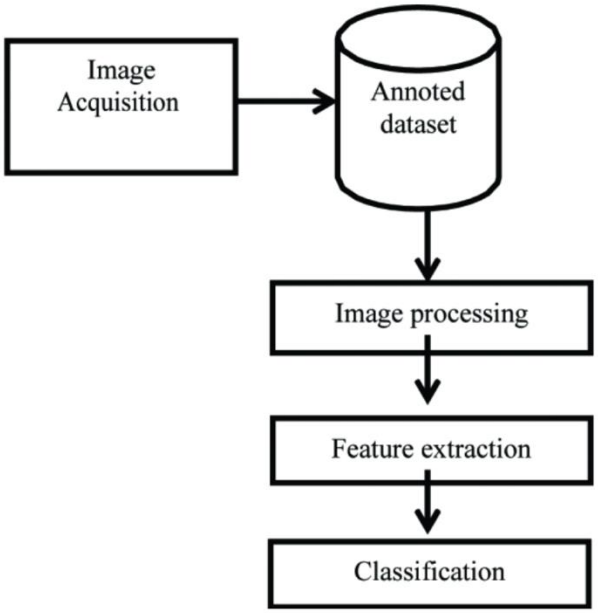


Figure 2.4: Image Classification Steps (Adapted from Shruthi (2019))

2.5.1 Random Forests

Also referred to as Decision Tree Ensembles, Random Forests represent one of the algorithms widely used for feature selection. The technique achieves feature reduction by carefully constructing a large set of trees against a target attribute. The usage statistics for each attribute are then used to find the most relevant subset of attributes (Silipo et al., 2015).

For instance, a large set of up to two thousand very shallow trees (two levels) can be generated. Each tree is then trained on a small fraction of the total number of attributes. RF then selects and retains the most informative attributes based on how often each attribute is selected as the best split in each tree. RF calculates and stores the attribute usage statistics which is later used to determine the most predictive attributes.

2.5.1.1 Algorithm

RF borrows the popular method of bootstrap aggregation or commonly referred to as bagging to create the training algorithm for the tree learners (Athey et al., 2019; cited in Cai et al., 2018). Given a training set of X = x1, ..., xn with responses Y = y1, ..., yn, bagging repeatedly (B times) selects a random sample with replacement of the training set and fits trees to these samples:

For $b = 1, ..., B$:

I. Sample, with replacement, n training examples from X, Y; call these Xb, Yb,

II. Train a classification or regression tree fb on Xb, Yb.

After training, predictions for unseen samples x' can be made by averaging the predictions from all the individual regression trees on x' as shown in Equation 2.1. When using classification trees, RF performs predictions by picking out the features with highest weights (James et al., 2013).

Equation 2.1: Prediction algorithm as used in Random Forest (Adapted from James et al. (2013))

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b(x')$$

To provide a degree of impurity or out-of-bag error of the predictions made by the RF algorithm, the estimate of uncertainty is calculated as the standard deviation of all predictions made from the individual regression trees on x' (James et al., 2013). Equation 2.2 provides the formula used:

Equation 2.2: Formulae to calculate out-of-bag error in Random Forest (Adapted from James et al. (2013))

$$\sigma = \sqrt{\frac{\sum_{b=1}^{B}(f_b(x') - \hat{f})^2}{B-1}}.$$

2.5.2 Principal Component Analysis

Analytics Vidhya (2016) simplifies the definition of Principal Component Analysis (PCA) to entail a technique for obtaining the relevant set of variables, denoted as components, from a large group of features as presented in a data set. PCA achieves this by projecting through the high dimensional data with an aim of collecting all information for all dimensions and pointing out the irrelevant dimensions. This allows the filtering of the feature set of the given high dimensional data while achieving minimal loss of information, making visualization obtained more significant.

2.5.2.1 Algorithm

In practice, PCA orthogonally mutates the initial n coordinates of a data set into a new set of n coordinates referred to as principal components (Sharma, 2018). The results of the transformation force the first principal component to obtain the largest possible variance. Similarly, each of the subsequent components obtains the highest possible variance but are constrained in that they must be uncorrelated with the preceding components. To obtain the highest information gain, defined by the variation in the data, PCA strives to retain only the first $m < n$ components to reduce

dimensionality in the data supplied (Sharma, 2018). PCA is more useful when dealing with 3 or higher dimensional data.

2.5.2.2 Limitations

Since the data set used requires normalization prior to performing PCA analysis, the PCA transformations become highly sensitive to relative scaling of the original feature set. Similarly, the new coordinates generated after a PCA transformation is done no longer exist as real system generated variables. This means that the final data obtained after performing a PCA analysis has lower interpretability. As a result, and as Sharma (Sharma, 2018) advises, PCA would not be a great analysis technique if interpretability of the results is key for a given analysis data project.

2.5.3 Singular Value Decomposition

In the space of linear algebra, Singular Value Decomposition (SVD) refers to a factorization of either real or complex matrices. SVD generalizes the decomposition of a normal square matrix to any $m * n$ matrix via an extension of the polar decomposition.

2.5.4 Uniform Manifold Approximation and Projection

Uniform Manifold Approximation and Projection (UMAP) is a dimensionality reduction algorithm proposed by McInnes and Healy in 2018. UMAP is a novel manifold method suitable for both visualization and dimensionality reduction of big data. The method is founded from the basis of algebraic topology and Riemannian geometry theoretical frameworks, resulting in a practically scalable algorithm with advent real world applications.

Three assumptions about the general characteristics of data are made while constructing the UMAP algorithm. These include the data is uniformly distributed on a Riemannian manifold, the Riemannian metric is locally constant or can be approximated as such and the manifold is locally connected. These assumptions make it possible to model the manifold with a fuzzy topological structure. The embedding is found by searching for a low dimensional projection of the data that has the closest possible equivalent fuzzy topological structure (McInnes et al., 2018).

2.5.4.1 Strengths

As McInnes and Healy (2018) explains, UMAP is a highly competitive algorithm which is applicable for data visualization and dimensionality reduction needs. These characteristics form the basis of the numerous benefits brought about by the algorithm.

While the t- Distributed Stochastic Neighbor Embedding (t-SNE) algorithm is currently seen as the best technique for data visualization capabilities, UMAP emerges as a highly competitive technique and presents even higher quality in data visualization. Additionally, the technique is seen to preserve more of the global structure of the original data, unlike PCA and as well demonstrates a superior run time performance. UMAP has no computational restrictions on embedding dimension, making it viable as a general-purpose dimension reduction technique for machine learning.

2.5.5 Convolutional Neural Networks

In this study, deep learning plays a major role in achieving the objectives of the research. Lindsay (2020) defines deep learning as a mathematical framework which emphasizes the understanding of successive layers of data representations in terms of how many layers contribute to a model as shown in Figure 2.5. These layers are learned through artificial neural networks.

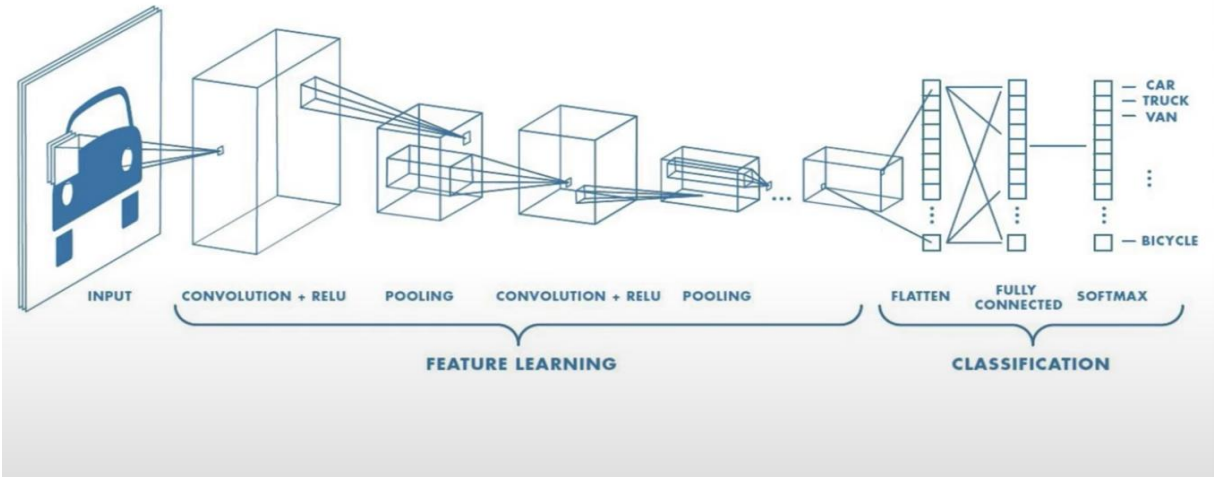


Figure 2.5: Image Showing Convolution Process (Adapted from Lindsay (2018))

The information goes through layered filters and is outputted as a purified sample. The algorithm transforms the data image into representations different from the original image as shown in Figure 2.6.

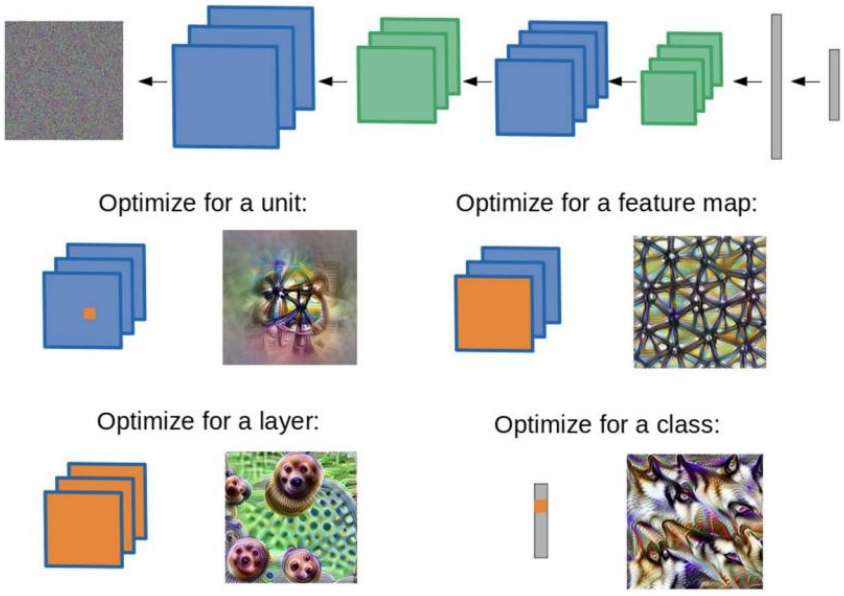


Figure 2.6: A Deep Learning Representation by a CNN Model (Adapted from Lindsay (2018))

The Convolution Neural Network (CovNet) is widely used for implementing the deep learning technique. It comprises of layers for feature extraction and classification. The advantage of CNN is its ability to identify the image features without supervision from the human which is good for timely detection of plant disease by non-experts (krishna et al., 2018).

2.6 Related Works

Different models have been proposed in predicting crop yielding in agriculture. The following highlights a few of the models that have been developed using machine learning algorithms.

2.6.1 Crop Yield Prediction Through Deep Learning

Kaneko and colleagues (2019) came up with a model for projecting crop produce in parts of the world where there lacked sufficient data for analysis. Most farmers in the developing countries have limited education on agricultural best practices and hence do not keep records of their farms. Kaneko et al. (2019) proposed that researchers could use information collected from developed countries to predict crop yield in places where data is sparse.

Initially, the researchers used a deep learning model on locally available remote sensing data from five African countries namely Kenya, Ethiopia, Tanzania, Malawi, Zambia and Nigeria to predict maize yields. The results were desirable in Kenya, Tanzania and Zambia with R^2 ranging from 0.50 to 0.56, and undesirable in Ethiopia, Malawi and Nigeria with R^2 ranging from - 0.60 to 0.13. On transferring data from other countries, Kaneko et al. (2019) obtained more accurate predictions in all the six countries as depicted in Table 2.2.

Table 2.2: R^2 values for LSTM-based crop yield prediction model and those when the model is combined with Gaussian Process (Adapted from Kaneko et al. (2019))

Country	LSTM	LSTM+GP
Kenya	0.49	0.56
Tanzania	0.40	0.50
Zambia	0.39	0.56
Ethiopia	-0.35	0.13
Malawi	-0.29	-0.09
Nigeria	-0.68	-0.60

Results after transferring data from other countries improved as shown in Table 2.3:

Table 2.3: R^2 values for LSTM-based crop yield prediction model and those when the model is combined with Gaussian Process (Adapted from Kaneko et al. (2019))

Country	LSTM	LSTM+GP
Kenya	0.82	0.82
Tanzania	0.74	0.80
Ethiopia	0.74	0.82
Malawi	0.25	0.55
Zambia	0.67	0.77
Nigeria	0.40	0.53
Average	**0.60**	**0.715**

2.6.2 Deep Neural Networks for Predicting Crop Yield

Khaki and Wang (2019) generated a model for predicting crop yield based on deep neural networks. The research involved analyzing large data samples provided by Sygenta which showed the genotypes and outputs of over 2000 maize hybrids grown in over 2000 locations.

For this research objective, the authors designed and developed a deep learning model that combined the linear and non-linear relationships between crop yield factors. Factors taken into consideration include the genotype and environmental factors like weather and soil. The model performed well with relatively accuracy scores and a differential root mean square error (RMSE) of 0.12. Table 2.4 illustrates these results.

Table 2.4: RMSE and R^2 values for the DNN crop yield prediction model (Adapted from Khaki & Wang, (2019))

Test variable	Validation RMSE	Validation correlation coefficient (%)
Yield	12.79	81. 91

Check yield	11.38	85.46
Average	**12.085**	**83.685**

2.6.3 Predicting Crop Yield based on Pre- and Mid-Season Data from Gamma Radio-Metrics Surveys or Electromagnetic Induction Surveys

This model, designed and developed by Filippi et al. (2017) uses data from electromagnetic or gamma radio-metrics surveys and remote sensing data like MODIS NDVI to predict crop yields. The researchers obtained crop yield data from farmers in Western Australia. Remotely sensed weather and vegetation data was collected from the national archives. Filippi and colleagues (2017) observed that the model produced better results as time passed by, which can be credited to the accumulation of data over time. For deep learning models, large data sets are required to increase their predictive ability.

2.6.4 Crop Yield Prediction through Machine Learning Model

Sangeeta (2020) proposes a machine learning model that takes into consideration environmental factors like rainfall, temperate, winds, and soil conditions to estimate crop yield. The researcher used data of the previous performances of crops under different environmental conditions to train the model. In determining the accuracy of the model, Sangeeta confirmed that the model generates more accurate results when the training data is more than testing data. The author also confirmed that the model would be irrelevant and perform poorly for regions with sparse datasets.

Abbas and colleagues (2020) created a model which combined proximal sensing and machine learning processes to predict crop yield. Proximal sensing techniques provide data about soil properties, land slope and Normalized Difference Vegetation Index (NDVI). The machine learning models, which were trained using data collected over three years, generated poor yield estimates. Figure 2.7 illustrates yield estimates achieved from this study. Despite this shortcoming, Abbas et al. (2020) argued that with larger datasets the models could generate better results.

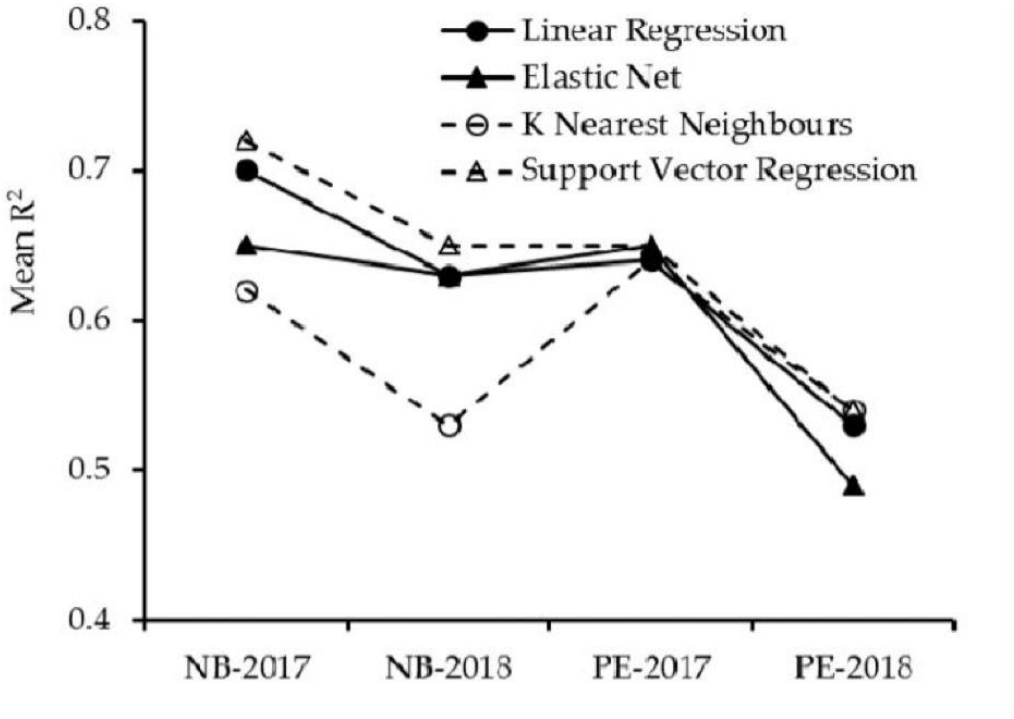


Figure 2.7: Comparative analysis of machine learning algorithms in predicting crop yield (Adapted from Abbas et al. (2020))

2.6.5 Crop Yield Prediction through Deep Gaussian Process

You et al. (2018) suggest that errors in deep learning models on remote sensed data can be minimized by introducing a Gaussian function. As demonstrated in Table 2.5, the final results show increased accuracy when convoluted neural networks are combined with the Gaussian process leading to reduced RMSE on crop yield predictions. This model can be used to provide real-time yield predictions using real-time remote sensing information. The only challenge with this model is its complexity preventing farmers without basic programming knowledge from using them.

Table 2.5: Comparison between RMSE of crop yield predictions obtained using CNN and CNN+GP models (Adapted from You et al. (2017))

Year	CNN	CNN+GP
2011	5.77	5.7
2012	5.91	5.68
2013	5.50	5.83
2014	5.27	4.89
2015	6.40	5.67
Average	**5.77**	**5.55**

2.7 Research Gap

Crop yield prediction remains a challenge with most of the existing models facing cost and performance challenges. Further, the inability to apply existing models at scale limits their applicability. These models use locally sensed data which is costly to acquire, noisy and limited to specific geographical regions. Of all the models reviewed, only two attempts to seek alternative and scalable approaches by use of remote sensed data to achieve crop yield prediction. This research proposed to combine the merits of existing techniques to come up with an inexpensive, scalable model that utilizes readily available satellite imagery data to perform crop yield predictions. This was achieved by harnessing the power of deep learning.

2.8 Conceptual Framework

Based on the literature reviewed and the various gaps identified, the study applied the conceptual framework as depicted in Figure 2.8 to combine techniques sited and come up with a model that could achieve the objectives of the research. The framework was confirmed to offer a workable

solution to the problem stated. The model developed through this framework achieved high levels of accuracy and confirmed its scalable applications to varying geographical regions and crop domains. Independent and dependent variables of the study were identified as time-series satellite imagery and crop yield prediction score respectively.

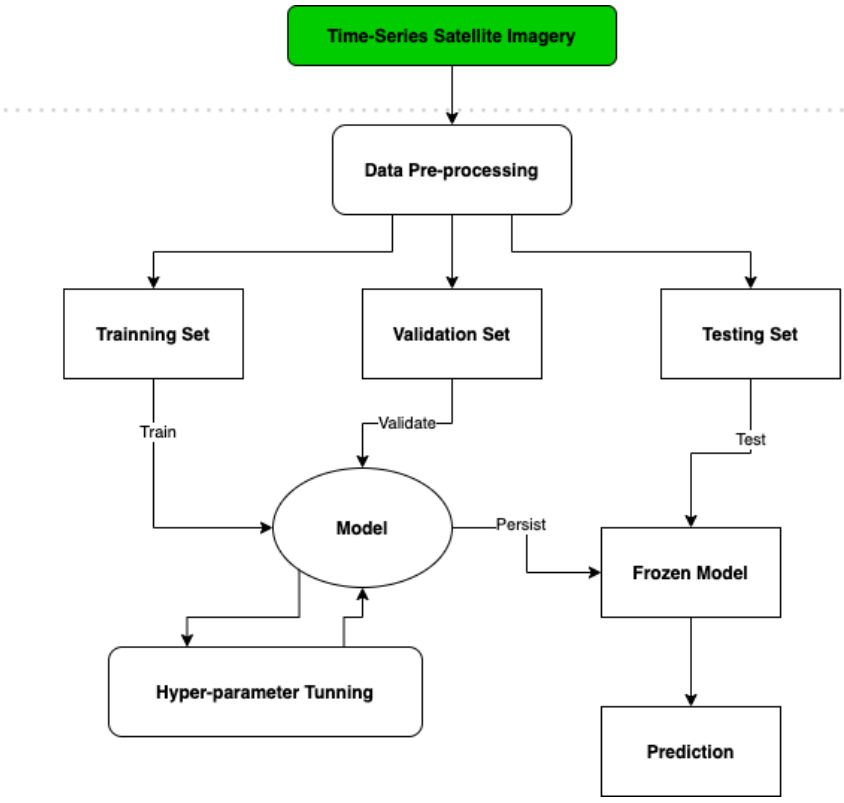


Figure 2.8: Conceptual Framework

Chapter 3: Methodology

3.1 Introduction

This chapter discusses system analysis approaches, architectures, design, development, implementation, and testing. Further, the chapter discusses the steps used in developing the model including data acquisition, data preprocessing, development, and validation on the model. The chapter concludes by evaluating the ethical considerations, reliability, and validity of the research.

3.2 Research Design

The study aimed at developing a model capable of predicting yielding capacity of a particular crop, given a time-series spectra data of the crop field. This involves the analysis of satellite imagery to extract quantifiable data to which statistical and computational techniques could be applied. Classifications are assigned to these images and are later used to draw inference while performing actual prediction tasks.

These characteristics of the study make it quantitative in nature (Kothari, 2017). The quantitative research design was used in the research where the data collected from the images taken can be quantified in numbers to represent the output. Similarly, the correlation research design was used in the study as features in a given time-series imagery can be detected and used to predict an outcome with no supervision or control by the researcher (Kothari, 2017).

Overall, the study adopted the mixed-methods research design where both the quantitative and correlational methods were used to structure, test and analyze the research. The design is chosen given the various attributes characterizing the mixed-method design and the diverse nature and goal of the study. The study is a problem driven research on a case where the mixed-method design constructs an approach that anchors most focus on examination of the research problem than the methodology used.

3.3 Population and Sampling

3.3.1 Population

Advancement in exploratory technologies and the consequent launch of optical earth orbit satellites such as the Sentinel-2A Satellite Sensor has catalyzed the generation and distribution of satellite imagery globally (Satellite Imaging Corp, 2021). Satellite images distributed includes images of interest for the purpose of conducting this research.

To assert a more focused study, the research picked out satellite imagery collected from the Sentinel-2A Satellite Sensor and targeting specific maize crop fields within the East African region. Images included a time-series set of data collect in the period between 2016 and 2019.

As illustrated in Appendices H and I, spectra bands from two main sources (Sentinel 2 and TERRACLIM respectively) were considered, summing the total number of bands to 30. Each band carry specific spectra data such as cloud mask, water vapor, precipitation accumulation, and wind speed among others. Each image collected included data for each of these bands.

Sentinel 2A collects multiple images for a given field within a month. However, for this study only one image was considered for each month. Images with the least cloud mask were given higher preference. Considering the 12 months for each year, the data collected produced 360 image bands or features for each crop field under study. Figure 3.1 illustrates a sample image from the dataset collected showing the visible spectra bands for a given month.

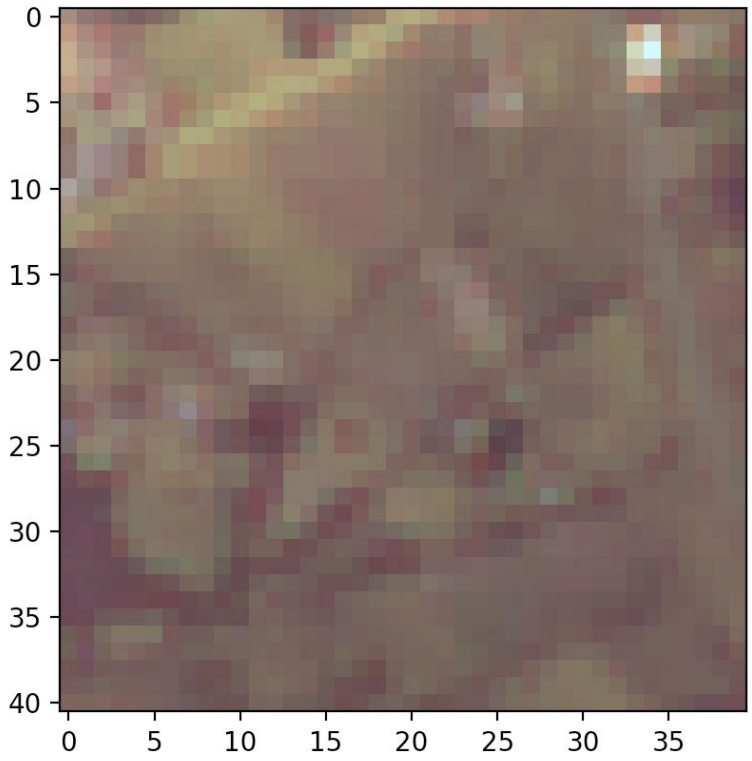


Figure 3.1: Sample Image Illustrating the Visible Bands

All images were presented in a 10m resolution and with 41px a side. The center of the image captured the field location. The image file name corresponded to a pre-assigned field ID for easy identification.

The total population consisted of 2977 train records and 1055 test records. Each record in the train population had a pre-assigned label for "yield", indicating its production capacity in tons per acre.

3.3.2 Sampling

The sampling technique used was simple random sampling representation of the data. This is because probability sampling supports the law of statistical regularity which states that on average if a random sample is chosen, it can carry the same characteristics as the population (Kothari,

2004). Sampling was done because of the random nature of the images used in the study. These images were pre-processed using data augmentation techniques.

Data acquired was further divided into three data sets. The train population of 2977 records were subdivided into two, 90% of the images used for training the model whereas 10% for validation. The test population of 1055 records formed the third set of data and was used to test the performance of the model when given a real-world-like prediction task.

3.3.3 Data Collection

The study used secondary source of data as published by Zindi (2020). This dataset comprised both satellite imagery datasets and crop-cut yield estimates data. Crop-cut yield estimates data was used to estimate and assign the "yield" capacity of each maize crop field in tons per acre for each of the growing seasons in the four years of collection. The yield was used to train the model developed, associating the detected features and attributes of a given satellite imagery with its yield capacity. Model training was done an effort to improve model performance and coverage by minimizing model error.

Consequently, model testing was done to evaluate accuracy, scalability, and performance of the model. This was done using real life satellite imagery collected using the Sentinel-2A Satellite Sensor for maize fields in the same region. This was provided as an extra dataset whose "yield" capacity of these extra dataset was not labelled (Zindi, 2020).

3.4 Model Development

3.4.1 Obtaining Data

This step involved identification, authorization and access of the data needed for the research. The activity involved writing a formal data access and use to Zindi, the rightful owners of the datasets. A positive response was received, authorizing the researcher to use and publicly publish the dataset, while observing necessary copyrights (Zindi, 2021).

Images acquired for the study were in the .pny format. The image file name corresponded to a pre-assigned field ID for easy identification. All additional data sets such as year and yield labels were acquired in the .csv formats except for metadata information that was represented in .docx format.

3.4.2 Data Pre-processing

This stage involved data engineering and feature engineering to select the features most relevant for the prediction task. The study employed various stages of data preprocessing beginning with missing value replacement and noise reduction to be analyzed and executed through smoothing or baseline reduction. Consequently, the study analyzed the data and perform the appropriate data transformations in an effort to conform the data to formats required in the analysis. Final step to be taken in data preprocessing was data feature reduction.

3.4.3 Development of Model

The model was developed using pytorch and sklearn libraries, which are open-source Python libraries.

3.4.4 Validation of Model

The model developed was validated using the mean square error Mean Squared Error (MSE) prediction quality estimator. Mean Squared Error (MSE) or Mean Squared Deviation (MSD) focuses on measuring the average squared difference between the actual value and the estimated values. It constitutes a risk function that corresponds to the expected value of the squared error loss (Lehmann & Casella, 2006; cited in Deisenroth et al., 2020).

Since MSE has the capability to measure quality of both a predictor and estimator functions, its definition and use varies depending on context. In this study, MSE was used to describe the quality of the CNN-based model as both an estimator and a predictor. Equation 3.1 illustrates the computation of MSE when used to determine quality of a predictor function. In this case, *n* is the number of instances, $\hat{Y}_{pred(i)}$ is the prediction of observation *i* and Y_i is the expected value.

Equation 3.1: Formulae for computing MSE of a predictor function (Adapted from Lehmann & Casella (2006) cited in Deisenroth et al. (2020))

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2.$$

Equation 3.2 illustrates computation of MSE when used to determine quality of an estimator function. In this case, the MSE of an estimator $\hat{\theta}$ is calculated with respect to an unknown parameter θ.

Equation 3.2: Formulae for computing MSE of an estimator function (Adapted from Lehmann & Casella (2006) cited in Deisenroth et al. (2020))

$$\mathrm{MSE}(\hat{\theta}) = \mathrm{E}_\theta\left[(\hat{\theta} - \theta)^2\right].$$

3.5 Research Reliability and Validity

Reliability in research is described by the levels of consistency of results obtained over a specified period of time. It is the ability to correctly instantiate the aggregate population used and results obtained from a research study. As such, a study is considered reliable in the event that its outcome can be replicated in similar approaches (Sudo, 2019). Similarly, research validity is considered as a critical factor in measuring a research's instrument. Validity gauges whether the findings indeed represent the real scenario of what they appear to be (Sudo, 2019).

Emphasis was placed to construct validity and reliability in this study. In an effort to further assert reliability and validity of the research and the model developed, this study applied Mean Squared Error (MSE) prediction quality estimator to assess the prediction outcomes obtained.

3.6 Ethical Considerations

Ethics define the standard of interaction between people, professionals, corporate or users and constitute a key framework that detail the quality of a research study (Helps, 2017). Aksnes, Langfeldt and Wouters (2019) describe research quality as a multidimensional notion that upholds originality, soundness and plausibility, societal value and scientific value as key traits. This study followed these quality standards, ensuring that all literary works used are obtained from reputable sources and are correctly cited. The data obtained and used in the study was presented as is, except in instances of duplicate or missing data entries that would depreciate the quality of the developed model.

In carrying out the research, utmost care was taken to ensure privacy of all parties involved. The study adhered to the moral and legal requirements of research. As a result, the researcher ensured that access to required data, its privacy and confidentiality, and its protection and storage was purely be limited to the research.

Chapter 4: System Analysis, Design and Architecture

4.1 Introduction

This chapter outlines the system analysis and architectural designs of the developed crop yield prediction system. System analysis is done to facilitate a full understanding of the needs and requirements of the system in efforts of creating a flawless construction. System architectural design follows the information processing conceptual framework as outlined in Figure 2.8. The section identifies and discusses in detail the various components of developed system and the interactions between each of these components. Use cases, sequence diagrams, context diagrams and entity relationship diagrams were all used to model and indicate these interactions.

4.2 System Analysis

System analysis purpose to understand and document the essential characteristics of a system under study. In this research, system analysis was conducted in two key steps including functional requirements and non-functional requirements analysis.

4.2.1 Functional Requirements

i. The system should allow user to specify the crop grown and upload a satellite imagery capturing spectra data for a particular crop field. Recently captured satellite images are preferred for a real-time prediction. Uploaded images should be in .pny format. Any other formats of data input should be rejected.

ii. The system should be able to detect yield impacting features of a given satellite imagery received from the uploaded data.

iii. The system should be able to associate the detected features and attributes of a given satellite imagery with the yielding capacity of the grown crop. Hence predict the yielding capabilities of the crop.

iv. The predicted yield capability should be valid based on the input given by the user.

4.2.2 Non-Functional Requirements

4.2.2.1 Usability

The system should be simple and straightforward. Among the main users of the system are farmers who might have minimal technical understanding of current technologies. Supporting simplicity will further promote validity and acceptance of the system.

4.2.2.2 Scalability

If there is an increase in the number of users or breath of data being uploaded to the system, the proposed solution should be able to handle the extra load, accepting new users and new data submissions and performing the adjacent predictions without breaking down.

4.2.2.3 Persistent Storage

The system should provide permanent storage for all predictions made. This data is used as training data for future predictions as a mean of improving accuracy of predictions made.

4.2.2.4 Accuracy

Predictions made from the system should be accurate since its outcomes implies profit expectations of precision farming and hence might affect livelihoods of the farming entities.

4.2.2.5 Supportability

The system should be accessible as a web application supporting accessibility across all major browsers in mobile and desktop devices.

4.2.2.6 Functionality Restoration

The administrator should be able to correctly restore the system to a functioning state in the event of a failure.

4.3 System Architecture

System architecture denotes the interactions between inputs, processes and the anticipated outputs of a given system. Figure 4.1 represents the system architecture of the system. The architecture

diagram aims at visualizing information flow from one system component to another. It models all steps input data has to go through for it to be processed fully and before it can be used for predictions. For this system, the architecture diagram also stands out as a blueprint from which other system design diagrams are modelled.

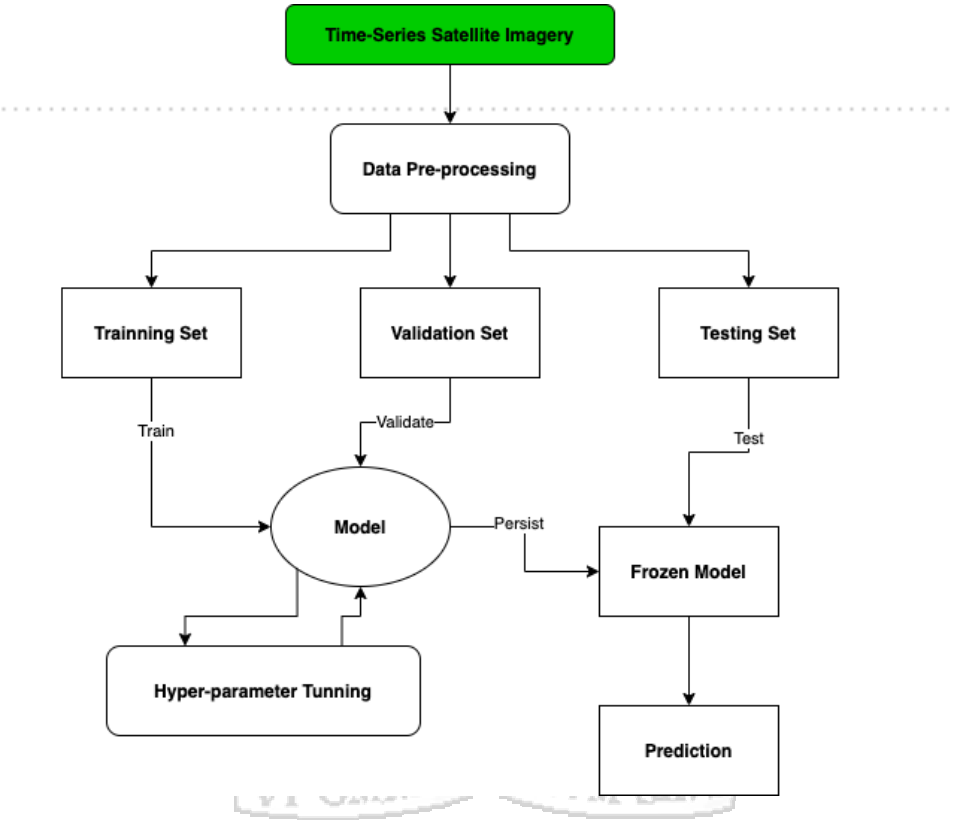


Figure 4.1: System Architecture Diagram

4.4 System Design

The design of the crop yielding prediction system in agriculture was executed in a manner to suggest conventional and reliable data collection from varied farming entities to form a formidable farming database with effaceable analytical tools. In order to achieve this, several diagrams modelled from the Unified Modelling Language (UML) were necessary to ensure proper design of the model for structured development and adherence to the agile model of the system development life cycle.

Several diagrams are represented in the rest of this section. They form the collective design approach taken in prototyping and development the algorithm. The first of these diagrams drawn to UML standards was the Use case diagram as shown in Figure 4.2. The diagram illustrates the different actors in the system and the relationships between actors.

4.5 Use Case Diagram

The main actors in the system include the system administrator and a typical agricultural practitioner such as a farmer. The administrator presents new data sets to the system, performs data pre-processing and model training. The data sets added builds up the system database, issues larger sets of data to be used in model training and hence creating room for more accurate predictions.

The actors who interact with the system have been identified and represented in Figure 4.2. A farmer interacts with the system by specifying the crop grown and uploads at least one recent satellite imagery containing the spectra information of the crop field. With these specifications and data, a prediction is made, and final output of the system comprise of the yielding capability of the crop. This step utilizes the functionality of the frozen model develop and persisted during model training. Table 4.1 provides extensive details on the main use cases in the system and their success scenarios.

Figure 4.2: System Use Case Diagram

The definition of actors in relation to the roles of the system available to them is shown in Table 4.1.

Table 4.1: Use Case Description

Actor	**Use Case**	**Description**
User - Farmer	Select Crop	A farmer should be able to select crop for which they want to predict yield capacity.
	Upload Image	The farmer should be able to upload at least one recent satellite imagery containing the spectra information of the crop field.

	Perform Prediction	The farmer should be able to trigger a prediction event based on the predefined parameters. Output of the system comprise of the yielding capability of the crop
Administrator	Pre-process Data	The system administrator should be able to pre-process large datasets in preparation for model training and update. Data pre-processing comprises of data standardization efforts such as cleaning, transformation, compression and reduction, preparing the data for use in analysis and predictions.
	Train Model	When performing model training, the administrator manipulates results obtained from the data pre-processing stage to train the model.
	Save Model	The administrator should be able to save the trained model.

4.6 Data Flow Diagrams

4.6.1 Context Diagram

The context diagram captures the high-level data flow and how the users interact with the model. It shows the various types of inputs and outputs from the classification as shown in Figure 4.3. The farmer and the system administrator being the main users.

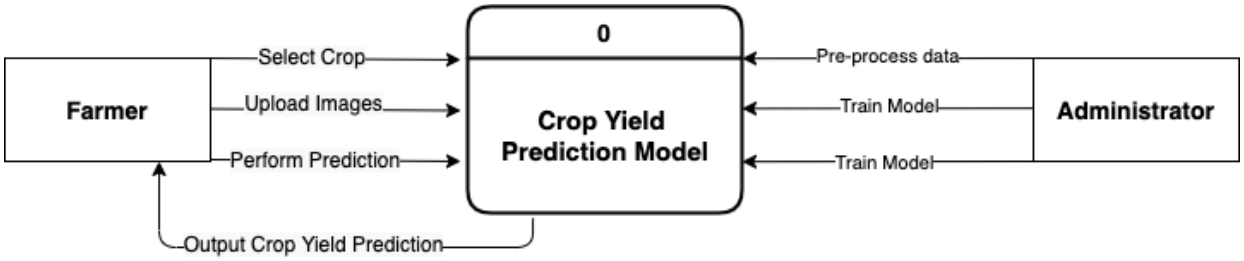


Figure 4.3: System Context Diagram

4.6.2 Sequence Diagram

Figure 4.3 outline the system sequence diagram. A farmer presents a satellite image for a specific crop field and uploads to the system. The uploaded data undergoes pre-processing to remove all noisy elements, making it ready for analysis and prediction. The system then extracts the key growth and yielding indicators of the crop from the uploaded data using the developed model and uses these as the main base model input. The system then performs the prediction using the model and outputs prediction results.

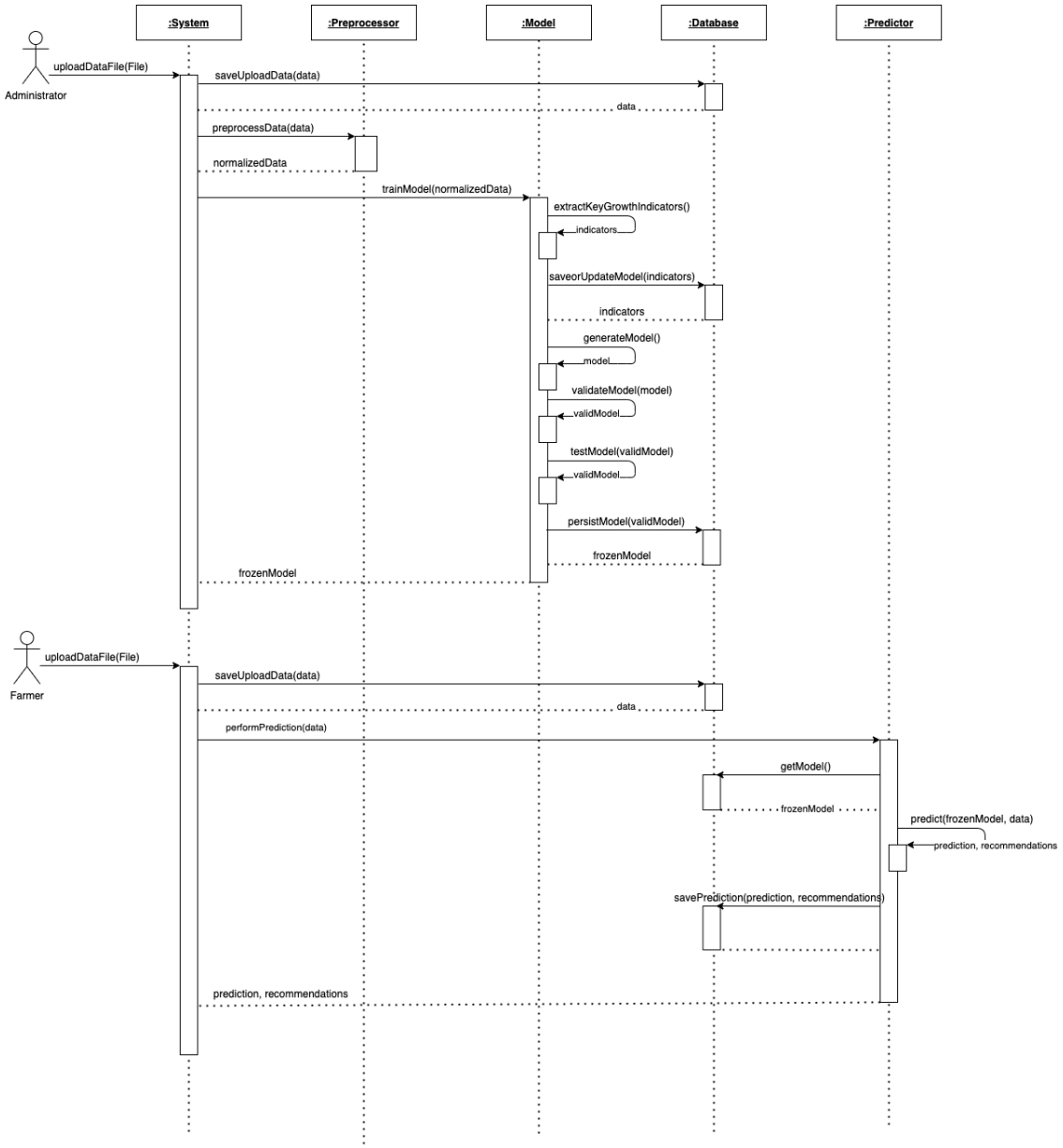


Figure 4.4: System Sequence Diagram

4.6.3 Entity Relationship Diagram

The entity relation diagram models the different tables in the system as well as the relationships these tables have as pertains the effective running of the system. The table is the most basic

component of the database. As such, it is important to have all relations properly designed in order to develop an efficient database for the system.

Indeed, the development of the database is important for this system. While data plays a key role to the operability of the system, the database acts as the single source of truth for all data used, capturing all data including the user inputs, model training, testing and validation data sets. The prototype takes advantage of the relations in the database SQL schema. Figure 4.5 effectively represents the entity relation diagram as modelled into the system for predicting the yielding potential a given crop.

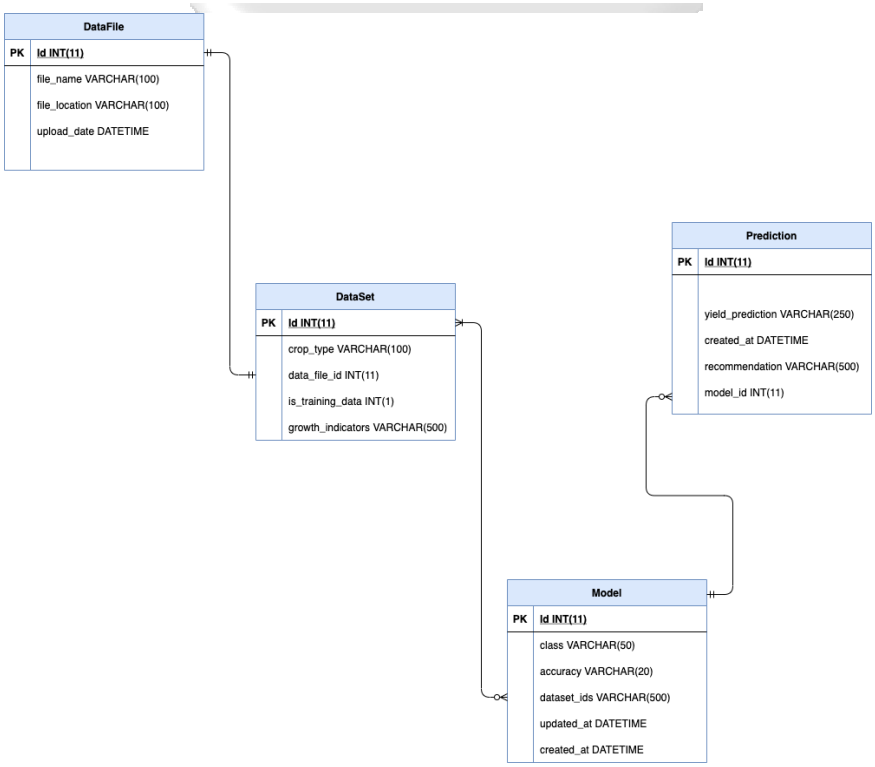


Figure 4.5: Entity Relationship Diagram

Chapter 5: System Implementation and Testing

5.1 Introduction

This chapter concentrates on the implementation and testing of the proposed crop yield prediction system. The focus on implementation is to bring a clear view of the different modules of the system and how each function. Testing on the other hand looks at whether the system is usable and functional so as to determine if the set objectives have been achieved.

5.2 Development Environment

Model development utilized Google Colab as the primary development and testing environment (Google Colabolatory, 2021). Alternative development included local development using a laptop or desktop, Digital Ocean servers and Kaggle Data Science and Machine Learning Code platform (Fuat, 2019). Although alternative options exist, Google colab was mostly preferred due to its ability to support deep learning processes free of charge.

Google Colab is a free cloud service which offers free GPU (Graphics Processing Unit) that were essential for building this model. The application of deep learning required very expensive hardware and powerful libraries such as TensorFlow, PyTorch, Keras, and OpenCV.

To setup this environment, a Google account for resource access was created and configured to use GPU hardware. Model training, validations and testing activities were using a common environmental setup.

5.3 Hardware Resources

Hardware resources used in system development were limited to those offered by the Google Colaboratory platform. Table 5.1 depicts hardware resources used:

Table 5.1: Hardware Resources

Hardware	**Specifications**	
Google Colab	GPU	1xTesla K80, having 2496 CUDA cores, compute 3.7, 12GB (11.439GB Usable) GDDR5 VRAM
	CPU	1xsingle core hyper threaded

5.4 Software Resources

Software resources used for the study mainly consisted of the programming language and software libraries used to build the model. The primary programming language used is python. The torch python library was used to provide two high-level features that played a key role in development of the model. These features include tensor computations such as NumPy and SciPy and the deep neural networks feature with gpu-ready capabilities. Torch was also used to save and load the trained model. Table 5.2 illustrates the overall software requirements of the developed model.

Table 5.2: Software Specifications

Software	**Library**	**Specification**
Python 3.9	sklearn	0.24.1
	Torch	1.8.1
	NumPy	1.19.4
	Pandas	1.2.4

5.5 Model Components

The developed model extends the basic principles of artificial neural networks. As illustrated in Figure 5.1, input, hidden and output layers consisted of the primary components of the model (Barry-Straume et al., 2018).

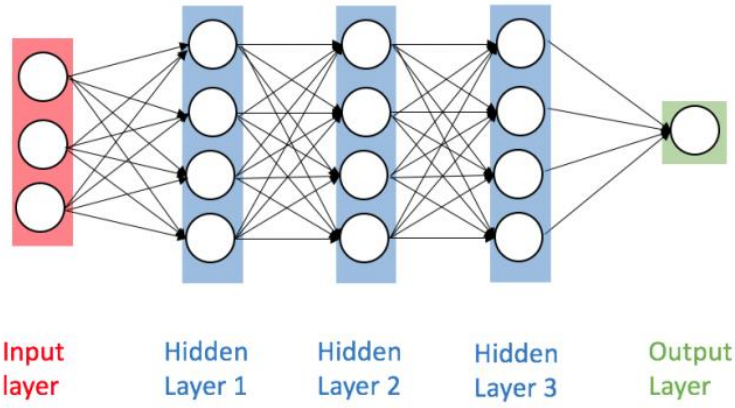


Figure 5.1: Components of Artificial Neural Networks (Adapted from Barry-Straume et al. (2018))

5.5.1 Storage

Since Zindi (2021) is a private platform, alternative data hosting solutions were required for easy access at manipulation during the development of the model. As such, acquired data was downloaded and uploaded to the Google Drive platform. The platform was chosen due to its ability to offer hosting of large data sets for free. Google Drive also offers free integration with the Google Colab platform, further easing the setups required for preparing the development environment. Figure 5.2 illustrates a code snippet used to create the connection into the Google Drive storage platform from Google Colab.

```
from google.colab import drive

drive.mount('/content/gdrive')
root_path = 'gdrive/MyDrive/colab/datasets/crop_yield/'
```

Figure 5.2: Google Drive storage platform integration

With the root path of the Google Colab operations set at the Google Drive folder, all output data, including the .pth files that stored the frozen model, were stored there.

5.5.2 Input Layer

The input layer comprised of the model component responsible for interactions with the external environment. It is through this layer that model training, test, and validation data sets were loaded. As illustrated in Appendix A, this layer expected image arrays of size (360, 32, 32) which were labelled with their "yield" capacity.

5.5.3 Output Layer

This included the final layer of the developed model. From it, model outputs were exposed to the external environment. While the major computations of the model occur within the hidden layers, results of these layers are forwarded to the output layer which in return exposes them for inference by the users.

5.6 Data Pre-processing

Understanding the dataset used proved to be a vital step for a successful prediction task. As a result, detailed examination of acquired datasets was done.

5.6.1 Field and Location Data Pre-Processing

All crop fields considered in the study were labeled with a unique Field_ID. GPS positions were used to identify the location of each field and was recorded during data capture. However, due to minor and major offset errors, some of the recorded positions fell at the edge of the field while others fell completely outside the field respectively.

In efforts to reduce possible errors that might result from this phenomena, manual review of field locations was done. Each field was assigned with a 'Quality' attribute. The labels 'Good', 'Medium', and 'Poor' were used to denote field locations within a single field, those adjusted to lie closer to the center of that field, and those with no obvious field associated with them respectively.

5.6.2 Image Data Pre-Processing

The developed model aims at generating a model that could perform crop yield prediction from remote sensed data. To achieve this, images formed a core input for the model. Any distortion to the uploaded images could lead loss of key spectra data attributes, resulting in a poorly trained and performing model.

To examine datasets that were in form of .pny files, an image visualized class was developed. Appendix G demonstrates code sample for the Image Visualizing function. Figure 5.3 demonstrates output from the image visualizing function. The output comprised of 12 images with each image representing each month of the year and showing the visible bands.

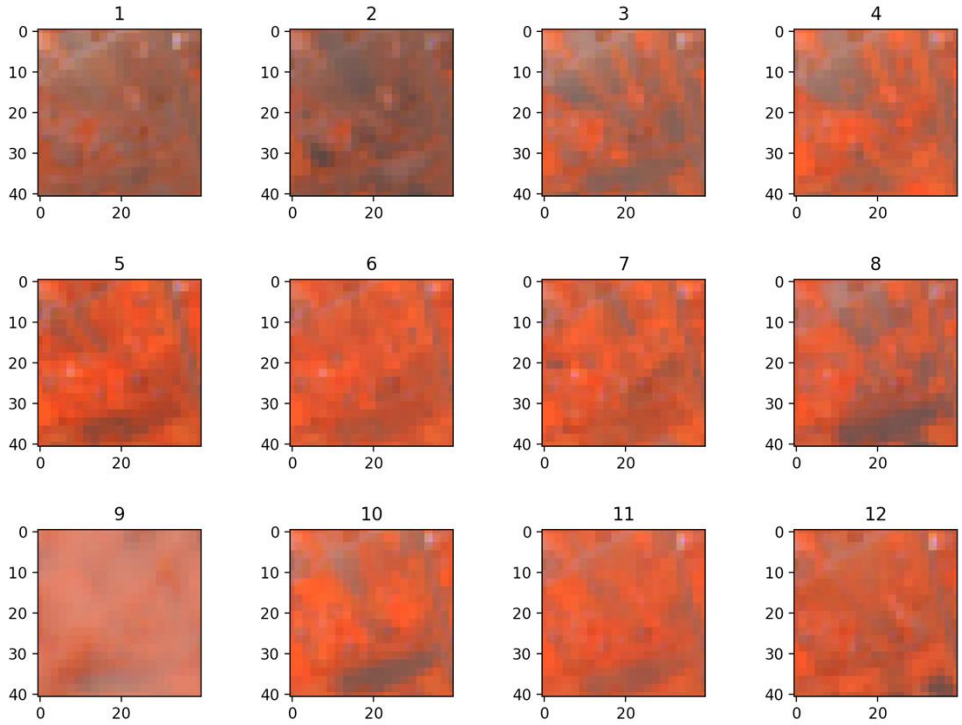


Figure 5.3: False Color Images for each Month in a Year

As illustrated in Appendix A, a dedicated class was developed through which images could be loaded and pre-processed uniformly. The Image processor class expected path to a folder containing the .pny images. It then loaded all the files in the repository provided while reshaping and transforming them into tensors using the torch python library. Figure 5.3 demonstrates sample use case for the image processor class

```
training_set = PreProcessor(
    X_train.reset_index(drop=True),
    y_train.reset_index(drop=True),
    'gdrive/MyDrive/colab/datasets/crop_yield/image_arrays_train'
)
validation_set  = PreProcessor(
    X_test.reset_index(drop=True),
    y_test.reset_index(drop=True),
    'gdrive/MyDrive/colab/datasets/crop_yield/image_arrays_train'
)
```

Figure 5.4: Loading and Processing Satellite Images

5.7 Model Implementation

The model was compiled on a loss function of mean squared error. This was the choice of function because the images represented a multiclass grouping. Meaning they were mutually exclusive on a case where Mean squared error loss functions perform relatively well in prediction of relative distances between inputs of continuous data.

The Adam optimizer was used. Adam was the best choice because of its ability to combine the benefits of AdaGrad and RMSProp which can manage noisy datasets in deep learning. The hidden layers consisted of convolution layers, batch normalization, dropout, and max pooling layers. The main purpose of the hidden layers was to provide maximum output to what was expected in the output layer. These were stacked in between the input layer and the output layer as shown in Appendix B.

5.7.1 Model Training

Appendix D demonstrates the model training function used in this research study. The train_test_split function from sklearn python library was used split the overall dataset into training and testing sets. 90% of all datasets acquired was used to train the model. Figure 5.4 demonstrates the splitting function used.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
    train['Field_ID'],
    train['Yield'],
    train_size=0.9,
    random_state=42)
```

Figure 5.5: Dataset splitting function

The training function tracked model training, recording loss index for each epoch trained. Figure 5.5 illustrates the performance progress of model training tracked in form of a loss curve.

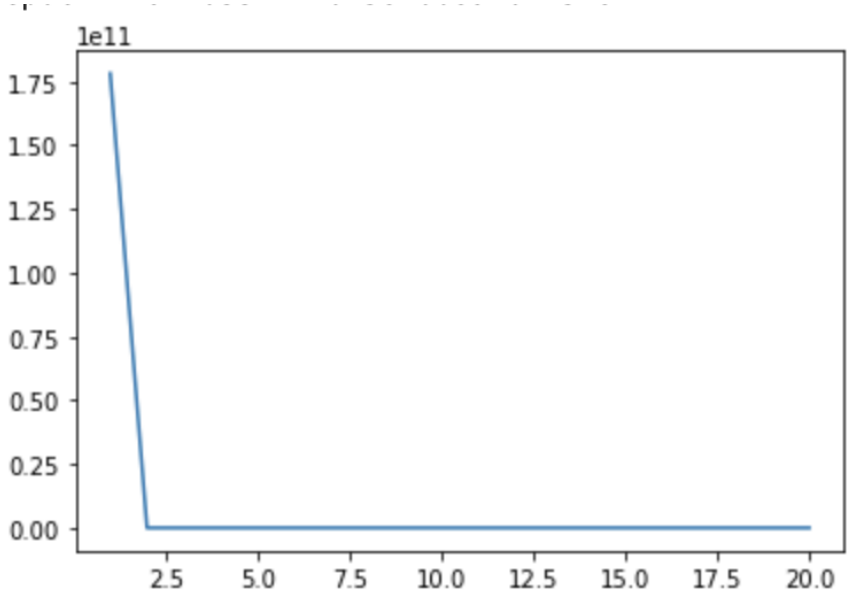


Figure 5.6: Training and Validation Loss

5.7.2 Model Validation

Appendix E demonstrates the model validation function used in this study. This was done in efforts of identifying accuracy of predictions made from developed model. 10% of the overall dataset was used for model validation purposes. Cumulative accuracy index was achieved by computing the sum correct predictions and dividing this by total test sample size. The model achieved an accuracy of 87%.

5.7.3 Model Testing

Testing was done to evaluate the performance of the model when given a real-world-like prediction task. To achieve this; yield capacity, year of data collection, and quality of crop field data points were stripped off from the test data set. The developed model was expected to estimate yield

capacity of a given field based on given satellite image data only. However, the test sets comprised of field locations known to have 'Good' or 'Medium' quality for effective validation of the test results. A total of 1055 image records were used to perform model testing. Appendix F demonstrates the model testing function used in this study.

5.7.4 Hyper-parameter Tuning

Parameter setting can make or break the performance of a given model. Other than determining model structure, the hyper-parameters can also be used to control running environments and system requirements such as memory used. For this study, there key parameters were considered while designing and tuning the model. These parameters include learning rate, epochs, and optimizers.

5.7.4.1 Optimizer

The adaptive moment estimation (Adam) optimizer was used in this study. Extending the classical stochastic gradient descent procedure, Adam optimizers function by calculating an exponential moving average of the gradient and the squared gradient. The optimizer was used update network weights iterative based in provided training data.

5.7.4.2 Learning Rate

This is the most crucial hyper-parameter. An optimal learning rate must be used for each model to guarantee optimal accuracy indices. Too low learning rates can result in low speeds while too high learning rates can result in overfitted models. A learning rate of 0.01 was used in this study.

5.7.4.3 Epoch

The epoch hyper-parameter allows splitting of model training into stages. Low number of epochs can result in highly undertrained model while high number of epochs might result in model overfitting. To avoid these extreme outcomes, this research study maintained an epoch count of 20.

5.7.4.4 Batch Size

The batch size hyper-parameter represents the number of examples from the training dataset used in the estimate of the error gradient. It is a key configuration to the model architecture. It heavily contributes to determining the learning parameters and controls the accuracy of the estimate of the error gradient when training neural networks. This study uses a batch size of 128. A higher batch size is preferred, but it also implies high memory requirements.

Chapter 6: Discussions

6.1 Introduction

The aim of this study was to design and develop a computer vision-based model capable of predicting yield capacity of a given crop by use of satellite imagery. In this regard, this chapter includes a discussion on the key findings of the study in relation to the literature on crop yield prediction. The chapter attempts to comparatively validate the findings of research while systematically illustrated how the questions posed by the study were answered. The chapter concludes by identifying the major limitations of the study.

6.2 Factors Influencing Crop Yielding in Agriculture

The study confirmed that the numerous factors affecting crop yielding, identified by the acronym GEMS, have greatly contributed to poor performance of existing yield prediction models. Collection of these data points is either costly or prone to high levels of noise. Furthermore, the design and development yield prediction solutions by use of these datasets results in poorly scalable models. This assertion is attributed by the fact that GEMS data is site specific and hence application of the same on a global level would be highly erroneous.

6.3 Techniques and Approaches used in Crop Yield Prediction

As discovered in chapter 2, numerous studies have been conducted in efforts to perform crop prediction in agriculture. A number of these have utilized various machine learning approaches, including Random Forests, Support Vector Machines and Deep Neural Networks among others. Most studies carried out relied on locally sensed data and crop-cut yield estimates to perform crop yield prediction within the same region as the region of data collection.

6.4 Computer Vision-based Model for Crop Yield Prediction Using Remotely Sensed Data

To overcome these challenges, this study leveraged an alternative source of data to design and develop an accurate and scalable deep learning model. Satellite imagery datasets were used as the primary and only source of data for training the model. This benefited the study in two major ways.

Firstly off, the approach automatically took care of the high dimensionality problem as demonstrated in the GEMS data. Second, satellite imagery data is readily available globally, a factor that greatly reduced the costs needed to collect real-time data for the study.

Image datasets collected from the Sentinel 2 satellite were pre-processed, trained and tested using a convolutional neural network. The training set was such that we needed to use data for three different classes to be able to validate the model. The testing set was used to ensure that the model's output values were nearly identical to the values initially in the dataset. Performance metrics that were used in the evaluation of these models are accuracy, precision, recall and F1-score. These metrics did not only help in selecting the best model but also in validating the model.

6.5 Validation of the Developed Model

Validation of the developed model was done using 10% of the overall training dataset. Reliability of the model in performing crop yield predictions was captured using an MSE loss function for each epoch trained. Cumulatively, and as illustrated in Figure 6.1, the model achieved an MSE score of 3.6. The cumulative MSE score was calculated from the loss index of the last 10 epochs trained. This is due the fact that MSE loss in a convolutional neural network decreases exponentially as the model trains and stabilizes to a flat curve after the model learns of all the important features in the dataset provided. Figure 5.5 demonstrates this assertion.

```
Begin Training::
epoch: 0 loss: 5285196126.74217
epoch: 1 loss: 9.805497732662424
epoch: 2 loss: 4.901175707406072
epoch: 3 loss: 4.05499409231577
epoch: 4 loss: 3.976066927686728
epoch: 5 loss: 3.9357040582426674
epoch: 6 loss: 3.8396807271679134
epoch: 7 loss: 3.8056066639369956
epoch: 8 loss: 3.7502568830330083
epoch: 9 loss: 3.862740108685 9
epoch: 10 loss: 3.7783668230792773
epoch: 11 loss: 3.8173172522387926
epoch: 12 loss: 3.719425228283345
epoch: 13 loss: 3.690411677492351
epoch: 14 loss: 3.6521714435092516
epoch: 15 loss: 3.645211614803596
epoch: 16 loss: 3.6120634075161084
epoch: 17 loss: 3.509577846856437
epoch: 18 loss: 3.5679510743092453
epoch: 19 loss: 3.643093476554053
tensor(1245.6632, dtype=torch.float64, grad_fn=<MseLossBackward>)
```

Figure 6.1: MSE Loss Scores of the developed model

This performance level was good and acceptable as compared to existing yield prediction models. Table 6.1 demonstrates the comparison of performance of the developed model to the existing yield prediction models for crops. Results for the LSTM, LSTM + GP and DNN models are as realized and discussed in chapter 2. The developed model was seen to outperform, even when compared with other models that use remotely sensed data to achieve the crop yield prediction task.

Table 6.1: Comparison of crop yield prediction models which use remotely sensed data

Model	**RMSE**	**R^2**
LSTM	-	60
LSTM+GP	-	71.5
DNN	12.085	83.685
CNN	3.6	-

6.6 Limitations of the Study

Three key limitations are identified from the study.

i. The model did not account for the varying genotypic characteristics that affect crop yielding independent of the spectra data of the crop field.
ii. Model training and testing was done on only one crop type.
iii. The model did not provide solutions to manage factors limiting crop yielding.

Chapter 7: Conclusion, Recommendations and Future Works

7.1 Conclusions

The primary aim of this research study was to design and develop an inexpensive, accurate and scalable technique for crop yield prediction. This was achieved by coupling a computer vision-based model with remote sensed data. To guarantee high performance of the model, the study employed data pre-processing strategies on the acquired satellite image data. Similarly, model hyper-parameter tuning was done in efforts of minimizing system requirement costs and optimize performance of the model.

Model validation was done using 10% of the training dataset while 1055 of a total of 4032 was used for model testing. The model was confirmed to offer superior performance as compared to existing models and techniques for crop yield prediction. With such as solution, it would be easy to scale the model and apply it in prediction of yield capabilities for different crops and in varying geographical regions.

7.2 Recommendations

The findings of this study are a step further in the fight against food insecurity in the world today. It would be important to combine these finding with existing techniques, forming hybrid approaches for improving agricultural policy and creating sustainable food channels. Immediate adoption of finding of the study in developing countries is highly recommended. Unlike developed countries, developing countries have limited alternative datasets that could be used in crop yield prediction tasks. Individual farmers are also encouraged to explore the numerous benefits of the developed technique. Some of the advantages they would harness is timely identification of ready markets as well as improved socio-economic planning.

7.3 Future Works

As presented earlier, the limitations of this research work open up opportunities for future areas of exploration and study. Below include the recommended future research areas within the domain of crop production and crop yield prediction:

i. Further research can be done to identify and discover the possibilities of using spectra data to recognize genotypic characteristics of the grown crop. This could help improve accuracy level of the yield prediction model and as well promote targeted outcomes from the prediction task.

ii. Further studies can be done to test and validate performance of the model in yield capacity prediction for crops other than the maize plant.

iii. The model should be enhanced such that it is capable of suggesting best practices to adopt on a given field in order to improve crop yielding capacity.

References

Abbas, F., Afzaal, H., Farooque, A. A., & Tang, S. (2020). Crop yield prediction through proximal sensing and machine learning algorithms. Agronomy, 10(7), 1046.

Aksnes, D. W., Langfeldt, L., & Wouters, P. (2019). Citations, Citation Indicators, and Research Quality: An Overview of Basic Concepts and Theories. SAGE Open, 9(1), 2158244019829575. https://doi.org/10.1177/2158244019829575

ANALYTICS VIDHYA. (2016, March 21). PCA: Practical Guide to Principal Component Analysis in R & Python. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2016/03/pca-practical-guide-principal-component-analysis-python/

Anitescu, C., Atroshchenko, E., Alajlan, N., & Rabczuk, T. (2019). Artificial neural network methods for the solution of second order boundary value problems. Computers, Materials and Continua, 59(1), 345–359. https://doi.org/10.32604/cmc.2019.06641

Athey, S., Tibshirani, J., & Wager, S. (2019). Generalized random forests. Annals of Statistics, 47(2), 1148–1178. https://doi.org/10.1214/18-AOS1709

Aubert, B. A., Schroeder, A., & Grimaudo, J. (2012). IT as enabler of sustainable farming: An empirical analysis of farmers' adoption decision of precision agriculture technology. Decision Support Systems, 54(1), 510–520. https://doi.org/10.1016/j.dss.2012.07.002

Barry-Straume, J., Tschannen, A., & Engels, D. W. (2018). An Evaluation of Training Size Impact on Validation Accuracy for Optimized Convolutional Neural Networks. 1(4), 18.

Cai, J., Luo, J., Wang, S., & Yang, S. (2018). Feature selection in machine learning: A new perspective. Neurocomputing, 300, 70–79. https://doi.org/10.1016/j.neucom.2017.11.077

Carolan, M. (2017). Publicising Food: Big Data, Precision Agriculture, and Co-Experimental Techniques of Addition. Sociologia Ruralis, 57(2), 135–154. https://doi.org/10.1111/soru.12120

Chaovalitwongse, W. A., Chou, C.-A., Liang, Z., & Wang, S. (2017). Applied optimization and data mining. Annals of Operations Research, 249(1–2), 1–3. https://doi.org/10.1007/s10479-017-2402-x

Chivenge, P., & Sharma, S. (2019). Precision agriculture in food production: Nutrient management. INTERNATIONAL WORKSHOP ON ICTs FOR PRECISION AGRICULTURE.

College of Food, Agricultural and Natural Resource Sciences. (2020). GEMS Platform. Data Driven Agricultural Innovation. https://gems.agroinformatics.org/webui/

Committee on Science Breakthroughs 2030: A Strategy for Food and Agricultural Research, Board on Agriculture and Natural Resources, Board on Atmospheric Sciences and Climate, Board on Life Sciences, Water Science and Technology Board, Division on Earth and Life Studies, Food and Nutrition Board, Health and Medicine Division, Board on Environmental Change and Society, Division of Behavioral and Social Sciences and Education, & National Academies of Sciences, Engineering, and Medicine. (2019). Science Breakthroughs to Advance Food and Agricultural Research by 2030 (p. 25059). National Academies Press. https://doi.org/10.17226/25059

Crowdflower. (2016). Data Science Report. http://visit.crowdflower.com/rs/416ZBE-142/images/CrowdFlower_DataScienceReport_2016.pd

Deisenroth, M. P., Faisal, A. A., & Ong, C. S. (2020). Mathematics for Machine Learning. Cambridge University Press.

Dodds, F., & Bartram, J. (2016). *The Water, Food, Energy and Climate Nexus: Challenges and an agenda for action*. Routledge & CRC Press. https://www.routledge.com/The-Water-Food-Energy-and-Climate-Nexus-Challenges-and-an-agenda-for/Dodds-Bartram/p/book/9781138190955

Filippi, P., Jones, E., Bishop, T., Acharige, N., Dewage, S., Johnson, L., ... & Whelan, B. (2017, October). A big data approach to predicting crop yield. In Proceedings of the 7th Asian-Australasian Conference on Precision Agriculture (pp. 16-18).

Franz, T. E., Pokal, S., Gibson, J. P., Zhou, Y., Gholizadeh, H., Tenorio, F. A., ... & Wardlow, B. (2020). The role of topography, soil, and remotely sensed vegetation condition towards predicting crop yield. Field Crops Research, vol. 252, p. 107788.

Fuat. (2019, March 25). Google Colab Free GPU Tutorial. Medium. https://medium.com/deep-learning-turkey/google-colab-free-gpu-tutorial-e113627b9f5d

Gaster, B. R. (2012). Heterogeneous computing with OpenCL. Morgan Kaufmann.

Gilbertson, J. K., & van Niekerk, A. (2017). Value of dimensionality reduction for crop differentiation with multi-temporal imagery and machine learning. Computers and Electronics in Agriculture, 142, 50–58. https://doi.org/10.1016/j.compag.2017.08.024

Google Colabolatory. (2021). Google Colaboratory. https://colab.research.google.com/drive/1mbFPJAwUxAnhQuzFWIyIUiVtHctVp_OC#scrollTo=KMMCMFi4zPbX&uniqifier=1

Journal of Family Therapy, 39(3), 348–365. https://doi.org/10.1111/1467-6427.12166

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning (Vol. 103). Springer New York. https://doi.org/10.1007/978-1-4614-7138-7

kaggle. (2020). Run Data Science & Machine Learning Code Online | Kaggle. https://www.kaggle.com/notebooks

Kaneko, A., Kennedy, T., Mei, L., Sintek, C., Burke, M., Ermon, S., & Lobell, D. (2019). Deep learning for crop yield prediction in Africa. In ICML Workshop on Artificial Intelligence for Social Good.

Karlsson, A., & Nessvi, C. (2018). - A study of the profitability in a Swedish context. 59.

Khaki, S., & Wang, L. (2019). Crop yield prediction using deep neural networks. Frontiers in plant science, vol. 10, p. 621.

Kothari, C. (2017). Research Methodology Methods And Techniques (Second Edition). New Age International (P) Ltd.,Publishers. https://idoc.pub/documents/research-methodology-methods-and-techniques-by-cr-kothari-pd49gwed36n9

krishna, M., Neelima, M., Mane, H., & Matcha, V. (2018). Image classification using Deep learning. International Journal of Engineering & Technology, 7, 614. https://doi.org/10.14419/ijet.v7i2.7.10892

Labaree, R. V. (2020). Research Guides: Organizing Your Social Sciences Research Paper: Types of Research Designs [Research Guide]. https://libguides.usc.edu/writingguide/researchdesigns

Lehmann, E. L., & Casella, G. (2006). Theory of Point Estimation. Springer Science & Business Media.

Liliane, T. N., & Charles, M. S. (2020). Factors Affecting Yield of Crops. In Agronomy—Climate Change & Food Security. IntechOpen. https://doi.org/10.5772/intechopen.90672

Lindsay, G. W. (2020). Convolutional Neural Networks as a Model of the Visual System: Past, Present, and Future. Journal of Cognitive Neuroscience, 1–15. https://doi.org/10.1162/jocn_a_01544

McInnes, L., Healy, J., & Melville, J. (2018). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. ArXiv:1802.03426 [Cs, Stat]. http://arxiv.org/abs/1802.03426

National Research Council. (1998). Precision Agriculture in the 21st Century: Geospatial and Information Technologies in Crop Management. https://doi.org/10.17226/5491

Ngoune Liliane, T., & Shelton Charles, M. (2020). Factors Affecting Yield of Crops. In Amanullah (Ed.), Agronomy—Climate Change and Food Security. IntechOpen. https://doi.org/10.5772/intechopen.90672

Pham, X., & Stack, M. (2018). How data analytics is transforming agriculture. Business Horizons, 61(1), 125–133. https://doi.org/10.1016/j.bushor.2017.09.011

Rao, S. S. (2019). Engineering Optimization: Theory and Practice. John Wiley & Sons.

Sangeeta, S. G. (2020). Design And Implementation Of Crop Yield Prediction Model In Agriculture. INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH, vol. 8, ISSUE 01.

Sapkota, T. B., Jat, M. L., Jat, R. K., Kapoor, P., & Stirling, C. (2016). Yield estimation of food and non-food crops in smallholder production systems. In Methods for measuring

greenhouse gas balances and evaluating mitigation options in smallholder agriculture (pp. 163-174). Springer, Cham.

Satellite Imaging Corp. (2021). Sentinel-2A SatelliteSensor | Satellite Imaging Corp. https://www.satimagingcorp.com/satellite-sensors/other-satellite-sensors/sentinel-2a/

Schemberger, E. E., Fontana, F. S., Johann, J. A., Souza, E. G. D., Schemberger, E. E., Fontana, F. S., Johann, J. A., & Souza, E. G. D. (2017). DATA MINING FOR THE ASSESSMENT OF MANAGEMENT AREAS IN PRECISION AGRICULTURE. Engenharia Agrícola, 37(1), 185–193. https://doi.org/10.1590/1809-4430-eng.agric.v37n1p185-193/2017

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural Networks, 61, 85–117. https://doi.org/10.1016/j.neunet.2014.09.003

Selçuk, Ş. B. (2019). Information Theory. In Essentials of Mathematical Methods in Science and Engineering (pp. 841–906). John Wiley & Sons, Ltd. https://doi.org/10.1002/9781119580294.ch19

Shafi, U., Mumtaz, R., García-Nieto, J., Hassan, S. A., Zaidi, S. A. R., & Iqbal, N. (2019). Precision agriculture techniques and practices: From considerations to applications. Sensors, 19(17), 3796.

Sharma, P. (2018, August 26). Dimensionality Reduction Techniques | Python. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2018/08/dimensionality-reduction-techniques-python/

Shruthi, U., Nagaveni, V., & Raghavendra, B. K. (2019). A Review on Machine Learning Classification Techniques for Plant Disease Detection. 2019 5th International Conference

on Advanced Computing Communication Systems (ICACCS), 281–284. https://doi.org/10.1109/ICACCS.2019.8728415

Silipo, R., Adae, I., Hart, A., & Berthold, M. (2015). Seven Techniques for Data Dimensionality Reduction. 21.

Solomatine, D. P. (2002). Data-driven modelling: Paradigm, methods, experiences. 38.

Stone, J. V. (2015). Information theory: A tutorial introduction (First edition). Sebtel Press.

Sudo, H. (2019). [Research Reliability Required in Academia]. Yakugaku Zasshi : Journal of the Pharmaceutical Society of Japan, 139(6), 891–898. https://doi.org/10.1248/yakushi.18-00193-4

United Nations National Assembly. (2015, September). *Transforming our World: The 2030 Agenda for Sustainable Development*. https://www.unfpa.org/resources/transforming-our-world-2030-agenda-sustainable-development

Valiya Veettil, A., & Mishra, A. k. (2020). Multiscale hydrological drought analysis: Role of climate, catchment and morphological variables and associated thresholds. Journal of Hydrology, 582, 124533. https://doi.org/10.1016/j.jhydrol.2019.124533

Wang, A. X., Tran, C., Desai, N., Lobell, D., & Ermon, S. (2018). Deep Transfer Learning for Crop Yield Prediction with Remote Sensing Data. Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies, 1–5. https://doi.org/10.1145/3209811.3212707

You, J., Li, X., Low, M., Lobell, D., & Ermon, S. (2018). Deep Gaussian Process for Crop Yield Prediction Based on Remote Sensing Data. 7.

Zindi. (2020). CGIAR Crop Yield Prediction Challenge. Zindi. Retrieved May 3, 2021, from https://zindi.africa/competitions/cgiar-crop-yield-prediction-challenge

Appendices

Appendix A: Data Pre-Processor Class

```
class PreProcessor(Dataset):
    def __init__(self, df, target, folder, transform=None):
        self.df = df
        self.length = len(df)
        self.transform = transform
        self.folder = folder
        self.target = target
    def __getitem__(self, index):
        img = np.load(f'{self.folder}/{self.df[index]}.npy').astype(np.float64)
        img.resize(360, 32, 32) # Reshaping the image to a 32* 32 px

        # if not self.transform:
        img = torch.from_numpy(img)
        return img, torch.tensor(self.target[index])
    def __len__(self):
        return self.length
```

Appendix B: The CNN Model

```
class CNN(nn.Module):
  def __init__(self):
    super().__init__()
    self.conv1 = nn.Conv2d(in_channels=360, out_channels=450, kernel_size=3, padding=1)
    self.conv2 = nn.Conv2d(in_channels= 450, out_channels = 200, kernel_size=3, padding=1)
    self.linear1 = nn.Linear(in_features=200*6*6, out_features=256)
    self.dropout1 = nn.Dropout(0.4)
    self.linear2 = nn.Linear(in_features=256, out_features=128)
    self.dropout2 = nn.Dropout(0.4)
    self.linear3 =  nn.Linear(in_features=128, out_features=64)
    self.dropout3 = nn.Dropout(0.4)
    self.out = nn.Linear(in_features=64, out_features=1)

  def forward(self, t):
    # Layer1
    t = self.conv1(t)
    t = F.relu(t)
    t = F.max_pool2d(t, kernel_size=4, stride=2)
    # Layer2
    t = self.conv2(t)
    t = F.relu(t)
    t = F.max_pool2d(t, kernel_size=5, stride=2)
    # Flatten and layer3
    t = t.reshape(-1, 200 * 6 * 6)
    t = self.linear1(t)
    t = F.relu(t)
    # Layer4
    t = self.dropout1(t)
    # Layer5
    t = self.linear2(t)
    t = F.relu(t)
    # Layer6
    t =self.dropout2(t)
    # Layer7
    t = self.linear3(t)
    t = F.relu(t)
    # Layer8
    t = self.dropout3(t)
    # Output
    return self.out(t)
```

Appendix C: Model Summary

```
CNN(
  (conv1): Conv2d(360, 450, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (conv2): Conv2d(450, 200, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (linear1): Linear(in_features=7200, out_features=256, bias=True)
  (dropout1): Dropout(p=0.4, inplace=False)
  (linear2): Linear(in_features=256, out_features=128, bias=True)
  (dropout2): Dropout(p=0.4, inplace=False)
  (linear3): Linear(in_features=128, out_features=64, bias=True)
  (dropout3): Dropout(p=0.4, inplace=False)
  (out): Linear(in_features=64, out_features=1, bias=True)
)
```

Appendix D: Model Training Function

```
def ModelTrainer(num_epochs, optimizer, model, train_loader, device):
    loss_vals=  []
    for epoch in range(num_epochs):
        epoch_loss= []
        total_epoch_loss = 0
        for i, (images, labels) in enumerate(train_loader):
            images = images.to(device)
            labels = labels.to(device)
            # Forward pass
            outputs = model(images)
            # Calculate Loss
            mse_loss = nn.MSELoss()
            loss = mse_loss(outputs.reshape(labels.shape[0]), labels)
            # Backward and optimize
            optimizer.zero_grad()
            # Calculate Weights
            loss.backward()
            total_epoch_loss += loss.item()
            epoch_loss.append(loss.item())
            # Update Weights
            optimizer.step()

            if (i+1) % 2000 == 0:
                print (f'Epoch [{epoch+1}/{num_epochs}], Step [{i+1}/{n_total_steps}], Loss: {loss.item():.4f}')

        loss_vals.append(sum(epoch_loss)/len(epoch_loss))
        print("epoch:", epoch,  "loss:", total_epoch_loss/len(train_loader))


    PATH = './cnn.pth'
    torch.save(model.state_dict(), PATH)
    plt.plot(np.linspace(1, num_epochs, num_epochs).astype(int), loss_vals)
```

Appendix E: Model Validation Function

```python
def ModelValidator (model, batch_size, testloader, device='cpu'):
    with torch.no_grad():
        n_correct = 0
        n_samples = 0

        for images, labels in testloader:
            images = images.to(device)
            labels = labels.to(device)
            outputs = model(images) # Pass Batch
            # max returns (value ,index)
            _, predicted = torch.max(outputs, 1)
            n_samples += labels.size(0)
            n_correct += (predicted == labels).sum().item()

        acc = 100.0 * n_correct / n_samples
        print(f'Accuracy of the network: {acc} %')
        return acc
```

Appendix F: Model Testing Function

```
import pandas as pd
import torch
import torch.optim as optim
from torch.utils.data import DataLoader
from utils import CropYieldDataLoader
from cnn import CNN


# Testing
ss = pd.read_csv('data/test_sample.csv')
test_set  = CropYieldDataLoader(
    ss['Field_ID'].reset_index(drop=True),
    ss['Yield'].reset_index(drop=True),
    'data/image_arrays_test'
)
test_loader = DataLoader(test_set, len(ss))

device = 'cpu'
model = CNN()
model.double()
optimizer = optim.Adam(model.parameters(), lr=0.01)
model.to(device)
model.load_state_dict(torch.load('model.cnn-best.pth'))

for each in test_loader:
    images, values = each
    images = images.to(device)
    values = values.to(device)
    #optimizer = optimizer
    test_values = model(images) # Pass Batch


test_values = test_values.cpu()
ss['Yield'] = test_values.detach().numpy()
ss.to_csv('test_results.csv', index=False)
```

Appendix G: Image Visualizing Function

```
from matplotlib import pyplot as plt
def Visualizer(imageset, folder = 'data/image_arrays_train'):
    # Look at a sample:
    fid = imageset['Field_ID'].sample().values[0]

    # File name based on Field_ID
    fn = f'{folder}/{fid}.npy'

    # Loading the data with numpy
    arr = np.load(fn)

    # Combine three bands for viewing
    rgb_jan = np.stack([arr[4], arr[3], arr[2]], axis=-1)

    # Scale band values to (0, 1) for easy image display
    rgb_jan = rgb_jan / np.max(rgb_jan)

    # View with matplotlib
    plt.imshow(rgb_jan)


    # View false colour images from each month in the year:
    fig, axs = plt.subplots(3, 4, figsize=(12, 8), facecolor='w', edgecolor='k')
    fig.subplots_adjust(hspace = .5, wspace=.001)
    axs = axs.ravel()
    for i in range(12):
        # False colour (band 8, 4 and 3)
        rgb = np.stack([arr[i*30 + 8], arr[i*30 + 4], arr[i*30 + 3]], axis=-1)
        # Scaling consistently
        rgb = rgb / 4000
        # View with matplotlib
        axs[i].imshow(rgb.clip(0, 1))
        axs[i].set_title(str(i+1))

    plt.show()
```

Appendix H: Sentinel 2A Spectra Bands

Band	Scale	Description
B1	0.0001	443.9nm (S2A) / 442.3nm (S2B) Aerosols
B2	0.0001	496.6nm (S2A) / 492.1nm (S2B) Blue
B3	0.0001	560nm (S2A) / 559nm (S2B) Green
B4	0.0001	664.5nm (S2A) / 665nm (S2B) Red
B5	0.0001	703.9nm (S2A) / 703.8nm (S2B) Red Edge 1
B6	0.0001	740.2nm (S2A) / 739.1nm (S2B) Red Edge 2
B7	0.0001	782.5nm (S2A) / 779.7nm (S2B) Red Edge 3
B8	0.0001	835.1nm (S2A) / 833nm (S2B) NIR
B8A	0.0001	864.8nm (S2A) / 864nm (S2B) Red Edge 4
B9	0.0001	945nm (S2A) / 943.2nm (S2B) Water vapor
B10	0.0001	1373.5nm (S2A) / 1376.9nm (S2B) Cirrus
B11	0.0001	1613.7nm (S2A) / 1610.4nm (S2B) SWIR 1
B12	0.0001	2202.4nm (S2A) / 2185.7nm (S2B) SWIR 2
QA10		Always empty
QA20		Always empty
QA60		Cloud mask

Appendix I: TERRACLIM Spectra Bands

Name	Units	Min	Max	Scale	Description
aet	mm	0*	3140*	0.1	Actual evapotranspiration, derived using a one-dimensional soil water balance model
def	mm	0*	4548*	0.1	Climate water deficit, derived using a one-dimensional soil water balance model
pdsi		-4317*	3418*	0.01	Palmer Drought Severity Index
pet	mm	0*	4548*	0.1	eference evapotranspiration (ASCE Penman-Montieth
pr	mm	0*	7245*		Precipitation accumulation
ro	mm	0*	12560*		Runoff, derived using a one-dimensional soil water balance model
soil	mm	0*	8882*	0.1	Soil moisture, derived using a one-dimensional soil water balance model
srad	W/m^2	0*	5477*	0.1	Downward surface shortwave radiation
swe	mm	0*	32767*		now water equivalent, derived using a one-dimensiona soil water balance model
tmmn	°C	-770*	387*	0.1	Minimum temperature
tmmx	°C	-670*	576*	0.1	Maximum temperature
vap	kPa	0*	14749*	0.001	Vapor pressure
vpd	kPa	0*	1113*	0.01	Vapor pressure deficit
vs	m/s	0*	2923*	0.01	Wind-speed at 10m

* estimated min or max value

Appendix J: Similarity Index Report

Curiginal

Document Information

Analyzed document	A Computer Vision Based Model for Crop Yield Prediction using Remote Sensing Data.docx (D103714729)
Submitted	5/4/2021 1:33:00 AM
Submitted by	
Submitter email	kiragu.daniel@strathmore.edu
Similarity	1%
Analysis address	library.strath@analysis.urkund.com

Sources included in the report

W	URL: https://ir.lib.uth.gr/xmlui/bitstream/handle/11615/52040/20161.pdf?sequence=1 Fetched: 5/3/2020 6:04:03 AM	1
W	URL: https://zindi.africa/competitions/cgiar-crop-yield-prediction-challenge Fetched: 5/4/2021 1:34:00 AM	1
SA	**CRG Project Technical Details_2021.pdf** Document CRG Project Technical Details_2021.pdf (D98601573)	1
SA	**TCNN_prediction.pdf** Document TCNN_prediction.pdf (D95665946)	1
W	URL: https://www.researchgate.net/publication/344560876_Design_And_Implementation_Of_Cr ... Fetched: 5/4/2021 1:34:00 AM	1
W	URL: https://deepai.org/publication/estimating-crop-yields-with-remote-sensing-and-deep ... Fetched: 5/4/2021 1:34:00 AM	1

Appendix K: Ethical Review Approval

Strathmore UNIVERSITY

15th October 2021

Mr Mburu Daniel,
danielmburu674@gmail.com

Dear Mr Mburu,

RE: A Tool for Crop Yield Prediction in Precision Agriculture using Uniform Manifold Approximation and Projection

This is to inform you that SU-IERC has reviewed and **approved** your above **SU- master's** research proposal. Your application reference number is **SU-IERC0926/20.** The approval period is **15th October 2021 to 14th October 2022.**

This approval is subject to compliance with the following requirements:

i. Only approved documents including (informed consents, study instruments, MTA) will be used
ii. All changes including (amendments, deviations, and violations) are submitted for review and approval by SU-IERC.
iii. Death and life-threatening problems and serious adverse events or unexpected adverse events whether related or unrelated to the study must be reported to SU-IERC within 48 hours of notification
iv. Any changes, anticipated or otherwise that may increase the risks or affected safety or welfare of study participants and others or affect the integrity of the research must be reported to SU-IERC within 48 hours
v. Clearance for export of biological specimens must be obtained from relevant institutions.
vi. Submission of a request for renewal of approval at least 60 days prior to expiry of the approval period. Attach a comprehensive progress report to support the renewal.
vii. Submission of an executive summary report within 90 days upon completion of the study to SU-IERC.

Prior to commencing your study, you will be expected to obtain a research license from National Commission for Science, Technology and Innovation (NACOSTI) https://research-portal.nacosti.go.ke/ and also obtain other clearances needed

Yours sincerely,

for: Prof Fred Were,
Chairperson; SU-IERC

STRATHMORE UNIVERSITY INSTITUTIONAL ETHICS REVIEW COMMITTEE (SU-IERC)
15 Oct 2021
TEL: +254 (0)70303400
P.O BOX 59857-00200
NAIROBI-KENYA

Ole Sangale Rd, Madaraka Estate. PO Box 59857-00200, Nairobi, Kenya. Tel +254 (0)703 034000
Email admissions@strathmore.edu www.strathmore.edu