Vegetation Index Based Crop Yield Prediction Model Using Convolution Neural Network: A Case Study of Kenya

By

Judith Chepngetich

114770

A Dissertation Submitted to the Faculty of Information Technology in partial fulfillment of the requirements for the award of Master of Science in Information Technology (MSc.IT) at Strathmore University

Faculty of Information Technology

Strathmore University

June 2020

Declaration and Approval

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.

Judith Chepngetich

[Signature]

June 2020

Approval

The thesis of Judith Chepngetich was reviewed and approved by the following:

Dr. Joseph Orero,

Dean, Faculty of Information Technology,

Strathmore University

Dr. Bernard Shibwabo,

Director of Graduate Studies,

Strathmore University

Abstract

Predicting the crop yield is a vital food security strategy that can help a country take suitable measures and come up with policies that will help in crop production management. Such predictions will also support the farmers and industries involved in crop production for strategizing the logistics of their business or farming activities. Having sufficient production plans can improve food sufficiency and avoid situations of food emergencies. Climate change has had a huge impact on food production with variations in crop yields, creating uncertainty. Most of the studies on crop yield prediction have been done based solely on weather data, which is sometimes inaccurate due to scarcity of weather information especially in developing countries, where there is poor record keeping and insufficient resources to collect data. The use of vegetation indices derived from remote sensing data overcomes these challenges by providing data that is easily accessible and gives a comprehensive and multidimensional analysis.

This study proposes a model that uses of vegetation index to predict crop yield using machine learning. Data from past crop yields in Kenya and vegetation greenness indices were the inputs applied to the algorithms. Various machine learning algorithms were applied and thereafter evaluated, so as to select the algorithm that gives better accuracy. To determine the accuracy level for the prediction model, the RMSE is calculated to compare actual and predicted values. The RMSE values obtained using convolution neural network for the three crops maize, rice and wheat were lower compared to those obtained using ridge regression, so it was selected as the optimal algorithm for the crop yield prediction model.

Table of Contents

List of Figures

List of Tables

List of Equations

List of Acronyms

ANN	-	Artificial Neural Networks
CNN	-	Convolutional Neural Networks
DNN	-	Deep Neural Network
EVI	-	Enhanced Vegetation Index
GIS	-	Geographical Information Systems
MAPE	-	Mean Absolute Percentage Error
MODIS	-	Moderate Resolution Imaging Spectroradiometer
NASA	-	National Aeronautics and Space Administration
NASS	-	National Agricultural Statistics Service
NDVI	-	Normalized Difference Vegetation Index
RF	-	Random Forest
SDAP	-	Severe Drought Area Prediction
SMI	-	Soil Moisture Index
SVM	-	Support Vector Machine
USDA	-	United States Department of Agriculture

Chapter 1: Introduction

1.1 Background

Uncertainty in crop yields is increasing because of change in climatic patterns, mainly attributed to the effects of climate change. Effective forecasting of crop yields is an important aspect of an agricultural economy. It provides a basis for advance planning, formulation and implementation of policies that involve vital decisions on crop procurement, distribution, price structure, import export decisions and timely resource allocation. A forecast also helps farmers decide in advance their future prospects and enables them take appropriate action in a timely manner. An estimation or prediction of crop yield is a complexity of various factors that affect a crop yield, mainly weather variables and agricultural input (Goyal, 2016).

Crop yield forecasting methods have evolved over time. A report by Hanuschak (2013) for the Food and Agricultural Organization (FAO) points out that initially, crop yield forecasting methods were characterized by traditional procedures known as crop cutting, which were based on statistical sampling methods. Crop cutting has been adopted by many countries and is still in use today with some revisions and improvement. These early methods were initially focused on final crop yield estimation at or near harvest, but have evolved over time into advance crop yield forecasting, giving estimates several months before a harvest.

Craig and Atkinson (2013) focused their report on crop area estimation, which is also a key determinant of crop production forecast. They discuss the technological advances that have been experienced in crop area estimation, which include remote sensing, computer processing and Geographical Information Systems (GIS). Ferencz et al.(2004) have also presented two ways of using satellite remote sensing data for crop yield estimation.

Greenness vegetation indices are algorithms derived from satellite remote sensing data, particularly spectral bands. These indices measure the concentration of green vegetation over an area, hence provide very useful information on plant health. Using vegetation indices provides a simple and reliable source of input for crop yield prediction since the data is readily available, as opposed to collecting weather data on the ground. Vegetation indices have also proven to have a good correlation with crop yield and have produced high precision models that have been used in crop yield prediction (Johnson, 2014).

1.2 Problem Statement

According to Lopez, Baruth, El Aydam, Eric and Garcia (2015) there has been an increase in world demand for agricultural products due to fluctuations of global production mostly caused by climate change, eventually leading to price volatility in agricultural markets and social unrest in some countries. This highlights the importance of crop monitoring and yield forecasting in anticipating production variations and in making well-informed and timely decisions or policies. In addition, crop yield prediction can help avert food crises and market disruptions by reducing speculation and contributing to overall increased food security.

Climate change has an intense effect in the developing world because they are more vulnerable to the effects of climate change since they rely primarily on natural resource dependent income sources for their livelihoods, especially agriculture. They also have few resources with which to adapt to the anticipated change in climatic patterns. Additionally, poor planning and management may lead to delay in response to changes in climatic conditions which could lead to economic and social damage (Kabubo-Mariara & Millicent, 2015).

Prediction of crop yields is important to farmers because it can aid in planning and management of variability in crop production due to climate change. It also enables them make informed financial decisions. Existing time series methods do not account for these cyclic changes, therefore do not account for variations. In addition, existing studies using machine learning techniques have focused on only one crop. Most of the studies on prediction have also been done based solely on weather data, without use of remote sensing data that provide data on the go and give a comprehensive and multidimensional analysis. This study will focus on building a crop yield prediction model by applying two machine learning algorithms and selecting the best, considering the past crop yield patterns and incorporating remote sensing data derived from satellite images.

1.3 Aim

The purpose of this research is to develop a crop yield prediction model using a machine learning algorithm, which will be used to predict food crop yield production.

1.4 Research Objectives

i. To review the challenges associated with existing methods for predicting crop yield.
ii. To analyze machine learning algorithms that can be used for building a prediction model.
iii. To develop a model for crop yield prediction.
iv. To perform testing and validation of the model developed.

1.5 Research Questions

i. What are the current methods used to forecast crop yields and their limitations?
ii. What machine learning algorithms are going to be used in developing the prediction model?
iii. How will the model be developed?
iv. How can the model be validated?

1.6 Justification

Ending hunger, achieving food security and improved nutrition and promoting sustainable agriculture, also known as SDG2, is one of the Sustainable Development Goals that has been highlighted to contribute to a sustainable future for all. To achieve this, measures have to be taken that work towards achieving this goal, including using technology to develop systems that can enhance food security and sustainability. One of the targets of SDG 2 is to adopt measures that ensure efficient operation of food commodity markets, which will help reduce price volatility (United Nations, 2015).

This study will contribute to the food security agenda through development of a model that will be used to predict crop yield production using vegetative greenness data, thus eliminating the uncertainty in crop production. If the crop yield is accurately predicted, farmers will have better response to climatic patterns change, thus making a crucial step towards attaining food security. In addition, there will be better management of food production, especially for staple food like maize which is normally prone to price volatility and fluctuations.

1.7 Scope and Limitation

The area of focus in this study is Kenya because of the fluctuating food production in recent years. It will also focus on the areas that are normally involved in agriculture as their source of income, considering foods produced locally.

Chapter 2: Literature Review

2.1 Introduction

In this chapter, various methods used to forecast the crop yield are explained. Studies that have been done on use of machine learning algorithms to predict the crop yields are also presented in this chapter. The literature is then summarized and contribution of this thesis discussed.

2.2 Crop Yield Estimation in Kenya

Agriculture plays a major role in the Kenyan economy, accounting for 26% of the country's GDP (FAO, 2020). Small scale farmers produce the bulk of agricultural produce with a production of 63% of the food produced in the country. Most of the small scale farmers rely on rain-fed farming systems, hence hugely affected by erratic weather patterns caused by climate change (FAO, 2015). Various methods of estimating crop yield have been adapted in the past by farmers where the yield estimates of the farms are determined in advance. The two major traditional methods used for crop yield estimation are crop cuts and farmer recall (Fermont and Benson, 2011). As explained by Fermont and Benson (2011), the crop cut method involves sampling small areas of the crop field known as subplots which are staked early in the season. The estimation is then done by dividing the production of the subplot by its area. This is done for several subplots and the average taken to be the yield. The farmer recall method is done after harvest and involves carrying out a survey on farmers' harvest and the yield is determined as the ratio of production to the area harvested. The drawback of these methods is that they are conducted during harvest or after harvest, hence do not predict crop yield in advance.

2.3 Crop Yield Prediction using Weather Data

Weather data has been the preferred parameter for crop yield prediction in many studies because weather variability has a direct impact on crop yield. Weather data also has the potential to provide reliable predictions. Crop yield prediction models based on weather data have been developed using statistical methods as well as machine learning methods. A weather based model was developed by Laxmi and Amrender (2011) for crop yield prediction in India, which took in maximum and minimum temperatures, rainfall and morning relative humidity as input variables for a Multilayer Perceptron (MLP) based neural network. Khaki and Wang (2019) used weather data in addition to soil data to predict crop yield using deep neural network (DNN). Several variables were subjected to feature selection in order to identify the weather variables that were

important to crop yield prediction. It emerged that solar ration, precipitation and temperature were the more important variables that gave better accuracy in the DNN model. The researchers in the study concluded that it was possible to make accurate predictions on crop yield using known weather conditions. The limitations of the above models is the sole reliance on weather data, which may be incomplete and might lead to misleading predictions.

2.4 Crop Yield Prediction using Greenness Satellite data

Technological advancements have ushered in new methods of crop yield estimation. Over the years, studies have revealed a shift to the use of satellite data and agro-metrological data to estimate crop yield (Rojas, 2007).The use of greenness vegetation indices for crop yield prediction has gained popularity in recent studies because satellite data has become easily available and accessible from various platforms. NDVI and EVI are the most widely used indices for crop yield predction. Petersen (2018) highlighted the importance of using vegetation indices in developing countries where weather data is scarce and unreliable.The study focuses on on crop yield prediction using vegetation health indices. (Mohamad (2019) created a new optimization model that was to compensate for the low resolution of satellite images caused by climate conditions. The new model was to improve on precision of crop yield estimation, hence an equation on energy balance was included. A comparison between actual production collected from farmers and the estimated crop yield based on the new model was done, and it proved that the model was efficient and sufficient to be used for prediction.

After calculation of the mean absolute percentage error (MAPE) for the estimated crop yield and actual crop yield, the error was less than 4%. Root-mean-square error (RMSE) was also used as a validation tool in the study to estimate the accuracy of the results, and the value obtained for RMSE was 0.039, which indicated that the developed model was of high accuracy. Figure 2.1: Absolute Errors for Farmers below shows the absolute errors of a sample of 15 farmers used to verify the results of the model.

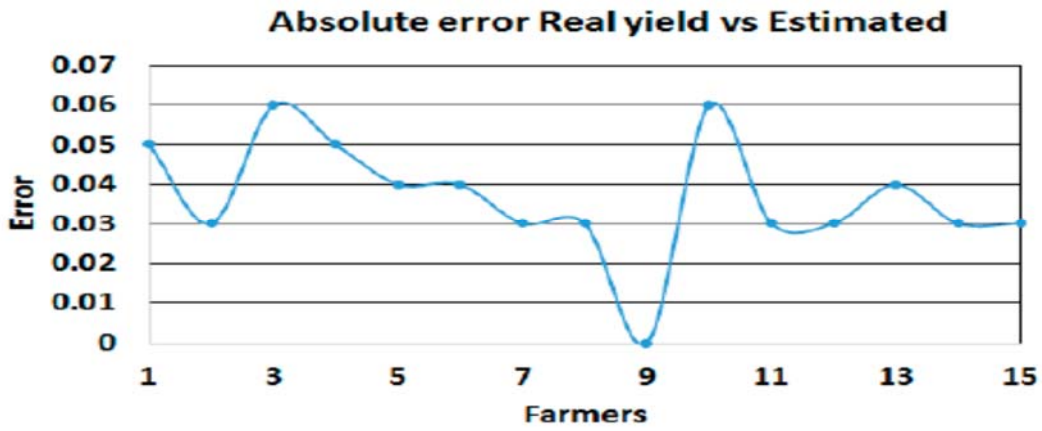


Figure 2.1: Absolute Errors for Farmers (Mohamad, 2019)

2.5 Machine Learning Algorithms for Prediction

Crop yield prediction is a prediction problem that seeks to utilize the prediction algorithms in machine learning (Ethem, 2010). The algorithms that can be used for this study are described in detail in this section, highlighting their strengths and weaknesses.

2.5.1 Neural Networks for Crop Yield Prediction

Artificial neural networks are a branch of artificial intelligence algorithms (Al-Sammarraie, Al-Mayali, & El-Ebiary, 2018). An Artificial Neural Network (ANN) mimics the working of the biological nervous system, such as the brain and the processing of information. The model of the information processing system is composed of a large number of highly interconnected processing elements known as neurons that work together to solve specific problems. Since ANNs learn by example, they are suitable for a specific application, such as pattern recognition or data classification. ANN learns by making adjustments to the synaptic connections that exist between the neurons after calculations are done (Siganos & Stergiou, 1996). **Error! Reference source not found.**2 below illustrates a neural network.

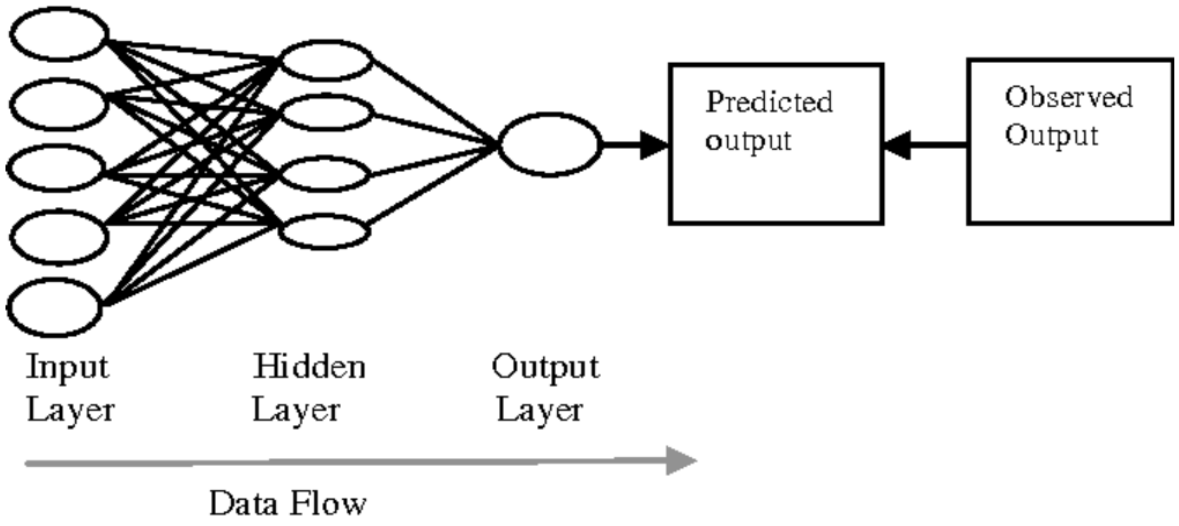


Figure 2.2: A Neural Network

Multi-layer perceptron (MLP) forms one type of artificial neural network. There is also a class of Single-layer perceptron ANN. In addition, there are Radial basis function networks and a whole group of recurrent/feedback networks (Gardner & Dorlinga, 1998).

Manjula and Narsimha (2016) adopted an Optimal Neural Network classifier (ONN) in spatial data mining to predict crop yield in India. Spatial data mining is defined in this study as "the process of extracting interesting knowledge from spatial databases". The prediction method underwent three major processes; preprocessing, feature reduction and prediction. Multilinear principal component analysis (MPCA) was used during the feature reduction phase. The prediction method was validated using prediction error and accuracy value. The study concluded that there is need to adopt technology to improve on food production and agriculture.

2.5.1.1 Crop Yield Prediction using Convolutional Neural Network

Priyanka, Soni and Malathy (2006) have highlighted Convolutional neural networks (CNNs, or ConvNets) as an essential tool in artificial neural networks and data mining. CNN is especially suited for image classification and pattern recognition data. In their study, CNN is used to detect the images, whereby an unknown image is given as input and the CNN is used to identify or classify this unknown image under different categories. This image is then divided into different pixels, with each pixel being identified as a receptive field. The receptive field was used to process

the portions of input image, resulting in a combined output for better representation of the image. This process was performed for each layer in the CNN.

You et al. (2017) introduced a new approach to Convolution Neural Network by incorporating a Gaussian process that would remove spatio-temporal errors and ultimately improve the accuracy of the algorithm. They further compared the performance of the algorithm with that of baseline methods like regression, decision tree and Long short-term memory (LSTM). The Gaussian based CNN proved to be more accurate since it had a 30 percent reduction in RMSE compared to the other methods. Table 2.1: RMSE Using Various Methods (You et al. 2017) below shows the RMSE of the various methods used.

Table 2.1: RMSE Using Various Methods (You et al. 2017)

	Baselines			Deep models			
Year	Ridge	Tree	DNN	LSTM	LSTM + GP	CNN	CNN + GP
2011	9.00	7.98	9.97	5.83	5.77	5.76	**5.7**
2012	6.95	7.40	7.58	6.22	6.23	5.91	**5.68**
2013	7.31	8.13	9.20	6.39	5.96	**5.50**	5.83
2014	8.46	7.50	7.66	6.42	5.70	5.27	**4.89**
2015	8.10	7.64	7.19	6.47	**5.49**	6.40	5.67
Avg	7.96	7.73	8.32	6.27	5.83	5.77	**5.55**

2.5.1.2 Using Deep Learning for Crop Yield Estimation

Kuwataa and Shibasaki (2016) presented a study that focused on using deep learning for crop estimation in the United States. The deep neural network estimated county-level corn yield for the entire country with a better accuracy that had not been achieved with other methods. The deep neural network developed had six hidden layers with 4000 neurons in each layer. The input data was composed of annual county level corn yield data, satellite based vegetative index data from MODIS EVI and gridded estimates of weather patterns for the country. In addition, a comparison was made with support vector machine and autoencoder algorithms to determine the one with a better level of accuracy.

The MODIS EVI data was then smoothed using wavelength shrinkage method in order to achieve data compression and signal de-noising using equation (2.1) and (2.2) respectively.

$$W\,f(x) = \int_{-\infty}^{+\infty} f(x) 1/\sqrt{a}\;\; \varphi(\frac{a-b}{a})\, dx \tag{2.1}$$

$$\lambda = \sqrt[\sigma]{2\log n} \tag{2.2}$$

φ represents a mother wavelet function, a a scaling parameter, and b a shifting parameter.

The result of the smoothed data obtained for the entire country was as shown in Figure 2.33.

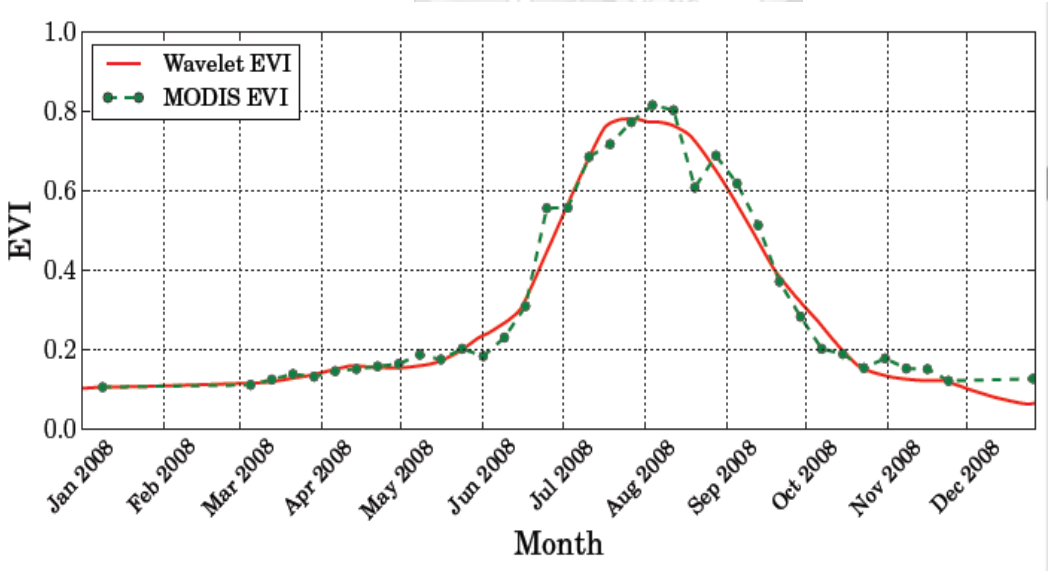


Figure 2.3: MODIS EVI and EVI Smoothed by the Wavelet Transform (Kuwataa & Shibasaki, 2016).

The three selected algorithms were then evaluated based on root mean square error (RMSE) and the coefficient of determination (R^2) and the estimation results were as shown in

Table *2.22*.

Table 2.2: Results of the Three Models (Kuwataa & Shibasaki, 2016).

	Daily input dataset		5-day accumulation dataset	
	RM SE	R^2	*RM SE*	R^2
SVM	20.4	0.727	17.7	0.792
Autoencoder	19.0	0.759	21.3	0.700
DNN (six hidden layers)	18.5	0.773	18.2	0.780

The results indicated that SVM had better accuracy with the 5-day accumulation dataset compared to the daily input set. However, for the autoencoder algorithm, better accuracy was achieved with the daily input dataset compared to the 5-day dataset. DNN achieved better results than SVM on the daily input dataset, and it was proposed as the most suitable algorithm for crop yield prediction since it extracted features better from dimensional input data.

2.5.2 Random Forest for Drought Areas Prediction

Haekyung, Kyungmin and Lee (2019) conducted a study to predict severe drought areas using random forest. Since most of the availale methods did not consider the uncertainty of the precipitation forecasts and spatial information, they designed a model called the severe drought area prediction (SDAP) model that would estimate soil moisture index (SMI) maps for several months using surface factors. The variables were all surface factors derived from remote sensing data, specifically related to soil moisture so that they could obtain spatial information of agricultural drought.

Haekyung, Kyungmin and Lee (2019) came up with 15 land surface factors that would be used as input variables in designing the model, and would undergo regression to the soil moisture index

using the random forest (RF) algorithm. They generated the drought function with the RF algorithm using the identified input variables.

Machine learning was performed using 62,500 data samples from the training area to generate the drought function f(x). The samples were split into two, the training data and the test data. The training data was 75% of the sample while the test data was 25%. The Random Forest Regressor python function of the scikit-learn library for machine learning was then applied. After tuning the coefficient of determination of the training data and test data, an optimal max_depth of 14 was found.

The performance of the drought function was verified using root mean square error (RMSE), normalized RMSE (NRMSE), mean absolute error (MAE) and R2.The results are displayed in Table 2.33 below:

Table 2.3: Evaluation of the Training Performance of the Drought Function (Haekyung, Kyungmin, & Lee, 2019).

RMSE	**NRMSE**	**MAE**	**R2**	**Max. SMI**	**Min. SMI**	**Max. Error**
0.05294	5.29%	0.03980	0.91	0.97940	0.05684	0.30352

2.5.3 KNN

K-Nearest Neighbour is a classification method that assigns input to the nearest neighbour based on similarity. Similarity is determined by calculating the Euclidean distance between neighbours, after which the new instance is assigned to the nearest cluster (Noronha, Divya, & Shruthi, 2016). The distance between two points in a plane with coordinates (x, y) and (a, b) is given by equation(2.3 below.

$$dist((x,y),(a,b)) = \sqrt{(x-a)^2 + (y-b)^2} \quad (2.3)$$

A study was done by Pavani and Beulet (2019) on the use of KNN for crop yield prediction in India.The model performed well in prediction of crop yield with a negligible difference between predicted and actual yield.

2.5.4 Ridge Regression

Ridge regression is a machine learning technique mainly used to overcome multicolinear problems. This technique reduces complexity of a model through coefficient shrinkage, hence suitable for high dimension settings (Ishwaran & Rao, 2014). Coefficient shrinkage is achieved by changing the hyperparameter alpha, in such a way that the higher the value of alpha, the lower the coefficients of ridge regression. The cost function of ridge regression is as shown in equation $min\| Y - X(\theta) \|^2 + \lambda \| (\theta) \|^2$

(2.4.

$$min \| Y - X(\theta) \|^2 + \lambda \| (\theta) \|^2 \quad (2.4)$$

Hernandez et al. (2015) developed a model for predicting crop yields using ridge regression, and they were able to estimate wheat yields under different irrigation conditions and growth stages.

2.5.5 Support Vector Machine (SVM)

Support vector machine is a classification technique which classifies data into different class labels, while having the ability to support both regression and classification tasks (Porchilambi & Sumitra, 2019). It works by creating a hyperplane that separates features of one class from another. Support vector machine has been used in crop yield prediction because of its simplicity, less susceptibility to overfitting and ability to use fewer parameters compared to other methods (Kodimalar & Surianarayanan, 2019). Figure 2.44 below provides an illustration of a support vector machine.

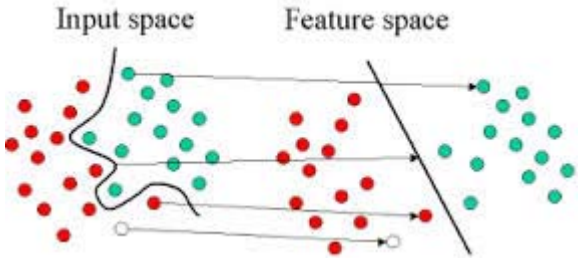


Figure 2.4: Support Vector Machine

A model was built by Ying-xue, Huan and Li-jiao (2017) to predict rice production in China using support vector machine. They came up with a yield module that was fed with soil data and meterological dat for types of rice, which in turn predicted rice yield for a given year. Different kernels and hyperparameters were used with five cross validation in order to determine the optimal kernel functions and hyperparameters that would give the least RMSE.They concluded that SVM could be used successfully for yield prediction and could be simulated in other regions as well.

2.5.6 Comparison of Various Learning Methods

A comparative study was done by Kumar, Kumar, and Vats (2018) on three machine learning algorithms namely K-Nearest Neighbor, Support Vector Machine and Least Squared Support Vector Machine to determine which one would give the lowest error rate and highest accuracy in crop yield prediction. In order to handle the issue of decision boundary and support vectors, (Kumar, Kumar, & Vats, 2018) decided to use a hyperplane so that they could obtain a straight line.

Linear square support vector machine, an extension of support vector machine was also chosen to handle the dataset that was complex and large, and because LSSVM works in ranges as opposed to data points.

The classifiers were evaluated through cross validation, Mean Squared Error (MSE) and Average Mean Squared Error (AMSE) and the results were as shown in Table 2.44, Figure 2.55 and Figure 2.6: Average Mean Squared Error of All Classifiers. respectively.

Table 2.4: Validation Error of Classifiers at Different Cross Validation Runs (Kumar, Kumar, & Vats, 2018).

Cross Validation	KNN	SVM	LS-SVM
1	0.307428	0.147108	0.0319429
2	0.33877	0.109105	0.0273692
3	0.342234	0.124982	0.0258275
4	0.32295	0.110263	0.0234979
5	0.32094	0.116354	0.0374257
6	0.323151	0.141801	0.0406318
7	0.313512	0.111455	0.0498942
8	0.331188	0.148668	0.0371499
9	0.344364	0.113936	0.0484682
10	0.304701	0.135623	0.0271545

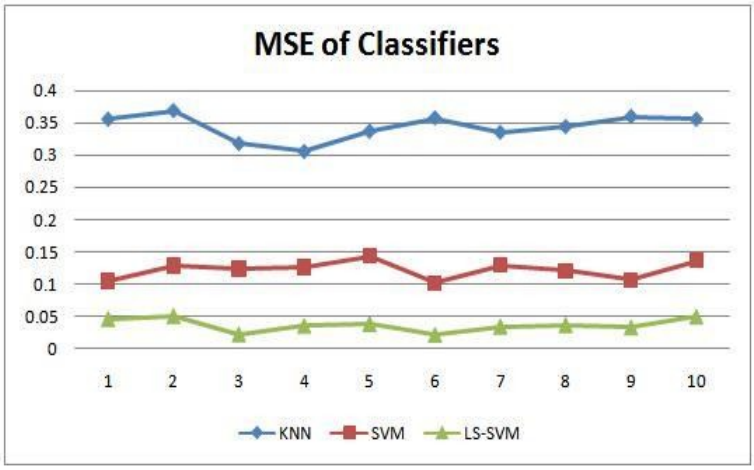


Figure 2.5: Mean Squared Error of All Classifiers (Kumar, Kumar, & Vats, 2018)

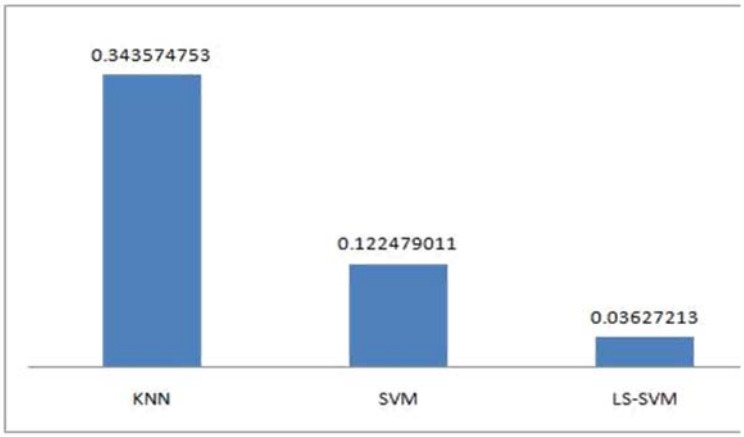


Figure 2.6: Average Mean Squared Error of All Classifiers (Kumar, Kumar, & Vats, 2018).

The LSSVM classifier had a better accuracy and lower error rate compared to KNN and SVM.

2.6 Contribution of This Study

This study applies the concept of machine learning to food crop yield prediction. It provides an opportunity to apply machine learning techniques, particularly convolution neural network, using variables that are relevant to the area of study, and incorporating satellite data to build a crop yield prediction model. It also provides an opportunity for crop yield prediction to be done at farmer level. The methods to be used in this study are ridge regression and convolution neural network.

2.7 Conceptual Framework

Figure 2.7 below is a conceptual framework outlining how the study was undertaken. Raw data was obtained from the ministry of Agriculture, Livestock and Fisheries and National Aeronautics and Space Administration (NASA) for past crop yields and vegetation greenness indices respectively. The data was then subjected to cleaning to handle missing values and achieve consistency.

Thereafter, data was explored further using various exploration techniques to understand the relationship of variables, uncover characteristics and bring to surface points of interest in the data. The data was then split into training and test sets, whereby the training set was used to train the model and the test set to validate the model. Finally, the performance of the model was evaluated using statistical tools.

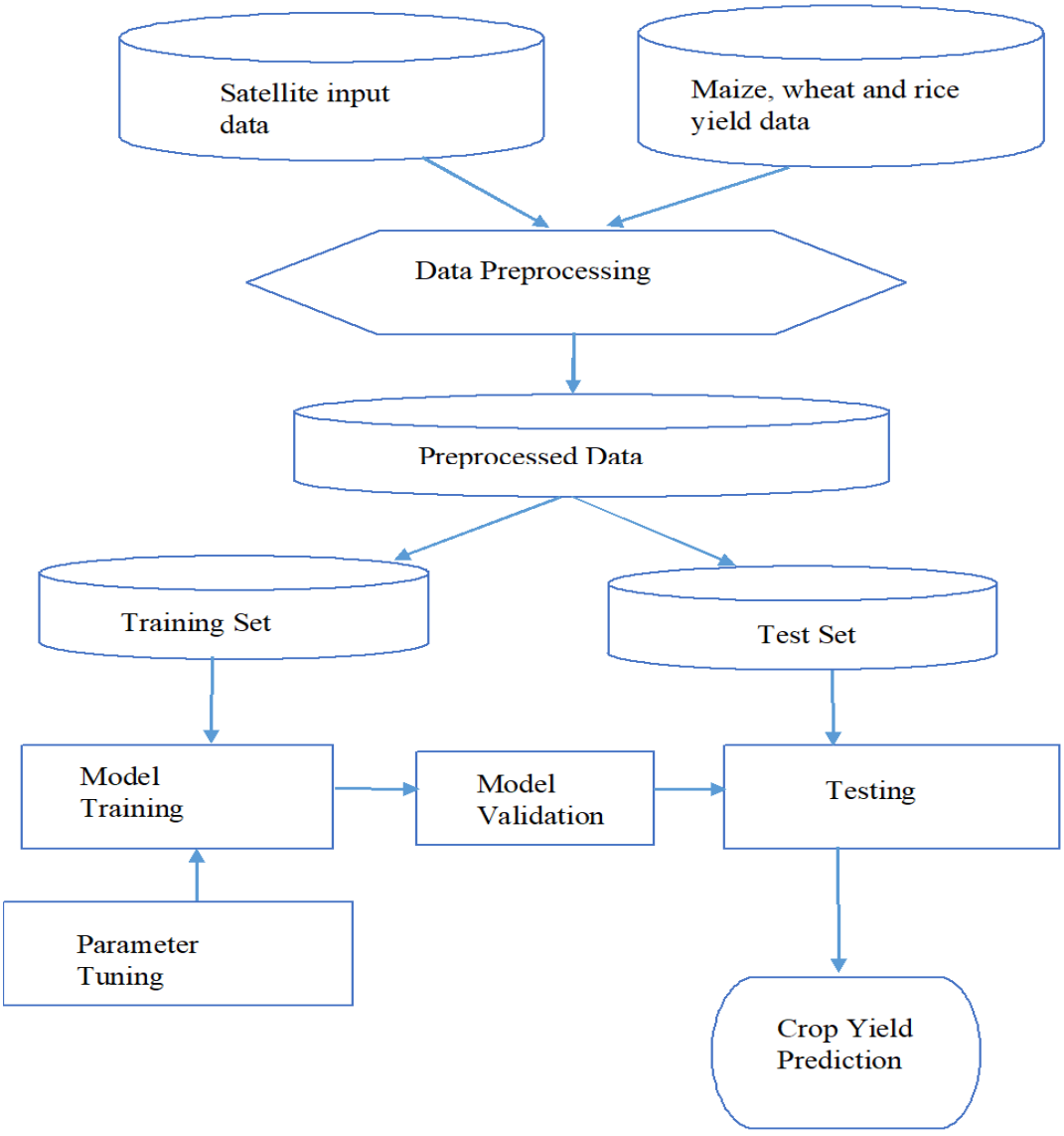


Figure 2.7: Conceptual Framework

Chapter 3: Research Methodology

3.1 Introduction

This chapter outlines the approach in whichthis study was carried out to meet the objectives set out in the research objectives. The methods of data collection, sampling and data analysis are also described in this chapter. The appropriate system development methodology for development of the prediction model is also discussed.

3.2 Research Design

Kothari (2004) defines research as an original contribution to the existing stock of knowledge making way for its advancement. He further describes it as a pursuit of truth with the help of study, observation, comparison and experiment, and is the search for knowledge through a systematic method of finding solution to a problem.

According to Luck and Rubin (2009), "A research design is the determination and statement of the general research approach or strategy adopted/or the particular project. It is the heart of planning. If the design adheres to the research objective, it will ensure that the client's needs will be served." This study adopted a quantitative research design, whereby secondary data sources were identified and data obtained from those sources. This was followed by data preprocessing and exploration. The study also met its stated objectives by reviewing studies that have been done on crop prediction using machine learning algorithms, while also highlighting their limitations. Two algorithms were explored and their performance evaluated to identify the best to be used to build the prediction model. Thereafter, a model for crop yield prediction was developed and subjected to testing and validation using the Root Mean Squared Error (RMSE).

3.3 Research Data

The study used secondary sources of data from satellite images that have been captured over the years which will assist to capture the vegetative greenness of an area. The Ministry of Agriculture online database was used to gather data on production yields for the past years to date.

3.3.1 Sampling Design

Samples of data were obtained from the identified online sources, where they were merged and cleaned for consistency. The data was then split into two sets, the training set that was fed as input to train the model, and the test data that was used to evaluate the models.

3.3.2 Data Preprocessing

Data was subject to cleaning so as to make the data suitable for training and developing the model. The raw data was checked for missing values, duplicates and errors before being split into the training and testing sets.

3.3.3 Feature Engineering

The data obtained from the secondary source was then subjected to feature engineering, which is a vital process where some features were removed to increase the prediction power and accuracy of the model being built. The raw data was transformed into features that better represent the research problem leading to more accuracy in the prediction model. The main features used were vegetation indices and water index.

3.4 System Development Methodology

This study adopted an iterative approach in development of the system whereby the model was trained continuously in iterations until a satisfactory level of performance was achieved. This comes with the advantage of flexibility in making changes to the model**,** other than a gated-step approach.

3.4.1 Phases of Iterative Development

IBM (2017) outlined the phases of iterative development for machine learning models as shown in Figure 3.1 below:

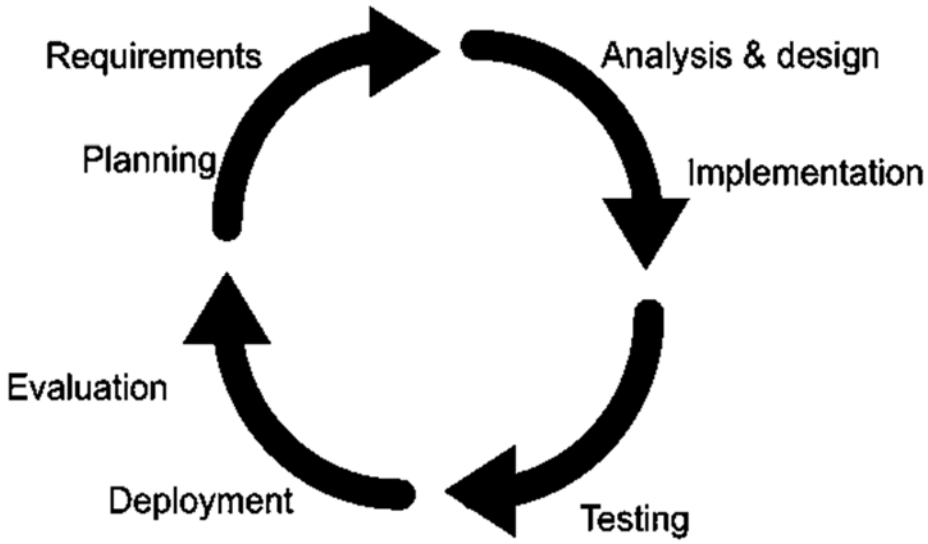


Figure 3.1 : Phases of Iterative Development (Adapted from IBM, 2017).

3.4.2 Planning Phase

This phase involves first understanding the problem and the requirements, so as to identify what sort of data is required. Data sources were identified and then extracted using extraction tools. The data was then cleaned to achieve consistency and reduce the possibility of errors. The tools used in building the model were also identified in this phase.

3.4.3 Analysis and Design Phase

Exploratory data analysis was performed in this phase so as to understand the data and features that are vital in achieving the objective of the study. Python was the main tool for visualization and any other statistical analysis. Use case diagram, partial domain diagram and a sequence diagrams were also used to model the proposed solution.

3.4.4 Implementation

Data was trained using two machine learning algorithms, convolution neural network and ridge regression. Python was the main programming language in use *with numpy*, *pandas* and *matplotlib.lyplot* libraries being the major libraries that provided a range of functions required. *Scikit-learn* library was also vital in performing regression.

3.4.5 Testing

The developed model was tested using the testing dataset from the split that was done after data preprocessing and cleaning. Data from the year 2012 to 2017 was used as the training data, while the data from the year 2018 was held out for testing.

3.5 Research Quality

Boaz and Ashby (2003) have outlined the dimensions of research to include quality and transparency in reporting so that the research can be appraised and used by others, whether the research was technically well executed, quality of the signal which seeks to find out whether the research answers policy and practice questions, and fitness for purpose of the research approach to the purpose of the study.

The test dataset was used to test the model once training had been done so as to determine the performance of the model. To compare various models, the Root Mean Squared Error (RMSE) was used as the metric.

RMSE is an evaluation metric used mostly in regression problems. The square root enables this metric to show large number deviations. RMSE was used to measure of the differences between values predicted by a model and the observed values. The equation is as shown below:

$$RMSErrors = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}} \quad (3.1)$$

3.6 Ethical Considerations

Data that was used for this study is data already available in the public domain since the sources are online databases available for free download. The main data sources for this study were the Ministry of Agriculture, Livestock and Fisheries online data portal and vegetation index data from NASA website. The data collected was only be used for the purpose of this study and was not shared with other parties.

Chapter 4: System Analysis, Design and Architecture

4.1 Introduction

The design and architecture of the crop yield prediction model are explored in this chapter. Firstly, the requirements for the system are outlined to ensure that the system designed operates as required. Thereafter the relationship between different objects are modelled using UML diagrams.

4.2 Requirements Analysis

4.2.1 Functional Requirements

i. The system should have the ability to recognize patterns from historical crop yield data and satellite data.
ii. It should also have the ability to predict crop yield using the model developed.
iii. The system should accept updates and additional input from the administrator in order to improve the model.

4.2.2 Non Functional Requirements

i. Performance-The developed model should have the ability to handle loads of data since training is done on a huge size of past crop yield data.
ii. Accuracy-The model should have a high level of accuracy so that it makes its predictions reliable and acceptable to the users.
iii. Reliability- The system should be available to the users at any given point in time and as required. This boosts the authenticity of the developed model.
iv. Maintainability- It should be easy to re-train the model and update the data when required so as to ensure that the accuracy of the model is improved with time.

4.3 System Design and Architecture

4.3.1 System Architecture

A system architecture is used to show the structure and behavior of various components of a system. It also provides a better understanding and view of how a system will meet the stated requirements and objectives. Figure 4.1: System Architecture below shows the architecture of the developed system. The prediction will be delivered by a mobile application, which picks the geolocation of a farmer located in a given area. Based on this location, the NDVI, EVI and NDWI

values are retrieved, which serve as input to the model. These retrieved values are also sent to the database after prediction so that the model can learn more from them. The model performs a prediction on the given input values then sends the predicted yield result to the mobile application. The farmer then receives the yield prediction on the application.

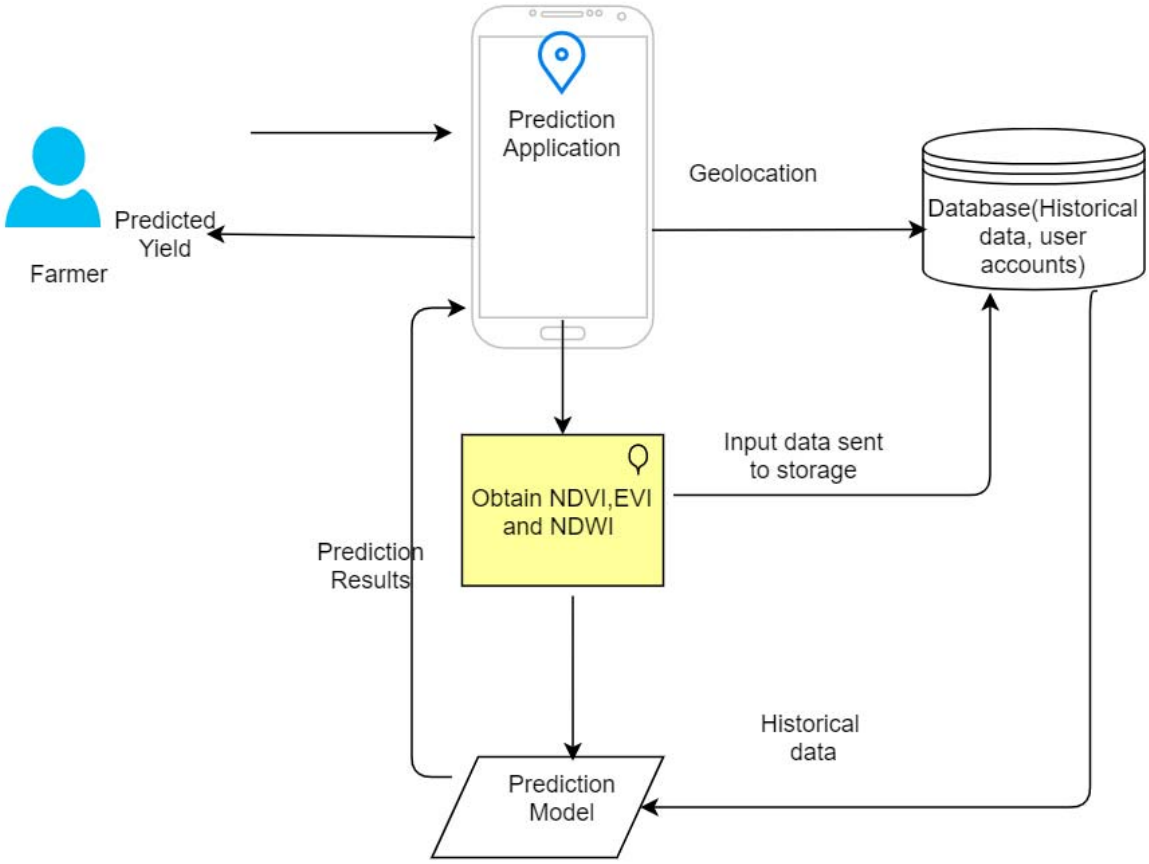


Figure 4.1: System Architecture

4.3.2 Use Case Diagram

A use case diagram is used to model the interactions of various actors with a system. The actors in the crop yield prediction system are made up of both primary and secondary actors. Farmers are the main users of the system who login and query about the predicted yield for a particular year,

hence are the primary actors. The administrator who updates the model with data is the secondary actor. The system use case diagram is as illustrated in Figure 4.2 below:

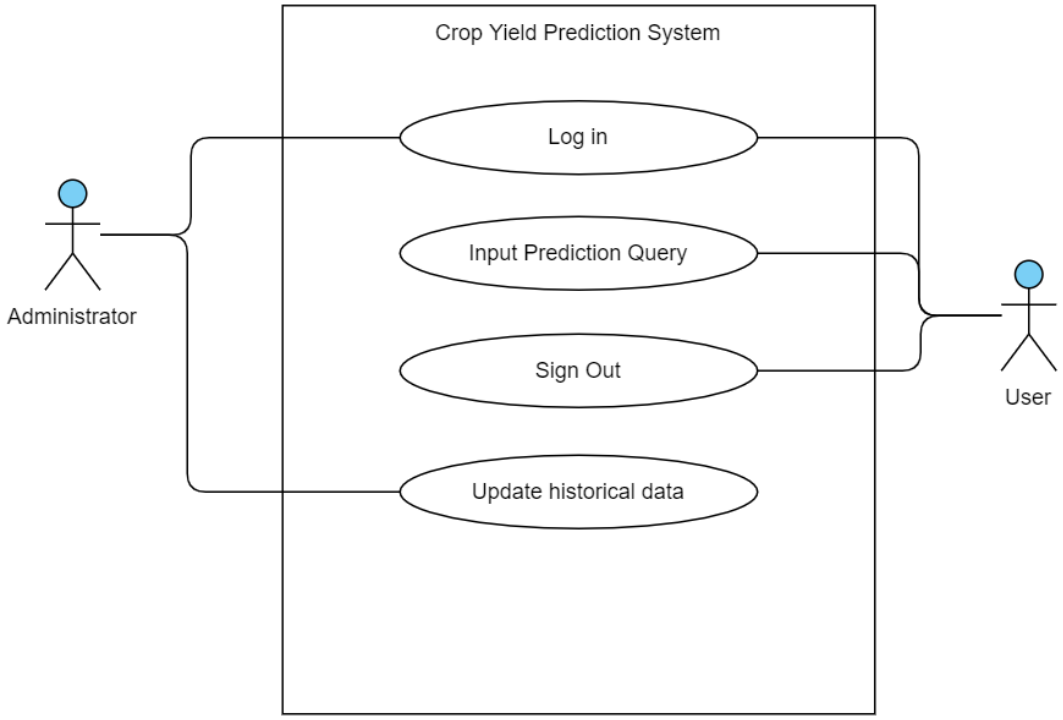


Figure 4.2: Use Case Diagram

4.3.3 Partial Domain Model

The partial domain model is a representation of a system being developed, that shows the main concepts of a particular domain being modelled. It has been used in this study to model the concepts pertinent to the crop yield prediction system, depicting the roles and rules relating to the data involved. Figure 4.3 below displays the partial domain model of the crop yield prediction system.

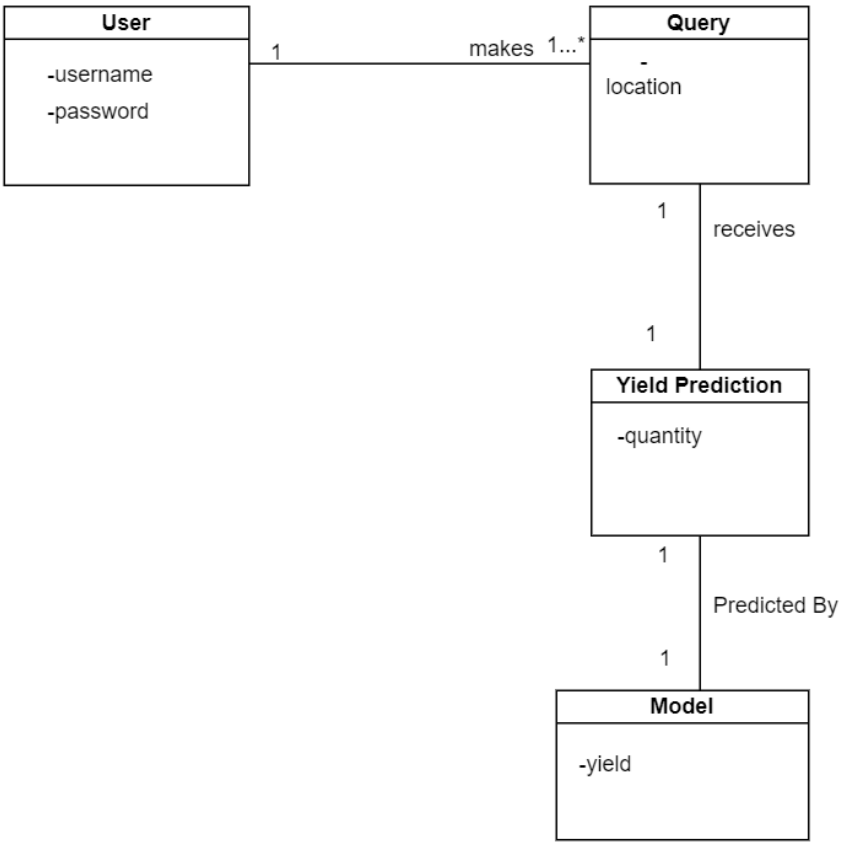


Figure 4.3: Partial Domain Model

4.3.4 Sequence Diagram

A sequence diagram has been used to illustrate the sequence of messages being sent and replied between various classes identified that provide functionality to the crop yield prediction system. It helps to understand the flow of logic of the system and how the intended functionality is achieved. The sequence diagram is as shown in Figure 4.4 below:

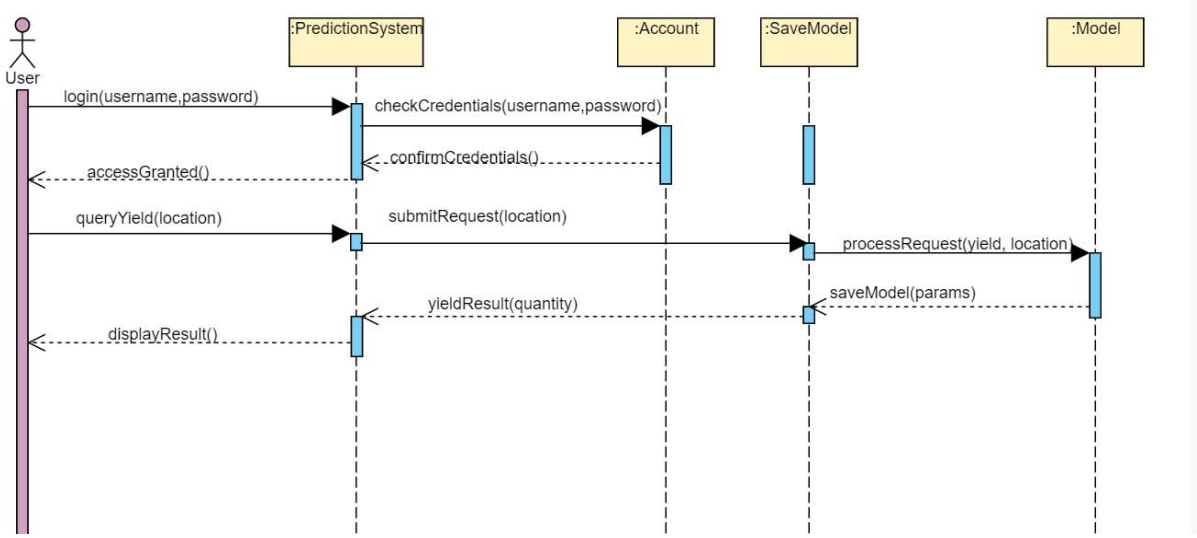


Figure 4.4: Sequence Diagram

4.3.5 Design Class Diagram

Class diagrams are important during system design since they allow direct mapping of classes to most of the object oriented programming languages. A class diagram identifies the attributes and methods of a class while also showing the relationship between objects. The class diagram used to model the crop yield prediction system is as shown in Figure 4.5 below:

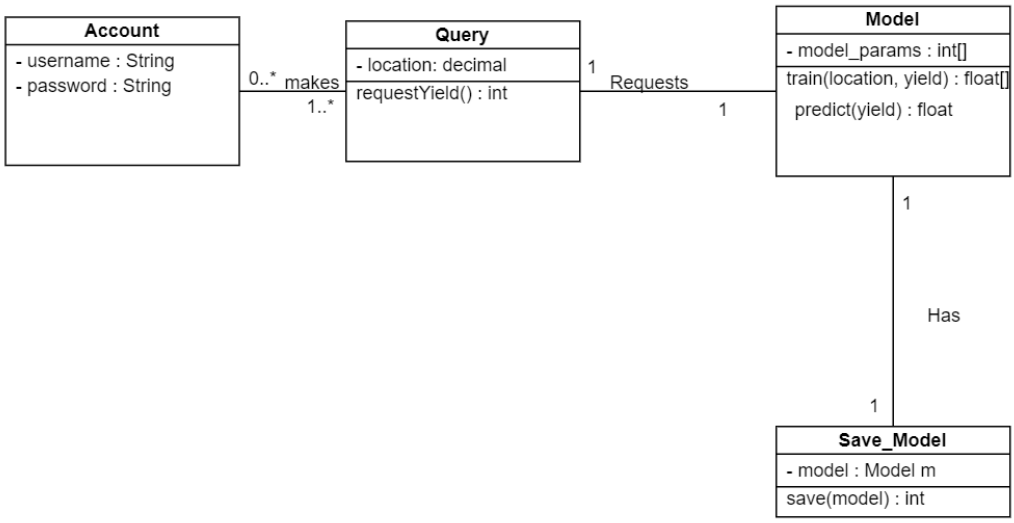


Figure 4.5: Class Diagram

Chapter 5: System Implementation and Testing

5.1 Introduction

This chapter details the implementation methods that have been used to develop the prediction model. The inputs to the model are and their sources are identified. The algorithm that has been used to develop the model is also illustrated with the results that have been achieved on the trained data. Steps that have been taken for data preprocessing are discussed, and the model evaluation method explained.

5.2 Hardware and Software Environment

The model was developed using python programming language. The key libraries that were used include numpy, pandas and keras. This study also leveraged the use of Google Earth Engine to extract data from the satellite images obtained from MODIS so as to save on computer resources required for processing and storing a large volume of satellite imagery data. Table 5.1: Hardware and Software Environmentbelow summarizes the environment used.

Table 5.1: Hardware and Software Environment

Platform	**Application**	**Specification**
Software	Numpy	Version 1.8.2
	Pandas	Version 1.0.1
	Keras	Keras 2.3.0
Hardware	RAM	16GB

5.3 Data Sources and Preprocessing

Data used in the study consisted of crop yield data at the county level obtained from the Ministry of Agriculture, Fisheries and Livestock and satellite imagery data obtained from MODIS. The crop yield corresponding to a harvesting season for a given year defined in tons was obtained for the respective counties that produces maize, wheat and rice. These crops were selected because they

are the staple food crops for local consumption in Kenya. The entire crop yield dataset used for the study was crop yield for the period 2012 to 2018.

Moderate Resolution Imaging Spectroradiometer (MODIS) imagery was accessed from the Google Earth Engine Platform via the Google Earth platform python API. MODIS sensor, which is aboard the Terra and Aqua satellites, gives a 2330 km wide swath view across the entire earth since 2000. Python code was then used to retrieve bands covering the major maize, wheat and rice producing counties for every year. The MODIS data was obtained for the years 2012 to 2018. Thereafter, three indices namely Normalized Difference Vegetation Index (NDVI), Enhanced Vegetation Index (EVI), and Normalized Difference Water Index were computed to measure the vegetation greenness that show the health of crops. The equations for obtaining the indices from the bands are as shown in the equations below:

$$NDVI = \frac{NIR - Red}{NIR + Red} \tag{5.1}$$

$$EVI = G * \frac{NIR - Red}{NIR + C1 * Red - C2 * Blue + L} \tag{5.2}$$

$$NDWI = \frac{Green - NIR}{Green + NIR} \tag{5.3}$$

NDVI is an index used to measure patterns of greenness in a given area, indicating how much vegetation is in the given area. NDVI uses two bands namely near infrared(NIR) and red band of the electromagnetic spectrum. The EVI equation is used to improve the vegetation signal because of its sensitivity to high biomass regions, and is used to reduce atmospheric influences that affect the NIR and red band reflectance. The atmospheric influences are incorporated into the equation by C1 and C2. It also reduces sensitivity to soil, represented by L in the equation. G is used as the gain or scaling factor. The monthly average of the three indices were used for every county

selected. Crop yield data from the year 2012 to 2017 was used as the training set, with 2018 data being used as the test set.

5.4 Training Machine Learning Models

5.4.1 Ridge Regression

Ridge regression is a machine learning algorithm suitable for predicting a quantity. It is a regression technique that prevents overfitting and reduces model complexity. It works by minimizing the impact of irrelevant features in the model. Ridge regression was performed with a regularization parameter alpha of a range of 0 to 20. Below is a sample code of how ridge regression was done.

```
print ('Ridge regression')

  for a in all_alphas:

    print ('alpha = ' + str(a))

    clf = Ridge(alpha=a, normalize = False)

    clf.fit(x_train, y_train)

    train_R2 = clf.score(x_train, y_train)

    train_RMSE = get_rmse(clf.predict(x_train), y_train)

    print ('RMSE on train =', train_RMSE)

    all_train_RMSE.append(train_RMSE)
```

5.4.2 Convolution Neural Network

Convolution neural network (CNN) is a deep learning neural network mostly used for image processing. CNN works by allowing each input image to pass through a series of convolution layers with filters (kernels), pooling, and fully connected layers (FC). It then applies a function to classify an object or perform prediction. The layers have various functions and each plays a unique role in the entire process. The layers are explained in detail below.

5.4.2.1 Convolution Layer

Convolution is the first layer to extract features from an input image. The advantage of Convolution is that it preserves the relationship between pixels by learning image features using small squares of input data and passes the result to next layer. It takes two inputs as an image matrix and as a filter. The filter is small and expands according to input data. The dot product of the filter and the input is calculated when the filter is mapped onto the input. This dot product results into an output matrix.

5.4.2.2 Pooling Layer

Pooling layers help in reducing the spatial size of the representation by reducing the dimensions of the image which decreases the required amount of computation and weights and increases the robustness. Spatial pooling also called subsampling helps to reduce dimensionality of each map but retains the important information. It can be of any of the three types: Max Pooling, Average Pooling and Sum Pooling, with each layer having a specific function. Max pooling takes the largest element from the rectified feature map. Average pooling takes the average of the elements in the map, while the sum of all elements in the feature map is known as sum pooling.

5.4.2.3 Fully Connected Layer

On this layer, a matrix is converted to a vector and fed into the fully connected layer like neural network to help map the representation between the input and the output. With the fully connected layers, features are combined together to create a model and a function is used to classify the outputs or make a prediction. The main functions used are Sigmoid function, Softmax function and ReLu function.

A histogram CNN architecture was adopted for this study; the raw images of the input data which consisted of NDVI, EVI and NDWI were mapped into histograms in order to reduce

dimensionality. The histograms were sequentially fed into the convolution layer, with a dimension of 46 x 46 x 3. The CNN architecture composed of 3 convolution blocks and one fully connected layer. Each convolution block has two convolution layers, with a kernel size of 3 x 3. The first convolution has 1 stride while the second convolution layer takes a 2 stride shift. A stride refers to the number of pixel shifts along one direction. Figure 5.1:Convolution Neural Network Architecture below shows the architecture used in this study.

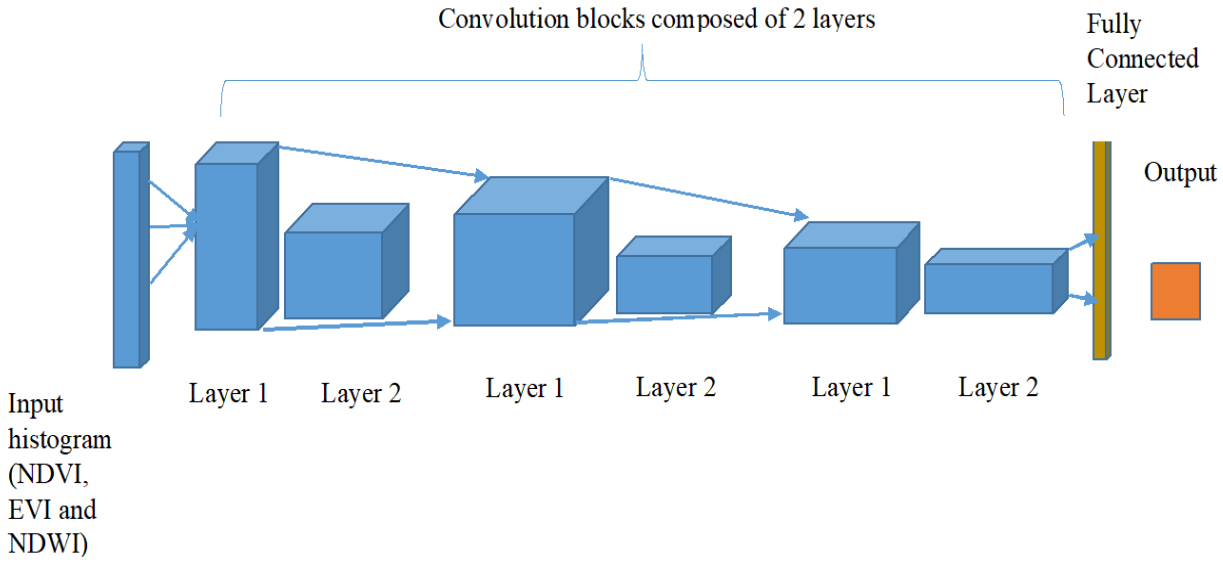


*Figure 5.1:*Convolution Neural Network Architecture

The model was thereafter trained with 1000 iterations and the hyper-parameters used were a batch size of 64, with optimizer Adam selected. The sample code below shows the neural model that was used to implement this.

```
class CNN_Model():
  def __init__(self, config, name):

    self.x = tf.placeholder(tf.float32, [None, config.W, config.H, config.C], name="x")
    self.y = tf.placeholder(tf.float32, [None])
    self.lr = tf.placeholder(tf.float32, [])
```

```
    self.keep_prob = tf.placeholder(tf.float32, [])

    self.conv1_1 = conv_relu_batch(self.x, 128, 3,1, name="conv1_1")
    conv1_1_d = tf.nn.dropout(self.conv1_1, self.keep_prob)
    conv1_2 = conv_relu_batch(conv1_1_d, 128, 3,2, name="conv1_2")
    conv1_2_d = tf.nn.dropout(conv1_2, self.keep_prob)

    conv2_1 = conv_relu_batch(conv1_2_d, 256, 3,1, name="conv2_1")
    conv2_1_d = tf.nn.dropout(conv2_1, self.keep_prob)
    conv2_2 = conv_relu_batch(conv2_1_d, 256, 3,2, name="conv2_2")
    conv2_2_d = tf.nn.dropout(conv2_2, self.keep_prob)

    conv3_1 = conv_relu_batch(conv2_2_d, 512, 3,1, name="conv3_1")
    conv3_1_d = tf.nn.dropout(conv3_1, self.keep_prob)
    conv3_2= conv_relu_batch(conv3_1_d, 512, 3,1, name="conv3_2")
    conv3_2_d = tf.nn.dropout(conv3_2, self.keep_prob)
    conv3_3 = conv_relu_batch(conv3_2_d, 512, 3,2, name="conv3_3")
    conv3_3_d = tf.nn.dropout(conv3_3, self.keep_prob)

    dim = np.prod(conv3_3_d.get_shape().as_list()[1:])
    flattened = tf.reshape(conv3_3_d, [-1, dim])

    self.feature = dense(flattened, 2048, name="feature")

    self.pred = tf.squeeze(dense(self.feature, 1, name="dense"))
    self.loss_err = tf.nn.l2_loss(self.pred - self.y)

self.train_op = tf.train.AdamOptimizer(self.lr).minimize(self.loss)
```

5.5 Model Evaluation

Evaluation of the models was done using RMSE. RMSE was chosen as the metric of evaluation is because it does not take the absolute value of an error, and given the fact that it squares the errors, it gives more weight to error identification. Many studies on prediction algorithms have adapted this metric and it has proven to be sufficient.

5.6 Model Selection

The two models were evaluated based on the experiments carried out. Convolution neural network was selected as the better model compared to ridge regression since it had better accuracy and performance with regards to reduction of RMSE.

5.7 Testing

The dataset was split into two sets, one for training and the other the test set. As indicated in chapter 5, crop yield data from the year 2012 to 2017 was used as the training set, with 2018 data being used as the test set. Testing was done for maize, wheat and rice crops as they were the main crops under study. The actual yield for maize for the year 2018 for each county was thereafter compared with the predicted yield in order to validate the model. Based on the trend depicted in Figure 5.2, it is evident that the model has generalization capabilities, and can be used for future predictions.

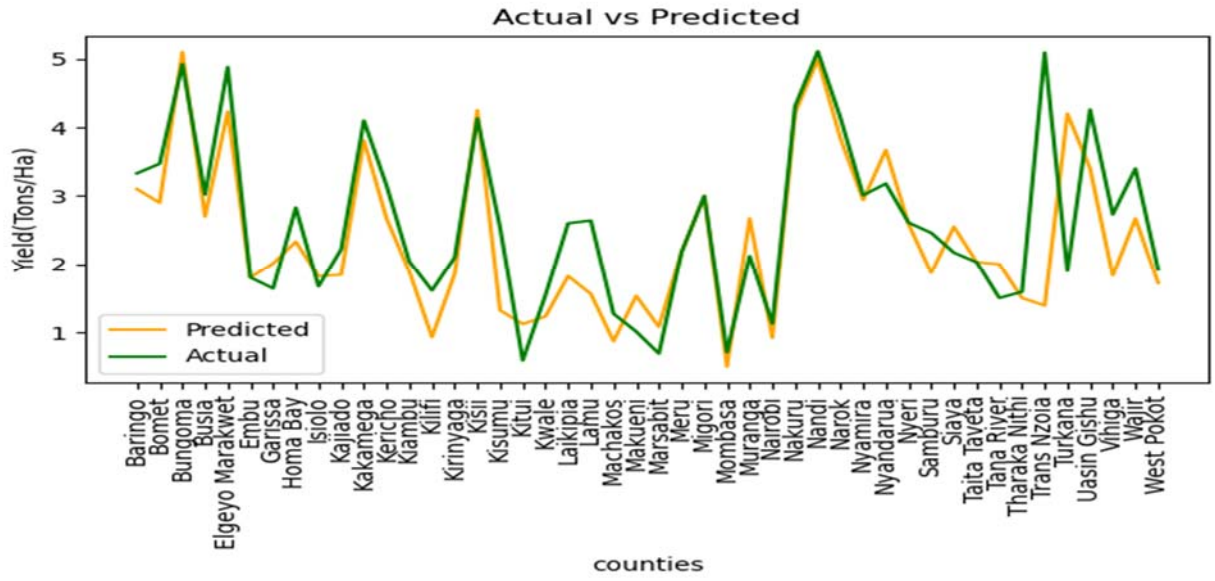


Figure 5.2: Actual versus Predicted Yield for Counties

Chapter 6: Discussions

6.1 Introduction

This chapter details the results of the study conducted, by setting out the research findings that answer the research questions identified in chapter one. The objective of this study was to develop a crop yield prediction model using machine learning algorithms, which can be used by a farmer to predict yield of their crop at the farm level. The two machine learning algorithms explored were convolution neural network and ridge regression. Input data fed into the model included two vegetation indices and one water index obtained from satellite imagery.

6.2 Research Findings

6.2.1 Crop Yield Prediction Methods

Various prediction or forecasting methods have always been adapted by farmers to predict future yield and production of crops. As discussed in chapter 2, traditional methods as well as modern technology based methods have been used in the past for crop prediction with varying results of performance and accuracy. Crop prediction can prove to be challenging sometimes because of variability of events that occur mid-season or before harvest. The use of remote sensing data, mainly from satellite imagery has played a big role in increasing the accuracy and reliability of the various modern prediction methods employed. Models based on agrometeorological data and plant measurements continue to be used by some farmers, even though technology based applications have gained popularity in recent years due to mobile phone and internet connectivity penetration to most rural and agriculture intensive areas. Most applications use machine learning algorithms in their prediction.

6.2.2 Machine Learning Algorithms for Prediction

Machine learning algorithms have been used to come up with various crop yield prediction models. The algorithms are composed of simple learning models and deep learning models which exhibit various strengths and weaknesses individually. Some of the simple learning models include linear regression, ridge regression, decision trees and support vector machine(SVM), while deep learning models constitute of convolution neural network and recurrent neural networks. Some of these models have been explained in detail in chapter two.

This study explores two of the machine learning models; convolution neural network and ridge regression. **Error! Reference source not found.**1 and Table 6.22 show the results of RMSE of three different crops calculated using the two algorithms, and it can be deduced that convolution neural network performed much better.

6.2.3 Developing the Model

As presented in chapter 5, two models were used to come up with the crop yield prediction model. One was based on a deep learning approach, while the other was a regression algorithm. The two models were then evaluated based on RMSE to identify the model with the best accuracy. The model with better accuracy was then selected as the base model for prediction.

6.2.4 Model Validation

In this study, RMSE was used to measure accuracy and compare the prediction errors of the two models used in the experiments. RMSE was calculated using the formula defined in equation (3.1).

The RMSE obtained for maize, wheat and rice using convolution neural network was 3.23, 5.92 and 6.55 respectively. Comparatively, the RMSE for maize, wheat and rice while using ridge regression was 7.52, 9.23 and 8.52 respectively. This demonstrates that convolution neural network had a better accuracy compared to the ridge regression model. Maize also recorded the lowest RMSE for both algorithms compared to wheat and rice. The poor performance of rice and wheat could be attributed to limited data available on the yield for these two crops over the years . Good RMSE results of convolution neural network could be attributed to the number of convolution layers incorporated in the algorithm, which enables unearthing of features from the input data that have a relationship with the crop yield.

Table 6.1: RMSE Results of Convolution Neural Network

	Train year	Test year	RMSE
Maize	2012-2017	2018	3.23
Wheat	2012-2017	2018	5.92
Rice	2012-2017	2018	6.55

Table 6.2: RMSE Results of Ridge Regression

	Train year	Test year	RMSE
Maize	2012-2017	2018	7.52
Wheat	2012-2017	2018	9.23
Rice	2012-2017	2018	8.52

Chapter 7: Conclusion, Recommendations and Future Work

7.1 Conclusion

Crop yield prediction is still a challenge to many farmers, since they forecast their yield based on traditional methods like crop cutting and farm recall. These methods have had low accuracy levels in the recent times owing to variability in weather patterns due to climate change. The main purpose of this study was to develop a crop yield prediction model using machine learning algorithms, which can be used by farmers to predict crop yield and enable farmers to make early informed decision regarding crop management. Various studies that have implemented machine learning were reviewed, and they displayed a good level of accuracy in crop yield prediction.

Two machine learning algorithms were developed and their performance measured based on the RMSE metric. Vegetation indices from satellite data were fed as input to the models, where they were examined under the two model experiments conducted. Thereafter, convolution neural network was selected as the optimal model for crop yield prediction because it exhibited better accuracy.

Based on the results of the model, it can be concluded that machine learning methods can be successfully used for crop yield prediction and can contribute to better crop management, planning and improved agricultural practices, which eventually lead to food security.

7.2 Recommendations

Recommendations based on this study are as outlined below:

i. The researcher recommends use of more data, especially on crop yield for better prediction. The model was trained based on yield data from 2012 to 2018, which was a limited dataset since machine learning algorithms require large amounts of data in order to understand patterns better and provide better accuracy.
ii. Tests on the developed model should be performed at ground level with farmers' involvement. This ensures better adoption of the application while obtaining recommendations from farmers on how the application can be improved for ease of use.

7.3 Future Work

i. The crop yield predictions obtained at farm level could be aggregated to obtain the national level yield prediction, with the assumption of mass adoption of the application by farmers.

ii. Future studies could also include recommendations sent out to farmers through SMS notifications, giving advice on agricultural practices and products that could improve their yield.

iii. IOT could also be incorporated to obtain soil parameters, which could contribute to overall better yield since they would give real time information on soil properties which can be used for improvement of the crop on time.

References

Al-Sammarraie, N. A., Al-Mayali, Y. M., & El-Ebiary, B. Y. (2018). Classification and Diagnosis using Back Propagation Artificial Neural Networks (ANN). *2018 International Conference on Smart Computing and Electronic Enterprise (ICSCEE)*, (pp. 1-5). doi:10.1109/ICSCEE.2018.8538383

Boaz, A., & Ashby, D. (2003). *Fit for purpose? Assessing research quality for evidence based policy and practice.* London: ESRC UK Centre for Evidence Based Policy and Practice.

Craig, M., & Atkinson, D. (2013). *A Literature Review of Crop Area Estimation. A Technical Report for FAO.*

Ethem, A. (2010). *Introduction to Machine Learning.* Cambridge, Massachusetts; London,England: The MIT Press.

FAO. (2015). *The Economic Lives of Smallholder Farmers.* Rome.

FAO. (2020). *FAO in Kenya*. Retrieved from FAO website: ttp://www.fao.org/kenya/fao-in-kenya/kenya-at-a-glance/en/

Ferencz, C., Bognar, P., Pasztor, S., Molnar, G., Timar, G., Hamar, D., . . . Ferencz, O. (2004). Crop Yield Estimation by Satellite Remote Sensing. *International Journal of Remote Sensing*, 4113–4149.

Fermont, A., & Benson, T. (2011). Estimating Yield of Food Crops Grown by Smallholder Farmers: A review in the Uganda context. *International Food Policy Research Institute (IFPRI)., 1097*.

Gardner, M., & Dorlinga, S. (1998). Artificial Neural Networks (the Multilayer Perceptron)—a Review of Applications in the Atmospheric Sciences. *Atmospheric Environment, 32*(14-15), 2627-2636. doi:10.1016/S1352-2310(97)00447-0

Goyal, M. (2016, December). Use of Different Multivariate Techniques for Pre-Harvest Wheat Yield Estimation in Hisar (Haryana). *International Journal of Computer & Mathematical Sciences, 5*(12).

Haekyung, P., Kyungmin, K., & Lee, D. K. (2019). Prediction of Severe Drought Area Based on Random. *MDPI*.

Hanuschak, G. A. (2013). *Timely and Accurate Crop Yield Forecasting and Estimation.* A Technical Report for FAO.

Hernandez, J., Lobos, G. A., Matus, I., del Pozo, A., Silva, P., & Galleguillos, M. (2015). Using Ridge Regression Models to Estimate Grain Yield from Field Spectral Data in Bread Wheat

(Triticum Aestivum L.) Grown under Three Water Regimes. *Remote Sensing, 7*, 2109-2126. doi:doi:10.3390/rs70202109

IBM. (2017, October 02). *https://developer.ibm.com/articles/cc-cognitive-big-brained-data-pt2/.* Retrieved April 23, 2019, from https://developer.ibm.com: https://developer.ibm.com/articles/cc-cognitive-big-brained-data-pt2/

Ishwaran, H., & Rao, S. J. (2014). Geometry and Properties of Generalized Ridge Regression in High Dimensions. *Contemporary Mathematics, 622*.

Johnson, D. M. (2014). An Assessment of Pre- and Within-season Remotely Sensed Variables for Forecasting Corn and Soybean Yields in the United States. *Remote Sensing of Environment, 141*, 116-128. doi:10.1016/j.rse.2013.10.027

Kabubo-Mariara, J., & Millicent, K. (2015, March). Climate Change and Food Security in Kenya. *Environment for Development Discussion Paper Series*.

Khaki, S., & Wang, L. (2019). Crop Yield Prediction Using Deep Neural Networks. *Frontiers in Plant Science, 10*, 621. doi:10.3389/fpls.2019.00621

Kodimalar, P., & Surianarayanan, C. (2019). An Approach for Prediction of Crop Yield Using Machine Learning and Big Data Techniques. *International Journal of Computer Engineering and Technology, 10*(3), 110-118.

Kothari, C. (2004). *Research Methodology Methods and Techniques.* New Delhi: New Age International (P) Ltd.

Kumar, A., Kumar, N., & Vats, V. (2018). Efficient Crop Yield Prediction using Machine Learning. *International Research Journal of Engineering and Technology (IRJET), 5*(6), 3151-3159.

Kuwataa, K., & Shibasaki, R. (2016, July 12-19). Estimating Corn Yield in The United States With Modis Evi and Machine Learning Methods. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, III*(8).

Laxmi, R. R., & Amrender, K. (2011). Weather Based Forecasting Model for Crops Yield Using Neural Network Approach. *Statistics and Applications, 9*(1&2), 55-69.

Lopez, L. R., Baruth, B., El Aydam, M., Eric, W., & Garcia, A. T. (2015, May 26). Crop monitoring and yield forecasting at global level: The GLOBCAST project from the European Commission. *32*.

Luck, D. J., & Rubin, R. S. (2009). *Marketing Research.* New Delhi: PHI Learning.

Manjula, A., & Narsimha, D. G. (2016). Crop Yield Prediction with Aid of Optimal Neural Network in Spatial Data Mining: New Approaches. *International Journal of Information & Computation Technology, 6*(1), 25-33.

Mohamad, A. M. (2019). Toward Precision in Crop Yield Estimation Using Remote Sensing and Optimization Techniques. *Agriculture, 9*(3), 54.

Noronha, P., Divya, .., & Shruthi, .. (2016, October). Comparative Study of Data Mining Techniques in Crop Yield Prediction. *International Journal of Advanced Research in Computer and Communication Engineering, 5*(3). doi:10.17148/IJARCCE

Pavani, S., & Beulet, A. S. (2019, December). Heuristic Prediction of Crop Yield using Machine Learning Technique. *International Journal of Engineering and Advanced Technology, 9*(3). doi:10.35940/ijeat.A1027.1291S319

Petersen, L. (2018). Real-time Prediction of Crop Yields from MODIS Relative Vegetation Health: A Continent-wide Analysis of Africa. *Remote Sensing*.

Porchilambi, K., & Sumitra, P. (2019, March). Machine Learning Algorithms for Crop Yield Prediction: A Survey. *Journal of Emerging Technologies and Innovative Research, 6*(3).

Priyanka, T., Pratishtha, S., & Malathy, C. (2018). Agricultural Crop Yield Prediction Using Artificial Intelligence and Satellite Imagery. *Eurasian Journal of Analytical Chemistry*.

Rojas, O. (2007). Operational Maize Yield Model Development and Validation Based on Remote Sensing and Agro-meteorological Data in Kenya. *International Journal of Remote Sensing, 28*(17), 3775-3793. doi:10.1080/01431160601075608

Siganos, D., & Stergiou, C. (1996). Neural Networks. *SURPRISE 96 Journal*, vol 4.

United Nations. (2015). *development.un.org/content/documents*. Retrieved April 19, 2019, from https://sustainabledevelopment.un.org: https://sustainabledevelopment.un.org/content/documents/21252030%20Agenda%20for%20Sustainable%20Development%20web.pdf

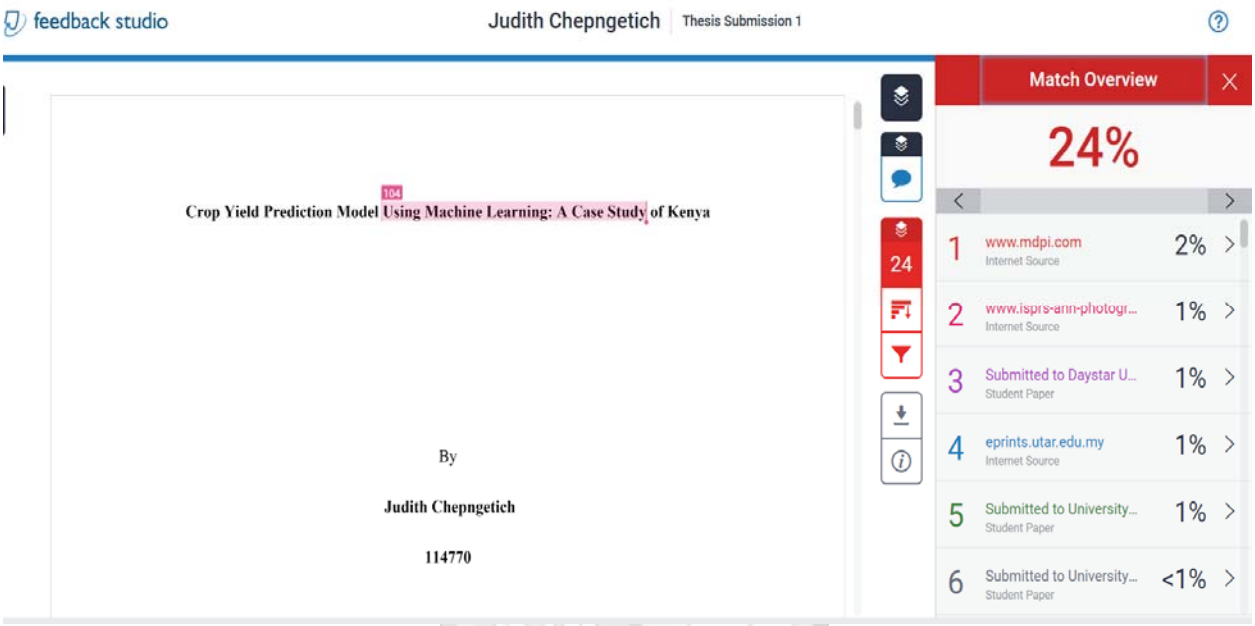
United States Department of Agriculture. (2012). *The Yield Forecasting and Estimating Program of NASS.*

Ying-xue, S., Huan, X., & Li-jiao, Y. (2017). Support Vector Machine-Based Open Crop Model (SBOCM): Case of Rice Production in China. *Saudi Journal of Biological Sciences, 24*. doi:10.1016/j.sjbs.2017.01.024

You, J., Li, X., Low, M., Lobell, D., & Ermon, S. (2017). Deep Gaussian Process for Crop Yield Prediction Based on Remote Sensing Data. *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, (pp. 4559-4565).

Appendix

Appendix 1: Originality Report



Appendix 2 : Ethical Clearance Certificate

Strathmore UNIVERSITY

12th November 2019

Ms Chepngetich, Judith
judith.chepngetich@strathmore.edu

Dear Ms Chepngetich,

RE: Crop Yield Prediction Model Using Machine Learning: A Case Study of Kenya

This is to inform you that SU-IERC has reviewed and **approved** your above research proposal. Your application approval number is **SU-IERC0571/19** .The approval period is **12th November, 2019 to 11th November, 2020.**

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 72 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 72 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://oris.nacosti.go.ke and also obtain other clearances needed.

Yours sincerely,

for: Dr Virginia Gichuru,
Secretary; SU-IERC

Cc: Prof Fred Were,
Chairperson; SU-IERC

STRATHMORE UNIVERSITY INSTITUTIONAL ETHICS REVIEW COMMITTEE (SU-IERC)
12 NOV 2019
TEL: +254 (0)703 034 000
P. O. Box 59857 - 00200
NAIROBI - KENYA

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

VT OMNES VNVM SINT

Appendix 3: Sample Maize Yield Data

COUNTY	Year	Harvested Area (HA)	Production (MT)	Yield (MT/HA)
Baringo	2012	39,753	71,867	1.81
Bomet	2012	32,697	73,278	2.24
Bungoma	2012	96,209	262,381	2.73
Busia	2012	41,990	50,102	1.19
Elgeyo/Marakwet	2012	31,533	91,964	2.92
Embu	2012	45,215	46,750	1.03

Garissa	2012	433	411	0.95
Homabay	2012	67,420	97,513	1.45
Isiolo	2012	987	979	0.99
Kajiado	2012	30,145	2,218	0.07
Kakamega	2012	76,285	172,308	2.26
Kericho	2012	41,971	98,549	2.35
Kiambu	2012	44,821	27,831	0.62
Kilifi	2012	70,672	18,625	0.26
Kirinyaga	2012	31,935	46,562	1.46
Kisii	2012	59,635	97,513	1.64
Kisumu	2012	33,395	97,513	2.92
Kitui	2012	77,210	55,238	0.72
Kwale	2012	39,490	13,225	0.33
Laikipia	2012	36,163	119,852	3.31
Lamu	2012	20,127	19,410	0.96
Machakos	2012	149,338	90,926	0.61
Makueni	2012	102,825	71,782	0.7
Mandera	2012	215	-4	0
Marsabit	2012	274	214	0.78
Meru	2012	96,244	142,499	1.48
Migori	2012	78,580	97,513	1.24
Mombasa	2012	1,003	451	0.45
Murang'a	2012	56,654	50,462	0.89
Nairobi	2012	1,018	852	0.84
Nakuru	2012	89,649	298,974	3.33
Nandi	2012	73,552	217,360	2.96
Narok	2012	109,940	184,295	1.68
Nyamira	2012	54,820	97,513	1.78
Nyandarua	2012	19,588	27,442	1.4
Nyeri	2012	33,593	26,067	0.78

Samburu	2012	1,227	1,024	0.83
Siaya	2012	76,385	97,513	1.28
Taita/Taveta	2012	13,986	5,113	0.37
Tana River	2012	6,269	5,357	0.85
Tharaka-Nthi	2012	26,779	38,192	1.43
Trans Nzoia	2012	104,421	432,342	4.14
Turkana	2012	2,252	3,529	1.57
Uasin Gishu	2012	91,310	257,649	2.82
Vihiga	2012	18,010	27,857	1.55
Wajir	2012	268	191	0.71
West Pokot	2012	33,035	110,674	3.35
Baringo	2013	29,117	55,805	1.92
Bomet	2013	30,620	72,236	2.36
Bungoma	2013	92,705	221,586	2.39
Busia	2013	45,898	63,230	1.38
Elgeyo/Marakwet	2013	32,015	101,336	3.17
Embu	2013	26,820	35,105	1.31
Garissa	2013	205	199	0.97
Homabay	2013	74,359	110,380	1.48
Isiolo	2013	1,015	1,073	1.06
Kajiado	2013	34,721	63,460	1.83
Kakamega	2013	73,463	172,683	2.35
Kericho	2013	41,170	80,022	1.94
Kiambu	2013	43,210	21,609	0.5
Kilifi	2013	647	310	0.48
Kirinyaga	2013	40,103	48,237	1.2

Appendix 4: Sample Wheat Yield Data

COUNTY	Year	Harvested Area (HA)	Production (MT)	Yield (MT/HA)	
Bungoma	2012	321	920	2.9	
Elgeyo Marakwet	2012	944	2,889	3.1	
Kericho	2012	58	210	3.6	
Laikipia	2012	5,991	19,158	3.2	
Marsabit	2012	60	206	3.4	
Meru	2012	15,265	41,387	2.7	
Nakuru	2012	31,657	101,180	3.2	
Nandi	2012	54	179	3.3	
Narok	2012	55,896	193,784	3.5	
Nyamira	2012	124	320	2.6	
Nyandarua	2012	2,326	10,086	4.3	
Nyeri	2012	5,252	13,925	2.7	
Samburu	2012	800	3,240	4.1	
Trans Nzoia	2012	2,320	5,977	2.6	
Uasin Gishu	2012	28,045	69,017	2.5	
Bungoma	2013	350	983	2.8	
Elgeyo Marakwet	2013	81	219	2.7	
Kericho	2013	40	170	4.3	
Laikipia	2013	5,060	20,250	4	
Marsabit	2013	42	108	2.6	
Meru	2013	13,295	58,662	4.4	
Nakuru	2013	31,292	82,057	2.6	
Nandi	2013	41	132	3.2	
Narok	2013	68,727	195,489	2.8	
Nyamira	2013	96	210	2.2	

Nyandarua	2013	1,408	3,510	2.5	
Nyeri	2013	5,420	16,639	3.1	
Samburu	2013	450	1,810	4	
Trans Nzoia	2013	1,957	4,834	2.5	
Uasin Gishu	2013	21,385	57,740	2.7	
Bungoma	2014	305	899	2.9	
Elgeyo Marakwet	2014	105	284	2.7	

Appendix 5: Sample Rice Yield Data

COUNTY	Season	Area (Ha)	Quantity (Ton)
Baringo	Long Rains	10	46
Bomet	Long Rains	0	0
Bungoma	Long Rains	44	143
Busia	Long Rains	493	734
Homabay	Long Rains	173	341
Kakamega	Long Rains	64	43
Kilifi	Long Rains	105	160
Kirinyaga	Long Rains	2,884	10,804
Kisumu	Long Rains	2,230	11,150
Kwale	Long Rains	718	417
Meru	Long Rains	18	9
Migori	Long Rains	311	464
Mombasa	Long Rains	0	0
Murang'a	Long Rains	32	46
Siaya	Long Rains	291	957
Taita Taveta	Long Rains	260	463
Tana River	Long Rains	252	217

Tharaka Nithi	Long Rains	12	11
Baringo	Short Rains	0	0
Bomet	Short Rains	0	0
Bungoma	Short Rains	60	201
Busia	Short Rains	965	4,348
Homabay	Short Rains	223	426
Kakamega	Short Rains	35	10
Kilifi	Short Rains	11	4
Kirinyaga	Short Rains	11,253	54,306
Kisumu	Short Rains	4,700	23,505
Kwale	Short Rains	31	11
Meru	Short Rains	18	12
Migori	Short Rains	210	440
Mombasa	Short Rains	0	0
Murang'a	Short Rains	38	88
Siaya	Short Rains	95	401
Taita Taveta	Short Rains	200	340
Tana River	Short Rains	229	229
Tharaka Nithi	Short Rains	2	1

Appendix 6: Location Data

location	lat	long
Mombasa	-4.0546598	39.6635895
Kwale	-4.17375	39.4520607
Kilifi	-3.63045	39.8499184
Tana River	-1.48256	40.0334091

Lamu	-2.27169	40.9020119
Taita Taveta	-3.39879	37.6833611
Garissa	-0.45275	39.6460114
Wajir	1.7471	40.0573196
Mandera	3.93725	41.8568802
Marsabit	2.33468	37.99086
Isiolo	0.35462	37.58218
Meru	0.04626	37.6558685
Tharaka Nithi	-0.33316	37.6458702
Embu	-0.53987	37.4574318
Kitui	-1.36696	38.0105515
Machakos	-1.52233	37.2652092
Makueni	-1.80409	37.6203384
Nyandarua	-0.27088	36.3791695
Nyeri	-0.42013	36.9475899
Kirinyaga	-0.49887	37.2803116
Muranga	-0.72104	37.1525917
Kiambu	-1.17139	36.8355598
Turkana	3.11988	35.5964203
West Pokot	1.23889	35.1119385
Samburu	1.09667	36.6980591
Trans Nzoia	0.52036	35.2699318
Uasin Gishu	1.01572	35.0062218
Elgeyo Marakwet	0.673056	35.508333
Nandi	0.20387	35.1049995
Baringo	0.49194	35.7430305
Laikipia	0.01667	37.0728302
Nakuru	-0.30719	36.0722504
Narok	-1.08083	35.871109

Kajiado	-1.85238	36.7768288
Kericho	-0.36774	35.2831383
Bomet	-0.78129	35.3415604
Kakamega	0.28422	34.7522888
Vihiga	0.081667	34.721389
Bungoma	0.5635	34.5605507
Busia	0.46005	34.1116905
Siaya	0.06116	34.2882309
Kisumu	-0.10221	34.7617111
Homa Bay	-0.52731	34.4571419
Migori	-1.06343	34.4731293
Kisii	-0.68174	34.7666588
Nyamira	-0.56333	34.9358292
Nairobi City	-1.28333	36.8166695

Appendix 7: Satellite Data Extraction Sample Code

```
import ee
import time
import sys
import numpy as np
import pandas as pd
import itertools
import os
import urllib

ee.Initialize()
```

```
def export_satimage(img,folder,name,region,scale,crs):
  task = ee.batch.Export.image(img, name, {
      'driveFolder':folder,
      'driveFileNamePrefix':name,
      'region': region,
      'scale':scale,
      'crs':crs
  })
  task.start()
  while task.status()['state'] == 'RUNNING':
    print ('Running...')
    # Perhaps task.cancel() at some point.
    time.sleep(10)
  print ('Done.', task.status())
if len(sys.argv) > 1 and os.path.exists(sys.argv[1]):
    csv_path = sys.argv[1]
else:
    csv_path = r'C:\Users\SPECTRE\Desktop\MSIT\Project data\county_locations.csv'
locations = pd.read_csv('county_locations.csv')

def appendBand(current, previous):
    # Rename the band
    previous=ee.Image(previous)
    current = current.select([0,1])
    accum = ee.Algorithms.If(ee.Algorithms.IsEqual(previous,None), current,
previous.addBands(ee.Image(current)))
    # Return the accumulation
    return accum

county_region = ee.FeatureCollection('users/Judiechep/County')
```

```
imgcoll = ee.ImageCollection("MODIS/006/MOD13Q1") \
    .filterBounds(ee.Geometry.Rectangle(33.8, -4.6,41.8, 5.5))\
   .filterDate('2012-01-01','2018-12-31')
img=imgcoll.iterate(appendBand)
img=ee.Image(img)

img_0=ee.Image(ee.Number(0))
img_5000=ee.Image(ee.Number(5000))

img=img.min(img_5000)
img=img.max(img_0)

for location, lat, lon in locations.values:
   try:
      fname = '{}'.format(location)
   except:
      pass
   offset = 0.11
   scale  = 500
   crs='EPSG:4326'

   # filter for counties
   region = ee.FeatureCollection.filterMetadata('COUNTY', 'equals', location.values)
   region = region.geometry().coordinates().getInfo()[0]
while True:
      try:
         export_satimage(img, 'County_data', fname, region, scale, crs)
      except:
         print ('retry')
         time.sleep(10)
```

```
            continue
        break
```

Appendix 8: CNN Model Sample Code

```
class CNN_Model():

    def __init__(self, config, name):


        self.x = tf.placeholder(tf.float32, [None, config.W, config.H, config.C], name="x")

        self.y = tf.placeholder(tf.float32, [None])

        self.lr = tf.placeholder(tf.float32, [])

        self.keep_prob = tf.placeholder(tf.float32, [])


        self.conv1_1 = conv_relu_batch(self.x, 128, 3,1, name="conv1_1")

        conv1_1_d = tf.nn.dropout(self.conv1_1, self.keep_prob)

        conv1_2 = conv_relu_batch(conv1_1_d, 128, 3,2, name="conv1_2")

        conv1_2_d = tf.nn.dropout(conv1_2, self.keep_prob)


        conv2_1 = conv_relu_batch(conv1_2_d, 256, 3,1, name="conv2_1")

        conv2_1_d = tf.nn.dropout(conv2_1, self.keep_prob)

        conv2_2 = conv_relu_batch(conv2_1_d, 256, 3,2, name="conv2_2")
```

```
conv2_2_d = tf.nn.dropout(conv2_2, self.keep_prob)


conv3_1 = conv_relu_batch(conv2_2_d, 512, 3,1, name="conv3_1")
conv3_1_d = tf.nn.dropout(conv3_1, self.keep_prob)
conv3_2= conv_relu_batch(conv3_1_d, 512, 3,1, name="conv3_2")
conv3_2_d = tf.nn.dropout(conv3_2, self.keep_prob)
conv3_3 = conv_relu_batch(conv3_2_d, 512, 3,2, name="conv3_3")
conv3_3_d = tf.nn.dropout(conv3_3, self.keep_prob)


dim = np.prod(conv3_3_d.get_shape().as_list()[1:])
flattened = tf.reshape(conv3_3_d, [-1, dim])


self.feature = dense(flattened, 2048, name="feature")


self.pred = tf.squeeze(dense(self.feature, 1, name="dense"))
self.loss_err = tf.nn.l2_loss(self.pred - self.y)


with tf.variable_scope('dense') as scope:
    scope.reuse_variables()
```

```
            self.dense_W = tf.get_variable('W')
            self.dense_B = tf.get_variable('b')
        with tf.variable_scope('conv1_1/conv2d') as scope:
            scope.reuse_variables()
            self.conv_W = tf.get_variable('W')
            self.conv_B = tf.get_variable('b')


        loss_err_summary_op = tf.summary.scalar("CNN/loss_err", self.loss_err)
        self.loss_reg = tf.add_n([tf.nn.l2_loss(v) for v in tf.trainable_variables()])
        loss_reg_summary_op = tf.summary.scalar("CNN/loss_reg", self.loss_reg)
        self.loss = config.loss_lambda * self.loss_err + (1 - config.loss_lambda) * self.loss_reg
        loss_total_summary_op = tf.summary.scalar("CNN/loss", self.loss)
        self.loss_summary_op = tf.summary.merge([loss_err_summary_op, loss_reg_summary_
op, loss_total_summary_op])


        self.train_op = tf.train.AdamOptimizer(self.lr).minimize(self.loss)
```

Appendix 9: Ridge Regression Sample Code

```
print ('Ridge regression')
```

```
for a in all_alphas:
    print ('alpha = ' + str(a))
    clf = Ridge(alpha=a, normalize = False)
    clf.fit(x_train, y_train)
    train_R2 = clf.score(x_train, y_train)
    train_RMSE = get_rmse(clf.predict(x_train), y_train)
    print ('RMSE on train =', train_RMSE)
    all_train_RMSE.append(train_RMSE)
```