SOIL QUALITY DETECTION FOR UPLAND RICE FARMING USING PRE-TRAINED CONVOLUTIONAL NEURAL NETWORK

BY

OLALEYE, Moses Abiola

MTech/SICT/2018/8283

DEPARTMENT OF COMPUTER SCIENCE

FEDERAL UNIVERSITY OF TECHNOLOGY

MINNA

NOVEMBER, 2021

SOIL QUALITY DETECTION FOR UPLAND RICE FARMING BASED ON PRE-TRAINED CONVOLUTIONAL NEURAL NETWORK

BY

OLALEYE, Moses Abiola

MTech/SICT/2018/8283

A THESIS SUBMITTED TO THE POSTGRADUATE SCHOOL FEDERAL UNIVERSITY OF TECHNOLOGY, MINNA, NIGERIA IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF MASTER OF TECHNOLOGY IN COMPUTER SCIENCE

NOVEMBER, 2021

ABSTRACT

One of the most important factors that affect the yields of farm produce is the quality and usability of the soil. The analysis of the soil quality and the usability of rice, a staple crop in the world, have been of interest to many researchers. There is increasing research in upland rice farming, which is making it to gain more attention. The major challenge upland environment poses is soil suitability; due to the nutrient demand of rice plant. Convolutional Neural Network is a deep learning technique that is commonly applied in the analysis of visual imagery. This study therefore has used pre-trained convolutional neural networks like AlexNet, Inception V3, ResNet18 and GoogLeNet to evaluate the usability of upland soils for rice farming. Soil samples and images were taken from Delta, Niger, Ogun and Osun states to the laboratory for analysis of soil PH and texture. Based on the soil quality score associated with each soil sample, the soil images were trained and classified as either "good for rice farming" or "not good for rice farming" using the selected pre-trained neural network. The results show high classification accuracy; however, ResNet18 performed best with an accuracy of 92.8%, slightly better than GoogLeNet. This studies further modified GoogLeNet model due to its portability to come up with a convolutional neural network model with an accuracy of 97.24%. Further research that explores additional soil parameters such as topography and moisture content is recommended.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure **Page**

GLOSSARY OF ABBREVIATIONS

Abbreviation	**Meaning**
2D	Two-Dimensional
ANN	Artificial Neural Network
CDD	Charge Couple Device
CEILAB	International Commission on Illumination
CNN	Convolutional Neural Network
DNN	Deep Neural Network
FARO	Forum of Artic Research Operators
GPU	Graphics Processing Unit
IITA	Institute of Tropical Agriculture
ILSVRC	Imagenet Large-Scale Visual Recognition Challenge
IRTP	International Rice Testing Programme
MAPE	Mean Absolute Percentage Error
NAREs	National Research Institutions
NERICA	New Rice for Africa
PCA	Principle Component Analysis
pH	Potential Hydrogen
PLS	Partial Least Square
ReLU	Rectified Linear Unit
RESNET	Residual Network
RGB	Red Blue Green

TDNN	Time Delay Neural Network
TV	Television
USAID	United States Agency for International Development
VGG	Visual Geometry Group
WARDA	West Africa Rice Development Association

CHAPTER ONE

1.0 INTRODUCTION

1.1 Background to the Study

A critical element in crop farming in the field of agriculture is the soil where crops are planted. The nature of the soil can go a long way to determine the degree of success achieved in farming. Therefore, vital properties of a soil have to be adequately measured and considered before embarking on farming on any scale. Some of these properties include the acidity or alkalinity of the soil, the macro and micro mineral constituents, soil structure, and its history (Keesstra *et al.*, 2016; Usman and Kundiri, 2016). However, finding a method that delivers a high degree of accuracy of soil properties for rice farming has been either sophisticated or lacks accuracy. Rice is major cereal food crop produced globally. The recent closure of the Nigerian border has spurred the production of rice. However, one of the major challenges in rice production is identifying a viable land.

Testing soil is very important to anyone embarking on rice farming in upland environment. Unfortunately, lack of knowledge, money, time and some resources does not afford a farmer the opportunity to carryout soil test (Kamble *et al.*, 2017). Software or apps aren't available for farmers to analyze their soils. They get tested mainly in Government laboratories at very high prices or through some other inefficient or time-consuming means. Researchers in the field of plant breeding and crop science since the early 90s have worked on producing rice varieties that can thrive in upland areas. Upland rice farming is therefore receiving more attention. This is attributed to incessant challenges associated with water areas, predominantly, flooding. However, there is a fundamental challenge with rice farming in upland area – soil type. Major challenges confronting lowland or swampy area rice include flooding and drought (Mohammed *et al.*, 2019). Out of fear of low yield, a farmer would even apply fertilizers even when not necessary. This is a situation when soil is capable of producing maximum yield. In some other cases the approximate amount of fertilizer to be added is not known. This also most often results in over- application. Sometimes, even when fertilizers are available, wrong application leads to soil degradation, pollution of the environment and eventual low rice yield.

From all plant nutrients Nitrogen (N) fertilizer rates deserve highest attention as too high rates may result in nitrate leaching, volatilization of N2O (greenhouse gas) and affect the farmers' profit. Too low rates will also depress the profit. The problem is accentuated by the fact that crops not only feed from soil inorganic but also from organic soil N. Most soil N tests do not consider the available organic soil N (Mengel *et al.*, 2006).

An upland environment is a naturally occurring area that is not flooded or irrigated. It is an undulating or levelled naturally drained soil with water supply through rainfall. Nearly 100 million people now depend on upland rice as their daily staple food. Almost two-thirds of the upland rice area is in Asia. Bangladesh, Cambodia, China, Northeastern India, Indonesia, Myanmar, Thailand, Nepal, and Vietnam are important producers (USAID, 2017).

Upland rice is grown in rainfed fields prepared and seeded when dry, much like wheat or maize. The ecosystem is extremely diverse, including fields that are level, gently rolling or steep, at altitudes up to 2,000 meters and with rainfall ranging from 1,000 to 4,500 mm yearly.

Soils range from highly fertile to highly weathered, infertile and acidic, but only 15 percent of total upland rice grows where soils are fertile and the growing season is long. Many upland farmers plant local rice that does not respond well to improved management practices—but these are well adapted to their environments and produce grains that meet local needs. Although the rice technology of the 1960s and 70s focused on irrigated rice, farmers in the uplands were not forgotten. Researchers produced cultivars adapted to poor soils and with improved blast resistance and drought tolerance. Some have outyielded traditional rice by more than 100 percent in evaluations (Dossou-Yovo *et al.*, 2016). Scientists at national agricultural research systems have crossed these improved rice with local cultivars and farmers are now beginning to grow the progeny. But more improvements are needed to meet the new challenges. New challenges are emerging in the world's upland rice farming areas, where already some of the world's poorest farmers try to wrest a living from fragile soils that are fast being degraded. Already the new upward pressures are resulting in a movement toward permanent agriculture and intensification of land use in upland areas. Those involved find themselves faced—in addition to the usual upland problems—with an urgent need to conserve the soil and the diversity of plant species and to cope with increasingly frequent and severe weed and disease infestations (Oladele *et al.,* 2019).

Rice is one of the major staple foods in Nigeria, consumed across all geopolitical zones and socioeconomic classes. Rice consumption is increasing rapidly in Nigeria because of the shift in consumer preference towards rice, increasing population growth, increased income levels, and rapid urbanization. It is commonly boiled and eaten with stew or vegetable soup. It is also used in the preparation of several local dishes that are eaten in every home, especially during festivals and ceremonies. However, rice production falls short of demand; the country depends heavily on rice importation of over 3 million tons annually, equivalent to over US$480 million in scarce foreign exchange. The Nigeria agricultural landscape is changing, with increased government policies aimed at stimulating private sector involvement and boosting local production. The efforts are starting to show results, as Nigeria's rice production rose from 3.7 million metric tons in 2017 to 4.0 million metric tons in 2018. For the record, the major rice producing states in Northern Nigeria are Kebbi, Borno, Kano, and Kaduna. Currently, most of the farmers producing rice rely on traditional technology with low use of improved input technologies. Average rice yields per unit area in the country are low and range between 2.0 and 3.0 t/ha compared to yields of 6–8 t/ha reported on research plots. It is important for farmers to adopt improved varieties and have a good knowledge of rice agronomy to increase rice production and productivity in the various states in Nigeria. Emphasis on the promotion of improved rice production technologies gained a fresh momentum following the recent policy of rice import restriction. Also, it warranted a need to equip extension agents with up-to-date information on crop production practices. Introduction 2 Guide to Rice Production in Northern Nigeria In this guide, we present the recommendations for achieving high rice yield in Northern Nigeria based on years of research of the International Institute of Tropical Agriculture (IITA) Ibadan, the West African Rice Center, (WARDA), Cotonou, Bénin; and the National Cereals Research Institute (NCRI), Badeggi (Omoigui *et al.*, 2020).

Rice is undoubtedly one of the most consumed cereals in the world. It is of the grass species *Oryza sativa* (Asian rice) or *Oryza glaberrima* (African rice). (Orluchukwu *et al.*, 2019). With regard to human nutrition and caloric intake, rice is the most important grain. Globally, the 9th highest importer of rice is Nigeria while it remains the highest importer in West Africa. Main varieties of rice produced in Nigeria are Fadama, Upland and Lowland rice (AgroNigeria, 2014).

In Nigeria, the growing ecology for rice is very vast and mostly underused. Upland rice has great potentials for possible expansions. Upland rice contributes to 35% of paddy fields, mangrove ecology (< 1%), irrigated rice field (15%), deep water (8%), and rain-fed lowland accounts for 45%. The arrival of FARO 45 and FARO 46 which are early maturing in nature and also, the introduction of NERICA has fostered the cultivation of rice by farmers in upland environment. As a result, the mangrove environment is being less cultivated compared with upland rice (Nahemiah, 2017). Collaborative efforts of institutions like rice research programme of International Institute of Tropical Agriculture (IITA), International Rice Testing Program (IRTP), West Africa Rice Development Association WARDA now called Africa Rice Center (AfricaRice), National Research Institutions (NAREs), lead to the invention of FAROs 38 to FARO 57. FAROs 35, 36 and 37 and later FARO 44 and 52. These varieties are changing rice farming from irrigate or shallow swamp to upland ecology.

A simple way to describe soil is referring to it as a sand and composition of organic matter that is essential for plant growth. Soil is made up of different proportions of sand, silt and clay. It exhibits several variations in their physical, mineralogical, chemical and biological properties. This is because soil is a heterogeneous unit. Knowledge of variability of soil properties is very crucial as this determines the productivity and usage of an area (Osujieke and Ezomon, 2019). Nigeria has a wide diversity of soil under different ecological conditions and with different levels of fertility. The different soils are a function of prevailing climatic condition, vegetative cover, and topography of the area among others (Sence Agric, 2014). Based on climate soil in Nigeria are grouped into northern zone of sandy soils, interior zone of laterite soils, southern belt of forest soils and alluvial soils Zones

Convolutional Neural Network in the aspect of deep neural network that is commonly used in image data analysis. The input image is passed through convolution filters, also known as convolution kernels, where input features are extracted to provide outputs known as feature maps. CNNs have found great use in natural language processing, agriculture (fruit grading, leave disease identification, soil classification and many more), image segmentation, image classification, financial time series and brain-computer interfaces (Szegedy *et al.,* 2015). CNNs are a type of multilayer perceptron that are standardized. In multilayer perceptron, each neuron in one layer is connected to neurons in the next layer. The first approach in developing CNN was taken when an article concerning the visual cortices of birds and monkeys was released by Hubel and Wiesel. The process of convolution was initiated by Kunihiko Fukushima; it was called necognitron. It actually inspired the research of Hubel and Wiesel. Yann Le Cunn performed an outstanding work in the history of CNN as he brought CNN to the level it attained to da by developing a 7 level convolutional model named as the LeNet (Ajit *et al.,* 2020).

Digital image processing as seen in the computing field is typified by algorithms used to process raw input images captured from sources like webcams, smartphones, cameras and so on. In contrast to analogue image processing, for example TV image that processes through electrical signal, digital image generally means using a digital computer for the purpose of processing 2D images. The mathematical expression of a gay level image is of the matrix form M x N. The elements are defined by function f(x, y) which corresponds to the amplitude, that is, the image brightness or gray level. The unit of an image with coordinate (x, y) is called its pixel, and is of an integer value (Pereira *et al.,* 2018).

Image processing and analysis is very has been a very effective tool in the field of agriculture for analysis of any agriculture-based substance. This technology has been put to use globally. This is because there are several devices and software out there today that are capable of capturing and manipulating digital images. One major plus to digital image analysis is that captured images aren’t destroyed while processing and the analysis is absolutely objective in nature. Image processing technique is done in agriculture by firstly capturing the image. The captured image is then processed and analyzed using a computer (Chandel and Singh, 2015).

1.2 Statement of the Research Problem

Since upland rice farming is gaining more attention, researchers are providing easier means of testing land usability. Traditional methods are time-consuming and not so reliable, technical methods are complex and expensive; for example use of special laboratories (Mahmoodi-Eshkaftaki *et al.,* 2020).

Various aspects of Computer Science have been used to address this problem on different levels, all with varying results. For example, some major computing areas that have been used include fuzzy logic, neural network, and image processing.

Therefore, this study improves on existing pre-trained CNN thereby developing a model that provides better classification accuracy for upland rice farming.

1.3 Aim and Objectives

The aim of this work is to develop a model that detects soil quality for rice farming in upland environment through pre-trained convolutional neural network. The objectives - are to:

i. Build dataset for soil quality assessment

ii. Evaluate Inception V3, AlexNet, GoogLeNet and ResNet18 CNN with the dataset and compare results

iii. Develop a CNN-based model that detects soil sample usability for upland rice farming

iv. Evaluate the performance of the model.

1.4 Scope of the Study

This work focuses on detecting soil quality for farming rice in upland areas. The dataset that were used in this work were collected from Niger, Osun, Ogun and Delta states in Nigeria. As a result, the model developed in this work cannot be used on dataset from lowland or swampy areas. Furthermore, the parameters measured from the soil dataset are pH values and texture. The measurements were taken at the Biotech laboratory, National Cereals Research Institute Badeggi, Niger State, Nigeria. Images of the samples were captured in situ with digital camera of 720 x 1600 and 13 Mega pixels; therefore, for best results, minimum of the camera specifications should be used. The computer specification used for the model training is AMD 3700x processor, 32GB DDR4 3200MHz RAM and NVIDIA 1060 6GB GDDR5.

1.5 Limitation of the Study

In this study, one major factor that influenced the methodology adopted and results, is the limited number of datasets.

1.6 Significance of Study

Methods used by rice farmers to know soil usability are either ineffective, time consuming or very expensive depending on the case. There's therefore need for a more accurate and easy way of checking if a soil type is compatible for rice farming in upland area. This work provides a very easy means for detecting soil usability for rice farming in upland areas by simply taking a snapshot of land portion of the intended farmland. Determining soil quality is very important to a rice farmer; as needs to avoid excessive acidic or basic soils. For soils that are usable, farmers will not have to buy or apply fertilizers as excess fertilizer results to soil acidification, pollution and increase financial expenditure.

CHAPTER TWO

2.0 LITERATURE REVIEW

2.1 Soil

Soil is the free surface substance that covering most land. It is made of organic matter and inorganic substances. It provides basic backing to plants and is also serves as their source of water and supplements. Soils are usually different in their physiochemical properties. Weathering and microbial activities and leaching all culminate to make varying soil types. Every soil type has its pros and cons when it comes to agricultural usage. The first thing usually notice about the soil is its colour, especially the top soil. Soil colour is caused by three main pigments to three main pigments which are; red - from iron and aluminium oxides, white - from silicates and salt, and black - source is organic matter Soil colour is reflects of the properties of the soil and the likely chemical processes going on underneath.

Soil is the most important natural non-renewable resource developed over a longer period of time due to weathering of rocks and subsequently enrichment of organic matter. Soil provides habitat for numerous microorganisms and serves as a natural medium for plant growth, thereby providing the plants with anchorage, nutrients and water to sustain the growth. Soil also serves as a universal sink for all types of pollutants, purifies ground water and is a major reserve of carbon in the universe. The role of soils to provide ecosystem services, maintenance of environmental/human health and ensuring the food security makes it as the most important and basic natural resource. Soil Science helps us to elaborate and understand how the soils provide all these services. Soil Science also provides us the basic knowledge dealing with the origin of the soil parent material, weathering of parent material and the formation of soils, morphological, physiochemical and biological features of soils, classification of soils and role of soils in the provision and maintenance of ecosystem services, food security and environmental quality (Hakeem *et al.,* 2016).

2.1.1 Soil pH

The pH of a soil sample is the most important aspect of its chemistry; as it determines nutrients availability to plants. It also determines the activity of microorganisms in the soil. Soil pH refers to the amount of hydrogen ions (H+) that are available in a solution. It is the measure of acidity or alkalinity of a soil sample. The pH value (scale) runs from 0 to 14. On the scale, 0 is the highest acidic value, 14 is the most basic value, while 7 is neutral. The primary factor that initially affects soil pH is its parent material. For instance, soils made mainly of granite tend to be acidic and examples of alkaline soils are limestone-based soils. However, soil pH changes over time either through natural or human influences. Some natural factors that affect soil pH are rainfall and climatic conditions. Human activities include fertilization with ammonium or sulfur containing fertilizers and the production of industrial by-products, for example sulfur dioxide and nitric acid.

The availability of nutrients in soils varies according to pH. For this reason, soil pH is extremely critical for plant growth and reproduction. The best pH for general nutrient availability is about 6.5. Figure 2.1 shows the effect of pH on nutrient availability.

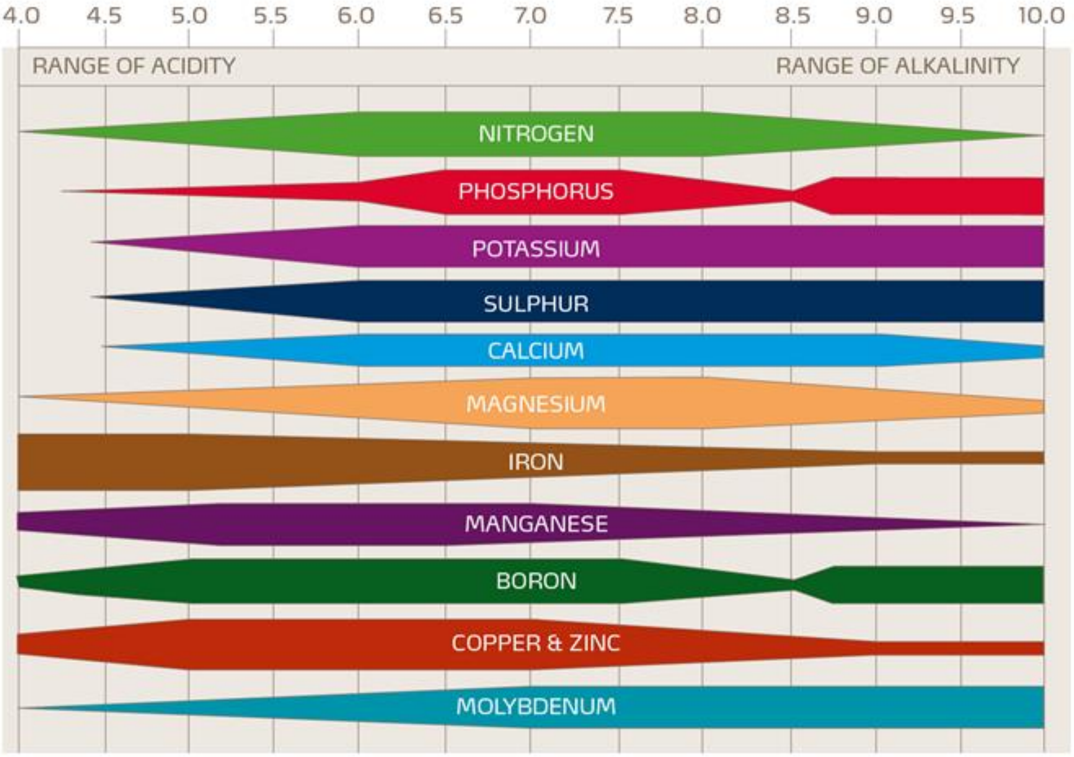


Figure 2. 1: Effect of pH on Nutrient Availability (Soil Properties, Part 3 of 3: Chemical Characteristics | Extension | University of Nevada, Reno, 2021)

Figure 2.1 shows how the macro and micro elements in the soil have influence on pH value and vis-versa. Soil pH is described as the “master soil variable” that affects soil chemical, biological, and even physical properties. It also affects processes that influence biomass yield and plant growth (Neina, 2019). The summary of the researcher’s work is seen in Figure 2.2. The work explains how soil pH affects processes that are connected with soil environment and how these processes induce changes in soil pH.

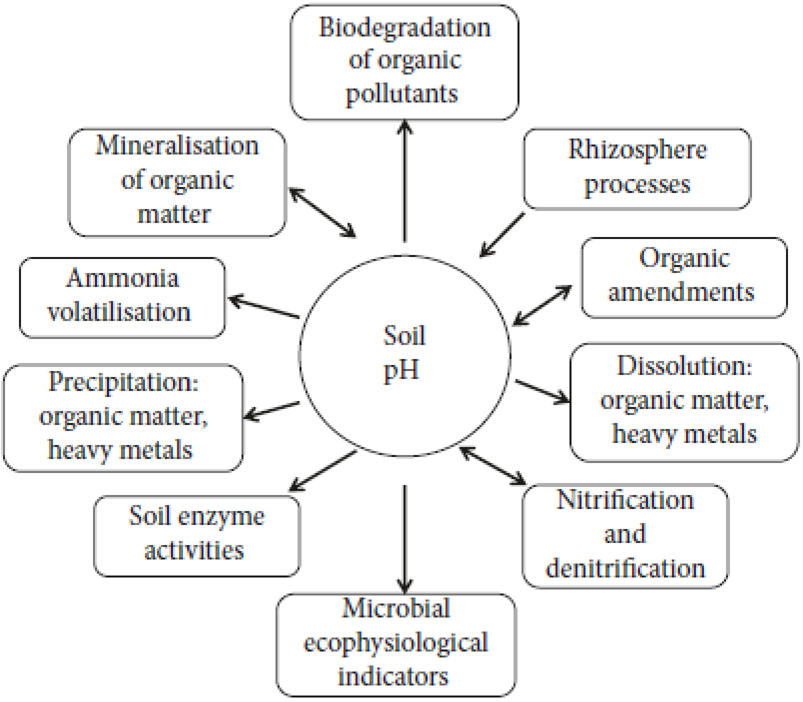

Biodegradation of organic pollutants
Mineralisation of organic matter
Rhizosphere processes
Organic amendments
Ammonia volatilisation
Soil pH
Dissolution: organic matter, heavy metals
Precipitation: organic matter, heavy metals
Soil enzyme activities
Nitrification and denitrification
Microbial ecophysiological indicators

Figure 2.2: Some biogeochemical processes and their relations with soil pH. (Neina, 2019)

Figure 2.2 shows that the pH of a soil can give an idea of what type of biological, geographical and chemical processes taking place in the soil. Conversely, these activities can also determine soil pH.

2.1.2 Soil Colour Correlated with pH Value

Important relationships between soil colour and its physiochemical attributes was established using NN. The work used neural network to predict soil colour. They found that several soil attributes could be predicted from its colour using only Red, Green and Blue (RGB) values from RGB system or L, a and b from CIELab system. The work gave high levels of accuracy (Aitkenhead *et al.*, 2013).

Soil RGB values were also related to pH values. The RGB values were used to derive the pH index of 80 soil samples and predicted the pH value of test data. The result yielded up to 70% accuracy, establishing correlation between soil colour and pH value (Kamble *et al.,* 2017). In another work, a 91% accuracy was obtained in deriving pH values from soil image samples. The methodology deployed involved using Principal Component Analysis for analysis of features and Support Vector Machine (SVM) for classification (Mahantesh, 2018). In table 2.1, the relationship between soil colour, pH and chemical constituents are compared.

Table 2.1: **Soil Characteristics, Colour and Management Implication (Forla *et al.*, 2015)**

Soil colour	Soil types and characteristics	Implications
Black	Peat/soils high in organic matter	Anaerobic conditions; drainage problems; low pH; high denitrification risk
	Vertosols	Workability; tillage problems; Zn deficiency
	Soils derived from limestone under reduced conditions	Deficiencies of P, Fe, Zn; drainage problems
White/pale/ bleached	Sandy soils	Nutrient deficiencies; leaching of nitrate, potassium, sulfate; low plant-available water
Red	Well-drained soils with high content of iron oxides	High P fixation; possible Al (and Mn) toxicities; low plant-available water
Yellow/yellow brown	Imperfectly drained to moderately well-drained soils with high content of iron oxides	Moderate P fixation; possible Mn toxicity; low plant-available water; compaction
Brown	Moderate soil organic matter levels and some iron oxides	Low to moderate P fixation; low to moderate plant-available water
Gleyed/grey/ blue grey	Near permanent waterlogging; anaerobic (reduced) conditions	Drainage problems; high denitrification risk; methane emission hazard
Mottles	Intermittent waterlogging; intermittent anaerobic (reduced) conditions	Intermittent drainage problems; denitrification risk when waterlogged; methane emission hazard when waterlogged

The properties of soil are by nature anisotropic. There is therefore need to study its associations and landscapes across regions in order for sustainable nutrient management. Sand and clay soil compositions are much more significant and relevant to the textural classification of the soils of the area than silt. Low nutrient soils can be improved by the addition of organic or farm residues to increase soil organic matter content. Also, improved management practices and cautious fertilizer application will go a long way to get the best from soil in farming practices (Adegbite *et al.,* 2019).

Analysis of soil texture in a laboratory is always difficult and not environmentally friendly. Over two days is required to get reports after sampling in a laboratory, and the procedure involves using hydrogen peroxide and sodium hydroxide as chemical dispersion reagents. The texture of a soil sample is mainly characterized arrangement of its particles. This includes clay (lower than 0.002 mm), silt (0.002 to 0.053 mm), and sand (0.053 to 2 mm). All forms of soil management practices in agricultural for example fertilizer application, controlling erosion, liming, irrigation and so on strongly depends on soil texture. Analysis of soil texture is mostly carried out with methods such as hydrometer method and pipette method. However, these are time consuming (Souza *et al.,* 2019). Soils are acidic when essential elements like Potassium, Sodium, Calcium and Magnesium which are held by soil particles are substituted by ions of hydrogen. When soils are formed under ample rainfall, they contain more acid content than the ones formed in arid conditions.

The pH of soils formed in little rainfall environment appears to be basic with approximate pH of about 7.0 Excessive application of nitrogenous fertilizers and manure under consistent farming over a number of years can result in soil acidification. Soil pH is immensely affected by Nitrogen content. Sources of Nitrogen such as fertilizers, manures and legumes all have or create some ammonium. This in turn heightens the acidity of the soil except that the plant directly absorbs the ammonium ions (Nutrition, 2018).

2.1.3 Soil Texture

In a research carried out to determine the effects of water regime/soil condition and soil texture (clay and sandy loam) on rice grain yield, the researchers used a greenhouse trial. They realized that soils with different textures significantly affected rice produced. They noted that the yield of clayey soil was 46% higher than sandy loam (Dou *et al.*, 2016). Computer scientists have also carried out studies on various aspects of soil imagery analysis and with the use of special tools and algorithms have been able to predict with a high degree of accuracy various soil properties. An analytical process to can be used to both predict and classify soil texture. In a research that worked with 63 soil samples and used PLS2 multivariate regression to correlate soil image data with standard laboratory values. The computer vision method used gave 100% match with the standard method. They concluded the process as being faster and environment friendly and cheaper (Souza *et al.*, 2019). Some other methods investigated the potentials of classifying soils based on texture through their RGB histograms. Soil images were taken with a CCD camera. There was linear correlation between silt and histogram variables as revealed by scattered plot (Sataloff *et al.*, 2012).

2.2 Upland Rice

Upland rice varieties are genetically modified rice in order to have high resistance to draught and with very little no requirement for water. They are rice grown in naturally well-drained soils, not having surface water accumulation or ground water supply, and normally not bunded. It is also called aerobic rice. There are new varieties of upland rice developed in Brazil. These have very high yield and also excellent grain quality. However, in places like Africa, Latin America and Asia cultivation of upland rice is carried out in soils that have low fertility. In fact, the most important yield limiting factor is soil nutrient deficiency. As the population of the world grows, the judicious use of soil will become more and more important in order to meet up with global food provision. This is because most human food are gotten from the soil. Therefore, it is necessary to improve upland rice yield in countries that are still developing (Fageria, 2014). FARO 45, FARO 46 and NERICA which are early maturing in nature are typical varieties developed to thrive in upland areas. When compared with lowland rice, upland rice has lower yield, this notwithstanding, it will continually be preferred because of low production cost and zero necessities for irrigation practices.

Best soil for rice should have adequate proportions of silt and clay, although there could be production variation from because of differences in soil conditions and cultivation in unsuitable soils. Rice grain yield in clay soil is 46% higher than in sandy loam soil averaged across cultivar and water regime. More panicle numbers are observed with continuous flooding treatment when compared to aerobic condition. Clay soil had 25% higher panicle number than in sandy loam soil. Cocodrie add 29% spikelet than Rondo, showing the greater resultant of rice cultivar on spikelet number over soil type and water management. In comparison, clay soil is 25% higher in water productivity than sandy loam soil. Results show that cultivar selection and soil texture are crucial determinants in deciding the water management approach to adopt (Dou *et al.*, 2016).

The best soil pH for rice growth in upland conditions is in the range of 5.5 – 6.5. In flooded conditions, it can rise to 7.0 to 7.2 (Shemahonge, 2013). Table 2.2 shows some major crops and their pH range for optimal yield.

Table 2.2: **Soil pH range for some common crops required for optimum yield (Mosaic Crop Nutrition, 2019)**

	pH Range	
5.0-5.5	**5.5-6.5**	**6.5-7.0**
Blueberries Irish Potatoes Sweet Potatoes	Barley Bluegrass Corn Cotton Fescue Grain Sorghum Peanuts Rice Soybeans Watermelon Wheat	Alfalfa Some Clovers Sugar Beets

Upland rice soil conditions differ from normal rice varieties. In the inland varieties, the soil is usually puddled and flooded to enable anaerobic conditions. In the case of upland rice, the soil is usually well drained, under tillage and aerobic in nature. Researchers Zhou *et al.*, (2014) reviewed several properties of rice crop farming and were able to highlight areas that need consideration and attention. Compaction of upland soil leads to poor germination. It also affects the development of roots in deep soil. Soil pH is usually affected by change in soil water content; this in turn affects the effectiveness and nature of nutrients in the soil.

2.3 Soil Quality Detection Models

Soil quality is regarded as the capacity of a soil to function. The assessment of a soil focuses on various aspects of the soil in order to measure the sustainability of soil management practices. De La Rosa and Sobral (2018) explored arable land identification, variations in crops, replenishment of organic matter, concentration of tillage, and justification of soil input. The quality of a soil is either inherent or dynamic in nature. As for inherent quality, it is the natural ability of the soil; for instance, clay soil naturally has higher water retaining capability than sandy soil. Also, a deep soil naturally has enough allowances for roots than surface bedrock soils. Such attributes are permanent and hardly change. As for dynamic property of a soil, it changes based on management practices used on the soil. One major goal of soil quality is to improve best practices for soil management and maximize the usage of potential soils. Therefore, the assessment soil-quality lies on its dynamic features in order to estimate the sustainability of soil management practices, but should be factored on inherent properties.

Venkat Rao (2019) worked on predicting soil quality using machine learning techniques. The paper suggests a solution in consideration of vital soil attributes and factors in order predict the soil quality. The researcher used amongst other testing algorithms Random Forest Algorithm, and using regression to increase efficiency. However, the prediction turned 70% accurate as revealed in resulting confusion matrix. Mukherjee and Lal (2014) in their research concluded that some major aspects of soil quality in relation to crop response are necessary requirements for predicting soil quality index.

Researchers Aitkenhead *et al.* (2016) designed a neural network model that is capable of estimating soil structure, texture, bulk density, pH and drainage category. The model provides estimates of these parameters from soils in a field. Each soil image (in their JPEG format) was converted to digital values and stored as three arrays of red, green and blue pixel values (with values ranging from 0 to 255).

The neural network was trained using 10-fold cross-validation. It involves producing ten subsamples of the training data, and ten ANN models being developed, each trained with 9 out of the 10 randomly-generated subsamples and was validated with the “missing” subsample. Each ANN was validated using a disparate subsample.

The neural network models used spatial covariates for Scotland’s landscape. Therefore, these exact models cannot be ineffective if applied elsewhere, especially in geographical regions with very different climate, soil or vegetation types. To achieve comparable levels of accuracy, models would need to be developed that are more applicable to regions of interest. This may be almost impossible for many areas of the world as there are no mapped information. Therefore, the method applied here is largely dependent on data availability, from soil surveys and partial datasets with appropriate accuracy.

Bhaskar Reddy *et al.* (2011) designed a system to measure the pH of blood using a microcontroller. The system is applicable to medical domain. It was tested among 15 patients and had a good result. The pH meter is digital in design and is based on the microcontroller. The meter was designed with two op-amps for summing and buffering. Similar technique was built on to develop a pH meter to measure the pH of soil. This time, a faster microcontroller was used (PIC16F876). It used a common cathode display and real time display (Al-Mashud *et al.,* 2014).

A fuzzy system was successfully designed by Abu *et al.* (2014) in order to control soil pH. The input to the system consisted of humidity, light intensity and temperature. The fuzzy method was used with Matlab software. The system also recognizes temperature, humidity and lighting changes. The prototype system was designed by Matlab software to perform simulation.

2.4 Image Processing in Soil Science

Digital image processing techniques is used in extracting useful information from an image by the analysis of the image. In the field of soil science, analysis of image is a potential means of measuring features of a soil sample or vegetation (Persson, 2011). Each element of an image otherwise known as a pixel forms a huge matrix of data. These pixels contain all the required information about the colour and brightness of an image. In the field of soil science image analysis is now getting more popular since the last decade. Areas like: macroporosity determination, root mass and length, organic matter content, soil albedo, water content, solute concentration and many more have been explored with digital image analysis.

Soil type detection using Constraint Clustering Algorithm was used by researchers Akepati and Kutakula (2018) to automatically classify soils. It had two steps; one was segmentation and the other, classification. Segmentation involves dividing digital image into multiple segments (sets of pixels) while classification was adapted to allocate classes to segmented images. Features of measured data are extracted. Support Vector machine algorithm was used under the segmentation. The major soils in their data include sand, peat and clay. They concluded that using machine vision to classify soil had higher accuracy.

Aziz *et al,* (2016) used artificial neural network to determine soil pH. The researchers made use of an existing database of soil samples containing their RGB (Red-Blue-Green) and pH values. Sample soil was therefore compared with the existing database to determine the pH value of the soil samples. The minimum errors were then determined. However, this method only delivered a wide range of possible pH values for soil samples.

In addition to the wide-ranged result, ANN has the black box (unexplainable outcome) challenge.

Researchers Kamble *et al.* (2017) implemented the calculation of soil pH using digital image processing. Eighty soil samples were collected and test in government laboratory. The work used digital image processing to determine the pH of the collected samples on the basis of RGB values. The work provided report of soil tested with likely deficient nutrient. They obtained results of about 60 to 70% accuracy compared with the values from government laboratories. The analysis of soil pore spaces is very useful in interpreting soil structure. This is mainly because soil physical and physiochemical parameters can be influenced by pore space.

In the work by (Nowak, 2015), the researcher opined image segmentation approach for detecting Pore structures. These pores have been earlier studied by soil tomography method. Density-based clustering method was used with kernel estimation. By this way, the researcher was able to identify innate data structure. The techniques of digital image processing and tomography are very useful in studying soil structure aggregates. The method used in the work is more objective than classical parametric and can be applied to data mining tasks.

2.5 Convolutional Neural Networks (CNN)

A typical CNN is made up of a deep learning algorithm. It accepts an input and allocate weights and biases to it (Saha, 2018). The weights and biases are attached to different parts of the object such that they can be differentiated from one another. CNNs have lower computational preprocessing compared to other algorithms that perform classification.

Old algorithms use filters manually, but in CNN, the network can learn filters with sufficient training. CNN architecture can be compared with the human brain connection model of neurons in the brain by copying the organization of the Visual Cortex. Convolution neural network is a multi-layer artificial neural network specially designed to handle two-dimensional input data. Each layer in the network is composed of multiple two-dimensional planes, and each plane consists of multiple independent neurons Composition, adjacent two layers of neurons connected to each other, and in the same layer of neurons are not connected between. CNNs are inspired by the early time delay neural networks and TDNNs. TDNN reduces the computational complexity in the network training process by sharing the weights in the time dimension, and is suitable for processing speech signals and time Sequence signal. CNNs use a weight-sharing network structure to make it more similar to a biological neural network, and the capacity of the model can be adjusted by changing the depth and breadth of the network, and has a strong assumption for natural images (statistical smoothness and local Correlation). Therefore, CNNs can effectively reduce the learning complexity of the network model, have fewer network connections and weight parameters, and are more likely to be trained than the full connected network with a considerable size (Al-Saffar *et al.*, 2017).

2.5.1 GoogLeNet architecture

GoogLeNet was proposed as deep CNN with the code name “Inception” (Szegedy *et al.*, 2015). It became the new setter of state-of-the-art classification and detection in ImageNet Large-Scale Visual Recognition Challenge in the year 2014 (ILSVC14). The major catch in the architecture is better utilization of computing resources. In order to achieve this, it was a carefully crafted design that allowed for adding to the depth and width of the network and the same time keeping the cost of computation invariable. In order to enhance quality, the design was based on Hebbian principle and ideology of multi-scale processing. A typical model used in the authors' submission for ILSVRC14 was the GoogLeNet which has 22 layers. The quality of the model was evaluated within the context of ImageNet data for recognition and successful classification. The most clear-cut means of improving the operation and results of deep neural network is by increasing the general size, which means the amount of levels of the network and the width (units at each level). The major aim of the inception model is based on finding out how an optimal local sparse structure in a CNN can be estimated and covered by accessible dense components. This network, although with 22 layers, it is compensated by reduced number if parameters of about 5 million when compared with a network like VGG that has about 138 million parameters. The inception layer has 1x1, 3x3, and 5x5 convolutions, a 3x3 max pooling and concatenated feature output. The convolution layer is each led by a 1x1 convolution to lower the depth of feature map and invariably reduce the number of parameters for computation. ReLU layer is also attached to the 1x1 convolution for normalized signal output. Figure 2.3 show GoogLeNet architecture and its 9 inception layers. At the end is a global average pooling layer that takes the average of individual map (Ajit *et al.,* 2020).

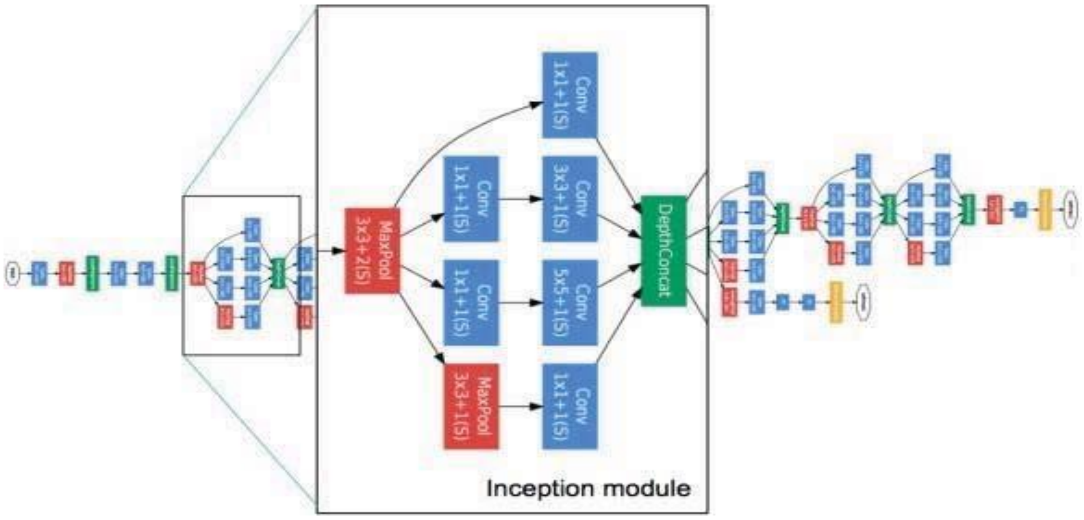


Figure 2. 3: GoogLeNet network with all the bells and whistles (Szegedy *et al.*, 2015)

2.5.2 AlexNet Architecture

AlexNet is an advanced CNN that has been trained on ImageNet dataset, which contains millions of images. CNNs had always been the go-to model for object recognition — they're strong models that are easy to control and even easier to train. They don't experience overfitting at any alarming scales when being used on millions of images. Their performance is almost identical to standard feedforward neural networks of the same size. The only problem: they're hard to apply to high resolution images. At the ImageNet scale, there needed to be an innovation that would be optimized for GPUs and cut down on training times while improving performance. The architecture consists of eight layers: five convolutional layers and three fully-connected layers. But this isn't what makes AlexNet special; these are some of the features used that are new approaches to convolutional neural networks (Deshmukh, 2019):

- **ReLU Nonlinearity**. AlexNet uses Rectified Linear Units (ReLU) instead of the tanh function, which was standard at the time. ReLU's advantage is in training time; a CNN using ReLU was able to reach a 25% error on the CIFAR-10 dataset six times faster than a CNN using tanh.

- **Multiple GPUs.** Back in the day, GPUs were still rolling around with 3 gigabytes of memory (nowadays those kinds of memory would be rookie numbers). This was especially bad because the training set had 1.2 million images. AlexNet allows for multi-GPU training by putting half of the model's neurons on one GPU and the other half on another GPU. Not only does this mean that a bigger model can be trained, but it also cuts down on the training time.
- **Overlapping Pooling.** CNNs traditionally "pool" outputs of neighboring groups of neurons with no overlapping. However, when the authors introduced overlap, they saw a reduction in error by about 0.5% and found that models with overlapping pooling generally find it harder to overfit.
- **Data Augmentation.** The authors used label-preserving transformation to make their data more varied. Specifically, they generated image translations and horizontal reflections, which increased the training set by a factor of 2048. They also performed Principle Component Analysis (PCA) on the RGB pixel values to change the intensities of RGB channels, which reduced the top-1 error rate by more than 1%.
- **Dropout.** This technique consists of "turning off" neurons with a predetermined probability (e.g. 50%). This means that every iteration uses a different sample of the model's parameters, which forces each neuron to have more robust features that can be used with other random neurons. However, dropout also increases the training time needed for the model's convergence.

Figure 2.4 shows AlexNet architecture, its various layers size at each layer and number of neurons at FC layers.

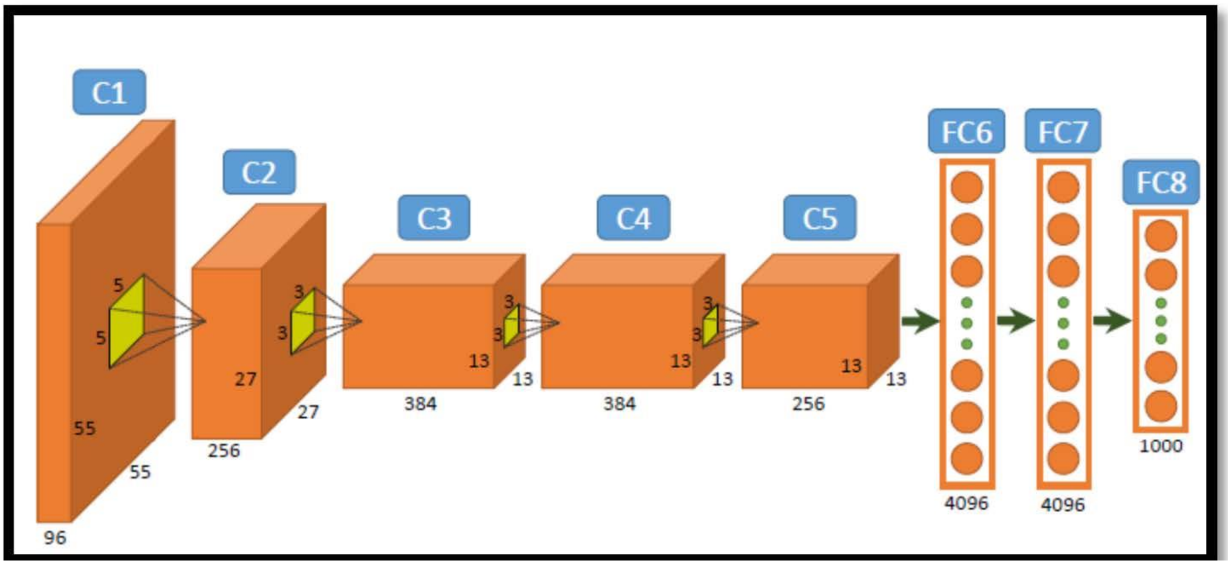


Figure 2. 4: Overall Alexnet Architecture (Samir *et al.*, 2020)

Table 2.3 gives a summary of computations that go on in each layer of the architecture

Table 2. 3: AlexNet Summary

Layer		Feature Map	Size	Kernel Size	Stride	Activation
Input	Image	1	227x227x3	-	-	-
1	Convolution	96	55 X 55 X 96	11x11	54	relu
	Max Pooling	96	27 X 27 X 96	3x3	22	relu
2	Convolution	256	27 X 27 X 256	5x5	11	relu
	Max Pooling	256	13 X 13 X 256	3x3	22	relu
3	Convolution	384	13 X 13 X 384	3x3	11	relu
4	Convolution	384	13 X 13 X 384	3x3	11	relu
5	Convolution	256	13 X 13 X 256	3x3	11	relu
	Max Pooling	256	6 X 6 X 256	3x3	22	relu
6	FC	-	9216	-	-	relu
7	FC	-	4096	-	-	relu
8	FC	-	4096	-	-	relu
Output	FC	-	1000	-	-	Softmax

AlexNet uses transfer learning approach. Transfer learning uses a pre-trained neural network. This technique allows the detachment of the last outer layers (classification layer) trained on a specific dataset, uses the remaining arrangement to retrain and get new weights related to interested classes. Transfer learning while training its newly assigned layers, plots the training progress plot which shows the training details with

2.5.3 ResNet:

This was the ILSVRC winner for 2015, ImageNet challenge. The version used at the ImageNet challenge had 152 layers; deep network. It exceeded human accuracy with about 3.6%. It used skip connection feature to solve the problem of vanishing gradient as performance dropped when the network got too deep. The input and output of one layer is copied to the next, in order words, learning the residual computation of the previous layer. At the end of each convolution is batch normalization, and about 65 million parameters were computed (He *et al.*, 2016). Figure 2.5 shows ResNet architecture and its layers.

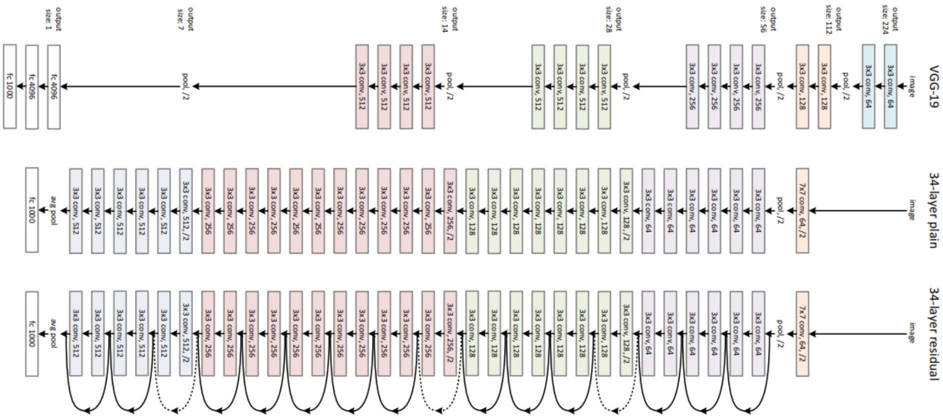


Figure 2. 5: ResNet Architecture

2.5.4 Inception V3

This is from the family of inception models. With auxiliary classifier, factorized 7x7 convolutions, and label smoothing, it has quite a number of reasonable modifications. The aim of factorizing convolutions is to limit number of parameters without attenuating efficiency of the network. Figure 2.6 shows the layers of Inception V3 and its overall architecture. The model is 42 layers deep but has a higher cost of computation compared with GoogLeNet (Szegedy *et al.*, 2016; Tsang, 2018).

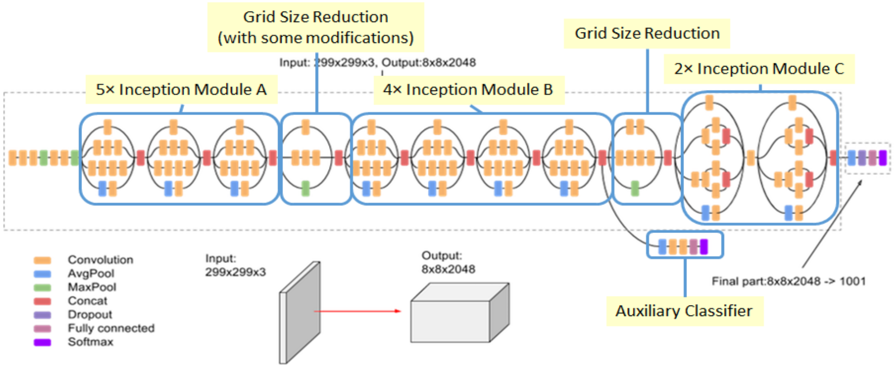


Figure 2. 6: Inception V3 Architecture

Table 2. 4: Summary, Review of Related Work

S/N	Author	Technique	Strength(s)	Weakness(es)
1	(Souza *et al.*, 2019)	Researchers used PIC16F876 Microcontroller	Sound stability and accuracy	Complexity in usage, Expensive, pH determination alone not sufficient for farming.
2	(Kumar *et al.*, 2014)	Digital Image Processing (Bayer filter technique), TNT Mips software	Was able to establish relationship between soil colour, their RGB values and their pH value	There were no measurements; soil colour, pH and RGB values were observed and mapped.
3	(Anami *et al.*, 2020)	CNN (VGG16 on paddy images)	Accuracy of classification as high as 95.08%	VGG16 has high parameters and computation cost
4	(Aziz *et al.*, 2016)	Neural Network	Model showed good performance	20% MAPE, limited pH range can be measured
5	(Kamble *et al.*, 2017)	Digital Image Processing (RGB correlation)	Large amount of data, Easy applicability of method in mobile environment	Unstable result and low precision (60% - 70%)
6	(Liang *et al.*, 2020)	Pre-trained models (AlexNet, VGG19)	High accuracy (AlexNet- 95%, VGG19- 91.8%)	Extraction of non-deep features
7	(Riese and Keller, 2019)	CNN (LucasResNet, LucasCNN)	Added to classification of hyperspectral data based on CNNs	Low accuracy of 70% on ResNet, only texture explored
8	(Utaminingrum and Robbani, 2016)	Scotect Algorithm, K-means, Segmentation, Munsell soil colour chart: for soil colour detection	High accuracy, up to 90%	Training and Test images must be taken by same camera, takes more time to process more and bigger image data. Larger neighbourhood makes image blurry.
9	(Dou *et al.*, 2016)	Hydrometer Procedure	Was able to establish that rice has higher yield under continuous	Only Rondo and Cocodrie cultivar tested

			flooding condition (clay) than in aerobic (sand condition)	
10	(Souza *et al.*, 2019)	Digital Image Processing, Multivariate Regression	Low-cost, faster, high accuracy	Soil texture not enough to determine soil usage for rice farming
11	(Shenbagavalli, 2011)	Image processing (Gray level thresholding, Low pass filter, Edge enhancement using Prewitt's Horizontal filtering)	Classification fairly accurate	Preprocessing stage time consuming
12	(Lima *et al.*, 2011)	Farmers' assessment of soil quality in rice production systems (Physical Observation: Soil Colour, Yield, Earthworm, Organic Matter)	Reasonable yield after several trials	Wastage of time and resources, low precision
13	(USAID, 2017)	Physical observation (land with good water retention capacity (contain some clay and/or organic matter)	Good yield after several trials	Inaccuracy and resource wastage
14	(Venkat, 2019)	Machine Learning (Random Forests Algorithm), increased performance with regression	Fast processing	Unreliable result
15	(Mahantesh, 2018)	Digital Image Processing (Matlab, Principal Component Analysis)	Provided a conceptual model that assists in defining and analyzing, GUI software implemented	pH determination alone not sufficient for farming
16	(Abu *et al.*, 2014)	Fuzzy logic (Matlab, input include temperature, light intensity and humidity)	Was able to create a simulation to regulate soil pH for rose plant	Instability in results

CHAPTER THREE

3.0 MATERIALS AND METHOD

3.1 Research Design

In Figure 3.1, it shows the system block diagram used to achieve the outlined objectives on soil quality detection. Major steps that led to the achievement of the objectives are defined.

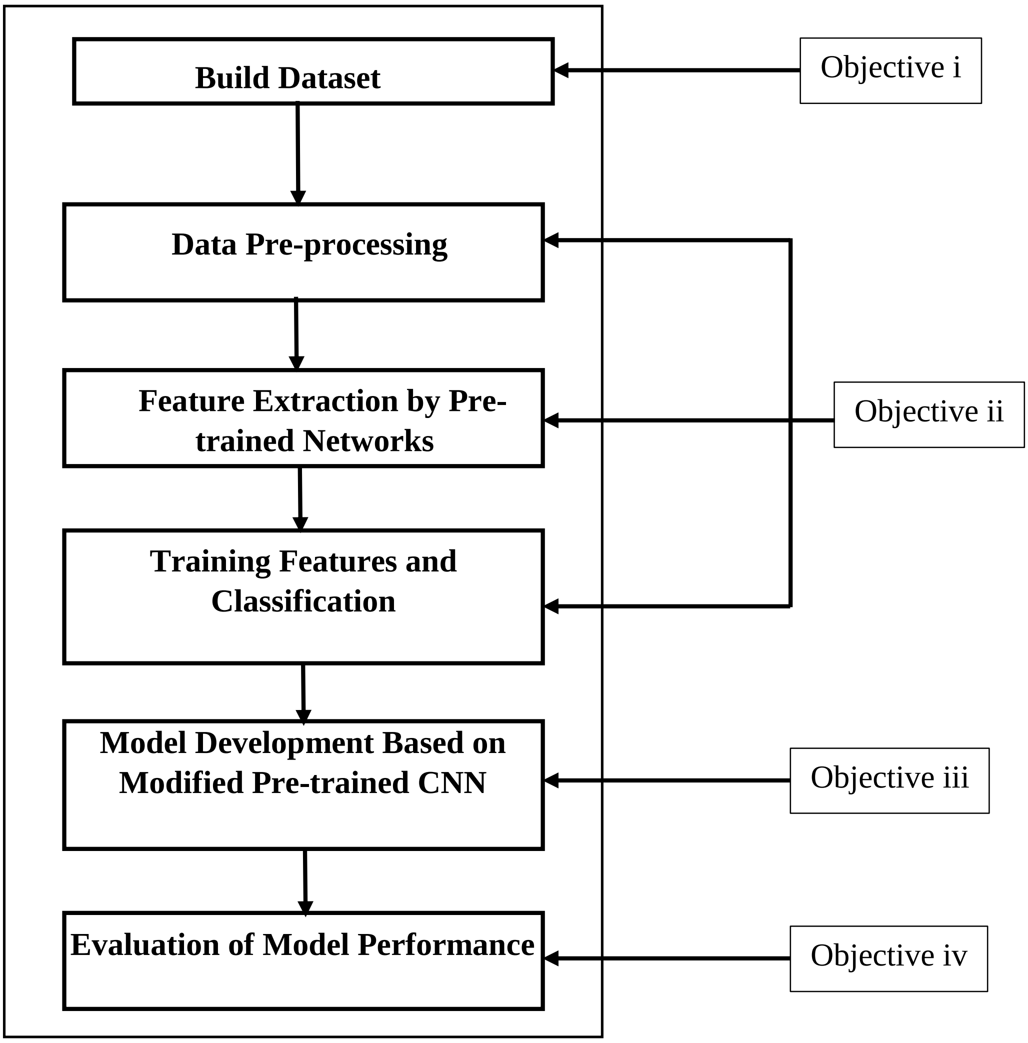

Build Dataset
Objective i
Data Pre-processing
Feature Extraction by Pre-trained Networks
Objective ii
Training Features and Classification
Model Development Based on Modified Pre-trained CNN
Objective iii
Evaluation of Model Performance
Objective iv

Figure 3.1: System Block Diagram

The image dataset is splitted into two classes. Each of the collected image data resized at the pre-processing stage to fit the requirement of the input layer of the network. Four pre-trained networks will be used for validation of results; they include Inception V3, AlexNet, ResNet18 and GoogLeNet. The features will be passed to SVM for training and classification.

3.1 Soil Dataset

Part of the soil samples were collected from Bida in Niger State, Osogbo in Osun state, Abeokuta in Ogun state and Warri in Delta state, Nigeria. Sixty-one (61) soil samples were collected observing standard procedure. Samples were taken by removing the topsoil; this means removing at least 10cm of the surface soil. The samples were analyzed in a laboratory at National Cereals Research Institute (NCRI), Badeggi Niger State, Nigeria. The pH values and textures of the samples were measured in the laboratory.
Soil sample photos were taken in-situ during the day with a digital camera of 720 x 1600 resolution and 13 megapixels. In order to improve results and make feature extraction process easier, the pictures were taken without foreign materials like stones and plants, thus making almost any part of the image potential region of interest. As seen in literatures, the pH indicates the presence of certain chemical components in the soil while the texture reflects the water retention capability of the soil. Upland rice varieties thrive in soils within the pH range of 5.5 to 6.5 and soil texture with particularly high clay content required for water retention.

After laboratory analysis of the sixty-one samples collected, sixteen were found to be good for upland rice farming while the remaining forty-five aren’t good for upland rice farming.

To improve the performance of the models and their results, additional 662 soil samples were gotten from United Stated Geological Survey (USGS) database.

Based on their pH values and texture results, the dataset is divided into two, 506 training set with 111 good for upland rice and 395 bad for upland rice farming. The second part of the dataset is 217 samples which is the testing set; 48 of them are good for upland rice farming and the remaining 169 being bad for upland rice farming. Table 3.1 shows numerical details of dataset collected.

Table 3. 1: **Soil Dataset**

Soil Dataset			
	Total	**Good for Rice**	**Bad for Rice**
Training Set (70%)	506	111	395
Testing Set (30%)	217	48	169
Total	723	129	564

A cropped section of some of the collected soil samples are displayed in Figure 3.2.

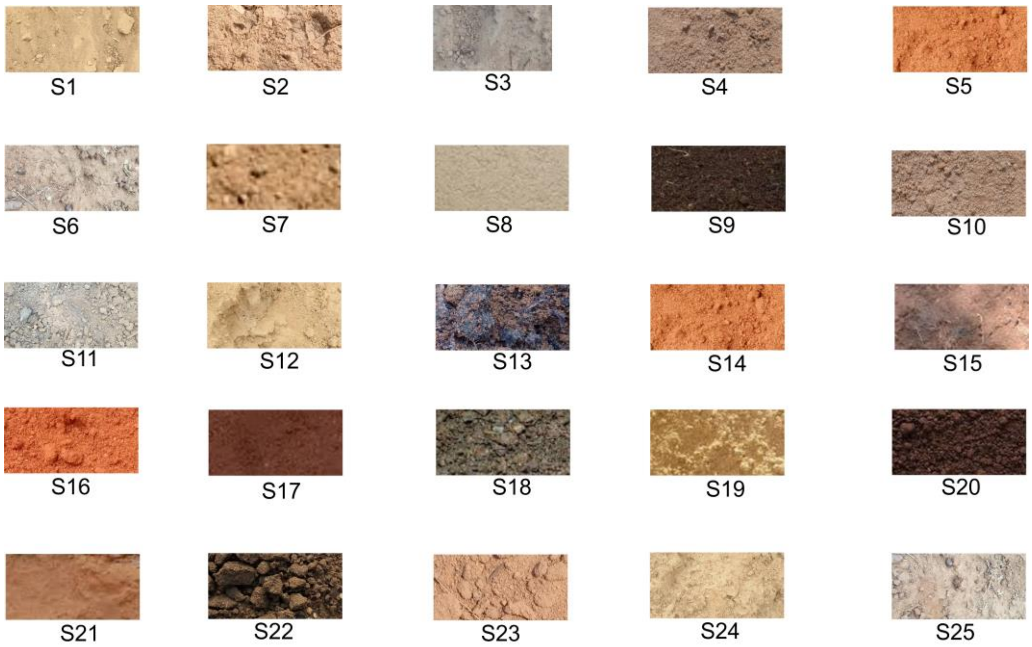


Figure 3.2: Samples, Soil Data

All samples range from clayey, loamy and sandy soils. Each image size is eventually modified to the input dimensions of the pre-trained network to be used. After the samples were collected and the images taken in-situ, the samples were taken to the laboratory for analysis Table 3.2 shows the result of the analysis of some of the samples collected.

Table 3. 2: Result of Soil Analysis (pH and Texture)

Sample	pH Value	Texture			Good for Upland Rice?
		% Sand	% Clay	%Silt	
1	7.66	95	2	3	NO, high pH, low clay
2	5.83	15	65	20	YES
3	6.67	97	2	1	NO, high pH, low clay
4	6.33	55	38	7	NO, low clay
5	6.92	71	28	1	NO, high pH, low clay
6	7.67	92	5	2	NO, high pH, low clay
7	5.89	45	51	4	YES
8	6.47	44	46	10	YES
9	6.35	54	39	7	NO, low clay
10	6.70	33	59	8	NO, high pH
11	6.95	31	62	7	NO, high pH
12	5.60	27	70	3	YES
13	6.64	87	12	1	NO, high pH, low clay
14	7.60	91	8	1	NO, high pH, low clay
15	6.33	88	11	1	NO, low clay
16	6.06	55	41	1	YES
17	7.12	97	2	1	NO, high pH
18	7.26	35	55	10	NO, high pH

The two classes here are: **YES**, for samples good for upland rice and **NO,** for samples that are not suitable for upland rice. For optimal yield, upland rice thrives in soil pH of 5.5 to 6.5 and a texture with about 40% clay (necessary for water retention).

3.2 Image Pre-processing, Feature Extraction, Training and Classification

There are various approaches to image-based data classification. A more effective approach is Transfer Learning where state of the art algorithms such as Inception V3, AlexNet, ResNet18, ResNet50, GoogLeNet and so on are used to extract features from images. These approaches use stored knowledge that is gained from solving other problems and then applying it to a different but related problem.

In this work, Inception V3, AlexNet, ResNet18 and GoogLeNet were tested on the dataset. One of the networks that performed well is then modified to give better performance and consequently the proposed model. Among the 4 models tested on the data, ResNet18 had the best result. Although ResNet18 had the best result, GoogLeNet architecture was modified due to its unique 1x1 convolutions which captures lower level features, light weight, fast training rate and portability in mobile environment.

In order to achieve better result with GoogLeNet, the inception layers were modified. Inception Module is the micro architecture in which the GoogLeNet macro architecture is built on. There is a total of 9 inception layers, each inception layer is made up of 1x1, 3x3, and 5x5 convolution filters and max pooling layer. The number convolution filters were modified to 3×3, 5×5, 7×7, 9×9, 25×25 and 28×28. Increasing the convolution filters allowed for lower-level and higher-level feature extraction from the images which in turn resulted in better model performance and higher accuracy. Furthermore, since GoogLeNet is designed to handle 1000 different object categories (classes) and millions of datasets, further modifications can be made since this study has only two categories and limited dataset, the inception layers were reduced until the model performance began to drop; the layers were therefore reduced to 5. The training parameters for the network shown in Table 3.3:

Table 3. 3: Network Parameters for Training

Parameter	Value / Name
Solver Name	SGDM
Initial Learning Rate	0.001
MiniBatchSize	150
MaxEpochs	100

3.2.1 Image pre-processing

Image pre-processing is carried out by augmenting images as they are passed into the GoogLeNet 36+--network. Images are resized to equal size. Input layer for the network is 300x300x3. 300x300 implies the width and the height of the input image, while the 3 represents the number of channels (RGB) of the image. The training images directory is loaded and their labels gotten after which they are passed into the network. After preprocessing, the images are pushed to the inception layers for extraction of features.

3.2.2 Feature extraction

The image features are extracted at the inception layers. The various convolution sizes used allows for optimum feature extractions. Also available at the inception layer is the max-pooling layer. Besides reducing spatial size, reduction in variance and computation time, the max-pooling layer is responsible for extracting the sharpest features of an image, especially as sharpest features are at the lowest representation of an image. Features extracted from all 5 inception layers are moved to the classification layer.

3.2.3 Classification

As with most CNN, the softmax layer is responsible for handling classification. Other activation functions include ReLU (no upper bound), sigmoid (ranges from 0 to 1), tanh (between -1 to 1); but softmax layer outputs a probability distribution which gives values of output that sums to 1. It is mainly used to normalize the output of a CNN to values of 0 to 1. This extra constraint helps training converge more quickly than it normally would. The softmax equation is given by:

$$h_\theta(x) = \frac{1}{1 + \exp(-\theta^T x)} \tag{3.1}$$

Here, h_θ is the scalar output of softmax,

The range of $h_\theta(x) \in R$, and $0 < h_\theta(x) < 1$

θ and x are the vectors of weights and input values respectively.

The flowchart shown at figure 3.2 summarizes the entire system procedure.

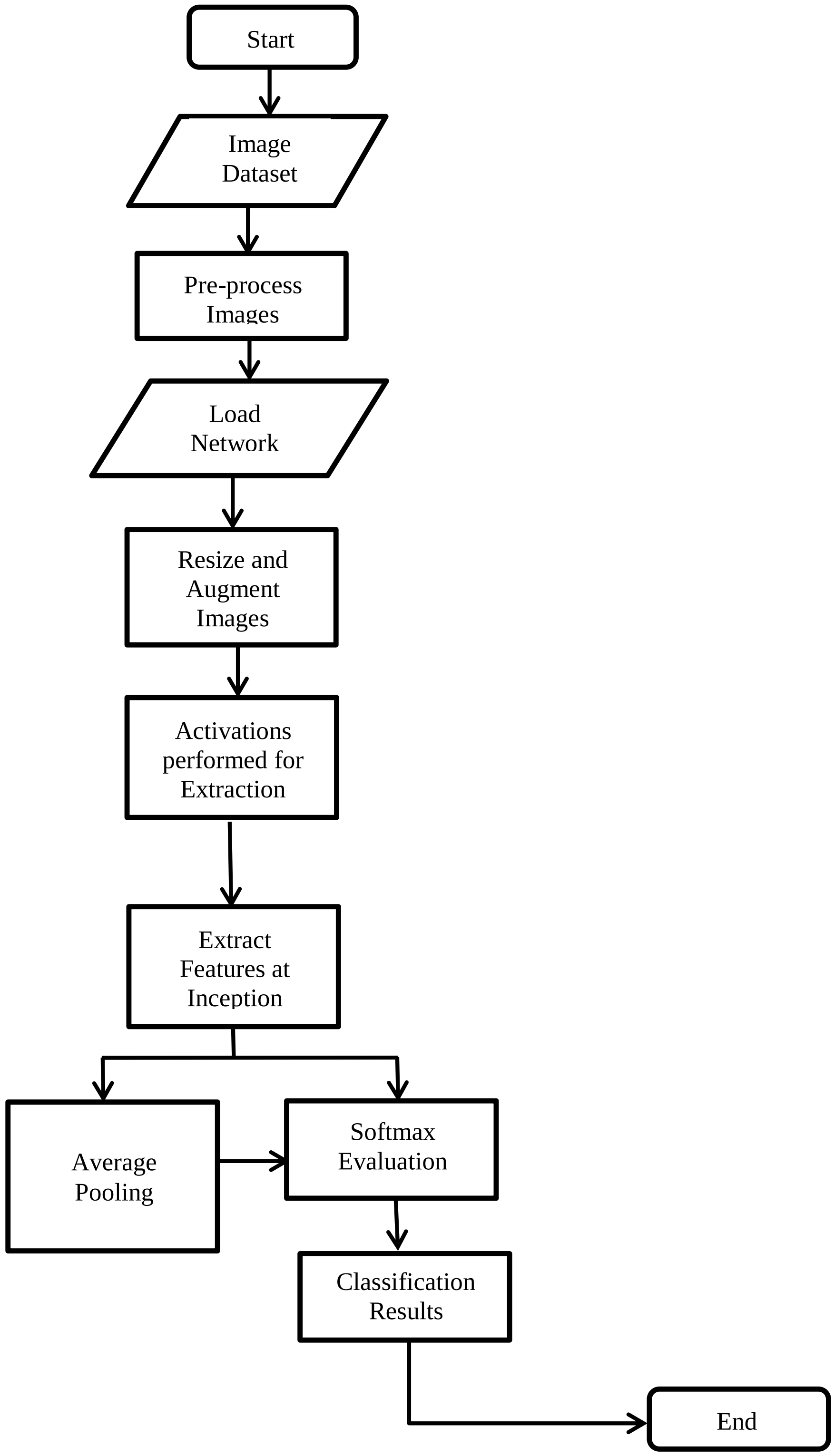


Figure 3.3: System Flowchart

3.2.4 System architecture

The architecture for both training and testing phases of the system is shown in Figure 3.4.

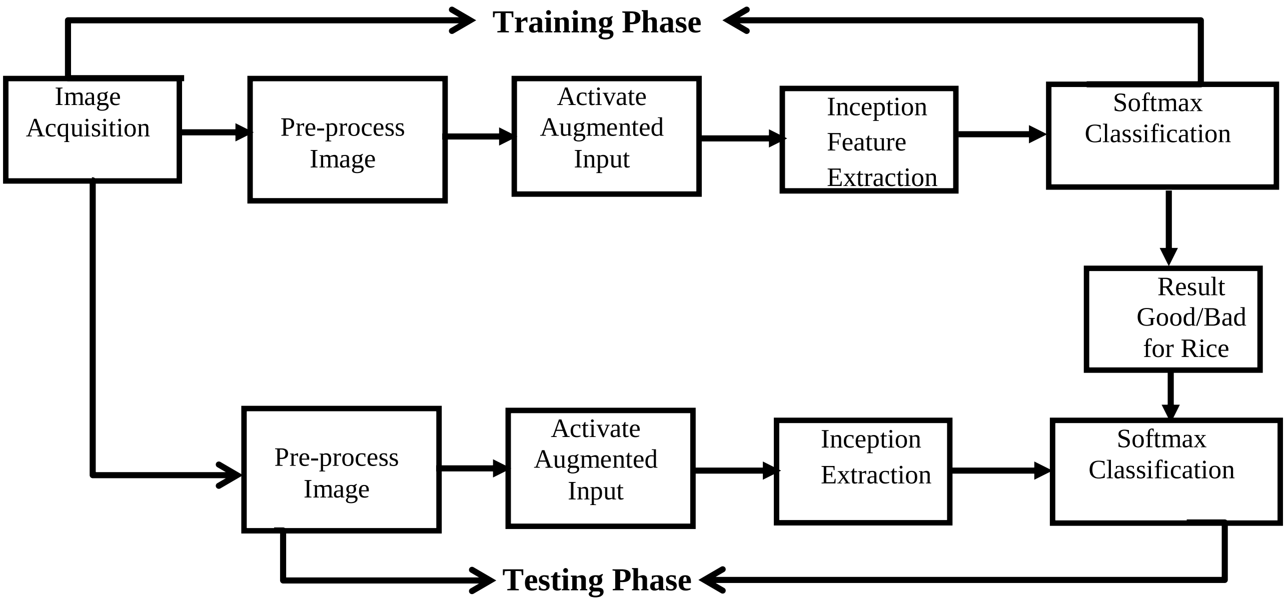


Figure 3.4: System Architecture

The image is resized to 300 x 300 at the input layer of the network over RGB colour channels. The image is augmented for better feature extraction, after which features are extracted at the max-pooling and convolution layers of the inception modules. Softmax function is responsible for probabilistic classification into one of two outputs; Yes, meaning good for rice farming and No meaning not good for rice farming.

3.3 System Model

3.3.1 Mathematical representation of soil classification based on pH and texture

Let the pH of a soil sample be represented as H

Let *T* be texture and sand, silt, and clay be *sa*, *si*, and *cl* respectively. Then,

T is a tuple such that

$T = (sa, si, cl)$ and

$sa \wedge si \wedge cl \subseteq T$

There are two classes represented by *C1* and *C2*

C1 represents the class of soil samples that meet criteria for upland rice farming

C2 represents the class of samples that do not meet the criteria for upland rice farming

Conditions for *C1* is given by:

$$sa, si, cl \in C1 \iff cl \geq 40 \quad (3.2)$$

$$H \in C1 \iff 5.5 \leq H \leq 6.5 \quad (3.3)$$

$$\therefore T \wedge H \in C1 \iff cl \geq 40 \wedge 5.5 \leq H \leq 6.5 \quad (3.4)$$

Conditions for *C2* is given by:

$$sa, si, cl \in C2 \iff cl < 40 \quad (3.5)$$

$$H \in C2 \iff H < 5.5 \vee H > 6.5 \quad (3.6)$$

$$\therefore T \wedge H \in C2 \iff cl < 40 \wedge (H < 5.5 \vee H > 6.5 \quad (3.7)$$

3.3.2 Modified GoogLeNet Model for Upland Rice Soil

There are 9 inception layers in GoogLeNet. The inception layers are the backbone of GoogLeNet architecture as they are made up of the convolution and max-pooling layers. Therefore, the inception layers of GoogLeNet are modified in this work to suite the dataset and to give the best possible result. The modified inception layer is shown in Figure 3.5.

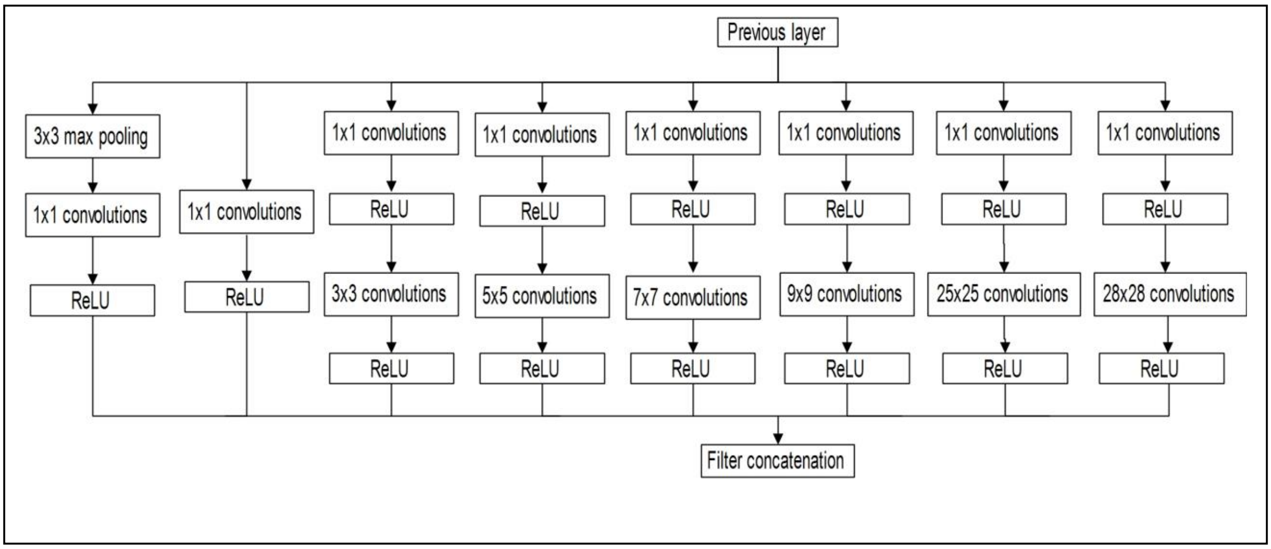


Figure 3.5: Modified Inception Module (Multiple convolutions and dimensionality reduction)

The modification was carried out in Matlab 2020a environment. Figure 3.6 shows the modification with Deep Network Designer in Matlab environment

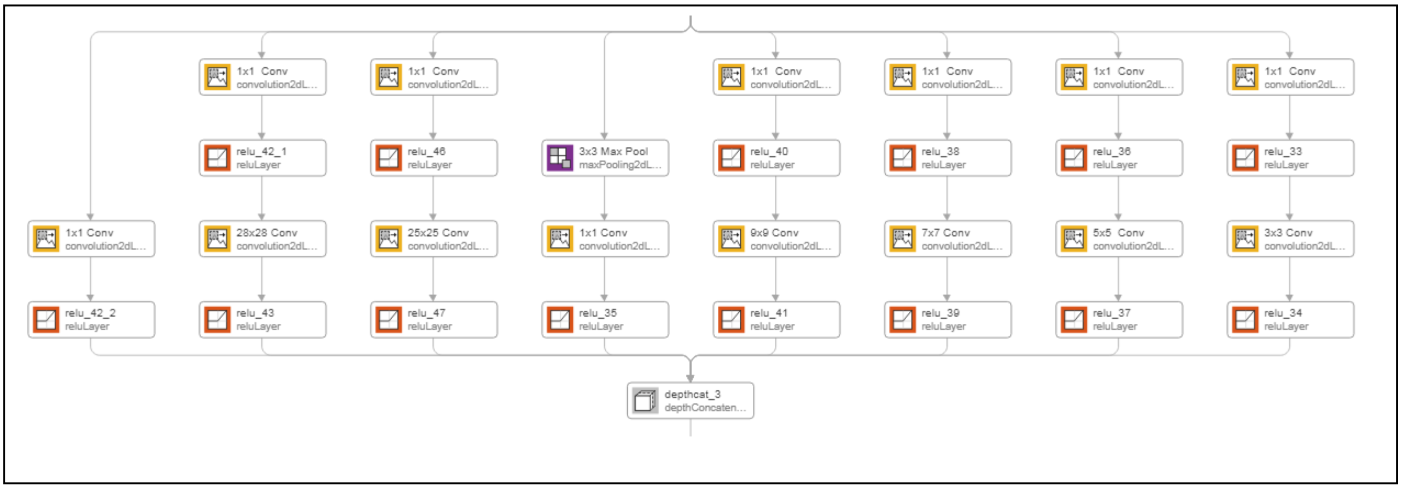


Figure 3.6: Modified Inception Module (Matlab Environment)

The original GoogleNet inception module has 1x1, 3x3 and 5x5 convolution layers, a 3x3 max-pooling layer and a concatenation layer. The inception layer has been further modified to have 1x1, 3x3, 5x5, 7x7, 9x9, 25x25 and 28x28 convolution layers. All 7 convolution layers as well as the max-pooling layer are performed in parallels to give a single concatenated output. While 1x1 learns patterns across the depth of the input, the other convolutions learn spatial patterns across all dimensional components of the input (height, width and depth). As with the original inception module, each layer extracts different kind of information. For instance, the output of the 3x3 convolution kernel gives a different information form the 5x5 kernel and the 3x3 max-pooling kernel also gives a different information. The 1x1 convolution allows for extraction of features at the most detailed level. Its main plus is that it reduces computational cost and time. To show this, for example, if we need to carry out a 3x3 convolution without 1x1 convolution on a 14x14x300, we need 30 filters. Figure 3.7 shows 3x3 filter without 1x1 convolutions.

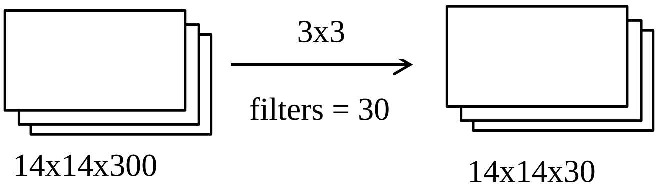


Figure 3.7: 3x3 Convolution without 1x1

This gives:

(3x3x300) x (14x14x30) operations

Which is equal to 15876000 operations, without 1x1 convolution.

Now, with the 1x1 convolution, we will be performing

(14x14x10) x (1x1x300) + (14x14x30) x (3x3x10) operations which is equal to 1117200 operations. Figure 3.8 shows the case of 3x3 with 1x1 convolutions.

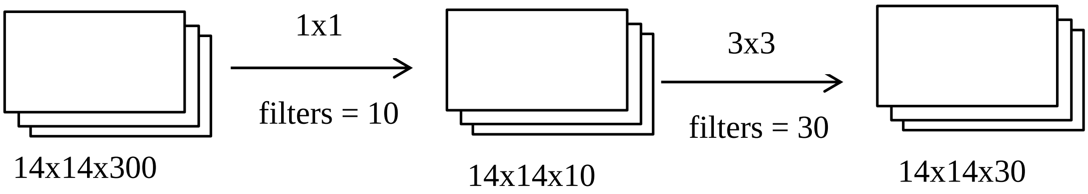


Figure 3.8: 3x3 Convolution with 1x1

Clearly, by using the 1x1 convolution, the number of operations performed in carrying out other convolutions is reduced. The 1x1 convolution also helps to reduce over-fitting problem and model size. With stacked 1x1 convolutions as seen in the modification, it allows for effective computation and more in-depth network as dimensions are reduced. The modification has multiple convolution filters; the idea is that the network will handle objects at different scales better.

The convolutional layers have ReLU (Rectified Linear Unit) layers attached to them. ReLU is an activation function. The function of the ReLU layer is to drop down any negative values from neurons to 0 while positive values remain unchanged, thereby ensuring non-linear transformation of data. ReLU is applied to the output of other functions. It speeds up training of a network as the computation results in a 0 or 1 depending on the sign of the neuron.

ReLU is mathematically is represented as:

$$\boldsymbol{y = max(0,x)} \tag{3.8}$$

The concatenation layer remains same; it is where all feature maps and general outputs and joined together as a single object for that module.

3.3.3 Modifying the GoogLeNet network

Considering that the number of convolutions in the inception module has increased, in order to mitigate the problems of complexity and computational cost of the entire model, 1x1 convolution and ReLU layers have been added to each convolution layer. However, we can further ensure better performance of the model reducing its number of inception modules. GoogLeNet was trained on 1000 object categories and with millions of image dataset. In this study there are just two categories (good for rice farming and not good for rice farming) and 762 datasets. This fact is going to be leveraged on by attempting to reduce the number of inception modules until the performance of the model begins to drop. It was observed that at 5 inception modules the network retained its best accuracy. The modified network that’s best for upland soil therefore has just 5 inception layers as against the 9 of GoogLeNet.

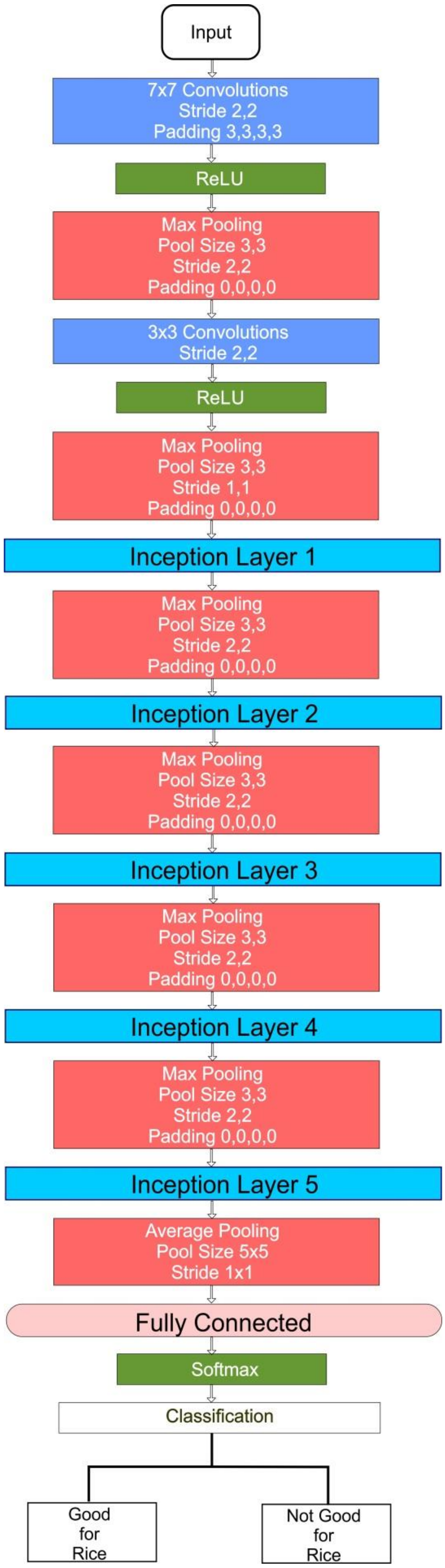
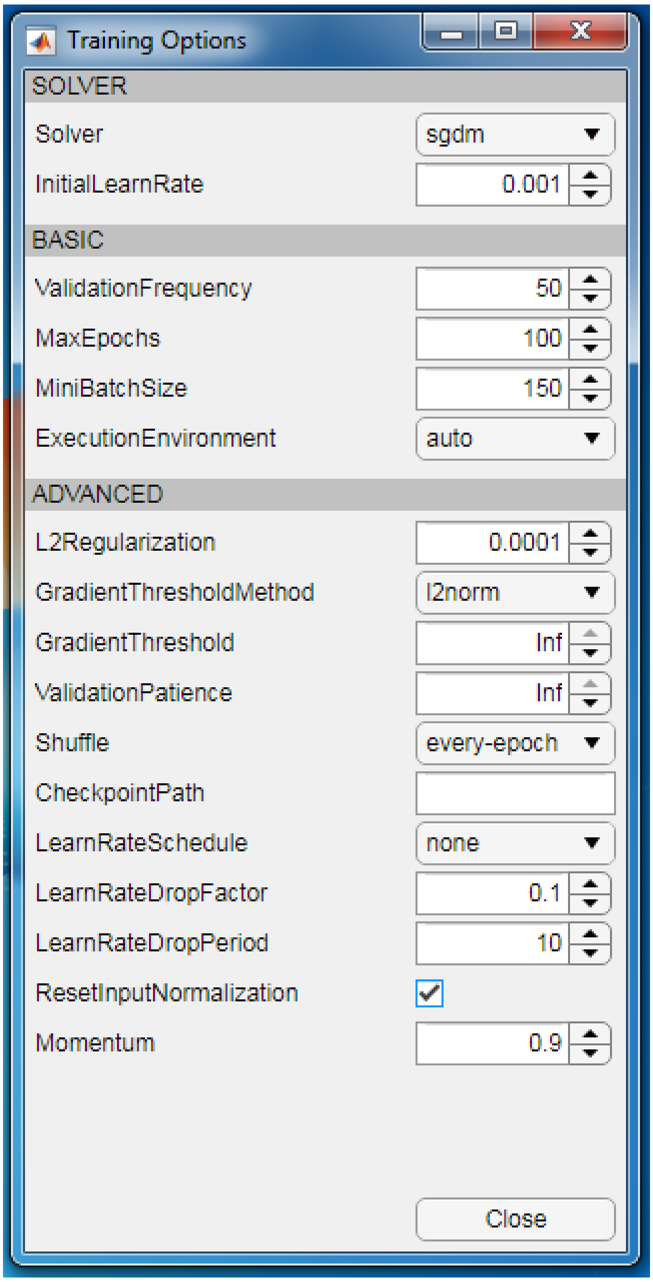


Figure 3. 9: Modified GoogLeNet Network for Soil Quality Detection

3.3.4 Matlab implementation of modified GoogLeNet network

The at the inception module and to the entire network were implement in Matlab environment. Matlab 2020a was used for the implementation.

Figure 3.10 shows training parameter fine-tuning for in Matlab environment:



SOLVER	
Solver	sgdm
InitialLearnRate	0.001
BASIC	
ValidationFrequency	50
MaxEpochs	100
MiniBatchSize	150
ExecutionEnvironment	auto
ADVANCED	
L2Regularization	0.0001
GradientThresholdMethod	l2norm
GradientThreshold	Inf
ValidationPatience	Inf
Shuffle	every-epoch
CheckpointPath	
LearnRateSchedule	none
LearnRateDropFactor	0.1
LearnRateDropPeriod	10
ResetInputNormalization	☑
Momentum	0.9

Figure 3.10: Training Parameter Fine-tuning

SDGM solver is used with an initial learning rate of 0.001. Best results were obtained at these tuning parameters.

In the Figure 3.11, it shows training data by class graph in Matlab environment:

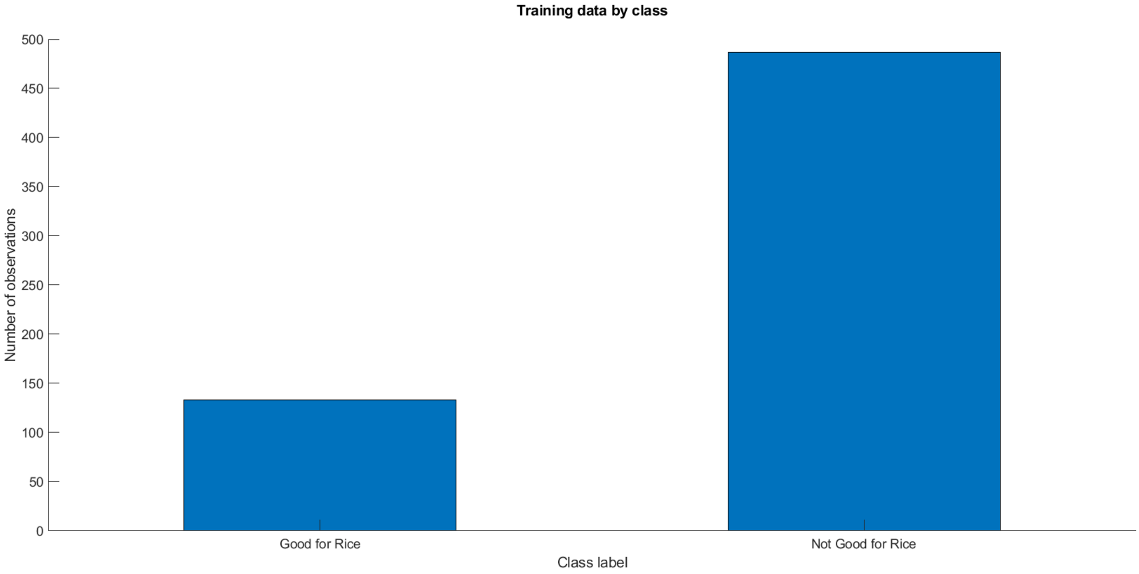


Figure 3.11: Training Data by Class

Figure 3.11 graphically summarizes the all training dataset that are good for rice farming and that are not good for rice farming.

In Figure 3.12, it shows validation data by class graph in Matlab environment:

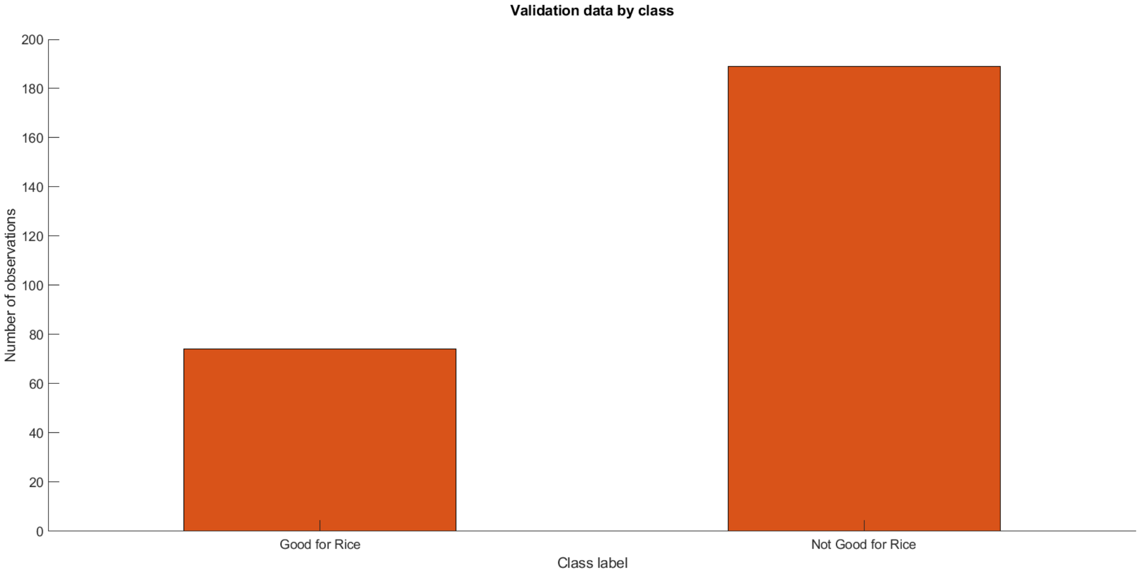


Figure 3.12: Validation Data by Class

Figure 3.11 graphically summarizes the all validation dataset that are good for rice farming and that are not good for rice farming.

CHAPTER FOUR

4.0 RESULTS AND DISCUSSIONS

4.1 Results

The various network models used on the soil data yielded varying results. AlexNet, ResNet18 and GoogLeNet were applied on training and test datasets. GoogLeNet performed better and therefore its architecture served as the basis of the creation of the model in this study.

The training and testing results for the three networks are given in Table 4.1.

Table 4. 1: Training and Validation Results with Four Algorithms

	Network				
	AlexNet	**ResNet18**	**GoogLeNet**	**Inception V3**	**Model**
Training Result	98.4%	98.7%	99.8%	100.0%	100.0%
Test/Validation Result	77.9%	92.8%	92.7%	91.8%	97.24%

The model created in this study gave a better performance over other tested network.

The classification matrix of the results of the various networks is shown form Figure 4.1 to Figure 4.11.

4.2 AlexNet Training and Testing Results

Figure 4.1 shows confusion matrix of AlexNet on training and testing dataset.

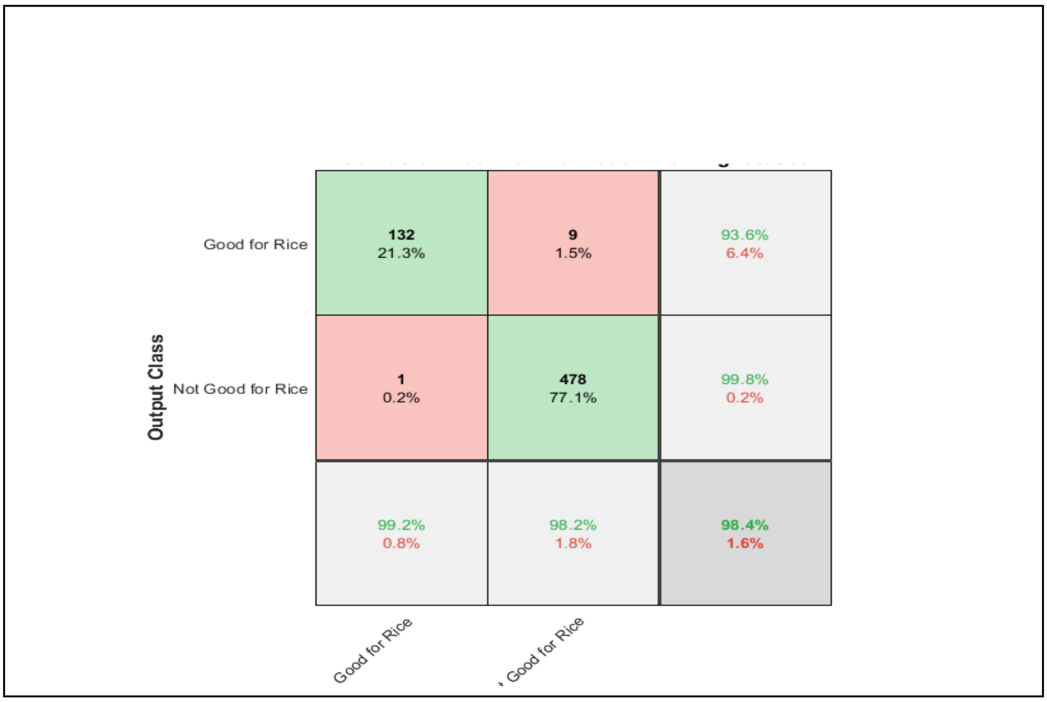


Figure 4.1: Confusion Matrix of AlexNet on Training Dataset

AlexNet training result shows 77.1% True Negative (TF), 21.3% True positive, and an overall accuracy of 98.4%. Figure 4.2 shows the confusion matrix of AlexNet Testing result.

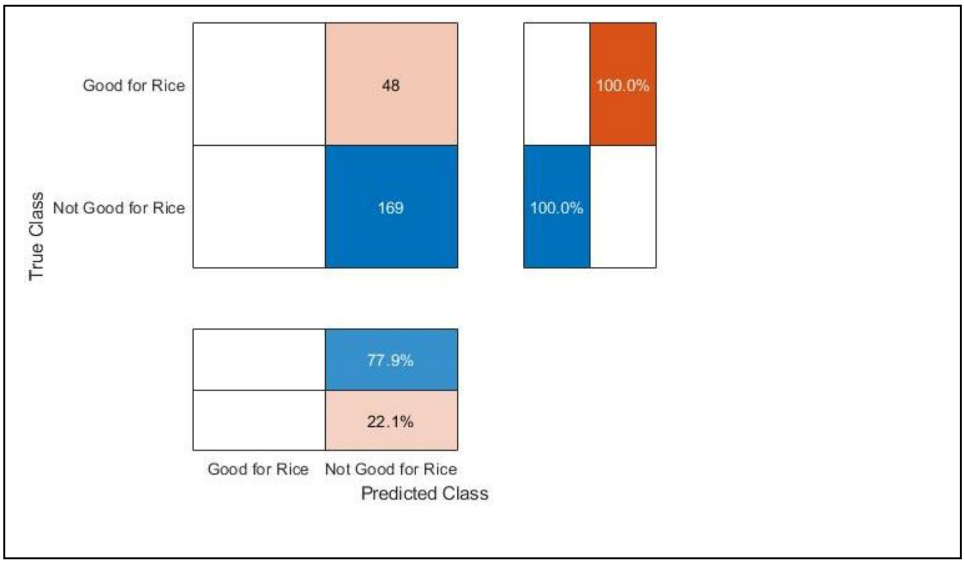
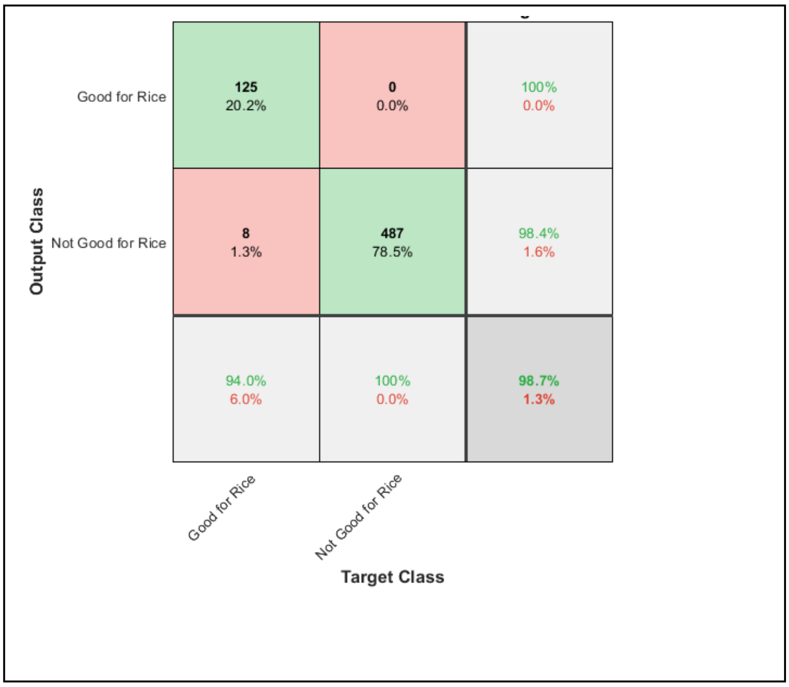


Figure 4.2: Confusion Matrix of AlexNet on Testing Dataset

AlexNet (with 8 layers) validation result shows an overall accuracy of 77.9%. The architecture has of 5 convolutional layers, 3 max-pooling layers, 2 normalization layers, 2 fully connected layers, and 1 softmax layer. It has an input layer of 277x277x3 and an overall of 60 million parameters, therefore, took longer time to train. The network had more misclassifications than other models used in this study.

4. 3 ResNet18 Training and Testing Results

The data was also run on ResNet18 and the result is shown in the figure 4.3.



Output Class

	Good for Rice	Not Good for Rice	
Good for Rice	125 20.2%	0 0.0%	100% 0.0%
Not Good for Rice	8 1.3%	487 78.5%	98.4% 1.6%
	94.0% 6.0%	100% 0.0%	98.7% 1.3%

Target Class

Figure 4.3: Confusion Matrix of ResNet18 on Training Dataset

ResNet18 training result shows 78.5% True Negative (TF), 20.2% True positive, and an overall accuracy of 98.7%. Figure 4.4 shows the confusion matrix of ResNet18 Testing result.

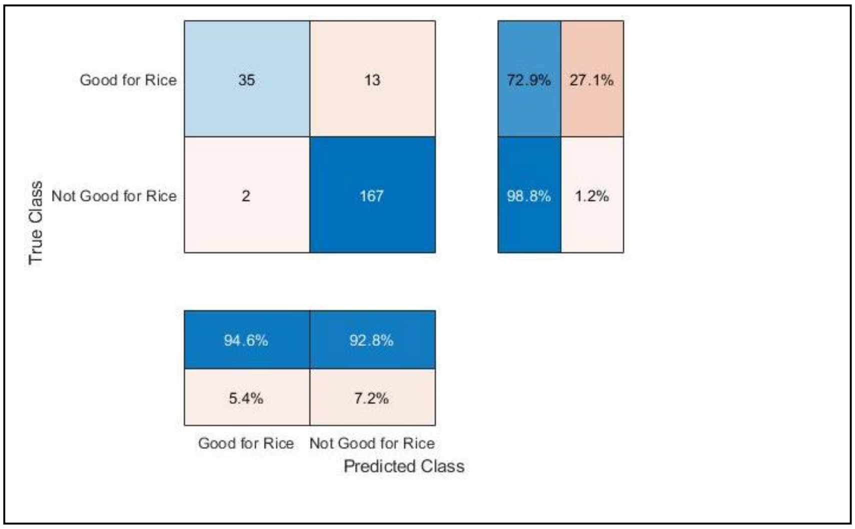


Figure 4.4: Confusion Matrix of ResNet18 on Testing Dataset

ResNet18 validation result shows an overall accuracy of 92.8%. The network has an input of 244x24x3, a depth of 18 and about 11 million number of parameters, therefore, compared with AlexNet, it took lesser time to obtain results. The network had the least misclassifications among other models used in this study.

4.4 GoogLeNet Training and Testing Results

The result of the samples when trained on GoogLeNet shown in the Figure 4.5.

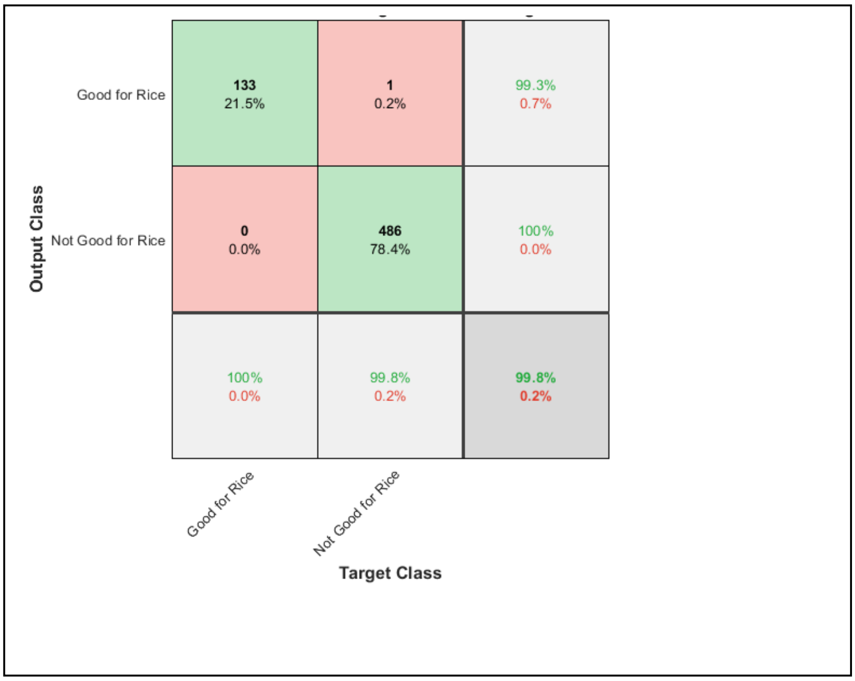


Figure 4.5: Confusion Matrix of GoogLeNet on Training Dataset

GoogLeNet training result shows 78.4% True Negative (TF), 21.5% True positive, and an overall accuracy of 99.8%. Figure 4.6 shows the confusion matrix of GoogLeNet testing result.

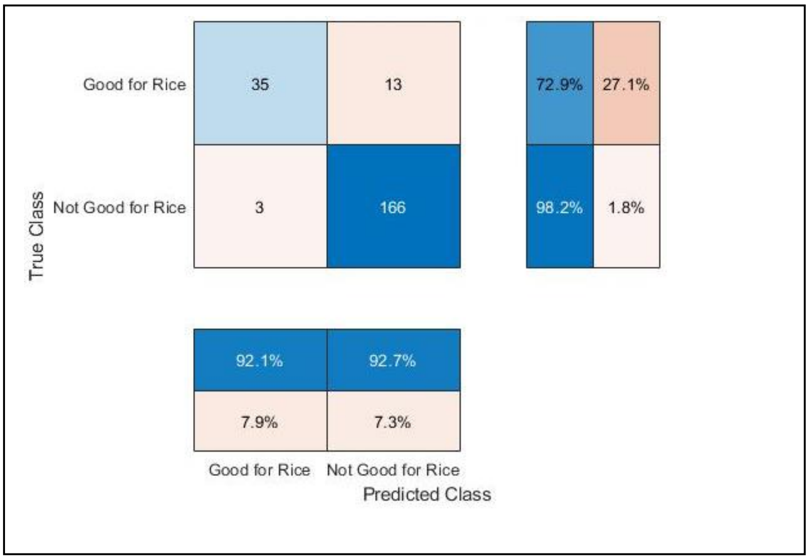


Figure 4. 6: Confusion Matrix of GoogLeNet on Testing Dataset

GoogLeNet validation result shows an overall accuracy of 92.7%. With an input layer of 244x244x3, a depth of 22 and parameters of 7 million (due to the 1x1 convolutions), it trained faster than ResNet18. The network training and validation time was fastest due to its light weight.

4.5 Inception V3 Training and Testing Results

The training result of the samples run on Inception V3 gave result shown in Figure 4.7.

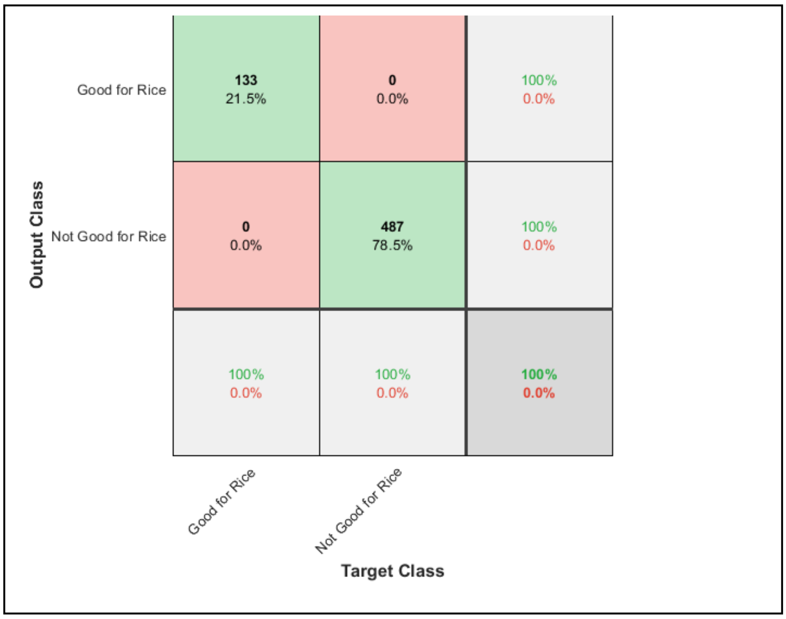


Figure 4. 7: Confusion Matrix of InceptionV3 on Training Dataset

InceptionV3 training result shows 78.5% True Negative (TF), 21.5% True positive, and an overall accuracy of 100%. Figure 4.8 shows the confusion matrix of InceptionV3 testing result.

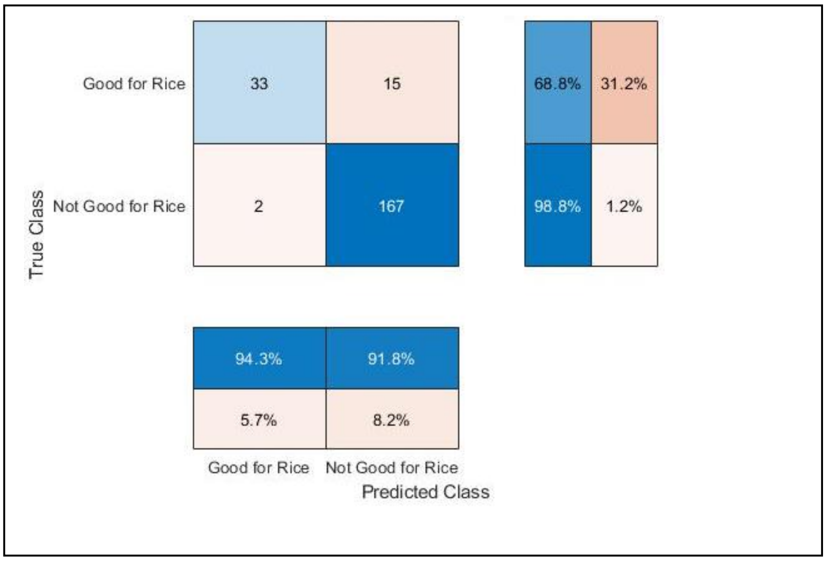
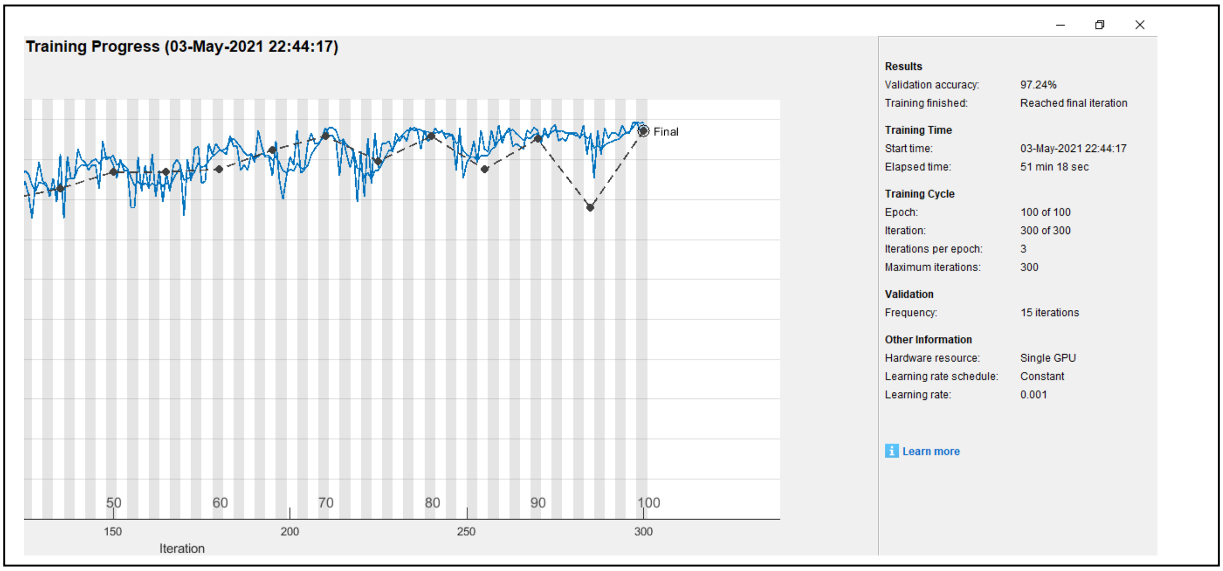


Figure 4.8: Confusion Matrix of InceptionV3 on Testing Dataset

Inception V3, a modification of GoogLeNet gave a testing result of 91.8% with the dataset. With an input of 299x299x3, 38 layers depth and factorized 7x7 convolutions in order to enhance feature detection, has a shortcoming in its number of parameters of 24 million, therefore, took longer time to achieve training and validation results.

4.6 Model Training and Validation Results

After the modifications made to GoogLeNet, the model was trained on the dataset.70% of the data was for training while the remaining 30% was for validation. The result of the training can be seen in Figure 4.9:



Training Progress (03-May-2021 22:44:17)

Results	
Validation accuracy:	97.24%
Training finished:	Reached final iteration
Training Time	
Start time:	03-May-2021 22:44:17
Elapsed time:	51 min 18 sec
Training Cycle	
Epoch:	100 of 100
Iteration:	300 of 300
Iterations per epoch:	3
Maximum iterations:	300
Validation	
Frequency:	15 iterations
Other Information	
Hardware resource:	Single GPU
Learning rate schedule:	Constant
Learning rate:	0.001

Figure 4. 9: Model Training and Validation Result

After fine-tuning the model in Matlab with the parameters in Figure 3.10, the training and validation process took about an hour. The model is 165 layers deep, this is due to the additional convolutional layers added to capture both low-level and high-level features. The training was 100% successful with a validation of 97.24%. Figure 4.10 shows a graphical representation of the results of the 4 pre-trained models and the modified

GoogLeNet model.

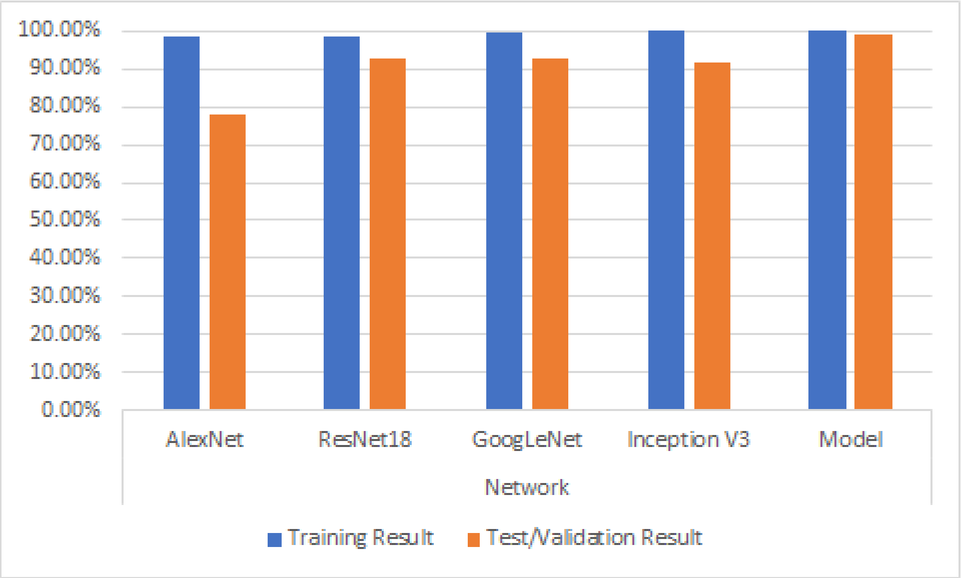


Figure 4.10: Graphical Representation of Results

Training and validation result as seen in Figure 4.10 shows that AlexNet had worst validation accuracy of 77.9% while the model performed best with an accuracy of 97.24%. Although AlexNet was the winner of 2012 ILSVRC, as at 2014 when GoogLeNet was invented, many of the flaws in AlexNet had been corrected in GoogLeNet. These include auxiliary classifiers, training speed, lower level feature extraction and lesser number of overall parameters. The model took advantage of these and modified GoogLeNet to obtained a more accurate level of feature extraction and overall classification accuracy.

4.7 VGG16 and VGG19 Testing Results

Other state-of-the-art CNN models used by other authors as explored in related studies (*Table 2.4*) shows that (Anami *et al.*, 2020) and (Liang *et al.*, 2020) used VGG16 and VGG19 respectively for land usage determination. Figure 4.10 show testing result on

VGG16 while Figure 4.11 shows testing result on VGG19

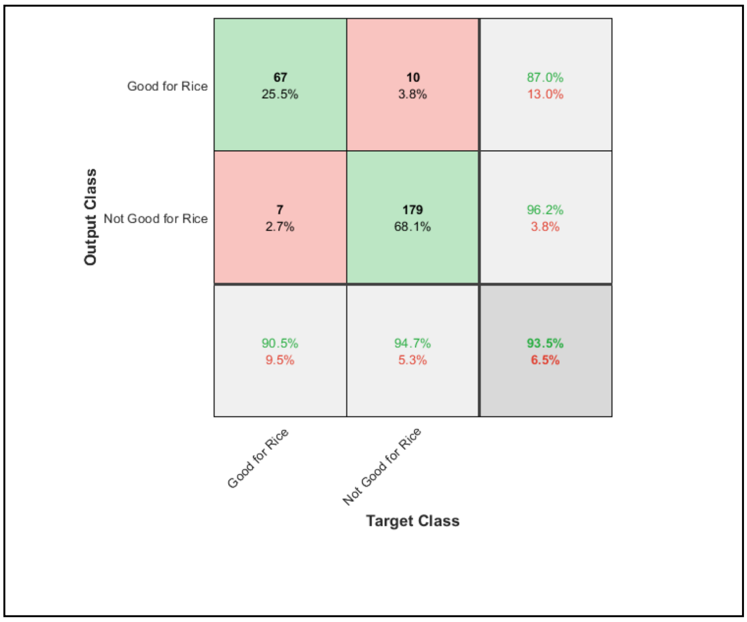


Figure 4. 11: Cross-evaluation: VGG16 on dataset

VGG16 (depth of 16) gave an overall classification accuracy of 93.5%.

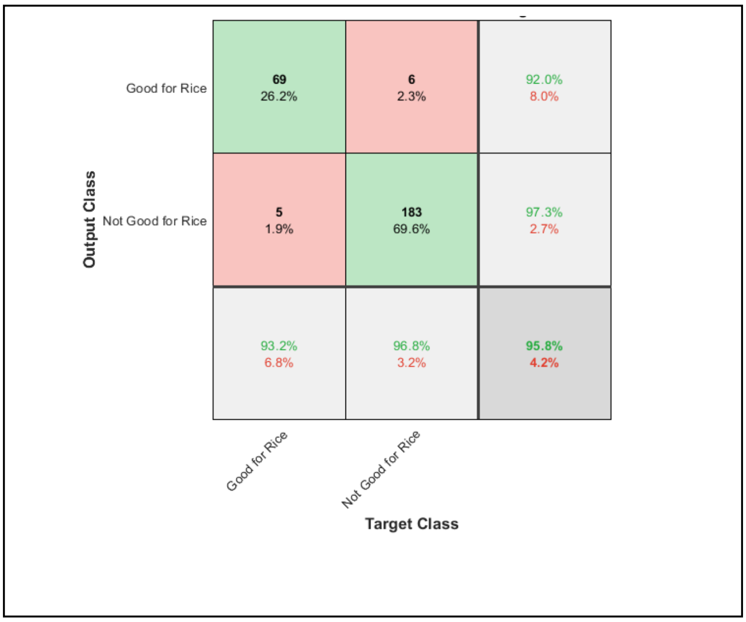


Figure 4. 12: Cross-evaluation: VGG19 on dataset

VGG19 (depth of 19) gave an overall classification accuracy of 95.8%. VGG networks are generally heavier and take longer time in training and testing. Both VGG16 and

VGG19 are very heavy. Both have 244x244x3 input layers. VGG16 has 138 million while VGG19 has 144 million parameters. Therefore, both networks had the longest training time.

4.8 Evaluation

The works of (Riese and Keller, 2019), (Anami *et al.*, 2020) and (Liang *et al.*, 2020) are three closely related work to this study. (Riese and Keller, 2019) used ResNet on Lucas dataset and had a result of 70% accuracy while (Anami *et al.*, 2020) used VGG16 model and obtained a classification accuracy of 95.08%. (Liang *et al.*, 2020) used AlexNet and VGG19 to test land usability with remote sensing and had 95% classification accuracy for AlexNet and 91.8% classification accuracy for VGG19 method.

The dataset in this study was run on these models and the following results were obtained as shown in Table 4.2.

Table 4. 2: Model Evaluation

Method(s)	**This Study**		**Other Authors**	
	Dataset	**Accuracy**	**Dataset**	**Accuracy**
ResNet18	This Study	92.8%	LUCAS (Riese and Keller, 2019)	70.0%
VGG16	This Study	93.5%	Self-generated (Anami *et al.*, 2020)	95.1%
AlexNet	This Study	77.8%	UC- Merced (Liang et) al., 2020	95.0%
VGG19	This Study	95.8%	UC- Merced (Liang et) al., 2020	91.8%

From Table 4.1, the results show that AlexNet had a classification accuracy of 77.8% this is mainly because most of the soil images are classified as false-positive by the model. Inception V3 gave a better performing result of 91.8% accuracy. It had fewer false-positives and false-negatives.

Furthermore, as seen in the Table 4.1 and in Figures 4.4 and 4.6, ResNet model performed slightly better than GoogLeNet. However, this study adopted GoogLeNet architecture. The reason is because of the peculiarity of the architecture of GoogLeNet; the model's architecture is designed in such a way that it can be deployed on mobile devices and has minimal computational parameters. After deriving the model from GoogLeNet, Figure 4.9 shows that the developed model performed better than other tested model with an accuracy of 97.24%.

The evaluation of the model was carried out by deploying other author's approaches on this study's dataset as seen in Table 4.2. The results show minimal differences in results. With high classification accuracies, the study shows that state-of-the-art CNN models can be used to solve may classification problems.

CHAPTER FIVE

5.0 CONCLUSION AND RECOMMENDATIONS

5.1 Summary

This study compared AlexNet, ResNet18, GoogLeNet, Inception V3 and additional architectures such as VGG16 and VGG19 by running 723 soil image dataset that have been collected from different sources on the pre-trained networks. The choice of pre-trained CNN has been made due to the low amount of dataset available. The dataset consists of soil images with two classes: "good for upland rice farming", and "not good for upland rice farming". The manual classification criteria of the soil samples was based primarily on laboratory results of the soil texture and pH values of each of the samples. Furthermore, the study modified GoogLeNet to achieve a unique model that performed better for the soil domain. The performance is based on accuracy of classification of image samples. The study finally compared the results and performance of the model with closely related work by other authors.

5.2 Conclusion

This study was able to develop a model that detects a soil's suitability for upland rice farming. The work leveraged on existing pre-trained algorithms which uses transfer learning. Of the four tested algorithms, ResNet18 performed better. However, GoogLeNet was modified to suit this case study. GoogLeNet inception layer and number of inception modules were modified. It was observed that increasing the number of convolutions in the inception module yielded better results. Also, the addition of 1x1 convolutions to each inception layer improved performance and reduced computational parameters. The model showed an overall accuracy of 97.24% which performed better than other tested networks. Increasing the number of convolutions also increases the chances of getting a better result, however, this usually comes with a cost of increased computation cost and complexity of network. In order to address these disadvantages, 1x1 convolutions was carried out before passing the image through every other convolution layer as seen in *fig 3.5*. This study's model has been able to shown better performance accuracy over other tested models.

5.3 Recommendation

It is recommended that further work be carried out with increased number of datasets. This would increase chances of better results. Increased training dataset would also widen the applicability of the model. Further studies could also consider including other soil parameters such as land topography and soil moisture content as factors.

5.4 CONTRIBUTION TO KNOWLEDGE

This research work was able to compare the performance of Inception V3, AlexNet, ResNet18 and GoogLeNet on the same soil image dataset. The results show that network models with deeper layers performed well. The research work was also able to develop a modified CNN model that is unique for soil domain. The model has better advantages in feature extraction and therefore had better classification accuracy the selected state-of-the-art models.

REFERENCES

Abu, M. A., Nasir, E. M. M., and Bala, C. R. (2014). Simulation of the soil pH control system using fuzzy logic method. *International Conference on Emerging Trends in Computer and Image Processing (ICETCIP2014)*, 15–19. http://psrcentre.org/images/extraimages/4 1214059.pdf

Adegbite, K. A., Okafor, M. E., Adekiya, A. O., Alori, E. T., and Adebiyi, O. T. V. (2019). Characterization and classification of soils of a toposequence in a derived savannah agroecological zone of Nigeria. *The Open Agriculture Journal*, 13(1), 44–50. https://doi.org/10.2174/1874331501913010044

AgroNigeria. (2014). Rice Farming: What you need to know. Retrieved on January 1, 2020 from https://agronigeria.ng/2014/01/10/rice-farming-what-you-need-to-know/

Aitkenhead, Coull, M., Towers, W., Hudson, G., and Black, H. I. J. (2013). Prediction of soil characteristics and colour using data from the National Soils Inventory of Scotland. *Geoderma*, 200–201, 99–107, doi:10.1016/j.geoderma.2013.02.013

Aitkenhead, M., Coull, M., Gwatkin, R., and Donnelly, D. (2016). Automated soil physical parameter assessment using smartphone and digital camera imagery. *Journal of Imaging*, 2(4). https://doi.org/10.3390/jimaging2040035

Ajit, A., Acharya, K., and Samanta, A. (2020). A review of convolutional neural networks. international conference on emerging trends in information technology and engineering. *International Conference on Emerging Trends in Information Technology and Engineering*, 2020, 1–5. https://doi.org/10.1109/ic-ETITE47903.2020.049

Akepati, D., and Kutakula, H. R. (2018). Machine vision techniques for soil detection in agricultural applications. *Journal of Semantics Scholar*, 2018, 1–6. Retrieved from https://www.semanticscholar.org/paper/Machine-Vision-Techniques-for-Soil-Detection-in-Shaik-Akepati/b1a0e7ec2a2265783e7fb70e13b821fb90e448ba

Al-Saffar, A. A., Hai, T., and Talab, M. A. (2017). Review of deep convolution neural network in image classification. *International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications*, 23(2), 123–124. https://doi.org/10.1080/02687030801943021

Al-Mashud, M. A., Uddin, M., and Serajul I. M. (2014). Design and implementation of microcontroller based digital soil pH meter. *Ulab Journal of Science and Engineering*, 5, 31–35. doi: 10.13140/2.1.4044.0645

Anami, B. S., Malvade, N. N., and Palaiah, S. (2020). Deep learning approach for recognition and classification of yield affecting paddy crop stresses using field images. *Artificial Intelligence in Agriculture*, 4, 12–20. https://doi.org/10.1016/j.aiia.2020.03.001

Aziz, M. M., Ahmed, D. R., and Ibrahim, B. F. (2016). Determination of the pH of soil by using neural network based on soil's colour. *International Journal of Advanced Research*, 6 (11),51-54. http://eprints.tiu.edu.iq/653/1/V6I11-01108.pdf

Bhaskar R. S., Ramana, V. C., and Malakondaiah, K. (2011). Design and development of microcontroller based fluoride meter. *Sensors and Transducers Journal*, 124(1), 64–71. Retrieved from https://www.sensorsportal.com/HTML/DIGEST/P_740.htm

Chandel, G. S., and Singh, P. K. (2015). Digital image processing applications in agriculture : A survey. *International Journal of Advanced Research in Computer Science and Software Engineering*, 5(3), 8–20. Retrieved from https://www.researchgate.net/publication/274418841_Digital_Image_Processing_Applications_in_Agriculture_A_Survey

De La Rosa, D., and Sobral, R. (2008). Soil quality and methods for its assessment. *Journal of Land Use and Soil Resources*, 167–200. https://doi.org/10.1007/978-1-4020-6778-5_9

Deshmukh, G. (2019). AlexNet: The architecture that challenged CNNs. Retrieved on January 13, 2020 from https://towardsdatascience.com/alexnet-the-architecture-that-challenged-cnns-e406d5297951

Dossou-Yovo, E. R., Brüggemann, N., Jesse, N., Huat, J., Ago, E. E., and Agbossou, E. K. (2016). Reducing soil CO_2 emission and improving upland rice yield with no-tillage, straw mulch and nitrogen fertilization in northern Benin. *Soil and Tillage Research*, 156(July), 44–53. https://doi.org/10.1016/j.still.2015.10.001

Dou, F., Soriano, J., Tabien, R. E., and Chen, K. (2016). Soil texture and cultivar effects on rice (Oryza sativa, L.) grain yield, yield components and water productivity in three water regimes. *PloS One*, 11(3), e0150549. https://doi.org/10.1371/journal.pone.0150549

Fageria, N. K. (2014). Communications in soil science and plant analysis nutrient management for improving upland rice productivity and sustainability. *Journal of Taylor and Francis*, 11, 37–41. https://doi.org/10.1081/CSS-120000394

For, G., (2015). Soil Constraints and Management Package (SCAMP). *Australiian Government Australian Centre for International Agricultural Research*, 11, 90. Retrieved from https://aciar.gov.au/node/9401

Hakeem, K. R., Akhtar, J., and Sabir, M. (2016). Soil science: Agricultural and environmental prospectives in soil science: *Agricultural and Environmental Prospectives*. 1, 1 - 430. https://doi.org/10.1007/978-3-319-34451-5

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer*

Vision and Pattern Recognition, 2016-December, 770–778. https://doi.org/10.1109/CVPR.2016.90

Kamble, U., Shingne, P., Kankrayane, R., Somkuwar, S., and Sandip, P. (2017). Testing of agriculture soil by digital image processing. *International Journal for Scientific Research and Development*, 5(01), 870–872. Retrieved from https://www.academia.edu/37366739/Testing_of_Agriculture_Soil_by_Digital_Image_Processing

Keesstra, S. D., Bouma, J., Wallinga, J., Tittonell, P., Smith, P., Cerdà, A., Montanarella, L., Quinton, J. N., Pachepsky, Y., Van Der Putten, W. H., Bardgett, R. D., Moolenaar, S., Mol, G., Jansen, B., and Fresco, L. O. (2016). The significance of soils and soil science towards realization of the United Nations sustainable development goals. *Soil-Corpernicus*, 2(2), 111–128. https://doi.org/10.5194/soil-2-111-2016

Kumar, V., Vimal, B. K., Kumar, R., Kumar, R., and Kumar, M. (2014). Determination of soil pH by using digital image processing technique. *Journal of Applied and Natural Science*, 6(1), 14–18. https://doi.org/10.31018/jans.v6i1.368

Liang, J., Xu, J., Shen, H., and Fang, L. (2020). Land-use classification via constrained extreme learning classifier based on cascaded deep convolutional neural networks. *European Journal of Remote Sensing*, 53(1), 219–232 https://doi.org/10.1080/22797254.2020.1809528

Lima, A. C. R., Hoogmoed, W. B., Brussaard, L., and Sacco, F. (2011). Farmers' assessment of soil quality in rice production systems. *NJAS - Wageningen Journal of Life Sciences*, 58, 31–38. https://doi.org/10.1016/j.njas.2010.08.002

Mahantesh, S. D. (2018). Analysis of Agricultural soil pH using digital image processing. *International Journal of Research in Advent Technology*, 6(8), 1812–1816. Retrieved from https://ijrat.org/downloads/Vol-6/august-2018/paper%20ID-68201809.pdf

Mahmoodi-Eshkaftaki, M., Haghighi, A., and Houshyar, E. (2020). Land suitability evaluation using image processing based on determination of soil texture–structure and soil features. *Soil Use and Management*, 36(3), 482–493. https://doi.org/10.1111/sum.12572

Maji, A., Ukwungwu, M., Nahemiah, D., Abo, M. and Bakare, S. (2017). Rice: History, research and development in Nigeria. In D. Nahemiah, I. Nkama, A. T. Maji, M. N. Ukwungwu (Eds.) *Rice in Nigeria: Traditional Recipes and Research Needs* (pp.1-10). Ronab Graphix

Mengel, K., Hütsch, B., and Kane, Y. (2006). Nitrogen fertilizer application rates on cereal crops according to available mineral and organic soil nitrogen. *European Journal of Agronomy*, 24(4), 343–348. https://doi.org/10.1016/j.eja.2005.12.001

Mohammed, U. A., Ibrahim, S., Hayatu, M., and Mohammed, F. A. (2019). Rice (Oryza Sativa L.) production in Nigeria: challenges and prospects. *Dutse Journal of Pure and Applied Sciences* (DUJOPAS), 5(2), 67–75. Retrieved from https://dujopas.com/dujopas-volume-5-no-2b-december-2019/

Mosaic Crop Nutrition. (2019). Soil pH - Nutrient Management. Retrieved on February, 02, 2020 from https://www.cropnutrition.com/nutrient-management/soil-ph

Mukherjee, A., and Lal, R. (2014). Comparison of soil quality index using three methods. *PLoS ONE*, 9(8). https://doi.org/10.1371/journal.pone.0105981

Neina, D. (2019). The Role of Soil pH in Plant Nutrition and Soil Remediation. *Applied and Environmental Soil Science*, 2019(3). https://doi.org/10.1155/2019/5794869

Nowak, P. (2015). Advances in intelligent systems and computing, Lviv Polytechnic National University, Lviv, Ukraine: Springer, Cham

Nutrition, C. (2018). Efficient fertilizer use guide for soil pH. Retrieved on February 02, 2020 from https://www.cropnutrition.com/efu-soil-ph

Oladele, S., Adeyemo, A., Awodun, M., Ajayi, A., and Fasina, A. (2019). Effects of biochar and nitrogen fertilizer on soil physicochemical properties, nitrogen use efficiency and upland rice (Oryza sativa) yield grown on an Alfisol in Southwestern Nigeria. *International Journal of Recycling of Organic Waste in Agriculture*, 8(3), 295–308. https://doi.org/10.1007/s40093-019-0251-0

Omoigui, L. O., Kamara, A. Y., and Kamai, N. (2020). Guide to rice production in northern Nigeria. *International Institute of Tropical Agriculture*. 2020, 1 - 27. https://www.iita.org/wp-content/uploads/2020/07/Guide-to-Rice-Production-in-Northern-Nigeria.pdf

Orluchukwu, J. A., Emem, A., and Omovbude, S. (2019). Effect of Agro- organic wastes and NPK fertilizer on upland rice performance in Port Harcourt, Rivers State, Nigeria. *Greener Journal of Agricultural Sciences*, 9(1), 102–109. https://doi.org/10.15580/gjas.2019.1.030919045

Osujieke, D. N., and Ezomon, I. P. (2019). Profile distribution of physical and chemical soil properties in Izombe. *Bulgarian Journal of Soil Science*. 01. https://doi.org/10.5281/zenodo.2587074

Pereira, C. S., Morais, R., and Reis, M. J. C. S. (2018). Recent advances in image processing techniques for automated harvesting purposes: A review. *2017 Intelligent Systems Conference, IntelliSys 2017,* 566–575. https://doi.org/10.1109/IntelliSys.2017.8324352

Persson, M. (2011). Image Analysis in Soil Science. *Semantic Scholar*. 05, 141–148.

Retrieved from https://www.semanticscholar.org/paper/Image-Analysis-in-Soil-Science-Persson/b73c539d35648a851d19d49107b8aa627b142d0d

Shenbagavalli R. (2011). Classification of soil textures based on laws features extracted from preprocessing images on sequential and random windows. *Bonfring International Journal of Advances in Image Processing*, 1(1), 15–18. https://doi.org/10.9756/bijaip.1004

Riese, F. M., and Keller, S. (2019). Soil texture classification with 1d convolutional neural networks based on hyperspectral data. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4(2/W5), 615–621. https://doi.org/10.5194/isprs-annals-IV-2-W5-615-2019

Saha, S. (2018). A comprehensive guide to convolutional neural networks — the ELI5 way. Retrieved on June 13. 2021 from https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Samir, S., Emary, E., El-Sayed, K., and Onsi, H. (2020). Optimization of a pre-trained AlexNet model for detecting and localizing image forgeries. *Information (Switzerland)*, 11(5). https://doi.org/10.3390/INFO11050275

Sataloff, R. T., Johns, M. M., and Kost, K. M. (2012). Soil texture classification algorithm using RGB characteristics of soil images. *Journal of the Faculty of Agriculture, Kyushu University*, 57(2), 5. https://doi.org/10.3182/20101206-3-JP-3009.00005

Shemahonge, M. I. (2013). Improving upland rice (Oryza Sativa L.) Perfomance through enhanced soil fertility and water conservation methods at Ukiriguru Mwanza, Tanzania. Sokoine *University of Agriculture. Morogoro, Tanzania*. http://www.suaire.sua.ac.tz/handle/123456789/554

Nevada, Reno. (2021). Soil Properties, Part 3 of 3: Chemical Characteristics | Extension University of Nevada, Reno. (2021). 5(Figure 2), 2–5. Retrieved from https://extension.unr.edu/publication.aspx?PubID=2749

Souza, D., Augusto, P., Morais, D. O., Souza, D. M. De, Thaís, M., and Carvalho, D. M. (2019). Predicting soil texture using image analysis. *Microchemical Journal*, 146(May), 455–463. https://doi.org/10.1016/j.microc.2019.01.009

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07-12-June-2015, 1–9. https://doi.org/10.1109/CVPR.2015.7298594

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. *Proceedings of the IEEE Computer*

Society Conference on Computer Vision and Pattern Recognition, 2016-Decem, 2818–2826. https://doi.org/10.1109/CVPR.2016.308

Tsang, S.-H. (2018). Review: Inception-v3 — 1st runner up (Image Classification) in ILSVRC 2015. Retrieved on June 14. 2021 from https://sh-tsang.medium.com/review-inception-v3-1st-runner-up-image-classification-in-ilsvrc-2015-17915421f77c

USAID. (2017). Growing upland rice in Nigeria. *Africe Rice Center (WARDA) Growth (Lakeland)*, 7, 2017. Retrieved from https://www.fao.org/fileadmin/user_upload/ivc/docs/uplandrice.pdf

Usman, S., and Kundiri, A. M. (2016). Role of Soil Science: an answer to sustainable crop production for economic development in sub-Saharan Africa. *International Journal of Soil Science*, 11(2), 61–70. https://doi.org/10.3923/ijss.2016.61.70

Utaminingrum, F., and Robbani, I. H. (2016). Scotect algorithm: A novel approach for soil color detection process using five steps algorithm. *International Journal of Innovative Computing, Information and Control*, 12(5), 1645–1653. http://www.ijicic.org/ijicic-120517.pdf

Venkat R. N., (2019). Prediction of soil quality using machine leaning techniques. *International Journal of Scientific and Technology Research*, 8(11), 1309–1313. http://www.ijstr.org/final-print/nov2019/Prediction-Of-Soil-Quality-Using-Machine-Leaning-Techniques.pdf

Zhou, W., Lv, T. F., Chen, Y., Westby, A. P., and Ren, W. J. (2014). Soil physicochemical and biological properties of paddy-upland rotation: *A review. Scientific World Journal*, 2014. https://doi.org/10.1155/2014/856352

APPENDIX

Network Model Code

```
%%Create Layer Graph
%%Create the layer graph variable to contain the network layers.
lgraph = layerGraph();
%%Add Layer Branches
%%Add the branches of the network to the layer graph. Each branch is a linear array of layers.
tempLayers = [
   imageInputLayer([300 300 3],"Name","imageinput","Normalization","none")
   convolution2dLayer([7 7],64,"Name","conv_7","Padding",[3 3 3 3],"Stride",[2 2])
   reluLayer("Name","relu_7")
   maxPooling2dLayer([3 3],"Name","maxpool_2","Stride",[2 2])
   convolution2dLayer([3 3],128,"Name","conv_8","Padding","same")
   reluLayer("Name","relu_8")
   maxPooling2dLayer([3 3],"Name","maxpool_4_1")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
   convolution2dLayer([1 1],20,"Name","conv_16","Padding","same")
   reluLayer("Name","relu_16")
   convolution2dLayer([25 25],20,"Name","conv_17","Padding","same")
   reluLayer("Name","relu_17")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
   convolution2dLayer([1 1],20,"Name","conv_12_1","Padding","same")
   reluLayer("Name","relu_12_1")
   convolution2dLayer([28 28],18,"Name","conv_13","Padding","same")
   reluLayer("Name","relu_13")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
   convolution2dLayer([1 1],20,"Name","conv_4","Padding","same")
   reluLayer("Name","relu_4")
   convolution2dLayer([5 5],18,"Name","conv_5","Padding","same")
   reluLayer("Name","relu_5")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
   maxPooling2dLayer([3 3],"Name","maxpool_1","Padding",[1 1 1 1])
   convolution2dLayer([1 1],32,"Name","conv_3","Padding","same")
   reluLayer("Name","relu_3")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
   convolution2dLayer([1 1],20,"Name","conv_10","Padding","same")
   reluLayer("Name","relu_10")
   convolution2dLayer([9 9],18,"Name","conv_11","Padding","same")
   reluLayer("Name","relu_11")];
lgraph = addLayers(lgraph,tempLayers);
```

```
tempLayers = [
    convolution2dLayer([1 1],20,"Name","conv_12_2","Padding","same")
    reluLayer("Name","relu_12_2")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    convolution2dLayer([1 1],20,"Name","conv_1","Padding","same")
    reluLayer("Name","relu_1")
    convolution2dLayer([3 3],18,"Name","conv_2","Padding","same")
    reluLayer("Name","relu_2")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    convolution2dLayer([1 1],20,"Name","conv_6","Padding","same")
    reluLayer("Name","relu_6")
    convolution2dLayer([7 7],18,"Name","conv_9","Padding","same")
    reluLayer("Name","relu_9")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    depthConcatenationLayer(8,"Name","depthcat_1")
    maxPooling2dLayer([3 3],"Name","maxpool_4_2","Stride",[2 2])];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    convolution2dLayer([1 1],20,"Name","conv_31","Padding","same")
    reluLayer("Name","relu_31")
    convolution2dLayer([25 25],40,"Name","conv_32","Padding","same")
    reluLayer("Name","relu_32")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    convolution2dLayer([1 1],20,"Name","conv_27_2","Padding","same")
    reluLayer("Name","relu_27_2")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    convolution2dLayer([1 1],20,"Name","conv_27_1","Padding","same")
    reluLayer("Name","relu_27_1")
    convolution2dLayer([28 28],36,"Name","conv_28","Padding","same")
    reluLayer("Name","relu_28")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    convolution2dLayer([1 1],20,"Name","conv_25","Padding","same")
    reluLayer("Name","relu_25")
    convolution2dLayer([9 9],36,"Name","conv_26","Padding","same")
    reluLayer("Name","relu_26")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    convolution2dLayer([1 1],20,"Name","conv_23","Padding","same")
    reluLayer("Name","relu_23")
    convolution2dLayer([7 7],36,"Name","conv_24","Padding","same")
    reluLayer("Name","relu_24")];
```

```
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    convolution2dLayer([1 1],20,"Name","conv_18","Padding","same")
    reluLayer("Name","relu_18")
    convolution2dLayer([3 3],36,"Name","conv_19","Padding","same")
    reluLayer("Name","relu_19")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    convolution2dLayer([1 1],20,"Name","conv_21","Padding","same")
    reluLayer("Name","relu_21")
    convolution2dLayer([5 5],36,"Name","conv_22","Padding","same")
    reluLayer("Name","relu_22")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    maxPooling2dLayer([3 3],"Name","maxpool_3","Padding",[1 1 1 1])
    convolution2dLayer([1 1],32,"Name","conv_20","Padding","same")
    reluLayer("Name","relu_20")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    depthConcatenationLayer(8,"Name","depthcat_2")
    maxPooling2dLayer([3 3],"Name","maxpool_4_3","Stride",[2 2])];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    convolution2dLayer([1 1],20,"Name","conv_42_2","Padding","same")
    reluLayer("Name","relu_42_2")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    convolution2dLayer([1 1],20,"Name","conv_42_1","Padding","same")
    reluLayer("Name","relu_42_1")
    convolution2dLayer([28 28],73,"Name","conv_43","Padding","same")
    reluLayer("Name","relu_43")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    convolution2dLayer([1 1],20,"Name","conv_46","Padding","same")
    reluLayer("Name","relu_46")
    convolution2dLayer([25 25],74,"Name","conv_47","Padding","same")
    reluLayer("Name","relu_47")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    maxPooling2dLayer([3 3],"Name","maxpool_4","Padding",[1 1 1 1])
    convolution2dLayer([1 1],32,"Name","conv_35","Padding","same")
    reluLayer("Name","relu_35")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    convolution2dLayer([1 1],20,"Name","conv_40","Padding","same")
    reluLayer("Name","relu_40")
    convolution2dLayer([9 9],73,"Name","conv_41","Padding","same")
    reluLayer("Name","relu_41")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    convolution2dLayer([1 1],20,"Name","conv_38","Padding","same")
    reluLayer("Name","relu_38")
    convolution2dLayer([7 7],73,"Name","conv_39","Padding","same")
    reluLayer("Name","relu_39")];
lgraph = addLayers(lgraph,tempLayers);

tempLayers = [
    convolution2dLayer([1 1],20,"Name","conv_36","Padding","same")
    reluLayer("Name","relu_36")
    convolution2dLayer([5 5],73,"Name","conv_37","Padding","same")
    reluLayer("Name","relu_37")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    convolution2dLayer([1 1],20,"Name","conv_33","Padding","same")
    reluLayer("Name","relu_33")
    convolution2dLayer([3 3],73,"Name","conv_34","Padding","same")
    reluLayer("Name","relu_34")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    depthConcatenationLayer(8,"Name","depthcat_3")
    maxPooling2dLayer([3 3],"Name","maxpool_4_4_1","Stride",[2 2])];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    maxPooling2dLayer([3 3],"Name","maxpool_5_1","Padding",[1 1 1 1])
    convolution2dLayer([1 1],32,"Name","conv_50_1","Padding","same")
    reluLayer("Name","relu_50_1")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    convolution2dLayer([1 1],20,"Name","conv_51_1","Padding","same")
    reluLayer("Name","relu_51_1")
    convolution2dLayer([5 5],146,"Name","conv_52_1","Padding","same")
    reluLayer("Name","relu_52_1")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    convolution2dLayer([1 1],20,"Name","conv_55_1","Padding","same")
    reluLayer("Name","relu_55_1")
    convolution2dLayer([9 9],146,"Name","conv_56_1","Padding","same")
    reluLayer("Name","relu_56_1")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    convolution2dLayer([1 1],20,"Name","conv_53_2_1","Padding","same")
    reluLayer("Name","relu_53_2_1")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    convolution2dLayer([1 1],20,"Name","conv_57_1","Padding","same")
    reluLayer("Name","relu_57_1")
    convolution2dLayer([28 28],146,"Name","conv_58_1","Padding","same")
    reluLayer("Name","relu_58_1")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    convolution2dLayer([1 1],20,"Name","conv_48_1","Padding","same")
    reluLayer("Name","relu_48_1")
    convolution2dLayer([3 3],146,"Name","conv_49_1","Padding","same")
    reluLayer("Name","relu_49_1")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    convolution2dLayer([1 1],20,"Name","conv_61_1","Padding","same")
    reluLayer("Name","relu_61_1")
    convolution2dLayer([25 25],148,"Name","conv_62_1","Padding","same")
    reluLayer("Name","relu_62_1")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    convolution2dLayer([1 1],20,"Name","conv_53_1_1","Padding","same")
    reluLayer("Name","relu_53_1_1")
    convolution2dLayer([7 7],146,"Name","conv_54_1","Padding","same")
    reluLayer("Name","relu_54_1")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    depthConcatenationLayer(8,"Name","depthcat_4_1")
    maxPooling2dLayer([3 3],"Name","maxpool_4_4_2","Stride",[2 2])];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    convolution2dLayer([1 1],20,"Name","conv_57_2","Padding","same")
    reluLayer("Name","relu_57_2")
    convolution2dLayer([28 28],146,"Name","conv_58_2","Padding","same")
    reluLayer("Name","relu_58_2")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    convolution2dLayer([1 1],20,"Name","conv_51_2","Padding","same")
    reluLayer("Name","relu_51_2")
    convolution2dLayer([5 5],146,"Name","conv_52_2","Padding","same")
    reluLayer("Name","relu_52_2")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    convolution2dLayer([1 1],20,"Name","conv_53_2_2","Padding","same")
    reluLayer("Name","relu_53_2_2")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    convolution2dLayer([1 1],20,"Name","conv_53_1_2","Padding","same")
    reluLayer("Name","relu_53_1_2")
    convolution2dLayer([7 7],146,"Name","conv_54_2","Padding","same")
    reluLayer("Name","relu_54_2")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    convolution2dLayer([1 1],20,"Name","conv_48_2","Padding","same")
    reluLayer("Name","relu_48_2")
    convolution2dLayer([3 3],146,"Name","conv_49_2","Padding","same")
    reluLayer("Name","relu_49_2")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    maxPooling2dLayer([3 3],"Name","maxpool_5_2","Padding",[1 1 1 1])
    convolution2dLayer([1 1],32,"Name","conv_50_2","Padding","same")
    reluLayer("Name","relu_50_2")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    convolution2dLayer([1 1],20,"Name","conv_55_2","Padding","same")
    reluLayer("Name","relu_55_2")
    convolution2dLayer([9 9],146,"Name","conv_56_2","Padding","same")
    reluLayer("Name","relu_56_2")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    convolution2dLayer([1 1],20,"Name","conv_61_2","Padding","same")
    reluLayer("Name","relu_61_2")
    convolution2dLayer([25 25],148,"Name","conv_62_2","Padding","same")
    reluLayer("Name","relu_62_2")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    depthConcatenationLayer(8,"Name","depthcat_4_2")
    averagePooling2dLayer([5 5],"Name","avgpool2d","Padding","same")
    fullyConnectedLayer(2,"Name","fc")
    softmaxLayer("Name","output")
    classificationLayer("Name","classoutput")];
lgraph = addLayers(lgraph,tempLayers);
% clean up helper variable
clear tempLayers;

%%Connect Layer Branches
%%Connect all the branches of the network to create the network graph.
lgraph = connectLayers(lgraph,"maxpool_4_1","conv_16");
lgraph = connectLayers(lgraph,"maxpool_4_1","conv_12_1");
lgraph = connectLayers(lgraph,"maxpool_4_1","conv_4");
lgraph = connectLayers(lgraph,"maxpool_4_1","maxpool_1");
lgraph = connectLayers(lgraph,"maxpool_4_1","conv_10");
lgraph = connectLayers(lgraph,"maxpool_4_1","conv_12_2");
lgraph = connectLayers(lgraph,"maxpool_4_1","conv_1");
lgraph = connectLayers(lgraph,"maxpool_4_1","conv_6");
lgraph = connectLayers(lgraph,"relu_13","depthcat_1/in6");
lgraph = connectLayers(lgraph,"relu_12_2","depthcat_1/in7");
lgraph = connectLayers(lgraph,"relu_3","depthcat_1/in1");
lgraph = connectLayers(lgraph,"relu_2","depthcat_1/in2");
lgraph = connectLayers(lgraph,"relu_5","depthcat_1/in3");
lgraph = connectLayers(lgraph,"relu_11","depthcat_1/in5");
lgraph = connectLayers(lgraph,"relu_17","depthcat_1/in8");
lgraph = connectLayers(lgraph,"relu_9","depthcat_1/in4");
lgraph = connectLayers(lgraph,"maxpool_4_2","conv_31");
lgraph = connectLayers(lgraph,"maxpool_4_2","conv_27_2");
lgraph = connectLayers(lgraph,"maxpool_4_2","conv_27_1");
lgraph = connectLayers(lgraph,"maxpool_4_2","conv_25");
lgraph = connectLayers(lgraph,"maxpool_4_2","conv_23");
lgraph = connectLayers(lgraph,"maxpool_4_2","conv_18");
lgraph = connectLayers(lgraph,"maxpool_4_2","conv_21");
lgraph = connectLayers(lgraph,"maxpool_4_2","maxpool_3");
lgraph = connectLayers(lgraph,"relu_32","depthcat_2/in8");
lgraph = connectLayers(lgraph,"relu_26","depthcat_2/in5");
lgraph = connectLayers(lgraph,"relu_28","depthcat_2/in6");
lgraph = connectLayers(lgraph,"relu_24","depthcat_2/in4");
lgraph = connectLayers(lgraph,"relu_27_2","depthcat_2/in7");
lgraph = connectLayers(lgraph,"relu_22","depthcat_2/in3");
lgraph = connectLayers(lgraph,"relu_20","depthcat_2/in1");
lgraph = connectLayers(lgraph,"relu_19","depthcat_2/in2");
lgraph = connectLayers(lgraph,"maxpool_4_3","conv_42_2");
lgraph = connectLayers(lgraph,"maxpool_4_3","conv_42_1");
lgraph = connectLayers(lgraph,"maxpool_4_3","conv_46");
lgraph = connectLayers(lgraph,"maxpool_4_3","maxpool_4");
lgraph = connectLayers(lgraph,"maxpool_4_3","conv_40");
lgraph = connectLayers(lgraph,"maxpool_4_3","conv_38");
lgraph = connectLayers(lgraph,"maxpool_4_3","conv_36");
lgraph = connectLayers(lgraph,"maxpool_4_3","conv_33");
lgraph = connectLayers(lgraph,"relu_42_2","depthcat_3/in7");
lgraph = connectLayers(lgraph,"relu_43","depthcat_3/in6");
lgraph = connectLayers(lgraph,"relu_41","depthcat_3/in5");
lgraph = connectLayers(lgraph,"relu_47","depthcat_3/in8");
lgraph = connectLayers(lgraph,"relu_34","depthcat_3/in2");
lgraph = connectLayers(lgraph,"relu_39","depthcat_3/in4");
lgraph = connectLayers(lgraph,"relu_37","depthcat_3/in3");
lgraph = connectLayers(lgraph,"relu_35","depthcat_3/in1");
lgraph = connectLayers(lgraph,"maxpool_4_4_1","maxpool_5_1");
lgraph = connectLayers(lgraph,"maxpool_4_4_1","conv_51_1");
lgraph = connectLayers(lgraph,"maxpool_4_4_1","conv_55_1");
lgraph = connectLayers(lgraph,"maxpool_4_4_1","conv_53_2_1");
lgraph = connectLayers(lgraph,"maxpool_4_4_1","conv_57_1");
lgraph = connectLayers(lgraph,"maxpool_4_4_1","conv_48_1");
lgraph = connectLayers(lgraph,"maxpool_4_4_1","conv_61_1");
lgraph = connectLayers(lgraph,"maxpool_4_4_1","conv_53_1_1");
lgraph = connectLayers(lgraph,"relu_50_1","depthcat_4_1/in1");
lgraph = connectLayers(lgraph,"relu_52_1","depthcat_4_1/in3");
lgraph = connectLayers(lgraph,"relu_49_1","depthcat_4_1/in2");
lgraph = connectLayers(lgraph,"relu_53_2_1","depthcat_4_1/in7");
lgraph = connectLayers(lgraph,"relu_58_1","depthcat_4_1/in6");
lgraph = connectLayers(lgraph,"relu_56_1","depthcat_4_1/in5");
lgraph = connectLayers(lgraph,"relu_62_1","depthcat_4_1/in8");
lgraph = connectLayers(lgraph,"relu_54_1","depthcat_4_1/in4");
lgraph = connectLayers(lgraph,"maxpool_4_4_2","conv_57_2");
lgraph = connectLayers(lgraph,"maxpool_4_4_2","conv_51_2");
lgraph = connectLayers(lgraph,"maxpool_4_4_2","conv_53_2_2");
lgraph = connectLayers(lgraph,"maxpool_4_4_2","conv_53_1_2");
lgraph = connectLayers(lgraph,"maxpool_4_4_2","conv_48_2");
lgraph = connectLayers(lgraph,"maxpool_4_4_2","maxpool_5_2");
lgraph = connectLayers(lgraph,"maxpool_4_4_2","conv_55_2");
lgraph = connectLayers(lgraph,"maxpool_4_4_2","conv_61_2");
lgraph = connectLayers(lgraph,"relu_53_2_2","depthcat_4_2/in7");
lgraph = connectLayers(lgraph,"relu_50_2","depthcat_4_2/in1");
lgraph = connectLayers(lgraph,"relu_52_2","depthcat_4_2/in3");
lgraph = connectLayers(lgraph,"relu_49_2","depthcat_4_2/in2");
lgraph = connectLayers(lgraph,"relu_54_2","depthcat_4_2/in4");
lgraph = connectLayers(lgraph,"relu_58_2","depthcat_4_2/in6");
lgraph = connectLayers(lgraph,"relu_62_2","depthcat_4_2/in8");
lgraph = connectLayers(lgraph,"relu_56_2","depthcat_4_2/in5");
%%Plot Layers
plot(lgraph);
```

Training Code

```
clc;
clear all;
load('C:\Users\ABIOLA\Desktop\All02\All\Matlab CNN model\My Model\My
Model_UnTrained.mat');
%training data
datapathtrain = fullfile('C:\Users\ABIOLA\Desktop\All02\All\Final Data\Train');
augimgsTrain = imageDatastore(datapathtrain,...
   'IncludeSubfolders',true,...
   'LabelSource','foldernames');
%testing data
datapathtest = fullfile('C:\Users\ABIOLA\Desktop\All02\All\Final Data\Test');
augimgsValidation = imageDatastore(datapathtest,...
   'IncludeSubfolders',true,...
   'LabelSource','foldernames');
%show Number of training and testing data
disp(['training images: ',num2str(numel(augimgsTrain.Files))]);
disp(['validation images: ',num2str(numel(augimgsValidation.Files))]);
inputSize = [300 300 3];
imgsTrain =
augmentedImageDatastore(inputSize,augimgsTrain,'ColorPreprocessing','gray2rgb');
imgsValidation =
augmentedImageDatastore(inputSize,augimgsValidation,'ColorPreprocessing','gray2rgb');
lgraph=lgraph_1;
% train options
options = trainingOptions('sgdm',...
   'MiniBatchSize',150,...
   'MaxEpochs',100,...
   'InitialLearnRate',1e-3,...
   'ValidationData',imgsValidation,...
   'ValidationFrequency',15,...
   'Verbose',1,...
   'ExecutionEnvironment','gpu',...
   'Shuffle','every-epoch',...
   'Plots','training-progress');
rng default
trainedGN = trainNetwork(imgsTrain,lgraph,options);
%Show Class Names
cNames = trainedGN.Layers(end).ClassNames
```

Testing Code

```
clc;
clear all;
%load testing data
datapath = fullfile('C:\Users\ABIOLA\Desktop\All02\All\Final Data\Test');
allImagess = imageDatastore(datapath,...
    'IncludeSubfolders',true,...
    'LabelSource','foldernames');
Test=allImagess.Labels;
inputSize = [300 300 3];
allImages = augmentedImageDatastore(inputSize,allImagess);

%my model
load('C:\Users\ABIOLA\Desktop\All02\All\Matlab CNN model\My Model\My Model_Trained.mat');
Predict = classify(trainedGN,allImages);
accuracy_my_model = sum(Predict == Test)/numel(Test)
figure (1)
cm = confusionchart(Test,Predict)
%cm = confusionmat(Test,Predict)
cm = bsxfun(@rdivide,cm,sum(cm,2))
figure,plotconfusion(Test,Predict)
title('Classification Matrix- Proposed Model')
%cm.RowSummary = 'row-normalized';
%cm.ColumnSummary = 'column-normalized';
%Alexnet
load('C:\Users\ABIOLA\Desktop\All02\All\MatlabCNNmodel\Alexnet\alexnet_TRD.mat');
```

```
Predict2 = classify(alexnet_TRD,allImages);
accuracy_Alexnet = sum(Predict2 == Test)/numel(Test)
figure (2)
cm = confusionchart(Test,Predict)
%cm2 = confusionmat(Test,Predict2)
cm2 = bsxfun(@rdivide,cm2,sum(cm2,2))
figure,plotconfusion(Test,Predict2)
title('Classification Matrix- Alexnet')
cm.RowSummary = 'row-normalized';
cm.ColumnSummary = 'column-normalized';
%InceptionV3
load('C:\Users\ABIOLA\Desktop\All02\All\Matlab CNN
model\InceptionV3\InceptionV3_TRD.mat');
Predict3 = classify(InceptionV3_TRD,allImages);
accuracy_InceptionV3 = sum(Predict3 == Test)/numel(Test)
figure (3)
%cm = confusionchart(Test,Predict)
cm3 = confusionmat(Test,Predict3)
cm3 = bsxfun(@rdivide,cm3,sum(cm3,2))
figure,plotconfusion(Test,Predict3)
title('Classification Matrix- InceptionV3')
cm.RowSummary = 'row-normalized';
cm.ColumnSummary = 'column-normalized';
%Resnet18
load('C:\Users\ABIOLA\Desktop\All02\All\Matlab CNN
model\Resnet18\resnet18_TRD.mat');
Predict = classify(resnet18_TRD,allImages);
accuracy_Resnet18 = sum(Predict == Test)/numel(Test)
figure (4)
cm = confusionchart(Test,Predict)
cm.Title = 'Classification Matrix- ResNet18';
cm.RowSummary = 'row-normalized';
cm.ColumnSummary = 'column-normalized';
%GoogleNet
load('C:\Users\ABIOLA\Desktop\All02\All\Matlab CNN
model\Googlenet\gnet_TRD.mat');
Predict = classify(Googlenet_TRD,allImages);
accuracy_Googlenet = sum(Predict == Test)/numel(Test)
figure (5)
cm = confusionchart(Test,Predict)
cm.Title = 'Classification Matrix- GoogleNet';
cm.RowSummary = 'row-normalized';
cm.ColumnSummary = 'column-normalized';
```