**ABSTRACT**

Food is considered as a basic need of human being which can be satisfied through agriculture. Agriculture plays a vital role in the economic growth of any country. With the increase of population, frequent changes in climatic conditions and limited resources, it becomes a challenging task to fulﬁl the food requirement of the present population. Methane Gas affects the soils, which can lead to restricted root growth, respiration, and nutrient and water uptake. Machine learning has helped improving gains in agriculture soil. A subset of artificial intelligence, namely machine learning, has a considerable potential to handle numerous challenges in the establishment of knowledge-based agriculture soil systems and find the chemical in paddies, seeds, vegetables, etc. The approach can accelerate agricultural research, identify sustainable practices, and help overcome future in good food demands. Application of machine learning in agriculture allows more efficient and precise agriculture soil to find the calories in fruits, vegetables, etc. The aim of the project is to detect whether the humidity, pH level, temperature, rainfall and food calorie. Agricultural intensification reduces soil methane absorption, lowering or eliminating the methane sink function. Well-aerated agricultural soils, unlike rice paddies, have received little attention for their methanotrophic potential, perhaps owing to their poor methane absorption capability. This proposed approach aims to improve the classification performance of the system. ANN systems are helping to improve the overall harvest quality and accuracy known as precision agriculture. ANN is used by capture the images of food products and get information about the quality. The research discusses ways to increase methane output via varied crop rotations that include food, feed, raw materials, and energy production.

**CHAPTER- 1**

**INTRODUCTION**

**1.1OVERVIEW BACKGROUND**

**Machine Learning**

Machine learning is a growing technology which enables computers to learn automatically from past data. Machine learning uses various algorithms for **building mathematical models and making predictions using historical data or information**. Currently, it is being used for various tasks such as **image recognition**, **speech recognition**, **email filtering**, **facebook auto-tagging**, **recommender system**, and many more.

In the real world, we are surrounded by humans who can learn everything from their experiences with their learning capability, and we have computers or machines which work on our instructions. But can a machine also learn from experiences or past data like a human does? So here comes the role of **machine learning**.



**Fig.1.1 Working flow of Machine Learning**

A machine learning system **learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it**. The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately. Machine learning is used in internet search engines, email filters to sort out spam, websites to make personalised recommendations, banking software to detect unusual transactions, and lots of apps on our phones such as voice recognition.

The goal of AI is to create computer models that exhibit “intelligent behaviours” like humans, according to [Boris Katz](https://www.csail.mit.edu/person/boris-katz), a principal research scientist and head of the Info Lab Group at CSAIL. This means machines that can recognize a visual scene, understand a text written in natural language, or perform an action in the physical world.

Machine learning is one way to use AI. It was defined in the 1950s by AI pioneer [Arthur Samuel](https://en.wikipedia.org/wiki/Arthur_Samuel) as “the field of study that gives computers the ability to learn without explicitly being programmed.”

Machine learning starts with data — numbers, photos, or text, like bank transactions, pictures of people or even [bakery items](https://www.newyorker.com/tech/annals-of-technology/the-pastry-ai-that-learned-to-fight-cancer), repair records, time series data from sensors, or sales reports. The data is gathered and prepared to be used as training data, or the information the machine learning model will be trained on. The more data, the better the program.

From there, programmers choose a machine learning model to use, supply the data, and let the computer model train itself to find patterns or make predictions. Over time the human programmer can also tweak the model, including changing its parameters, to help push it toward more accurate results. (Research scientist Janelle Shane’s website [AI weirdness](https://aiweirdness.com/) is an entertaining look at how machine learning algorithms learn and how they can get things wrong — as happened when [an algorithm tried to generate recipes](https://aiweirdness.com/post/190569291992/ai-recipes-are-bad-and-a-proposal-for-making-them) and created chocolate chicken cake.)

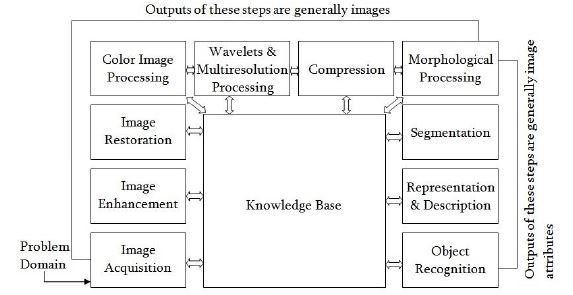
**IMAGE PROCESSING**

Image processing is the most important part of the machine vision. Machine Vision is becoming more and more important these days because it is being used in manufacturing, inspection of parts, medical applications and robot guiding. Machine vision means making the machines to interact with the environment as human beings do in terms of seeing. But there is much work left to be done. Visual data are most complex and most useful sensory input for humans. Machine vision is concerned with the interpretation of similar visual data. Image Processing is the science of modifying and analyzing pictures. For analysis of images we need to find their edges first. Edges information can be used for segmentation of images or for locating their boundary in formation. Edge detection is important because edges give the compact description of the objects and objects can be reconstructed from the edge information.

Vision begins with transformation of a flux of photons into a set of intensity values at an array of sensors. The first step in visual information processing is to obtain a compact description of the raw intensity values. The primitive elements of the initial description should ideally be complete in the sense of representing the full information contained in the image, and meaningful (that is, capturing significant properties of the three-dimensional surfaces around the viewer). Physical edges are among the most of objects since they correspond to object boundaries or to changes in surface orientation or material properties.

**BLOCK DIAGRAM OF IMAGE PROCESSING:**

Fundamental steps in digital image processing are shown in Fig. Image acquisition digitizes the image captured by camera. Image enhancement is the process of manipulating an image so that the results are more suitable for specific applications. Image restoration improves an appearance of an image, which tends to probabilities model of image degradation Morphological processes are the tools of extracting image components that are useful in the description and presentation of an image. Image segmentation is the most difficult ask in digital image processing which separates objects from the background. Representation makes the decision whether to represent data as boundary or as a complete region. Recognition is the process that assigns label to an object based on information provided by its descriptor.



**Fig.1.2** **Block Diagram of Image Processing**

**Python**

Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis. Python is a general-purpose language, meaning it can be used to create a variety of different programs and isn’t specialized for any specific problems.

This versatility, along with its beginner-friendliness, has made it one of the most-used programming languages today. A survey conducted by industry analyst firm Red Monk found that it was the second-most popular programming language among developers in 2021

Python is a very popular general-purpose interpreted, interactive, object-oriented, and high-level programming language. Python is dynamically-typed and garbage-collected programming language.

It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). Python supports multiple programming paradigms, including Procedural, Object Oriented and Functional programming language. Python design philosophy emphasizes code readability with the use of significant indentation.

Python is an open source and high-level programming language designed by Guido van Rossum. Released in 1991, this programming language’s design philosophy aims at code readability, which is why it’s one of the easiest languages to learn.

In addition, [Python](https://kinsta.com/blog/php-vs-python/) is an object-oriented, reflective, functional, procedural, and structural language. All these approaches of Python help programmers write logical and clear code for projects of all scales.

**Methane Soil Food Calorie Value Prediction**

The Methane soil Food calorie Value Prediction predicts the current Food calorie of the methane soil with the help of image acquisition and then applies machine learning algorithms to predict the current Food calorie of the methane soil and then recommends which crop to grow. The Food calorie value is predicted based on the food calorie to which is taken of the methane soil and the quality may be compromised sometimes which may lead to in accuracy. The proposed system of this paper includes capturing methane soil image, pre-processing, feature extraction and then recognition of methane soil Food calorie value. After all of this crop recommendation happens. So, this is the major disadvantage that the image quality should be up to the mark for good accuracy which is not possible every time and climatic conditions also play a major role. In uses same technology but detects the methane soil image processing with the help of python coding. This paper uses ANN classifier to classify images based on upon their RCB values and label the images with methane soil notation. In uses same technique with support vector machine algorithm for classification of the methane soil types.

**1.2 Objective**

The main objective of the project is to provide a platform with a simple user interface where user can get the expected price of a crop in a particular crop and in addition to this the Food calorie value of methane soil which is required in order to grow a particular crop in a particular area. To achieve this machine learning regression algorithms will be considered and trained for use. Also a web application will be developed using which users can give input and get the predicted values as output. The objective is make sure right information is given to users using machine learning algorithms. Also making it accessible for the target audience by providing a web application with a simple user interface so that it can be easily used and operated by the users. In addition to this useful links will also be present which can be helpful for the users for example links with list of fertilizers, maintenance of fields, etc.

**1.3 MOTIVATION AND PROBLEM STATEMENT**

Agriculture plays an important role when it comes to the GDP of this country also is a source of income for about 60 percent of the population. The backbone of the Indian economy is the agriculture sector this means the financial system of India greatly depends on agriculture and there is a need to improve overall production in which use of technologies like machine learning can play an important role. Also farming is not only limited to farmers these days as many people are looking and investing in this sector after exploring it‟s potential. People who are knew in this field requires accurate and relevant information as they rely only on past experiences of other farmers, our project looks to solve this problem and provide adequate information to all also the farmers who are relying on past experience will also have accurate information. Food calori value plays a very important role when it comes to yield and production quality of the crops and due to lack of appropriate knowledge it leads to financial losses and many times the quality of the demand is not met. A system which provides right information about what is the Food calori value required for a particular crop to grow in a particular area is missing and our project will fill this void. Also farmers currently need to rely on past experience for estimating the price of the crop in the mandi and they have idea particularly of two or three markets near their area. Here also a system is missing where a farmer can check price of crops in different markets of the district also in addition to this other than minimum price which is set by they have to relevant source of information of what price to expect.

**CHAPTER 2**

**LITERATURE SURVEY**

**[1] N.Zhang,M.Wang,andN.Wang,‘‘Precisionagriculture—Aworldwide overview,’’ Comput. Electron. Agricult., vol. 36, nos. 2–3, pp. 113–132, 2020.**

Agriculture plays a vital role in the economic growth of any country. With the increase of population, frequent changes in climatic conditions and limited resources, it becomes a challenging task to fulﬁl the food requirement of the present population. Precision agriculture also known as smart farming have emerged as an innovative tool to address current challenges in agricultural sustainability. The mechanism that drives this cutting edge technology is machine learning (ML). It gives the machine ability to learn without being explicitly programmed. ML together with IoT (Internet of Things) enabled farm machinery are key components of the next agriculture revolution. In this article, authors present a systematic review of ML applications in the ﬁeld of agriculture. The areas that are focused are prediction of soil parameters such as organic carbon and moisture content, crop yield prediction, disease and weed detection in crops and species detection. ML with computer vision are reviewed for the classiﬁcation of a different set of crop images in order to monitor the crop quality and yield assessment. This approach can be integrated for enhanced livestock production by predicting fertility patterns, diagnosing eating disorders, cattle behaviour based on ML models using data collected by collar sensors, etc. Intelligent irrigation which includes drip irrigation and intelligent harvesting techniques are also reviewed that reduces human labour to a great extent. This article demonstrates how knowledge-based agriculture can improve the sustainable productivity and quality of the product.

Artiﬁcial intelligence (AI) has smoothly penetrated

in a number of monitoring and control applications including the

agriculture. However, research efforts towards low-power sensing

devices with fully-functional AI on board are still fragmented.

In this work, we present an embedded system enriched with the

AI ensuring the continuous analysis and in-situ prediction of

the growth dynamics of plant leaves. The embedded solution

is grounded on a low-power embedded sensing system with

a Graphics Processing Unit (GPU) and is able to run the

neural networks-based AI on board. We use a Recurrent Neural

Network (RNN) called the Long-Short Term Memory network

(LSTM) as a core of AI in our system. The proposed approach

guarantees the system autonomous operation for 180 days using

a standard Li-ion battery. We rely on the state-of-the-art mobile

graphical chips for ’smart’ analysis and control of autonomous

devices. This pilot study opens up wide vista for a variety of

intelligent monitoring applications, especially in the agriculture

domain. Also, we share with the research community the Tomato

Growth datase

**[2] D.Shadrin,A.Menshchikov,A.Somov,G.Bornemann,J.Hauslage,and M. Fedorov, ‘‘Enabling precision agriculture through embedded sensing with artiﬁcial intelligence,’’ IEEE Trans. Instrum. Meas., vol. 69, no. 7, pp. 4103–4113, Jul. 2020.**

Artiﬁcial intelligence (AI) has smoothly penetrated

in a number of monitoring and control applications including the

agriculture. However, research efforts towards low-power sensing

devices with fully-functional AI on board are still fragmented.

In this work, we present an embedded system enriched with the

AI ensuring the continuous analysis and in-situ prediction of

the growth dynamics of plant leaves. The embedded solution

is grounded on a low-power embedded sensing system with

a Graphics Processing Unit (GPU) and is able to run the

neural networks-based AI on board. We use a Recurrent Neural

Network (RNN) called the Long-Short Term Memory network

(LSTM) as a core of AI in our system. The proposed approach

guarantees the system autonomous operation for 180 days using

a standard Li-ion battery. We rely on the state-of-the-art mobile

graphical chips for ’smart’ analysis and control of autonomous

devices. This pilot study opens up wide vista for a variety of

intelligent monitoring applications, especially in the agriculture

domain. Also, we share with the research community the Tomato

Growth datase

Artiﬁcial intelligence (AI) has smoothly penetrated

in a number of monitoring and control applications including the

agriculture. However, research efforts towards low-power sensing

devices with fully-functional AI on board are still fragmented.

In this work, we present an embedded system enriched with the

AI ensuring the continuous analysis and in-situ prediction of

the growth dynamics of plant leaves. The embedded solution

is grounded on a low-power embedded sensing system with

a Graphics Processing Unit (GPU) and is able to run the

neural networks-based AI on board. We use a Recurrent Neural

Network (RNN) called the Long-Short Term Memory network

(LSTM) as a core of AI in our system. The proposed approach

guarantees the system autonomous operation for 180 days using

a standard Li-ion battery. We rely on the state-of-the-art mobile

graphical chips for ’smart’ analysis and control of autonomous

devices. This pilot study opens up wide vista for a variety of

intelligent monitoring applications, especially in the agriculture

domain. Also, we share with the research community the Tomato

Growth datase

Artiﬁcial intelligence (AI) has smoothly penetrated in a number of monitoring and control applications including the agriculture. However, research efforts towards low-power sensing devices with fully-functional AI on board are still fragmented. In this work, we present an embedded system enriched with the AI ensuring the continuous analysis and in-situ prediction ofthe growth dynamics of plant leaves. The embedded solution is grounded on a low-power embedded sensing system with a Graphics Processing Unit (GPU) and is able to run the neural networks-based AI on board. We use a Recurrent Neural Network (RNN) called the Long-Short Term Memory network(LSTM) as a core of AI in our system. The proposed approach guarantees the system autonomous operation for 180 days usinga standard Li-ion battery. We rely on the state-of-the-art mobile graphical chips for ’smart’ analysis and control of autonomous devices. This pilot study opens up wide vista for a variety of intelligent monitoring applications, especially in the agriculture domain. Also, we share with the research community the Tomato Growth dataset.

**[3] Emadi, M.; Taghizadeh-Mehrjardi, R.; Cherati, A.; Danesh, M.; Mosavi, A.; Scholten, T. Predicting and Mapping of Soil Organic Carbon Using Machine Learning Algorithms in Northern Iran. Remote Sens. 2021, 12, 2234.**

Estimation of the soil organic carbon (SOC) content is of utmost importance in understanding the chemical, physical, and biological functions of the soil. This study proposes machine learning algorithms of support vector machines (SVM), artificial neural networks (ANN), regression tree, random forest (RF), extreme gradient boosting (XGBoost), and conventional deep neural network (DNN) for advancing prediction models of SOC. Models are trained with 1879 composite surface soil samples, and 105 auxiliary data as predictors. The genetic algorithm is used as a feature selection approach to identify effective variables. In terms of accuracy, DNN yielded a mean absolute error of 0.59%, a root mean squared error of 0.75%, a coefficient of determination of 0.65, and Lin’s concordance correlation coefficient of 0.83. The SOC content was the highest in udic soil moisture regime class with mean values of 3.71%, followed by the aquic (2.45%) and xeric (2.10%) classes, respectively. Soils in dense forestlands had the highest SOC contents, whereas soils of younger geological age and alluvial fans had lower SOC. The proposed DNN (hidden layers = 7, and size = 50) is a promising algorithm for handling large numbers of auxiliary data at a province-scale, and due to its flexible structure and the ability to extract more information from the auxiliary data surrounding the sampled observations, it had high accuracy for the prediction of the SOC base-line map and minimal uncertainty.

**[4] Chen, L.; Ren, C.; Li, L.; Wang, Y.; Zhang, B.; Wang, Z.; Li, L. A Comparative Assessment of Geostatistical, Machine Learning, and Hybrid Approaches for Mapping Topsoil Organic Carbon Content. ISPRS Int. J. Geo-Information 2019, 8, 174**

Accurate digital soil mapping (DSM) of soil organic carbon (SOC) is still a challenging subject because of its spatial variability and dependency. This study is aimed at comparing six typical methods in three types of DSM techniques for SOC mapping in an area surrounding Changchun in Northeast China. The methods include ordinary kriging (OK) and geographically weighted regression (GWR) from geostatistics, support vector machines for regression (SVR) and artificial neural networks (ANN) from machine learning, and geographically weighted regression kriging (GWRK) and artificial neural networks kriging (ANNK) from hybrid approaches. The hybrid approaches, in particular, integrated the GWR from geostatistics and ANN from machine learning with the estimation of residuals by ordinary kriging, respectively. Environmental variables, including soil properties, climatic, topographic, and remote sensing data, were used for modeling. The mapping results of SOC content from different models were validated by independent testing data based on values of the mean error, root mean squared error and coefficient of determination. The prediction maps depicted spatial variation and patterns of SOC content of the study area. The results showed the accuracy ranking of the compared methods in decreasing order was ANNK, SVR, ANN, GWRK, OK, and GWR. Two-step hybrid approaches performed better than the corresponding individual models, and non-linear models performed better than the linear models. When considering the uncertainty and efficiency, ML and two-step approach are more suitable than geostatistics in regional landscapes with the high heterogeneity. The study concludes that ANNK is a promising approach for mapping SOC content at a local scale.

**[5] Gehl, R.J.; Rice, C.W. Emerging technologies for in situ measurement of soil carbon. Clim. Chang. 2007, 80, 43–54.**

Carbon sequestration in the terrestrial biosphere is critical to mitigating the increasing anthropogenic CO2content of the atmosphere. However, improved efﬁciency of methods for soil C measurement is important to better estimate terrestrial C inventories and ﬂuxes at a regional and global scale. Laboratory based measurement of soil C involves intensive, time consuming, and costly methodology that limits applicability for large land areas. Recently, research efforts have focused on measuring soil C in situ using a variety of methods. These methods include Laser Induced Breakdown Spectroscopy (LIBS), Inelastic Neutron Scattering (INS), near-infrared spectroscopy (NIRS), and remote sensing.

**[6] Heung, B. Bulmer, C.E. Schmidt, M.G. Predictive soil parent material mapping at a regional scale: A Random Forest approach. Geoderma 2014, 214, 141–154.**

Agriculture sector is the backbone of the Indian economy and the aim of the project is to help this sector grow more. To meet the requirements of the growing demand modernization is a must need and use of technology is really important. The contribution of the agriculture sector in the Gross Domestic Product is nearly 18% and 40% is the contribution in the Net Domestic Product. Being such a high contributor in the economy it has more potential to grow but due to lack of implementation of proper technology which could directly help the farmers or people related to the sector it has led to downfall of this sector. India being one of the largest populated country in the world, if this sector goes down then the demands need to meet by importing which would decrease the economy and make it difficult for the poor and middle class families as food would not be cheaper as it is today. Although a lot of policies have been made related to this sector to make sure the farmers have low risk as they are provided with economical support and introduction of Minimum Support Price which makes sure that nothing is wasted but a very few policies have been made to increase the quality and yield of production. The number of policies which help farmers increase their capacity are a very few. In the twenty first century farmers don‟t only need policies but they technology which could help them directly by increasing their yield and quality of production.

**[7] G.-J. Horng, M.-X. Liu, and C.-C. Chen, ‘‘The smart image recognition mechanism for crop harvesting system in intelligent agriculture,’’ IEEE Sensors J., vol. 20, no. 5, pp. 2766–2781, Mar. 2022.**

This study proposed a harvesting system based on the Internet of Things technology and smart image recognition. Farming decisions require extensive experience; with the proposed system, crop maturity can be determined through object detection by training neural network models, and mature crops can then be harvested using robotic arms. Keras was used to construct a multilayer perceptron machine learning model and to predict multiaxial robotic arm movements and position. Following the execution of object detection on images, the pixel coordinates of the central point of the target crop in the image were used as neural network input, whereas the robotic arms were regarded as the output side. A MobileNet version 2 convolutional neural network was then used as the image feature extraction model, which was combined with a single shot multibox detector model as the posterior layer to form an object detection model. The model then performed crop detection by collecting and tagging images. Empirical evidence shows that the proposed model training had a mean average precision (mAP) of 84%, which was higher than that of other models; a mAP of 89% was observed from the arm picking results.

**[8] W. Wu, S. Ding, and J. Lv, ‘‘Application research of neural networks in fruit and vegetable harvesting robot,’’ in Proc. 10th World Congr. Intell. Control Autom., Jul. 2022, pp. 1790–1795.**

The fruit and vegetable picking has posed a great challenge on the production and markets during the harvest season. Manual picking cannot fully meet the rapid requirements of each market, mainly due to the high labor-intensive and time-consuming tasks, even the aging and shortage of agricultural labor force in recent years. Alternatively, smart robotics can be an efficient solution to increase the planting areas for the markets in combination with changes in cultivation, preservation, and processing technology. However, some improvements still need to be performed on these picking robots. To document the progress in and current status of this field, this study performed a bibliometric analysis. This analysis evaluated the current performance characteristics of various fruit and vegetable picking robots for better prospects in the future. Five perspectives were proposed covering the robotic arms, end effectors, vision systems, picking environments, and picking performance for the large-scale mechanized production of fruits and vegetables in modern agriculture. The current problems of fruit and vegetable picking robots were summarized. Finally, the outlook of the fruit and vegetable picking robots prospected from four aspects: structured environment for fruit planting, the algorithm of recognition and positioning, picking efficiency, and cost-saving picking robots. This study comprehensively assesses the current research status, thus helping researchers steer their projects or locate potential collaborators.

**[9] Y. Mekonnen, S. Namuduri, L. Burton, A. Sarwat, and S. Bhansali, ‘‘Review—Machinelearningtechniquesinwirelesssensornetworkbased precision agriculture,’’ J. Electrochem. Soc., vol. 167, no. 3, Jan. 2020, Art. no. 037522.**

The use of sensors and the Internet of Things (IoT) is key to moving the world’s agriculture to a more productive and sustainable path. Recent advancements in IoT, Wireless Sensor Networks (WSN), and Information and Communication Technology (ICT) have the potential to address some of the environmental, economic, and technical challenges as well as opportunities in this sector. As the number of interconnected devices continues to grow, this generates more big data with multiple modalities and spatial and temporal variations. Intelligent processing and analysis of this big data are necessary to developing a higher level of knowledge base and insights that results in better decision making, forecasting, and reliable management of sensors. This paper is a comprehensive review of the application of different machine learning algorithms in sensor data analytics within the agricultural ecosystem. It further discusses a case study on an IoT based data-driven smart farm prototype as an integrated food, energy, and water (FEW) system.

**[10] Bou Kheir, R.; Greve, M.H.; Bøcher, P.K.; Greve, M.B.; Larsen, R.; McCloy, K. Predictive mapping of soil organic carbon in wet cultivated lands using classiﬁcation-tree based models: The case study of Denmark. J. Environ. Manag. 2021, 91, 1150–1160.**

Soil organic carbon (SOC) is one of the most important carbon stocks globally and has large potential to affect global climate. Distribution patterns of SOC in Denmark constitute a nation-wide baseline for studies on soil carbon changes (with respect to Kyoto protocol). This paper predicts and maps the geographic distribution of SOC across Denmark using remote sensing (RS), geographic information systems (GISs) and decision-tree modeling (un-pruned and pruned classification trees). Seventeen parameters, i.e. parent material, soil type, landscape type, elevation, slope gradient, slope aspect, mean curvature, plan curvature, profile curvature, flow accumulation, specific catchment area, tangent slope, tangent curvature, steady-state wetness index, Normalized Difference Vegetation Index (NDVI), Normalized Difference Wetness Index (NDWI) and Soil Color Index (SCI) were generated to statistically explain SOC field measurements in the area of interest.

**CHAPTER 3**

**METHODOLOGY**

**3.1 Introduction**

Methodology may be defined as systematic knowledge of the best way of setting to work. In the development and progress of the sciences methodology has played a very important role. So also in the realm of agricultural research, methodology is a vital necessity, and my plea -therefore is for its greatest possible utilization. General methodology is undoubtedly the most valuable tool of trade in all occupations, but is probably the most neglected of subjects. Without methodology we would possess no scientific knowledge. Logic provided the foundation-stone, then followed mathematics, and finally came statistical method. These three comprise what may be termed methodology proper. In the wider sense, however, we may include both ordinary method or practice, as represented in art and ‘craft, and scientific method or practice--i.e., techniques applied in the laboratory and field. In another category we have mechanical aids which are indispensable in scientific technique and which are responsible for a certain amount of modification and adaptation of technique due to their limitations. Accurate information that makes crop history more yielding is the main reason for making the decision, and in particular it makes risking a high factor and their risk management. This paper presents a crop yield prediction system, which helps the farmers to predict the crop yield. An ANN is developed to obtain the data from pH sensor and moisture sensor, wherein this data values are trained and tested and also it has been used to predict the crop yielding in the particular field.

**3.2 Dataflow Diagram**

**INPUT IMAGE**

**PREPROCESSING**

**(Noise Removal and Filtering)**

**IMAGE SEGMENTATION**

**FEATURE EXTRACTION**

**CLASSIFICATION**

**OUTPUT IMAGE**

**Fig.3.1 Dataflow Diagram**

**3.3 Module Description**

**3.3.1 Pre-Processing**

The aim of preprocessing is an improvement of an image data that suppresses unwanted distortions or enhances some image features important for further processing. The pre preparing associated with transformation, picture resize, noise removing and improves the quality and produces an image where in details can be perceived precisely. To remove such unwanted data in an image, number of the image pre-processing systems is required in order to better perception of the images. The way toward upgrading pixel intensity and image quality additionally are managed after pre handling. In this research, the pre-processing techniques is used for removing the noise in the image.. Different filtering techniques can be adapted for noise declining to improve the visual quality as well as reorganization of images. The filters are Median filter are used to enhance the clarity of soil images.

**i. Grayscale Conversion:**

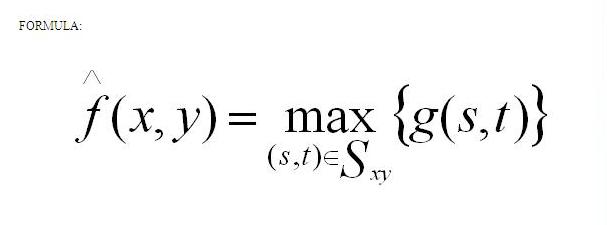
The filtering strategies like Gaussian and median channel are balanced for noise removal method. it is easy to remove the noise by using conversion of colours images to gray scale images. The image acquired is in RGB color. It is converted into gray scale because it carries only the intensity information which is easy to process instead of processing three components R (Red), G(Green), B(Blue).

**ii. Filtering Methods:**

Filtering technique for improving a picture , in filtering are essentially wont to cover either the high frequencies inside the image, for instance smoothing the image, or the low frequencies, i.e., improving or perceiving edges inside the image. For instance, you will filter a picture to pressure certain feature or evacuate different highlights. Number of techniques are available and thusly as well as can be expected depends on the picture and the manner in which it will be utilized. Picture filtering is important for a few applications, including smoothing, sharpening, expelling noise, and edge acknowledgment. Filtering methods Input pictures are influenced by various noise either Gaussian channel, Median channel, etc. so noise picture process is managed to improve the image quality utilizing the filtering strategy respectively. The detail of filter described below.

**3.3.1.1 Median Filter:**

The median filter is the filtering technique used for noise removal from images and signals. Median filter is very crucial in the image processing field as it is well known for the **preservation of edges during noise removal.** Order-statistics filter also known as Median filter, which exchanges the estimation of a pixel by the middle of the gray levels in the region of that pixel. The median is a rank command statistic and in a intelligence the main stream of the pixel values included determines the result. The expression of Median filter,

…..(1)

The first estimation of the pixel is incorporated into the calculation of the median. Median filters are extremely standard for assured sorts of arbitrary noise. They give astonishing noise diminish capacities, with stunningly less clouding than a linear smoothing filter of comparative size. The median is figured by first sorting all the pixel values from the window in numerical order. A while later supplanting the pixel being considered with the inside (middle) pixel value. Median filters are viable for the bipolar and unipolar impulse noise. Median filters are mainly reasonable within the sight of both unipolar and bipolar impulse noise.

**Algorithm for median filter**

To remove salt and pepper noise from the corrupted image the below described algorithm is used.

**Step 1**: A two dimensional window (denoted by 3×3 W) of size 3x3 is selected and centered around the processed pixel p(x, y) in the corrupted image.

**Step 2**: Sort the pixels in the selected window according to the ascending order and find the median pixel value denoted by Pmed), maximum pixel value (Pmax) and minimum pixel value (Pmin) of the sorted vector V0. Now the first and last elements of the vector V0 is the Pmin and Pmax respectively and the middle element of the vector is the Pmed.

**Step 3**: If the processed pixel is within the range Pmin < P(x, y) < Pmax , Pmin > 0 and Pmax < 255,it is classified as uncorrupted pixel and it is left unchanged. Otherwise p(x, y) is classified as corrupted pixel.

**Step 4:** If p(x, y) is corrupted pixel, then we have the following two cases:

**Case 1:** If Pmin < Pmed < Pmax and 0 < Pmed < 255, replace the corrupted pixel p(x, y) with Pmed.

**Case 2:** If the condition in case 1 is not satisfied then Pmed is a noisy pixel. In this case compute the difference between each pair of adjacent pixel across the sorted vector V0 and obtain the difference vector VD .Then find the maximum difference in the VD and mark its corresponding pixel in the V0 to the processed pixel.

**Step 5:** Step 1 to step 4 are repeated until the processing is completed for the entire image.

**3.3.2 Segmentation:**

Image segmentation is the process of partitioning a digital image into multiple image segments, also known as image regions or image objects (sets of pixels). By dividing an image into segments, the process only the important segments of the image instead of processing the entire image. The purpose of Image segmentation is to partition an image into meaningful regions with respect to a particular application. Image segmentation is a method of separating the image from the background and read the contents. In this research through study has been done on most commonly used edge detection techniques such as Canny-edge detection. Essentially, in image partitions are different objects which have the same texture or color. The image segmentation results are a set of regions that cover the entire image together and a set of contours extracted from the image. All of the pixels in a region are similar with respect to some characteristics such as color, intensity, or texture. Adjacent regions are considerably different with respect to the same individuality. The different approaches are,

1. by finding boundaries between regions based on discontinuities in intensity levels,
2. thresholds based on the distribution of pixel properties, such as intensity values, and
3. based on finding the regions directly.

Thus the choice of image segmentation technique is depends on the problem being considered. **Steps in Edge detection:**

**A.** **Filtration:** Every image is associated with some intensity values, random change in these values can result in noise. Some common noise is: salt and pepper noise, impulse noise etc. Noise can result in difficulties in effective edge detection; hence image has to be filtered in order to reduce the noise content that leads to loss of edge strength . It is also termed as Smoothening.

**B. Enhancement:** Improving the quality of image is termed as enhancement. It aims to produce an image which is better and more suitable than original. A filter is applied in order to enhance the quality of edge in image.

**C. Detection:** Several methods are adopted to determine which points are edge points and which a edge pixels should be discarded as noise.

Edge detection is the problem of fundamental importance in image analysis. Edge detection techniques are generally used for finding discontinuities in gray level images. To detect consequential discontinuities in the gray level image is the important common approach in edge detection. Image segmentation methods for detecting discontinuities are boundary based methods.

**3.3.2.1 Canny-Edge Detection:**

The Canny edge detection technique is one of the standard edge detection techniques. It was first created by John Canny for his Master’s thesis at MIT in 1983, and still outperforms many of the newer algorithms that have been developed. To find edges by separating noise from the image before find edges of image the Canny is a very important method. Canny method is a better method without disturbing the features of the edges in the image. It is a better method because it extracts the features in an image without disturbing its features. There are certain criteria to improve current methods of edge detection. The first and most obvious is low error rate. It is important that edges occurring in images should not be missed. The second criterion is that the edge points be well localized i.e. the distance between the edge pixels as found by the detector and the actual edge should be minimum. A third criterion is to have only one response to a single edge.

**Importance of Canny**

Despite of number of edge detection techniques available canny algorithm is considered because it contains a number of adjustable parameters which can affect the computation time and effectiveness of the algorithm.

a) The size of the Gaussian filter: The smoothing filter used in the first stage directly affects the results of the detection of small, sharp lines. A larger filter causes more blurring, smearing out the value of an given pixel over a larger area of image.

b) The use of two thresholds with hysteresis allows more flexibility than in a single-threshold. A threshold set too high can miss important information. On the other hand, a threshold set too low will falsely identify irrelevant information (such as noise) as important.

The edge detection in this technique is optimized with regard to the following criteria.

a) Maximizing the signal-to-noise ratio of the gradient.

b) Edge localization for ensuring the accuracy of edge.

c) Minimizing multiple responses to a single edge.

**The algorithmic steps are as follows:**

**Step 1:** Convolve image f(r, c) with a Gaussian function to get smooth image

f^(r, c). f^(r, c)=f(r,c)\*G(r,c,6)

**Step 2:** Apply first difference gradient operator to compute edge strength then edge magnitude and direction are obtain as before.

**Step 3:** Apply non-maximal or critical suppression to the gradient magnitude.

**Step 4:** Apply threshold to the non-maximal suppression image.

**3.3.3 Feature Extraction**

Feature extraction increases the accuracy of learned models by extracting features from the input data. This phase of the general framework reduces the dimensionality of data by removing the redundant data. Feature extraction is a subset of feature engineering. Data scientists turn to feature extraction when the data in its raw form is unusable. Feature extraction transforms raw data into numerical features compatible with machine learning algorithms. One common application is raw data in the form of image files—by extracting the shape of an object or the redness value in images, data scientists can create new features suitable for machine learning applications.

Feature extraction cuts through the noise, removing redundant and unnecessary data. This frees machine learning programs to focus on the most relevant data. The most accurate machine learning models are those developed using only the data required to train the model to its intended business use. Including peripheral data negatively impacts the model’s accuracy.

Feature extraction plays a key role in image processing. Along with other tools, this technique is used to detect features in digital images such as edges, shapes, or motion. Once these are identified, the data can be processed to perform various tasks related to analysing an image.

**3.3.3.1 Gray-level run-length matrix (GLRLM)**

Gray-level run-length matrix (GLRLM) is a matrix from which the texture features can be extracted for texture analysis. The GLRLM method is a way of extracting higher order statistical texture features.

A gray level run can be described as a line of pixels in a certain direction with the same intensity value. The number of such pixels defines the gray level run length and the number of occurrences is called the run length value.

Here a run length is considered to be a number of neighbouring pixels that possess the same grey intensity in a particular direction. In this work only seven GLRLM features will be extracted and these parameters are Short Run Emphasis (SRE), Long Run Emphasis (LRE), Gray level non-uniformity (GLN), Run length non-uniformity (RLN), Run Percentage (RP), Low Gray Level Run Emphasis (LGLRE), and High Gray Level Run Emphasis (LGLRE).

Such a huge computation poses a big problem for real-time system. For other tasks, this preprocess of GLRLM construction and feature extraction would cause the whole process time consuming. Though for a single image, it may not take much time, but for hundreds of slices from an MRI examination, it may take hours to process them. Hence, it is important and beneficial if we can accelerate it. With the rapid development of massively parallel Graphics Processing Unit (GPU) computing technology, typically the CUDA framework, there is an affordable solution by parallelizing the GLRLM construction and features extraction. Such acceleration examples are reported for Gray Level Co-occurrence Matrix (GLCM) based features. It achieved 19 fold speedup for computing GLCM and Haralick features on microscope images of 1344 × 1024 pixels and 12 bit gray level depth. Dixon and Ding reported 7 fold speedup for GLCM construction and 9.83 fold speedup for feature extraction on diffraction images.

**3.3.4 Classification**

A machine or intelligent computer program learns and extract knowledge from the data, builds a framework for making predictions or intelligent decisions. Thus, the ML process is divided into three key parts, i.e. data input, model building, and generalization. Generalization is the process for predicting the output for the inputs with which the algorithm has not been trained before. ML algorithms are mainly used to solve complex problems where human expertise fails such as weather prediction, spam ﬁltering, disease identiﬁcation in plants, pattern recognition. Today, due to the availability of innovative algorithms and large data sets through internet resources industries and research communities are widely using ML algorithms for solving a diverse set of problems. DL is the subﬁeld of the family of ML algorithms which is trained from large set sand uses an artiﬁcial neural network (ANN) to make intelligent decisions. The AI systems are applicable in each farming operation as some of them even extend beyond the conventionally recognized steps. In this section we will discuss the state of art techniques proposed/implemented by various researchers and practitioners worldwide.

**3.3.4.1Artiﬁcial Neural Network:**

In predictive modeling and forecasting, as well as nonlinear and impermanent time series of processes where there is no exact solution and clear relationship to recognize and describe them, artiﬁcial neural networks have shown good performance. The frequently used ANN model is referred to as the multilayer perceptron(MLP). This model is occasionally used as a substitute for a feed-forward network. The MLP requires a well-known output so that to learn and train the network; this type of neural network is referred to as a supervised network. MLP produces a model that plots the input to the output using training data so that subsequently, the model is applied to predict the output when it is unknown. In the present study, and after some preliminary tests to choose the model, multilayer feed-forward back-propagation ANN was applied. The ANN models are well adapted for modeling nonlinear behaviour. They have the capacity of learning for complex relationships between multiple inputs and output variables. The ANN model was run in R using the package “nnet.” The best structure for the ANN model was obtained by changing the size (number of units in the hidden layer). ANN accurately predicts the rainfall and crop yield across different regions of globe. ANN model best predicts the nitrate, potassium, phosperous, content and water requirement in system.

Inspired by the properties of biological neural networks, Artificial Neural Networks are statistical learning algorithms and are used for a variety of tasks, from relatively simple classification tasks to computer vision and speech recognition. Artificial neural networks are implemented as a system of interconnected processing elements, called nodes, which are functionally analogous to biological neurons. The connections between different nodes have numerical values, called weights, and by altering these values in a systematic way, the network is eventually able to approximate the desired function. The hidden layers can be thought of as individual feature detectors, recognizing more and more complex patterns in the data as it is propagated throughout the network. For example, if the network is given a task to recognize a face, the first hidden layer might act as a line detector, the second hidden takes these lines as input and puts them together to form a nose, the third hidden layer takes the nose and matches it with an eye and so on, until finally the whole face is constructed. This hierarchy enables the network to eventually recognize very complex objects.

**Input Layer:**

As the name suggests, it accepts inputs in several different formats provided by the programmer.

**Hidden Layer:**

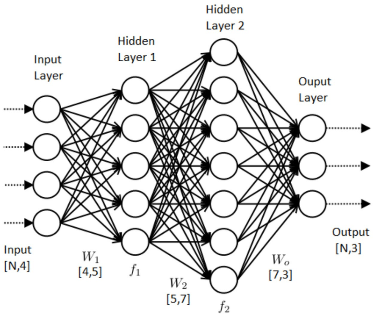
The hidden layer presents in-between input and output layers. It performs all the calculations to find hidden features and patterns.

**Output Layer:**

The input goes through a series of transformations using the hidden layer, which finally results in output that is conveyed using this layer. The artificial neural network takes input and computes the weighted sum of the inputs and includes a bias. This computation is represented in the form of a transfer function.

What is Artificial Neural Network……(2)

It determines weighted total is passed as an input to an activation function to produce the output. Activation functions choose whether a node should fire or not. Only those who are fired make it to the output layer. There are distinctive activation functions available that can be applied upon the sort of task we are performing.



**Fig.3.2 Artificial Neural Network**

**CHAPTER-4**

**REQUIREMENTS**

**4.1 HARDWARE REQUIREMENTS**

The hardware requirements and their specifications of the system for the project implementation.

• Processor: Intel i3

• Hard disk: 250 GB

• RAM: 4 GB

**4.2 SOFTWARE REQUIREMENTS**

The software Requirements used for the implementation of the program.

• IDE: Jupyter

• Front End: Anaconda Python

• Back End: Dataset

**4.3 INTRODUCTION TO SOFTWARE**

**4.3.1 Introduction to Python**

Python is an interpreted, high-level and general-purpose programming language. Python’s design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a “batteries included” language due to its comprehensive standard library. Python was created in the late 1980s, and first released in 1991, by Guido van Rossum as a successor to the ABC programming language. Python 2.0, released in 2000, introduced new features, such as list comprehensions, and a garbage collection system with reference counting, and was discontinued with version 2.7 in 2020. Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible and much Python 2 code does not run unmodified on Python 3.

Python interpreters are supported for mainstream operating systems and available for a few more (and in the past supported many more). A global community of programmers develops and maintains C Python, a free and open-source reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and C Python development. It currently ties with Java as the second most popular programming language in the world.

**a.Features**

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by meta programming and meta objects (magic methods)). Many other paradigms are supported via extensions, including design by contract and logic programming. Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.

Python’s design offers some support for functional programming in the Lisp tradition. It has filter, map, and reduce functions; list comprehensions, dictionaries, sets, and generator expressions. The standard library has two modules (itertools and functools) that implement functional tools borrowed from Haskell and Standard ML.

The language’s core philosophy is summarized in the document The Zen of Python (PEP 20), which includes aphorisms such as:

• Beautiful is better than ugly.

• Explicit is better than implicit.

• Simple is better than complex.

• Complex is better than complicated.

Rather than having all of its functionality built into its core, Python was designed to be highly extensible. This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications. Van Rossum’s vision of a small core language with a large standard library and easily extensible interpreter stemmed from his frustrations with ABC, which espoused the opposite approach.

Python strives for a simpler, less-cluttered syntax and grammar while giving developers a choice in their coding methodology. In contrast to Perl “there is more than one way to do it” motto, Python embraces a “there should be one—and preferably only one—obvious way to do it” design philosophy. Alex Martelli, a Fellow at the Python Software Foundation and Python book author, writes that “To describe something as ‘clever’ is not considered a compliment in the Python culture”.

Python’s developers strive to avoid premature optimization, and reject patches to non-critical parts of the CPython reference implementation that would offer marginal increases in speed at the cost of clarity. When speed is important, a Python programmer can move time-critical functions to extension modules written in languages such as C, or use PyPy, a just-in-time compiler. Cython is also available, which translates a Python script into C and makes direct C-level API calls into the Python interpreter. An important goal of Python’s developers is keeping it fun to use. This is reflected in the language’s name — a tribute to the British comedy group Monty Python — and in occasionally playful approaches to tutorials and reference materials, such as examples that refer to spam and eggs (from a famous Monty Python sketch) instead of the standard foo and bar.

A common neologism in the Python community is pythonic, which can have a wide range of meanings related to program style. To say that code is pythonic is to say that it uses Python idioms well, that it is natural or shows fluency in the language, that it conforms with Python’s minimalist philosophy and emphasis on readability. In contrast, code that is difficult to understand or reads like a rough transcription from another programming language is called unpythonic. Users and admirers of Python, especially those considered knowledgeable or experienced, are often referred to as Pythonistas.

**b.OpenCV**

Open CV (Open-Source Computer Vision Library) is a open source computer vision software library for the purpose of machine learning. Open CV was developed to serve the purpose of computer vision applications and to stimulate the usage of machine perception in the commercially viable products. Open CV is a BSD-licensed product which is easy for the utilization and modification of the code. The library contains more than 2500 advanced algorithms including an extensive set of both typical and state-of-the-art computer vision and machine learning algorithms. These algorithms can be employed for the detection and recognition of faces, identification of objects, extra fake notes of 3 D models of objects, production of 3 D point clouds from stereo cameras, stitching images together for production of a high resolution image of an entire scene, finding similar images from an image database, removing red eyes from images taken using flash, following ye movements, recognition of scenery and establishing markers to overlay it with intensified reality etc. It includes C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. Open CV mainly involves real-time vision applications taking advantage of MMX and SSE instructions when available. A full-featured CUDA and Open CL interfaces are being progressively developed. There are over 500 algorithms and about 10 times functions that form or back those algorithms. Open CV is written inherently in C++ and has a template interface that works harmoniously with STL containers.

**c.IDLE**

IDLE is Pythons Integrated Development and Learning Environment. IDLE is completely coded in Python, using the tkinter GUI toolkit. It works mostly uniformly on Windows, Unix and macOS. It has a Python shell window (interactive interpreter) with colorizing of error messages, code input and code output. There is a multi-window text editor with multiple undo, Python colorizing, smart indent, call tips, auto completion, and other features. Searching within any window, replacing within editor windows and searching through multiple files is possible. It also has configuration, browsers and other dialogs as well.

**CHAPTER-5**

**EXPERIMENT ANALYSIS**

**5.1 Sample Code:**

**Soil Input Image**

from PIL import Image

import stepic

import matplotlib.pyplot as plt

import cv2

import numpy as np

import scipy.misc

import random

import scipy.ndimage

import skimage.filters

import sklearn.metrics

im1=100

im1=cv2.imread('train/gas/4.jpg')

plt.imshow(im1)

cv2.imshow("Input Image", im1)

cv2.waitKey()

**RGB to Gray**

im2 = cv2.cvtColor(im1, cv2.COLOR\_BGR2GRAY)

cv2.imshow("Input Image", im2)

cv2.waitKey()

plt.imshow(im2)

**Image Size**

print(im1.size)

**Median Filter**

median\_filtered = scipy.ndimage.median\_filter(im1, size=3)

plt.imshow(median\_filtered, cmap='gray')

cv2.imshow("median filtered image", median\_filtered)

cv2.waitKey()

plt.title('median filtered image')

**Grayscale Image Histogram**

counts, vals = np.histogram(im2, bins=range(2 \*\* 8))

plt.plot(range(0, (2 \*\* 8) - 1), counts)

plt.title('Grayscale image histogram')

plt.xlabel('Pixel intensity')

plt.ylabel('Count')

**Canny Detector**

import numpy as np

import os

import cv2

import matplotlib.pyplot as plt

def Canny\_detector(img, weak\_th = None, strong\_th = None):

img = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

img = cv2.GaussianBlur(img, (5, 5), 1.4)

gx = cv2.Sobel(np.float32(img), cv2.CV\_64F, 1, 0, 3)

gy = cv2.Sobel(np.float32(img), cv2.CV\_64F, 0, 1, 3)

mag, ang = cv2.cartToPolar(gx, gy, angleInDegrees = True)

mag\_max = np.max(mag)

if not weak\_th:weak\_th = mag\_max \* 0.1

if not strong\_th:strong\_th = mag\_max \* 0.5

height, width = img.shape

for i\_x in range(width):

for i\_y in range(height):

grad\_ang = ang[i\_y, i\_x]

grad\_ang = abs(grad\_ang-180) if abs(grad\_ang)>180 else abs(grad\_ang)

if grad\_ang<= 22.5:

neighb\_1\_x, neighb\_1\_y = i\_x-1, i\_y

neighb\_2\_x, neighb\_2\_y = i\_x + 1, i\_y

elif grad\_ang>22.5 and grad\_ang<=(22.5 + 45):

neighb\_1\_x, neighb\_1\_y = i\_x-1, i\_y-1

neighb\_2\_x, neighb\_2\_y = i\_x + 1, i\_y + 1

elif grad\_ang>(22.5 + 45) and grad\_ang<=(22.5 + 90):

neighb\_1\_x, neighb\_1\_y = i\_x, i\_y-1

neighb\_2\_x, neighb\_2\_y = i\_x, i\_y + 1

elif grad\_ang>(22.5 + 90) and grad\_ang<=(22.5 + 135):

neighb\_1\_x, neighb\_1\_y = i\_x-1, i\_y + 1

neighb\_2\_x, neighb\_2\_y = i\_x + 1, i\_y-1

elif grad\_ang>(22.5 + 135) and grad\_ang<=(22.5 + 180):

neighb\_1\_x, neighb\_1\_y = i\_x-1, i\_y

neighb\_2\_x, neighb\_2\_y = i\_x + 1, i\_y

if width>neighb\_1\_x>= 0 and height>neighb\_1\_y>= 0:

if mag[i\_y, i\_x]<mag[neighb\_1\_y, neighb\_1\_x]:

mag[i\_y, i\_x]= 0

continue

if width>neighb\_2\_x>= 0 and height>neighb\_2\_y>= 0:

if mag[i\_y, i\_x]<mag[neighb\_2\_y, neighb\_2\_x]:

mag[i\_y, i\_x]= 0

weak\_ids = np.zeros\_like(img)

strong\_ids = np.zeros\_like(img)

ids = np.zeros\_like(img)

for i\_x in range(width):

for i\_y in range(height):

grad\_mag = mag[i\_y, i\_x]

if grad\_mag<weak\_th:

mag[i\_y, i\_x]= 0

elif strong\_th>grad\_mag>= weak\_th:

ids[i\_y, i\_x]= 1

else:

ids[i\_y, i\_x]= 2

return mag

frame = im1

canny\_img = Canny\_detector(frame)

plt.figure()

f, plots = plt.subplots(2, 1)

plots[0].imshow(frame)

plots[1].imshow(canny\_img)

**Feature Extraction**

import numpy as np

import cv2

from matplotlib import pyplot as plt

# read the image

img = im1

# convert image to gray scale image

gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

# detect corners with the goodFeaturesToTrack function.

corners = cv2.goodFeaturesToTrack(gray, 27, 0.01, 10)

corners = np.int0(corners)

# we iterate through each corner,

# making a circle at each point that we think is a corner.

for i in corners:

x, y = i.ravel()

cv2.circle(img, (x, y), 3, 255, -1)

plt.imshow(img), plt.show()

**Histogram**

import cv2

import matplotlib.pyplot as plt

img\_bgr = im1

plt.imshow(img\_bgr)

plt.show()

color = ('b', 'g', 'r')

for i, col in enumerate(color):

histr = cv2.calcHist([img\_bgr],

[i], None,

[256],

[0, 256])

plt.plot(histr, color = col)

plt.xlim([0, 256])

plt.show()

height, width, \_ = img\_bgr.shape

for i in range(0, height - 1):

for j in range(0, width - 1):

pixel = img\_bgr[i, j]

pixel[0] = 255 - pixel[0]

pixel[1] = 255 - pixel[1]

pixel[2] = 255 - pixel[2]

img\_bgr[i, j] = pixel

plt.imshow(img\_bgr)

plt.show()

color = ('b', 'g', 'r')

for i, col in enumerate(color):

histr = cv2.calcHist([img\_bgr],

[i], None,

[256],

[0, 256])

plt.plot(histr, color = col)

plt.xlim([0, 256])

plt.show()

**Edge Image**

import numpy as np

import cv2 as cv

from matplotlib import pyplot as plt

img = im1

classes = ['gas', 'nongas']

edges = cv.Canny(img,100,200)

plt.subplot(121),plt.imshow(img)

plt.title('Original Image'), plt.xticks([]), plt.yticks([])

plt.subplot(122),plt.imshow(edges,cmap = 'gray')

plt.title('Edge Image'), plt.xticks([]), plt.yticks([])

plt.show()

**Predicted Results**

def predict(lst):

w = ''.join(lst).split('/')[-2]

if(w == classes[0]):

print("Gas")

else:

print("Non Gas")

imagee = "test/gas/4.jpg"

img2 = cv2.imread(imagee)

print(img2.shape)

predict("Result: " + imagee)

**Food Input Image**

import cv2

from PIL import Image

from matplotlib import pyplot as plt

from skimage import data

from skimage.feature import blob\_dog, blob\_log, blob\_doh

from math import sqrt

from skimage.color import rgb2gray

import glob

from skimage.io import imread

image\_path = "C:/Users/Lenovo/Documents/Project/Code/images/carrot/4\_6.jpg"

image\_file = Image.open(image\_path)

image\_file.save("image\_name.jpg", quality=95)

example\_file = glob.glob(image\_path)[0]

im = imread(example\_file)

plt.imshow(im)

plt.show()

**RGB to Gray**

img = cv2.imread(image\_path)

gray = cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY)

ret, thresh = cv2.threshold(gray,0,255,cv2.THRESH\_BINARY\_INV+cv2.THRESH\_OTSU)

plt.imshow(gray)

**Canny-Edge Image**

import numpy as np

import cv2 as cv

from matplotlib import pyplot as plt

img = cv.imread("images/carrot/4\_6.jpg",0)

edges = cv.Canny(img,100,200)

plt.subplot(121),plt.imshow(img)

plt.title('Original Image'), plt.xticks([]), plt.yticks([])

plt.subplot(122),plt.imshow(edges,cmap = 'gray')

plt.title('Canny Edge Image'), plt.xticks([]), plt.yticks([])

plt.show()

**Calorie Value**

import numpy as np

import cv2

from matplotlib import pyplot as plt

img = cv2.imread(image\_path)

mask = np.zeros(img.shape[:2],np.uint8)

bgdModel = np.zeros((1,65),np.float64)

fgdModel = np.zeros((1,65),np.float64)

rect = (50,50,450,290)

cv2.grabCut(img,mask,rect,bgdModel,fgdModel,5,cv2.GC\_INIT\_WITH\_RECT)

mask2 = np.where((mask==2)|(mask==0),0,2).astype('uint8')

img = img\*mask2[:,:,np.newaxis]

res = img.size

calorie=res/1800

plt.imshow(img),plt.colorbar(),plt.show()

print('calorie value is',calorie)

**Gas Comparision**

fig = go.Figure()

fig.add\_trace(go.Bar(

x=crop\_summary.index,

y=crop\_summary['N'],

name='Nitrogen',

marker\_color='indianred'))

fig.add\_trace(go.Bar(

x=crop\_summary.index,

y=crop\_summary['P'],

name='Phosphorous',

marker\_color='lightsalmon'))

fig.add\_trace(go.Bar(

x=crop\_summary.index,

y=crop\_summary['K'],

name='Potash',

marker\_color='crimson'))

fig.update\_layout(title="N, P, K values comparision between crops",

plot\_bgcolor='white',

barmode='group',

xaxis\_tickangle=-45)

fig.show()

**Comparision between rainfall, temperature and humidity**

fig = px.bar(crop\_summary, x=crop\_summary.index, y=["rainfall", "temperature", "humidity"])

fig.update\_layout(title\_text="Comparision between rainfall, temperature and humidity",

plot\_bgcolor='white',

height=500)

fig.update\_xaxes(showgrid=False)

fig.update\_yaxes(showgrid=False)

fig.show()

**Correlation between different features**

fig, ax = plt.subplots(1, 1, figsize=(15, 9))

sns.heatmap(cropdf.corr(), annot=True,cmap='Wistia' )

ax.set(xlabel='features')

ax.set(ylabel='features')

plt.title('Correlation between different features', fontsize = 15, c='black')

plt.show()

**Accuracy**

from sklearn.metrics import accuracy\_score

accuracy=accuracy\_score(y\_pred, y\_test)

print('LightGBM Model accuracy score: {0:0.4f}'.format(accuracy\_score(y\_test, y\_pred)))

**5.2 PERFORMANCE METRICS**

Performance metrics are defined as figures and data representative of an organization’s actions, abilities, and overall quality. There are many different forms of performance metrics, including temperature, humidity, rainfall, pH level, nitrogen gas, potassium gas , phosphorous and overall quality in agriculture. Performance metrics can vary considerably when viewed through different industries.

If the variables rise together then the two deviations tend to be either positive together or negative together, so their products tend to be positive and the expected value yields a positive covariance. If one rises as the other falls then positive deviations in one variable tend to be associated with negative ones in the other so the products tend to be negative and the expected value yields a negative covariance. On the other hand, if there is no discernible pattern in the behaviour of the two variables then positive and negative products will tend to cancel and the covariance will be near zero. However, the size of the covariance is not predictable in any given situation, other than that its absolute value must be no greater than the product of the standard deviations of the two variables. So dividing the covariance by this product of standard deviations yields the correlation, whose value must therefore lie between -1 and +1.

**Accuracy:**

Accuracy is often the most used metric representing the percentage of correctly predicted observations, either true or false. To calculate the accuracy of a model performance, the following equation can be used:

**Accuracy=TP+TN/TP+FP+FN+TN**

accuracy score: 0.9864

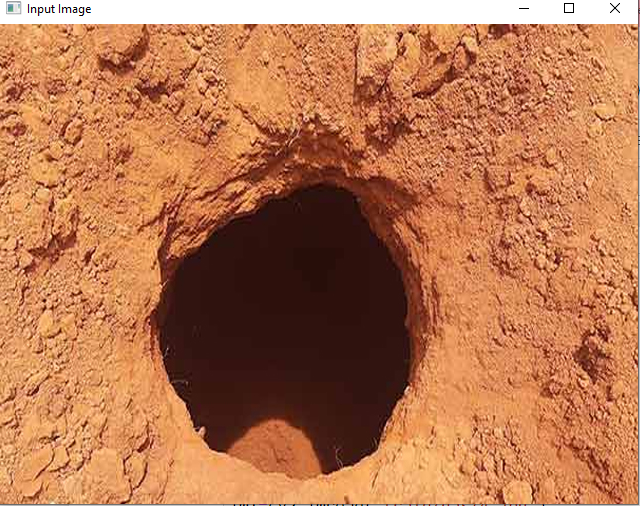
**CHAPTER – 6**

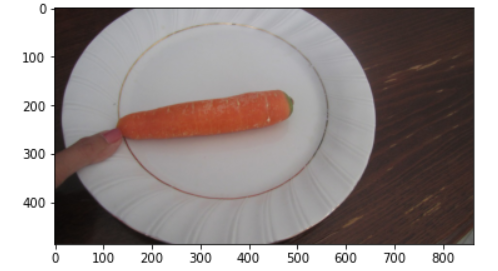
**RESULT AND DISCUSSION**

All the approaches described in the previous chapter, namely, Noises, Filters, Canny-Edge and ANN classification are applied on several images. The results are shown in below. Generally speaking, conclusions regarding the performance of these methods can be made by observing the results.

**1. INPUT IMAGE**

Below figure shows the soil and food input image.

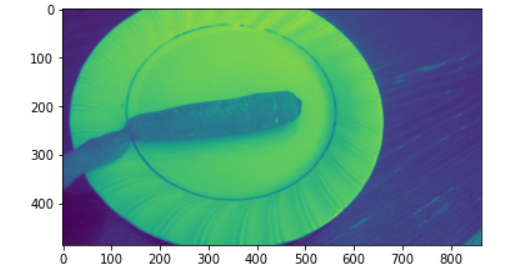




**2. GRAY SCALE CONVERSION**

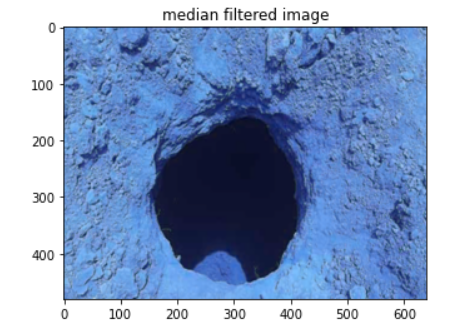
It is easy to remove the noise by using conversion of colours images to gray scale images. Below figure shows the Gray scale image.

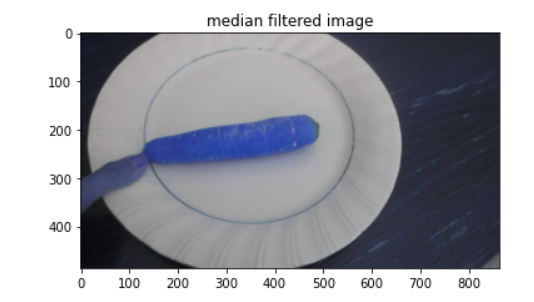




**3. MEDIAN FILTER**

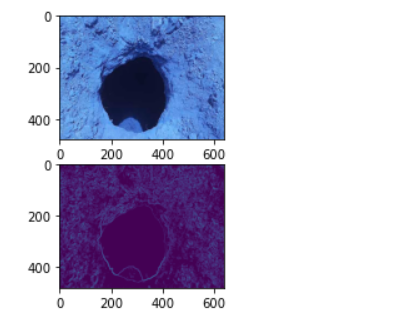
Median Filter is a one type of noise removing technique. Below figure shows the Median Filter.

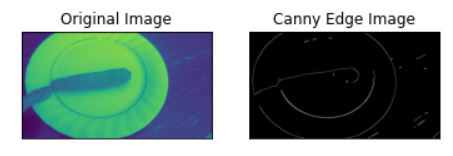




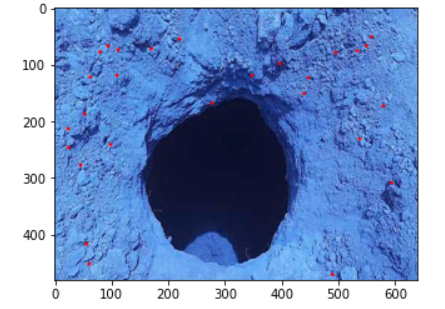
### 4.CANNY-EDGE

Canny-Edge is a one type of segmentation process. Below figure shows the Canny-Edge.





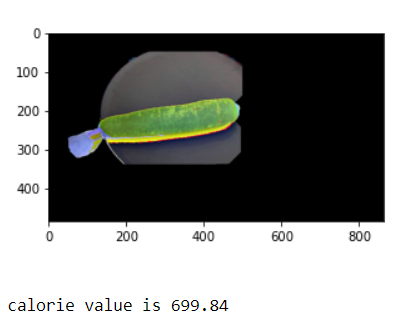
**5.FEATURE EXTRACTION**



**6.OUTPUT**

(480, 640, 3)

Gas



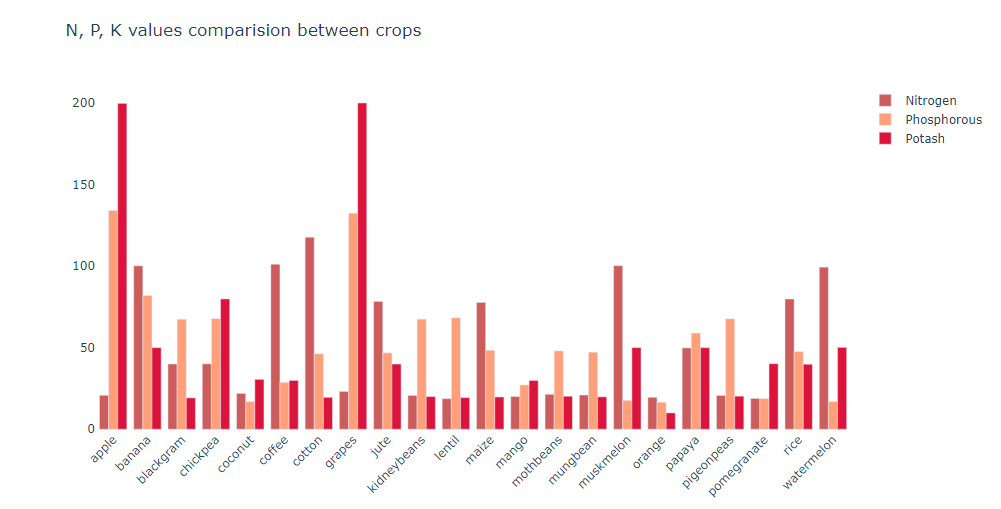
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S.No | N | P | K | Temperature | Humidity | pH | Rainfall | Label |
| 0 | 90 | 42 | 43 | 20.879744 | 82.0027 | 6.50 | 202.9 | Rice |
| 1 | 85 | 58 | 41 | 21.770462 | 80.3196 | 7.03 | 226.6 | Wheat |
| 2 | 60 | 55 | 44 | 23.004459 | 82.3207 | 7.84 | 263.9 | Maize |
| 3 | 74 | 35 | 40 | 26.491096 | 80.1583 | 6.98 | 242.8 | Beans |
| 4 | 78 | 42 | 42 | 20.130175 | 81.6048 | 7.62 | 262.7 | Peas |

**Table No.5.1 Gas Measurements for paddies**

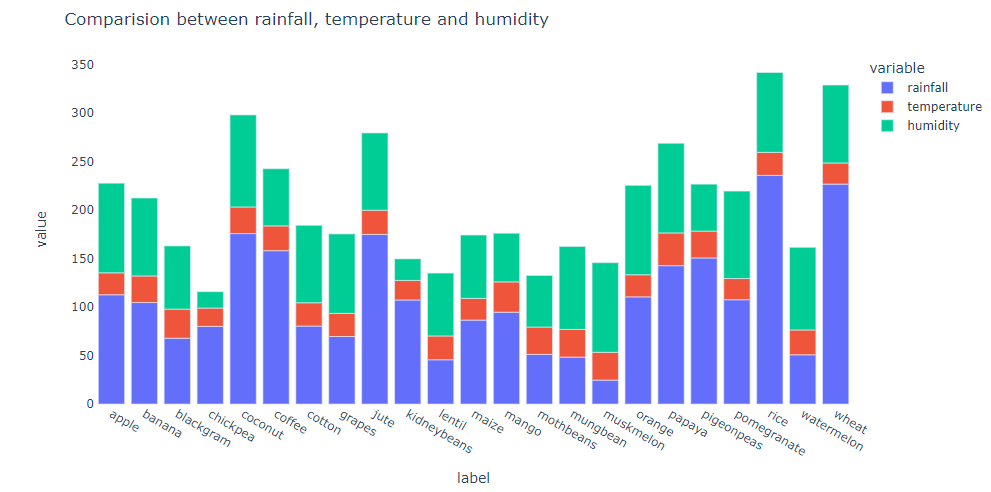
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S.No | N | P | K | Temperature | Humidity | pH | Rainfall | Label |
| 0 | 20 | 134 | 199 | 22.630942 | 92.3333 | 5.92 | 112.65 | Apple |
| 1 | 100 | 82 | 50 | 27.376798 | 80.3581 | 5.98 | 104.6 | Banana |
| 2 | 40 | 67 | 19 | 29.973340 | 65.1184 | 7.13 | 67.88 | Blackgram |
| 3 | 40 | 67 | 79 | 18.872847 | 16.8604 | 7.33 | 80.05 | Chickpea |
| 4 | 21 | 16 | 30 | 27.409892 | 94.8442 | 5.97 | 175.6 | Coconut |

**Table No.5.2 Gas Measurements for fruits**

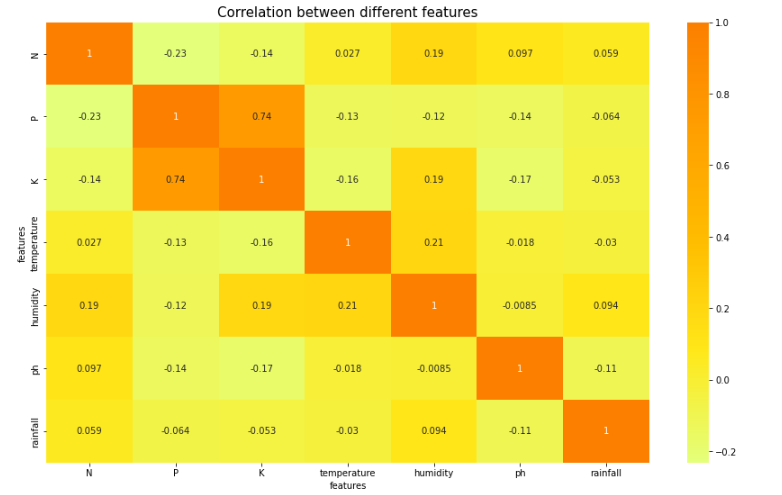
It is defined as figures and data representative of an organization’s actions, abilities, and overall quality. There are many different forms of performance metrics, including temperature, humidity, rainfall, pH level, nitrogen gas, potassium gas, phosphorous and overall quality in agriculture. Performance metrics can vary considerably when viewed through different industries.



**Fig.5.1 N,P,K values comparision between crops**



**Fig.5.2 Comparision between rainfall ,temperature and humidity**



**Fig.5.3 Correlation between different features**

### CHAPTER-7

**CONCLUSION**

Artificial neural network has the potential of playing an important role in meeting the food requirement of entire world. Agriculture is now has been revolutionized with artificial neural network. Food is considered as a basic need of human being which can be satisfied through agriculture. Agriculture plays a vital role in the economic growth of any country. With the increase of population, frequent changes in climatic conditions and limited resources, it becomes a challenging task to fulﬁl the food requirement of the present population. Machine learning has helped improving gains in agriculture soil. The work will surely be very useful for agriculture field. These innovative tools are assisting monitor soil parameters, and temperature conditions, rainfall and improve other agriculture related tasks. Accurate prediction and classification through cognitive ability of machines is difficult in varying geographical conditions. This paper investigated and analyzed the humidity, pH level and food calorie, temperature, rainfall. As more data are made publicly available from direct measurements, more synthesis tools will be needed to interpret the enormous amount of information in datasets. With the availability of more data comes the need for methods to use these data to understand how soil properties. Five predictors (air temperature, soil temperature, fertilizer amendment class, soil classification, crop) were selected to develop the models based on results from step-forward feature selection and correlation analysis. Among the f ML algorithms, the prediction based on Artificial Neural Network models achieved the highest accuracy. Soil classification was the most important predictor variable evaluated in the ANN (Artificial Neural Network) models produced from ML methods.

### CHAPTER-8

### FUTURE SCOPE

In near future artificial intelligence systems, robotics and smart sensor technology will automate the whole farming process. IoT enabled smart sensors, actuators, satellite images, robots, drones are some of the key technological revolutions that boosted the agriculture industry. These components play a vital role in collecting real-time data and accordingly making decisions without human support. Metaheuristic algorithms can be explored for nodes localization in agriculture fields in order to optimize the sensor deployment in the field and keep the minimize cost to farmers. Smart techniques play an important component in precision agriculture as these techniques quickly complete the work and reduces human labour. More ML, DL and hybrid algorithms can be explored in the agriculture industry for sustainable use of available resources. In future primarily concerned with testing the capabilities of ML algorithms for simulating methene fluxes using minimal predictors temperature, soil moisture, soil type, air temperature, cropping system, and the type of fertilization in the dataset. Soil chemical properties e.g., pH, CEC, organic carbon, and clay content were not considered as predictors because there were many values missing in the dataset that could introduce bias and reduce model performance. Future work can explore the suitability of applying other ML algorithms, e.g., artificial neural networks and XGBoost, to the image dataset to determine if hyper parameter tuning can be improved and increase the performance of the algorithms. The data which is being used to train the models and predict output is not a real time data, so with the use of IOT devices which is setup right in the farm which will give real time data. This will give real time and give better accuracy and things will be automated automatically and be more useful for the users.

**REFERENCES**

[1] Manpreet Kaur, Heena Gulati, Harish Kundra., “Data Mining in Agriculture on Crop Price Prediction: Techniques and Applications” International Journal of Computer Applications (0975 – 8887) Volume 99– No.12, August 2014.

[2] Rohith R1 , Vishnu R2 , Kishore A3 , Deeban Chakkarawarthi4, “Crop Price Prediction and Forecasting System using Supervised Machine Learning Algorithms”, IJARCCE,Vol. 9, Issue 3, March 2020

[3] B.Sahithi , T.Saheli , D.Ramanika , N.Anil Kumar, “Crop Price Prediction System using Machine learning Algorithms” , Quest Journals, Volume 6 ~ Issue 1 (2020) pp: 14-20

[4] Rachana P S1, Rashmi G2, Shravani D3, Shruthi N4, Seema Kousar R5, “CROP PRICE FORECASTING SYSTEM USING SUPERVISED MACHINE LEARNING ALGORITHMS”, IRJET, Volume: 06 Issue: 04 | Apr 2019

[5] Shabnam Shaikh, Gargi Dhuri, Harshada Mhetre, Prajakta Raut, Prof. Amol Dhakne, “Methanesoil Food calori Value Prediction And Crop Recommendation”, IRJEET, (E- ISSN 2348-1269, P- ISSN 2349-5138)

[6] Utpal Barman, Ridip Dev Choudhury , “Prediction of Methanesoil food calori using Smartfood calorione based Digital Image Processing and Prediction Algorithm”.

[7] Kingsley JOHN 1,\*, Isong Abraham Isong 2 , Ndiye Michael Kebonye, “Article Using Machine Learning Algorithms to Estimate Methanesoil Organic Carbon Variability with Environmental Variables and Methanesoil Nutrient Indicators in an Alluvial Methanesoil”.

[8] “Developing Crop Price Forecasting Service Using Open Data from Taiwan Markets”, Yung-Hsing Peng, ChinShun Hsu, and Po-Chuang Huang, 2017 IEEE.

[9] “Agricultural Production Output Prediction Using Supervised Machine Learning Techniques”, Md. Tahmid Shakoor, Karishma Rahman, Sumaiya Nasrin Rayta, Amitabha Chakrabarty, 2015 IEEE

[10] Monali Paul, Santosh K. Vishwakarma, Ashok Verma, “Analysis of Methanesoil Behavior and Prediction of Crop Yield using Data Mining approach”, 2015.