# ESTIMATION OF ABOVE GROUND BIOMASS(AGB) USING REGION OF INTEREST(ROI)

**A PROJECT REPORT**

***Submitted by***


**DARSHETA D (111919104024)**

**MARIYA ANTHONY VISUVASAMMAL S (111919104070)**

**MARY RAJAM A (111919104071)**


***in partial fulfillment for the award of the degree***

***of***

***BACHELOR OF ENGINEERING***

**IN**

**COMPUTER SCIENCE AND ENGINEERING**


**S. A. ENGINEERING COLLEGE, CHENNAI 600 077**

**ANNA UNIVERSITY: CHENNAI 600 025**


**APRIL 2023**

# ANNA UNIVERSITY: CHENNAI 600025

## BONAFIDE CERTIFICATE

Certified that this Project Report **"ESTIMATION OF ABOVEGROUND BIOMASS (AGB) USING REGION OF INTEREST(ROI)"** is the bonafide work of **"DARSHETA D(111919104024), MARIYA ANTHONY VISUVASAMMAL S (111919104070), MARY RAJAM A(111919104071)"**, who carried out the project work under my supervision.

**SIGNATURE**

Dr.R.Geetha, M.E., Ph.D.,
**HEAD OF THE DEPARTMENT**
Professor
Department of
Computer Science and Engineering
S.A.ENGINEERING COLLEGE
Avadi-Ponnamallee Main Road,
Thiruverkadu Post,
Chennai-600 077.

**SIGNATURE**

Dr.R.Geetha, M.E., Ph.D.,
**SUPERVISOR**
Professor
Department of
Computer Science and Engineering
S.A.ENGINEERING COLLEGE,
Avadi-Ponnamallee Main Road,
Thiruverkadu Post,
Chennai-600 077.

**Submitted to Project and Viva Examination held on _____**

**INTERNAL EXAMINER** **EXTERNAL EXAMINER**

## ACKNOWLEDGEMENT

We owe a great many thanks to a great many people who helped and supported us during the completion of our project.

We take this opportunity to express our profound gratitude and deep regards to our Founder Chairman **(Late)** Shri**. D. SUDHARSSANAM, M.L.A,** our Chairman Shri**. D. DURAISWAMY**, our humble Secretary Shri**. D. DASARATHAN** our correspondent Shri.**S. AMARNAATH**, and our Director Shri**.D.SABARINATH**, for their exemplary guidance, monitoring and constant encouragement throughout the course of this thesis. The blessing, help and guidance given by them time to time shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to thank our beloved Principal **Dr.S.RAMACHANDRAN M.E., Ph.D.,** for extending his support. Also, we sincerely thank **Dr. R. GEETHA, M.E., Ph.D.,** Head of the Department of Computer Science and Engineering as well as our internal guide for her guidance and encouragement in our project work which helped us in completing this task through various stages.

We also obliged to convey our special and sincere gratitude to our project coordinator **Ms. S. ARUMAI SHINY, M.E.,** Assistant Professor**,** CSE Department and **Ms .G. BELSHIA JEBAMALAR, M.E.,** Assistant Professor for their encouragement and concern during the review session along with their valuable advice, knowledge and expert guidance in our project.

Lastly, we thank almighty, our parents, all the teaching and non- throughout teaching staff for their support and guidance our project and for their constant encouragement without which this project would have not been possible.

## ABSTRACT

The management of the temperature by forests is crucial, because they provide a diverse spectrum of ecological functions. The terminology AGB is defined as "the aboveground standing dry mass of live or dead matter from tree or shrub (woody) life forms, calculated as a mass per unit area". AGB is critical in the research of the carbon cycle and climate change in the worldwide terrestrial environment. The ROI-based AGB estimation method works well at the local level. The biomass of the subtropical woods in the areas of India, Indonesia and Thailand is estimated in this project using data from Deep Globe. The model uses Convolutional Neural Network (CNN) to predict the level of carbon in the atmosphere. The accurate level of carbon which is absorbed by the trees is found using empirical formulae. By the collaboration of several datasets and modelling algorithms, this study offers a novel ROI-based method for estimating AGB for subtropical forests that improves accuracy by above 90%. The main factor that is essential for estimating the carbon absorbed by the trees will be the height and diameter of the tree. The height and diameter are found first and then certain calculations are performed to find the carbon trapped inside the tree. The project is done using MATLAB by incorporating the Deep learning technique i.e., CNN along with Region of Interest (ROI) and Gray Level Co-occurrence Matrix(GLCM).



**Keywords:** MATLAB, neural networks, region of interest, rgb, biomass estimation, carbon absorption.

# TABLE OF CONTENTS

# List of Figures

# CHAPTER 1

# INTRODUCTION

Above Ground Biomass estimation plays a major role in estimating the carbon level present in the atmosphere. The amount of carbon present in the atmosphere is something that is considered as very important to reduce the life-threatening factors. One among those factors is global warming. In general, the trees absorb the carbon di oxide present in the environment which is let out by humans and releases oxygen which is very essential for humans to lead life. This balances our eco system. As the trees are absorbing carbon di oxide it can be easily used to detect the amount of carbon present in the environment.

This work mainly aims on the amount of carbon absorbed by the trees so that the level of carbon can be found. Based on the amount of carbon absorbed by the trees, it can be found whether the carbon level in the particular area is high, low or average. In that way the amount of carbon can be found and also the hazardous effects of the carbon can also be controlled by taking further steps like planting more trees and reducing the reason for carbon emission. This helps in balancing out the carbon in the atmosphere in such a way that only essential carbon will be present in the atmosphere.

By using Region of Interest, the amount of carbon can be estimated in trees by focussing on a particular area. Region of Interest is used to take a particular area in the image and the image is analysed to clean all the noises and enhance the image in such a way that it is perfect to do further estimations. After enhancing the image is processed to separate the green area which is the essential part when it comes to estimating the carbon level. With the help of Histogram, the image can be represented in terms of graph by representing it in the form of graph. The graph is drawn based on the RGB values in the image. The green, blue and red value is represented separately which can then be used to separate different shades of green.

In the segmentation process, K-means clustering and Binary Classification is used to analyse the image and cluster different elements in the image. With the help of CNN the level of carbon present in the atmosphere will be estimated. Apart from estimating the carbon, using classification, various features of Image such as Skewness, Contrast, Corelation, Energy, Mean, Standard Deviation, Entropy, RMS, Variance, Kurtosis will be extracted. Finally using empirical formulae which is based on the height and width of the tree, the amount of carbon trapped in the tree will be calculated. The carbon level is found with the help of height and width of the tree which is found by the model that we created and using the height and width of the tree green weight is found using the formula When we half the amount of green weight, it gives dry weight and when we half the amount of green weight that we found, we will get the Carbon level of the tree or a specific area.

### 1.1 Machine Learning

Machine learning (ML) is a data analysis technology that automates the generation of analytical models. It is an example of artificial reasoning, which is generally defined as a machine's ability to understand human behaviour and act in human-like ways. Machine learning (ML) uses historical or previous data as input and bases new output values on the historical data. Massive amounts of data are used by ML algorithms both throughout the learning process and when making predictions about the future. The stages include the following: observation, data analysis, data checking, data verification, data prediction. Image recognition is a noteworthy and comprehensive example of ML in service. Considering the power of a digital image, it may distinguish an object as the pixels in photos with shading or high contrast. Accessible information can be categorised by ML into groupings, which are then defined by rules by examiners. When the grouping is done, the experts can determine the likelihood of a problem. ML has the ability to separate structured data from unstructured data. The most popular method of commenting on for predictive investigation devices is computerised by an ML computation. Machine learning is mostly used for generation, prediction, and

detection. This phrase describes a system's ability to acquire and assimilate information through broad-based perceptions as well as to develop and expand by picking up new knowledge rather than having it already be part of the system. ML is currently being used in a variety of projects and fields. Examples include clinical finding, image handling, forecasting, planning, learning association, and more. Results from ML applications are reliant on prior knowledge. Many businesses uncover new efficiencies and reduce waste with the aid of machine learning algorithms in order to save money.

### 1.2 Deep Learning

Deep learning is a subset of a larger collection of machine learning approaches that combine representation learning with artificial neural networks. There are three different types of learning: supervised, semi-supervised, and unsupervised. Deep-learning Deep neural networks, deep belief networks, deep reinforcement learning, recurrent neural networks, convolutional neural networks, and transformers have been used in areas such as computer vision, speech recognition, natural language processing, machine translation, bioinformatics, drug design, medical image analysis, climate science, material inspection, and board game programmes. These applications have produced outcomes that are equivalent to, and in some cases superior to, those obtained by traditional methodologies. ANNs were inspired by biological information processing and distributed communication nodes. In a variety of ways, ANNs vary from biological brains. Artificial neural networks, in particular, are static and symbolic, whereas most live species' organic brains are dynamic (plastic) and analogue. The term "deep" in deep learning refers to the network's use of several layers. Early research proved that a linear perceptron cannot be a universal classifier, but a network with a nonpolynomial activation function and one unbounded width hidden layer might. Deep learning is a modern variation that focuses on an infinite number of bounded-size layers, allowing for practical application and streamlined implementation while preserving theoretical universality under mild conditions. In depth, Deep learning allows layers to be

heterogeneous and stray significantly from physiologically informed connectionist models for the sake of efficiency, trainability, and understandability.

### 1.3 Region of Interest(ROI)

A region of interest (ROI) refers to a sample of the collection of data that has been defined for a specific purpose. The ROI idea is frequently used in a range of applications. It is occasionally beneficial to process a certain subregion of a photograph while leaving the rest alone. Image subregions can be easily expressed using Mathematica Graphics primitives such as the Point, Line, Circle, Polygon, or just an array of vertex locations.

### 1.4 MATLAB

MATLAB (acronym for "MATrix LABoratory") is a patented multi-paradigm language for programming and numeric computing environment developed by MathWorks. MATLAB provides capabilities for matrix manipulation, function and data visualisation, algorithm implementation, user interface design, and interaction with programmes written in other languages. Though MATLAB is designed mainly for numerical calculation, symbolic computing skills are available through an optional toolbox that uses the MuPAD symbolic engine. Simulink is a software add-on that allows graphical multi-domain simulation and model-based design for dynamic and embedded systems. MATLAB can call functions and subroutines developed in the C and Fortran programming languages. A wrapper function that takes and returns MATLAB data types is written. MEX files (MATLAB executables) are automatically loadable object files generated by the compilation of such routines. More two-way Python interface has been introduced since 2014. Perl, Java, ActiveX, and.NET libraries may all be called within MATLAB, and many MATLAB libraries are designed to be wrappers for Java or ActiveX libraries. Calling MATLAB from Java is more challenging, although it is doable using a MathWorks-supplied MATLAB toolbox or an undocumented approach referred to as JMI (Java-to-MATLAB Interface).

## 1.5 Open CV

OpenCV (Open-Source Computer Vision Library) is a machine learning and computer vision software open source library. OpenCV was created to offer a standard foundation for computer vision applications and to speed up the incorporation of machine perception into commercial goods. Changes and applications in business It is used for facial detection and identification, object identification, classification of human movement in footage, tracking of camera movement, and tracking of moving objects. The library contains over 2500 optimisation algorithms, such as a full set of classic and modern machine learning algorithms. To get high resolution of the whole picture, extract the 3D object model, construct a 3D point cloud from the stereo camera, and subsequently merge the photos. Set up markers like augmented reality overlays. Each member of the OpenCV community has over 47,000 users from all around the world and a predicted download volume of approximately 18 million. Companies, research groups, and government agencies all make extensive use of the library. Google, Yahoo, Microsoft, as well as Intel, IBM, Sony, Honda, and Toyota are just a few of the companies that use this collection. OpenCV is commonly used in applications such as Applied Mind, Video Surf, and Zeitera. Street view photography, Israeli CCTV intruder identification, keeping track of of Chinese mining machines, assisting robots in shifting and picking up items in willow garages, drowning cases in Europe as a whole augmented reality displays in Spain and New York, trash tracking in the region of Turkey, tagging of all items worldwide of factories inspect products in Japan, and even performing facial recognition quickly. It supports C++, Python, Java, and MATLAB and is interoperable with Windows, Linux, Android, and Mac OS. When MMX and SSE instructions are available, OpenCV concentrates on real-time vision applications. CUDA and OpenCL interfaces are completely working. It is now actively evolving. There are over 500 algorithms and several functions that comprise or assist these algorithms. Approximately ten times. OpenCV is developed in C++ and provides a template interface that works well with STL containers.

### 1.6 Tensor Flow

TensorFlow is an open-source and free data flow and differentiated programming framework for a variety of applications. This is a representational math library used in ML applications like neural networks. It may be utilised for Google research and manufacturing. TensorFlow is Google Brain's second generation system. On February 11, version 1.0 was published, and the original implementation is now operating on selected devices. TensorFlow may run on many GPUs and central processing units (with optional CUDA and SYCL additions for general-purpose GPU computation). Tensor Flow- is available for Linux, macOS, Windows, and mobile computing devices (including Android & iOS) in 64-bit version. Tensor Flow neural the term is used with multidimensional data matrices called tensors, and the flexible architecture allows computers to be easily deployed on many platforms (CPU, GPU, TPU), as well as desktop to servers' groups mobile devices, & Tensor Flow is used along with multidimensional data matrices known tensors. Jeff Dean revealed during the Google I/O conference held around June 2016 that TensorFlow was cited in 1,500 GitHub projects, just 5 of which were owned by Google. Unlike other deep learning numerical libraries such as Theano, TensorFlow is intended for study and manufacturing systems, such as Rank Brain in Google Search as well as the intriguing Deep Dream. It is capable of running on single-processor systems, GPUs, mobile devices, and huge distributed systems along with hundreds of machines.

### 1.7 Keras

Keras is an API for individuals rather than machines. Keras follows best practises to decrease cognitive load: it offers a uniform and straightforward API, minimises user input for typical use cases, and delivers clear and responsive error messages. It also includes detailed documentation and developer tutorials. Keras offers numerous frequently used neural network building pieces, including as layers, goals, triggers, optimizers, and many tools to ease the processing of picture and text input and the code necessary for deep writing. The neural network code is available on GitHub, as well as the community support forum includes a GitHub problem page as well as a

Slack channel. Keras is a lightweight Python deep learning package that works with Theano and Tensor Flow. Deep learning study and development of models should be implemented as rapidly and easily as feasible. It is compatible with Python 2.7 or 3.5, if the fundamental framework is supplied, it can operate without issue on GPU and CPU. It is distributed under the MIT licence. Keras is created and maintained by Google developer François Cholet in accordance with the four guiding principles listed below: Modularity: The model may be thought of as either a series or a graph. A deep learning model's challenges are separate components that may be assembled in any way.

# CHAPTER 2
# LITERATURE SURVEY

**1) "**Above-ground biomass estimates based on active and passive microwave sensor imagery in low biomass savanna ecosystem"-2018

By exploring the connection between the space-borne radar data and AGB at ten tonnes per hectares as well as below, this work intends to fill a gap in the scant research done for low primary production zones like grasslands, steppes, or savannas. Introduce inactive brightness temperatures as an additional covariate for biomass estimation, with the concept that it contains information in addition to the microwave backscatter from active sensors. The results indicate that large-scale estimates of AGB for grassland and savanna may be obtained with high precision (R2 up to 0.52). It was also revealed that the inclusion of passive radar can increase the quality of AGB calculations in regard to explained variance for certain cases.

**2) "**Estimation of maize above-ground biomass based on stem-leaf separation strategy integrated with LiDAR and optical remote sensing data" -2019

In this work, the stem-leaf separation method paired along with drone LiDAR and a multispectral imagery is presented for estimating the AGB in maize. First, the correlation matrix was utilised to optimise LiDAR structure parameters (LSPs) and spectral vegetation indices (SVIs). Based on the screening indicators, the SVIs and LSPs were subjected to multivariate linear regression (MLR) with the above-ground leaf biomass (AGLB) and above-ground stem biomass (AGSB), respectively. Simultaneously, utilising the AGLB and AGSB, all SVIs produced from multispectral data and all LSPs created from LiDAR data were sent to PLSR. Finally, the AGB was determined by adding the AGLB and AGSB estimates from the MLR and PLSR techniques, respectively. Furthermore, combining LiDAR and multispectral data produces more accurate AGB estimations than either LiDAR or multispectral data by

itself, with R2 increasing significantly by a value of 0.13 and 0.30, accordingly, and the RMSE and NRMSE decreasing by 22.89 and 54.92 g/m2 and 4.46% & 7.65%, respectively. This study improves AGB accuracy in predictions and suggests a new monitoring guideline by combining multispectral and LiDAR data.

**3) "**A method to avoid spatial overfitting in estimation of grassland above-ground biomass on the Tibetan Plateau" -2021

Proper grasslands above-ground biomass (AGB) analyses are crucial for the long-term use and protection of grasslands resource and the environment. In this study, a random forest (RF) model was programmed with forward feature selection (FFS) and leave location out the cross-validation (LLO-CV) techniques to predict the dry weight (DW) of grassland AGB based on a variety of characteristics. The model's performance was R2 = 0.66, with an average Root Mean Square Error (RMSE) of 503.86 kg DW/ha & a mean absolute error (MAE) of 376.51 kg DW/ha. The geographic distribution of grassland AGB extended from north-west to south-east throughout the Tibetan Plateau (TP) between 2001 and 2018. Grassland AGB increased more than it dropped over the study period (70.6% vs 29.4%, correspondingly). This study describes a robust method for increasing the flexibility of machine learning methods for forecasting grasslands AGB in unknown places.

**4) "**Biomass estimation and mapping of Cangio Mangrove biosphere reserve in South of Vietnam using ALOS-2 PALSAR-2 data" -2019

In this study, radar data from the ALOS-2 PALSAR-2 satellite is used to create biomass estimation models and ultimately a biomass database in the Can Gio Mangrove Biosphere Reserve. The estimations of biomass were assessed using a single variable regression and multivariate regression technique, using 30 plots of sample data for the model used for training and 15 samples plot for the validation model, in addition to the coefficient of determination (R 2) and RMSE. HV polarisation was shown to be substantially related to biomass in regression analysis, with linear models, exponential

approaches (R 2 = 0.69; RMSE = 28.73), and polynomial models (R 2 = 0.76; RMSE = 28.03) exhibiting the greatest association. Nonetheless, no significant relationship was found between the HH polarisation & above-ground biomass, considering the linear model (R 2 = 0.42), the exponential model of biomass (R 2 = 0.46), or the model based on polynomials (R 2 = 0.42). Multiple linear regression was carried out as well between the radar image features and the field biomass. R 2 relationship between biomass along with two distinct variables (HH and HV) equaled 0.79, as well as the root mean squared error (RMSE) was 29.78. Nonetheless, the model combining the HV variable and eight texture variables yielded an improved outcome (R 2 = 0.81; RMSE = 27.76), describing 81% of the difference in forest biomass. This model was used to construct the biomass above ground mapping at Can Gio the biosphere Reserve in southern Vietnam.

**5) "**High-accuracy Machine Learning Models to Estimate above Ground Biomass over Tropical Closed Evergreen Forest Areas from Satellite Data"-2021

Manually done field measurements have traditionally been employed to estimate the amount of biomass stored in forested areas, which is expensive, time consuming, and doesn't scale well across large areas. Using a large-scale training dataset generated from a thorough biomass mapping project in the Democratic Republic of the Congo (DRC), this study investigates the possibility for calculating the total amount of Above Ground Biomass (AGB) using publicly available satellite data and machine learning methods. Various model designs were investigated, encompassing recent state of the art tree based models as well as deep neural network (DNN) models. DNN models were identified to provide a slight improvement in accuracy and may be used for further fine tuning using small local datasets for use outside of the DRC.

**6) "**Above-ground biomass references for urban trees from terrestrial laser scanning data"-2021

This paper discusses the possibility for terrestrial laser scanning (TLS) approaches to collect highly accurate data on urban tree structure and AGB. Before being felled and weighed, a total of fifty urban trees in 7 Swiss cities were surveyed using TLS and standard forest inventory procedures. Quantitative structure modelling was used to reconstruct the TLS cloud of points tree building, volume, and AGB. AGB estimates resulting from TLS were contrasted with AGB estimates constructed solely on forests trees allometries around breast height. The relationships between multiple tree parameters as AGB predictions were studied. while contrasted with destructively harvested references, TLS-derived AGB estimates performed well, having a R2 of 0.954 (RMSE = 556 kilogrammes) vs 0.837 (RMSE = 1159 kg) for allometrically produced AGB estimations. According to a correlation analysis, several TLS-derived hardwood volumes estimations, combined with trunk widths and tree crown measurements, outperform tree height in identifying total wood AGB. TLS-based wood volume calculations have the ability to accurately estimate tree AGB irrespective of the type of tree, size, or form. This allows us to acquire highly precise non-destructive AGB estimates that may be used to construct new allometric equations without the need for time-consuming destructive harvesting.

**7) "**Above-ground biomass prediction by Sentinel-1 multitemporal data in central Italy with integration of ALOS2 and Sentinel-2 data" -2018

The objective of this study is to employ Sentinel-1 SAR multitemporal data, in addition to multispectral & SAR data at other wavelengths, to fine-scale map the above-ground biomass (AGB) at the provincial scale in a Mediterranean woodland environment. When using integrated sensors and an upper bound of 400 Mg ha1, the regression findings demonstrate great accuracy in prediction (R2 14 0.7). Despite the fact that broadleaf woodlands had a varying reactivity in backscatter all through the course of the year, multitemporal SAR data proved effective in picking appropriate

Sentinel-1 data. In the event SAR multifrequency information or a mix of SAR and optical information were employed, prediction accuracy was equivalent.

**8) "**Assessment of volume and Above-Ground Biomass in Araucaria Forest through satellite images, comparing different methods in the south of Chile"-2022

The research presented compares findings from several existing and novel methodologies for biomass estimate utilising data of varying resolutions, including open and commercial images (e.g., Sentinel-2A, Landsat-8, Quick Bird, etc.). As a pilot research area, the goal is to determine the influence of resolution on the estimate of forest stand characteristics in Araucaria forest type. To estimate volume and aboveground biomass from ground samples, precise allometric equations are utilised. To generate detailed biomass values by tree component, a non-linear model with more than one covariate is utilised (e.g., wood, foliage, etc)

**9) "**The use of machine learning methods to estimate aboveground biomass of grasslands"-2021

This research investigates the use of machine learning (ML) techniques towards calculating the above-ground biomass (AGB) in grasslands environments. We do a literature review of the topic using 26 recent studies to identify common practises, including the link between evaluating efficiency and the ML technique used, sources of data, and size (local/regional). To establish the link between study characteristics and estimate accuracy, we use descriptive and correlation analysis. Despite a rise in the number of studies and application examples, there is no evidence that the estimating performance of the algorithms has improved over time. The number of field samples collected, the source of the RS data, and the species in the mix of grasslands systems are the most important variables in explaining estimate accuracy in all approaches. Accuracy increases with sensor proximity to the field, implying that field spectroscopy is more accurate than satellite imagery on average. There is no evidence that one machine learning technique is more appropriate for this issue than another. The research

literature also has significant limits in terms of the application of ML algorithms. For instance, just a few studies confirmed the models, casting doubt on their wider applicability. Despite these constraints, and in light of the proven progress, we predict that ML techniques when combined with RS/proximal information will keep on improving and be beneficial for farm management throughout the near future.

**10) "**Modelling Alpine Grassland Above Ground Biomass Based on Remote Sensing Data and Machine Learning Algorithm: A Case Study in East of the Tibetan Plateau, China" - 2022

When combined with meteorology, soil, topography, geography, and in situ measured AGB data (RF), a total of five out of the ten variables (elevation, slope, aspect, topographic position, temperature, precipitation, and the concentration of clay and sand in the soil) produce significant effects on grassland AGB, with R2 0.04-0.39 and RMSE 859.68-1075.09 kg/ha, respectively. The RMSE of the RF model was lowered from 26.45% to 44.27% when compared with univariate/multivariate parameter models. In the meantime, RF models can explain 89.41% of the variation in AGB across the grass growing season. This study provided a more suitable RS inversion model which made use of MODIS vegetation indexes and other influencing factors. Furthermore, the precision of calculations based on MODIS data has been significantly improved.

# CHAPTER 3
# SYSTEM ANALYSIS

## 3.1 EXISTING SYSTEM

The already existing model to estimate the above ground biomass and it is done using various machine learning techniques such as XG Boost, support Vector Machine (SVM), random forest and also Neural network. The model does not give a specific value of carbon which is present in the particular area or of the specific tree. The Existing system gives an accuracy of only about 90% which is only the estimation of above ground biomass (AGB) and the estimation is based on a particular area. The XGBoost model performed best when Landsat 8 and Sentinel-1A pictures were used as predictor variables. In contrast to the LR technique, the F-test results showed that the RF and XGBoost algorithms significantly improved AGB estimation. On XGBoost, the parameter optimisation on XG Boost was shown to be more significant than on Random Forest. The XGBoost model is an effective model for AGB estimation that can alleviate overestimation and underestimation issues. This study presents a novel method for calculating AGB value for the subtropical-forest via remote-sensing by combining several sensor datasets and modelling techniques.

### 3.1.1 Disadvantages of existing system

The Existing system does not cover a large area of land and is focussed on only a particular area. The work only focussed on the amount of above ground biomass in the atmosphere and the exact carbon level in the atmosphere is not found. The average AGB of the research region discovered in this study is less than that of a subtropical forest. There is a high number of secondary forests in the research region, and the distribution of age classes is different; it is made up of young as well as middle aged forests, with a very few dominating, mature, and over-mature woods. Saturation is the primary cause of AGB underestimate based on remote sensing data in forests with high AGB values. Moreover, greater spatial as well as radiometric resolution remote sensing data, like

LIDAR and hyper spectral data, or the technique of mixed pixel decomposition, data cleaning, and variable selection, may be solutions for AGB estimation.

## 3.2 PROPOSED SYSTEM

The Proposed System uses Region of Interest (ROI) to particularly select an area and find out the carbon absorbed by the trees in that particular area. The system uses the RGB colour gradient to effectively differentiate the parts of the tree to give accurate results about the amount of carbon absorbed by the parts of the tree present above the soil by using GLCM. The features of the image are obtained using the CNN which helps in differentiating the different parts and areas in the image. To estimate the level of Carbon, Convolutional Neural Network (CNN) model is used which tells whether high, average or low carbon is present in the selected area and also using empirical formulae which is based on the height and width of the tree which will be obtained from the model, is used to find the exact value of carbon in the particular tree or particular area.

### 3.2.1Advantages of Proposed System

The System uses a variety of datasets which contains both aerial and drone images which can be used to estimate the carbon content. The images can be processed and the carbon can be estimated using various techniques such as Region of Interest (RoI), Gray-Level Co-occurrence Matrix (GLCM) and Convolutional Neural Network. In the previous model, exact amount of carbon is not found whereas in this work using CNN and few empirical formulae, the exact amount of carbon is found and also the level of carbon is shown. In addition to it, the features of the image is also found to know about the image more.

## 3.3 HARDWARE REQUIREMENTS

- Device name DESKTOP-0PLO7PN
- Processor Intel(R) Core(TM) i7-7600U CPU @ 2.80GHz 2.90 GHz

- Installed RAM 8 GB
- System type 64-bit operating system, x64-based processor
- Hard disk 512 SSD

## 3.4 SOFTWARE REQUIREMENTS

- Operating system : Windows 11
- IDE : MATLAB AND SIMULINK
- Coding Language : MATLAB

# CHAPTER 4

# SYSTEM IMPLEMENTATION

## 4.1 Introduction

System implementation (i.e., quality assurance) is the process of establishing how an information system should be produced (i.e., physical system design), verifying that the data within the system is operational and used, and confirming that the information system meets quality requirements. It requires both starting from scratch and constructing an entirely novel system from scratch. People have charge of the process for usage and evaluation after it has been implemented. It comprises instructing clients about how to operate the system as well as making preparations for a seamless transition.

## 4.2 Architecture Diagram

An architectural diagram is a graphical depiction of how software system components are physically implemented. It displays the overall framework of the system's software along with the interactions, restrictions, and limitations that exist between each component. The dataset is collected and pre-processed to clean the data to remove all the noises and enhance the picture quality. After pre-processing the data is in a form which is suitable for making the further estimation, the data is trained in such a way that the data will be suitable to follow all our further process. Once the model is done training, testing is done with new data to check whether the data is trained well. Then the steps to estimate the carbon level is implemented. The model's success or failure is determined by the model's accuracy, which will be found once the model

implemented.

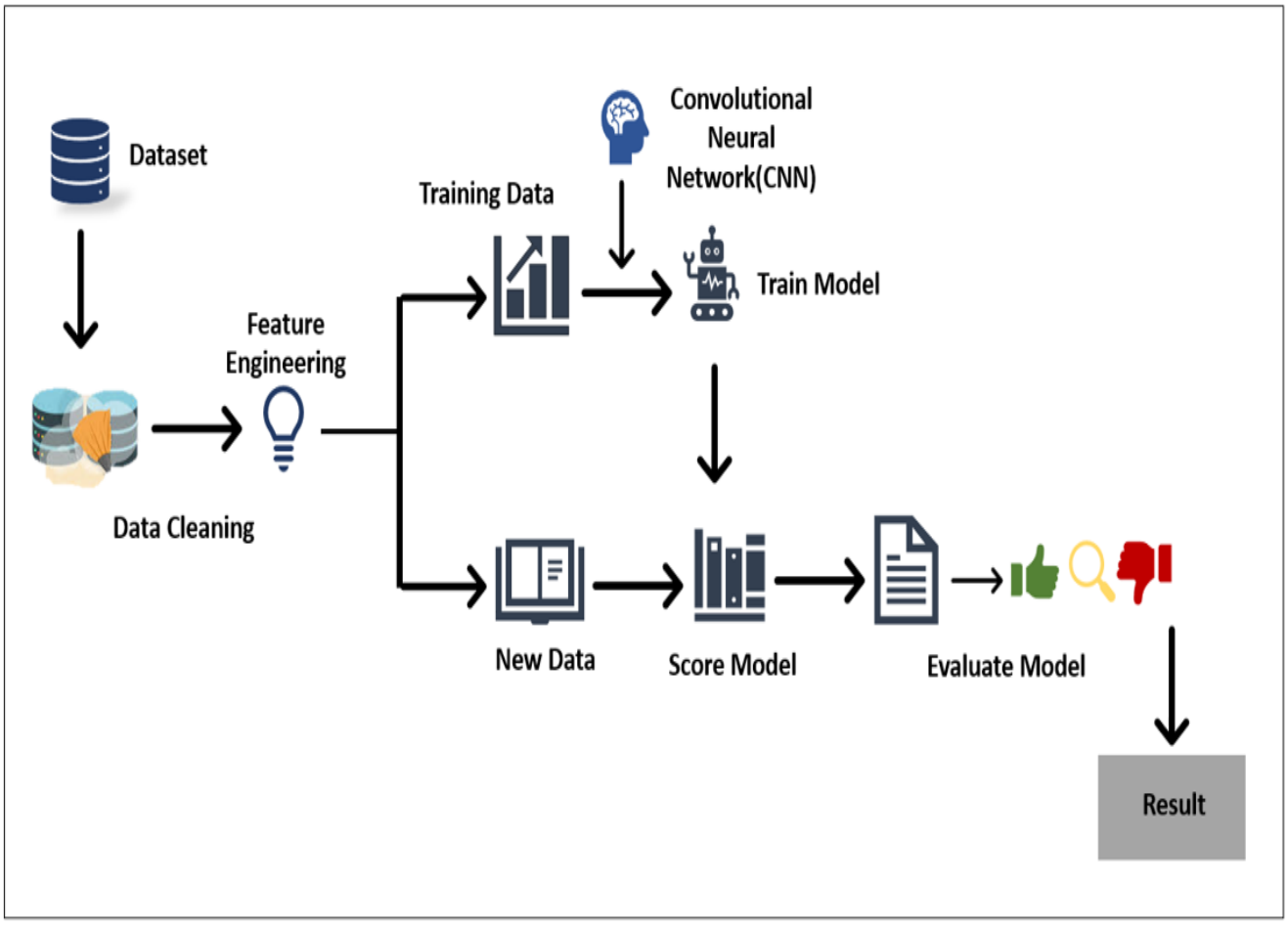


**Fig 4.1 Architecture Diagram**

## 4.3 Modular Description

**4.3.1 Green area detection** Green area identification is a critical step in carbon estimates since it allows us to estimate the quantity of carbon stored by vegetation in a specific region. The greener space there is, the more carbon is sequestered through photosynthesis. For green area detection, Region of Interest and GLCM method is used. Region of Interest (ROI) is used to take into picture a particular area in the image. The objective was to design and implement a model in MATLAB that will detect the green area in an image similar to the training image. The green area is detected to find the area filled with trees so that it will be easy to implement the model to find the amount of carbon present in the particular area. The carbon can be only found with the help of the

height and width of the tree. Overall, green area identification is an important procedure for carbon estimate since it helps us to better comprehend the amount of carbon sequestered by vegetation in a certain region. By integrating several detection technologies, we may obtain a more accurate estimate of carbon sequestration, which can be utilised to influence conservation and management policies.

**4.3.2Range of RGB in image** RGB is an abbreviation for red, green, and blue, the main colours used to produce all colours in digital pictures. Each and every pixel in an RGB image is depicted by three numbers that describe the intensity of red, green, and blue light that should be used to form the pixel's colour. Most digital picture formats use 8 bits to represent the values for each channel (red, green, and blue), which implies that each channel can have 256 potential values ranging from 0 to 255. As a result, the RGB values of an image vary from 0 to 255 for each channel. The image is analysed to find the amount of Red, Green and Blue (RGB) in the image. The RGB value in the image is found using the mean value. Our goal is to train a custom deep learning model to detect the trees in the image. The RGB is found to take into account only the green colour which represents the area filled with green where we will make the further estimations.

**4.3.3 Chrominance** In colour theory and image processing, chrominance refers to the colour information in an image that is distinct from luminance or brightness information. Chrominance is a term used in digital image processing to describe the two-colour channels of an image that are generated from red, green, and blue (RGB). The idea is to separate the brightness information of an image from its colour. Since we need to find the shades of green, applying the chrominance will help to differentiate the colours and find the different types of green. The chrominance blue and red image is applied to the image and the brightness is reduced so that the green can be differentiated in the image.

**4.3.4 Segmentation** The process of separating an image or a bigger data set into many smaller, more manageable segments or areas is referred to as segmentation.

Segmentation is frequently used in image processing to separate objects or regions of interest from the background or other sections of the picture. The segmentation process includes k-means clustering. Using Region of Interest, we take into account a particular area and separate the green area from the image. K-means clustering is used to group similar objects in the image. Here in our project, trees, water bodies are all grouped according to their own characteristics.

**4.3.5 Success and loss rate** Success and loss rates are performance metrics commonly used in machine learning and data analysis tasks, particularly in classification tasks. The image given as the input to the model is processed and the success loss rate graph is computed. Both the success rate and loss rate are computed to know whether the image will produce successful results for carbon estimation. The dataset contains almost 4000 images in total and all the images are tested to check whether they are suitable to make further estimations. Out of 4000 images after testing, 300 image is found to be suitable for making the carbon estimation. In practise, success and loss rates are often determined using a held-out test dataset in which the real labels of the instances are known and compared to the projected labels given by the classifier or model. The test dataset is frequently a small portion of the original set of data utilised for training the classifier or framework, and the size and composition of the dataset may have an influence on the accuracy and generalizability for the outcome metrics.

**4.3.6 Model Improvement** Model improvement refers to the process of enhancing the performance or accuracy of a machine learning model or algorithm. Model improvement is an iterative process that involves analysing the model's performance, identifying areas of weakness or inefficiency, and making modifications to improve the model's accuracy or efficiency. Model improvement requires careful analysis and experimentation to determine which methods are most effective for a given problem. It is important to balance performance improvements with the complexity and computational cost of the model, and to validate the improvements using appropriate evaluation metrics and test datasets.

### 4.3.7 Calculation of Carbon

**Calculating green weight (GW)** The estimation of the mass of the tree when it is mortal is known as the Green Weight. This includes all the wood content and any moisture in the tree. This allow to calculate the above-ground green weight of a tree based on the tree's diameter and height. To find the green weight, insert the values obtained for diameter (centimetres) and height (metres) into the relevant equation.

For trees with diameter < 28 cm: GW = 0.0577 x d2 x h

For trees with diameter > 28 cm: GW = 0.0346 x d2 x h

**Calculating dry weight (DW)** Dry weight represents the mass of the wood in the tree when dried in an oven, so the moisture is removed. To find the dry weight just multiply green weight by 50 percent.

$DW = GW \times 0.5$

**Calculating carbon storage (C)** Quantity of carbon in the wood of the tree is denoted as the carbon storage. This represents both the entire quantity of carbon that the tree sequesters in addition to the amount of carbon that is taken in from the atmosphere during photosynthesis. To find carbon storage, multiply dry weight by 50 percent.

$C = DW \times 0.5$

### 4.4 Specification of Modules

**4.4.1 Dataset Identification** Dataset identification refers to the process of identifying and selecting an appropriate dataset for a particular machine learning or data analysis task. The dataset is a collection of data that is used to train, validate, or test a ML model or to perform data analysis tasks. The input dataset is obtained from https://www.kaggle.com/datasets/quadeer15sh/augmented-forest-segmentation for analysis. The dataset is publicly available for research analysis. In general, it contains a

training dataset with 4000 MNIST images which is fed into the model to check whether the images are suitable to make further estimations. And in the test dataset 300 aerial images are found suitable to proceed with our calculations.

**4.4.2 Pre-Processing** Cleaning, converting, and collecting data for study or modelling is a critical stage in machine learning and data analysis. Pre-processing helps to ensure that the data is accurate, relevant, and appropriate for the specific analysis or modelling task. The photos are pre-processed with random rotation (the greatest angle of rotation was 30 degree), horizontally flips, shearing, zooming, cropping, and minor random noise disturbance. Image processing increases the quality of the image making it suitable to do calculations which provides further accurate results. The noises in the image are removed and is concerted to a form which is suitable for doing the carbon estimation. Pre-processing is essential for ensuring that the data is suitable for analysis or modelling and can have a major influence on the effectiveness of the resultant models or analyses. Pre-processing techniques are often applied iteratively, with the effects of each step evaluated before moving on to the next step.

**4.4.3 Transfer Learning -CNN** Transfer learning is a machine learning along with deep learning approach in which a previously trained model is utilised as the starting point for an entirely novel assignment or issue. Transfer learning may considerably enhance learning process speed and accuracy, especially for applications that have limited data for training or computational resources. Machine learning algorithms require a large quantity of data to perform good categorisation. The concept of machine learning was combined with transfer learning. A model that has already been trained is a network that has been previously trained on a large dataset, typically for a big-scale image classification task. Transfer learning for the classification of images works on the principle that if a predictive algorithm is trained on an enormous general dataset, it will effectively serve as a generic model. This enables training and comparison of the model. Transfer learning utilising CNNs has the potential to significantly reduce the amount of training information and computational materials required for a new assignment while

retaining high accuracy and performance. It is particularly useful in computer vision applications such as image categorization, object identification, and semantic segmentation.

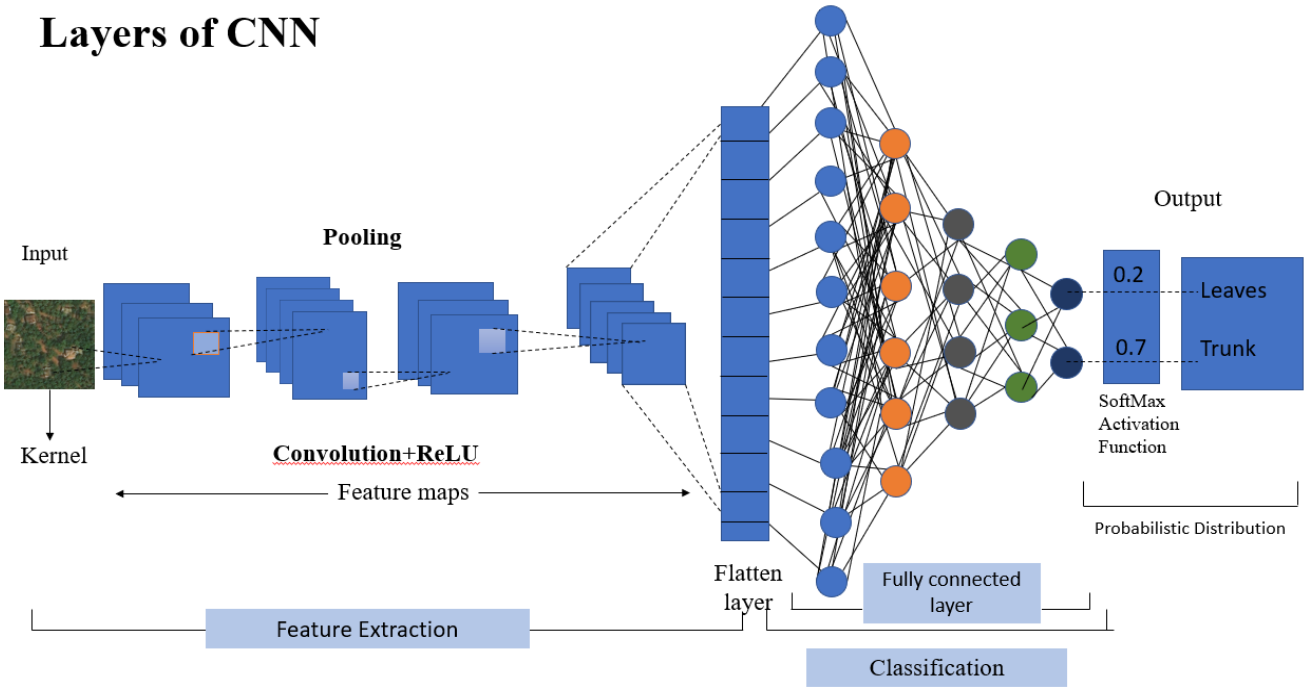


**Fig 4.2 Layers of CNN**

### 4.4.4 U- Net Architecture

The design of the U-Net is used for semantic segmentation. It is made up of two paths: one shrinking and one expanding. The U-Net design is made up of an encoder and a decoder that are linked by a bottleneck layer. The network's encoder conducts downsampling and feature extraction, while the network's decoder does upsampling and feature reconstruction. The bottleneck layer connects the encoder and decoder, allowing the network to keep spatial information intact. The U-Net design has a characteristic U-shaped topology, with skip links between the encoder and decoder levels. The skip connections allow the network to incorporate information from several picture scales, enhancing its capacity to appropriately segment complicated forms and structures. The network's encoder is generally made up of many convolutional layers followed by a pooling layer that downsamples the feature maps to minimise spatial dimensionality.

The network's decoder conducts upsampling using transposed convolutional layers to restore the feature maps to their original dimensions. The U-Net architecture's skip connections connect the relevant levels of the network's encoder and decoder components, allowing the network to recover spatial information lost during downsampling. The skip connections serve to prevent the vanishing gradient problem from occurring in deep neural networks. Other forms of segmentation tasks, such as semantic segmentation, instance segmentation, and panoptic segmentation, may also be handled by the architecture, which can be tweaked and expanded. Overall, the U-Net architecture is a strong and adaptable tool for picture segmentation, particularly in biomedical imaging.

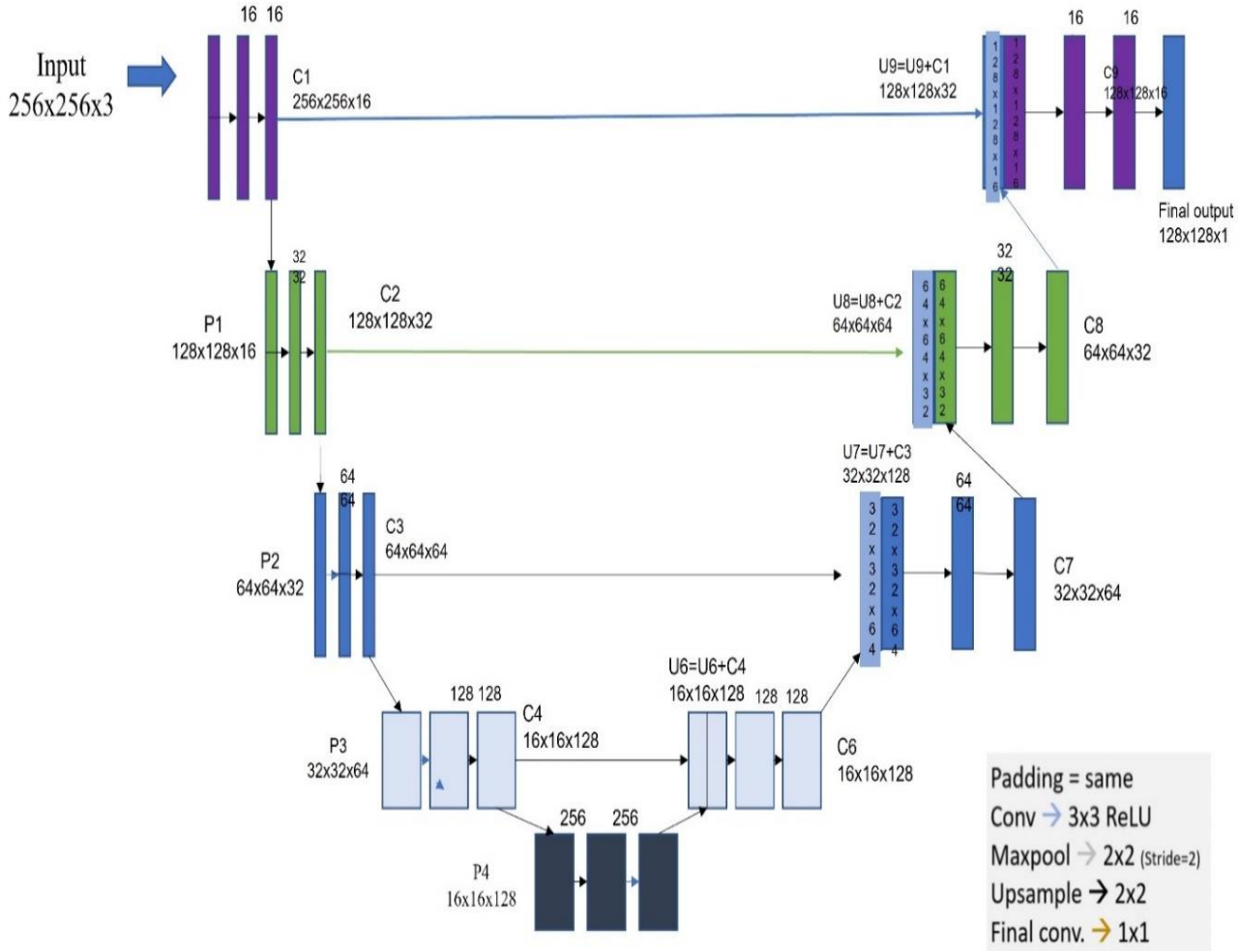


**Fig 4.3 Architecture of U-Net**

**F1-score**

The F1 score is an estimation of the accuracy for a model based on binary classification that is dependent on recall as well as precision. The value it holds can range from 0 to 1, having greater values signifying better performance.

The F1 scoring formula is as follows:

2 * (precision * recall) / (precision + recall) = F1 score

where accuracy is defined as the number of true positives divided by the total of true positives and false positives, and recall is defined as the number of true positives divided by the sum of true positives and false negatives.

where precision is defined as the number of true positives divided by the total of true positives and false positives and recall is defined as the numbers of true positives divided by the total amount of true positives and false negatives.

**Precision = True Positives / (True Positives + False Positives)**

**Recall = True Positives / (True Positives + False Negatives).**

The F1 score is frequently used to assess machine learning model performance in binary classification situations, where the aim is to predict one of the two possible results, which can be either true or false, positive or negative, or yes or no. It is especially beneficial when the data set is skewed, implying that a particular class is far more prevalent than the other. In such instances, accuracy alone may be deceptive, and the F1 score gives a more balanced assessment of the model's efficacy.

The F1 Score is the weighted mean of Recall and Precision. As a result, F1 Score takes false positives as well as false negatives into account. Although it is not as intuitive as accuracy, F1 is sometimes more handy than accuracy when there is an unequal

distribution of class. If the cost of false negatives as well as false positives is comparable, accuracy is the preferable option. If the cost of false negatives and also false positives differs, recall and accuracy must be evaluated.

Accuracy is defined as the proportion of correctly anticipated observations to total observations. It appears that if a model has great accuracy, it is the best. It is a key element that the levels of false positives as well as false negatives were almost identical for our symmetric datasets.

**Accuracy Score = (TP + TN)/ (TP + FN + TN + FP)**

**Confusion matrix**

A confusion matrix is the matrix that summarises a machine learning model's performance on a set of test data. It is frequently used to assess the effectiveness of classification algorithms, which are designed to predict a category label for each input occurrence. The matrix shows how many true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) the model produced on the test data.

### 4.4.5 WORKING METHODOLOGY

The Working methodology of Above-Ground Biomass model is described in fig 4.5. It contains four phases where an input image is selected. The dataset contains 4000 images and it is given as input to the training step and the image is processed. After the model is trained, a new image is given as input to the model. Once the image is found to be from our trained dataset, the model does the pre=processing step where we use Histogram to represent the image in a graph which takes the RGB values from the picture as the base for plotting the graph. The RGB value is taken into account to differentiate green from the image as it denotes the area where the trees are present. The mean of all the three colours is also calculated and displayed.

In segmentation process k-means classifier and binary classifier is used along with Region of Interest (ROI), where K-means classifier is used to classify the

different elements that is present in the image and ROI is used to take into account a particular area to classify and make the estimation. CNN model is used to determine the level of carbon trapped inside a single tree or the carbon present in that particular area.

In classification, we will extract the features of the image like Skewness, Contrast, Corelation, Energy, Mean, Standard Deviation, Entropy, RMS, Variance, Kurtosis to know more about the particular image. Using empirical formulae, the exact amount of carbon trapped inside the tree or that particular area is found by taking into account only the height and width of the tree by computing the dry weight. The half of dry weight will give the green weight and half of that will give the value of carbon.

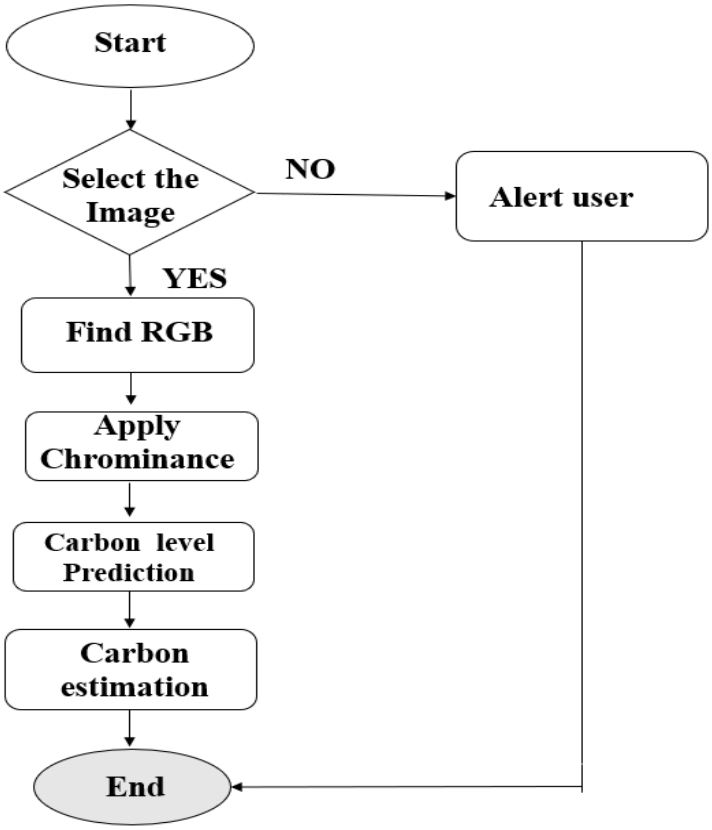


**Fig 4.4 Working methodology**

# CHAPTER 5

# SYSTEM TESTING

## UNIT TESTING

Unit testing is a method of software testing that involves each of the parts or units of the software system that are examined independently in order to verify that they work as intended. A function, method, class, or module can all be considered units. The goal of unit testing is to ensure that each component of the software system works as intended and satisfies the design criteria. Each unit is tested individually during unit testing, often by writing a test case that exercises the unit and comparing the results to expected behaviour. Unit tests are often automated and easy to execute, providing feedback on any regressions or new defects encountered during development. Unit testing assists in the early detection and isolation of flaws, minimising the cost of bug patches and enhancing the overall functionality of a software system. It also helps to ensure that each part of the software's architecture works properly, making the system simpler to maintain and modify over time. To guarantee that the software package as a whole is working effectively, unit testing usually works in conjunction with additional testing methodologies such as integration evaluation and system testing.

## INTEGRATION TESTING

Integration testing is a sort of software testing in which the interfaces as well as interactions among different elements or parts of a software system are examined. Integration testing is carried out to detect flaws that may develop when different components are combined and to ensure that they work together as intended. The components or modules are joined and evaluated as a group during integration testing to ensure that they interact and integrate appropriately with one another. Top-down testing, bottom-up testing, or a combination of the two can all be used for integration testing. System testing, which evaluates the complete software system to ensure that it fits its design criteria and needs, is frequently performed after integration testing.

## VALIDATION TESTING

Validation testing is a part of software testing that is used to ensure whether a software system or application is fit for purpose and meets the needs of its stakeholders. Validation testing ensures that a software system is appropriate for its intended purpose and meets the demands of its users. Validation testing is usually done at the end of the software development life cycle, when the software has been completed and is ready for release. It entails putting the software system through its paces in a real-world setting to ensure that it fulfils the demands of its users and serves its intended purpose. The ultimate purpose of validation testing is to guarantee that the software system provides the value and advantages expected. This sort of testing aids in the identification of any faults or issues that may affect the usability, performance, or security of the software system, allowing them to be rectified before to the system's release to the clients.

## VERIFICATION TESTING

Verification testing is a type of software testing that guarantees the design and technological requirements of a software system or app are satisfied. The purpose of testing for verification is to guarantee that the software system was appropriately designed and that it meets the requirements and design criteria. During the requirements collecting and design stages of the software's development life cycle, verification testing is frequently performed. It entails testing the software system to ensure that it satisfies its functional and technical requirements and is constructed in accordance with its design standards. Verification testing's ultimate purpose is to guarantee that the software system is developed correctly and fits its design parameters, lowering the chance of faults and issues later in the software development life cycle. Verification testing identifies any inconsistencies or holes in the design or requirements of the software system, allowing them to be corrected before the software system is constructed and delivered to the testing and validation phases.

## USER ACCEPTANCE TESTING

User acceptance testing (UAT) is a type of testing performed to verify whether or not a software system meets the requirements and demands of the user. The purpose of UAT is to ensure that the software platform is prepared for use by end users and meets their needs. After the software system has completed device, integration, and system testing, UAT is frequently conducted towards the end of the software's development cycle. End-users or customers who will use the software system in reality carry it out. The primary purpose of UAT is to ensure that the software platform meets the user's requirements and is ready for release. It facilitates in the identification of any flaws or problems that may affect the accessibility, efficiency, or functionality of the software system, that may then be corrected prior to the software system is provided to end users. UAT is an essential stage of testing in software development since it provides feedback from customers, which helps to ensure that the software system in question provides the value and benefits that were expected. The effective completion of UAT signifies that the software system is prepared for release to production and will meet the end-users' demands and expectations.

# CHAPTER 6

# RESULTS AND DISCUSSIONS

## Data Pre-processing

The Histogram is employed in the data the pre-processing stage in this study. The histogram is a graphical representation that shows the number of pixel in a picture at each distinct intensity value. The histogram graph is drawn to indicate the amount of green, blue and red amount in the image and the mean for the range of colours is also found. The mean is also computed to mathematically say how much the red, green and blue is present is the image.

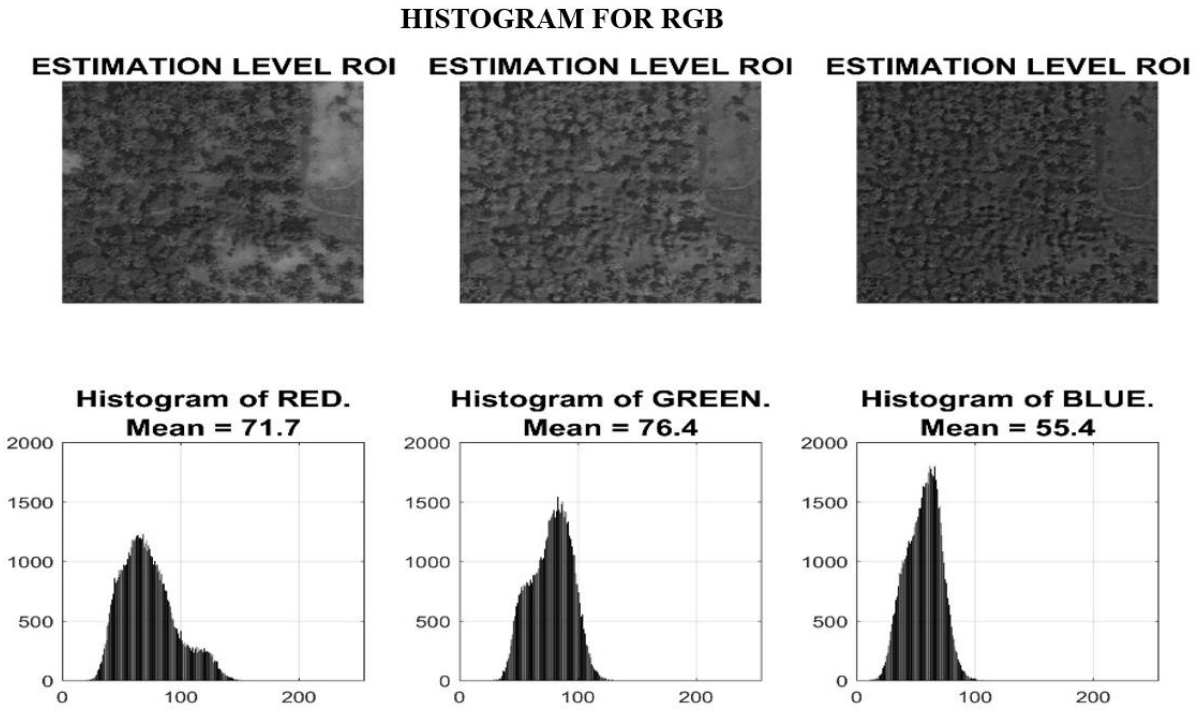


**Fig 6.1(Region of Interest)**

## Applying Chrominance

Hue is added to the image to separate and find the different shades of green. Finding the different shades of green is important as it denotes the area filled with trees. Blue and Red hue is added to differentiate the differentiate the shades of green. It will

also decrease the brightness of the image and makes it easier to separate out the shades of green in the image.

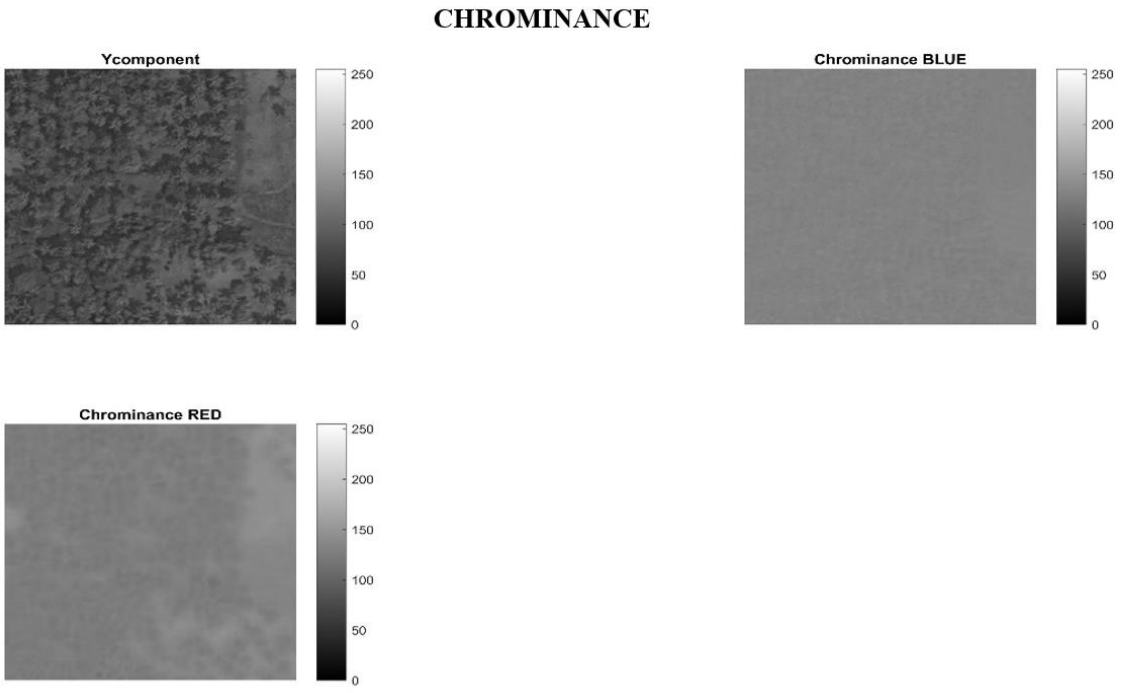


**Fig 6.2(Applying Chrominance)**

**Data Segmentation**

The segmentation process will provide accurate results by taking only the green value from the RGB mean value that is found in the pre-processing step. Region of interest (ROI) takes in a certain area of an image where further estimation can be done. In this work ROI is used to separate and analysis the areas where the green value is high denoting that it has trees present in that particular area. In the ROI segment, the area without the trees (areas other than green) is represented and in the segmentation image, the area with tress (areas that are green) is represented. Then the cluster of this both image is represented finally.

# SEGMENTATION

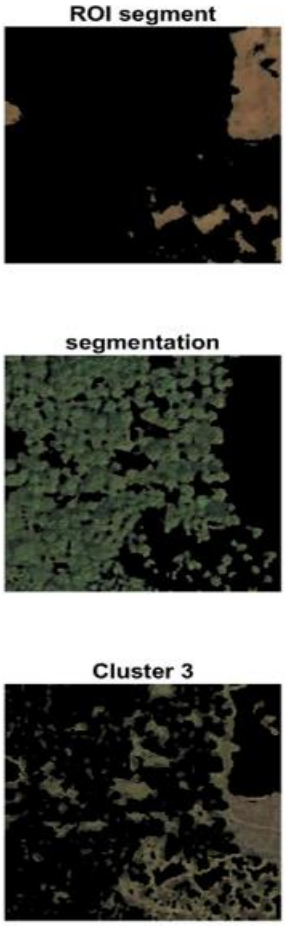


**Fig 6.3(Segmentation)**

**Success and Loss rate**

The dataset containing 4000 aerial and landscape images are given as input to the model for training and testing. The figure below (Fig 6.1) is plotted with Batch number which is nothing but the samples that are used to train and test the model in X axis and Success and loss rate is represented in the Y axis. The 4000 images are let into the model for testing and training and the model will check all the images and checks whether the image will be suitable for making further estimation. Upon checking all the images, the model found that among all the images, 300 will work best for the estimations that are needed for finding the carbon value.

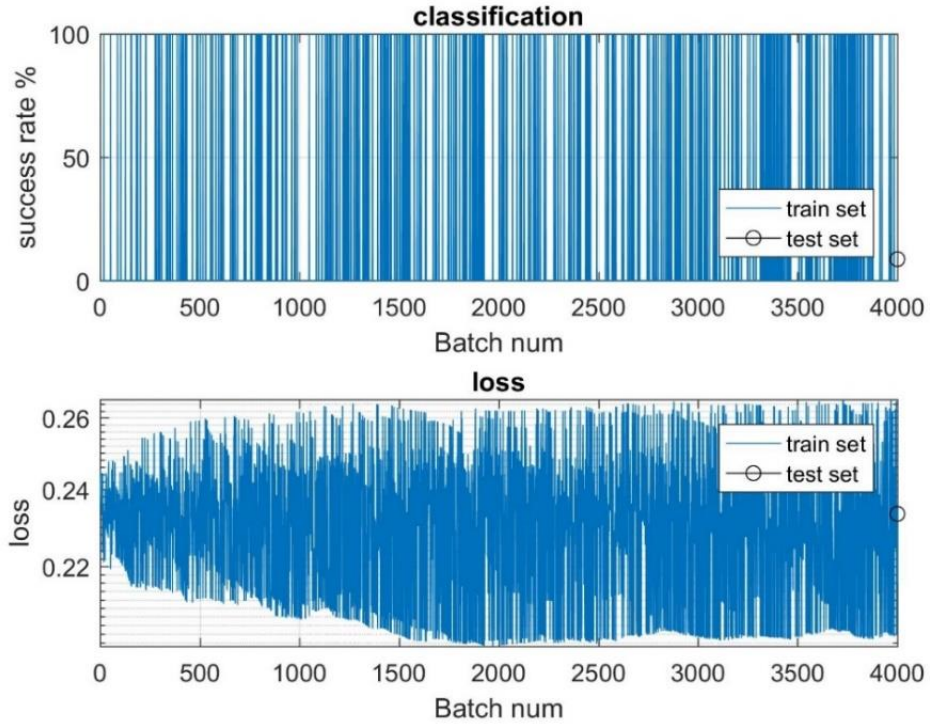


**Fig 6.4(Success and Loss Rate)**

## Confusion Matrix

The confusion Matrix allows us to find Recall and Precision, which, along with Accuracy, is used to measure the performance of Machine Learning and Deep Learning Models. The accuracy for the particular image that is chosen as the input to do the estimation in segmentation and classification step accounts to provide an accuracy of about 91.7%. This accuracy is only for this particular image and not the overall accuracy of the model that is built. The accuracy differs from image to image.

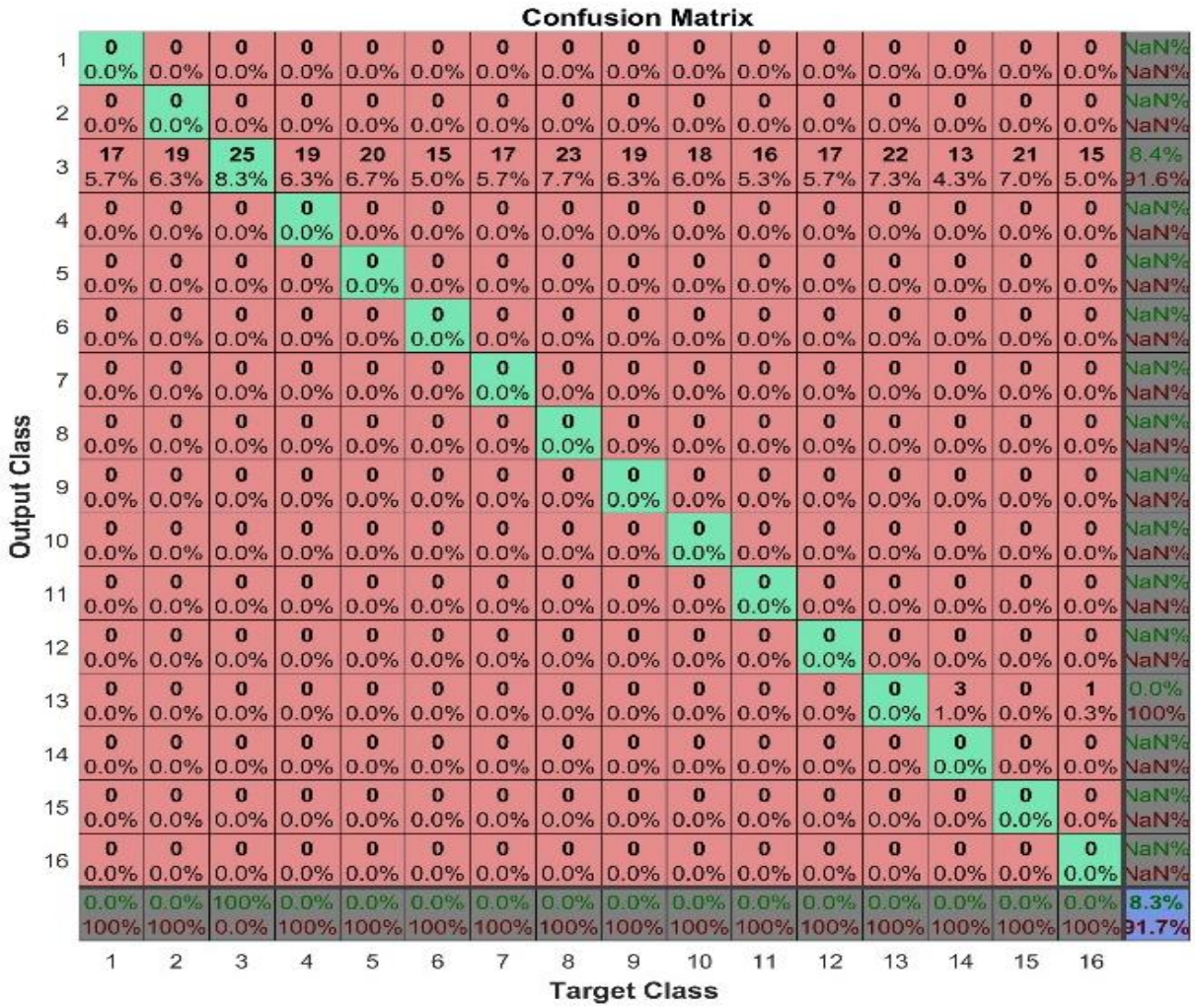


**Fig 6.5(Confusion matrix)**

In our model, we are training the MNIST images which is mainly in the form of matrix . These matrix consists of categorical vector of image labels. N by M matrix is used ,where N represents number of classes and M represents number of observations present in the images. The confusion matrix is mainly used to compare the results of true labels of trained data and predicted labels. The X-axis denotes target class which is the true test labels. The Y-axis denotes the predicted class i.e., output class. These two are represented in rows and columns. Rows are predicted class and columns are true test class. Each cell gives the information about the number of observations and percentage of total number of observations for that particular cell. The right end column species the precision values of the model and the bottom row species the recall values of the model.

The diagonal cells gives the information about correctly classified observations. The other cells are providing the details of incorrectly classified observations of the class. Even though the underlying classes doesn't provide any observations, it displays the results with zero percentage. The output and target class must have same number of elements. The bottom right cell gives the percentage of overall accuracy by comparing the test values with the predicted values. We obtained over 91.7% accuracy in our model.

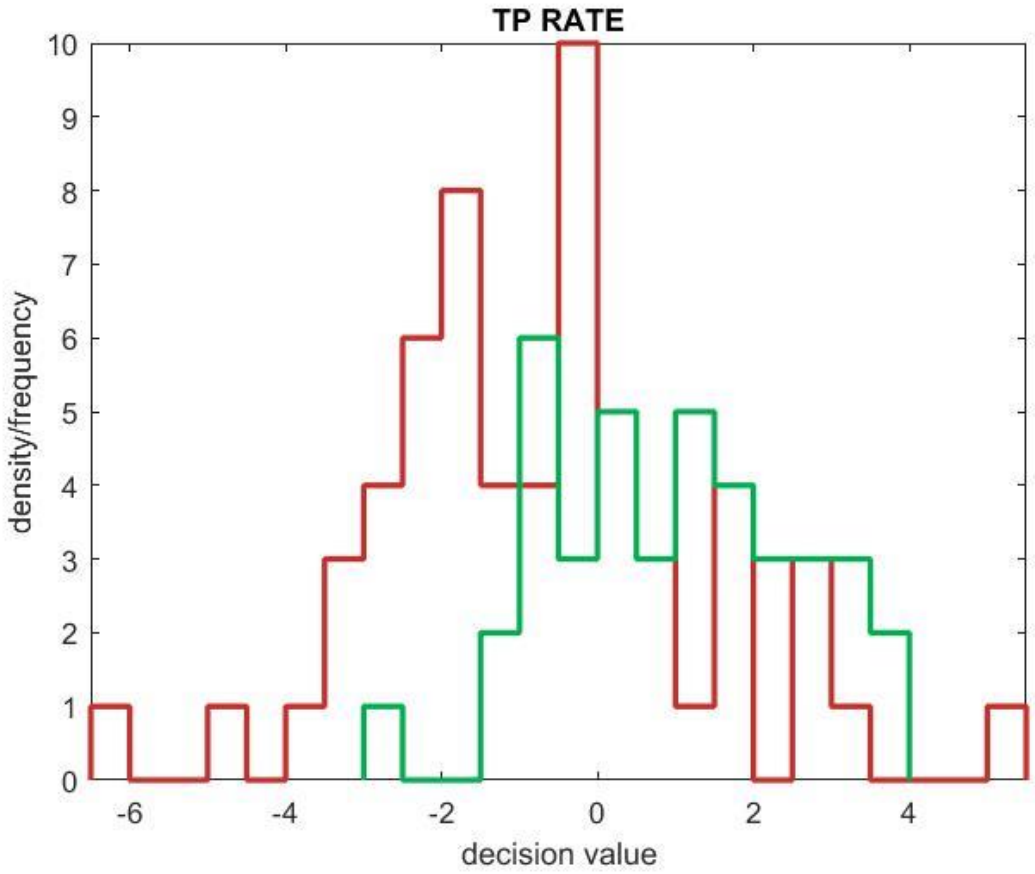


**Fig 6.6(TP Rate)**

The above graph represents the TP rate (True Positive) of the model. Where X axis denotes the predicting values which is positive or negative. While, Y axis denotes the actual values which gives whether the predicting values are true or false. By combining actual and predicted values we get to know about TP(true Positive) ,FP(False Positive) ,TN(True negative) ,FN(False Negative). This graph compares the above mentioned

four fields for measuring the performance of classification models .The Green colour line in the positive area gives TP ,where in negative area gives TN .Likewise, The Red Colour line in the positive area gives FP, where in negative area gives about FN.

**Performance Measure**

The performance measure of the classification model is calculated by ROC curve i.e., receiver operating characteristic curve. The performance of the model is done based on the two parameters ,that is True Positive Rate and False Positive Rate .The graph gives the Performance of model at every thresholds with above mentioned parameters. For Precision, the threshold must set to higher values for the effective model performance. A good classification model will have an ROC curve that is close to the top left corner of the plot, which represents high TPR and low FPR, indicating that the model is able to accurately classify positive and negative samples.

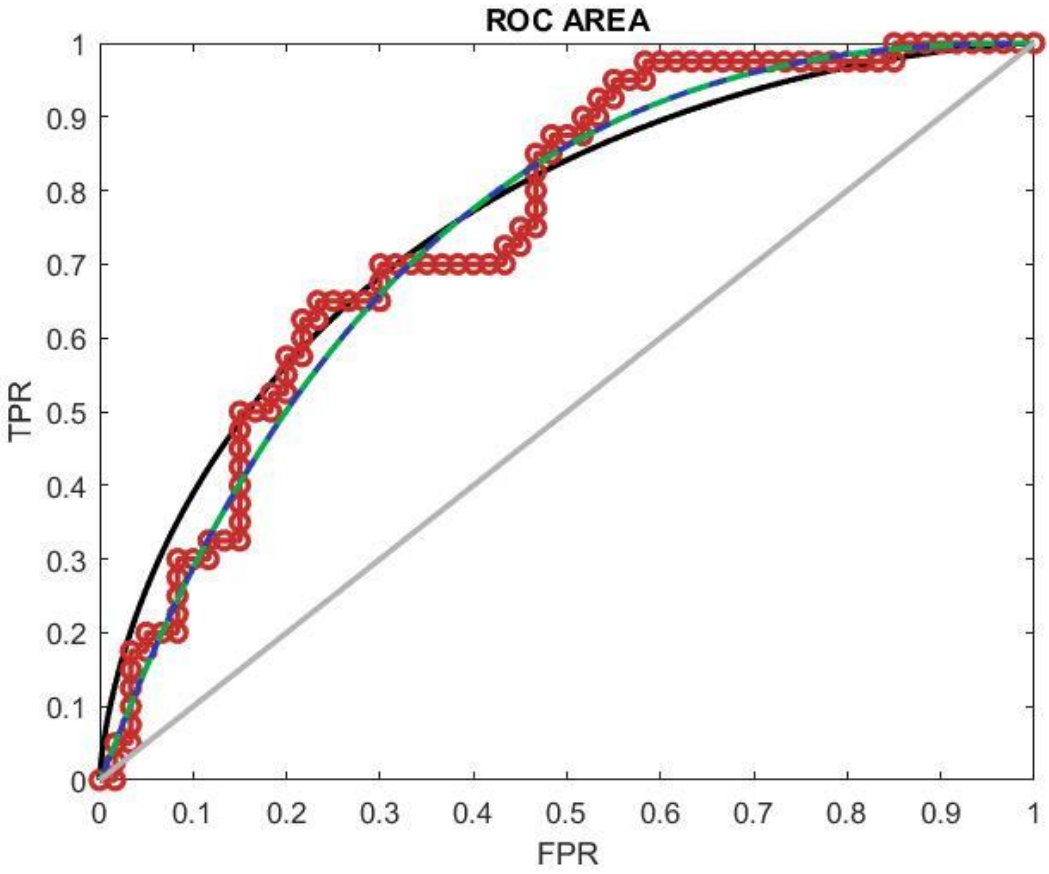


**Fig 6.7(ROC curve)**

**Accuracy**

The False positive rate provides the fraction of incorrect predictions in positive class. The FP rate is found with the help of Precision and Recall. The True Positive Rate is taken for the value of recall and the Positive Predictive Value is taken for the Precision. It denotes the correctly classified positive samples divided by the number of positive samples. For accuracy ,both the positive and negative outcomes should be taken into account for the effective model because, if the model's False positive increases even though it has high precision still it consider as non effective model. They are plotted as the graph. In our graph, both precision and recall are high. So, it denotes that our model is effective at different thresholds.

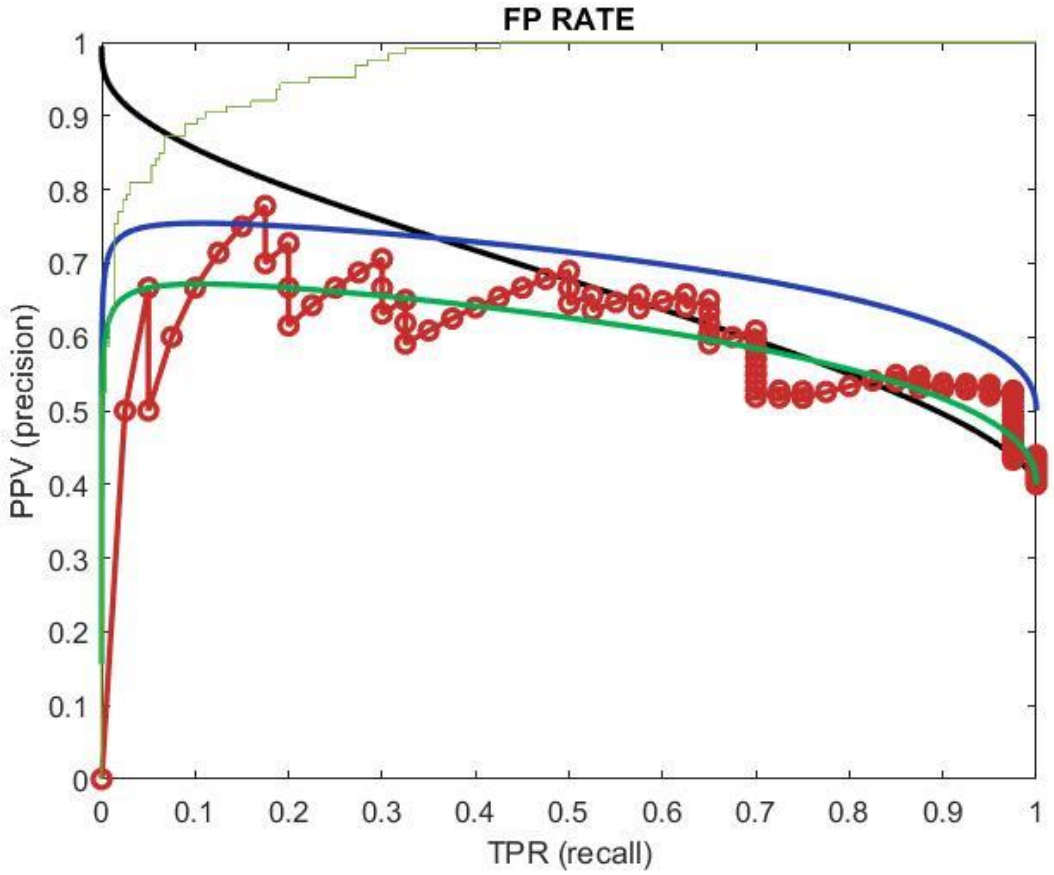


**Fig 6.8(FP Rate)**

**Precision and Recall**

Recall and precision are two commonly used evaluation metrics in classification tasks in machine learning. The recall measures the fraction of positive examples that are correctly identified by the classifier, while the precision measures the fraction of predicted positive examples that are actually positive. A perfect classifier has both recall and precision equal to 1. This graph is used to identify the pattern between the predicted values and actual values. This graph helps to plot the relation between the false positive and true positive rate. This is very helpful to develop an efficient classification system.

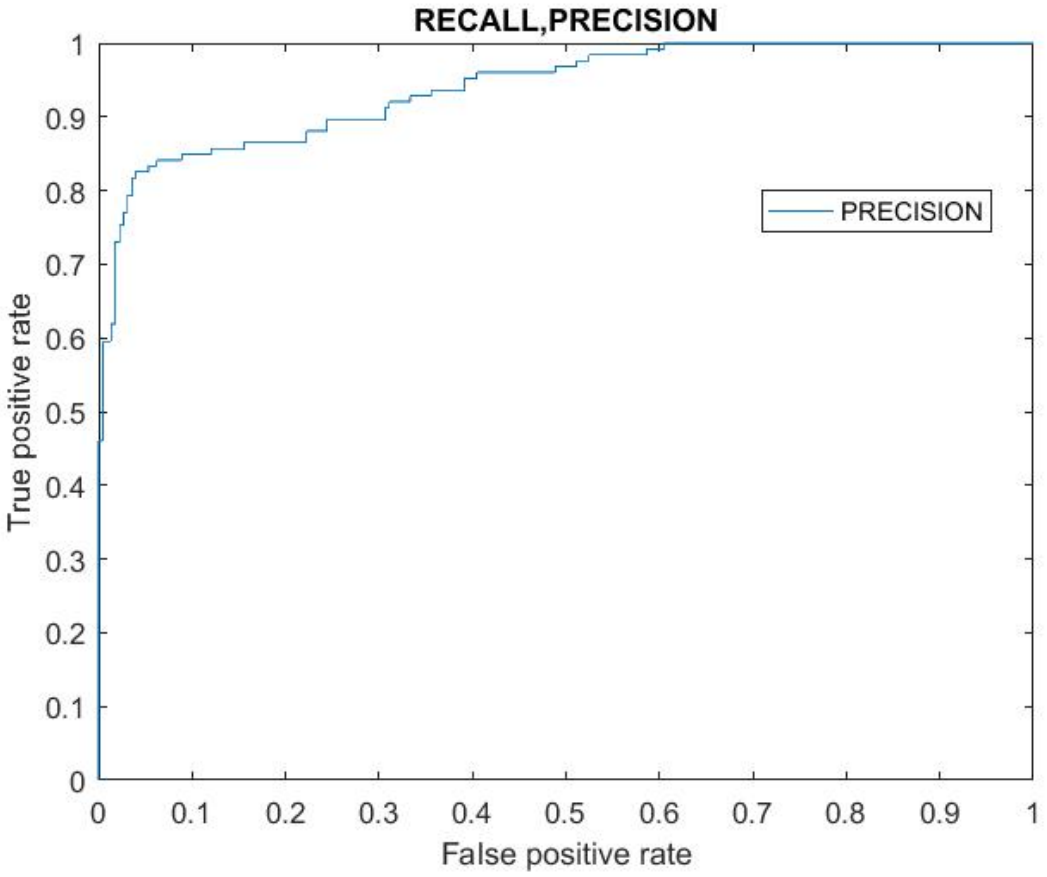


**Fig 6.9(Recall and Precision)**

# CHAPTER 7

# CONCLUSION AND FUTURE ENHANCEMENT

## 7.1 CONCLUSION

The rising global mean air temperature and other climatic changes are being driven by a rise in atmospheric greenhouse gas concentrations, the most significant of which is carbon dioxide (CO2). The exchange of CO2 between the atmosphere and the biosphere moderates the atmospheric concentration; the net exchange is the consequence of photosynthesis that takes in and respiration that releases CO2. Satellite sensor data may be used to estimate CO2 exchange on a regional or global scale. With the help of the data that we obtained from Deep Globe dataset, we can estimate the amount of carbon trapped inside the trees in a particular area which will help in reducing the amount of carbon in the atmosphere and maintain a balanced proportion of carbon in the atmosphere. Region of interest along with Convolutional Neural Network (CNN) is used to implement the model. The Region of Interest is used to take in a particular area of the image to do further calculations and CNN is used to extract the feature of the aerial image to tell the level of carbon present in that area. The estimation is mainly based on RGB values of the image which will give a better way to differentiate the different shades of green and find the area which contains the most number of trees. As tress play an important role in carbon cycle, trees are the main factor in this work. ROI along with GLCM with the help of CNN model is used to estimate both the level of carbon and also the exact amount of carbon present in that particular area by using aerial image.

By implementing this model, the overall accuracy accounted to 92%. The accuracy changes for each and every image and that depends on the input image that is fed into the model. This work also tells the level of carbon present in that particular area and also gives the features of the image for better understanding.

## 7.2 FUTURE ENHANCEMENT

In future work, profound plan of estimating the carbon with any sort of image and not just an aerial image will be proposed, as aerial image can be accessed only with permission to access the satellite data from the original source. By making the model this way, more people can work and know about the effects of carbon and encourages people to work for the environment and also it is easy to find the dataset as it will contain images taken from a normal camera or drone image.

The future scope of the proposed work is to include an alarm which will prompt the user if the carbon rate exceeds the pre-determined safe level of carbon that should be present in the environment. This will surely benefit the environmentalists, as it alerts them to take actions to balance the carbon level. This will help the environmentalist to make changes to the environment in a long run.

# CHAPTER 8
# PSEUDOCODE

Input

Forest Aerial Image

Procedure

Load the input images

Detect the range of RGB in the input image using histogram

Apply chrominance of red and blue in the image

Segment the areas in the image using ROI

Apply k-means cluster for combine green area

Split the dataset used for learning(training and testing)

Setting for CNN parameters

Start the parameters

Test the model

Sort the classes into groups and create a confusion matrix.

Extract CNN graph for RGB in the image and convert into grayscale image

Extract green areas height and width

Apply it in the empirical formula for Estimating Carbon trapped inside the Tree

Output:Carbon level in the particular area is estimated.

# CHAPTER 10

# APPENDIX

## APPENDIX 1 SOURCE CODE

```
clc;    % Clear the command window.
workspace;  % Make sure the workspace panel is showing.
format longg;
format compact;
fontSize = 15;
[baseFileName, folder] = uigetfile('*.*', 'Specify an image file');
fullFileName = fullfile(folder, baseFileName);
if ~exist(fullFileName, 'file')
  % Didn't find it there.  Check the search path for it.
  fullFileName = baseFileName; % No path this time.
  if ~exist(fullFileName, 'file')
    % Still didn't find it.  Alert user.
    errorMessage = sprintf('Error: %s does not exist.', fullFileName);
    uiwait(warndlg(errorMessage));
    return;
  end
end
[Bio_Mass_Estimation colorMap] = imread(fullFileName);
[rows columns numberOfColorBands] = size(Bio_Mass_Estimation);
% If it's an indexed image (such as Kids),  turn it into an rgbImage;
```

```
if numberOfColorBands == 1

  Bio_Mass_Estimation = ind2rgb(Bio_Mass_Estimation, colorMap); % Will be in the 0-1 range.

  Bio_Mass_Estimation = uint8(255*Bio_Mass_Estimation); % Convert to the 0-255 range.

end

% Display the original color image full screen

imshow(Bio_Mass_Estimation);

title('Madras eye', 'FontSize', fontSize);

% Enlarge figure to full screen.

set(gcf, 'units','normalized','outerposition', [0 0 1 1]);

% if strcmpi(button, 'Cancel')

%   return;

% end

redChannel = Bio_Mass_Estimation(:, :, 1);

greenChannel = Bio_Mass_Estimation(:, :, 2);

blueChannel = Bio_Mass_Estimation(:, :, 3);

% Display the color channels.

subplot(2, 4, 2);

imshow(redChannel);

title('ESTIMATION LEVEL ROI', 'FontSize', fontSize);

subplot(2, 4, 3);

imshow(greenChannel);

title('ESTIMATION LEVEL ROI', 'FontSize', fontSize);
```

```
subplot(2, 4, 4);
imshow(blueChannel);
title('ESTIMATION LEVEL ROI', 'FontSize', fontSize);
imgID = 2;
% Get the means of each color channel
meanR = mean2(redChannel);
meanG = mean2(greenChannel);
meanB = mean2(blueChannel);
% Let's compute and display the histograms.
[pixelCount grayLevels] = imhist(redChannel);
subplot(2, 4, 6);
bar(pixelCount);
grid on;
caption = sprintf('Histogram of RED.\nMean = %.1f', meanR);
title(caption, 'FontSize', fontSize);
xlim([0 grayLevels(end)]); % Scale x axis manually.
% Let's compute and display the histograms.
[pixelCount grayLevels] = imhist(greenChannel);
subplot(2, 4, 7);
bar(pixelCount);
grid on;
caption = sprintf('Histogram of GREEN.\nMean = %.1f', meanG);
title(caption, 'FontSize', fontSize);
```

```
xlim([0 grayLevels(end)]); % Scale x axis manually.
% Let's compute and display the histograms.
[pixelCount grayLevels] = imhist(blueChannel);
subplot(2, 4, 8);
bar(pixelCount);
grid on;
caption = sprintf('Histogram of BLUE.\nMean = %.1f', meanB);
title(caption, 'FontSize', fontSize);
xlim([0 grayLevels(end)]); % Scale x axis manually.
% specify the desired mean.
desiredMean = mean([meanR, meanG, meanB])
% message = sprintf('ALL OVER ESTIMATION LEVEL ',...
%   meanR, meanG, meanB, desiredMean);
% uiwait(helpdlg(message));
% Linearly scale the image in the cropped ROI.
correctionFactorR = desiredMean / meanR;
correctionFactorG = desiredMean / meanG;
correctionFactorB = desiredMean / meanB;
redChannel = uint8(single(redChannel) * correctionFactorR);
greenChannel = uint8(single(greenChannel) * correctionFactorG);
blueChannel = uint8(single(blueChannel) * correctionFactorB);
% Recombine into an RGB image
% Recombine separate color channels into a single, true color RGB image.
```

```
correctedRgbImage = cat(3, redChannel, greenChannel, blueChannel);
figure;
% Display the original color image.
subplot(2, 4, 5);
imshow(correctedRgbImage);
title('BIO MASS ESTIMATION ROI', 'FontSize', fontSize);
% Enlarge figure to full screen.
set(gcf, 'units','normalized','outerposition',[0 0 1 1]);
% Display the color channels.
subplot(2, 4, 2);
imshow(redChannel);
title('BIO MASS ESTIMATION ROI', 'FontSize', fontSize);
subplot(2, 4, 3);
imshow(greenChannel);
title('BIO MASS ESTIMATION ROI', 'FontSize', fontSize);
subplot(2, 4, 4);
imshow(blueChannel);
title('BIO MASS ESTIMATION ROI', 'FontSize', fontSize);
% Let's compute and display the histograms of the corrected image.
[pixelCount grayLevels] = imhist(redChannel);
subplot(2, 4, 6);
bar(pixelCount);
grid on;
```

```
caption = sprintf('Histogram of RED.\nMean = %.1f', meanR);
title(caption, 'FontSize', fontSize);
xlim([0 grayLevels(end)]); % Scale x axis manually.
% Let's compute and display the histograms.
[pixelCount grayLevels] = imhist(greenChannel);
subplot(2, 4, 7);
bar(pixelCount);
grid on;
caption = sprintf('Histogram of GREEN.\nMean  = %.1f', meanG);
title(caption, 'FontSize', fontSize);
xlim([0 grayLevels(end)]); % Scale x axis manually.
% Let's compute and display the histograms.
[pixelCount grayLevels] = imhist(blueChannel);
subplot(2, 4, 8);
bar(pixelCount);
grid on;
caption = sprintf('Histogram of BLUE.\nMean  = %.1f', meanB);
title(caption, 'FontSize', fontSize);
xlim([0 grayLevels(end)]); % Scale x axis manually.
% Get the means of the corrected ROI for each color channel.
meanR = mean2(redChannel);
meanG = mean2(greenChannel);
meanB = mean2(blueChannel);
```

```
correctedMean = mean([meanR, meanG, meanB])
fontSize = 10;
rchannel =  Bio_Mass_Estimation( :,: ,1);
gchannel = Bio_Mass_Estimation( :,: ,2);
bchannel = Bio_Mass_Estimation( :,: ,3);
   R = Bio_Mass_Estimation;
R(:,:,2:3) = 0;
G = Bio_Mass_Estimation;
G(:,:,[1 3]) = 0;
B = Bio_Mass_Estimation;
B(:,:,1:2) = 0;
subplot(2,2,2)
image(R);
pause(1);colorbar;
subplot(2,2,3)
image(G);
pause(1);colorbar;
subplot(2,2,4)
image(B);
pause(1);colorbar;
%%
%%extracting chrominace , chrominace blue and red
```

```
Ydash = 16+(0.2567890625  * rchannel)+ (0.50412890625 * gchannel) +( 0.09790625 * bchannel);

Cb= 128+(-0.14822265625 * rchannel)-(0.2909921875 * gchannel) + (0.43921484375* bchannel);

Cr = 128+(0.43921484375  * rchannel)- (0.3677890625 * gchannel) -( 0.07142578125 * bchannel);

%% JPEg format

%Ydash= 0+( 0.299* rchannel)+ (0.587 *gchannel) +(0.114 * bchannel);

%Cb= 128+(-0.168736 * rchannel)-(0.331264 * gchannel) + (0.5* bchannel);

%Cr = 128+(0.5  * rchannel)- (0.418688 * gchannel) -( 0.081312 * bchannel);

%%

 Ymean=  (mean(mean(Ydash)));

 Cbmean= (mean(mean(Cb)));

 Crmean= (mean(mean(Cr)));

 Crstd= std2(Cr);

figure

set(gcf, 'units','normalized','outerposition',[0 0 1 1]);

subplot(2,2,1)

imshow(Ydash); title ('Ycomponent');

pause(1);colorbar;

subplot(2,2,2)

imshow(Cb); title ('Chrominance BLUE');

pause(1);colorbar;

subplot(2,2,3)
```

```
imshow(Cr); title ('Chrominance RED');

pause(1);colorbar;

%% segmentation

disp('Segmentation.')

%%

cform = makecform('srgb2lab');

% Apply the colorform

lab_he = applycform(Bio_Mass_Estimation,cform);

% Classify the colors in a*b* colorspace using K means clustering.

% Since the image has 3 colors create 3 clusters.

% Measure the distance using Euclidean Distance Metric.

ab = double(lab_he(:,:,2:3));

nrows = size(ab,1);

ncols = size(ab,2);

ab = reshape(ab,nrows*ncols,2);

nColors = 3;

[cluster_idx cluster_center] = kmeans(ab,nColors,'distance','sqEuclidean', ...

                                      'Replicates',3);

%[cluster_idx cluster_center] =
kmeans(ab,nColors,'distance','sqEuclidean','Replicates',3);

% Label every pixel in tha image using results from K means

pixel_labels = reshape(cluster_idx,nrows,ncols);

%figure,imshow(pixel_labels,[]), title('Image Labeled by Cluster Index');
```

```
% Create a blank cell array to store the results of clustering
segmented_images = cell(1,3);
% Create RGB label using pixel_labels
rgb_label = repmat(pixel_labels,[1,1,3]);
for k = 1:nColors
    colors = Bio_Mass_Estimation;
    colors(rgb_label ~= k) = 0;
    segmented_images{k} = colors;
end
figure, subplot(3,1,1);imshow(segmented_images{1});title('ROI segment');
subplot(3,1,2);imshow(segmented_images{2});title('segmentation');
subplot(3,1,3);imshow(segmented_images{3});title('Cluster 3');
set(gcf, 'Position', get(0,'Screensize'));
signal1 = feature_ext(Bio_Mass_Estimation);
[cA1,cH1,cV1,cD1] = dwt2(signal1,'db4');
[cA2,cH2,cV2,cD2] = dwt2(cA1,'db4');
[cA3,cH3,cV3,cD3] = dwt2(cA2,'db4');
DWT_feat = [cA3,cH3,cV3,cD3];
G = pca(DWT_feat);
g = graycomatrix(G);
stats = graycoprops(g,'Contrast Correlation Energy Homogeneity');
Contrast = stats.Contrast;
Correlation = stats.Correlation;
```

```
Energy = stats.Energy;

Homogeneity = stats.Homogeneity;

Mean = mean2(G);

Standard_Deviation = std2(G);

Entropy = entropy(G);

RMS = mean2(rms(G));

%Skewness = skewness(img)

Variance = mean2(var(double(G)));

a = sum(double(G(:)));

Smoothness = 1-(1/(1+a));

Kurtosis = kurtosis(double(G(:)));

Skewness = skewness(double(G(:)));

feat = [Contrast,Correlation,Energy,Homogeneity, Mean, Standard_Deviation, Entropy,
RMS, Variance, Smoothness, Kurtosis, Skewness];

disp('------------------------------------------------------------');

disp('Contrast = ');

disp(Contrast);

disp('Correlation = ');

disp(Correlation);

disp('Energy = ');

disp(Energy);

disp('Mean = ');

disp(Mean);
```

```
disp('Standard_Deviation = ');
disp(Standard_Deviation);
disp('Entropy = ');
disp('RMS = ');
disp(Entropy);
disp(RMS);
disp('Variance = ');
disp(Variance);
disp('Kurtosis = ');
disp(Kurtosis);
disp('Skewness = ');
disp(Skewness);
load Trainset.mat
 xdata = meas;
 group = label;
acc=accuracy_image(feat);
disp(acc);
addpath('../../Training' , '../../mdCNN' , '../../utilCode' );
numTest=300;numTrain=100;
x=randi(16,1,numTrain+numTest)-1;
xBin=[mod(x,2) ;mod(floor(x/2),2) ;mod(floor(x/4),2) ;mod(floor(x/8),2)];
% 'hide' the 4 bits inside a larger vector padded with random bits and fixed bits
```

```
samples = [repmat((1:10)',1,size(xBin,2)) ; xBin ;
rand(10,size(xBin,2))/2*mean(xBin(:))];

dataset=[];

for idx=1:size(samples,2)

   if (idx>numTrain)

      dataset.I_test{idx-numTrain} = samples(:,idx-numTrain);

      dataset.labels_test(idx-numTrain)=x(idx-numTrain);

   else

   dataset.I{idx} = samples(:,idx);

   dataset.labels(idx)=x(idx);

   end

end

net = CreateNet('../../Configs/1d.conf');  % small 1d fully connected net,will converge
faster

net   =  Train(dataset,net, 100);

checkNetwork(net,Inf,dataset,1);

 hsv=rgb2hsv(Bio_Mass_Estimation);

   hueImage = hsv(:,:, 1);

   figure, imshow(hueImage);

   meann = mean2(feat);

   meann= meann*100;

   y = round(meann);

   disp(y);

   run('predict.m');
```

```
pause(1);
% result = cnn_classifier(feat,meas,label);
% helpdlg(result);
load('Accuracy_Data.mat')
Accuracy_Percent= zeros(200,1);
for i = 1:800
data = Train_Feat;
groups = ismember(Train_Label,1);
% groups = ismember(Train_Label,0);
[train,feat] = crossvalind('HoldOut',groups);
cp = classperf(groups);
 classperf(cp,feat);
Accuracy = cp.CorrectRate*2;
Accuracy_Percent(i) = Accuracy.*100;
end
Max_Accuracy = max(Accuracy_Percent);
sprintf('Accuracy of cnn with 800 iterations is: %g%%',Max_Accuracy)
% warning('on','all');
    fontSize = 10;
    % Compute and plot the red histogram.
    hR = figure
    [countsR, grayLevelsR] = imhist(rchannel);
    maxGLValueR = find(countsR > 0, 1, 'last');
```

```
maxCountR = max(countsR);
bar(countsR, 'r');
grid on;
xlabel('GRAY VALUE');
ylabel('PIXEL');
title('CNN GRAPH', 'FontSize', fontSize);
% Compute and plot the green histogram.
hG = figure
[countsG, grayLevelsG] = imhist(gchannel);
maxGLValueG = find(countsG > 0, 1, 'last');
maxCountG = max(countsG);
bar(countsG, 'g', 'BarWidth', 0.95);
grid on;
xlabel('GRAY VALUE');
ylabel('PIXEL');
title('CNN GRAPH', 'FontSize', fontSize);
% Compute and plot the blue histogram.
hB = figure
[countsB, grayLevelsB] = imhist(bchannel);
maxGLValueB = find(countsB > 0, 1, 'last');
maxCountB = max(countsB);
bar(countsB, 'b');
grid on;
```

```
    xlabel('GRAY VALUE');
    ylabel('PIXEL');
    title('CNN GRAPH', 'FontSize', fontSize);
    % Set all axes to be the same width and height.
    % This makes it easier to compare them.
    maxGL = max([maxGLValueR,  maxGLValueG, maxGLValueB]);
% if eightBit
%         maxGL = 255;
% end
    maxCount = max([maxCountR,  maxCountG, maxCountB]);
% axis([hR hG hB], [0 maxGL 0 maxCount]);
    % Plot all 3 histograms in one plot.
    figure
    plot(grayLevelsR, countsR, 'r', 'LineWidth', 2);
    grid on;
    xlabel('Gray Levels');
    ylabel('Pixel Count');
    hold on;
    plot(grayLevelsG, countsG, 'g', 'LineWidth', 2);
    plot(grayLevelsB, countsB, 'b', 'LineWidth', 2);
    title('ALL OVER REGION', 'FontSize', fontSize);
   gray_img=rgb2gray(Bio_Mass_Estimation);
[height_val,width_val,w_h_img]=Estimation_carbon_level( gray_img );
```

```
figure,imshow(w_h_img),title('BIO MASS CARBON LEVEL Image');impixelinfo;

% msgbox(strcat('The Estimation of carbon level
are=',num2str(width_val),':',num2str(height_val)));

if (num2str(width_val)>28)

    GW= 0.0577*((width_val)*2)*(height_val);

    DW= GW/2;

    Carbon= DW/2;

    msgbox(strcat('The Estimation of carbon level =',num2str(Carbon)));

else (num2str(width_val)<28)

    GW= 0.0346*((width_val)*2)*(height_val);

    DW= GW/2;

    Carbon= DW/2;

    msgbox(strcat('The Estimation of carbon level =',num2str(Carbon)));

end
```

**Carbon Estimation Level**

```
function [ height,width,out_img ] = Estimation_carbon_level ( gray_img

level = graythresh(gray_img);

seg_img = im2bw(gray_img,level);

[L num] = bwlabel(seg_img);

   for j = 1:num

      if length(find(L == j)) < 20

         L(L==j) = 0;

      end
```

```
    end
    L1 = imdilate(L,ones(3));
    [labe num] = bwlabel(L1);
    %%impixelinfo
    %hold on
    box = regionprops(labe,'BoundingBox');
    m_bounv=0;
    m_bounl=0;
    for j = 1:num
        r_size=  round(box(j).BoundingBox);
        if(m_bounv<(r_size(4)*r_size(3))  )
            m_bounv=r_size(4)*r_size(3);
            m_bounl=j;
        end
    end
    %hold off
 if(m_bounl~=0)
r_size=  round(box(m_bounl).BoundingBox);
 r_img=(seg_img(r_size(2):r_size(2)+r_size(4)-1,r_size(1):r_size(1)+r_size(3)-1,:));
 else
   r_img=zeros(32,32);
 end
seg_img=~r_img;
```

```
w_h_img=seg_img;
%Program for height measurement
pos=0;
height_val=0;
center_pos=1;
for i=1:size(seg_img,1)
   for j=1:size(seg_img,2)
      if(seg_img(i,j)==0 && pos==0)
         pos=pos+1;
         center_pos=j;
      elseif(seg_img(i,center_pos)==1 && pos==1)
       break;
      end
     if(j==center_pos && pos==1)
        w_h_img(i,j)=1;
        height_val=height_val+1;
     end
   end
end
% figure, imshow(w_h_img);
%Program for width measurement
col_pos=size(seg_img,2);
row_pos=1;
```

```
for i=1:size(seg_img,1)
    for j=1:size(seg_img,2)
        if(seg_img(i,j)==0 && col_pos> j )
            col_pos=j;
            row_pos=i;
            break;
        end
width_val=0;
%for i=1:size(seg_img,1)
    for j=col_pos:size(seg_img,2)
        if(seg_img(row_pos,j)==1 )
         %break;
        else
          w_h_img(row_pos,j)=1;
          width_val=width_val+1;
        end
%end
width= width_val;
out_img=w_h_img;
height=height_val;
end
```

## APPENDIX 2 Output Screenshots

**Fig A2.1(Input Image)**

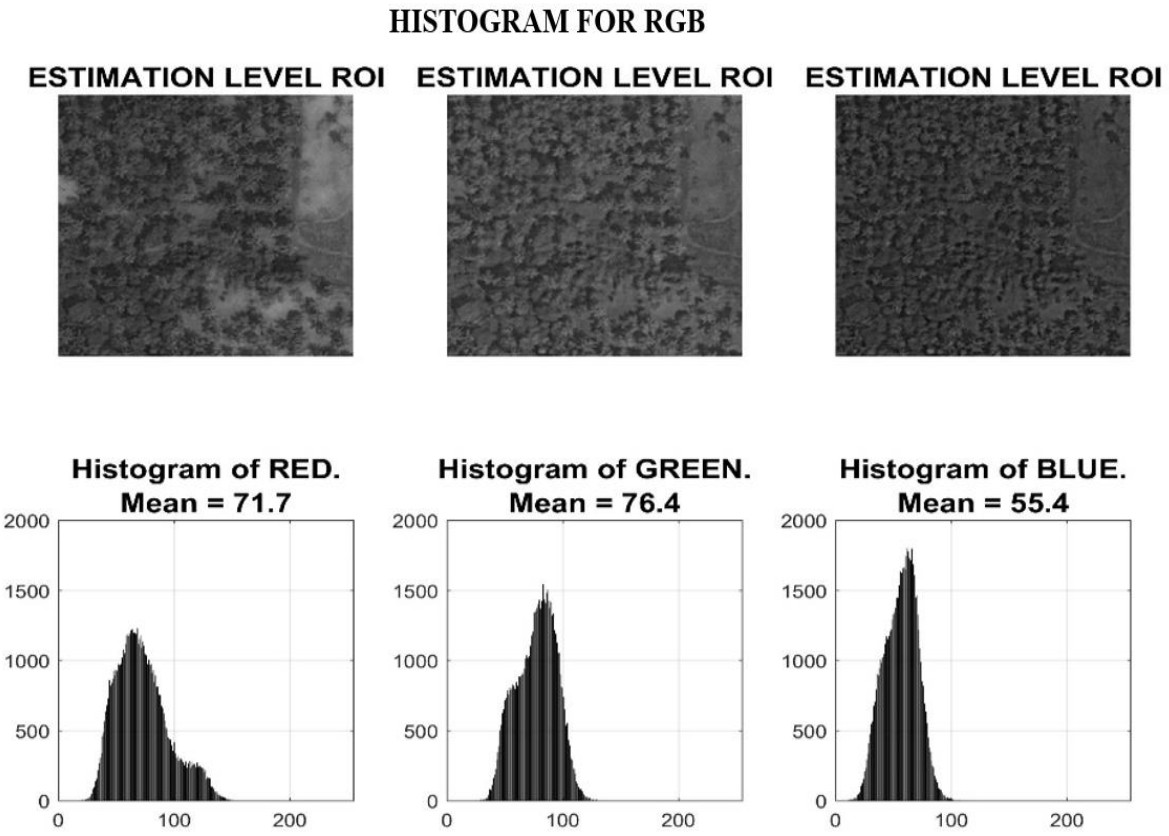


**Fig A2.2(Region of Interest)**

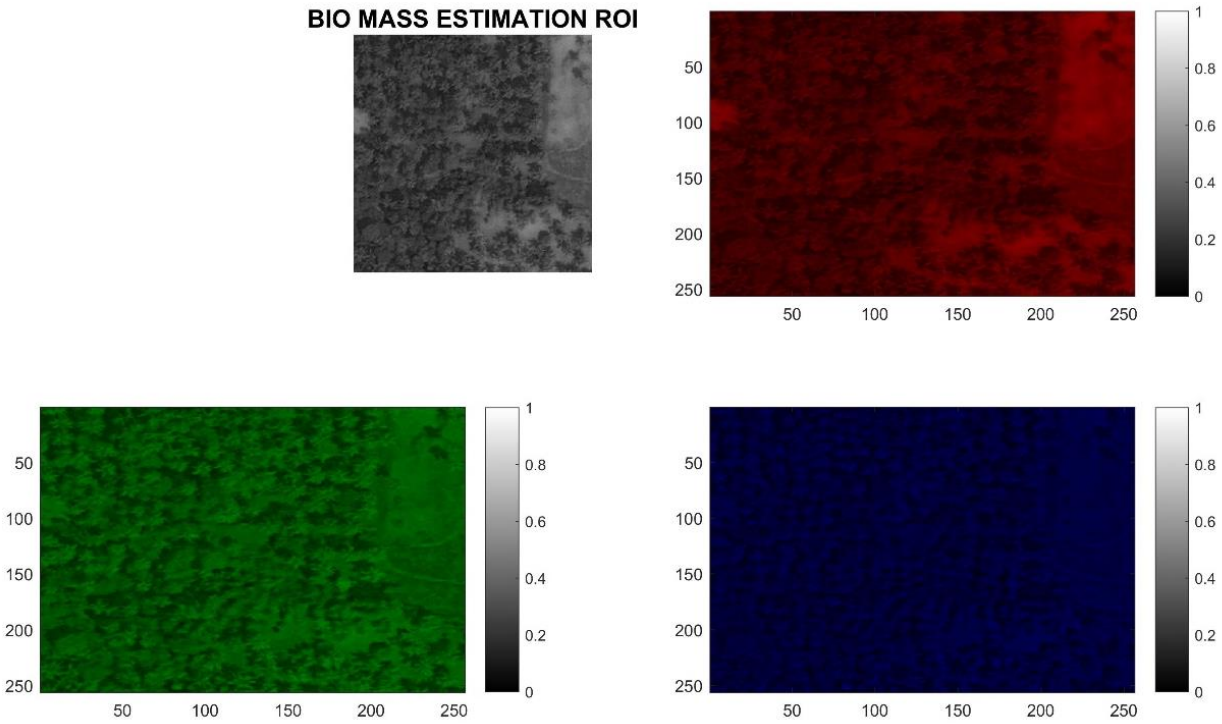


**Fig A2.3(Applying Chrominance)**

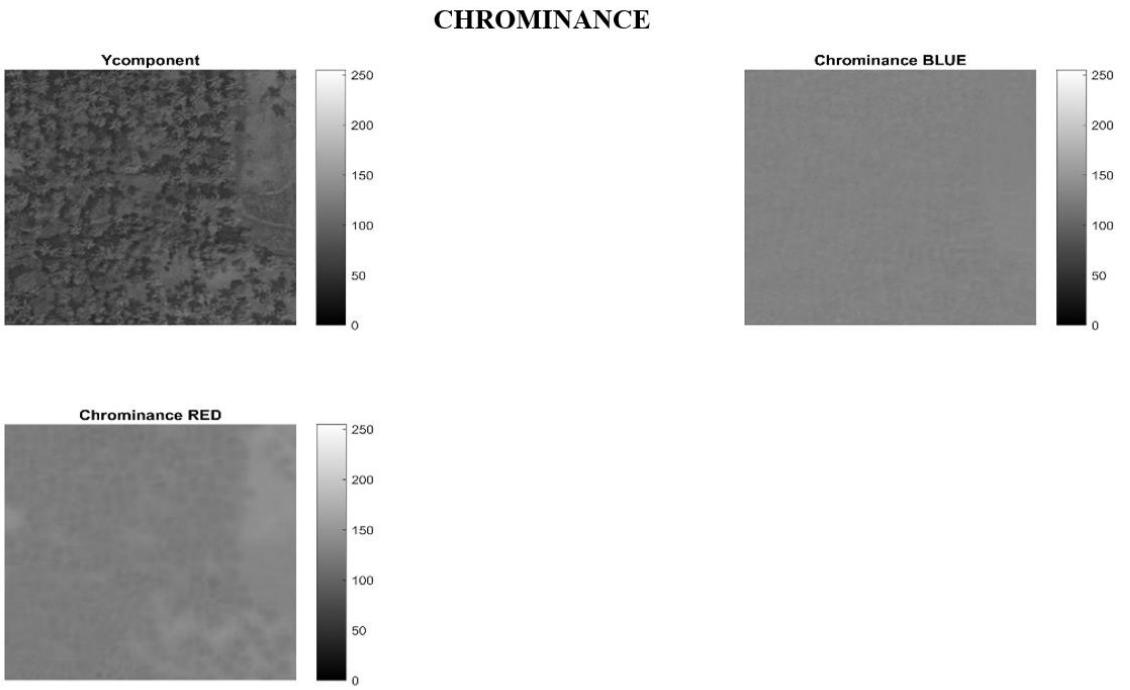


**Fig A2.4(Applying Hue)**

## SEGMENTATION

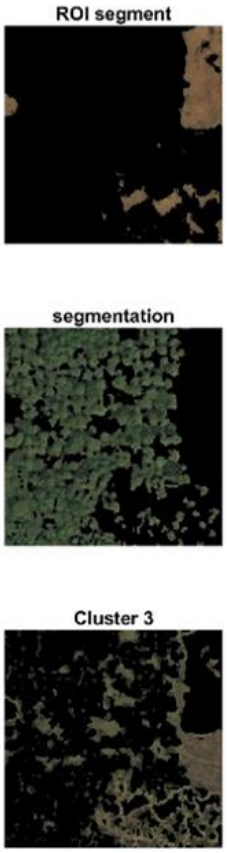


**Fig A2.5(Segmentation)**

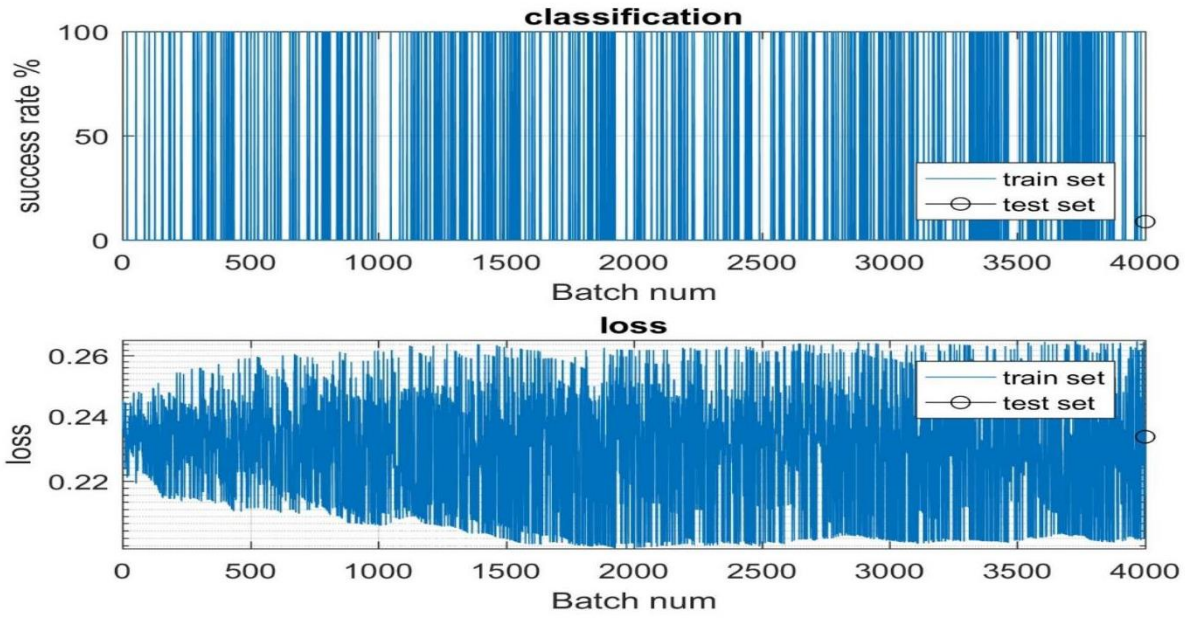


**Fig A2.6(Success and Loss rate)**


**Fig A2.7(Carbon Level)**

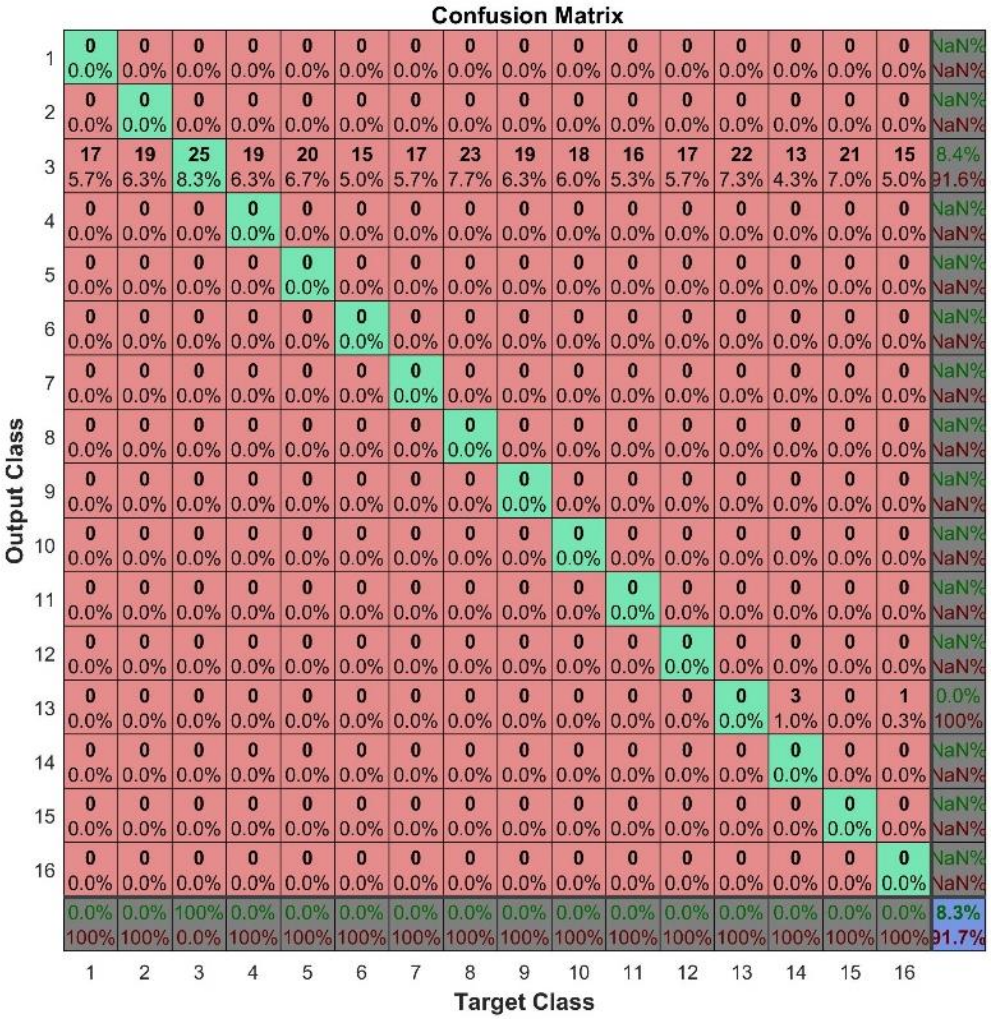


**Fig A2.8(Confusion Matrix)**

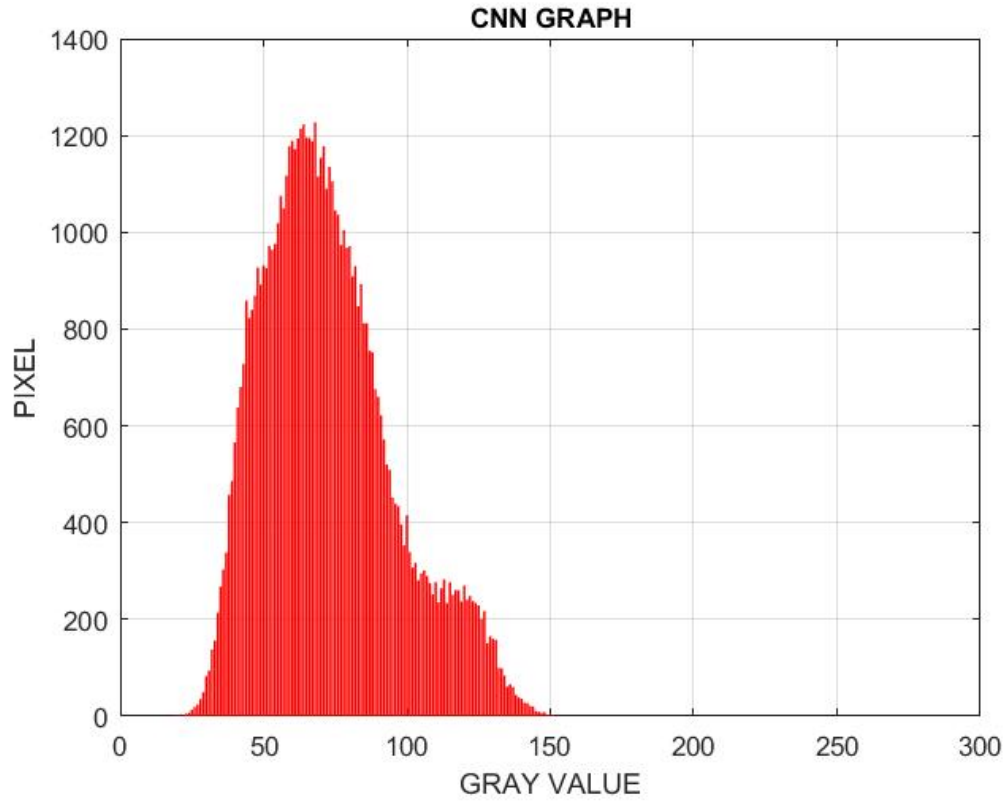


**Fig A2.9(Level of Red)**

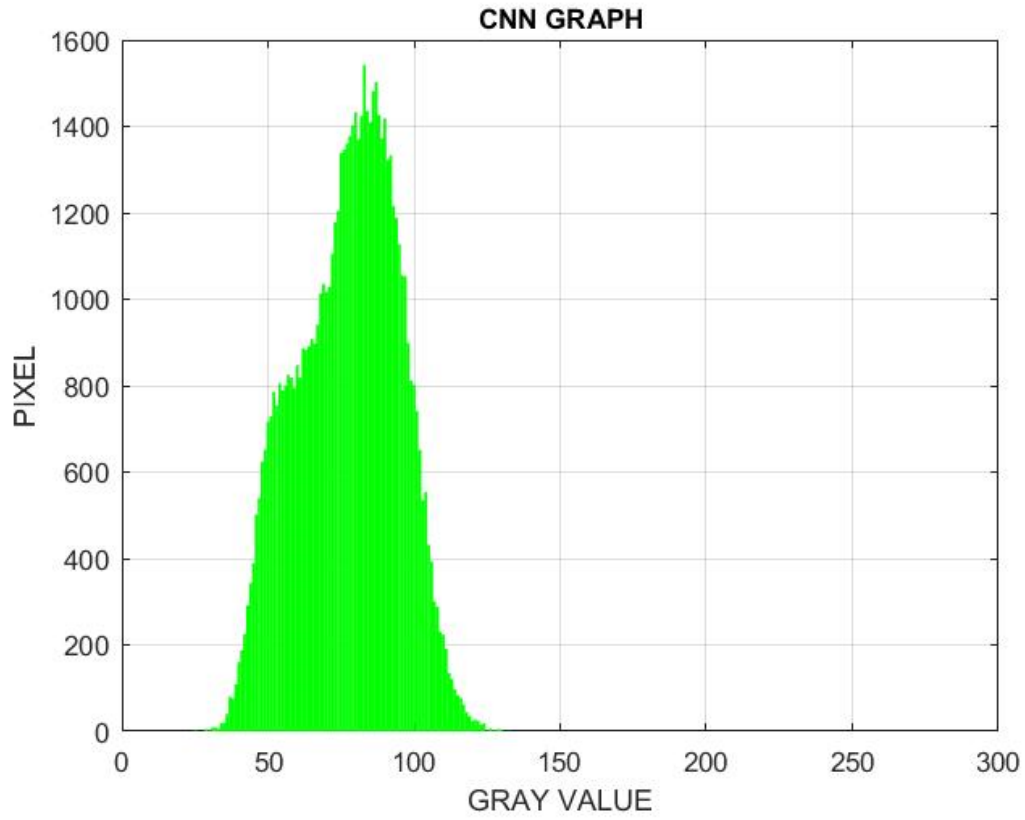


**Fig A2.10(Level of Green)**

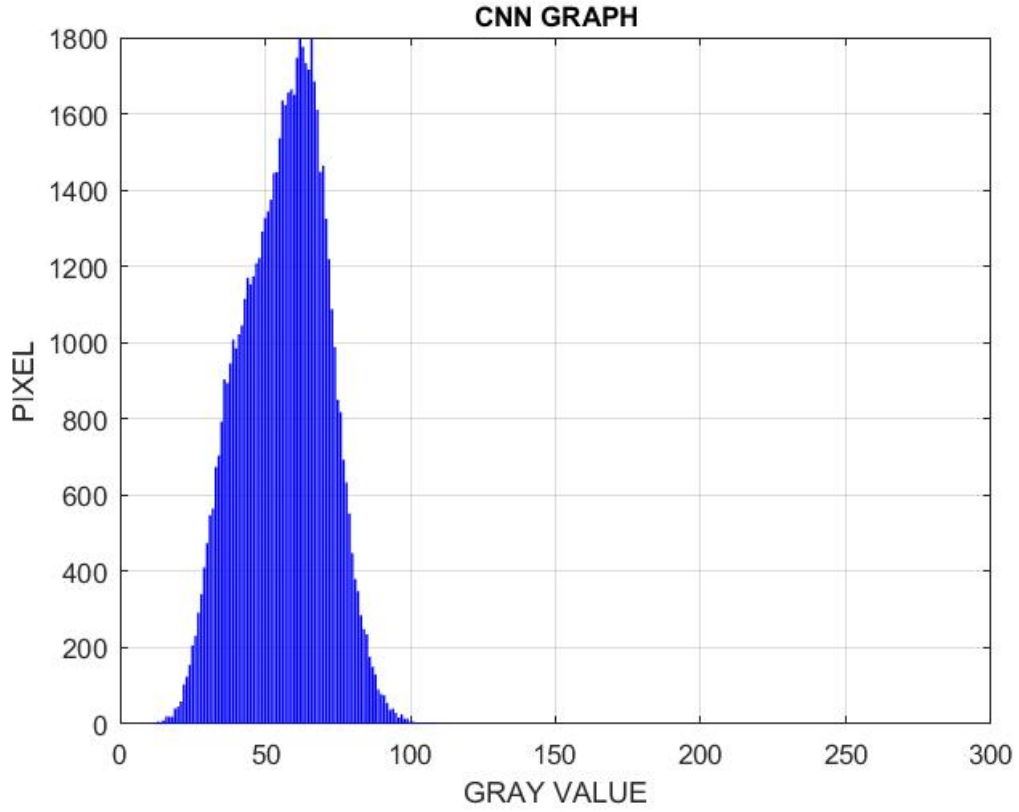


**Fig A2.11(Level of Blue)**

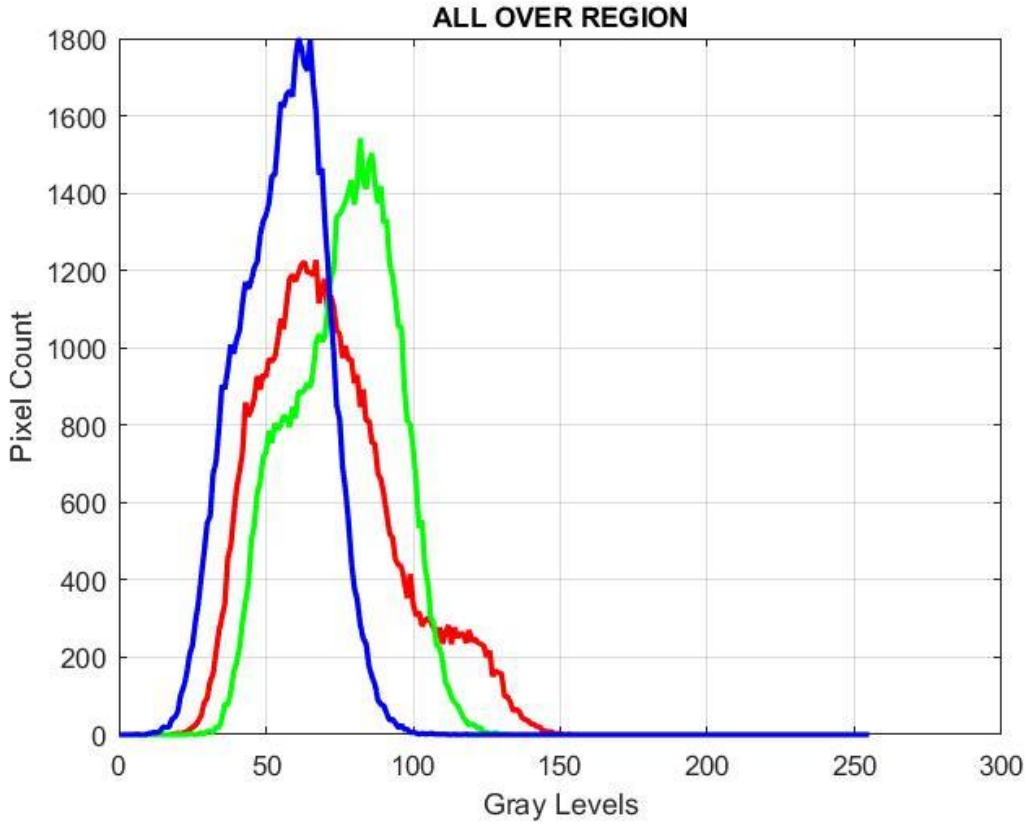


**Fig A2.12(Range of RGB)**


**Fig A2.13(Amount of Carbon)**

## CHAPTER 11

# REFERENCES

[1] Huiyi Su1, Wenjuan Shen1, Jingrui Wang3, Arshad Ali1 and Mingshi Li1,2* "Machine learning and geostatistical approachesfor estimating aboveground biomass in Chinese subtropical forests"
https://doi.org/10.1186/s40663-020-00276-7

[2] Hui Yu a,*, Yufeng Wu a, Liting Niu a
, Yafan Chai a, Qisheng Feng b, Wei Wang c,Tiangang Liang b "A method to avoid spatial overfitting in estimation of grassland above-ground biomass on the Tibetan Plateau"
https://doi.org/10.1016/j.ecolind.2021.107450

[3]Mohamed Musthafa, Gulab Singh, Akshay Patil, Nela Bala Raju, Shradha Mohanty "Forest Above Ground Biomass Estimation Using Multi-Sensor Geostatistical Approach"
https://ieeexplore.ieee.org/document/9323642

[4] Maurizio Santoro1, Oliver Cartus1, Urs Wegmüller1 "Estimation Of Forest Above-Ground Biomass With C-Band Scatterometer Backscatter Observations"
https://ieeexplore.ieee.org/document/9323600

[5] Hasituya1,2, Chen Zhong-xin1,2, Wu Wen-bin1,2, Qing Huang1,2 "Estimation of Above-ground Biomass Carbon Storage in Hulunbeier Grassland based on Remotely Sensed Data"
https://doi.org/10.1109/Agro-Geoinformatics.2015.7248119

[6] Hoscilo Agata1, Lewandowska Aneta, Ziółkowski Dariusz, Stereńczak Krzysztof, Lisańczuk Marek, Schmullius Christiane and Pathe Carsten “Forest Aboveground Biomass Estimation Using A Combination Of Sentinel-1 And Sentinel-2 Data” https://doi.org/10.1109/IGARSS.2018.8517965

[7] Mohamed Musthafa* and Gulab Singh “Improving Forest Above-Ground Biomass Retreival Using Multi-Sensor L- and C- Band SAR Data and Multi-Temporal SpaceBorne LiDAR Data” https://doi.org/10.3389/ffgc.2022.822704

[8] Andreas Braun, Julia Wagner, Volker Hochschild “Above-Ground Biomass Estimates Based On Active And Passive Microwave Sensor Imagery In LowBiomass Savanna Ecosystem” https://www.researchgate.net/publication/329582330_Above-ground_biomass_estimates_based_on_active_and_passive_microwave_sensor_imagery_in_low-biomass_savanna_ecosystems

[9] L. Duncanson1· J. Armston1 M. Disney· V. Avitabile· N. Barbier· K. Calders S. Carter· J. Chave· M. Herold· T. W. Crowther· M. Falkowski J. R. Kellner · N. LabrièreR. Lucas·N.MacBean·R.E. McRoberts · V. Meyer · E. Næsset · J. E. Nickeson“The Importance of Consistent Global Forest Aboveground Biomass Product Validation” https://doi.org/10.1007/s10712-019-09538-8

[10] Fardin Moradi, Seyed Mohammad Moein Sadeghi, Hadi Beygi Heidarlou, Azade Deljouei, Erfan Boshkar, Stelian Alexandru Borz “Above-ground biomass estimation in a Mediterranean sparse coppice oak forest using Sentinel-2 data” https://doi.org/10.15287/afr.2022.2390

[11] Mikhail Urbazaev1,2* , Christian Thiel1, Felix Cremer1, Ralph Dubayah3, Mirco Migliavacca4, Markus Reichstein4 and Christiane Schmullius1 “Estimation of forest aboveground biomass and uncertainties by integration of field measurements, airborne LiDAR, and SAR and optical satellite data in Mexico”
https://doi.org/10.1186/s13021-018-0093-5

[12] Yaohui Zhu1,2,3, Chunjiang Zhao1,2,3, Hao Yang2,3, Guijun Yang2,3,Liang Han2,4, Zhenhai Li2,3, Haikuan Feng2,3, Bo Xu2,3, Jintao Wu2,3 and Lei Lei2,3 “Estimation of maize above-ground biomass based on stem-leaf separation strategy integrated with LiDAR and optical remote sensing data”
http://dx.doi.org/10.7717/peerj.7593

[13] LUONG, V. N.1* – TU, T. T. 1 – KHOI, A. L.1 – HONG, X. T. 1 – HOAN, T. N.2 – THUY, T. L. H.2 “Biomass Estimation And Mapping Of Can Gio Mangrove Biosphere Reserve In South Of Viet Nam Using Alos-2 Palsar-2 Data”
http://dx.doi.org/10.15666/aeer/1701_015031

[14] Daniele De Rosa a,* , Bruno Basso b, Matteo Fasiolo c, Johannes Friedl a, Bill Fulkerson d, Peter R. Grace a, David W. Rowlings a “Predicting pasture biomass using a statistical model and machine learning algorithm implemented with remotely sensed imagery”
https://doi.org/10.1016/j.compag.2020.105880

[15] Tiago G. Morais a,*, Ricardo F.M. Teixeira a, Mario Figueiredo b, Tiago Domingos a “The use of machine learning methods to estimate aboveground biomass of grasslands: A review”
https://doi.org/10.1016/j.ecolind.2021.108081

[16] Ktawut Tappayuthpijarn1 and Bernd S Vindevogel “High-accuracy Machine Learning Models to Estimate above Ground Biomass over Tropical Closed Evergreen Forest Areas from Satellite Data”
https://www.researchgate.net/publication/359679000_High-accuracy_Machine_Learning_Models_to_Estimate_above_Ground_Biomass_over_Tropical_Closed_Evergreen_Forest_Areas_from_Satellite_Data

[17] M.A. Stelmaszczuk-Górska, C.J. Thiel and C.C. Schmullius “Remote Sensing for Aboveground Biomass Estimation in Boreal Forests”
https://www.nature.com/articles/s41598-020-67024-3

[18] F. Pirotti, E. Kutchartt, E. Csaplovics 3 “Assessment Of Volume And Above-Ground Biomass In Araucaria Forest Through Satellite Images, Comparing Different Methods In The South Of Chile”
https://doi.org/10.5194/isprs-archives-XLII-3-W12-2020-331-2020

[19] Thota Sivasankar, Junaid Lone, K K Sarma, Abdul Qadir, Raju P.L. N. “Estimation of Above Ground Biomass Using Support Vector Machines and ALOS/PALSAR data”
https://vjs.ac.vn/index.php/jse/article/view/13690

[20] Daniel Kükenbrink, , Oliver Gardi, Felix Morsdorf, Esther Thürig, Andreas Schellenberger and Lukas Mathys “Above-ground biomass references for urban trees from terrestrial laser scanning data”
https://academic.oup.com/aob/article/128/6/709/6165072

[21] Simon L. Lewis, Bonaventure Sonke, Terry Sunderland, Serge K. Begne, Gabriela Lopez-Gonzalez, Geertje M. F. van der Heijden, Oliver L. Phillips, Kofi Affum-Baffoe,

Timothy R. Baker, Lindsay Banin, Jean-Franc¸ois Bastin, Hans Beeckman, Pascal Boeckx, Jan Bogaert, Charles De Cannie`re, Eric Chezeaux, Connie J. Clark, Murray Collins, Gloria Djagbletey, Marie Noe¨l K. Djuikouo, Vincent Droissar, Jean-Louis Doucet, Cornielle E. N. Ewango, Sophie Fauset, Ted R. Feldpausch, Ernest G. Foli, JeanFranc¸ois Gillet, Alan C. Hamilton, David J. Harris, Terese B. Hart, Thales de Haulleville, Annette Hladik, Koen Hufkens, Dries Huygens, Philippe Jeanmart, Kathryn J. Jeffery, Elizabeth Kearsley, Miguel E. Leal, Jon Lloyd, Jon C. Lovett, Jean-Remy Makana, Yadvinder Malhi, Andrew R. Marshall, Lucas Ojo, Kelvin S.-H. Peh, Georgia Pickavance, John R. Poulsen, Jan M. Reitsma, Douglas Sheil"Above-ground biomass and structure of 260 African tropical forests"
https://royalsocietypublishing.org/

[22] Baoping Meng , Tiangang Liang, Shuhua Yi, Jianpeng Yin , Xia Cui, Jing Ge , Mengjing Hou, Yanyan Lv, and Yi Sun "Modeling Alpine Grassland Above Ground Biomass Based on Remote Sensing Data and Machine Learning Algorithm: A Case Study in East of the Tibetan Plateau, China"
https://ieeexplore.ieee.org/abstract/document/9106792

[23] Lalit Kumar 1,*ID and Onisimo Mutanga 2 "Remote Sensing of Above-Ground Biomass"
http://dx.doi.org/10.3390/rs9090935