A project report on

Land Cover Detection using Machine Learning

Submitted in partial fulfilment of the requirements for the degree of

Bachelor Of Technology

in

Information Technology

by

NAME: Paras Dang

REG NO: 19BIT0378

Under the guidance of

Dr. Kumaresan P.

SCHOOL OF INFORMATION TECHNOLOGY & ENGINEERING

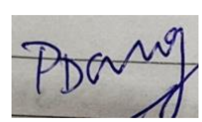
VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

DECLARATION

I hereby declare that the project entitled "Land Cover Detection System" submitted by **Paras Dang** (19BIT0378) for the award of the degree of Bachelor of Technology in "Information Technology respectively, to VIT is a record of bonafide work carried out under the supervision of Dr. Kumaresan P. I would really like to thank Dr. Kumaresan P for his extreme support.

I further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place : Vellore
Date : 3/04/2022



CERTIFICATE

This is to certify that the project entitled "Land Cover Detection" submitted by Paras Dang, 19BIT0378, SITE, VIT University, for the award of the degree of Bachelor of Technology in Information Technology is a record of bonafide work carried out by him under my supervision during the period Dec 2022 - May 2023, as per the VIT code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diplomain this institute or any other institute or university. The project fulfils the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

Place: **Vellore**

Date: **3rd April 2023**

Signature of the Guide

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

Abstract & Introduction

In every region of the world, information spreads very quickly due to the available learning techniques of the known era, which has also reduced the time consumption of information consumption. There are various platforms like movies that are available online, business stages that are happening on the internet, all these information are available which can be referenced in large content.

If we use color bars that are pure in nature, it will help some public authorities to update the land information. It will be a sensing platform, so it will take less time and the cost we will incur will not be too high, in this way the aircraft system will be used, which should be unmanned. Here we will use a method for optical aerial imaging that will fall under decision tree land use classification. At the beginning, a proprietary parcel map will be in play to obtain land cover information using a maximum likelihood classifier. Then, to establish the relationship between land use and land cover age, we generate a decision tree. A well-structured parcel map will be created based on the geographical characteristics of the parcels. This land use can be widely used in landscape and urban planning thanks to the flexible classification method. Using aerial imagery, an object-oriented approach for image- based analysis is used to map urban land cover. The investigation of image classification and segmentation will proceed through the Analytical Hierarchy process. Fuzzy classification helps in classifying buildings as high or low, as well as roads and grasslands, as these are land cover types for image segmentation.

Interactions

16/12/22 0^{th} Review, Prof Kumaresan P
27/01/23 1^{st} Review Mock, Prof Kumaresan P
01/02/2023 1^{st} Review, Prof Bharanidharan N & Prof Sudha M
13/03/2023 2^{nd} Review Prof Bharanidharan N & Prof Sudha M
3^{rd} Review – 17^{th} April

ACKNOWLEDGEMENT

It is my pleasure to express with deep sense of gratitude to Dr. Kumaresan P, Senior Assistant Professor, SITE, Vellore Institute of Technology, for his constant guidance, continual encouragement, understanding; more than all, he taught me patience in my endeavor. My association with him is not confined to academics only, but it is a great opportunity on my part of work with an intellectual and expert in the field of technology.

I would like to express my gratitude to DR.G.VISWANATHAN, Chancellor VELLORE INSTITUTE OF TECHNOLOGY, VELLORE, MR. SANKAR VISWANATHAN, DR. SEKAR VISWANATHAN, MR.G V SELVAM, Vice – Presidents VELLORE INSTITUTE OF TECHNOLOGY, VELLORE, DR. RAMBABU KODALI, Vice – Chancellor, DR. S. NARAYANAN, Pro-Vice Chancellor and Dr. S. Sumathy, Dean, School of Information Technology & Engineering (SITE), for providing with an environment to work in and for his inspiration during the tenure of the course.

In jubilant mood I express ingeniously my whole-hearted thanks to Dr. Usha Devi, HoD, all teaching staff and members working as limbs of our university for their not-self-centered enthusiasm coupled with timely encouragements showered on me with zeal, which prompted the acquirement of the requisite knowledge to finalize my course study successfully. I would like to thank my parents for their support.

It is indeed a pleasure to thank my friends who persuaded and encouraged me to take up and complete this task. At last but not least, I express my gratitude and appreciation to all those who have helped me directly or indirectly toward the successful completion of this project.

Place: Vellore
Date: 3rd April 2023

Paras Dang
19BIT0378

CHAPTER 2

Literature Reviews

TITLE	AUTHORS	MERITS	DEMERITS
Prediction of land use and land cover change through cell-based self-adaptive deep learning with multi-source data. (2019)	Lin Mu, Lizhe Wang, Yuewei Wang, Xiaodao Chen, and Wei Han	Trustworthy and reliable, which means to gain explainability. Maintaining overhead control at regular levels,Fast and efficient, but also as accurate as the most modern algorithms	Heavy weight, High prediction complexity for large datasets Responsive to dataset size
Testing the effectiveness of using high resolution data from GF-1 in land cover classifications(2018)	Xiaofeng Wang, Chaowei Zhou, Xiaoming Feng, Changwu Cheng and Bojie Fu	It can scale in proportion to the amount of training data available.	Difficulty in achieving better performance Higher correlations in prediction errors, Not thoroughly explored
Semi-MCNN: Semi-supervised Multi-CNN EnsembleLearning Method for Urban Land Cover Classification Using Submeter HRRS Images	Runyu Fan, Ruyi Feng, Lizhe Wang, Jining Yan, and Xiaohan Zhang	Ability to further reduce the required level of human effort Demonstrates high robustness and imperceptibility Simple to use and interpret	They have not been thoroughly researched. Accepts the size of the data file, Unable to meet current network business requirements
Convolutional	Miae Kim,	He found	The variance of the

Neural Network-Based Land Cover Classification Using 2D Spectral Reflectance Curve Plots with Multitemporal Satellite Imagery	Junghee Lee, Daehyeon Han, Minso Shin, JunghoIm, Junghye Lee, Lindi J. Quackenbush, and Zhu Gu	attractive results,Optimal learning on extracted features, Computational complexity is significantly reduced	predictions cannot be reduced; therefore, it cannot reduce the generalization error. Poor application performance,Cannot be implemented in real time.
Evaluation and prediction of changes in ecosystem service values based on land use/land cover changes with a random forest and cellular automata model in the Qingdao metropolitan area, China	Xiaochuan Qin and Bihong Fu	A relatively simple and computationally undemanding method It can improve worst-case performance Reduce resource consumption while meeting reliability requirements	Very computationally intensive when training the model Higher correlation in prediction errors Manual annotation requires human labor.
Land cover changes can be determined by statistical evaluation by comparing temporal or differently dated land cover maps.	(Choukiker & Dohare, 2021)	Efficient and Fast	Not cost effective
Land Cover Detection using Transition Matrix	Isabel et al. ., 2011; Lu et al., 2014	As accurate as the most modern algorithms.	High prediction complexity for large datasets.
Land cover classification using 2D plots of spectral reflectance curves with multitemporal	Junghye Lee, Lindi J. Quackenbush, and Zhu Gu	Computational complexity is significantly reduced	Poor application performance,Cannot be implemented in real time.

satellite imagery.			
Land Cover Detection using High Level remote sensing	Xiaoming Feng, Changwu Cheng and Bojie Fu (2018)	It reduces generalization error. Discriminative features learned by the model.	Higher correlation in prediction errors,
The prediction of urban land use and land cover (LULC)	Yuewei Wang, Xiaodao Chen	Maintaining overhead control at regular levels,Fast and efficient	High prediction complexity for large datasets Responsive to dataset size

CHAPTER 3

ISSUES IN EXISTING SYSTEM

1. Huge payloads
2. Problems existing in their respective network structures.
3. This system is Opportunistic and not controllable
4. Cannot restrain behaviour pattern in very complex systems
5. High complexity, inaccuracy, and inadequacy

SCOPE

The main contributions of this project are:

- Data Analysis
- Data Pre-processing
- Training the Model
- Testing of Test Data

PROPOSED SYSTEM

The proposed framework consists of low-density localities (LDR), medium-density neighbourhoods (MDR), high-density localities (HDR), commercial regions, forests, agricultural areas, open spaces (parks), and streets. The Google Earth Street View application was used to help decide on the type of land use of the parcels. Different selection trees were prepared for images with different spatial targets in SPSS, a measurement product.

The main points we used to construct the selection tree were pack area, stand length, and the proportion of each land cover type within the stand. Chi-square programmed association detection (CHAID) was used as the chosen strategy for growing the tree. The proposed framework relies on a tree selection based land use planning strategy that is guided using an optical aerial symbology that has red, blue and green groups and a property bundle map.

Then, these created images are used to characterize the land use using a similar methodology to decide on the impacts of the land use objective. The proposed framework only uses the red, blue, and green groups of ortho-symbolism to meet the goal of delivering a land use map with minimal cost and to move the handling of UAV optical symbology.

CHAPTER 4

Algorithm to be used and Why?

<u>Maximum Likelihood Estimation</u>

Maximum likelihood estimation (MLE) is a probabilistic approach to determining model parameter values. The parameters could be defined as blueprints of the model, since this is what the algorithm works on. MLE is a widely used technique in machine learning, time series, panel data, and discrete data. The motive of MLE is to maximize the likelihood of values for a parameter to achieve the desired results.The proposed system utilizes a probability-based approach to classify each pixel into a specific land use class. The probability that a pixel belongs to a particular class is calculated, and then the pixel is assigned to the class with the highest probability. This approach allows for accurate and efficient land use classification. Machine learning models can generally be formulated using maximum likelihood estimation, which provides a reliable and systematic approach for predictive modelling as an optimization problem. The maximum likelihood classifier is a statistical method that is used to classify data based on the maximum likelihood estimate. It is based on the idea of finding the parameters that maximize the likelihood of the observed data. In the example given, the data generating process is assumed to follow a Gaussian distribution, and the maximum likelihood classifier is used to estimate the mean and standard deviation of the distribution that best fit the observed data. The maximum likelihood estimate is obtained by maximizing the likelihood function, which is a function of the parameters of the distribution. The likelihood function gives the probability of observing the data given a set of parameter values. The maximum likelihood estimate is the set of parameter values that maximize this function. The maximum likelihood classifier is a parametric method, which means that it assumes a specific form for the distribution of the data. In this case, the Gaussian distribution is assumed. The parameters of the distribution, i.e., the mean and standard deviation, are estimated using the maximum likelihood method. The maximum likelihood classifier is a powerful tool for data classification, and it is widely used in various applications such as image classification, speech recognition, and natural language processing. However, it is important to note that the assumptions made about the distribution of the data can affect the accuracy of the classification. Hence, it is crucial to choose the appropriate model and make sure that the assumptions made about the data are reasonable.

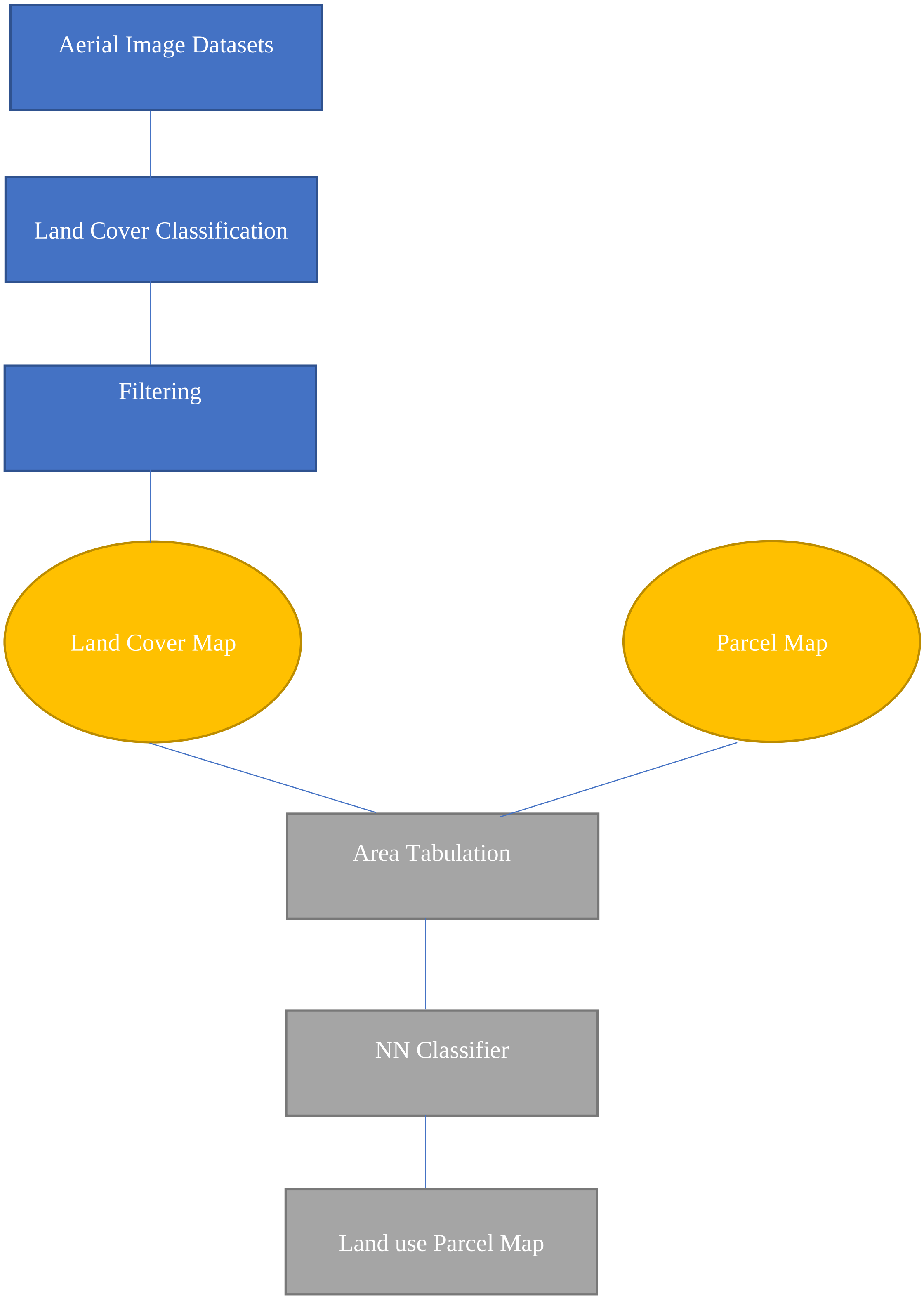
Advantages

a) Predictable and good way to deal with boundary assessment problems.

b) Have attractive numerical and optimality properties.

c) Mitigates the computational intricacy of greatest probability assessment.

Architecture Design



Aerial Image Dataset

An aerial image dataset is a collection of images captured by cameras mounted on airborne platforms such as drones, satellites, airplanes, or helicopters. These images provide a bird's-eye view of the earth's surface and can be used for a variety of applications, including urban planning, environmental monitoring, agriculture, and disaster management. The aerial image dataset typically contains high-resolution images, which are obtained through remote sensing techniques. These images are usually in the form of RGB (red, green, blue) or multispectral images, which can be used to extract information about the land surface, such as vegetation cover, water bodies, buildings, and roads. Aerial image datasets can be processed using machine learning algorithms to extract useful information from the images. For example, object detection algorithms can be used to identify buildings or vehicles, while image segmentation techniques can be used to distinguish different land covers. The information extracted from the aerial image dataset can be used for decision-making purposes, such as planning land use, assessing environmental impacts, or responding to natural disasters. It is important to note that aerial image datasets may have limitations, such as cloud cover or atmospheric distortions, which can affect the quality of the images. Furthermore, acquiring aerial images can be expensive and time-consuming, requiring specialized equipment and trained personnel. Nevertheless, aerial image datasets continue to be an important tool for a variety of applications that require a broad view of the earth's surface.

Land Cover Classification

Land cover classification is a process of categorizing the land surface into different classes based on the physical and biological characteristics of the area. It involves the identification and mapping of different land cover types, such as forests, grasslands, wetlands, water bodies, and urban areas. Land cover classification is typically done using remote sensing techniques, which involve the use of aerial or satellite images to capture information about the land surface. These images are then analyzed using image processing and machine learning algorithms to extract information about the land cover types present in the area. The classification process involves several steps, including image preprocessing, feature extraction, and classification. During the image preprocessing step, the images are corrected for atmospheric distortions, radiometric variations, and geometric distortions to ensure accuracy in subsequent steps.

Next, feature extraction techniques are used to identify and extract information about the different land cover types in the images. This may involve the use of spectral, textural, and contextual features to distinguish between different land cover types.

Finally, classification algorithms are used to assign each pixel in the image to a specific land cover class based on its features. The classification algorithms can be supervised or unsupervised, depending on whether or not training data is available.

The resulting land cover maps can be used for a variety of applications, such as natural resource management, land use planning, and environmental monitoring. Land cover classification can help identify areas that are vulnerable to environmental degradation or habitat loss, and can assist in the development of strategies to mitigate these issues. Overall, land cover classification is an important tool for understanding and managing the earth's surface, and remote sensing techniques have made it possible to conduct this analysis on a large scale with high accuracy and precision.

Filtering

Filtering is the process of removing noise and irrelevant information from a dataset to improve the accuracy of a model. It involves the use of mathematical techniques to identify and eliminate unwanted data points, so that the remaining data can be used to train a more accurate and effective model.

There are several types of filtering techniques used in machine learning, including:

1. Gaussian filtering: This technique uses a Gaussian filter to smooth out the data by convolving the data with a Gaussian kernel. This helps to reduce the noise and make the data more uniform.
2. Median filtering: This technique replaces each pixel in an image with the median value of its neighboring pixels. This helps to remove any outliers or noise that might be present in the data.
3. Low-pass filtering: This technique removes high-frequency noise from the data by attenuating the high-frequency components of the signal. This can be useful in audio and image processing applications.
4. High-pass filtering: This technique removes low-frequency noise from the data by attenuating the low-frequency components of the signal. This can be useful in image processing applications, where low-frequency noise can appear as a uniform background.
5. Band-pass filtering: This technique filters the data to pass only a specific frequency range, while rejecting frequencies outside of that range. This can be useful in signal processing applications, such as EEG analysis or audio filtering.

Filtering is an important step in machine learning, as it helps to improve the quality of the data used to train a model. By removing unwanted noise and irrelevant information, filtering can help to increase the accuracy and efficiency of a model, and can lead to better results in real-world applications.

Land Cover Map

A land cover map is a type of map that represents the different types of land cover in a particular geographic area. It provides information about the types of vegetation, water bodies, urban areas, and other types of land use present in the area.The map is created using a variety of data sources, such as satellite imagery, aerial photographs, and ground surveys. This data is then analyzed and classified into different categories, such as forests, grasslands, water bodies, and urban areas, based on their unique spectral characteristics. Land cover maps are important for a variety of reasons. They can be used to monitor changes in land use over time, which can help policymakers and land managers make informed decisions about conservation, land use planning, and natural resource management. Land cover maps can also be used to identify areas that are at risk for environmental hazards, such as flooding, landslides, or wildfires. Overall, land cover maps are a valuable tool for understanding the spatial distribution and characteristics of different types of land cover in a given area. They help provide a detailed understanding of the physical and ecological features of the landscape, which can inform decision-making about land management and conservation.

Parcel Map

A parcel map is a type of map that displays the boundaries of individual land parcels or properties within a geographic area. These maps are typically created and maintained by local government agencies, such as county assessor's offices or land surveying departments. Parcel maps are important for a variety of purposes. They are used to determine property boundaries, ownership, and other legal and financial information related to land parcels. They are also used by real estate professionals, appraisers, and developers to assess property values and identify potential development opportunities.
Parcel maps are typically created using a combination of surveying, mapping, and GIS (Geographic Information System) technology. The process involves collecting and analyzing data about the location, size, and shape of individual land parcels, as well as any natural or man-made features that may affect their boundaries. Once the data is collected and analyzed, it is then used to create a map that accurately represents the location and boundaries of each individual parcel. These maps may include detailed information such as property lines, easements, zoning districts, and other relevant data.
Overall, parcel maps are a valuable tool for understanding the physical and legal characteristics of land parcels within a given area. They are used by a wide range of professionals to inform decision-making about land use, development, and other important activities related to the use and management of land.

Area Tabulation

Area tabulation is a process of summarizing and organizing geographic data into a table format, typically based on a set of predefined geographic regions or zones. The tabulation process involves aggregating data values for each region or zone and creating a table that displays the total values for each region or zone. Area tabulation is commonly used in GIS (Geographic Information System) analysis and spatial data management. It allows users to analyze and compare data for different regions, such as states, counties, or census tracts quickly and efficiently. The tabulation process can be used to calculate a wide range of data values, including population, land use, demographic characteristics, and other relevant information. The resulting tables can be used to identify trends, patterns, and relationships within the data and to inform decision-making about a wide range of topics, such as public health, land use planning, and environmental management. Overall, area tabulation is a powerful tool for summarizing and analyzing geographic data, allowing users to gain insights quickly and easily into complex patterns and relationships within the data. It is a critical component of many GIS applications and is widely used in a variety of fields, from urban planning to public health to environmental science.

NN Classifier

A nearest neighbor (NN) classifier is a machine learning algorithm used for classification tasks. It is a type of instance-based learning, where the algorithm makes predictions based on the similarity between new data points and the training data. The NN classifier works by comparing the new data point to the training data points, calculating the distance between them, and finding the closest data point(s). The classification of the new data point is then determined by the class of the closest data point(s). The "nearest neighbor" refers to the closest training data points, which can be determined using a variety of distance metrics, such as Euclidean distance or Manhattan distance. The number of nearest neighbors used for classification is typically determined by a parameter called "k". For example, a 3-NN classifier would use the class of the three closest training data points to classify the new data point. The NN classifier is a simple and effective algorithm that can be used for a variety of classification tasks, such as image recognition, natural language processing, and anomaly detection. It does not require a complex model or prior assumptions about the data, which makes it easy to implement and interpret. However, it can be sensitive to outliers and noisy data, and may not perform as well as more complex algorithms on large or high-dimensional datasets. Overall, the NN classifier is a powerful and flexible machine learning algorithm that is commonly used in a variety of applications. It provides a straightforward way to classify new data points based on their similarity to the training data, making it a valuable tool for many types of data analysis and decision-making.

Land Use Parcel Map

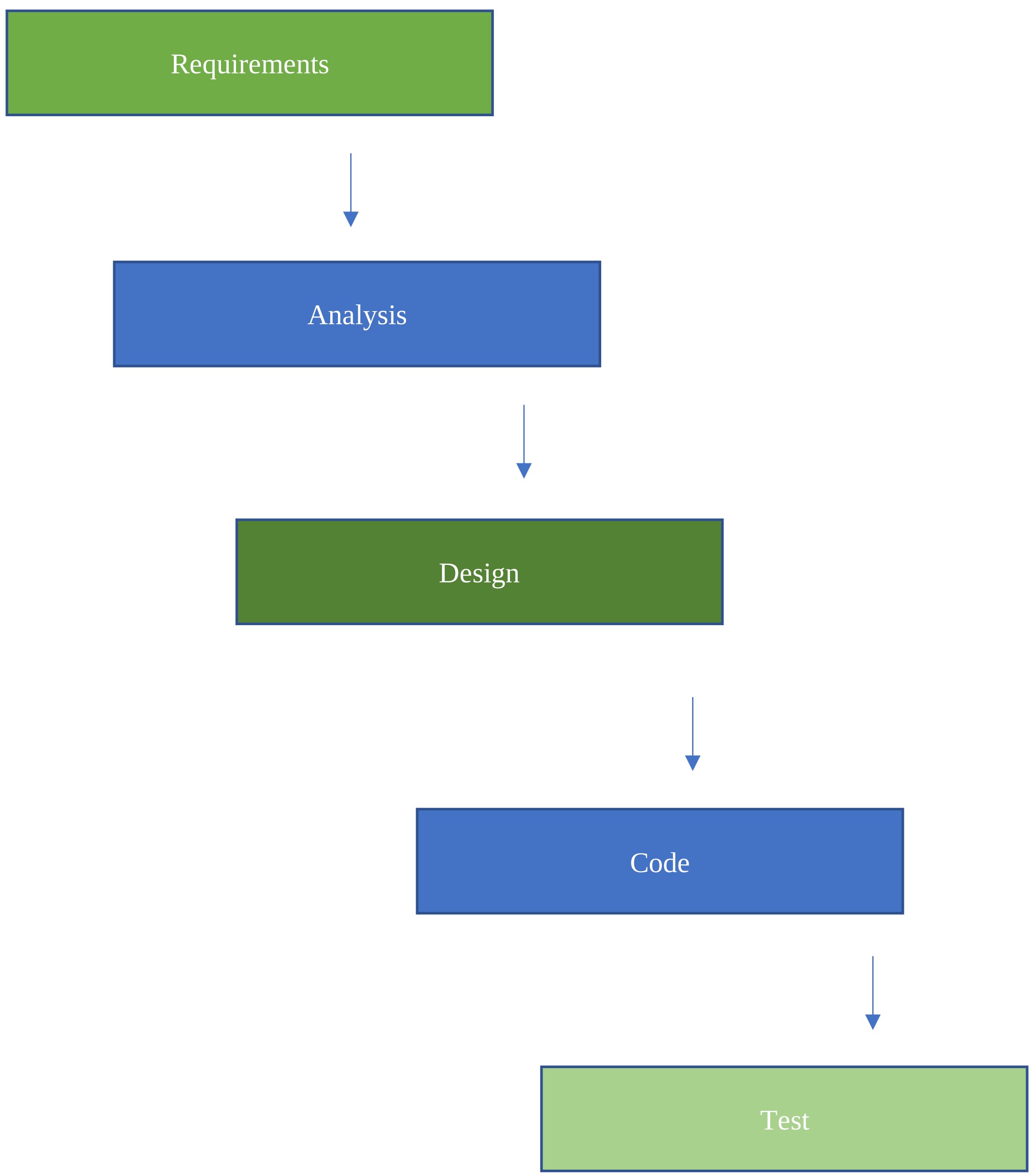
A land use parcel map is a type of map that displays the different land uses within individual parcels or properties within a geographic area. These maps are typically created and maintained by local government agencies, such as planning departments, to inform land use planning and zoning decisions. Land use parcel maps provide a detailed understanding of how individual parcels are being used within a given area. They show the different types of land uses, such as residential, commercial, industrial, agricultural, and open space, and the distribution of these land uses across the landscape. The maps are created using a combination of GIS (Geographic Information System) technology and land use planning data, such as zoning maps and land use permits. The process involves collecting and analyzing data about the land use of each individual parcel, and then classifying each parcel into one of the land use categories. Once the data is collected and analyzed, it is then used to create a map that accurately represents the land use of each individual parcel within the area of interest. These maps may include detailed information such as the location and boundaries of each parcel, the area of each parcel, and the land use category of each parcel. Overall, land use parcel maps are a valuable tool for understanding the distribution and characteristics of different land uses within a given area. They provide information that can be used to inform decision-making about land use planning and zoning, and they are used by a wide range of professionals, such as urban planners, real estate developers, and government officials, to inform land use policy and management.

CHAPTER 6

<u>System Analysis</u>

Waterfall Method

It is a linear and sequential software development methodology. In this approach, the application development process is divided into a series of predefined phases, with each phase being completed before moving on to the next.



- **Requirement Gathering and Documentation:** In this initial phase, the project team collects and analyses information about the problem to be solved. This can be done through various methods such as interviews, questionnaires, or brainstorming sessions. By the end of this phase, a comprehensive requirements document should be created and assigned to the team for further development.

- **System Design:** Using the requirements document as a reference, the team begins designing the system solution. This phase involves specifying hardware and software requirements, as well as creating system architecture and design models. No coding takes place during this phase.

- **Implementation:** In this phase, software development begins with coding and programming. Web application programmers take the design specifications and create a functional product. Source code is written in small pieces that are later integrated during this phase or at the start of the next.

- **Testing:** Once the coding is completed, thorough testing of the product can begin. Testers methodically search for and report any issues that they find. Serious problems may require the project to return to phase one for reassessment.

- **Delivery/Deployment:** With successful testing, the solution is complete and ready to be deployed or released to the client. The project team submits the deliverables and oversees the deployment process.

- **Maintenance:** After the final solution is implemented, it will require ongoing maintenance and support. The project team will need to address any issues that arise and may need to create patches and updates to ensure the solution continues to function correctly. Major issues may require the project to return to phase one for reassessment.

Proposed Methodology

The proposed system includes several land use types, such as low-density residential areas (LDR), medium density residential areas (MDR), high density residential areas (HDR), commercial areas, forest, crop lands, open space (parks), and roads. To determine the land use type of each parcel, Google Earth Street View was utilized. Decision trees were constructed in SPSS using various spatial resolutions, and parcel area, parcel length, and the ratio of each land cover type within a parcel were used as features. The decision tree growing up method CHAID was employed for tree construction. Our proposed system uses optical aerial imagery with red, blue, and green bands, as well as an Ownership parcel map for land use classification. The aerial imagery was resampled at 20cm to 50cm and 50cm to 100cm resolutions. This approach was applied to classify land use and investigate the effects of spatial resolution. Dataset was obtained from Kaggle.

System Implementation

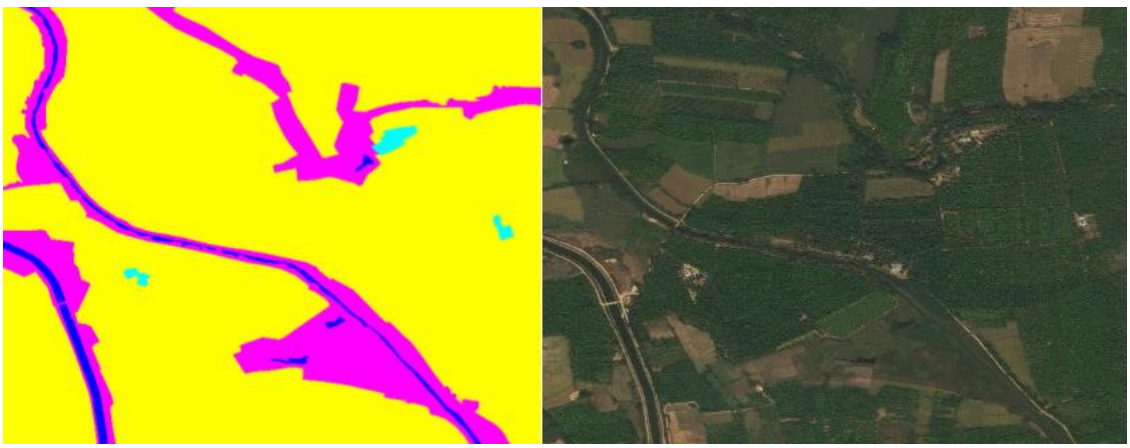
Module 1: Data Collection And Pre-processing

The initial step in any mining technique is to collect the data needed for analysis. The first phase of data cleaning is crucial in pre-processing to obtain cleaned data for further processing. The data transformation process converts the collected data into a unique format suitable for the mining technique. Typically, the downloaded dataset contains unstructured, noisy, and irrelevant data. To make the data appropriate for pattern mining and analysis, it must be subjected to data pre-processing. Data pre-processing not only enhances data quality but also reduces the dataset's size.

(i)

Module 2: Feature Selection and Feature Extraction

Both feature selection and feature extraction are utilized to reduce model complexity and overfitting by reducing dimensionality. Dimensionality reduction is one of the most critical components of training machine learning models. Feature selection entails selecting a subset of features from the original set to decrease model complexity, improve computational efficiency, and decrease generalization errors caused by irrelevant features. Feature extraction is the process of deriving information from the original features set to create a new feature subspace. The primary objective of feature extraction is to compress the data while retaining most of the pertinent information. Similar to feature selection techniques, these methods are used to decrease the number of features from the original set to decrease model complexity, model overfitting, enhance model computation efficiency, and decrease generalization errors.



(ii)

Module 3: Model Training

One approach to quickly evaluating the algorithm's performance on your problem is to create a train and test split of your dataset. The training dataset is employed to prepare and train the model. The test dataset is considered as new data where the algorithm's output values are withheld. We gather predictions from the trained model on the inputs from the test dataset and compare them to the withheld output values of the test set.

By comparing the predictions and withheld outputs on the test dataset, we can calculate a performance measure for the model on the test dataset. This estimate represents the skill of the algorithm when making predictions on unseen data. The final machine learning model is the model used to make predictions on new data. To train a model, access to the data, several utility functions, and multiple iterations/passes through the dataset are required. The training process effectively utilizes both functions repeatedly: The model's parameters are initially randomly instantiated. Next, the model's score is checked. If the score is deemed insufficient, (often referred to as the cost or loss function), the model's parameters are adjusted. This adjustment is accomplished through the use of a selected optimization algorithm to update the model's parameters. This process is iterated until an acceptable score is obtained. Once the training phase is completed, the final model is obtained, and it can be used to make predictions on new data.

(iii)

Hardware Requirements

- Processor that have 1.4 GHz 64-bit processor
- Disk with Space 100GB Free Space
- RAM Minimum 8GB
- Graphics Device Super VGA (1024 x 768) or higher-resolution

Software Requirements

- Python
- Anaconda
- Jupyter Notebook
- TensorFlow Package
- Plotly Package
- Matplotlib Package

GANTT Chart

Date	Acitivity	Description of the Activity	Guide Remarks
16/12/22	1	Abstract, Problem Description and analyzing Research Papers.	Go on, good ideology.
1/02/2023	2	Features, and thoughts on Implementation and various algorithms to be used.	Try to create an impact by using algorithm to minimize issues.
13/03/2023	3	Implementation and Features to be used.	Complete the implementation and usage of features
3/04/2023	4	Conclusion and Future scope.	Validation Pending

Python

Python is a high-level, interpreted programming language with dynamic semantics and object-oriented features. It offers built-in data structures, dynamic typing, and dynamic binding, which makes it an excellent choice for Rapid Application Development and for scripting or gluing existing components. The language's easy-to-learn syntax emphasizes readability and reduces the cost of program maintenance. Python supports modularity and code reuse through its modules and packages. The Python interpreter and standard library are freely available in both source and binary form and can be used on all major platforms.

Jupyter Notebook

Jupyter Lab is a popular digital development environment for code, data, and research papers. Its versatile interface allows users to design and manage workflows in various fields, such as data science, computational statistics, scientific computing, and AI. The Jupyter Notebook is the primary web application for creating and sharing computational reports. It combines code and outputs into a single document that includes visuals, narrative text, mathematical equations, and other rich media. In essence, it's a single file where you can execute code, display outputs, add annotations, explanations, charts, and make your work more reliable, transparent, repeatable, and shareable. Using Notebooks is now an integral part of the data science workflow at companies worldwide. If you work with data, using a Notebook will accelerate your workflow and make it more reliable to discuss and share your results. As a client-server application, the Jupyter Notebook App allows you to edit and run your notebooks through a web browser. The application can be installed on a local machine without Internet access or can be deployed on a remote server, where you can access it via the Internet. Its core components are the kernels and a dashboard. A kernel is a program that runs and inspects the user's code. The Jupyter Notebook App has a kernel for Python code, but there are also kernels available for other programming languages. The dashboard of the application not only shows you the notebooks you've created and can resume but also can be used to manage the kernels: you can see which ones are running and shut them down if necessary.

Jupyter Notebook Features:

- Pluggable authentication: Manage users and authentication with PAM, OAuth, or integrate with your own identity management system.
- Integrated deployment: Deploy the Jupyter Notebook to large numbers of users in your organization on a centralized infrastructure on- or off-site.
- Container-friendly: Use Docker and Kubernetes to scale your deployment, isolate client processes, and work on software deployment.
- Live coding environments: Code can be edited and run in real-time with input provided directly in the application.
- Code meets data: Attach your Notebook to your data to provide collaborative software management and data access within your organization.

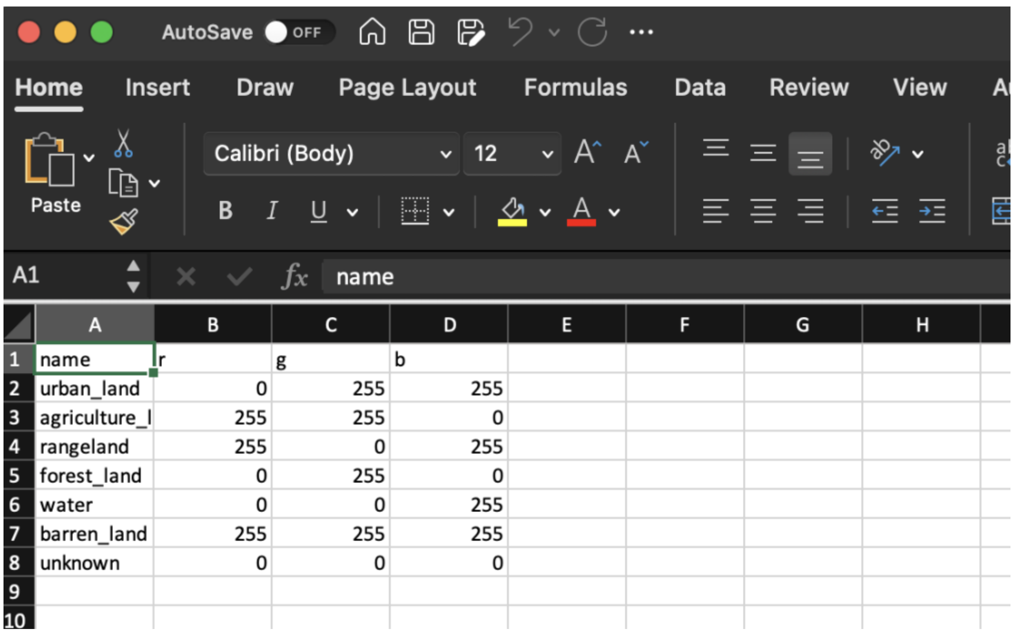
Tensor Flow

Profound learning is a subfield of artificial intelligence that uses algorithms inspired by the structure and function of the brain. It involves analyzing large amounts of unstructured data, such as images, videos, audio, and text, unlike traditional AI, which often uses structured data. TensorFlow, created by Google, is a popular AI framework used to design, build, and train deep learning models. TensorFlow is named after the operations that brain networks perform on multi-layered data arrays or tensors. Using tf.keras, you can design, fit, evaluate, and use deep learning models to make predictions in just a few lines of code. TensorFlow is also highly scalable and can run on CPU, GPU, or a cluster of these systems for training purposes. When working with large amounts of data, distributed computing is necessary, which is where TensorFlow comes in handy by running the code in a distributed way. GPUs, like those made by Nvidia, are popular for performing mathematical calculations, including network replication, which is critical in deep learning. TensorFlow also has integration with C++ and Python API for faster development. A tensor is a mathematical object represented as arrays of higher dimensions, and these arrays of data with different sizes and positions are handled as inputs to the neural network. Tensors can have more than three, four, or five layers, and they help keep the data organized and perform analysis around that data.

Code

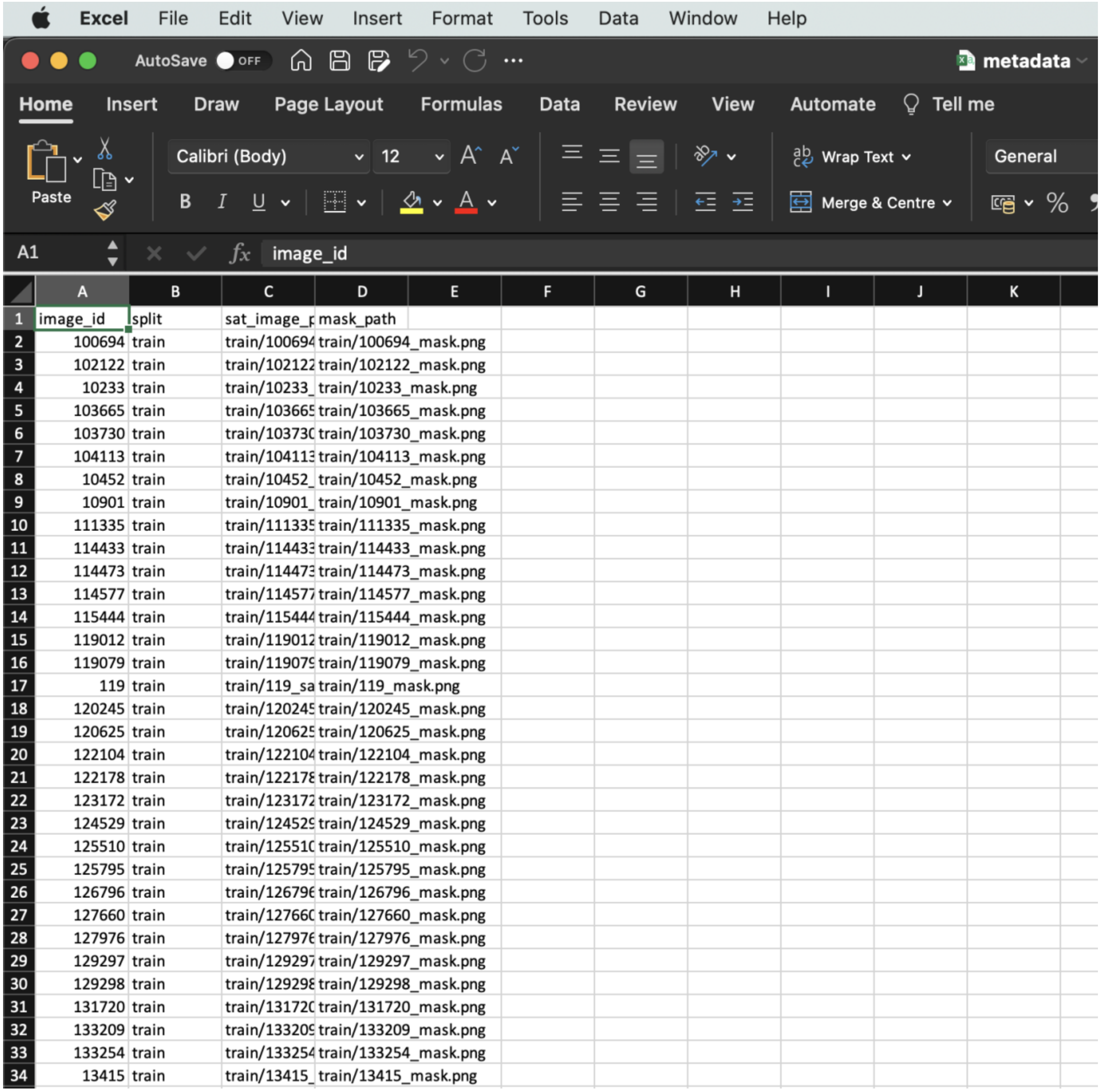
```
import warnings

warnings.filterwarnings("ignore")

import os

import cv2

import tqdm

import random

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

%matplotlib inline

import torch

import torch.nn as nn
```

Dataset CSV Files



	A	B	C	D
1	name	r	g	b
2	urban_land	0	255	255
3	agriculture_l	255	255	0
4	rangeland	255	0	255
5	forest_land	0	255	0
6	water	0	0	255
7	barren_land	255	255	255
8	unknown	0	0	0

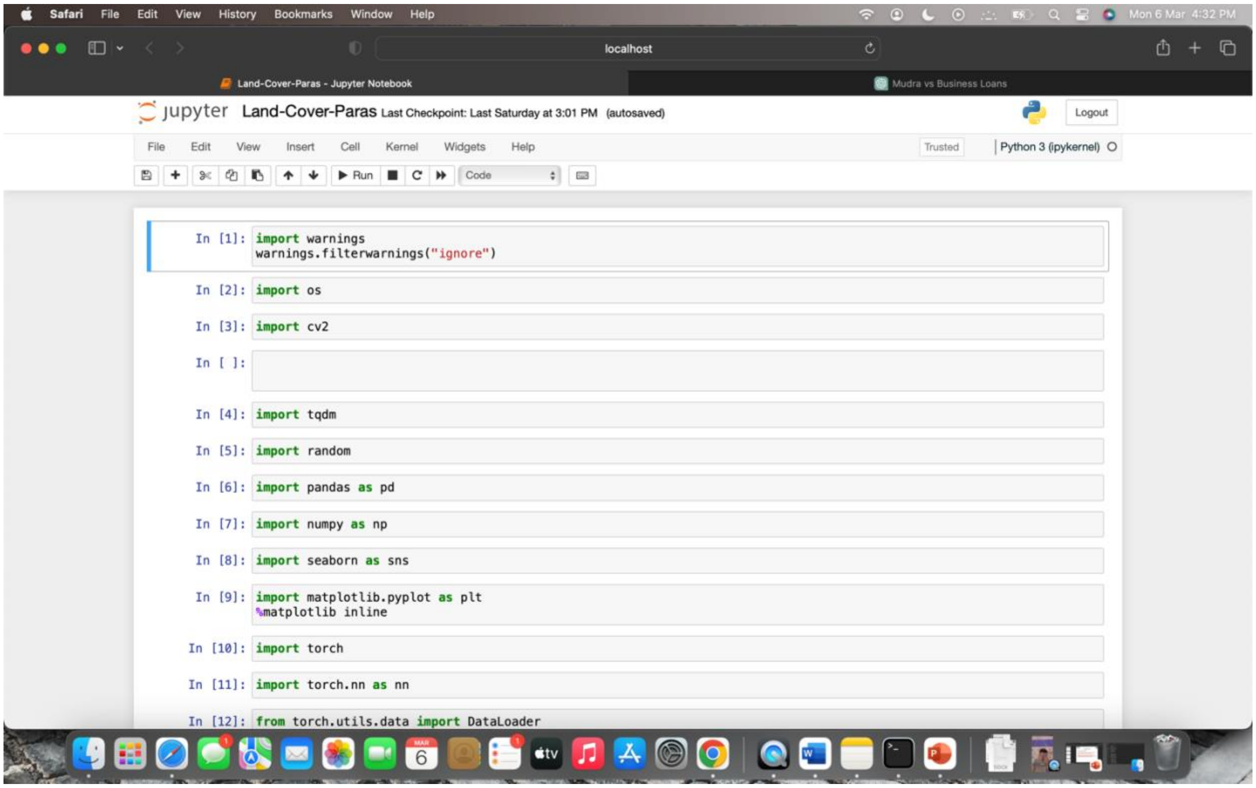
(iv.i)



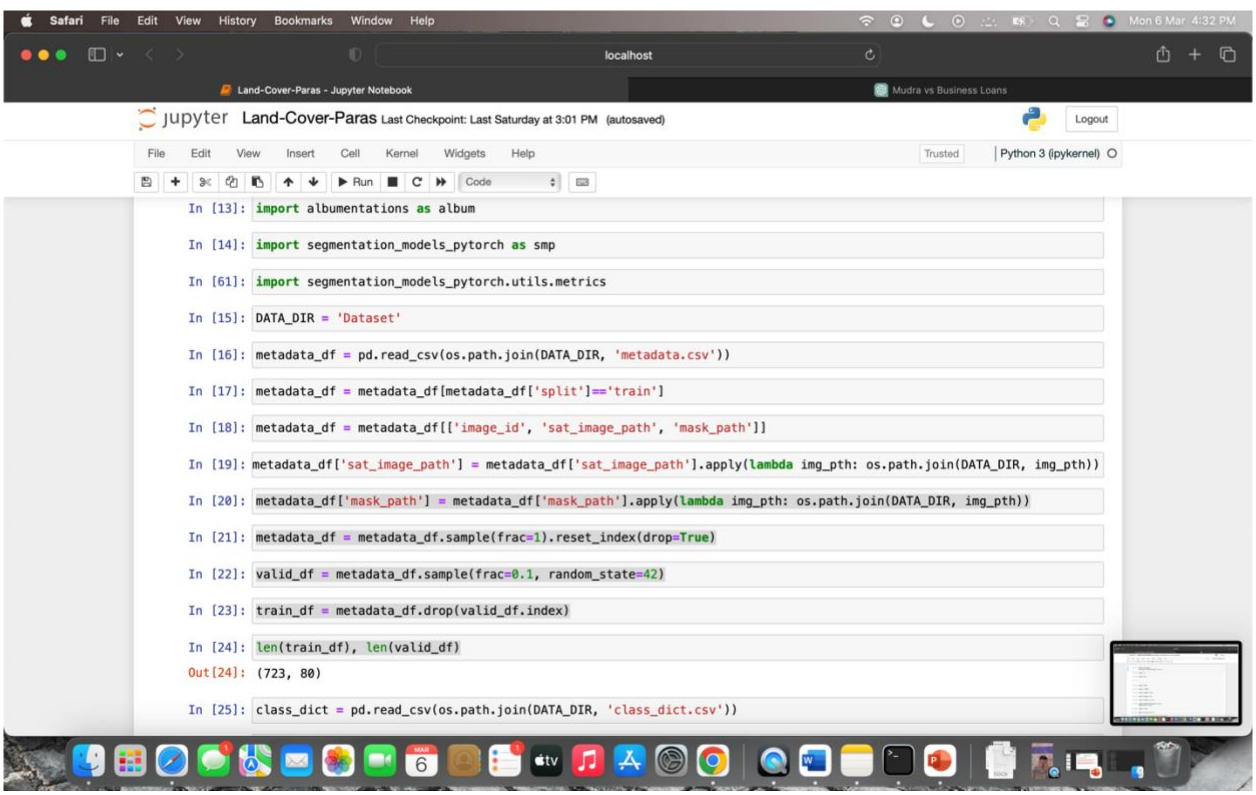
image_id	split	sat_image_p	mask_path
100694	train	train/100694	train/100694_mask.png
102122	train	train/102122	train/102122_mask.png
10233	train	train/10233_	train/10233_mask.png
103665	train	train/103665	train/103665_mask.png
103730	train	train/103730	train/103730_mask.png
104113	train	train/104113	train/104113_mask.png
10452	train	train/10452_	train/10452_mask.png
10901	train	train/10901_	train/10901_mask.png
111335	train	train/111335	train/111335_mask.png
114433	train	train/114433	train/114433_mask.png
114473	train	train/114473	train/114473_mask.png
114577	train	train/114577	train/114577_mask.png
115444	train	train/115444	train/115444_mask.png
119012	train	train/119012	train/119012_mask.png
119079	train	train/119079	train/119079_mask.png
119	train	train/119_sa	train/119_mask.png
120245	train	train/120245	train/120245_mask.png
120625	train	train/120625	train/120625_mask.png
122104	train	train/122104	train/122104_mask.png
122178	train	train/122178	train/122178_mask.png
123172	train	train/123172	train/123172_mask.png
124529	train	train/124529	train/124529_mask.png
125510	train	train/125510	train/125510_mask.png
125795	train	train/125795	train/125795_mask.png
126796	train	train/126796	train/126796_mask.png
127660	train	train/127660	train/127660_mask.png
127976	train	train/127976	train/127976_mask.png
129297	train	train/129297	train/129297_mask.png
129298	train	train/129298	train/129298_mask.png
131720	train	train/131720	train/131720_mask.png
133209	train	train/133209	train/133209_mask.png
133254	train	train/133254	train/133254_mask.png
13415	train	train/13415_	train/13415_mask.png

(iv.ii)

Code Implementation (v)



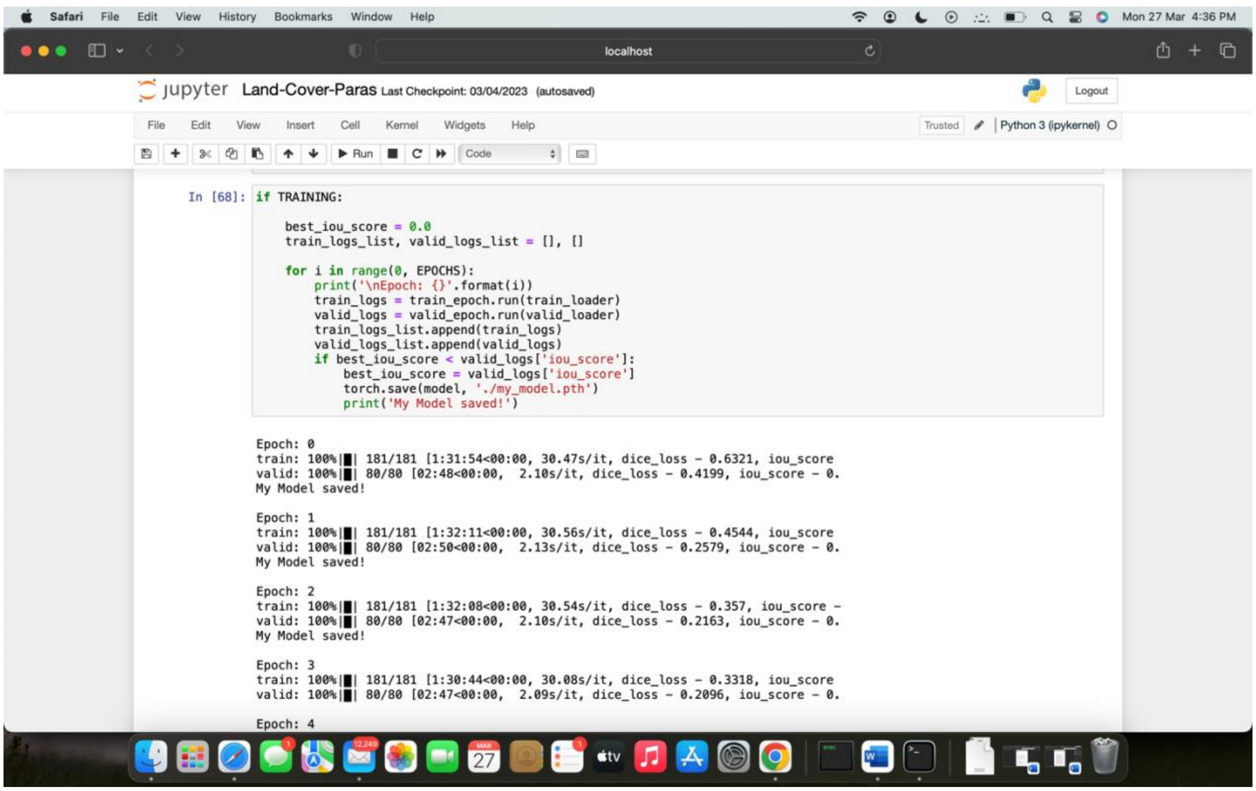
```
In [1]: import warnings
        warnings.filterwarnings("ignore")

In [2]: import os

In [3]: import cv2

In [ ]:

In [4]: import tqdm

In [5]: import random

In [6]: import pandas as pd

In [7]: import numpy as np

In [8]: import seaborn as sns

In [9]: import matplotlib.pyplot as plt
        %matplotlib inline

In [10]: import torch

In [11]: import torch.nn as nn

In [12]: from torch.utils.data import DataLoader
```

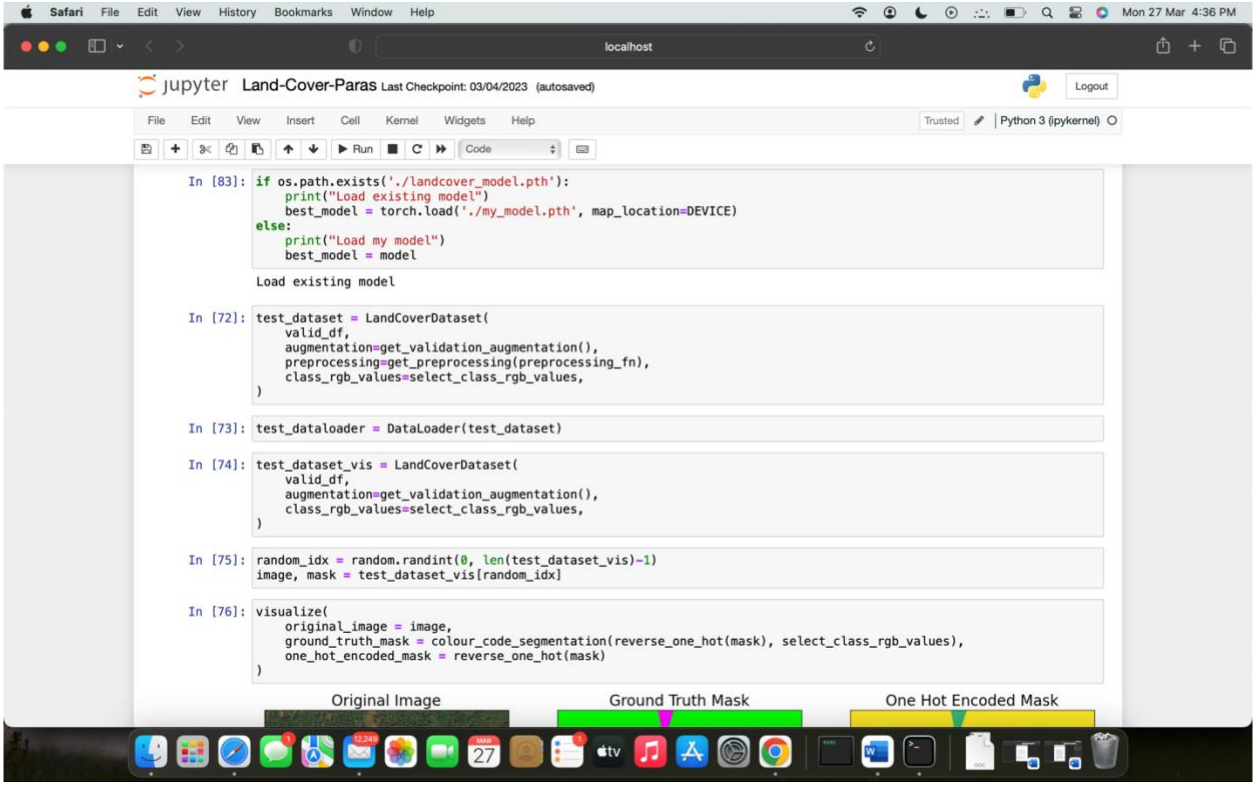


```
In [13]: import albumentations as album

In [14]: import segmentation_models_pytorch as smp

In [61]: import segmentation_models_pytorch.utils.metrics

In [15]: DATA_DIR = 'Dataset'

In [16]: metadata_df = pd.read_csv(os.path.join(DATA_DIR, 'metadata.csv'))

In [17]: metadata_df = metadata_df[metadata_df['split']=='train']

In [18]: metadata_df = metadata_df[['image_id', 'sat_image_path', 'mask_path']]

In [19]: metadata_df['sat_image_path'] = metadata_df['sat_image_path'].apply(lambda img_pth: os.path.join(DATA_DIR, img_pth))

In [20]: metadata_df['mask_path'] = metadata_df['mask_path'].apply(lambda img_pth: os.path.join(DATA_DIR, img_pth))

In [21]: metadata_df = metadata_df.sample(frac=1).reset_index(drop=True)

In [22]: valid_df = metadata_df.sample(frac=0.1, random_state=42)

In [23]: train_df = metadata_df.drop(valid_df.index)

In [24]: len(train_df), len(valid_df)
Out[24]: (723, 80)

In [25]: class_dict = pd.read_csv(os.path.join(DATA_DIR, 'class_dict.csv'))
```

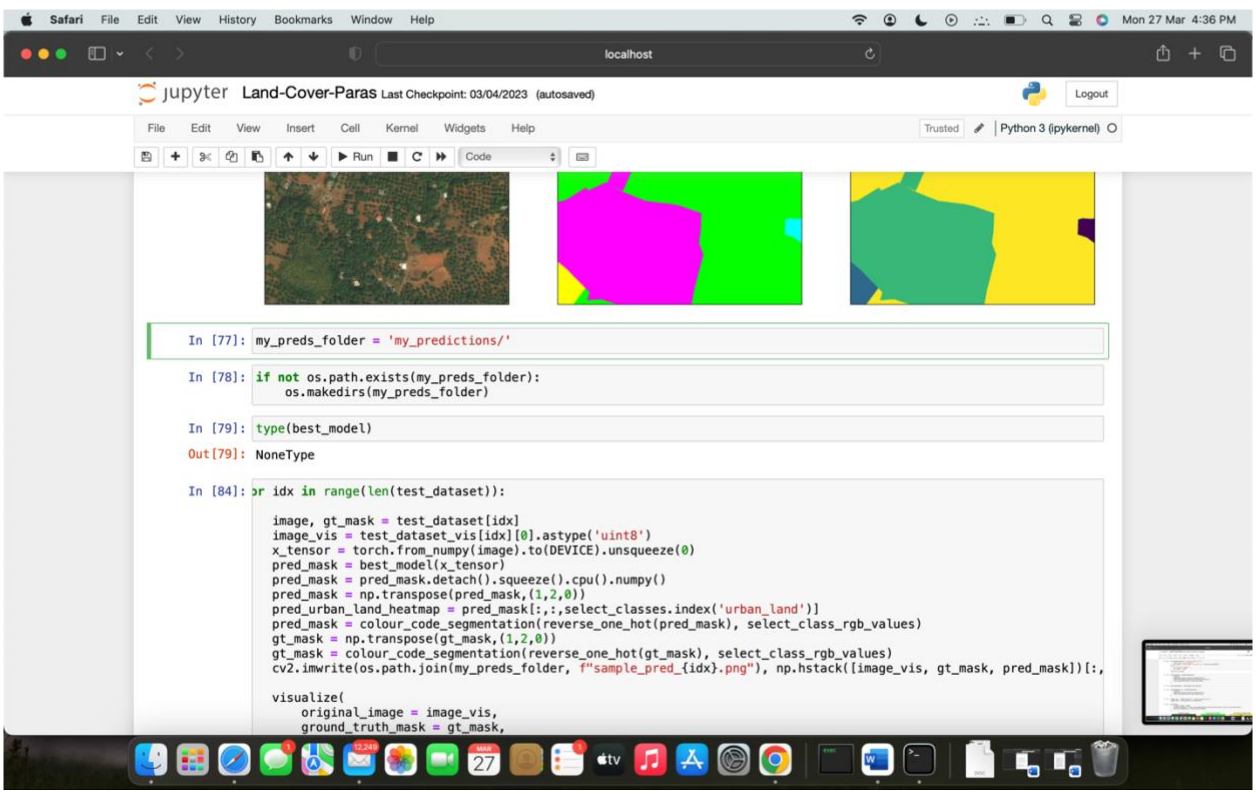
Expected Outcome

- Will Tolerate Variations
- Improve the operational efficiency.
- Improve the quality and consistency of data
- Can be adopted for better prediction in industrial applications.

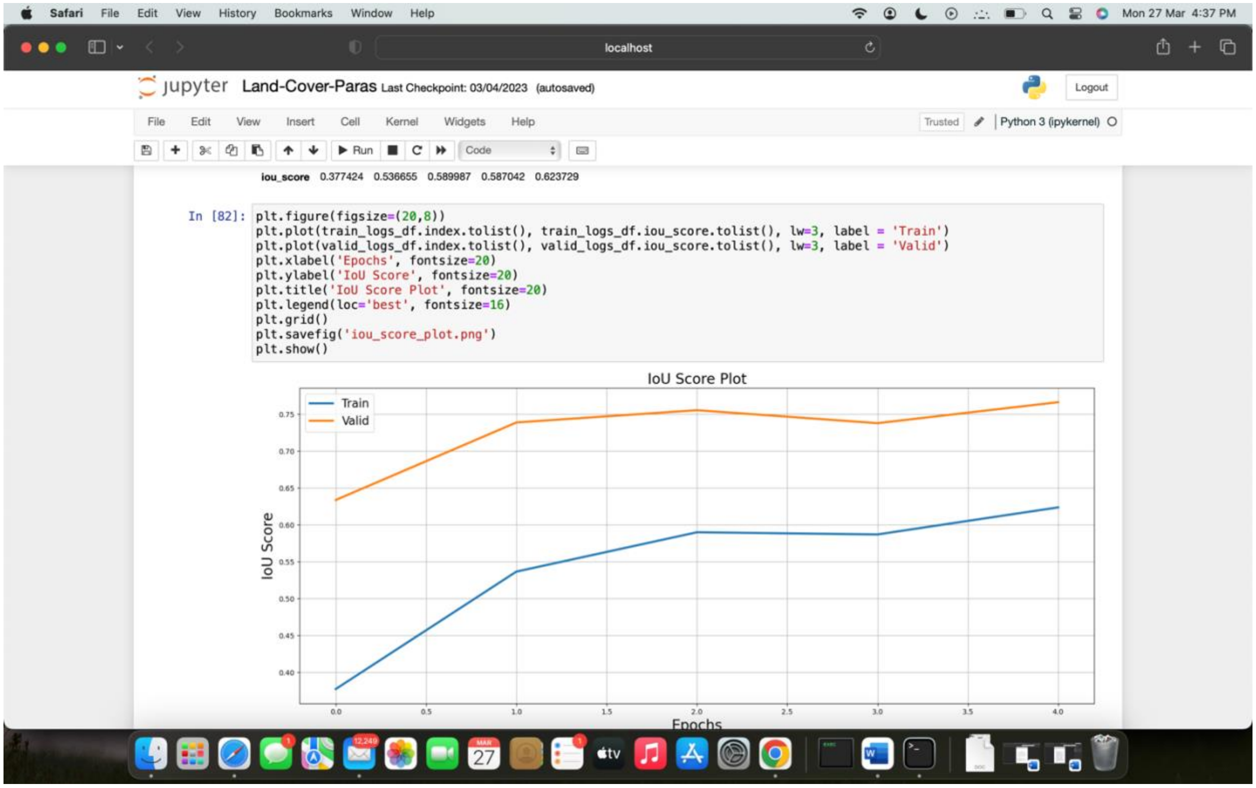




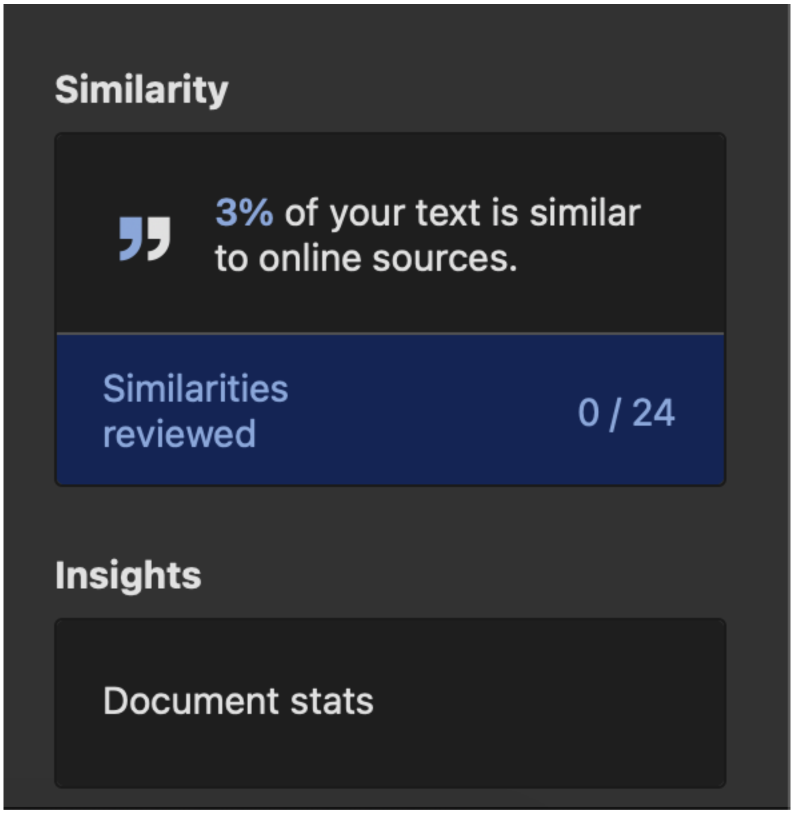
```
In [83]: if os.path.exists('./landcover_model.pth'):
             print("Load existing model")
             best_model = torch.load('./my_model.pth', map_location=DEVICE)
         else:
             print("Load my model")
             best_model = model

Load existing model

In [72]: test_dataset = LandCoverDataset(
             valid_df,
             augmentation=get_validation_augmentation(),
             preprocessing=get_preprocessing(preprocessing_fn),
             class_rgb_values=select_class_rgb_values,
         )

In [73]: test_dataloader = DataLoader(test_dataset)

In [74]: test_dataset_vis = LandCoverDataset(
             valid_df,
             augmentation=get_validation_augmentation(),
             class_rgb_values=select_class_rgb_values,
         )

In [75]: random_idx = random.randint(0, len(test_dataset_vis)-1)
         image, mask = test_dataset_vis[random_idx]

In [76]: visualize(
             original_image = image,
             ground_truth_mask = colour_code_segmentation(reverse_one_hot(mask), select_class_rgb_values),
             one_hot_encoded_mask = reverse_one_hot(mask)
         )
```

```
In [77]: my_preds_folder = 'my_predictions/'

In [78]: if not os.path.exists(my_preds_folder):
             os.makedirs(my_preds_folder)

In [79]: type(best_model)

Out[79]: NoneType

In [84]: or idx in range(len(test_dataset)):

           image, gt_mask = test_dataset[idx]
           image_vis = test_dataset_vis[idx][0].astype('uint8')
           x_tensor = torch.from_numpy(image).to(DEVICE).unsqueeze(0)
           pred_mask = best_model(x_tensor)
           pred_mask = pred_mask.detach().squeeze().cpu().numpy()
           pred_mask = np.transpose(pred_mask,(1,2,0))
           pred_urban_land_heatmap = pred_mask[:,:,select_classes.index('urban_land')]
           pred_mask = colour_code_segmentation(reverse_one_hot(pred_mask), select_class_rgb_values)
           gt_mask = np.transpose(gt_mask,(1,2,0))
           gt_mask = colour_code_segmentation(reverse_one_hot(gt_mask), select_class_rgb_values)
           cv2.imwrite(os.path.join(my_preds_folder, f"sample_pred_{idx}.png"), np.hstack([image_vis, gt_mask, pred_mask])[:,

           visualize(
               original_image = image_vis,
               ground_truth_mask = gt_mask,
```


Safari File Edit View History Bookmarks Window Help
Mon 27 Mar 4:37 PM
localhost
jupyter Land-Cover-Paras Last Checkpoint: 03/04/2023 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
iou_score 0.377424 0.536655 0.589987 0.587042 0.623729
In [82]:
plt.figure(figsize=(20,8))
plt.plot(train_logs_df.index.tolist(), train_logs_df.iou_score.tolist(), lw=3, label = 'Train')
plt.plot(valid_logs_df.index.tolist(), valid_logs_df.iou_score.tolist(), lw=3, label = 'Valid')
plt.xlabel('Epochs', fontsize=20)
plt.ylabel('IoU Score', fontsize=20)
plt.title('IoU Score Plot', fontsize=20)
plt.legend(loc='best', fontsize=16)
plt.grid()
plt.savefig('iou_score_plot.png')
plt.show()
IoU Score Plot
Train
Valid
IoU Score
Epochs

Similarity Report



Conclusion

The study aimed to test the impacts of different spatial resolutions on land use classification, based on the transitional effects of land cover changes. Initially, optical airborne imagery with natural spectral bands was used as input to generate land cover information using the Maximum Likelihood Classifier. Then, a Ownership package map was used as auxiliary data to improve the accuracy of land cover classification results. To avoid interference between different ground objects, data was extracted for each type of ground object separately. Additionally, the optimal scale division was found to be more effective for ground object extraction of all sizes. The results indicate that this approach yields optimal classification results with improved accuracy.

Benefits

- Consistent approach to parameter estimation problems.
- Have really great mathematical and optimality properties.
- Helps mitigate the computational complexity of maximum likelihood estimation.

Future Work

One possible approach to this problem could be to use quantum algorithms to analyze satellite data and reconstruct land cover patterns. Quantum algorithms can potentially provide significant speedup over classical algorithms, which can be particularly useful when dealing with large datasets. So in future I would like to work on Land Cover Reconstruction using Quantum Computing.

References

1. J.-S. Lee and E. Pottier, Polarimetric Radar Imaging: From Basics to Application. Boca Raton, FL, USA: CRC Press, 2009.
2. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, arXiv:1409.1556. [Online]. Available: http://arxiv.org/abs/1409.1556
3. S.-W. Chen and C.-S. Tao, "PolSAR image classication using Polarimetric-Feature-Driven deep convolutional neural network," IEEE Geosci. Remote Sens. Lett., vol. 15, no. 4, pp. 627631, Apr. 2018.
4. W. Yang, T. Zou, D. Dai, and Y. Shuai, "Supervised land-cover clas- sication of TerraSAR-X imagery over urban areas using extremely randomized clustering forests," in Proc. Joint Urban Remote Sens. Event, May 2009 ,pp. 16.
5. F. Changyong et al., "Log-transformation and its implications for data analysis," Shanghai Arch. Psychiatry, vol. 26, no. 2, p. 105, 2014.
6. K. J. Wallace, "Classication of ecosystem services: Problems and solu- tions," Biol. Conservation, vol. 139, pp.235246, Oct. 2007. Di Gregorio, Land Cover Classication System, Classication Con- cepts, Software Version 3. Rome, Italy:
7. Food and Agriculture Organization of the United Nations, 2016. G. P. Asner et al., "High-resolution forest carbon stocks and emissions in the Amazon," Proc. Nat. Acad. Sci., vol. 107, pp. 1673816742, Sep. 2010.Baccini et al., "Estimated carbon dioxide emissions from tropical deforestation improved by carbondensity maps," Nature Climate Change, vol. 2, pp. 182185, 2012.
8. GFOI, Integrating Remote-Sensing and Ground-Based Observations for Estimation of Emissions and Removals of Greenhouse Gases in Forests: Methods and Guidance from the Global Forest Observations Initiative. Geneva, Switzerland: Group on Earth Observations, 2014.
9. BFD, National Forest and Tree Resources Assessment 20052007 Bangladesh. Dhaka, Bangladesh: Bangladesh Forest Department, Ministry of Environment and Forest (MoEF), Food and Agriculture Organization of the United Nations (FAO), 2007. Di Gregorio, Land Cover Classication System, User Manual, Software Version 3. UN-REDD Programme.
10. Rome, Italy: Food and Agriculture Organization of the United Nations, 2016.
11. L. Costello and M. Piazza, Proceedings from the Training Survey Design and Data Management Using Open Foris Collect for NFI and Carbon Stock Assessment in Bangladesh. Dhaka, Bangladesh: Food and Agricul- tural Organization of the United Nations, Nov. 2015. Di Gregorio, Recommendations on the Land and Forest Classication System of Bangladesh, Training Workshop on Land Cover Classication in the context of REDD+ in Bangladesh. UN-REDD Programme. Dhaka, Bangladesh: Food and Agriculture Organization of the United Nations, 2013. Hadi, T. Udita, and R. Jalal, Proceedings of the National Consultation Conference on Land Cover
12. Classication System in Bangladesh. Dhaka, Bangladesh: FAO of the United National and Bangladesh Forest Department, Mar. 2016.
13. G. Franceschini et al., Production Chain for Land Cover Mapping in Bangladesh. Dhaka, Bangladesh: Food and Agriculture Organization of the United Nations, 2016.

14. K. Z. Tasnim, G. Franceschini, and R. Jalal, Proceedings of the Workshop on the Accuracy Assessment and Quality Checking of Land Cover Map of Bangladesh 2015. Dhaka, Bangladesh: FAO of the United Nations and Bangladesh Forest Department, 2017.
15. 12. M. Costallo and J. Piezza, Proceedings from the Training Survey data and Management data services Using Open Foris Collection for NFI and Carbon Stock Assessment in Bangladesh. Dhaka, Bangladesh: Food and Agricul- tural Organization of the United Nations, Nov. 2015.
16. B. Ji Gragorio, configuration of the Forest and land Classication System, Training Workshop on Land Use Classication for the usage of RADD+ in. UA-RADD Programme: healthcare and natural Organization of the foreign land, 2018.
17. B. Yadi, F. Ldita, and C. Galal, Convincing of the International ask for journal on Land use Classication, Frequent ask of countries, July. 2015.
18. M. Francaschine es ap., Land use coverage production chain : healthcare and Natural Organization of the United Nations, 2016.
19. A. J. Hasnim, I. Francaschine, and M. Halal, P Accurate Examination and Quality verification of Land Use. 2019.
20. U. wlwfsson, K. P. Roody, A. B. Stahman, and K. M. Waadcock, "Changing of land services can be more accurate: Calculating volume certainty using stratied estimation," pp. 122131, Feb. 2013.