Satellite Imagery Land use classification using Machine Learning

CS39440 Major Project Report

Author: Abdullah Durrani (abd15@aber.ac.uk)
Supervisor: Dr/Prof. Tossapon Boongoen (tob45@aber.ac.uk)

23rd April 2024
Version: 1.0 (Draft)

This report was submitted as partial fulfilment of a BSc degree in Artificial Intelligence and Robotics (GH76)

Department of Computer Science
Aberystwyth University
Aberystwyth
Ceredigion
SY23 3DB
Wales, U.K.

Declaration of originality

I confirm that:

- This submission is my own work, except where clearly indicated.
- I understand that there are severe penalties for Unacceptable Academic Practice, which can lead to loss of marks or even the withholding of a degree.
- I have read the regulations on Unacceptable Academic Practice from the University's Academic Registry (AR) and the relevant sections of the current Student Handbook of the Department of Computer Science.
- In submitting this work I understand and agree to abide by the University's regulations governing these issues.

Name Abdullah Muhammad Khan Durrani

Date 23/04/24

Consent to share this work

By including my name below, I hereby agree to this project's report and technical work being made available to other students and academic staff of the Aberystwyth Computer Science Department.

Name Abdullah Muhammad Khan Durrani

Date 23/04/24

Acknowledgements

I would like to thank my supervisor Prof. Tossapon Boongoen for all his help in supporting me through this project I could not have gotten to here writing this report without Your encorigment to always do my best.

Abstract

This report states the process utilized in developing a python application that employs the K-means clustering algorithm for the classification of satellite imagery. The primary objective of this project was to simplify and enhance the usage of satellite data across various sectors by providing a user-friendly tool for image analysis.

The application incorporates a single classifier which can be expanded to include other unsupervised classifiers as well as supervised ones , but the K-means algorithm, which is renowned for its efficiency in segmenting images into clusters based on pixel similarity. This method is particularly advantageous for categorizing land use and identifying patterns or changes in satellite images.

Research was conducted to ensure the optimal implementation of the K-means algorithm, with studies indicating its effectiveness in handling large datasets and its robustness in producing significant clusters that are meaningful in the context of satellite imagery.

Designed to be intuitive, the application allows users of varying technical expertise to engage with satellite data analysis. The system facilitates real-time processing of images, providing immediate feedback and results of the effectiveness of the Kmeans implementation, and allows for the user to compare images overtime visually highlighting differences. which are crucial for timely decision-making in areas such as environmental monitoring and urban planning among other areas.

Contents

List of Figures

List of Tables

Chapter 1

Background & Objectives

1.1 Introduction

In the world of Satellite Imagery analysis, the tools needed to provide robust and efficient classification and unlock the full potential of satellite imagery across various disciplines such as environmental monitoring, disaster response and urban planning is crucial. This project was initiated by recognizing this need for an application that leverages machine learning to simplify and enhance the classification of satellite imagery

The Selection of the K-means clustering algorithm for this task was for several reasons. Its simplicity, effectiveness, and arguably the most important that it is unsupervised as opposed to the CAST (Decision trees), Random forrest and Artificial Neural Network approaches which are supervised learning methods. The K-means classifier allows the segmentation of large datasets of satellite imagery into meaningful clusters based on Pixel similarity. This approach helps in categorizing Land use and Specifically in detecting changes over time in regard to land use as we will see.

1.2 Background

The main idea of this project centers on the advanced utilization of satellite imagery for practical real world issues [1]. Satellite imagery provides a view of the earth's surface that most individuals don't usually look at, it provides critical data for environmental monitoring, urban development, strategic planning, land use control, security issues, and many more useful and important applications. The application of machine learning algorithms to satellite imagery enhances the ability to categorize and analyze the data more efficiently and in a less time-consuming way.

1.2.1 Problem Overview

The increase in satellite imagery presents both an opportunity and a challenge. With satellites continuously orbiting and capturing visuals of earth's surface, there exists a vast collection of images that contain data which can enhance the decision taken in certain fields across multiple sectors. However, the volume of this data makes it difficult to process and analyze efficiently with conventional methods. Which may be inaccurate and or incredibly slow and can be influenced to ignore certain information based on biases and or pressure to complete the work on time. This can cause issues for time-sensitive matters such as environmental changes, disaster response and or Urgent urban developmental needs.

Thus, the problem requires advanced tools that can handle the large amount of data taken from satellite imagery and convert it into usable insights swiftly and accurately. Traditional Image processing methods often fall short in managing the spatial and spectral diversity found in satellite imagery. Other Machine Learning Methods, specifically those most used for land use classification, require to be trained on data which takes time to collect and is expensive to store depending on how much data is collected. Not to mention, the training itself will take time that you might not have in an Emergency situation [2], and the model will be only as good as the training data used to train it.

Because of these Issues, The development of an application that uses unsupervised algorithms for satellite imagery classification addresses these challenges. By using an unsupervised approach to classification, it saves time on training and allows the user to segment the image by clusters which represent different land uses and makes it easier to see difference over time in land use in the area. It aims to enhance the accessibility and utility of satellite imagery. making it actionable for experts and informative for the layman.

1.2.2 Problem analysis

To address the issues of satellite imagery analysis, A solution must include several key features. these features would aim to ensure that the system not only meets the immediate needs of various users but also adapt to evolving technological changes. These include the following:

- **Real-Time Processing:** *The ability to process data and analyze it swiftly is crucial. Developing systems that offer near realtime processing capabilities ensure actionable insights when*
- **sufficient in time and cost** *Should be quick and easily used on standard computer hardware. It should minimize setup times and make efficient use of hardware resources to ensure cost-effectiveness and accessibility to a broader range of users*
- **Accessibility and Usability:** *Should be able to be used and be understood by experts and laymen. this means a Simple-to-use user interface with simple to understand Image Analysis that is actionable to the users specific needs*

- **Accuracy:** *The program needs to be accurate to a certain degree to be actionable.To achieve it needs to be able to give metrics on its performance after completion so that the user can make informed decisions about the results*
- **Difference Map:** *The program should be able to compare the area to a previous version of the area and to tell the user visually how much change has occurred*
- **Unsupervised Machine learning Method:** *The program should use an unsupervised machine learning method*
- **Continuous Improvement and Learning:** *Since this field is ever evolving the system must be designed for easy maintenance and updates. The code should be clear and well-documented, facilitating the integration of new features and the adaptation of the system to incorporate the latest research findings and technological improvements.*

1.3 Project Analysis

This project aims to develop a sophisticated tool for analyzing satellite imagery, leveraging the capabilities of unsupervised machine learning technologies. Opting for an unsupervised machine learning approach provides significant advantages over supervised methods, particularly in scenarios where labeled data is scarce or costly to obtain.

Supervised methods require extensive labeled datasets for training, which can be both time-consuming and expensive to prepare, especially for complex image datasets like satellite imagery. In contrast, unsupervised machine learning automates the analysis without the need for labeled data, significantly enhancing speed and reducing costs. By integrating advanced unsupervised algorithms, the system can process vast amounts of imagery data, identifying inherent patterns and changes more quickly and accurately than supervised methods.

This automation not only accelerates the analytical process but also reduces reliance on potentially biased or inconsistent manual labeling. Furthermore, unsupervised systems can continuously adapt and improve as they are exposed to new data, learning from the inherent structure and variations without the need for continuous retraining with new labeled data.

Moreover, leveraging unsupervised machine learning reduces the overhead associated with maintaining and updating large labeled datasets, offering a unified approach that can be easily scaled and adapted to various needs without significant modifications. This adaptability is crucial for dynamic applications like environmental monitoring and urban planning [3], where conditions can evolve rapidly and unpredictably.

Ultimately, this scalability and cost-effectiveness make advanced satellite image analysis more accessible to a broader range of users, from experts to laymen in related fields, enhancing the potential impact of the technology across multiple sectors.

1.3.1 Supervised vs. Unsupervised

As stated in the previous few sections, an unsupervised approach would be better than a supervised approach for this project here is why:

- **Elimination of Labeling Requirements:** *One of the most significant challenges with supervised learning is the necessity for labeled datasets, which are used to train the models. Labeling satellite images can be labor-intensive and costly, requiring expert knowledge and a considerable amount of time. An unsupervised approach, however, does not require labeled data. It automatically detects patterns and anomalies in the data based on the inherent characteristics of the images, such as pixel values and textures.*
- **Adaptability to New Data:** *Satellite imagery is continuously evolving due to changes on the Earth's surface and differences in data acquisition techniques. Supervised models often need retraining or fine-tuning when new types of data are introduced, which can be a cumbersome and resource-intensive process. Unsupervised learning methods are inherently more flexible, adapting to new patterns and changes in the data without the need for retraining. This adaptability makes unsupervised learning particularly suitable for the dynamic nature of satellite imagery,*
- **Scalability and Cost-Effectiveness:** *Handling the vast amounts of data generated by satellite technologies can be more scalable under an unsupervised framework. Since there is no need to label new data, continually*
- **Discovery of Unknown Patterns,** *Unsupervised learning is particularly adept at identifying patterns and clusters that were not previously anticipated. In satellite imagery, this capability is crucial as it can reveal unexpected changes or new phenomena that a supervised model, trained on previously known labels, might overlook.*
- **Reduced Bias and Greater Objectivity:** *Since Unsupervised Learning does not rely on Labels it is not susceptible to biases that might appear in the training dataset of a Supervised learning method*

All of these factors played a role in why the Project uses an unsupervised machine learning algorithm, specifically the previously mentioned K-means clustering algorithm. As it not only optimizes resource utilization but also enhances the analytical capabilities of the system.

1.3.2 Choice of Machine learning Algorithm

In selecting The appropriate machine learning algorithm for the classification of satellite imagery in our project, the K-means clustering algorithm from the sklearn library was chosen due to its specific characteristics that align well with the requirements of the project, Specifically:

- **Simplicity:** *K-means is one of the simplest clustering algorithms to implement and understand. it's computationally efficient as well especially when dealing with large datasets, like those typically generated by satallite imagery.*
- **Good metrics for calculating Accuracy of the Clusters:** *Silhouette Score, Davies Bouldin Index, and Inertia provide valuable insights into the clustering performance, but they measure different aspects. The Silhouette Score assesses how appropriately data points have been grouped compared to other clusters -1 being it's in the wrong cluster and 1 being this is definitely the correct cluster, which reflects separation and cohesion, while Inertia focuses on the compactness of the clusters. A high number is not very compact meaning the cluster is not very uniform and a lower number meaning it's a compact cluster. Employing both metrics together allows for balancing cluster quality through Inertia and relevance through the Silhouette Score, facilitating more accurate and reliable cluster analysis. Additionally, the Davies-Bouldin Index (DBI) enhances this evaluation by measuring the ratio of intra-cluster distances to inter-cluster distances. A lower DBI value indicates better clustering as it signifies that the clusters are both compact and well-separated from each other. Integrating DBI into the assessment provides a comprehensive view of cluster validity, For the Specific image.*
- **Well-defined Clusters** *K-means works best with well seperated data and can produce very tight clusters which are very useful in our project when we need to delineate different types of land use from each other*
- **Robustness and consistency:** *K-means often produces consistent results, making it reliable for repeat analyzes under similar conditions. This predictability is valuable in applications requiring repeated deployment, like monitoring changes in environmental landscapes over time through satellite images.*

1.3.3 Primary Objective

The primary objective of the project is to develop an advanced application capable of efficiently processing and analyzing satellite imagery using a machine learning algorithm. This tool aims to transform satellite data into actionable insights swiftly and accurately, enhancing decision-making in various sectors. By utilizing machine learning techniques the application seeks to outperform non-machine learning techniques by offering greater precision and faster processing times, thereby providing rapid and cost-effective image analysis. This will make satellite imagery more accessible and useful to both experts and laymen, helping them to understand and react to changes in the landscape efficiently and effectively for the users specific uses.

1.4 Process

1.4.1 Methodology

In the course of this project, Multiple Methodologies were considered, particularly agile methodologies. Like Scrum, Extreme Programming, and Kanban, as well as plan driven ones like the Waterfall approach. In the beginning of the project, the idea was to go for a plan driven Waterfall method.

Overtime however, FDD or Feature Driven Development was fully chosen as the methodology for its specific advantages in managing software development projects through iterative and incremental progress, Unlike the Waterfall method which for a solo developer puts too much strain in the beginning on rigid planning, making problems later on when designs or features have to change for unspecified reasons.

1.4.2 Feature Driven Development (FDD) Overview

Feature Driven Development is a client-centric, architecture driven and pragmatic software development Methodology. Its main focus is to deliver working software in a timely manner consistently. FDD is usually used in a collaborative setting. However, even as this was a solo project, it was straightforward to adapt it to being a single developer scenario unlike waterfall which is much less able to be adapted to a solo methodology

FDD compartmentalizes the planning into Features/functions of a whole overall model of the program which serves as a blueprint of the program. Each feature is then given a priority in the plan, and in each iteration of the Program you plan design and code that feature and then move on to the next. and by the time you finish, you have a fully implemented completely working program ready for the user to use and enjoy.

1.4.3 Development environment, Libraries, and Programing language

For this project, it was fully developed in the anaconda development environment. As it ensures that the program will be Cross platform, has a wide array of machine learning and non-machine learning libraries which were essential for this project such as

- **numpy: 1.23.5**
- **matplotlib: 3.80**
- **scikit-Learn: 1.30**
- **cv2: 4.60**
- **Pillow: 10.2.0**
- **tk: 8.6.12**
- **math**

It was also used as it supports python 3.11.5, which is the version of python this program was developed on. Plus, there was already prior experience for this environment as it was used for a previous python module. Although a new Fresh anaconda environment was created for this project instead of the old one being reused.

1.4.4 Version Control System

For this project, GitHub was used as the version control system. As GitHub is widely recognized for its robust functionality in managing changes to code, collaborating on a project, storing code securely over time, Documentation support, and Integration with other tools. Such tools as The IDE Pycharm on which this program was developed

1.4.5 Documentation of diagrams

The simple UML diagrams used in this report have been created using plantUML scripts from [4] and the line graphs were generated using code

Chapter 2

Experimentation

2.1 K-Means clustering

This section is about how the Kmeans clustering algorithm works mathematically. As it is important that we learn how the Classifier functions so that we can understand how Clustering occurs, what the processes are, and how we can manipulate it to get the best possible results. The steps are as follows:

- **Initialization:** *The first step in k means clustering involves selecting K initial centroids, where k is the number of clusters you want to identify in the dataset. The centroids can be randomly selected from the datapoints, or you can provide a heuristic or specific strategy on which they are chosen to enhance the quality and or speed of convergence.*
- **Assignment step** *In this step each data point in the dataset is assigned to the nearest centroid. the nearest is determined by the euclidian distance between the data point and the centroid this is mathematically determined by*

$$C_i = \{x_p : \|x_p - m_i\| \leq \|x_p - m_j\| \, \forall j, 1 \leq j \leq k\}$$

where C_i is the set of data points assigned to cluster i, x_p is a data point, and m_i is the centroid of cluster i. This means each data point x_p is assigned to cluster i if the distance from x_p to m_i is the smallest among all centroids.

- **Update Step** *Once all data points have been assigned to clusters, the centroids need to be recalculated. This is done by taking the mean of all points assigned to each cluster—the position of the centroid of each cluster is updated to the mean position of all points belonging to that cluster. The formula for updating the centroid of each cluster is:*

$$m_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$$

where $|C_i|$ is the number of data points in cluster i, and $\sum_{x \in C_i}$ is the sum of all data points in cluste i

- **Iteration** *In this step steps 2 and 3 are repeqted iterativly until the centroilds stop moving significantly, this means that the cluster is now stabilized and the algorithm has converged, the number of iterations has been spent. This allows the cluster assignmenst and centeroid positions to reflect the data accurately.*
- **Convergence** *The algorithm has converged when the centroids have stabilized and or alternatively when the assignment of points to clusters longer change*

2.2 Metrics Used

When Employing Clustering algorithms such as K-means, assessing the quality of the algorithm is important as it validates the effectiveness of the analysis. Because of this, I use three crucial metrics to evaluate the K-means algorithm Silhouette score, Inertia and DBI(Davies-Bouldin Index)

2.2.1 Silhouette Score

The Silhouette Score is a measure of how similar an object is to its own cluster compared to other clusters. The score is calculated for each data point and can range from -1 to +1, where a high value indicates that the object is well-matched to its own cluster and poorly matched to neighboring clusters Mathematically its definition is:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

$a(i)$ is the mean distance between i and all other data points in the same cluster.

This measures how well it i is assigned to its cluster (the smaller, the better).

$b(i)$ is the minimum mean distance from ii to all points in any other cluster, of which i is not a member.

This measures how poorly i is matched to its neighboring cluster (the larger, the better).

The overall Silhouette Score for the dataset is the mean Silhouette Score of all individual points. This score provides a succinct measurement of how appropriately the data has been clustered.

2.2.2 Inertia

Inertia, also known as the within-cluster sum of squares, measures the compactness of the clusters, which ideally should be as small as possible . It is calculated by summing the squared distances between each data point and its nearest centroid. Mathematically its definition is:

$$W(C) = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2$$

C_i is the set of all points in cluster i,

mu_i is the centeroid of cluster i

Outer Sum: $\sum i = 1^k$ iterates over each cluster from 1 to k.

Inner Sum: $\sum x \in C_i$ sums over all points x within each cluster C_i.

$\|x - \mu_i\|^2$ computes the squared Euclidean distance between a point x and the cluster centroid mu_i, which is the norm squared of the vector difference.

k is the number of clusters

2.2.3 Davies-Bouldin Index (DBI)

The DBI is Defined as the Average similarity measure of each cluster with its most similar cluster, where similarity is the ration of within cluster distances to between cluster differences. The Goal is to minimize the DBI, as a lower DBI score indicates a better clustering division. It has no dependency on the Shape or Density of the Cluster, It is easy to compute, and it is a Simple interpretation as it directly quantifies the trade-off between the compactness of clusters and their separation. The Mathematical definition for it is:

$$DBI = \frac{1}{k} \sum_{i=1}^{k} \max_{i \neq j} R(i, j)$$

where k is the number of clusters, and $R(i, j)$ is the similarity measure between clusters i and j where $R(i, j)$ is defined as $R(i, j) = \frac{s_i + s_i}{d_{ij}}$

s_i is the average distance to all points in cluster i to the centroid of cluster i (intra-cluster distance)

d_{ij} is the distance between the centroids of clusters i and j (inter-cluster distance)

s_j is similarly the average intra-cluster distance for cluster j.

2.3 Analysis of Clustering Metrics: Determining the Optimal Number of Clusters for the Al Dhannah Dataset

The selection of the optimal number of clusters (k) in K-means clustering is crucial for achieving the best possible grouping of data points. We can analyze this by reviewing graphs of the Average Inertia for Different k, Average Silhouette Scores for Different k, and a combined plot of Silhouette Score vs.Inertia so that we can find the best K for the Al Dhannah City dataset as each dataset will have a different best K.

2.3.1 Average Inertia for different K

As illustrated in FigureB.1, As we can see from the Graph, there is a sharp decline in inertia as the number of Clusters increases from 2 to around 5 followed by a more gradual decrease. The Lower the Inertia value, the more Compact the Clusters meaning higher quality clusters.

We can apply the elbow rule here which states that the optimal K is where the inertia curve begins to flatten in this graph the curve begins to flatten around K=4. this suggests that increasing the number of clusters beyond this point results in Diminishing results

2.3.2 Averge Silhouette Scores for Different K

As illustrated in FigureB.2, The Average Silhouette Score graph presents a different perspective, highlighting the average silhouette score of clusters as k varies. The silhouette score measures how similar an object is to its own cluster compared to other clusters, with higher values generally indicating more appropriate clustering.

The graph shows the highest silhouette score at k=2, which then declines steadily as more clusters are added. This indicates that at k=2, the clusters are more distinct and well-separated compared to higher values of k.

2.3.3 Silhouette Score vs. Inertia

As illustrated in FigureB.3, The combined graph Plots these scores against each other for the value of k this visual representation helps us see the trade-off between Inertia and the Silhouette score. If K is high in this dataset, inertia is low, so K = 2 is too much in favor of the Silhouette Score, the Clusters are separate, but the clusters are not as dense as they should be. K = 4, on the other hand, is perfect as The silhouette Score is not as good as K = 3 and the inertia is not as good as K = 5.

Using insights from Both previous graphs. We can determine that K = 4 is the most balanced choice for the Current Dataset as it provides a compromise between having distinct clusters and ensuring that the clusters are compact enough [5]. This is suggested by the elbow in the inertia graph and relatively good silhouette scores for K = 4 in the silhouette score graph.

Chapter 3

Design and Implementation

As we are following the Feature Driven Development methodology which was discussed in section 1.4.2 the Design, Implementation, and Testing Chapters of the report have been merged For each iteration of the Project a section has been created to report on the progress made for each iteration

3.1 Iteration 0

This iteration was spent researching and creating a design prototype as well as figuring out the feature list of the application. This iteration was also spent researching where to collect data

3.1.1 Initial Design

The initial design of the Satellite Image Classification System was conceived to provide a robust, user-friendly platform for the analysis and classification of satellite imagery using machine learning techniques. The design focused on creating a modular, scalable, and intuitive application that could be easily adapted to meet the diverse needs of users ranging from environmental scientists to urban planners.

The Initial design would be object orientated, and did not currently have a specific Machine learning Model in mind for the application there wer 4 different models in line for the spot Random Forrest, CAST(Decision tree), CNNs and K-means clustering Initially the Criteria for Model selection was Performance, Scalability and Accuracy.

Below is an initial Architecture for what the Program should have been able to do it consists of 3 Layers A User interface Layer, Data processing layer and a machine learning Layer

- **User interface Layer**

- – **Functionality** *The UI layer, provides the primary interaction point for users. It handles tasks like loading images, initiating the classification process, displaying results, and enabling comparative analysis of images.*
  – **Technology** *It utilizes the Tkinter library, a standard GUI toolkit in Python, which supports building desktop applications with graphical elements like buttons, canvases, and dialog boxes.*
- **Data Processing Layer**
- – **Functionality** *This layer, is responsible for preparing the image data for analysis. This includes preprocessing tasks such as resizing, and normalizing the pixel data of the image, and making sure the images are in the optimal format for clustering.*
  – **Output** *It outputs the Image as Usable data for the machine Learning layer*
- **Machine Learning layer**
- – **Functionality** *he core analytical capabilities of the system are handled by the machine learnning algorithm to the processed image data, classifing the images into distinct land uses based on their features.*
  – **Features** *It calculates key metrics after classification testing for accuracy, and classifier Performance*

3.1.2 Use case Diagram

The use case diagram provided in FigureB.4 offers a visual representation of the interactions between the user and the Satellite Image Classification System. It outlines the key functionalities and flow of operations within the system, highlighting how users engage with the software to achieve their objectives

3.1.3 Activity Diagram

This activity diagram provided in figureB.5shows how the user will interact with the program and what the program should be doing depending on what the user does

3.1.4 Data Collection

Data for the Project was collected is collected from rom the Copernicus web browser, a part of the Copernicus Earth Observation Program. This program provides a comprehensive suite of satellite data encompassing a wide range of environmental and security applications, making it an invaluable resource for our system. Specifically, we took data From the Sentinel 2a satellite. We chose JPG over the TIFF and PNG because of three main reasons:

- **Accessibility** *Jpg is one of the most common image formats on the internet, it's highly compatible and support across various platforms and devices. This makes JPG an ideal choice for ensuring that the application is accessible to a broad audience*
- **Ease of use** *By using Jpg, the system simplifies the user experience, as most users are already familiar with handling and viewing JPG files. This familiarity eliminates potential barriers to entry*
- **Memory Constraints** *PG images offer the advantage of compression, which reduces file sizes significantly. This compression enables more efficient storage and faster transmission of images, which is particularly beneficial when dealing with large datasets typical in satellite imagery*

3.1.5 Feature List

A feature List B.1 for the required features in the Land Use Classification application. the features are designed to provide a comprehensive toolset for users, ensuring not only the functionality to process and classify images but also to manage resources effectively

3.2 Iteration 2

During the second iteration of developing the land use classifier project, the primary focus was on implementing the crucial image loading functionality. This feature enables users to upload their satellite imagery into the system, setting the stage for subsequent processing and analysis of the image

3.2.1 Development phase

during the development phase of iteration 2, We developed image processing functions using the pillow and open CV libraries. These libraries provided the necessary tools for checking image integrity and performing necessary transformations such as format, conversion and resizing useful for machine learning applications

we did a series of B.2tests which succeeded all of them except one, Which was a semi-fail because The corrupted image in the test was not detected by the program but it was also not used by the program and when the program you tried to use it, the program did not crash but sent an internal error so it was a semi-fail After implementing the image loader for the project.

3.3 Iteration 3

During the third iteration of the program, the primary focus was on getting the next two features from the feature list up and running. These were the image viewer and the classification configure settings. Both of these culminated in The creation of the main window class. This was the main way for the user to interact with the program from this main window. They could view the images, load their own images onto the program. enter k for the K-means classifier which we had chosen by this point to be the classifier that we would develop next. and pressing the start classification button which didn't really do anything at this point it was just more for show

3.3.1 Development Phase

Integration of key features: image viewer development purpose was to allow users to visually inspect and manage the satellite images they uploaded functionality. The image viewer was designed to display images within the application Interface. The Classification configuration settings purpose was to provide users with the ability to configure the parameters for the k-means clustering algorithm which was selected as the classification technique. users could enter the number of clusters for the k means algorithm. The setting is crucial for turning the classification process to meet the specific needs of the user, such as distinguishing between different types of geographical features or land uses. The main window class now acts as a central hub for user interactions. Within the applications design, the main window class was developed to integrate various functionalities, including image loading, viewing and configuration settings into a single, coherent interface. This design approach ensures that users have a centralised and intuitive interface from which they can control all major aspects of the application

3.4 Iteration 4

Iteration 4 of the land use classification system marked a pivotal development phase, introducing the core ImageProcessor and KProcessor classes. The ImageProcessor class handles initial image manipulations, resizing images for uniformity, flattening them into arrays, normalizing data, and reshaping these into 3D arrays suitable for clustering. These pre-processed images are then passed to the KProcessor class, which applies K-means clustering, calculates key metrics like silhouette score, inertia, and Davies-Bouldin Index (DBI), and performs cluster remapping to ensure color consistency across visual outputs. This remapping involves sorting the indices of cluster centers by the sum of their coordinates and creating a new, orderly mapping from original to sorted indices. This sophisticated data processing and clustering functionality, integrated back into the main window class, significantly enhances the system's ability to provide robust image classification and analysis, making this iteration a substantial leap forward in the system's development.

Manual Testing on this Iteration was done in 2

3.4.1 the Development Phase

Iteration 4 marked, a significant advancement in the development of the land use classifier system with the introduction of the km processing class and the image processor class. This phase represented the core of the project functionality where the primary processing and clustering operations were implemented. Overview of new classes and functionalities: The image processor class's purpose is to prepare the raw satellite images for clustering by performing a series of pre-processing steps. like image resizing the image to ensure uniformity across all images. It Flattens the image and normalizes the values of the image. Reshapes the flattened array into a 3D array appropriate for the clustering process. The km processing class purpose handles the clustering of pre-processed image data using the k means algorithm and calculates relevant clustering metrics. Functionality: The k means algorithm applies clustering to segment the image data into a specified number of clusters. metric calculation computes key performance metrics like silhouette score, inertia and the Davies bolden index or DBI to assess the quality of the clustering cluster. Remapping adjust the labelling of clusters to ensure consistency in colour mapping across different plots. Enhancing the visual coherence of cluster representations and is important for the last feature which is difference maps The processed data, along with the clustering results and metrics, are then passed back to the Main Window class. This integration allows users to interact dynamically with the processed images, view clustering results, and analyze the performance metrics directly through a user-friendly interface.

3.5 Iteration 5

Iteration 5 of the land use classification focused on the development of the display results page a used the clusters that had been taken from the km processing object and used matplotlib to clot the cluster. Be it singular or plural with the metrics that had been calculated for that specific plot, so that would be the silhouette score inertia and Davies Bowden index

3.5.1 the Development Phase

iteration 5 of the land use classifier system was centred around the development of the display results page which plays a crucial role in visualising the outcomes of the land use classification process this phase leveraged the class clustering data processed by the km processing class from the previous iterations utilising the popular python library matplotlib to create informative and interactive visualisations of the clustered satellite images, key features and functionalities developed cluster visualisation. The system now integrates functionality to display the clustered images directly on the results page when showing a single cluster or multiple clusters. The visualisations are designed to be clear

and distinguishable enhancing the user's ability to interpret the data integration of matrix alongside the visual representations key clustering metrics such as the silhouette score inertia and Davies Bowden index are displayed. These metrics are crucial for assessing the quality and effectiveness of the clustering providing users with quantitative basis to evaluate the segmentation results.

Although Iteration 5 was more focused and compact compared to other phases, its contributions were pivotal for the overall project. It set the stage perfectly for the final iteration, enabling a seamless completion of the program. This iteration not only refined critical elements but also ensured that the foundation was robust, allowing the subsequent development phase to proceed without any major obstacles and culminate successfully.

3.6 Interation 6

Iteration 6 of the land use classifier system was focused on the development of the difference map creater. A difference map is a plot that is the difference between two other plots. The way you calculate a difference map is that it's the absolute value of the difference of two maps. So what that gives us is an inverse of the two maps which highlights the differences between the two maps this is useful as you can see the differences between the same location at different times this gives the user the ability to look for differences in land use over time.

The purpose of the difference map creator is that it is designed to highlight variations between two geographical images of the same location captured at different times. By identifying these changes, users can effectively monitor and analyse alterations in land use environmental shifts or development progress over time. It is also particularly valuable for users such as environmental planners, conservationists, and urban developers who need to keep track of changes in land use, vegetation cover urban expansion or environmental degradation. It provides a clear visual representative representation of changes enhancing decision-making by providing concrete data on how areas have evolved

Chapter 4

Critical Evaluation

4.1 Methodology

The methodology adopted for the project was primarily based on feature-driven development with certain elements incorporated from the waterfall methodology, but then later removed. Originally, the project was envisioned to follow a waterfall approach which is categorised by a sequential linear process of software development. However, it was determined that the rigid structure of the waterfall methodology was not ideally suited for this project, particularly because it has to be undertaken by a solo developer. This is the decision to shift towards feature driven development was made to leverage its flexibility and efficiency [6], which are better suited for managing complex evolving projects by a single developer fdd focuses on delivering tangible working software repeatedly and in a timely manner. This approach breaks down the project into manageable chunks of work centred around individual features, making it more adaptable to changing requirements and easier to handle independently

4.2 Technologies

The development environment selected for this project was anaconda 3. Which offers several significant advantages for deploying applications across various operating systems anaconda 3. Supports compatibility with both Windows and Linux, which broadens the potential user base by ensuring that the project can operate on the most widely used platform. There are some key advantages to using anaconda3. Cross-platform compatibility which was already covered. Rich library support for machine learning is another big one. Anaconda is renowned for its comprehensive suite of machine learning libraries that are readily available and easy to install libraries such as sci-fi, scikit-learn and tensorflow. There's also the ease of library management anaconda simplifies the process of library management with its conda package manager conda allows for the easy installation, updating and management of libraries and dependencies. However, there were still issues with anaconda 3. Specifically, related to setting up environments which were a bit challenging for someone new to anaconda 3

as there was a previous environment already available for this project. However, the library would not install any of the specific packages needed for this project at that time. Also, there were some libraries are very resource intensive, specifically tensorflow which was going to be used in this project. However, it was having difficulty running on my machine so it had to be discarded.

4.3 Requirements

The objective of this project was to create a application that can take satellite imagery and use a machine learning algorithm on it and classify the land use and be able to tell you the differences between different times of that same land using the land uses that it has classified this objective was Matt and completed. However it could have been done better with more options for the user, not just gay clustering you could have had different classifiers and the ability to train your own classifiers kind of like weka in a way but all features that were put on the feature list have been implemented in this project some issues did occur such as the an ability for tensorflow to work properly on my machine is one of them. Another is the late final implementation of the program. It could have been implemented way sooner, but because of certain circumstances the program was implemented a lot later and so is a lot more rough looking than it needed to be

4.4 Design

The design of the system was effectively implemented and functions well. However, there is potential for enhancement through integration of alternative technologies that could significantly refine and improve the overall architecture and user interface. While many features of the system were developed based on thorough research, some aspects could have benefited from further exploration or a different methodological approach. Potential areas of improvement were the GUI, time management and even more machine learning algorithms could have been used. Supervised methods could have been used if there was a way to find annotated data which exists, but there was not enough time to find this data for supervised machine learning classifiers. On the topic of the GUI that was used was Tkinter, but the GUI planned to be used was qtpy. But because of time constraints because the implementation of the program was done later than needed to have been done, we had to stick with TKinter so overall, the current system design is robust and meets the outline recommendations embracing alternative technologies, however, could have improved it a lot allowing for more options for the user and an overall better looking finished application.

4.5 implementation

The implementation of project features generally aligns well with the original design, maintaining functionality and performance. Although minor, non-recurring bugs have

been identified, they do not significantly disrupt the overall operation. Most challenges stemmed from hardware constraints, particularly involving TensorFlow for image processing. The initial plan to use TensorFlow was eventually abandoned in favor of unsupervised learning methods, which circumvented the hardware limitations and allowed smooth implementation of most features without significant issues

4.6 Testing

This section identifies an area for significant improvement in the current development phase, the scope and depth of testing procedures. While core functionality is like the image processor and decay means processors were successfully tested. A broader testing strategy is necessary to ensure the overall quality and robustness of the project time constraints limited the ability to perform manual testing beyond core functionalities and the main window interface. To address this. The following recommendations are suggested for future projects. Developing a comprehensive test plan, prioritise testing, leverage automation tools, and allocate sufficient resources to testing by implementing these recommendations. Hopefully there will be a better job next time to be able to do a better job of testing

4.7 Future work

Should the project be extended, numerous enhancements could be considered to augment its capabilities significantly. Potential upgrades include advanced training options for classifiers which would enhance their accuracy and efficiency. Improvements to the user interface could be made to facilitate more interactive data exploration such as enhanced zoom capabilities for plots and images, thereby improving user engagement and analytical precision. Further, the integration of more powerful classifiers would allow for deeper insights and more robust data handling [7], specifically expanding the machine learning capabilities to include processing of teeth images and analysis of individual bands from satellite imagery could enable more detailed and specialized analysis compared to the current limitations of working with compiled jpegs. This could significantly enhance the application utility in fields requiring high-fidelity image analysis such as environment monitoring and geographic information systems

4.8 Conclusion

In conclusion, the project was successfully completed. Despite the challenges posed by stringent timelines and certain managerial issues, the compressed schedule significantly constrained the scope of extensive testing, which is a critical phase in any project development life cycle. Nonetheless, the technical execution was handled competently reflecting a deep understanding of the project's goals and methodologies. I particularly valued the collaborative experience with my supervisor Tossapon Boongoen, whose

guidance was instrumental in refining the design and helping me overcome obstacles as they arose. His encouragement was pivotal in encouraging my performance fostering a conductive learning environment. Throughout this project, I gained significant experience with agile methodologies, especially feature driven development FDD. This experience has not only enriched my professional skills, but also prepared me to integrate these methodologies into future projects more effectively

References

[1] Y. Wang, G. Cai, L. Yang, N. Zhang, and M. Du, “Monitoring of urban ecological environment including air quality using satellite imagery,” *PLOS ONE*, vol. 17, no. 8, p. e0266759, 2022.

[2] “Flood detection in urban areas using satellite imagery and machine learning,” *Water*, vol. 14, no. 7, pp. 1140–1140, 2022.

[3] *Dynamic monitoring of urban planning based on image data fusion in multi-source remote sensing*. CRC Press eBooks, 2022, pp. 494–502.

[4] arwen vaughan, “the expert’s designtools.” [Online]. Available: planttext.com

[5] K. K. Kolawole and B. W. Adebayo, “Performance evaluation student result using k-means clustering,” *International journal of communication and information technology*, vol. 3, no. 1, pp. 01–05, 2022.

[6] F. Belli, T. Tuglular, and E. Ufuktepe, “A new approach to event- and model-based feature-driven software testing and comparison with similar approaches,” *International advanced researches and engineering journal*, 2022.

[7] “K-textures, a self-supervised hard clustering deep learning algorithm for satellite image segmentation,” *Frontiers in Environmental Science*, vol. 10, 2022.

[8] S. R. Vijayalakshmi and S. M. Kumar, “Performance analysis of vegetation area classifications in satellite images using machine and deep learning approaches,” pp. 1–6, 2022.

[9] “Machine learning-based land use and land cover mapping using multi-spectral satellite imagery: A case study in egypt,” *Sustainability*, vol. 15, no. 12, pp. 9467–9467, 2023.

[10] S. 2A, “Data source.” [Online]. Available: https://dataspace.copernicus.eu/browser/

[11] “Semantic segmentation using k-means clustering and deep learning in satellite image,” vol. 2019, pp. 192–196, 2019.

[12] M. H. Salim, “Monitoring urban areas for climate change adaptation using remotely sensed indicators in the urbangreeneye project,” 2023.

[13] C. J. West, “A comparison of software project architectures: agile, waterfall, spiral, and set-based,” Ph.D. dissertation, Massachusetts Institute of Technology, 2018.

[14] T. Xiong, “Application of remote sensing technology in sustainable urban planning and development,” *Applied and Computational Engineering*, vol. 3, no. 1, pp. 283–288, 2023.

[15] “A distance metric for uneven clusters of unsupervised k-means clustering algorithm,” *IEEE Access*, vol. 10, pp. 86 286–86 297, 2022.

[16] Z. Nawaz, S. Aftab, and F. Anwer, “Simplified fdd process model,” *International Journal of Modern Education and Computer Science*, vol. 9, no. 9, pp. 53–59, 2017.

[17] S. D. Mishra, N. Monath, M. Boratko, A. Kobren, and A. McCallum, “An evaluative measure of clustering methods incorporating hyperparameter sensitivity,” *Proceedings of the ... AAAI Conference on Artificial Intelligence*, vol. 36, no. 7, pp. 7788–7796, 2022.

Appendices

Appendix A

Use of Third-Party Code, Libraries and Generative AI

1.1 Third Party Code and Software Libraries

The only Software libraries used are:

- **numpy: 1.23.5**
- **matplotlib: 3.80**
- **scikit-Learn: 1.30**
- **cv2: 4.60**
- **Pillow: 10.2.0**
- **tk: 8.6.12**
- **math**

1.2 Generative AI

No Generative AI was used to create any part of this project

Appendix B

Graphs and tables

2.1 Figures

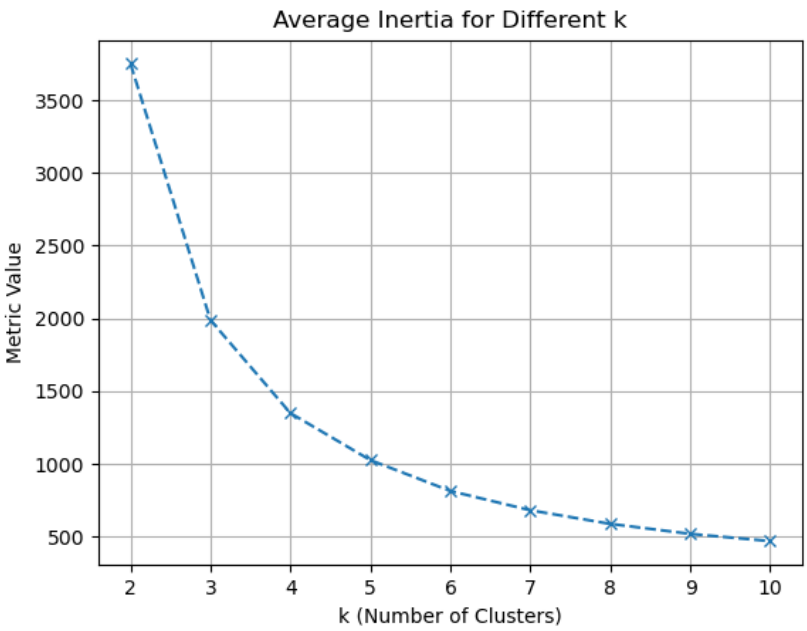

Figure B.1: Graph for Average Inertia for K

2.2 tables

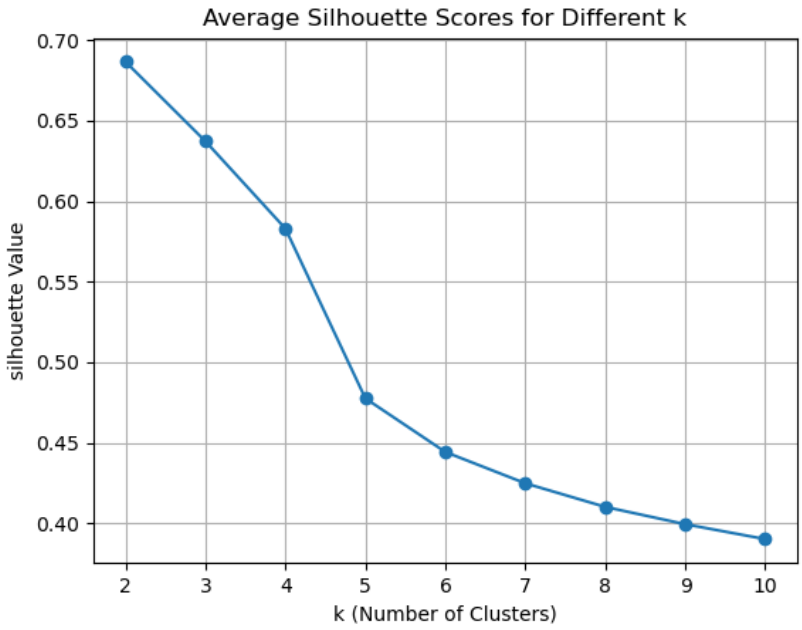

Figure B.2: Graph for Average Silhouette scores for K

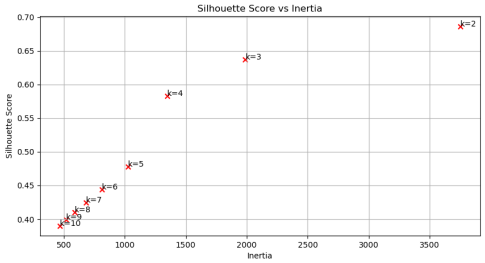

Figure B.3: Graph for the Silhouette Score vs. Inertia

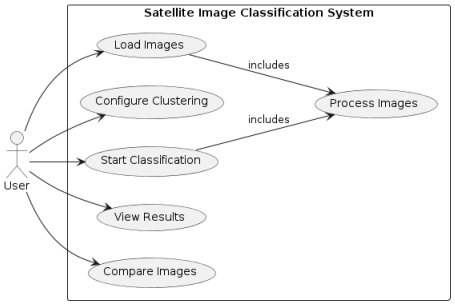

Figure B.4: Use case Diagram

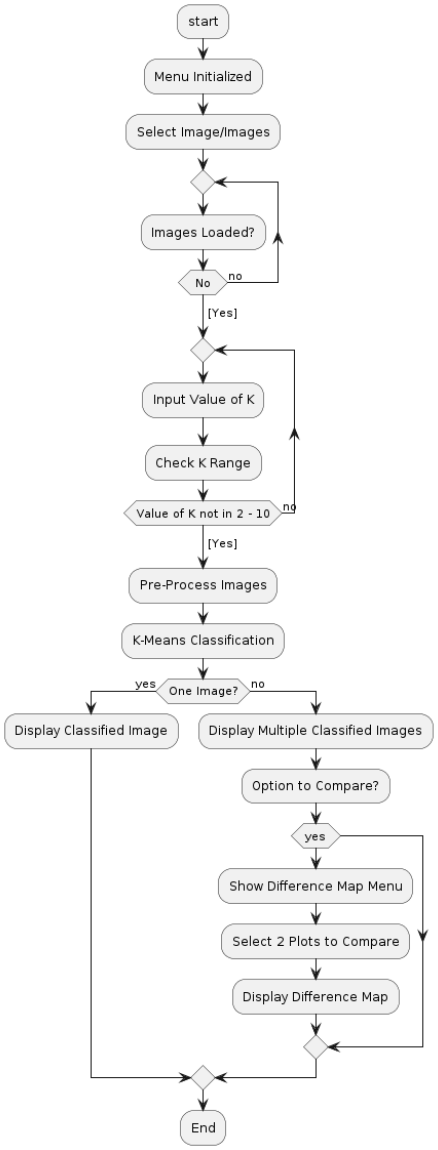
start
Menu Initialized
Select Image/Images
Images Loaded?
No
no
[Yes]
Input Value of K
Check K Range
Value of K not in 2 - 10
no
[Yes]
Pre-Process Images
K-Means Classification
yes
One Image?
no
Display Classified Image
Display Multiple Classified Images
Option to Compare?
yes
Show Difference Map Menu
Select 2 Plots to Compare
Display Difference Map
End

Figure B.5

Table B.1: Feature List

Feature Number	Feature Name	Feature Description	Acceptance Test
1	Load Satellite Imagery	Users should be able to upload and store satellite images	Users can upload images in supported formats (e.g., JPG). The system confirms successful uploads
2	View Images	Users should be able to view and manage their uploaded satellite images	All uploaded images are displayed in the user dashboard with options to zoom and pan
3	Configure classification	Users should be able to configure parameters for classification	Users can select and set parameters such as number of clusters
4	Start Classification	Users should be able to initiate the classification process on selected images	Classification process starts, and the system displays processing status.
5	View Classification Results	Users should be able to view and analyze the results of image classifications	Results, including cluster maps and metrics are displayed for user analysis.
6	Change Detection	Users should be able to compare classification results over time	comparison interface allows user to view a diff map of the two areas with the differences highLighted

Table B.2: Test Descriptions

Test Number	Test Name	Test Description	Pass or Fail
1	File Format Compatibility Test	Upload a file which is jpg or png	Pass
2	File Size and Resolution Test	Upload a very large Image	Pass
3	Corrupted File Handling	Upload a corrupted image file	Fail
4	Multi-file Upload Test	Upload multiple Files at once	Pass
6	Upload Limits and Restrictions	Upload as many files as You can, should be able to	Pass