Future University in Egypt.
Faculty of Computers and Information Technology.
Department of Information Systems.

Graduation Project:

Development of an Enterprise System for Satellite Image Processing Using Artificial Intelligence.

Faculty of Computer Science and Information Technology.

Information Systems Department.

Prepared by	
20194462	**Ahmed HossamEldeen Elsaeed Morgan.**
20193056	**Abdelrahman Kamel Sayed Hassan.**
20191521	**Mohamed Yosr Mohamed Mohsen Mohamed.**

Supervised by
Prof/ Safaa Mohamed Hassan Said.
TA/ Warda Mustafa Saied Mahmoud.

February 2024.

وَمَا أُوتِيتُمْ مِنَ الْعِلْمِ إِلَّا قَلِيلًا (الإسراء:85). صدق الله العظيم

Abstract.

Satellite imagery has become an indispensable source of information for numerous industries and domains. From monitoring agricultural lands and urban development to assessing environmental changes and aiding disaster management, satellite images provide valuable insights that assist in decision-making processes. However, the vast amount of data generated by satellites poses significant challenges in terms of processing and extracting meaningful information in a timely manner. This is where the application of artificial intelligence (AI) techniques comes into play, offering promising solutions to automate and enhance satellite image analysis.

The objective of this graduation project is to develop an enterprise system that leverages the power of AI for satellite image processing. The system aims to overcome the limitations of traditional methods and enable efficient and accurate extraction of relevant information from satellite imagery, the system seeks to improve the accuracy and efficiency of image classification, object detection, and feature extraction processes.

The developed enterprise system encompasses several modules, each contributing to the overall goal of efficient satellite image processing. The data ingestion module ensures seamless integration with satellite data sources, enabling the system to acquire and store large volumes of satellite images. This module establishes a robust framework for data management, facilitating easy access to diverse satellite datasets.

The preprocessing module plays a crucial role in preparing the acquired satellite data for subsequent analysis. It focuses on tasks such as noise reduction, image enhancement, and spatial calibration, which are essential for improving the quality and reliability of the data. By addressing these preprocessing challenges, the system ensures that the subsequent analysis is performed on accurate and well-prepared data.

Acknowledgement.

It has been a great opportunity to gain lots of experience in real-time projects, followed by the knowledge of how to design and analyze real projects. For that, we want to thank all the people who made it possible for students like us. Special thanks to the graduation Project Unit for the efforts they made to provide us with all useful information and make the path clear for the students to implement all the education periods in real-time project design and analysis. We would like to express our special thanks of gratitude to Prof. Safaa Mohamed Hassan for her great support, motivation to us, supervision, continuous help, and fruitful discussion during the progress of this work and her critical reading of the document. And Engineer Warda Mustafa who helped us to make the project the way it has now and helped us to earn many skills.

Grateful thanks are also due to Dr. Amira Idrees, head of the Information System Department in FUE, for her fruitful discussion and advice during the pre-request presentation of our graduation project.

Secondly, we would like to thank our university FUE.
Special thanks to Prof. Ebada Sarhan.

Contents

1. Table of Figure.

Introduction

Chapter One.

1.1- Overview.

The objective of this project is to enhance the quality of satellite imagery using image processing techniques and artificial intelligence (AI). Satellite imagery plays a vital role in various fields, including environmental monitoring, urban planning, and disaster management. However, satellite images often suffer from inherent limitations such as noise, and spatial distortions. This project aims to address these issues by employing radiometric enhancement, spatial enhancement, and spectral enhancement techniques.

Radiometric enhancement techniques focus on improving the brightness, contrast, and dynamic range of satellite images. By adjusting the pixel values based on statistical analysis and image characteristics, radiometric enhancement enhances the visibility of details and improves the overall visual quality of the images. This technique ensures that important information within the satellite imagery is accurately represented, facilitating better analysis and interpretation.

Spatial enhancement techniques aim to mitigate spatial distortions present in satellite images. These distortions can arise due to various factors, including sensor artifacts, atmospheric effects, and platform vibrations. By applying spatial filtering and correction algorithms, spatial enhancement techniques improve the spatial resolution, sharpness, and geometric accuracy of the images. This enables more precise measurements and analysis, particularly in applications where fine details are crucial.

Spectral enhancement techniques address issues related to inconsistent spectral characteristics across satellite images. Different satellite sensors and acquisition conditions can lead to variations in spectral responses, making it challenging to compare and analyze images from different sources or time periods. Spectral enhancement techniques involve spectral calibration, normalization, and correction algorithms to standardize the spectral information across images. This ensures consistent spectral signatures and facilitates reliable analysis and comparison of the data.

In addition to these image processing techniques, artificial intelligence is utilized to further enhance the quality of satellite imagery. Machine learning and deep learning algorithms are trained on a diverse dataset of satellite images to learn patterns and extract meaningful information. These AI models can aid in noise reduction, feature extraction, and image restoration, leading to improved image quality and more accurate analysis.

The project involves the development of algorithms and methodologies to implement radiometric enhancement, spatial enhancement, and spectral enhancement techniques. Extensive experimentation and evaluation are conducted using real-world satellite datasets to validate the effectiveness of the proposed methods. Performance metrics such as image quality measures, spatial accuracy, and spectral consistency are used to assess the improvements achieved through the image processing techniques and AI models.

The outcomes of this project have significant implications for various industries and applications that rely on satellite imagery. By improving the quality of satellite images, the project enhances the accuracy and reliability of analysis, decision-making, and monitoring processes. The findings contribute to the advancement of image processing and AI techniques in the field of satellite imagery, opening possibilities for improved environmental monitoring, urban planning, disaster management, and more.

1.2- Problem Statement.

Satellite imagery plays a crucial role in various industries and applications, including environmental monitoring, urban planning, and disaster management. However, the quality of satellite images often suffers from inherent limitations such as noise, spatial distortions, and spectral inconsistencies. These limitations pose significant challenges in accurately interpreting and extracting meaningful information from the imagery, leading to suboptimal decision-making and analysis outcomes.

The existing techniques for satellite image enhancement have limitations in effectively addressing the specific challenges associated with radiometric, spatial, and spectral quality improvements. Conventional methods often fail to deliver satisfactory results due to their inability to handle the complex characteristics and variations present in satellite images. Additionally, the lack of automated and intelligent approaches hinders the efficient processing and enhancement of large-scale satellite datasets.

Furthermore, the absence of advanced artificial intelligence (AI) techniques limits the potential for automated analysis and restoration of satellite images. Machine learning and deep learning algorithms have demonstrated immense capabilities in image processing tasks, but their application to satellite imagery enhancement is relatively unexplored.

Therefore, there is a pressing need to develop an innovative and comprehensive solution that utilizes advanced image processing techniques and AI algorithms to overcome the limitations and improve the quality of satellite imagery. This solution should address the challenges of radiometric enhancement, spatial enhancement, and spectral enhancement in a unified and intelligent manner, enabling accurate interpretation, analysis, and utilization of satellite imagery across various domains.

By addressing these challenges, the proposed solution aims to enhance the accuracy and reliability of satellite image analysis, thereby improving decision-making processes and supporting effective monitoring and management in industries that rely on satellite data. The development of an advanced enterprise system that integrates state-of-the-art image processing techniques and AI algorithms will pave the way for significant advancements in satellite imagery quality and utilization, ultimately benefiting a wide range of applications and industries.

1.3- Scope and Objectives

1.3.1 Scope

The scope of this project encompasses the development of an enterprise system for improving the quality of satellite imagery using image processing techniques and artificial intelligence (AI). The project focuses on addressing the challenges associated with radiometric enhancement, spatial enhancement, and spectral enhancement of satellite images. The system will be designed to handle large-scale satellite datasets and facilitate automated processing and analysis. The project's scope also includes evaluating the effectiveness of the developed system using real-world satellite imagery datasets.

1.3.2 Objectives

- Develop algorithms and methodologies for radiometric enhancement to improve the brightness, contrast, and dynamic range of satellite images.
- Design and implement spatial enhancement techniques to mitigate spatial distortions, enhance spatial resolution, and improve geometric accuracy in satellite imagery.
- Explore spectral enhancement methods to standardize spectral characteristics across satellite images and facilitate consistent analysis and comparison.
- Incorporate artificial intelligence techniques to enhance image quality, reduce noise, and restore details in satellite imagery.
- Develop an enterprise system that integrates the developed algorithms and AI models, providing a unified platform for efficient and automated processing of satellite imagery.
- Evaluate the performance and effectiveness of the developed system using real-world satellite datasets, assessing improvements in image quality, spatial accuracy, and spectral consistency.
- Demonstrate the applicability and benefits of the developed system in various industries and applications that rely on satellite imagery, such as environmental monitoring, urban planning, and disaster management.
- Provide comprehensive documentation, including technical specifications and user guides, to facilitate the deployment and utilization of the developed system in practical settings.
- Contribute to the advancement of image processing techniques and AI applications in the field of satellite imagery, paving the way for further research and development in this domain.

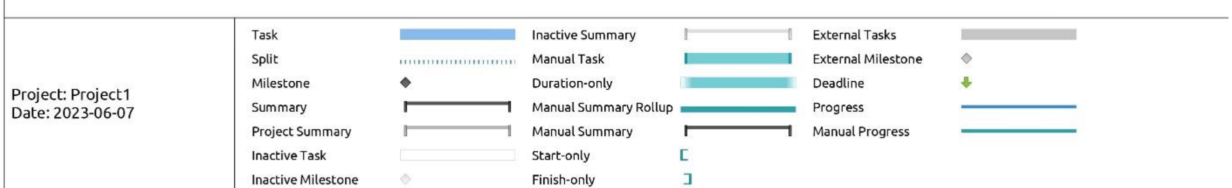
1.4- Work Methodology.

- Requirement Gathering:
  Gather the specific requirements for the website, considering factors such as user interface, functionality, etc.
- Design and Wireframing:
  Design the website's layout, user interface, and navigation structure. Create wireframes to visualize the website's functionality and flow. Ensure a user-friendly and intuitive design.
- Data Acquisition:
  Acquire relevant satellite imagery datasets, including images with different radiometric, spatial, and spectral characteristics. Curate a diverse dataset that covers a wide range of scenarios.
- Image Processing Algorithms:
  Develop algorithms for radiometric enhancement, spatial enhancement, and spectral enhancement of satellite images. Implement image processing techniques such as histogram equalization, noise reduction, spatial filtering, and spectral calibration.
- AI Integration:
  Incorporate artificial intelligence techniques into the website to enhance image quality and facilitate automated processing. Utilize machine learning or deep learning algorithms to learn patterns, extract features, and improve image restoration.
- Website Development:
  Develop the website, incorporating the designed user interface and integrating the developed image processing algorithms. Implement features such as image upload, processing, and visualization.
- Testing and Evaluation:
  Test the website's functionality, performance, and user experience. Evaluate the effectiveness of the implemented image processing techniques and AI models using real-world satellite image datasets. Measure metrics such as image quality, accuracy, and processing speed.
- Documentation and Reporting:
  Document the technical specifications, algorithms used, and methodologies employed in developing the website. Prepare user documentation and guides to assist users in utilizing the website's features effectively.

By following this work methodology, the project aims to develop a user-friendly website that utilizes image processing techniques and AI to enhance the quality of satellite imagery. The methodology ensures a systematic approach from requirement gathering to deployment, with a focus on usability, effectiveness, and continuous improvement to deliver an enhanced user experience and improved satellite image quality.

1.5- Work Plan (Gantt Chart).

ID	Task Mode	Task Name	Duration	Start	Finish	Predecessors
1		**1 Project Initiation Phase**	**2 wks**	**2023-06-07**	**2023-06-20**	
2		1.1 Define project objectives, scope, and deliverables.	3 days	2023-06-07	2023-06-09	
3		1.2 Identify project stakeholders and establish communication channel	3 days	2023-06-07	2023-06-09	
4		1.3 Set up project management tools and documentation templates.	3 days	2023-06-07	2023-06-09	
5		1.4 Conduct initial project kickoff meeting.	2 days	2023-06-07	2023-06-08	
6		**1.5 Milestone: Finish Project Initiation Phase.**	1 day	2023-06-07	2023-06-07	
7		**2 Requirement Analysis and Design Phase**	**3 wks**	**2023-06-27**	**2023-07-17**	
8		2.1 Conduct user research and gather requirements for the website.	6 days	2023-06-27	2023-07-04	
9		2.2 Create a detailed design plan, including wireframes and UI/UX considerations.	6 days	2023-06-27	2023-07-04	
10		2.3 Define the required image processing techniques	9 days	2023-06-27	2023-07-07	
11		**2.4 Milestone: Finish Requirement Analysis and Design Phase**	0 days	2023-06-27	2023-06-27	
12		**3 Data Acquisition and Preprocessing Phase**	**2 wks**	**2023-07-15**	**2023-07-27**	
13		3.1 Identify and acquire relevant satellite imagery datasets.	6 days	2023-07-17	2023-07-24	
14		3.2 Preprocess the acquired data, including resizing, format conversion, and metadata extraction.	6 days	2023-07-17	2023-07-24	
15		**3.3 Milestone: Finish Data Acquisition and Preprocessing Phase**	0 days	2023-07-15	2023-07-15	
16		**4 Algorithm Development Phase**	**8 wks**	**2023-08-31**	**2023-10-25**	
17		4.1 Develop algorithms for radiometric enhancement, spatial enhancement, and spectral enhancement.	21 days	2023-08-31	2023-09-28	
18		4.2 Implement image processing techniques and integrate AI models.	12 days	2023-08-31	2023-09-15	
19		4.3 Validate and optimize the algorithms for enhanced image quality.	6 days	2023-08-31	2023-09-07	
20		**4.4 Milestone: Finish Algorithm Development Phase.**	0 days	2023-08-31	2023-08-31	
21		**5 Website Development Phase**	**10 wks**	**2023-09-01**	**2023-11-09**	
22		5.1 Set up the website infrastructure, including server configuration and database setup.	10 days	2023-09-22	2023-10-05	
23		5.2 Develop front-end components and user interface.	6 days	2023-09-01	2023-09-08	
24		5.3 Implement image upload, processing, and visualization functionalit	6 days	2023-09-01	2023-09-08	
25		5.4 Integrate the developed algorithms and AI models into the website	1 day	2023-09-01	2023-09-01	
26		**5.5 Milestone: Finish Website Development Phase.**	5 days	2023-09-01	2023-09-07	
27		**6 Testing and Evaluation Phase**	**4 wks**	**2023-11-27**	**2023-12-22**	
28		6.1 Conduct thorough testing of the website's functionality, performance, and user experience.	1 day	2023-11-27	2023-11-27	
29		6.2 Evaluate the effectiveness of the implemented image processing techniques and AI models.	1 day	2023-11-27	2023-11-27	
30		6.3 Collect user feedback and make necessary refinements.	1 day	2023-11-27	2023-11-27	
31		**6.4 Milestone: Finish Testing and Evaluation Phase**	1 day	2023-11-27	2023-11-27	
32		**7 Documentation and Deployment Phase**	**2 wks**	**2023-11-10**	**2023-11-23**	
33		7.1 Prepare technical documentation, including specifications and methodologies used.	1 day	2023-11-10	2023-11-10	
34		7.2 Create user documentation and guides for website usage.	1 day	2023-11-10	2023-11-10	
35		7.3 Deploy the website to a suitable web hosting platform or server.	1 day	2023-11-10	2023-11-10	



Page 1

Figure 1: Project Gantt Chart (illustrates all major system tasks).

1.6- Work Organization.

1. **Chapter One:** An Introduction and Overview of Our Enterprise System.
2. **Chapter Two:** Related Work and State of The Art.
3. **Chapter Three:** Proposed Solutions The ideas on which we built the system.
4. **Chapter Four:** Data Used in The System.
5. **Chapter Five:** Design and Simulation Setup of The System.
6. **Chapter Six:** System Analysis.
7. **Chapter Seven:** Implementation.
8. **Chapter Eight:** Testing And Evaluation.
9. **Chapter Nine:** Conclusions And Future Work.
10. **Chapter Ten:** Manuals and User Guide.

Related Work

Chapter Two.

2.1.1 Background.

Satellite imagery is an important tool for understanding and monitoring the Earth's surface. It provides a way to observe large areas of the planet remotely and non-invasively. Satellite images are used for a wide range of applications, including environmental monitoring, disaster management, urban planning, and agriculture. However, satellite images often suffer from various types of degradation, such as low contrast, noise, and blur, which can affect the accuracy of the information extracted from them. Image enhancement techniques can be used to improve the quality of satellite images and make them more suitable for various applications. In this project, we propose a method for enhancing satellite images using radiometric, spatial, and spectral. Satellite imagery is acquired using sensors that capture electromagnetic radiation reflected or emitted from the Earth's surface. The sensors can be passive or active, and they operate in different spectral bands, ranging from visible and near-infrared to thermal infrared and microwave. The raw satellite images captured by the sensors often suffer from various types of degradation, such as low contrast, noise, and blur, which can affect their visual quality and information content. Image enhancement techniques are used to improve the quality of satellite images and make them more suitable for various applications. Radiometric enhancement techniques are used to adjust the brightness and contrast of the image to improve its visual quality. Spatial enhancement techniques are used to filter and sharpen the image to improve its spatial resolution. Spectral enhancement techniques are used to enhance specific bands of the image to highlight certain features. In recent years, deep learning techniques have been applied to satellite image enhancement, including convolutional neural networks (CNNs), generative adversarial networks (GANs), and attention mechanisms. These techniques have shown promising results in enhancing the visual quality and information content of satellite images. Satellite Image Enhancement: Satellite image enhancement is a process of improving the quality of satellite images by applying various image processing techniques. Satellite images are acquired by sensors mounted on orbiting satellites that capture electromagnetic radiation reflected or emitted from the Earth's surface. However, the raw satellite images often suffer from various types of degradation, such as low contrast, noise, and blur, which can affect their visual quality and information content. Satellite image enhancement techniques are used to improve the quality of these images to make them more suitable for various applications, including environmental monitoring, disaster management, urban planning, and agriculture. Satellite image enhancement techniques can be grouped into three categories: radiometric enhancement, spatial enhancement, and spectral enhancement. Radiometric enhancement techniques are used to adjust the brightness and contrast of the image to improve its visual quality. Spatial enhancement techniques are used to filter and sharpen the image to improve its spatial resolution. Spectral enhancement techniques are used to enhance specific bands of the image to highlight certain features. Some commonly used techniques for satellite image enhancement include histogram equalization, contrast stretching, filtering methods such as the Laplacian filter, the Sobel filter, and the unsharp masking method, and spectral enhancement methods such as principal component analysis

(PCA), the Bovey transform, and the IHS transform. Satellite image enhancement techniques play an important role in improving the accuracy of the information extracted from satellite images, which can be used in a variety of applications, such as environmental monitoring, disaster management, urban planning, and agriculture.

2.1.2 Artificial intelligence (AI):

It refers to machines, chiefly computers working like humans. In AI, machines perform tasks like speech recognition, problem-solving and learning, etc. Machines can work and act like a human if they have enough information. So, in artificial intelligence, knowledge engineering plays a vital role. The relation between objects and properties is established to implement knowledge engineering. Artificial Intelligence can be divided into different categories based on the machine's capacity to use past experiences to predict future decisions, memory, and self-awareness.

2.1.3 Machine learning:

Is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns, and make decisions with minimal human intervention. Machine learning is an application of artificial intelligence (AI) that provides systems with the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves. The process of learning begins with observations or data, such as examples, direct experience, or instruction, to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers to learn automatically without human intervention or assistance and adjust actions accordingly.

2.1.4 Data Used (Material):

Raster and vector are two very different, but common data formats used to store geospatial data. Vector data use X and Y coordinates to define the locations of points, lines, and areas (polygons) that correspond to map features such as fire hydrants, trails, and parcels. As such, vector data tends to define centers and edges of features. Raster data uses a matrix of square areas to define where features are located. These squares, also called pixels, cells, and grids, typically are of uniform size, and their size determines the detail that can be maintained in the dataset. Because raster data represent square areas, they describe interiors rather than boundaries as is the case with vector data. Raster data is stored as a grid of values which are rendered on a map as pixels. Each pixel value represents an area on the Earth's surface. Vector data structures represent specific features on the Earth's surface and assign attributes to those features. An example of discrete raster data is population density. Continuous data examples are temperature and elevation measurements. Two types of raster satellite data were used in this study, Landsat 8, and Sentinel 2 Images.

2-1 Characteristics of Landsat-8 and Sentinel- 2:

band	Description	Wavelength	Resolution
1-	Violet-Deep blue	0.43-0.45	30
2-	Blue	0.45-0.51	30
3-	Green	0.53-0.50	30
4-	Red	0.64-0.67	30
5-	Near infrared	0.85-0.88	30
6-	Shortwave infrared	1.57-1.65	30
7-	Shortwave infrared	2.11-2.29	30
8-	Panchromatic	0.50-0.68	15
9-	Cirrus clouds	1.36-1.38	30
10-	Thermal infrared	10.62-11.19	30
11-	Thermal infrared	11.50-12.51	30

Spectral Bands	Wavelength	Spatial Resolution
Band 1(Aerosol)	443	60
Band 2 (Blue)	490	10
Band 3 (Green)	560	10
Band 4 (Red)	665	10
Band 5 (vegetation Red-Age)	705	20
Band 6 (Vegetation Red-Age)	740	20
Band 7 (Vegetation Red-Age)	783	20
Band 8 (NIR)	842	10
Band 8A (Vegetation Red-Age)	865	20
Band 9 (NIR)	945	60
Band 10 (MIR)	1375	60
Band 11 (MIR)	1610	20
Band 12 (MIR)	2190	20

- Both Sentinel-2 and Landsat 8 are satellite remote sensing systems that provide valuable data for a variety of applications related to land use, agriculture, forestry, and water resource management. However, there are some differences between them in terms of their spatial resolution, spectral range, revisit time, data format, data size, and availability.
- In terms of spatial resolution, Sentinel-2 has a higher resolution of 10, 20, or 60 meters depending on the spectral band, while Landsat 8 has a spatial resolution of 30 meters for most of its spectral bands, except for the panchromatic band, which has a higher resolution of 15 meters.
- Regarding the spectral range, both systems cover a range of bands from the visible to the thermal infrared regions, with Sentinel-2 covering 13 bands and Landsat 8 covering 11 bands.

In terms of revisit time, Sentinel-2 has a shorter revisit time of 5 days, while Landsat 8 has a revisit time of 16 days.

- The data format for Sentinel-2 is CHDF, while Landsat 8 data is distributed in the Level-1 data product format. The size of Sentinel-2 data can range from tens of megabytes to several gigabytes, depending on the level of processing, while a single Landsat 8 scene can be several hundred megabytes in size.
- Both systems are widely used for a variety of applications, and their data is freely available through different platforms. Sentinel-2 data is available through the European Space Agency's (ESA) Copernicus Open Access Hub, while Landsat 8 data is available through the USGS Earth Explorer and Glove's platforms. Access to both systems can be facilitated through various software platforms, such as ESA's Sentinel Application Platform (SNAP), Google Earth Engine, QGIS, and ENVI.
- Overall, while Sentinel-2 has a higher spatial resolution and shorter revisit time, Landsat 8 covers a slightly narrower spectral range and may have a smaller data size. The choice of system to use will depend on the specific application and the user's requirements for spatial resolution, spectral range, and revisit time.

2-2 Techniques:

- **Histogram Stretching:**
  Image histogram stretching is a contrast enhancement technique that adjusts the distribution of pixel intensities. By expanding or compressing the range of intensity values, histogram stretching enhances the dynamic range of an image. This is particularly useful in images with low contrast, where details may be obscured. The process involves stretching the histogram to cover the entire available intensity range, improving the visibility of details and overall image quality. Histogram stretching is a straightforward yet powerful method for enhancing the visual appeal of images without introducing new information.

- **Image Transformation:**
  Image Transformation involves the manipulation of spatial characteristics within an image without altering pixel intensity. This process is crucial in various computer vision applications, where aligning or adjusting images is essential. Common transformations include rotation, scaling, translation, and shearing. For instance, in image registration, transformations are used to align multiple images, facilitating comparison or analysis. Additionally, correcting perspective distortions caused by camera angles is a key aspect of image transformation, ensuring accurate representation and analysis of visual data.

- **Image Filtration:**
  Image Filtration focuses on modifying pixel values based on their local neighborhood, utilizing various filters for specific effects. Filters like Gaussian blur is employed for smoothing or reducing noise, while Sobel operators and Laplacian filters enhance edges and highlight specific features. Filtration is integral in image processing, aiding in tasks such as edge detection, feature extraction, and noise reduction. The versatility of filters allows for the customization of image appearance and the extraction of relevant information, contributing to the effectiveness of computer vision algorithms.

- **Super Resolution**
  Super resolution is a sophisticated technique aiming to enhance the spatial resolution of an image, generating a higher-resolution version than the original. This process is vital in applications where increased image quality is crucial, such as medical imaging or surveillance. Advanced algorithms, including deep learning approaches like convolutional neural networks, are often employed for super-resolution tasks. These algorithms learn intricate patterns and relationships in the data, allowing them to generate finer details and achieve impressive upscaling results. Super resolution contributes significantly to improving the visual fidelity of images, enabling better analysis and interpretation in various fields.

2-3 Literature Survey In recent years:

Several studies have proposed different techniques for enhancing satellite images. Radiometric enhancement techniques, including histogram equalization, contrast stretching, and gamma correction, have been widely used to improve the brightness and contrast of satellite images. Spatial enhancement techniques, including filtering and sharpening methods such as the Laplacian filter, the Sobel filter, and the unsharp masking method, have been used to improve the spatial resolution of satellite images. Spectral enhancement techniques, including principal component analysis (PCA), the Bovey transform, and the IHS transform, have been used to highlight specific features of satellite images. Deep learning techniques have also been used for satellite image enhancement. Convolutional neural networks (CNNs) have been used to learn the mapping between degraded and enhanced images. Generative adversarial networks (GANs) have been used to generate high-quality satellite images from low-quality ones. Attention mechanisms have been used to focus on specific regions of the image during the enhancement process.

2-4 Literature Survey:

In recent years, several studies have proposed different techniques for enhancing satellite images. Radiometric enhancement techniques, including histogram equalization, contrast stretching, and gamma correction, have been widely used to improve the brightness and contrast of satellite images. Spatial enhancement techniques, including filtering and sharpening methods such as the Laplacian filter, the Sobel filter, and the unsharp masking method, have been used to improve the spatial resolution of satellite images. Spectral enhancement techniques, including principal component analysis (PCA), the Bovey transform, and the IHS transform, have been used to highlight specific features of satellite images. Deep learning techniques have also been used for satellite image enhancement. Convolutional neural networks (CNNs) have been used to learn the mapping between degraded and enhanced images. Generative adversarial networks (GANs) have been used to generate high-quality satellite images from low-quality ones. Attention mechanisms have been used to focus on specific regions of the image during the enhancement process.

Proposed Solutions.

Chapter Three.

3 Concept for Image Processing and Enhancement.

Image processing is a multidisciplinary field that involves the application of computational algorithms to digital images. The fundamental objective is to manipulate, analyze, and extract information from visual data for diverse applications. At its core, image processing encompasses a broad spectrum of techniques designed to enhance, transform, or derive insights from digital images.

The process begins with the acquisition of digital images through various devices such as cameras, satellites, or medical imaging equipment. These images are represented digitally, with each pixel corresponding to a specific location and carrying intensity or color information. The representation serves as the foundation for subsequent processing steps.

One crucial aspect of image processing is enhancement, where techniques are employed to improve the visual quality of images. This may involve adjustments to contrast, brightness, or the reduction of noise, contributing to images that are more suitable for analysis or human perception.

Image transformation is another key concept, involving modifications to the spatial domain or intensity values of an image. This can include geometric corrections or remapping intensity values to achieve specific objectives in image processing.

Filters play a significant role in image processing by allowing the modification or extraction of features from images. Linear filters, such as convolution, and non-linear filters, like median filtering, are commonly applied for tasks such as blurring, sharpening, or noise reduction.

Image restoration techniques focus on recovering or improving the quality of degraded images, addressing issues like noise, blurring, or compression artifacts. These methods contribute to the restoration of images for better interpretability and analysis.

Segmentation involves dividing images into meaningful regions or segments, facilitating object recognition and further analysis. This step is crucial for understanding the content and structure of images in applications ranging from medical imaging to computer vision.

Image compression is employed to reduce the size of images, optimizing storage space and facilitating efficient transmission without compromising acceptable image quality. Feature extraction is another essential concept, involving the identification and extraction of relevant information from images, laying the groundwork for subsequent analysis or pattern recognition.

Pattern recognition, often incorporating machine learning algorithms, focuses on recognizing and classifying patterns or objects within images based on extracted features. Morphological image processing involves operations that analyze and manipulate the shape or structure of objects in an image, contributing to tasks like object detection and image understanding.

Ultimately, image processing plays a crucial role in numerous fields, including medicine, remote sensing, computer vision, surveillance, and entertainment. Ongoing advancements in hardware capabilities and algorithmic development continue to drive the evolution of image processing techniques, expanding their applications and impact across diverse domains.

3-1 Purpose of Image Enhancement.

Image processing for enhancement involves a set of techniques and algorithms designed to improve the visual quality, interpretability, and overall utility of digital images. The goal is to reveal hidden details, reduce noise, enhance contrast, and make the images more suitable for analysis, visualization, or human perception. Here are key concepts related to image processing for enhancement:

1. **Contrast Enhancement:**
    - The process of increasing the visual difference between the darkest and brightest parts of an image.
    - Techniques like histogram stretching, histogram equalization, and contrast stretching are commonly used to enhance contrast.
2. **Brightness Adjustment:**
    - Manipulating the overall brightness level of an image to make it more visually appealing or suitable for specific applications.
    - Techniques like gamma correction or linear scaling are employed for adjusting brightness.
3. **Histogram Equalization:**
    - A method to enhance the contrast of an image by redistributing the intensity values across a wider range.
    - It aims to make the histogram of the image more uniform.
4. **Spatial Filtering:**
    - The application of filters to modify the pixel values of an image based on their spatial relationships.
    - Techniques include convolution with various filter kernels for tasks such as blurring, sharpening, or edge detection.
5. **Frequency Domain Processing:**
    - Analyzing and enhancing images in the frequency domain using techniques like Fourier transform.
    - High-pass and low-pass filtering in the frequency domain can be used for tasks such as sharpening or noise reduction.
6. **Dynamic Range Adjustment:**
    - Adjusting the range of intensity values in an image to better utilize the available bit-depth.
    - This helps in enhancing details in both bright and dark regions of the image.

7. **Multi-scale Image Enhancement:**
    - Applying enhancement techniques at different scales or resolutions to address features of varying sizes.
    - Multi-scale methods can enhance both fine details and global structures in an image.
8. **Adaptive Enhancement:**
    - Tailoring enhancement techniques based on the local characteristics of an image.
    - Adaptive histogram equalization, adaptive filtering, and local contrast enhancement methods fall under this category.
9. **Image Dehazing:**
    - Removing or reducing the effects of atmospheric haze in satellite or outdoor images.
    - Dehazing techniques enhance visibility and improve image quality in scenarios with poor atmospheric conditions.
10. **Super Resolution:**
    - Increasing the spatial resolution of an image to reveal finer details.
    - Super resolution techniques use algorithms, including deep learning models, to generate high-resolution images from lower-resolution counterparts.
11. **Color Enhancement:**
    - Improving the color balance and vibrancy of an image.
    - Techniques like color correction and histogram equalization can be applied to enhance color features.
12. **Edge Enhancement:**
    - Emphasizing edges or boundaries within an image.
    - Edge enhancement techniques, such as gradient-based methods, help highlight important structural information.

3-2 Proposed image Enhancement Techniques.

3-2-1 Histogram Stretching.

Histogram stretching, a key technique in image processing, aims to improve image contrast by expanding the range of pixel intensities. It works by analyzing the distribution of intensity values in an image and then adjusting these values to cover a wider range, thus enhancing the visual quality.

This process involves mapping the original intensity values to a new range, typically from 0 to 255 for an 8-bit grayscale image. By stretching or compressing the intensity values while maintaining their relationships, details become more discernible, and the overall contrast of the image is improved.

Histogram stretching methods like Histogram Equalization, Standard Deviation Stretching, and Minimum-Maximum Normalization offer distinct ways to achieve this enhancement. They vary in their approaches, from redistributing pixel values for uniformity (Histogram Equalization) to adjusting contrast based on statistical measures (Standard Deviation Stretching) or scaling intensities to fit within a specified range (Minimum-Maximum Normalization). These methods are vital for various applications, aiding in tasks like medical imaging analysis, satellite imagery interpretation, and general image enhancement in photography.

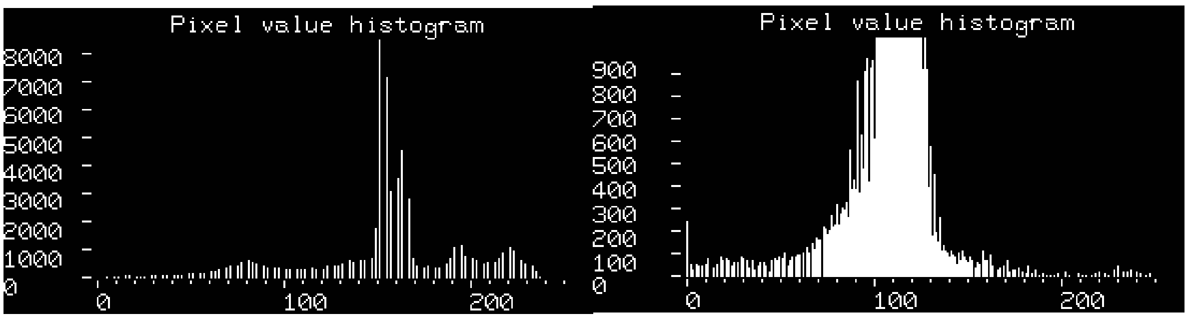


Figure 2 Pixel Value with and Without Histogram.

3-1-1 Histogram Equalization.

Histogram Equalization is a technique used in image processing to enhance contrast by redistributing the intensity values in an image's histogram. Here are the key aspects and functions of histogram equalization:

1. **Contrast Enhancement:** The primary goal is to improve the visual quality of an image by spreading out the intensity values more evenly across the entire histogram. This process tends to make the image more vivid and detailed.
2. **Histogram Analysis:** It begins with analyzing the histogram of the image to understand the distribution of pixel intensities.
3. **Cumulative Distribution Function (CDF):** Histogram equalization works by calculating the Cumulative Distribution Function of the histogram. The CDF represents the cumulative frequencies of pixel intensities in the image.
4. **Intensity Transformation:** Based on the CDF, a mapping function is created to reassign pixel values. This transformation effectively stretches the histogram to fill the entire available intensity range.
5. **Enhanced Contrast:** After the equalization process, the resulting image exhibits improved contrast, making previously subtle details more visible.
6. **Uniform Distribution:** Ideally, histogram equalization aims to achieve a uniform distribution of pixel intensities, ensuring a balanced representation of brightness levels.
7. **Applications:** It finds applications in various fields, such as medical imaging for better analysis of X-rays or MRI scans, in satellite imagery interpretation, and in general photography to enhance image quality.
8. **The equation for Histogram Equalization:** involves the calculation of a transformation function that maps the original pixel intensities to their adjusted values based on the cumulative distribution function (CDF) of the image histogram.

The equation for Histogram Equalization involves the calculation of a transformation function that maps the original pixel intensities to their adjusted values based on the cumulative distribution function (CDF) of the image histogram.

1. **Histogram Calculation:** Compute the histogram $H(rk)$, where rk represents the intensity level.
2. **Normalized Histogram**: Calculate the probability density function $p(r_k) = \frac{H(rk)}{MN}$ where M is the image width, N is the image height, and MN is the total number of pixels.
3. **Cumulative Distribution Function (CDF):** Compute the cumulative sum of probabilities up to intensity level r_k to obtain the CDF, denoted as CDF (r_k): $CDF(r_k) = \sum_{l}^{k} = 0\, p(r_j)$
4. **Transformation Function:** Now, the transformation function T (r_k) is formulated using the CDF: $T(r_k) = round((L-1) \times CDF(r_k))$ where round ensures that the calculated values are rounded to the nearest integer. This function maps the original pixel intensity r_k to its new value.
5. **Output Image:** Apply the transformation function T (r_k) to all pixels in the image to generate the equalized image.

The resulting T (r_k) acts as a mapping function that adjusts the intensity values of the original image to achieve histogram equalization, enhancing the contrast by redistributing the pixel intensities more uniformly across the available intensity range.

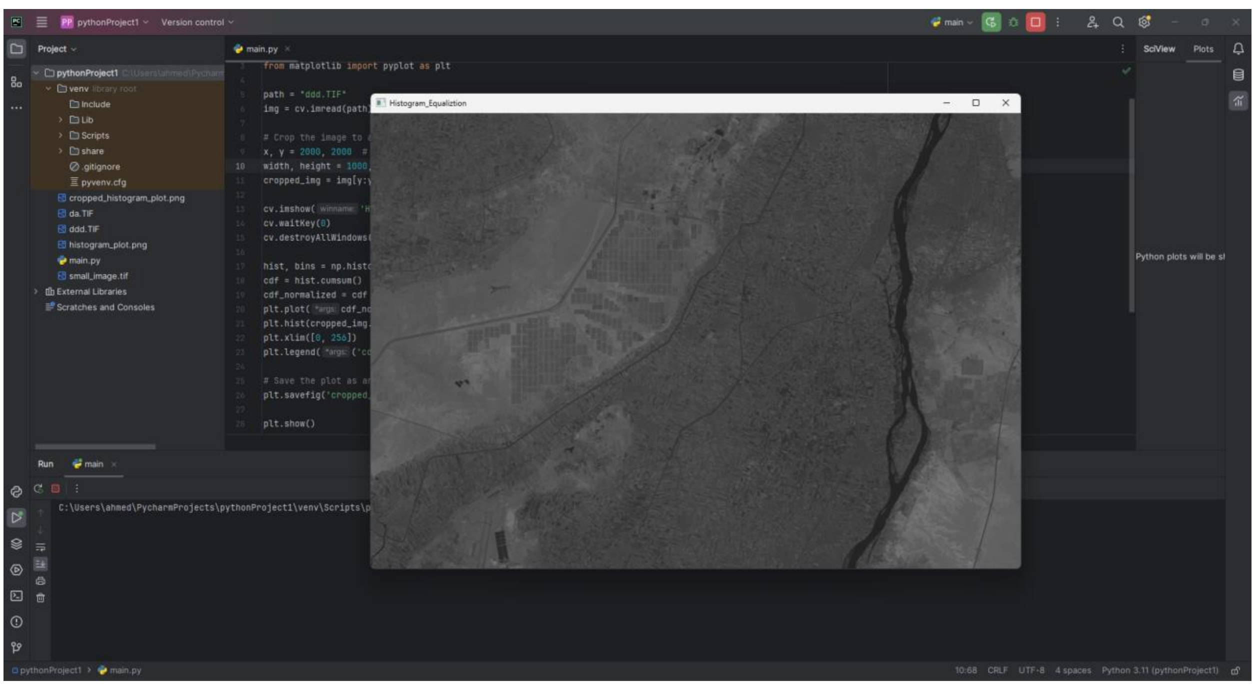


Figure 3 Output of Histogram Equalization.

3-1-2 Standard Deviation Stretching.

The Standard Deviation Stretching method in histogram stretching involves adjusting image contrast based on statistical measures like the mean and standard deviation of pixel intensities. This technique aims to expand or compress the range of pixel values to match a desired statistical distribution, often centered around the mean with a specific standard deviation.
The process starts with calculating statistical parameters such as the mean and standard deviation from the image histogram. Then, these parameters are utilized to normalize and scale the pixel intensities accordingly. By stretching or compressing the intensity values while maintaining their statistical characteristics, the contrast in the image is modified.
Standard Deviation Stretching provides a controlled means of adjusting contrast, making it particularly useful when fine-tuning specific details in an image is crucial. This method allows for tailored adjustments based on statistical properties, offering a way to enhance or diminish contrast while considering the distribution of pixel intensities.

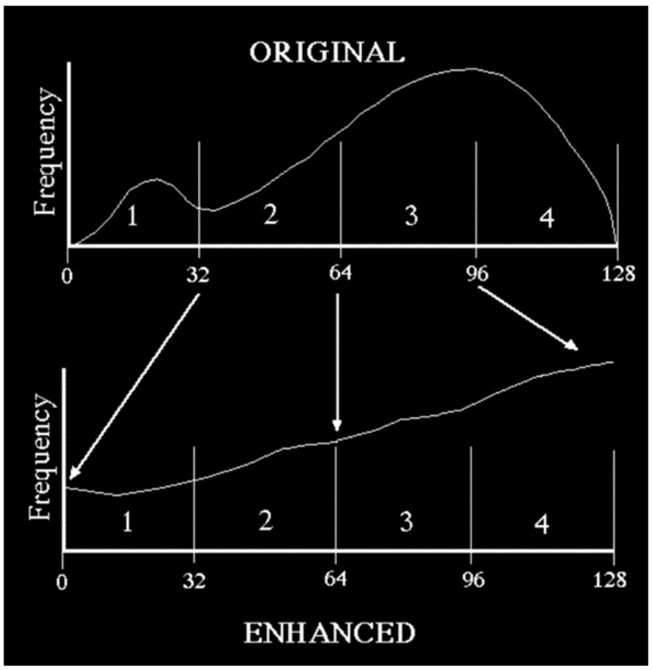


Figure 4 Standard Deviation Stretching Graph.

The equation for Standard Deviation Stretching involves scaling and adjusting the pixel intensities based on the statistical properties of the image, typically using the mean (μ) and standard deviation (σ).

Let's assume an image with pixel intensities ranging from 0 to L-1 (where L is the number of intensity levels, typically 256 for 8-bit images):

13. Calculation of Statistical Parameters:

- Calculate the mean (μ) and standard deviation (σ) of the pixel intensities in the image.

14. Normalization:

- Normalize the pixel intensities by subtracting the mean (μ) and then dividing by the standard deviation (σ): $NormalizedIntensity = \dfrac{OriginalIntensity - \mu}{\sigma}$.

15. Adjustment and Scaling:

- Scale the normalized intensities to fit within the desired range, often from 0 to L-1 (0 to 255 for 8-bit images): $AdjustedIntensity = round(\,NormalizedIntensity \times NewMaxIntensity\,)$

16. Output Image:

- Apply the adjusted intensities to all pixels in the image, generating the image with modified contrast based on the statistical properties.

This process adjusts the contrast of the image by altering the pixel intensities according to their statistical characteristics, allowing for controlled enhancements or reductions in contrast while considering the mean and standard deviation of the image's intensity distribution.

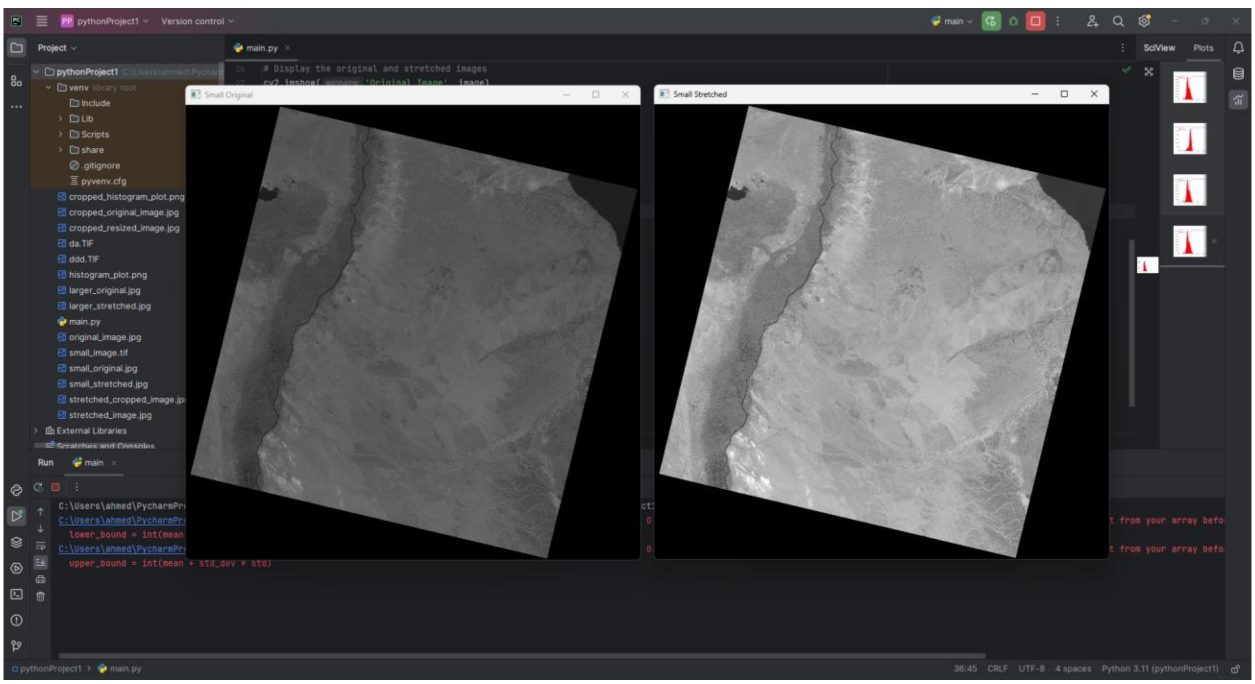


Figure 5 Output of the Standard Deviation.

3-1-3 Min-Max Stretching.

The Minimum-Maximum Stretching technique in histogram stretching aims to enhance image contrast by scaling and adjusting pixel intensities to fit within a predefined range, typically from 0 to 255 for 8-bit grayscale images. This method ensures that the entire range of available intensity values is utilized effectively, thereby improving image quality.

The process begins by determining the minimum and maximum pixel intensities present in the image histogram. These values define the range of intensity levels within the image.

Next, the intensities are linearly transformed to fit within the desired range. The formula for this transformation involves scaling the original intensities proportionally to cover the full range of values, ensuring that the minimum intensity maps to 0 and the maximum intensity maps to 255.

$$AdjustedIntensity = \frac{OriginalIntensity - MinIntensity}{MaxIntensity - MinIntensity} \times 255$$

After the transformation, the pixel intensities span the entire available range, effectively utilizing the full spectrum of brightness levels and enhancing image contrast. The Min-Max Stretching technique provides a straightforward way to adjust contrast by scaling pixel intensities, making it beneficial in various image processing applications for improving visual quality and detail visibility.

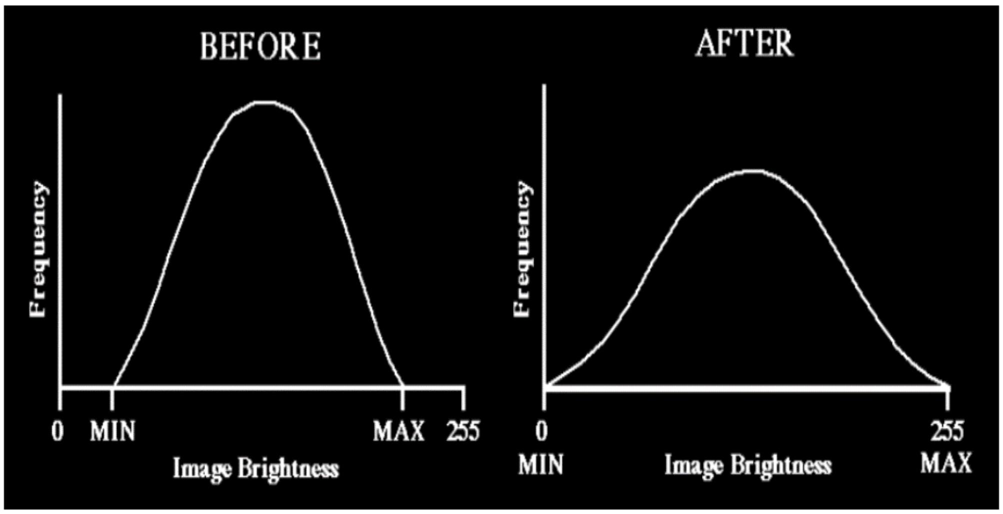


Figure 6 Min-Max Stretching Graph.

The equation for Minimum-Maximum Stretching involves scaling and adjusting pixel intensities to fit within a predefined range, often from 0 to 255 for 8-bit images. Here's the formula:

Let's assume an image with pixel intensities ranging from 0 to L-1 (where L is the number of intensity levels, typically 256 for 8-bit images):

1. **Finding Minimum and Maximum Intensities:**
    - Determine the minimum (Min Intensity) and maximum (Max Intensity) pixel intensities present in the image histogram.

2. **Normalization:**
   - Normalize the pixel intensities to a range from 0 to 1:

$$NormalizedIntensity = \frac{OriginalIntensity - MinIntensity}{MaxIntensity - MinIntensity}$$

3. **Adjustment and Scaling:**
   - Scale the normalized intensities to fit within the desired range (usually 0 to L-1 or 0 to 255 for 8-bit images): $AdjustedIntensity = round(\,NormalizedIntensity \times (\,L-1)\,)$.

This process linearly transforms the original pixel intensities to map them to a new range, ensuring that the minimum intensity value becomes 0 and the maximum intensity value becomes the maximum value of the specified range (L - 1 or 255 for 8-bit images). The rounded value ensures that the adjusted intensities are integers within the defined range.

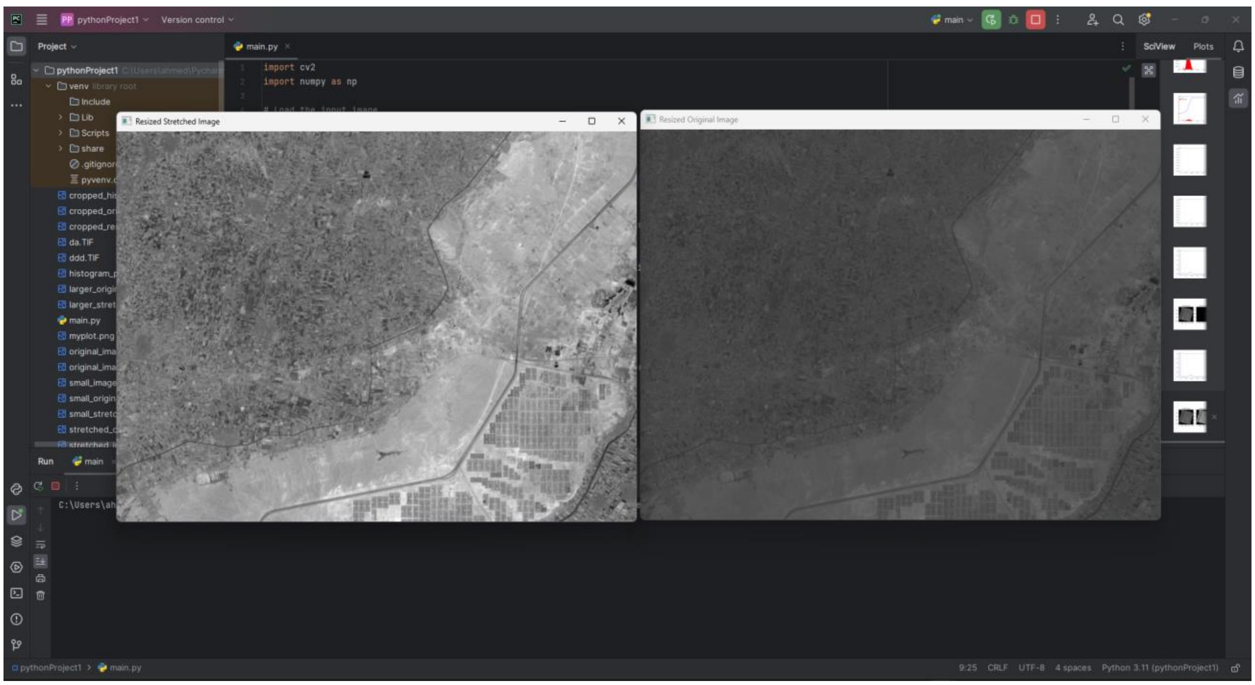


Figure 7 Output of the Min-Max Stretching.

3-3 Images Filter.

Satellite image filtering involves a range of techniques applied to satellite imagery to enhance quality, reduce noise, and extract pertinent information for analysis. These filters work by modifying pixel values based on mathematical operations or specific algorithms. They play a vital role in preprocessing satellite data, aiding in better interpretation and analysis.

These filters can improve image clarity by smoothing, highlighting edges, reducing noise, or enhancing contrast. Smoothing filters help create cleaner images by averaging neighboring pixel values, while edge detection filters emphasize abrupt intensity changes, aiding in feature extraction. Additionally, contrast enhancement filters widen intensity ranges, making details more discernible.

Satellite image filters are indispensable in various fields. In agriculture, they assist in monitoring crop health. In urban planning, they aid in infrastructure analysis. Environmental monitoring benefits from these filters by tracking changes in landscapes. Moreover, in disaster management, these filters contribute to assessing affected areas efficiently. Overall, these techniques help researchers and analysts derive insights and make informed decisions from satellite data, crucial for numerous applications across diverse domains.

3-3-1 Median Filter.

In satellite image processing, the median filter serves as a valuable tool for noise reduction and enhancement. Satellite images often suffer from various types of noise, including speckle noise in radar images or random noise introduced during image acquisition. The median filter effectively mitigates such noise without significantly blurring the edges or fine details present in the images.

By replacing each pixel's value with the median value within a specified neighborhood, the median filter helps in reducing noise while preserving important image features. In satellite imagery, where precise information extraction is crucial, maintaining the integrity of edges and details is essential. The median filter achieves this by effectively addressing noise without compromising the overall quality of the image.

Its robustness against outliers and ability to preserve edge information make it a preferred choice for satellite image processing. It's particularly useful in remote sensing applications for land cover classification, feature extraction, and environmental monitoring where clear, noise-free imagery is essential for accurate analysis and decision-making. The median filter's capability to enhance satellite images by reducing noise while retaining critical details makes it an invaluable tool in improving data quality for various applications in satellite-based research and analysis.

1	**4**	**0**	**1**	**3**	**1**
2	2	4	2	2	3
1	0	1	0	1	0
1	2	1	0	2	2
2	5	3	1	2	5
1	1	4	2	3	0

1	4	0	1	3	1
2	1	1	1	1	3
1	1	1	1	2	0
1	1	1	1	1	2
2	2	2	2	2	5
1	1	4	2	3	0

*Figure 8 Median Filtering using 3*3 Sampling Matrix.*

The equation for the median filter involves replacing each pixel's value with the median value within a defined neighborhood or kernel. Let's consider a square kernel of size n×n pixels:

1. **Selecting the Neighborhood:**
   - For each pixel in the image, create a window centered around that pixel with a size $n \times n$ (if n is odd) or $(n+1) \times (n+1)$ (if n is even).
2. **Sorting Pixel Values:**
   - Collect the pixel values within the defined window and arrange them in ascending order.
3. **Determining the Median:**
   - Select the middle value from the sorted list of pixel intensities. If the number of pixels in the window is even, take the average of the two middle values.
4. **Replacement:**
   - Replace the original pixel value with the computed median value.

The formula for the median filter doesn't involve mathematical operations in the same way as some other filters. Instead, it relies on sorting and selecting the middle value within the defined neighborhood for each pixel, making it computationally efficient for noise reduction while preserving image details.

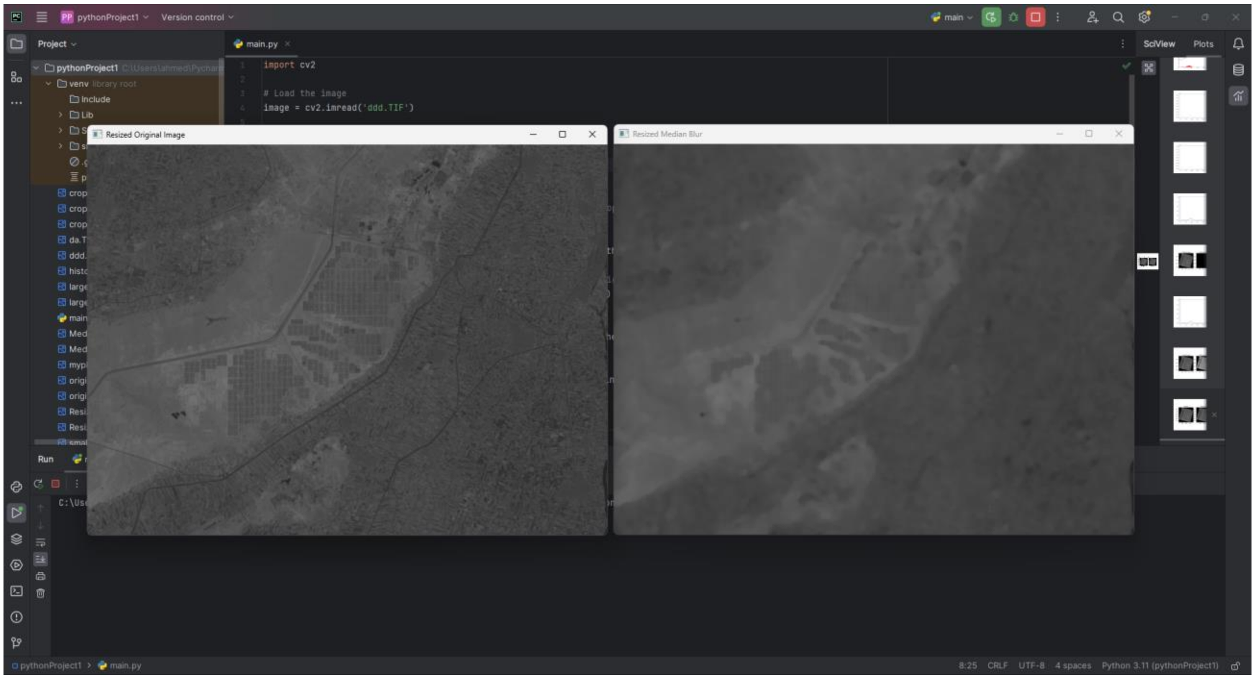


Figure 9 Output of the Median Filter.

3-3-2 High Pass Filter.

The High Pass Filter is a spatial domain image processing technique used to enhance or extract high-frequency components, such as edges and fine details, from an image while attenuating or eliminating low-frequency components, including smooth areas or gradual transitions in intensity.

This filter works by emphasizing rapid intensity changes or edges in an image while suppressing or removing slow variations or homogeneous regions. It achieves this by subtracting a smooth version of the image from the original, effectively extracting the high-frequency information.

Mathematically, the high pass filter is implemented by convolving the image with a kernel, such as the Laplacian or Sobel operator. These kernels highlight areas of high gradient or change in intensity, accentuating edges, and fine structures within the image. The resulting image emphasizes features with significant intensity variations while reducing or eliminating smoother areas.

High pass filtering finds applications in edge detection, sharpening images, and extracting texture information from images. It aids in tasks like object detection, image enhancement for computer vision applications, and analysis where accentuating high-frequency components is essential for precise interpretation and understanding of the image content.

-1	-1	-1
-1	8	-1
-1	-1	-1

Figure 10 High Pass Filter Matrix.

The equation for a basic high pass filter can be implemented using convolution with a kernel represent the original image intensity at $HighPass(x,y) = I(x,y) - (I * K)(x,y)$.

Here, (I * K) represents the convolution operation between the image (I) and the high pass filter kernel (K). The result of this convolution is a smoothed version of the original image.

The choice of the high pass filter kernel (K) depends on the specific requirements of the application. Common choices include the Laplacian kernel or Sobel operator, both of which emphasize high-frequency components by detecting rapid changes in intensity.

The high pass filter operation effectively subtracts the smoothed version of the image from the original, emphasizing edges and fine details. The resulting image highlights areas with significant intensity variations, making it useful for tasks like edge detection and image sharpening in image processing applications.

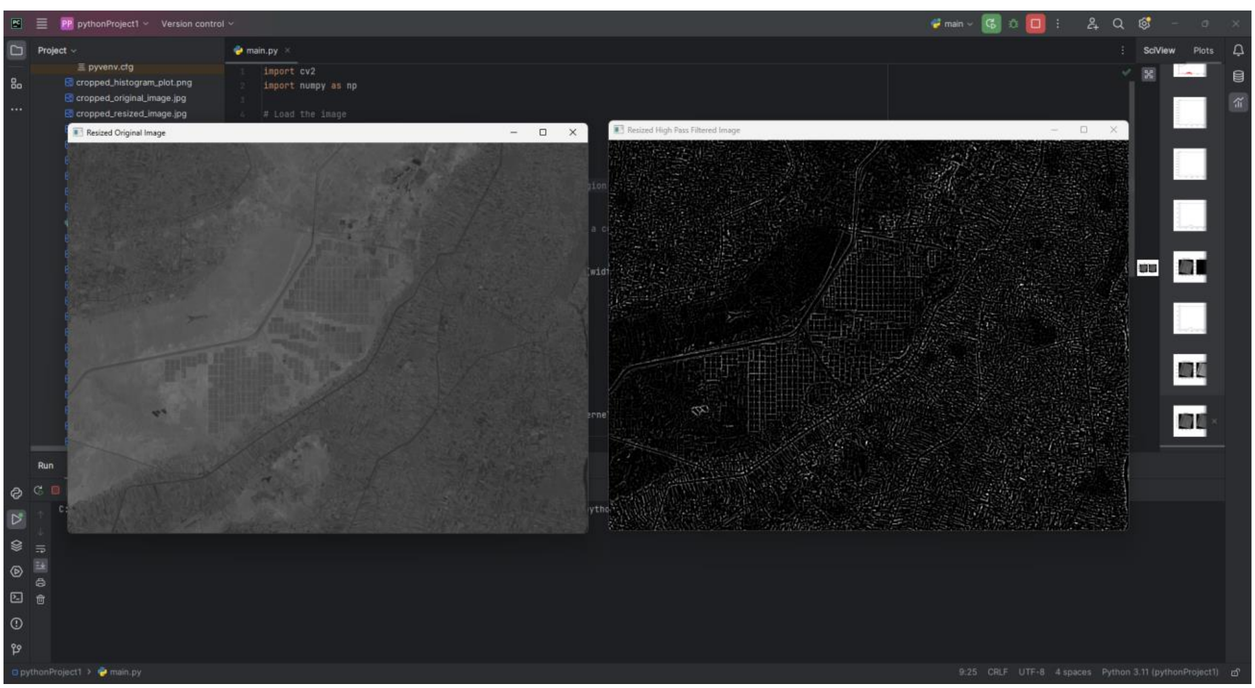


Figure 11 The Output for High Pass Filter.

3-3-3 Low Pass Filter.

Low pass filters in satellite image processing are essential tools used to reduce noise, blur or smooth images, and eliminate high-frequency components while preserving low-frequency information. These filters work by attenuating or suppressing rapid changes in intensity, such as noise or fine details, allowing for a clearer representation of broader features and overall trends within the image.

By convolving the image with a low pass filter kernel, such as a Gaussian or averaging filter, high-frequency noise is diminished, and fine details are smoothed out. This process helps in improving the image quality, reducing the effects of noise inherent in satellite imagery, and facilitating better interpretation and analysis.

In satellite image processing, low pass filters play a crucial role in preprocessing data. They aid in enhancing the signal-to-noise ratio, emphasizing larger-scale features, and reducing the impact of small-scale variations that might not be of primary interest in certain applications. Moreover, they contribute to better visualization of general trends and patterns in satellite data, making them valuable in fields such as land cover classification, environmental monitoring, and terrain analysis where a clearer, noise-free representation of broader features is essential for accurate interpretation and decision-making.

$$LPF = \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

Figure 12 Low Pass Filter Matrix.

The equation for a basic low pass filter involves convolution of the image with a kernel designed to attenuate high-frequency components while preserving low-frequency information. Let I(x,y) represent the original image intensity at position (x,y), and K be the low pass filter kernel. The convolution operation is denoted by:

$$LowPass(x, y) = (I * K)(x, y$$

Here, I∗K represents the convolution operation between the image II and the low pass filter kernel KK. The result of this convolution is a smoothed version of the original image, emphasizing low-frequency components and reducing high-frequency noise or details.

The choice of the low pass filter kernel (K) determines the filtering characteristics. Common choices include Gaussian filters or average filters. Gaussian filters, for instance, emphasize smoother transitions and suppress high-frequency noise more uniformly across the image.

The low pass filter operation is pivotal in satellite image processing to improve image quality by reducing noise and emphasizing broader features while minimizing the impact of fine-scale variations that might not be relevant for certain analysis or interpretation purposes.

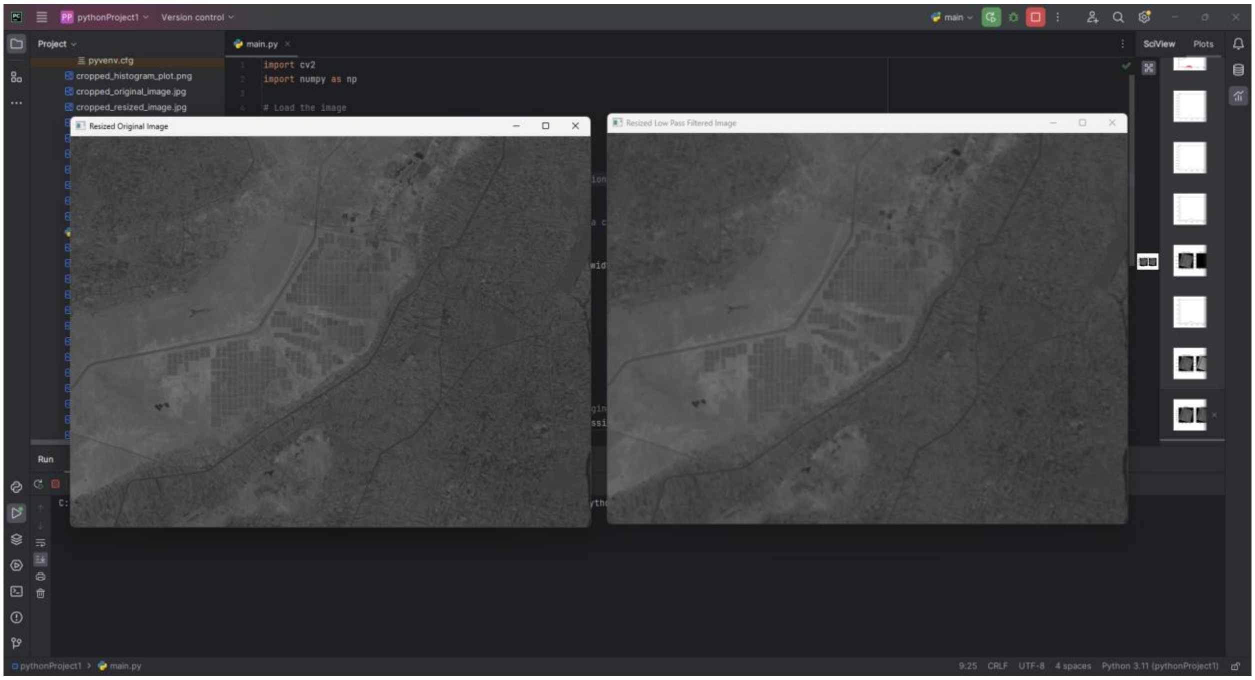


Figure 13 The Output of Low pass filter.

3-3-4 Directional Filter (Sobel Filter).

The Sobel filter, a type of directional filter used in satellite image processing, focuses on edge detection by emphasizing changes in intensity across an image. It comprises two kernels, one for detecting horizontal edges and another for vertical edges, operating by convolving these kernels with the image to highlight edge features in specific directions.

The Sobel filter calculates gradients by applying a weighted average to neighboring pixels along the x and y axes, accentuating areas of rapid intensity change. This process effectively identifies edges, delineating boundaries between different regions or objects within the image.

In satellite image analysis, the Sobel filter aids in feature extraction, object detection, and edge enhancement. It helps in identifying distinct features such as roads, buildings, or terrain variations by emphasizing their boundaries and edges. By highlighting these details, the filter assists in interpreting and analyzing satellite data for various applications like land cover classification, urban planning, and environmental monitoring.

The Sobel filter's directional approach in edge detection proves valuable in satellite image processing, enabling the extraction of meaningful features and aiding in the accurate interpretation and analysis of satellite imagery for diverse applications in remote sensing and geospatial analysis.

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \qquad \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$(A) \qquad\qquad (B)$$

Figure 14 Soble Filter Matrix.

The Sobel filter consists of two kernels, one for detecting horizontal edges and another for vertical edges. These kernels are convolved with the image to detect edges in specific directions. Let I(x,y) represent the original image intensity at position (x,y), and Kx and Ky be the Sobel filter kernels for horizontal and vertical edges respectively.

The horizontal Sobel kernel (Kx) typically looks like:

$$K_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$ The vertical Sobel kernel (Ky) typically looks like: $$K_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

The Sobel filter operation involves convolving the image with both kernels separately:

$$SobelHorizontal(x,y) = (I * K_x)(x,y) \quad SobelVertical(x,y) = (I * K_y)(x,y)$$

These convolutions result in two images: one emphasizing horizontal edge and another highlighting vertical edges. The magnitude and direction of the gradient at each pixel can be computed using these horizontal and vertical edge images, aiding in edge detection and feature extraction in satellite image processing.

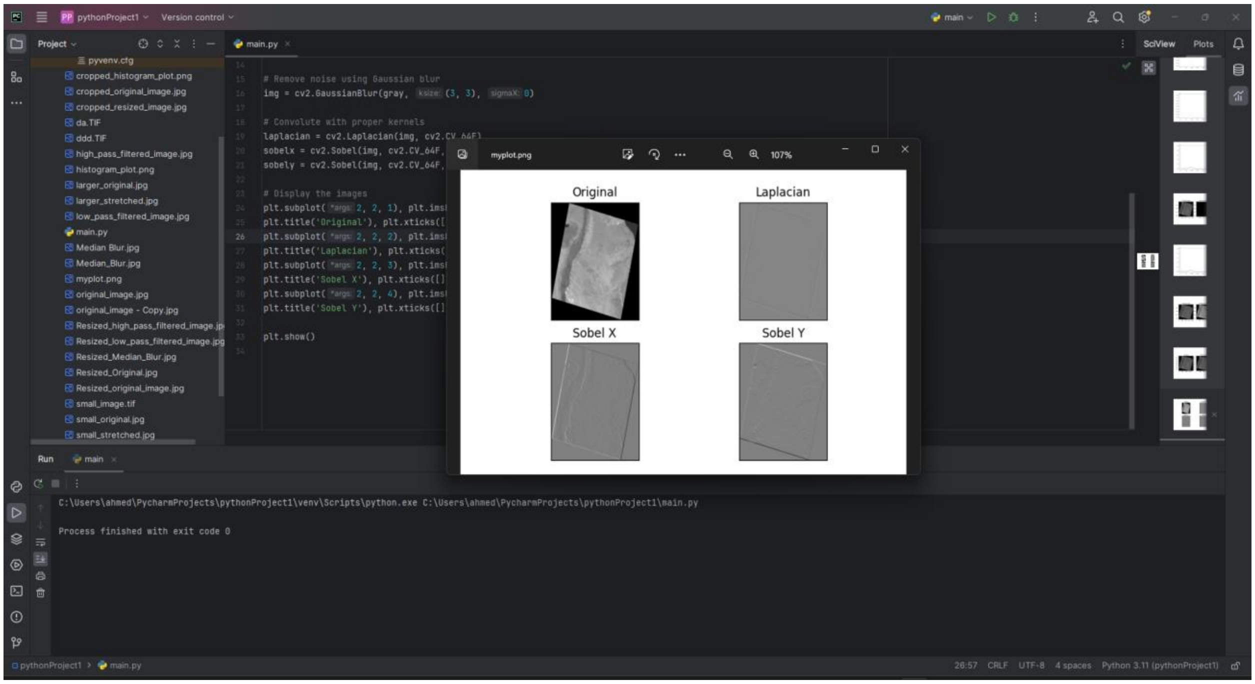


Figure 15 The output of Soble Filter.

3-4 Image Transformation.

Satellite image transformation refers to refining raw satellite data for improved interpretation and application. Initially, the data undergoes preprocessing steps, such as calibration and correction, to remove distortions and enhance quality. Techniques like contrast enhancement and filtering are then applied to improve visibility and highlight specific features within the imagery.

Further processing involves extracting valuable information, like land cover types or urban areas, using methods such as edge detection and object-based analysis. Fusion techniques combine data from different sensors or sources to create comprehensive, informative images useful for various applications. Dimensionality reduction methods, like PCA, streamline complex data, aiding computational efficiency while retaining essential information.

One crucial application involves change detection, where multiple images from different times are compared to identify alterations in the landscape, such as urban expansion or environmental changes. These transformations enable better decision-making across fields like agriculture, environmental monitoring, and disaster response by providing clearer, more insightful satellite imagery.

3-4-1 Principal Component Analysis (PCA).

Principal Component Analysis (PCA) is a powerful technique in satellite image processing used for dimensionality reduction, noise reduction, and feature extraction. In this context, PCA helps in managing the vast amount of data generated by satellite sensors by identifying patterns and reducing redundancy.

In satellite image processing, PCA operates by transforming the original high-dimensional data, representing pixel values across multiple spectral bands, into a smaller set of uncorrelated variables known as principal components. These components capture the most significant variability in the data.

By identifying the directions of maximum variance in the satellite image data, PCA enables the extraction of key information while discarding less critical or redundant information. This reduction in dimensions not only aids in visualization but also accelerates subsequent analysis and interpretation.

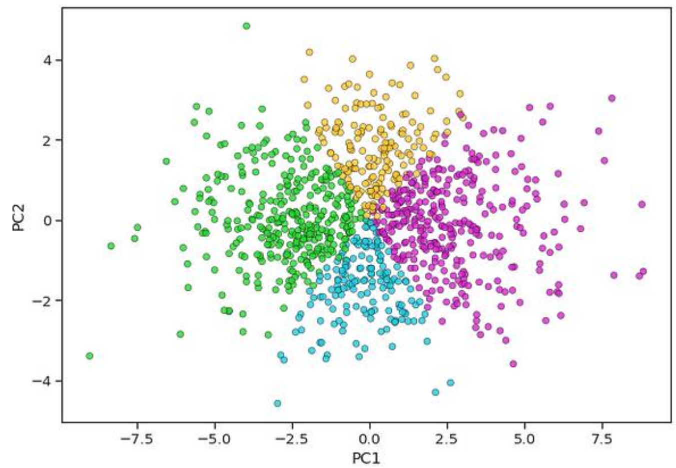


Figure 16 Scatterplot after PCA reduced from 3-dimensions to 2-dimensions.

The equation for Principal Component Analysis (PCA) involves several steps, but the fundamental mathematical operation revolves around eigenvalue decomposition or singular value decomposition of the covariance matrix of the data. Let's denote the data matrix as X with dimensions n×p, where n represents the number of observations or samples, and pp represents the number of variables (features).

1 **Data Standardization:** The first step is to standardize the data by subtracting the mean and dividing it by the standard deviation for each variable, ensuring that all variables are on a similar scale.

2 **Covariance Matrix:** The covariance matrix C is computed from the standardized data matrix X. The covariance between variables i and j is calculated as:

$$C = \frac{1}{n-1}(X - \overline{X})^T(X - \overline{X})$$

3 **Eigenvalue Decomposition:** PCA involves finding the eigenvectors and eigenvalues of the covariance matrix C. The eigenvectors (V) and eigenvalues (Λ) satisfy the equation:

$$C.V = V.A$$

4 **Selecting Principal Components:** The eigenvectors in V are sorted in descending order based on their corresponding eigenvalues in Λ. These eigenvectors represent the principal components.

5 **Projection onto Principal Components:** To transform the data onto the new lower-dimensional space formed by the principal components, you compute the matrix multiplication: $Y = X.\overline{V}$

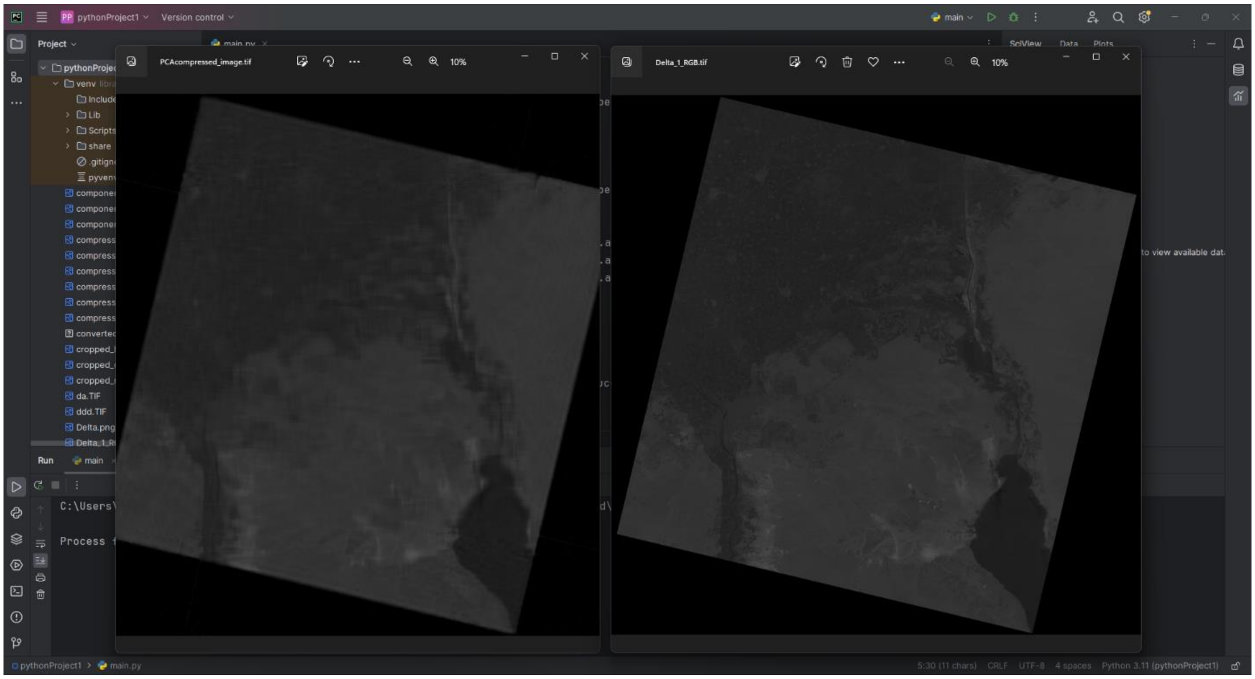


Figure 17 The Output of Using Principal Component Analysis (PCA) Function.

3-4-2 Independent Component Analysis (ICA).

Independent Component Analysis (ICA) is a statistical method used for separating a multivariate signal into additive, independent components. It assumes that the observed data is a linear combination of independent source signals, each with its own statistical distribution. The goal of ICA is to uncover these underlying independent components from the observed mixtures without prior information about the sources.

The process of ICA involves finding a remixing matrix that, when multiplied with the observed data, separates it into independent components. The algorithm iteratively adjusts this remixing matrix to maximize the statistical independence of the resulting components.

ICA is a powerful technique for uncovering hidden factors or sources from mixed observations, providing a valuable tool for signal processing and data analysis where the sources are not explicitly known or easily separable.

Independent Component Analysis (ICA) aims to decompose a multivariate signal into statistically independent components. The core equation involves finding a remixing matrix that separates observed mixtures into their original independent sources.

Given an observed mixed signal matrix X of dimensions n×p, where n represents the number of observations or samples, and pp represents the number of variables (mixed sources).

The equation for ICA can be represented as follows:

1. **Assumptions:**
   - $X = AS$, where:
     - X is the observed mixed signal matrix.
     - A is the mixing matrix (unknown).
     - S is the source matrix of independent components.
2. **Objective:**
   - Find the remixing matrix WW such that $Y = XW$ yields independent components Y.
3. **ICA Objective Function:**
   - The ICA algorithm minimizes the statistical dependence or maximizes the statistical independence of the components.
   - One common measure used for non-Gaussian sources is maximizing non-Gaussian or negentropy.
4. **Steps:**
   - ICA typically employs optimization techniques like gradient descent to iteratively update the remixing matrix W to achieve independence in the transformed components.
5. **Matrix Representation:**
   - X is the observed mixed signal matrix.
   - A is the mixing matrix.
   - S is the source matrix.
   - W is the remixing matrix.
   - Y is the matrix of independent components.

3-4-3 Normalized Difference Vegetation Index (NDVI).

Normalized Difference Vegetation Index (NDVI) transformation is a widely used technique in remote sensing and satellite imagery analysis to assess vegetation health and density. NDVI calculates a numerical indicator that quantifies vegetation abundance by analyzing the difference between near-infrared (NIR) and red light reflected by vegetation.

The formula for NDVI is: $NDVI = \frac{(NIR - Red)}{(NIR + Red)}$

Here's a brief overview of the NDVI transformation:

1. **Input Data:** NDVI is computed using satellite images captured in the red and near-infrared bands. Vegetation strongly absorbs red light and reflects near-infrared light, providing the basis for NDVI calculation.
2. **Calculation:** NDVI is calculated on a pixel-by-pixel basis for an entire image using the formula mentioned above. Higher NDVI values indicate healthier and more dense vegetation, while lower values correspond to less vegetation cover.
3. **Interpretation:** NDVI values typically range from -1 to 1, although they are often presented from 0 to 1 in practice. Healthy vegetation usually yields values closer to 1, as they reflect more NIR and absorb more red light. Bare soil or non-vegetated areas have lower NDVI values.
4. **Applications:** NDVI transformations are widely used in various fields, including agriculture, forestry, environmental monitoring, and land-use planning. It helps in assessing vegetation health, detecting changes in plant growth, estimating crop yields, and identifying areas affected by drought or other environmental stresses.
5. **Mapping and Visualization:** NDVI maps derived from satellite images provide valuable insights into spatial variations in vegetation cover, allowing for the identification of areas with thriving vegetation and those needing attention due to stress or degradation.

NDVI transformation is a powerful tool for monitoring and understanding vegetation dynamics over large areas, aiding in various applications related to agriculture, ecosystem analysis, and environmental management.

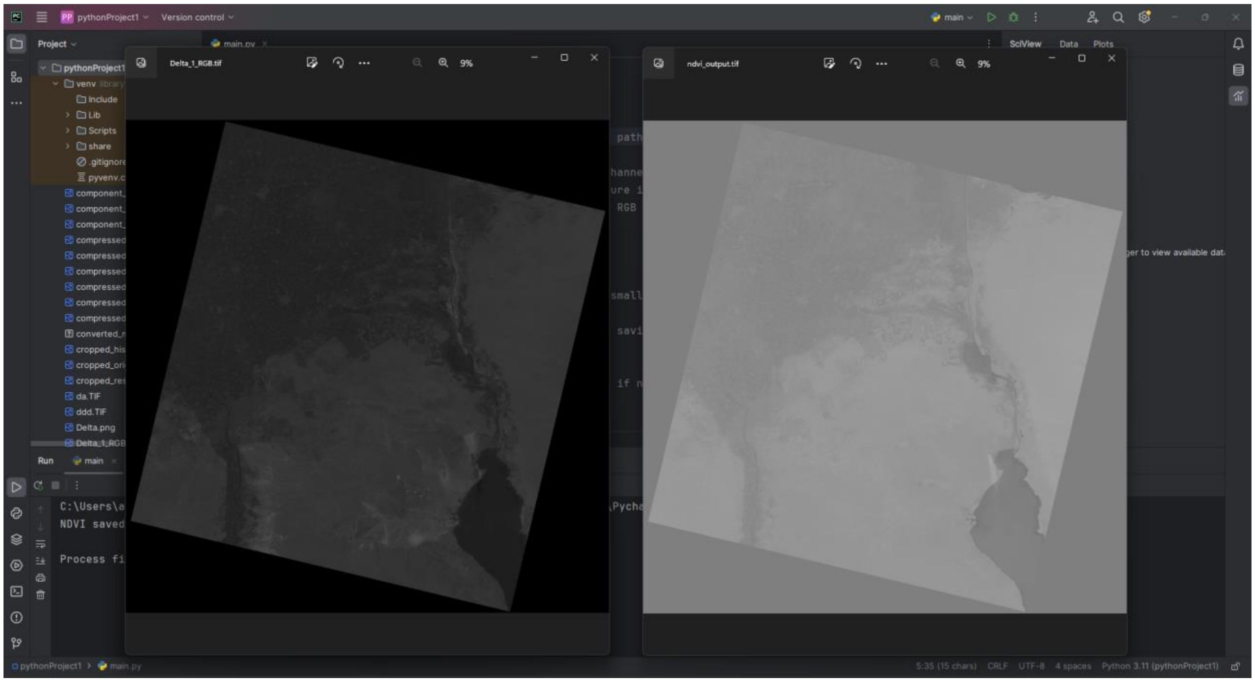


Figure 18 The Output of Using Normalized Difference Vegetation Index (NDVI).

3-4-4 Minimum Noise Fraction (MNF).

The Minimum Noise Fraction (MNF) transformation is a technique used in satellite image processing to reduce noise and enhance the information content of hyperspectral or multispectral data. It aims to decorate the bands in an image and reorient them to represent information in decreasing order of significance.

Here's a concise overview of the MNF transformation:

1. **Noise Reduction:** MNF is designed to reduce noise and extract meaningful information from satellite images. It identifies the principal components that contain the most significant information while minimizing the impact of noise present in the data.
2. **Decorrelation:** MNF transforms the original bands of the image into a set of new uncorrelated variables or components. These components are ordered by their significance, with the first few components capturing the essential information present in the image.
3. **Statistical Approach:** MNF employs a statistical approach to identify the sources of variation in the data. It separates the noise components from the signal components, allowing for better interpretation and analysis of the image data.
4. **Feature Extraction:** MNF transformation enables the extraction of meaningful features from satellite imagery. These features could represent various aspects like land cover types, vegetation health, geological formations, or other relevant information present in the image.

5. **Applications:** MNF-transformed images find applications in various fields such as agriculture, environmental monitoring, geological surveys, and land-use planning. The enhanced images aid in better understanding and decision-making by highlighting important features and reducing noise.

   MNF transformation is particularly useful in dealing with the challenges of multispectral or hyperspectral data, enhancing the utility of satellite imagery by reducing noise and emphasizing relevant information. It serves as a valuable preprocessing step, facilitating better analysis and interpretation of satellite data for numerous applications in remote sensing and geographical studies.

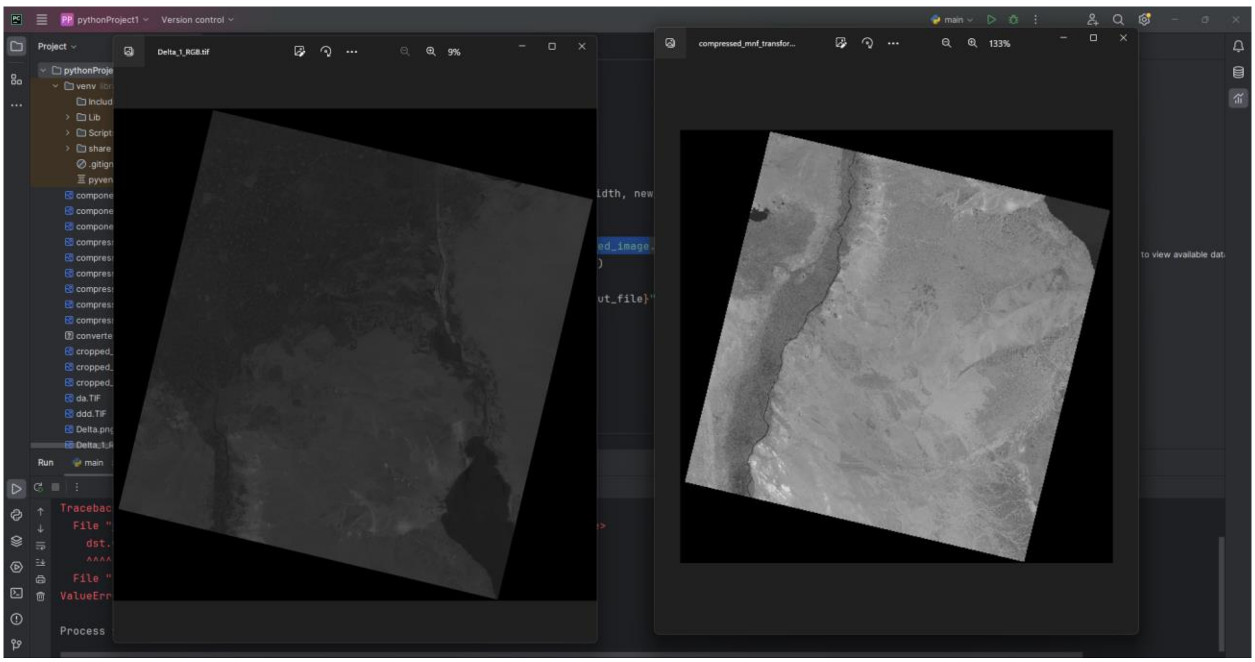


Figure 19 The Output of Using Minimum Noise Fraction (MNF).

3-5 Super Resolution.

Satellite image super-resolution is a cutting-edge technology that addresses the demand for high-quality, detailed visual information obtained from Earth observation satellites. As satellites capture vast regions of the Earth's surface, the images they produce often face challenges related to limited spatial resolution, causing a loss of fine details. Super-resolution techniques aim to enhance the clarity and sharpness of these satellite images by reconstructing higher-resolution versions from their lower-resolution counterparts.

The process involves employing advanced algorithms and computational methods to extrapolate missing information and improve the overall visual fidelity of satellite imagery. This technology has significant implications across various fields, including environmental monitoring, agriculture, urban planning, disaster management, and defense. By providing finer details and improved spatial accuracy, satellite image super-resolution enables more precise analysis, interpretation, and decision-making, empowering industries and researchers to extract valuable insights from Earth observation data. The continuous refinement of super-resolution techniques contributes to the ongoing evolution of satellite imaging capabilities, fostering advancements in our understanding of the Earth's dynamic and complex landscapes.

3-5-1 Enhanced Super-Resolution Generative Adversarial Networks (ESRGAN 2X).

Enhanced Super-Resolution Generative Adversarial Networks (ESRGAN) represent a notable advancement in the realm of satellite image super-resolution, leveraging state-of-the-art deep learning techniques to enhance the quality of satellite imagery. ESRGAN builds upon the foundation of traditional Super-Resolution Generative Adversarial Networks (SRGAN) by introducing enhancements that significantly improve the perceptual quality of the reconstructed images.

ESRGAN incorporates a unique architecture, combining the power of Generative Adversarial Networks (GANs) with sophisticated feature extraction and refinement mechanisms. The GAN framework involves a generator and a discriminator working in tandem. The generator aims to produce high-resolution images from low-resolution inputs, while the discriminator evaluates the authenticity of the generated images against real high-resolution samples.

$$D_{RA}\left(X_r, X_f\right) = \sigma(C(R) - \sum(C(F)]) \rightarrow 1$$

$$D_{RA}\left(X_r, X_f\right) = \sigma(C(F) - \sum(C(R)]) \rightarrow 0$$

1. **Feature Extraction Mechanisms:**
   - ESRGAN employs advanced feature extraction modules to better capture and understand the intricate details present in satellite imagery.
   - These mechanisms help the model learn complex patterns and structures, enhancing its ability to reconstruct high-resolution content.

2. **Perceptual Loss Function:**
   - ESRGAN incorporates a perceptual loss function, which evaluates the visual similarity between the generated and real images based on perceptual features.
   - This ensures that the enhanced images not only exhibit higher resolution but also maintain visual coherence and realism.

3. **Adversarial Training with Improved Discriminator:**
   - The adversarial training process in ESRGAN involves an improved discriminator that effectively distinguishes between real and generated images.
   - This adversarial component encourages the generator to produce high-quality outputs that closely resemble authentic high-resolution satellite images.

4. **Post-Processing Refinement:**
   - ESRGAN introduces post-processing refinement techniques to further enhance the quality of the generated images.
   - These refinements contribute to reducing artifacts and improving overall visual fidelity.

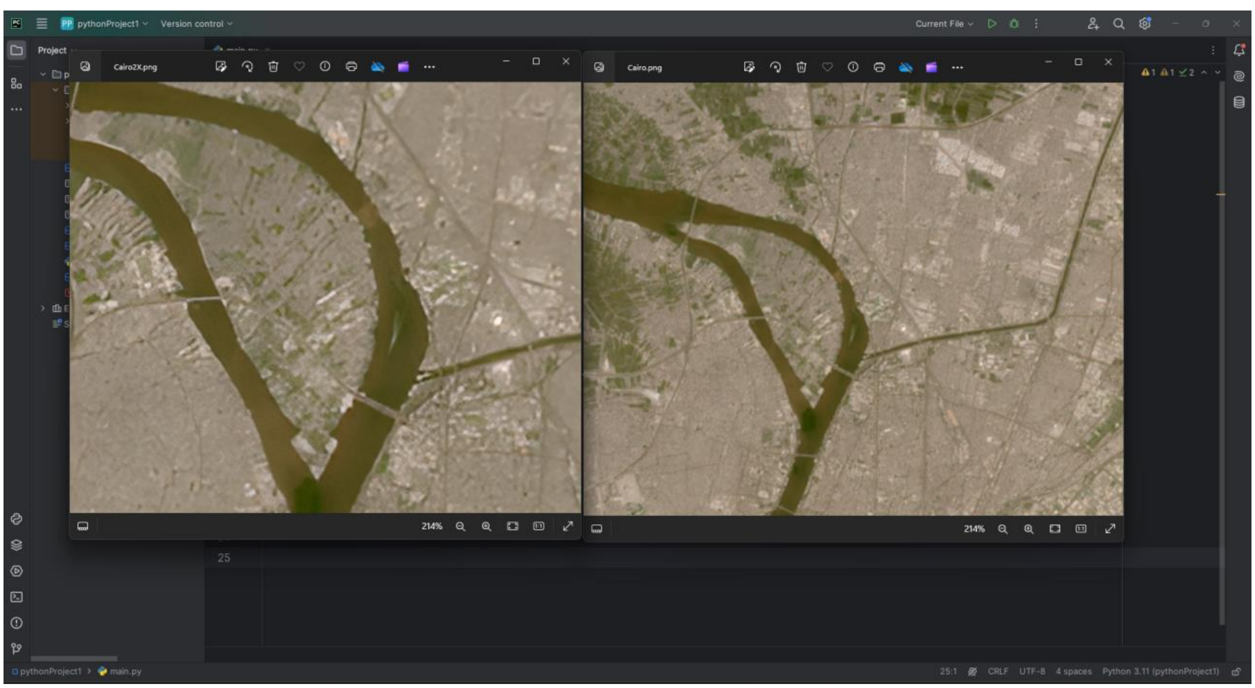


Figure 20 The Output of Using EsrGAN 2X.

3-5-2 Generative Adversarial Networks (GAN 4X).

Satellite super-resolution using Generative Adversarial Networks (GANs) is a cutting-edge technique in remote sensing and image processing that aims to enhance the spatial resolution of satellite imagery beyond the native capabilities of the imaging sensors. GANs, a type of artificial intelligence model, consist of a generator and a discriminator that work in tandem to generate high-resolution images from low-resolution inputs.

In the context of satellite imagery, GANs can be employed to address the limitations of traditional imaging systems by generating realistic and detailed high-resolution images from lower resolution sources. The process typically involves training the GAN on pairs of high and low-resolution satellite images, allowing the generator to learn the mapping between the two resolutions. Once trained, the GAN can then generate enhanced images with improved spatial details.

The advantages of using GANs for satellite super-resolution include:

1. **Preservation of Contextual Information:** GANs excel in retaining contextual information during the super-resolution process, ensuring that the generated high-resolution images maintain the overall scene context of the original low-resolution input.
2. **Realistic Image Synthesis:** GANs can generate visually realistic images, which is crucial in satellite imagery where maintaining the authenticity and integrity of the data is essential for accurate interpretation.
3. **Adaptability to Various Satellite Sensors:** GAN-based super-resolution techniques are versatile and can be adapted to different satellite sensors and platforms, allowing for the enhancement of imagery from various sources.
4. **Reduced Artifacts:** GANs can mitigate common artifacts associated with traditional upscaling methods, providing cleaner and more visually appealing high-resolution images.

$$Min_G \; Max_D \; V(D,G) = \mathrm{E}_{X\sim p\, data(x)[Log\, D\, (X)]} + \mathrm{E}_{z\sim p(z)\{Log(1-D(G(z))]}$$

where V(D,G) is the adversarial loss, λ is a weighting parameter, and L content is the content loss. The content loss can be computed using various metrics such as mean squared error (MSE) or perceptual loss, which measures the perceptual difference between images.

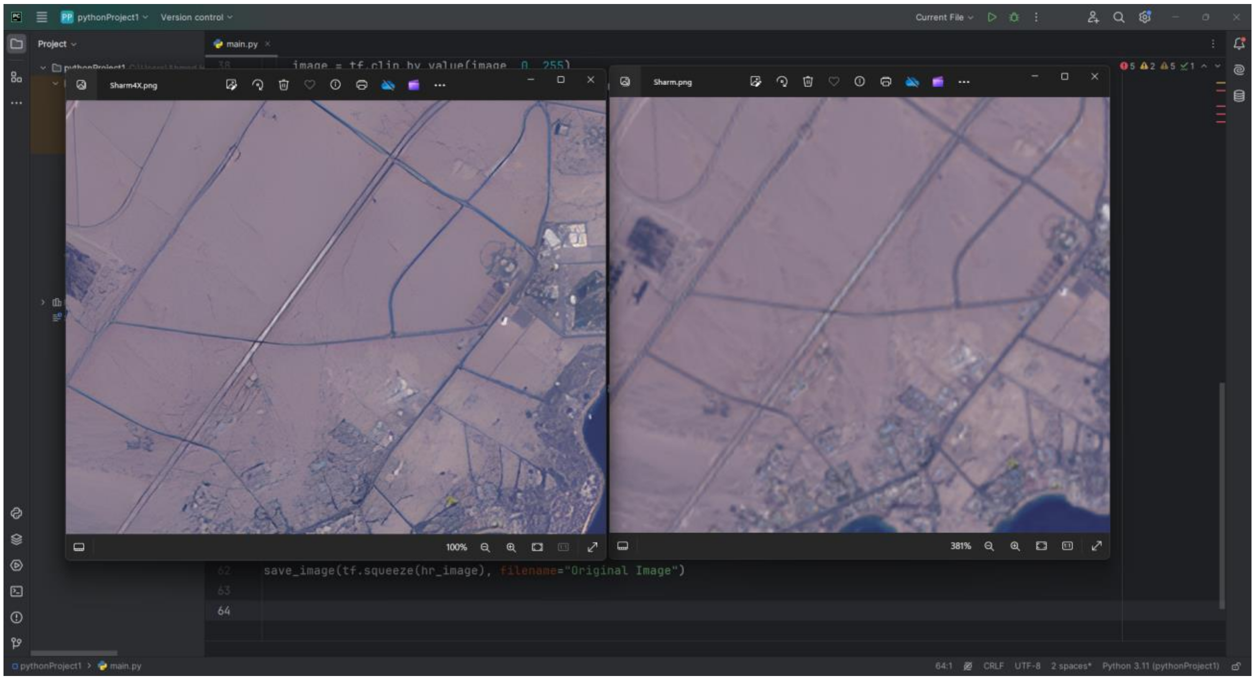


Figure 21 The Output of Using GAN 4X.

Dataset Used.

Chapter Four.

4-1 Data Used.

4-1-1 Landsat 8

Landsat characteristics.

Landsat 8 orbits the Earth in a sun-synchronous, near-polar orbit, at an altitude of 705 km, inclined at 98.2 degrees, and completes one Earth orbit every 99 minutes. The satellite has a 16-day repeat cycle. Operational Land Imager (OLI) - Built by Ball Aerospace & Technologies Corporation.

Spectral Band Number	Spectral Band Name	Wavelength (µm)	Spatial Resolution (m)
1	Coastal aerosol	0.43-0.45	30
2	Blue	0.45-0.51	30
3	Green	0.53-0.59	30
4	Red	0.64-0.67	30
5	Near-Infrared (NIR)	0.85-0.88	30
6	Shortwave-Infrared (SWIR) 1	1.57-1.65	30
7	Shortwave-Infrared (SWIR) 2	2.11-2.29	30
8	Panchromatic	0.50-0.68	15
9	Cirrus	1.36-1.38	30
10	Thermal-Infrared (TIRS) 1	10.60-11.19	100
11	Thermal-Infrared (TIRS) 1	11.50-12.51	100

Figure 22 Characteristics of Landsat 8 Spectral band.

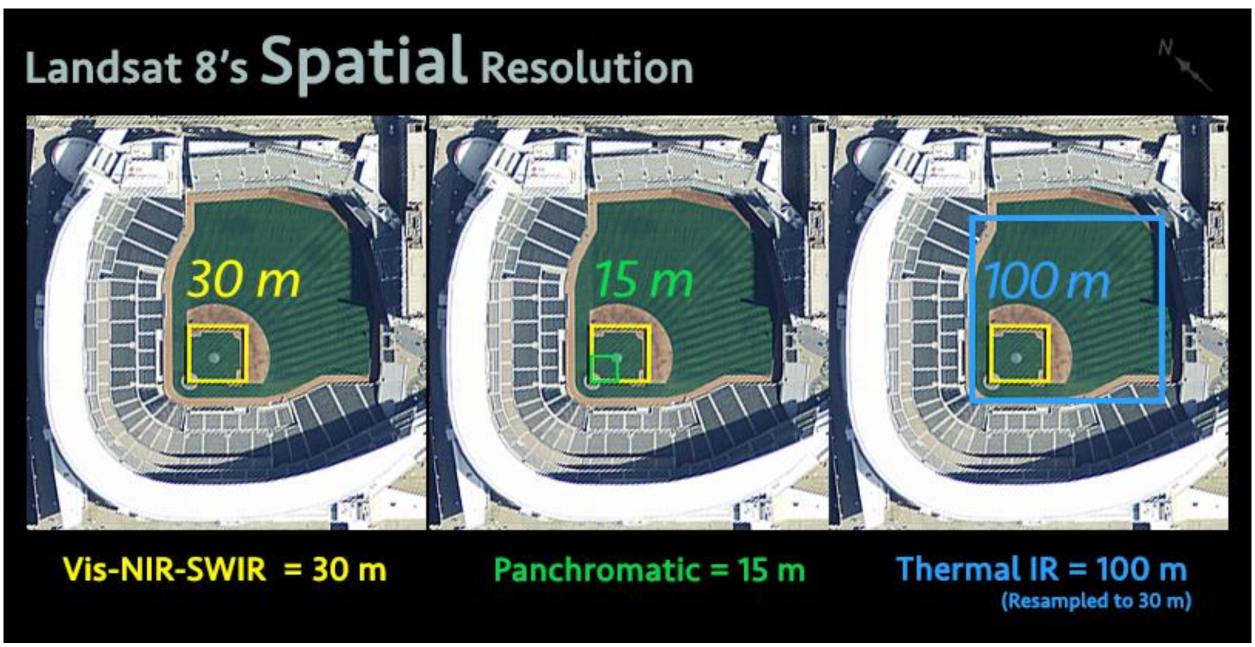


Figure 23Landsat 8 spatial resolution with 30 m, 15 m, 100m.

Landsat satellites collect data along a wide ground track that spans 185 kilometers (115 miles) but with a spatial resolution that allows them to see the human signature on the landscape. Each Landsat pixel covers a 30 by 30-meter area (98 by 98 feet), about the size of a baseball diamond. This still image shows the Landsat path over Minneapolis, the site of the 2014 Major League Baseball All Star game and reveal the individual pixels.

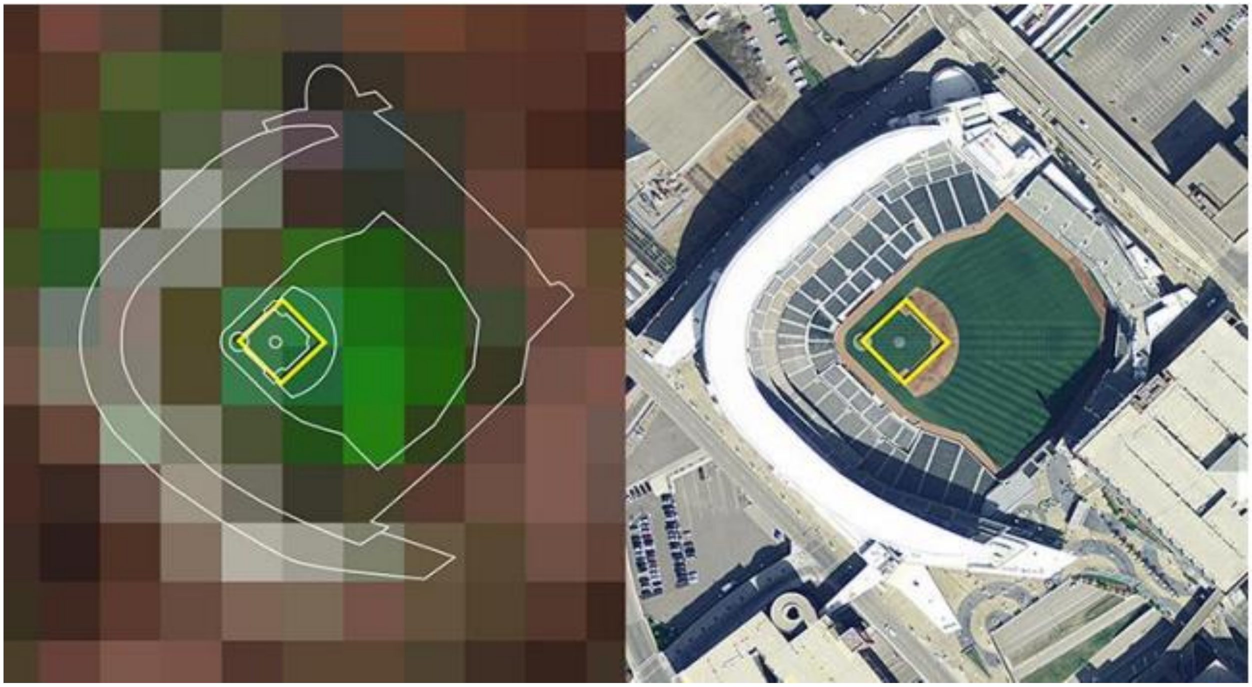


Figure 24 Picturing a pixel.

The yellow square is 30 meters on the side, exactly the size of a single pixel of Landsat data and slightly larger than a baseball diamond. The aerial photograph on the right shows Target Field in Minneapolis. The Landsat data on the left shows healthy vegetation in green while streets and buildings range from reddish-brown to white. Landsat's spatial resolution of 30 meters allows us to see the landscape at a human scale while simultaneously collecting data across a broad swath 185 kilometers (or 115 miles) wide.

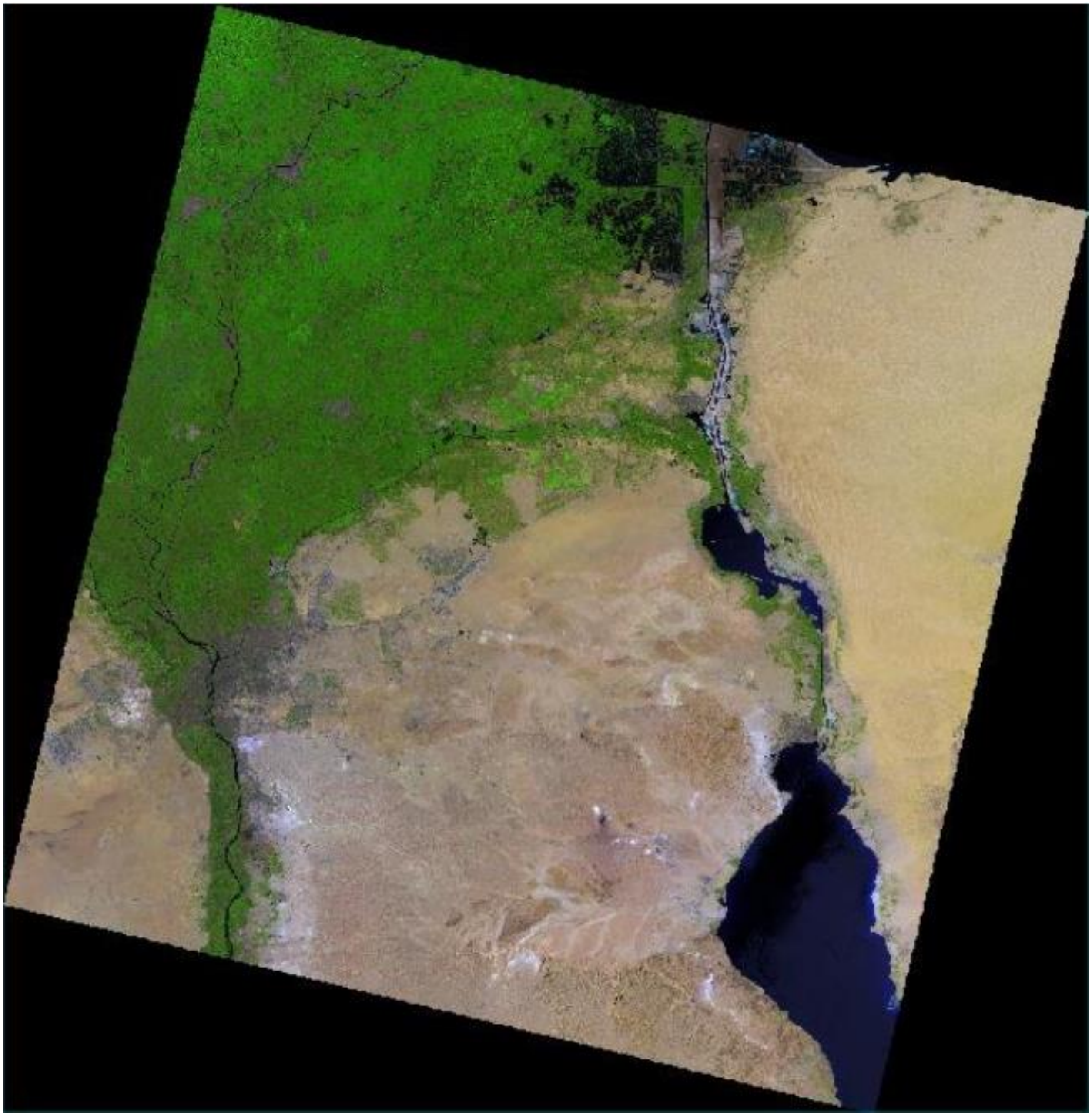


Figure 25 The study area using Landsat 8 satellite Image.

Landsat Data Acquisition.

In the first, we used the https://earthexplorer.usgs.gov/ website to download Landsat satellite images.

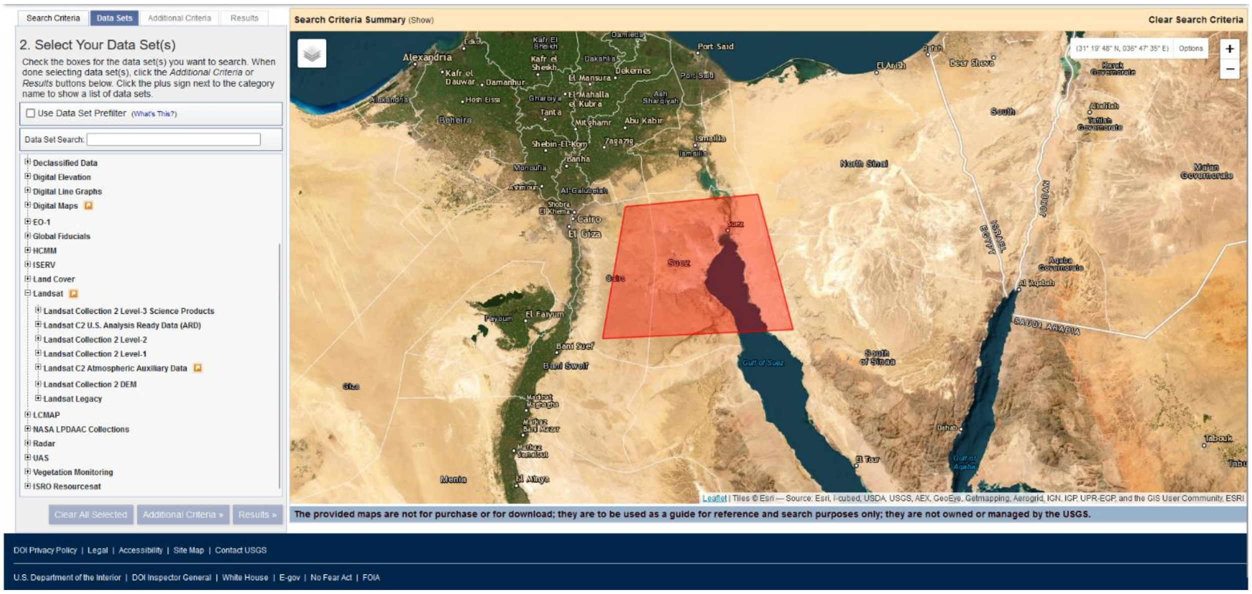


Figure 26 Landsat Dataset Download.

Secondly, we used the ArcMap software to compose the Landsat 1,2,3,4,5,6,7 Bands.

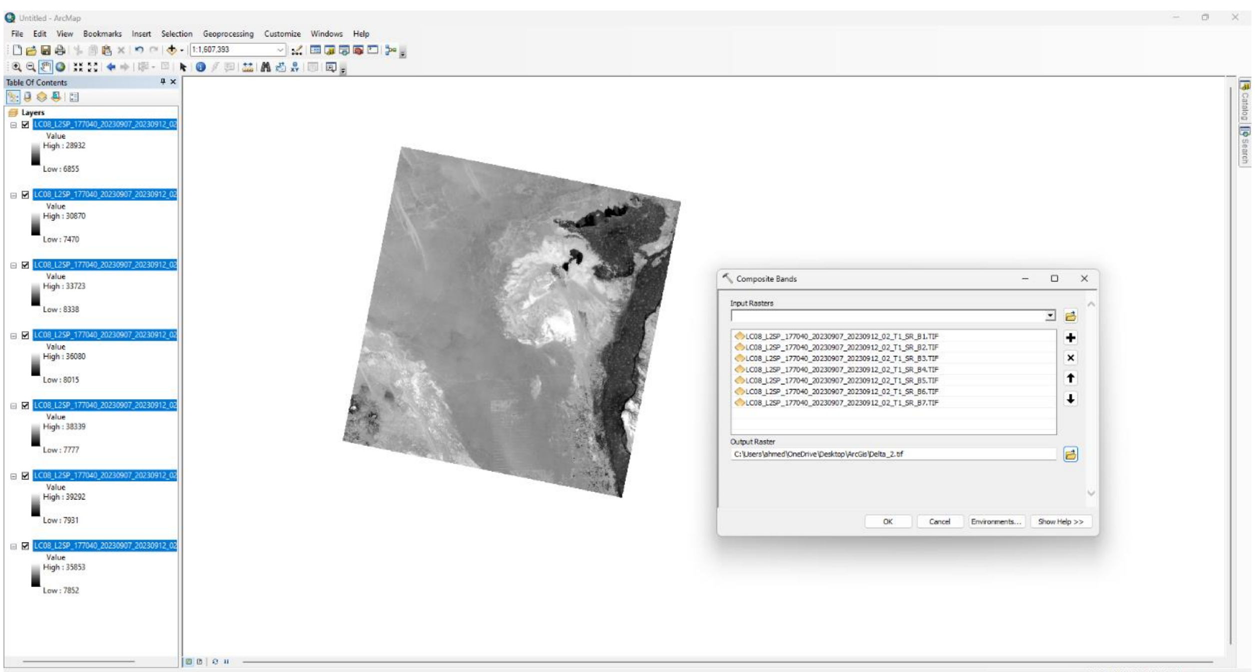


Figure 27 Landsat Composite Bands.

4-1-2 Sentinel 2.

Sentinel-2 Characteristics.

The Sentinel-2 mission is based on a constellation of two identical satellites in the same orbit. Each satellite carries an innovative wide swath high-resolution multispectral imager with 13 spectral bands for a new perspective of our land and vegetation.

1- The Sentinel-2 satellites each carry a single multi-spectral instrument (MSI) with 13 spectral channels in the visible/near infrared (VNIR) and short wave infrared spectral range (SWIR). Within the 13 bands.
2- Systematic global coverage of land surfaces from 56° S to 84° N.
3- Revisiting every 10 days under the same viewing angles.
4- Spatial resolution of 10 m, 20 m and 60 5- 290 km field of view.

Sentinal-2 Band	**Sentinel-2 A**		**Sentinel-2 B**		**Spectral Resolution (m)**
	Central wavelength (nm)	Bandwidth (nm)	Central wavelength (nm)	Bandwidth (nm)	
Band 1-Coastal aerosol	442.7	21	21	60	60
Band 2-Blue	492.4	66	66	10	10
Band 3-Green	559.8	36	36	10	10
Band 4-Red	664.6	31	31	10	10
Band 5-Vegetation red edge	704.1	15	16	20	20
Band 6-Vegetation red edge	740.5	15	15	20	20
Band 7-Vegetation red edge	782.8	20	20	20	20
Band 8-NIR	832.8	106	106	10	10
Band 8A- Narrow NIR	864.7	21	22	20	20
Band 9-Water vapor	945.1	20	21	60	60
Band 10-SWIR-Cirrus	1373.5	31	30	60	60
Band 11-SWIR	1613.7	91	94	20	20
Band 12-SWIR	2202.4	175	185	20	20

Figure 28 Characteristics of Sentinel 2 spectral data.

Sentinel -2 Data Preparation.

In the first, we used the https://dataspace.copernicus.eu/ website to download Sentinel satellite images.

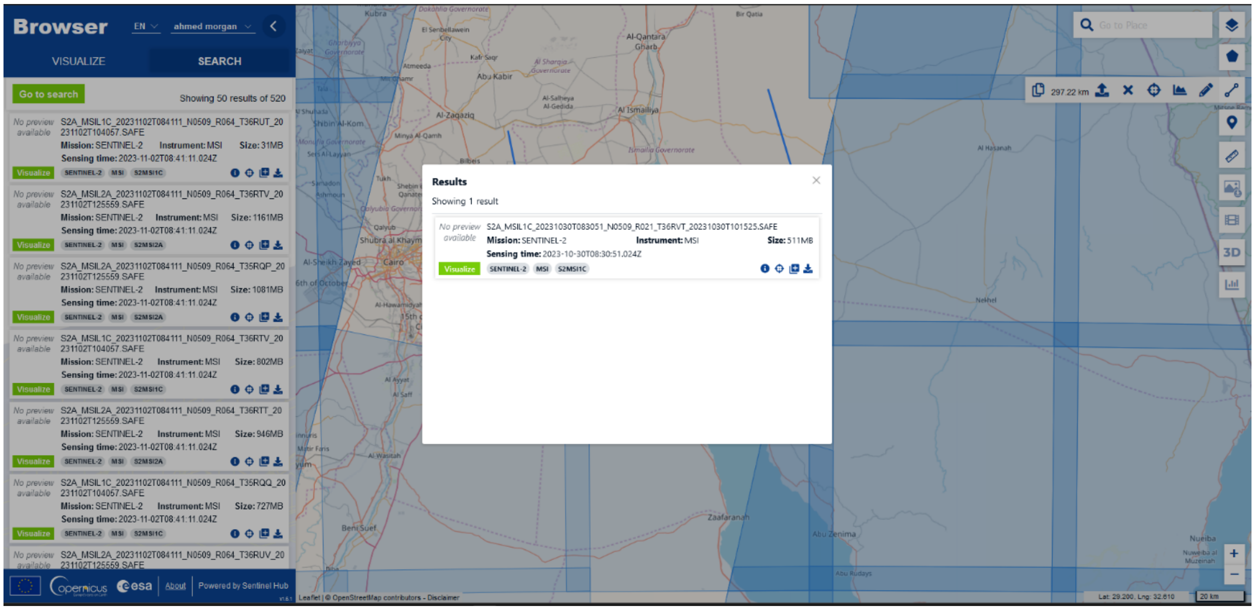


Figure 29 Sentinel -2 Dataset Download.

Secondly, we used the ArcMap program to compose the Sentinel -2 Bands.

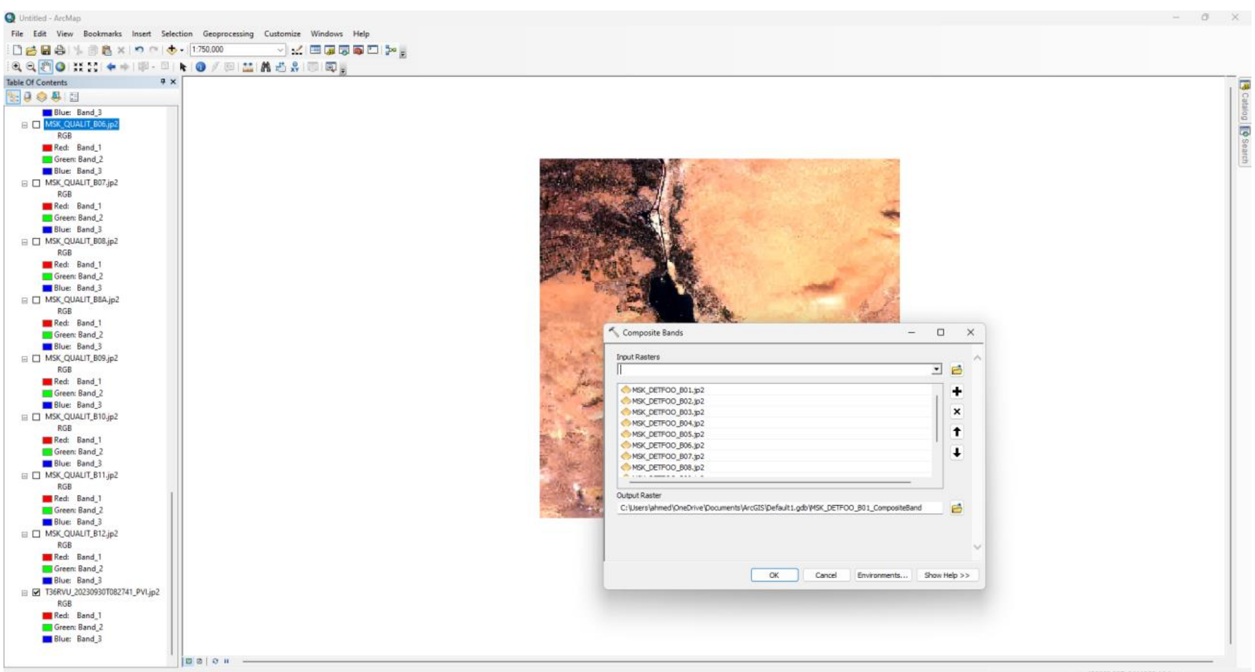


Figure 30 Sentinel-2 Composite Bands.

4-1-3 Alos-2

Alos-2 Radar Data Characteristics.

As of my last knowledge update in September 2021, I can provide you with some information about the ALOS AVNIR-2 (Advanced Visible and Near Infrared Radiometer type 2) satellite sensor. Please note that there may have been developments or changes since that time, so it's a good idea to verify the latest information if needed. Here are some characteristics of ALOS AVNIR-2:

1. **Sensor Type:** ALOS AVNIR-2 is a remote sensing instrument that operates in the visible and near-infrared (VNIR) spectrum.
2. **Launch Date:** ALOS (Advanced Land Observing Satellite), also known as "Daichi," was launched on January 24, 2006. AVNIR-2 was one of its instruments.
3. **Purpose:** AVNIR-2 is designed for Earth observation and is used for various applications, including land use and land cover classification, disaster monitoring, forestry, agriculture, and environmental monitoring.
4. **Spectral Bands:** AVNIR-2 has four spectral bands, covering the visible and near-infrared regions of the electromagnetic spectrum. These bands typically include red, green, blue, and near-infrared channels, which are essential for distinguishing different surface features.
5. **Spatial Resolution:** AVNIR-2 provides high spatial resolution imagery, typically around 10 meters. This fine spatial resolution allows for detailed monitoring and analysis of the Earth's surface.
6. **Coverage:** The instrument can capture images in a wide swath, which means it can cover a significant area on each pass. This is important for monitoring large regions efficiently.
7. **Revisit Time:** The ALOS satellite system was designed to provide relatively short revisit times for Earth observation, making it useful for time-sensitive applications.
8. **Data Products:** ALOS AVNIR-2 generates various data products, including orthorectified images, panchromatic and multispectral data, and other derived products suitable for different remote sensing applications.
9. **Applications:** The data from ALOS AVNIR-2 has been used in a wide range of applications, including agriculture, forestry, urban planning, environmental monitoring, disaster management, and more.
10. **Operational Status:** Please note that the information I have is accurate as of 2021. The operational status and specific details about the satellite and its sensors may have changed since then. Therefore, it's advisable to check with the Japan Aerospace Exploration Agency (JAXA) or relevant space agencies for the latest information regarding ALOS AVNIR-2.

Characteristics	Description
Sensor Type.	Remote sensing instrument operating in VNIR spectrum.
Launch Date.	January 24, 2006 (as part of ALOS satellite).
Purpose.	Earth observation for land use, disaster monitoring, forestry, agriculture, environmental moni toring.
Spectral Bands.	Four bands in visible and near-infrared regions.
Spatial Resolution.	Approximately 10 meters.
Coverage.	Wide swath coverage for efficient monitoring.
Revisit Time.	Designed for relatively short revisit times.
Products Data.	Orthorectified images, panchromatic and multispectral data, derived products.
Applications.	Agriculture, forestry, urban planning, environmental monitoring, disaster management, etc.
Status Operational.	As of 2021; check with JAXA or relevant space agencies for latest information.

AVNIR-2 Characteristics.	
Title	**Description**
Number of bands	4
Wavelength (micrometres)	Band 1: 0.42 -0.50 Band 2: 0.52 - 0.60 Band 3: 0.61 - 0.69 Band 4: 0.76 -0.89
Spatial resolution	10m (at Nadir)
Swath width	70km (at Nadir)
Signal to noise ratio	> 200
MTF	Band 1 through 3: > 0.25 Band 4: > 0.2
Number of detectors	7.100/Band
Pointing angle	-44 to +44 degrees
Bit length	8 bits/pixel
Data downlink rate	120Mbps
Bit length	8 bits/pixel
Data compression	Lossless compression (onboard)

Figure 31 Alos Avnir-2 Characteristics

Alos-2 Data Preparation.

In the first, we used https://search.asf.alaska.edu/ website to download Alos-2 satellite images.

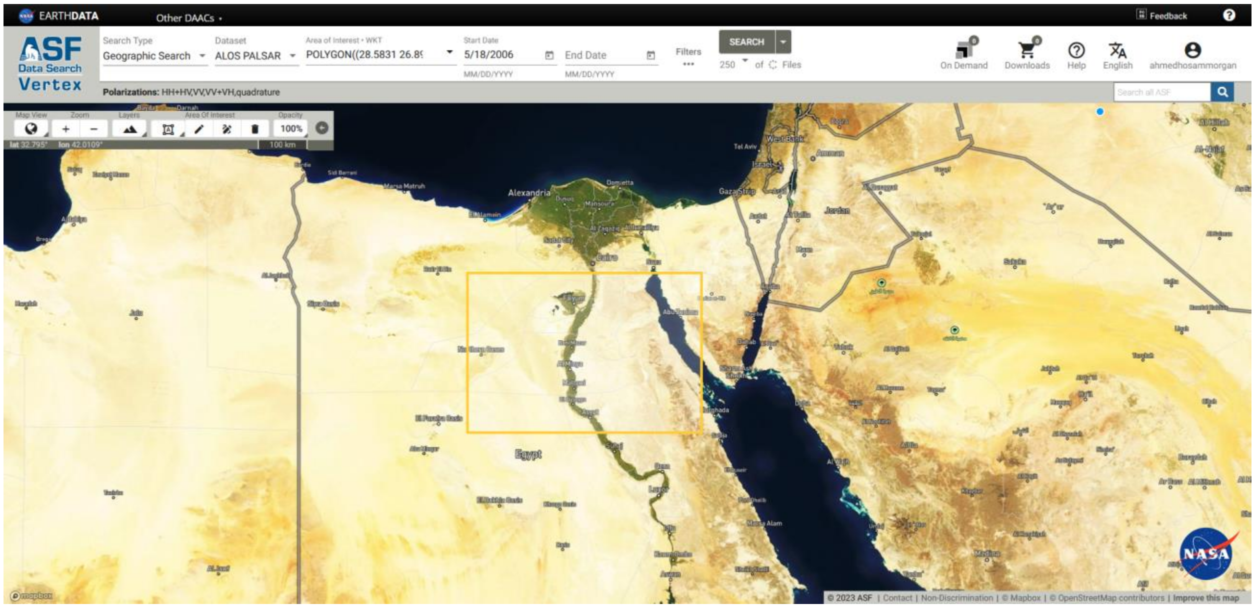


Figure 32 Alos-2 Dataset Download.

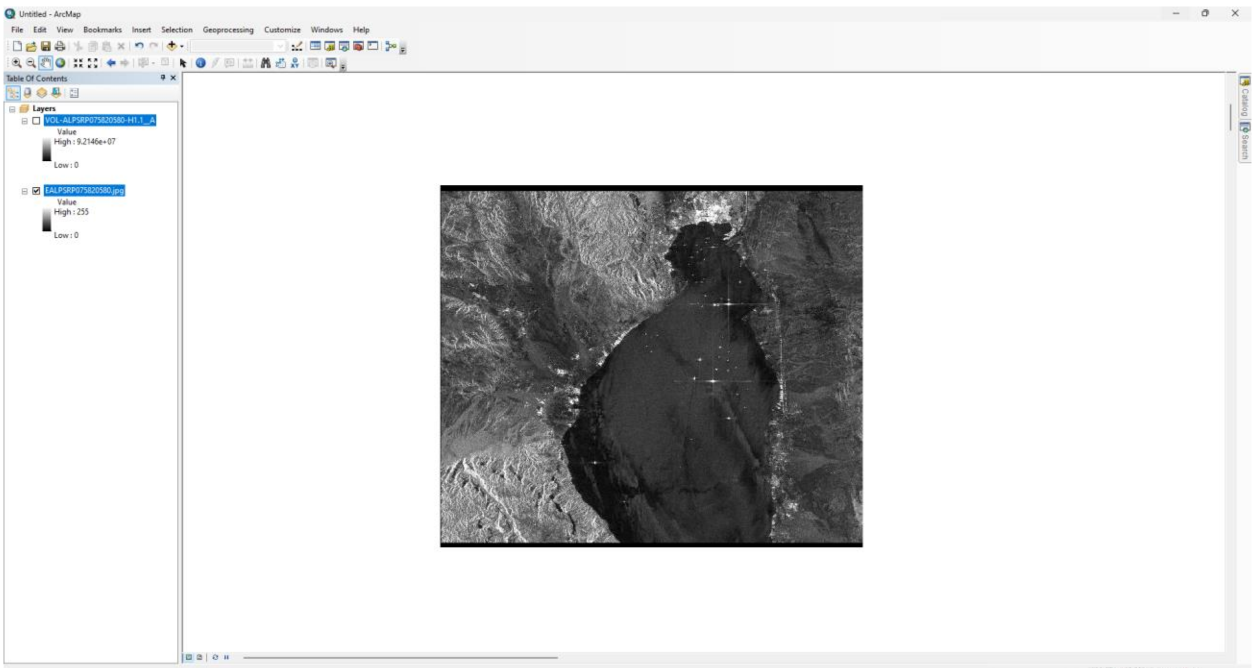


Figure 33 Alos Dataset View

Design / Simulation Setup.

Chapter Five.

5 Solution Methodology.

The automated enterprise system is based on Django Framework and design pattern in its development process. The model contains pure application data and layers that handle all system requests. Model should not contain logic that describes how to present the data to the end user. View is responsible for presenting and visualizing the model's data to the end user. View knows the way to access the model's data, but it does not know how and what does this data means or how the user can manipulate it. Controller usually lies between the view and the model; its major role is to listen to events that are triggered by the view and then executes the appropriate reaction to these fired events. The methodology for the development process of our automated. It depends on the standard Software Development Life Cycle (SDLC) process which includes (Requirements Analysis, System Design, Implementation, Testing, and Deployment).

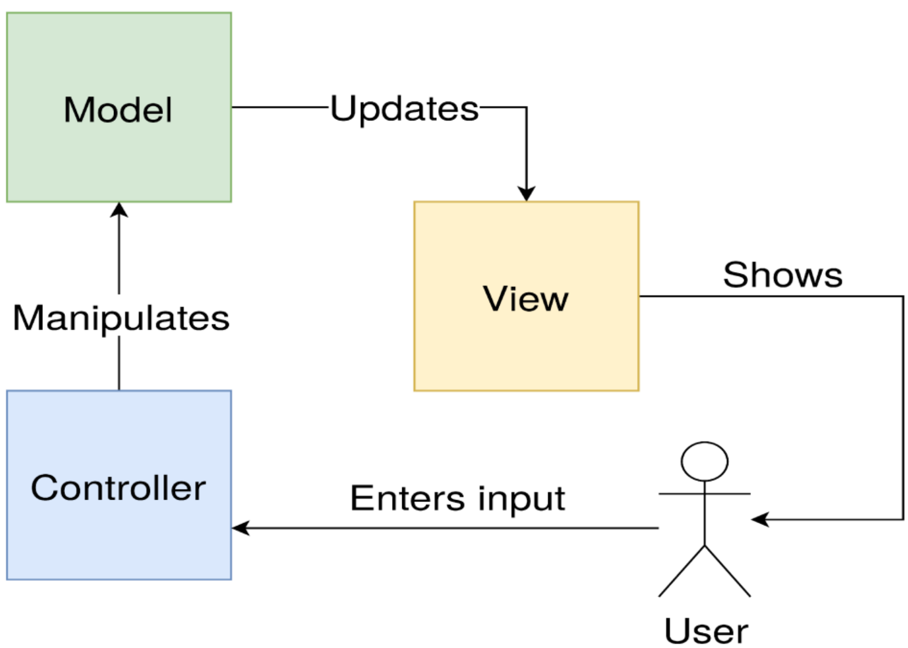


Figure 34 Structure of Django.

And in the development of an enterprise system for satellite image processing using AI within the Django framework, the Software Development Life Cycle (SDLC) follows a structured approach. Here's an overview of the SDLC stages used in this project:

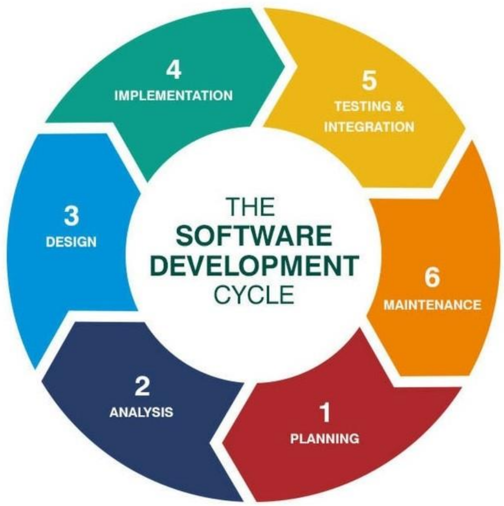


Figure 35 Software Development Life Cycle (SDLC).

5-1 Design / Simulation Setup.

The main purpose of the system design is to translate the gathered business requirements and business processes into a technical design that will be used to develop the proposed automated enterprise system functions.

5-1-1 System Architecture.

Describe the technical direction, overall technical solution in terms of platform(s), operating system(s), network, language(s), hardware, supporting software and tools to be used in the creation and maintenance of the application.

Hardware Requirements are as follow:

- Processors: Intel Core i7 9 GEN CPU.
- Memory: 16GB RAM.
- Graphics: GeForce® GTX 1650 with 4GB GDDR5.
- Disk Drive: SSD 500 GB.

Software Requirements are as follow:

- Programming Language: HTML, CSS, JavaScript, Python 3.6 or above.
- IDE: PyCharm, Visual Studio Code.
- Operating System: Windows 10,11 and Linux Based Systems.

The Libraries Requirements are as follow:

- Pandas.
- NumPy.
- Matplotlib.
- OpenCV.
- Scikit-learn.
- TensorFlow.
- PyTorch.
- Plotly.
- Rasterio.
- Scipy.
- Imageio.
- Pillow.
- Django.
- Six.

5-2 System Architecture.

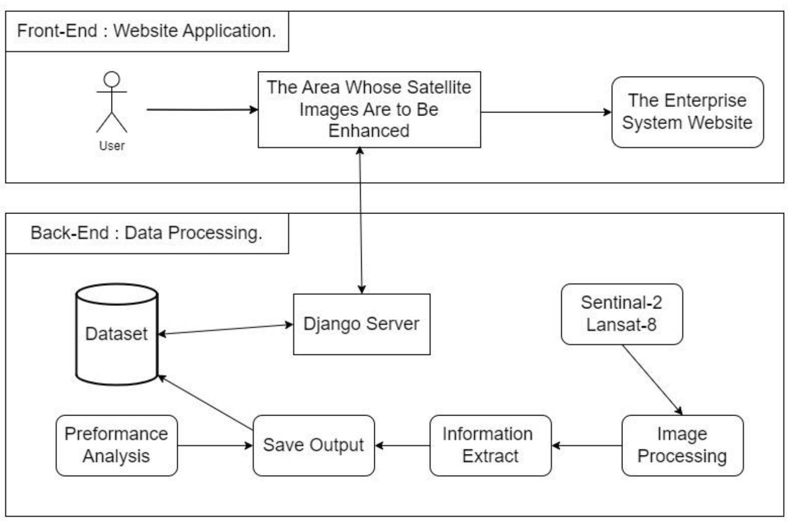


Figure 36 System Architecture.

5-3 System Interface Hierarchy.

Deliver and define a diagram of navigation hierarchy that explains how a user transfers or moves over the user interface.

5-3-1 System Process.

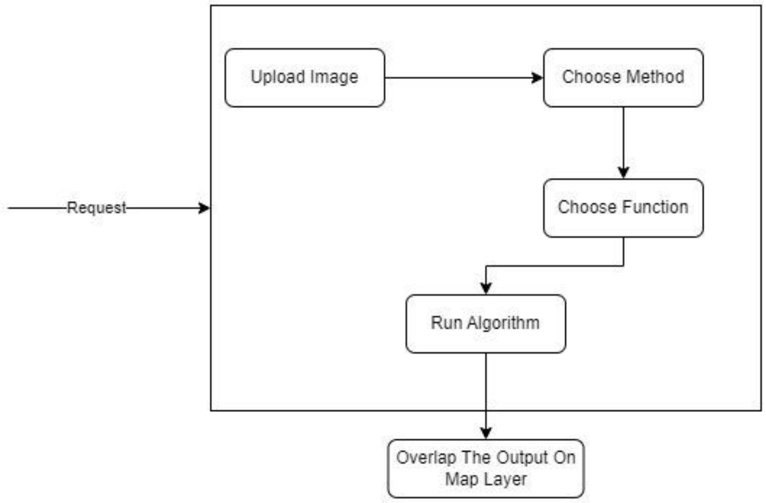


Figure 37 System Process.

5-3-2 System View.

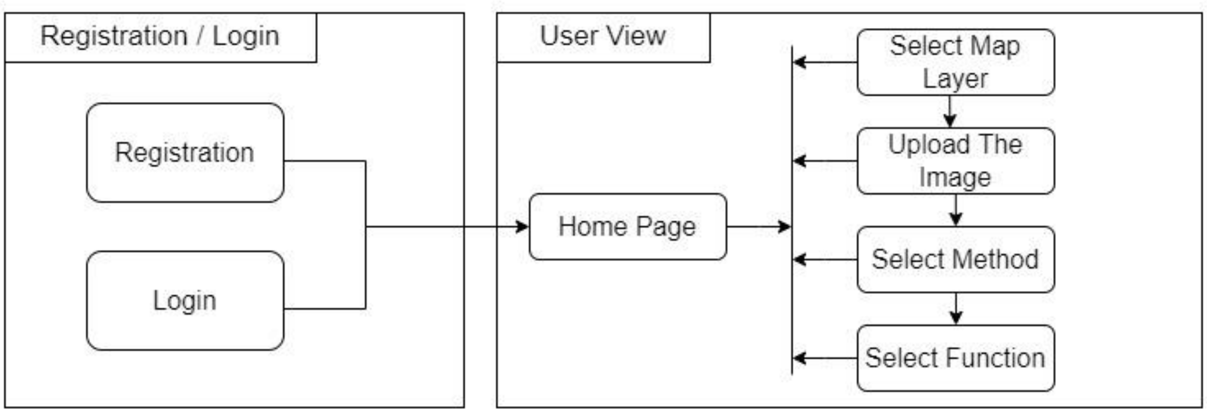


Figure 38 System View.

5-3-3 Sign in.

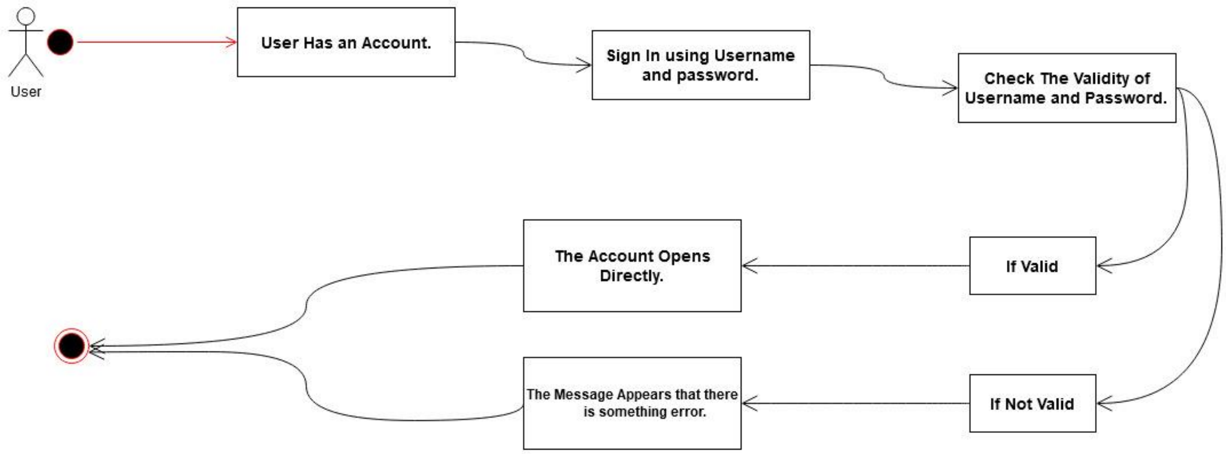


Figure 39 Navigation Hierarchy to sign in.

5-3-4 Run The Method or Function.

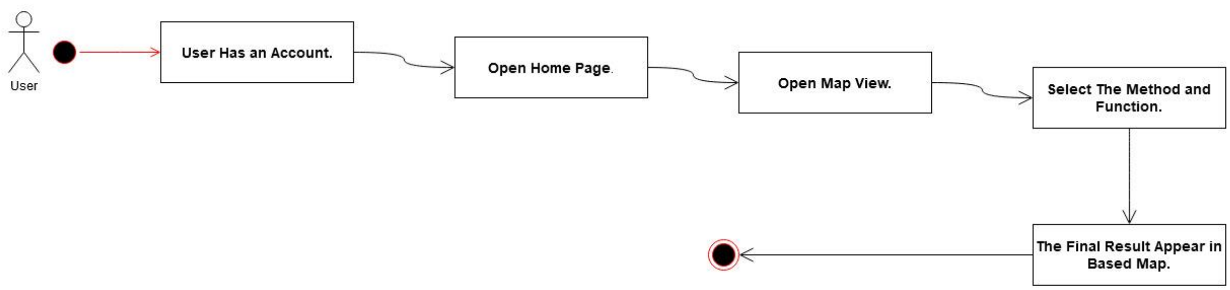


Figure 40 Navigation Hierarchy to Run a Method or Function.

5-3-5 Get in touch.

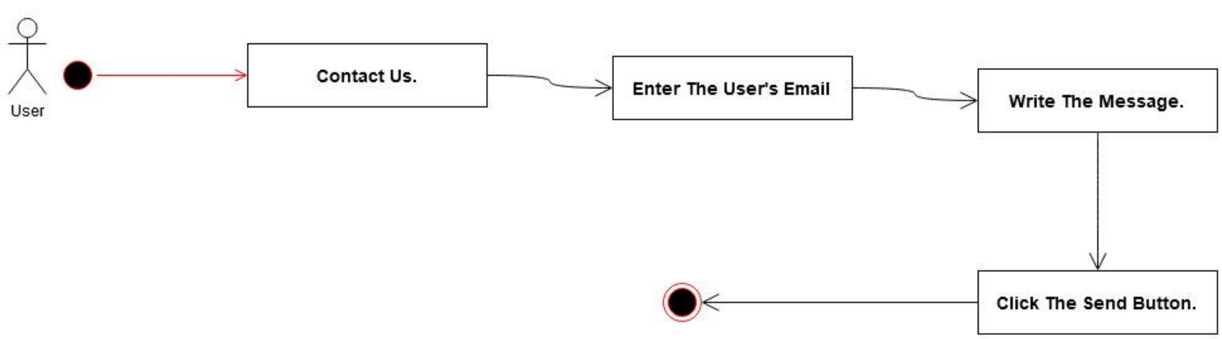


Figure 41 Navigation Hierarchy to get in touch.

5-4 Mockup Illustrates Expected Viewers.

5-4-1 Image Function and Methods Main Screen.

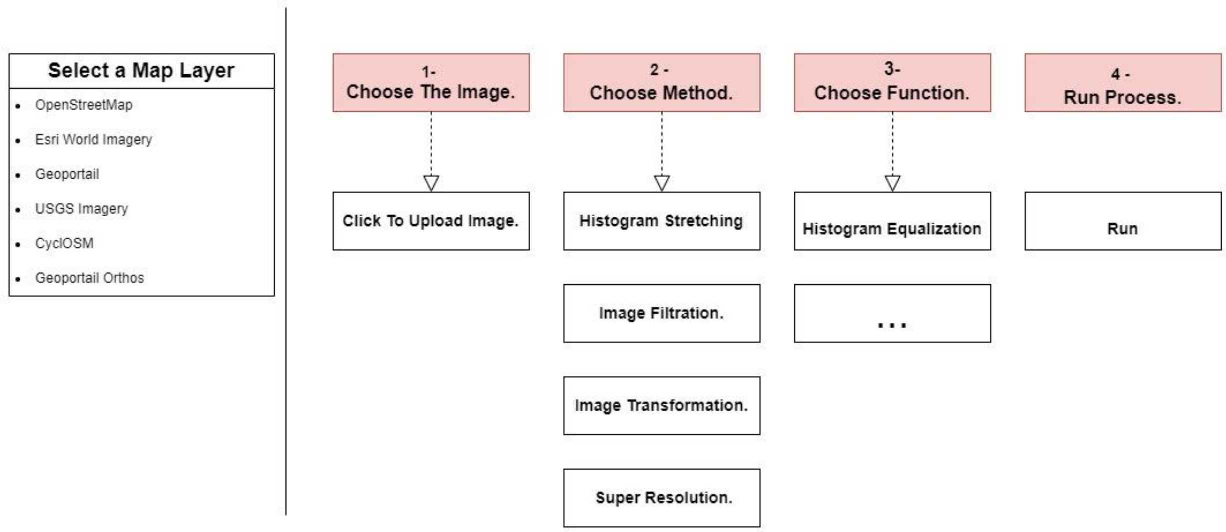


Figure 42 Mock-up illustrates expected viewer for Image Function and Methods Main Screen.

5-4-2 Histogram Stretching Functions Screen.

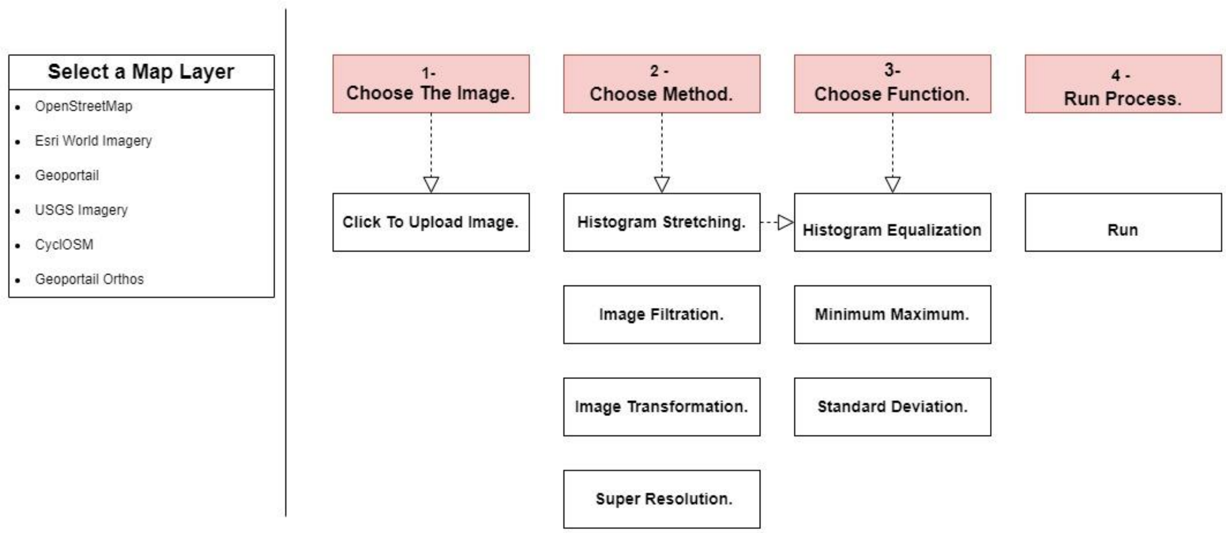


Figure 43 Mock-up illustrates expected viewer for Histogram Stretching.

5-4-3 Image Filtration Functions Screens.

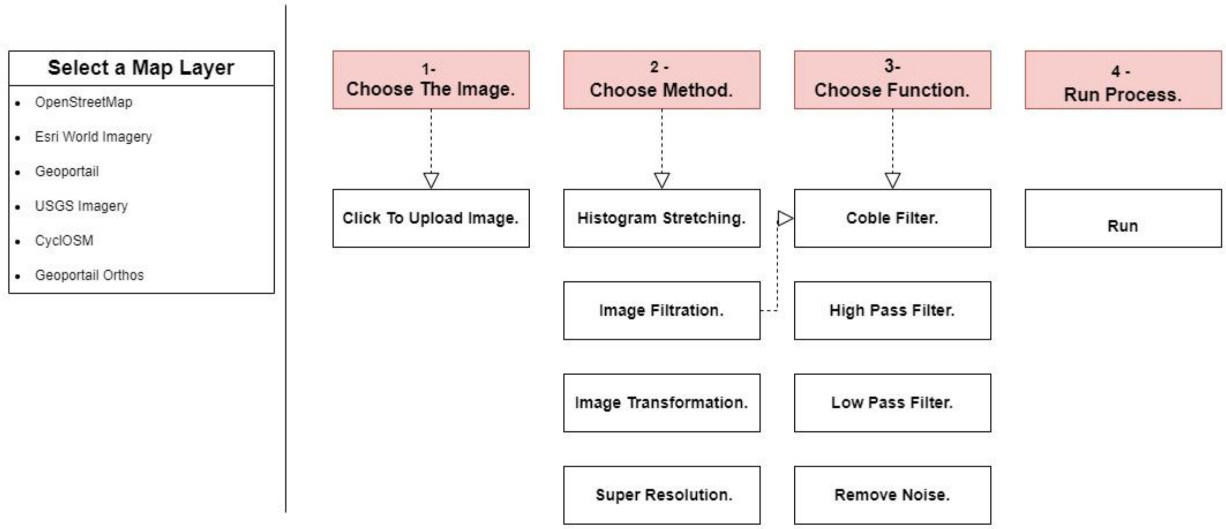


Figure 44 Mock-up illustrates expected viewer for Image Filtration.

5-4-4 Image Transformation Functions Screens.

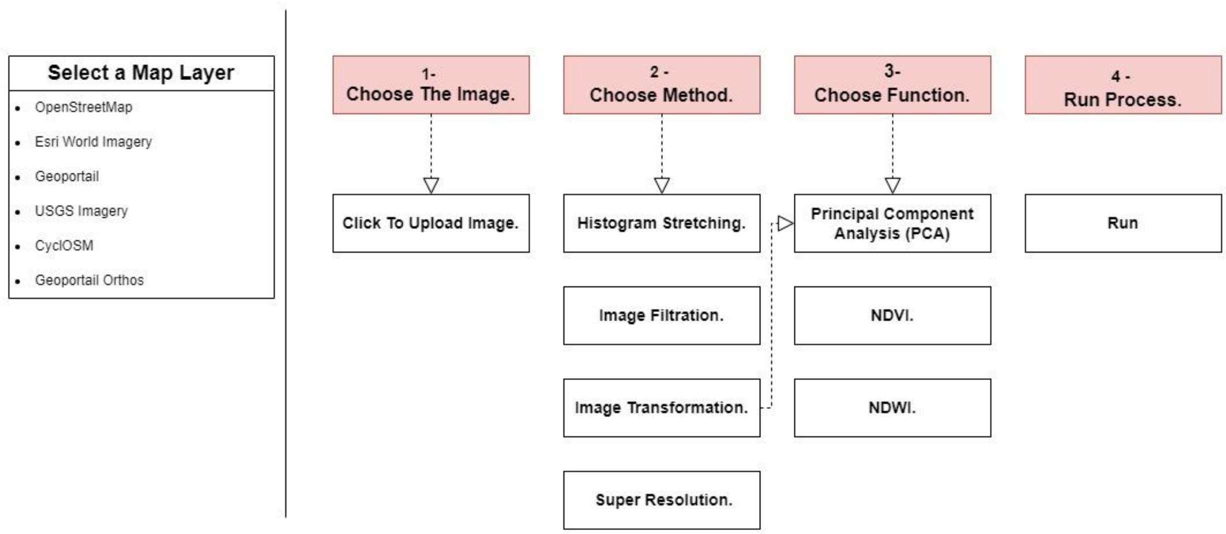


Figure 45 Mock-up illustrates expected viewer for Image Transformation.

5-4-5 Super Resolution Functions Screens.

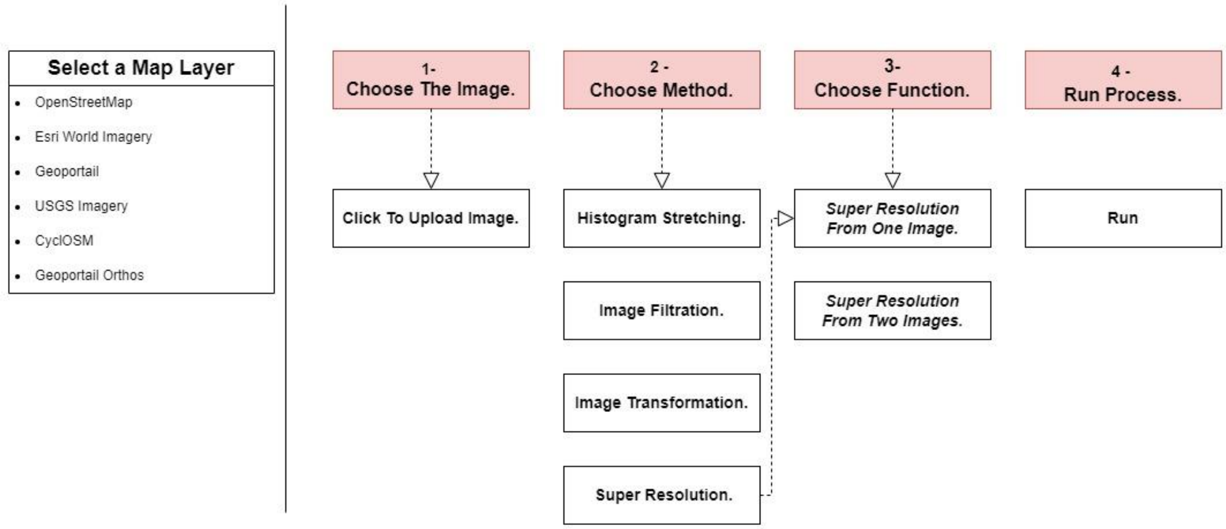


Figure 46 Mock-up illustrates expected viewer for Super Resolution.

System Analysis.

Chapter Six.

6 Proposed System.

6-1 Overview

Enhancing satellite images involves refining the quality and clarity of data captured by satellites orbiting the Earth. These images are vital for various applications like environmental monitoring, urban planning, agriculture, and disaster management. However, satellite images often face challenges such as low resolution, noise, atmospheric interference, and limited spectral information. Enhancing these images involves sophisticated algorithms that aim to improve resolution, remove noise, correct distortions, and enhance details, ensuring better interpretation and analysis of the data they provide. This process is crucial in extracting valuable insights and aiding decision-making across multiple fields.

6-2 Functional Requirements.

A functional requirement is the function of a software. The function is described as a set of inputs, behavior, and outputs. This system defines the capabilities and functions that a system must be able to perform successfully.

6-2-1 Website's Functional Requirements.

- Welcome Page.
- Sign in.
- Sign up.
- Home Page.
- About US Page.
- Choose The Map Layer List Part.
- Description of Selected Map Layer.
- Map View Part.
- Upload Image Part.
- Choose Function.
- Choose Method.
- Processing Part.
- The Final Output Appear in The Map Layer.
- In Case of Choose Super Resolution, User Can Select the Pretrained Dataset.
- Change Account Button.
- Login Button.
- Logout Button.
- Go Back to the Welcome Page Button.

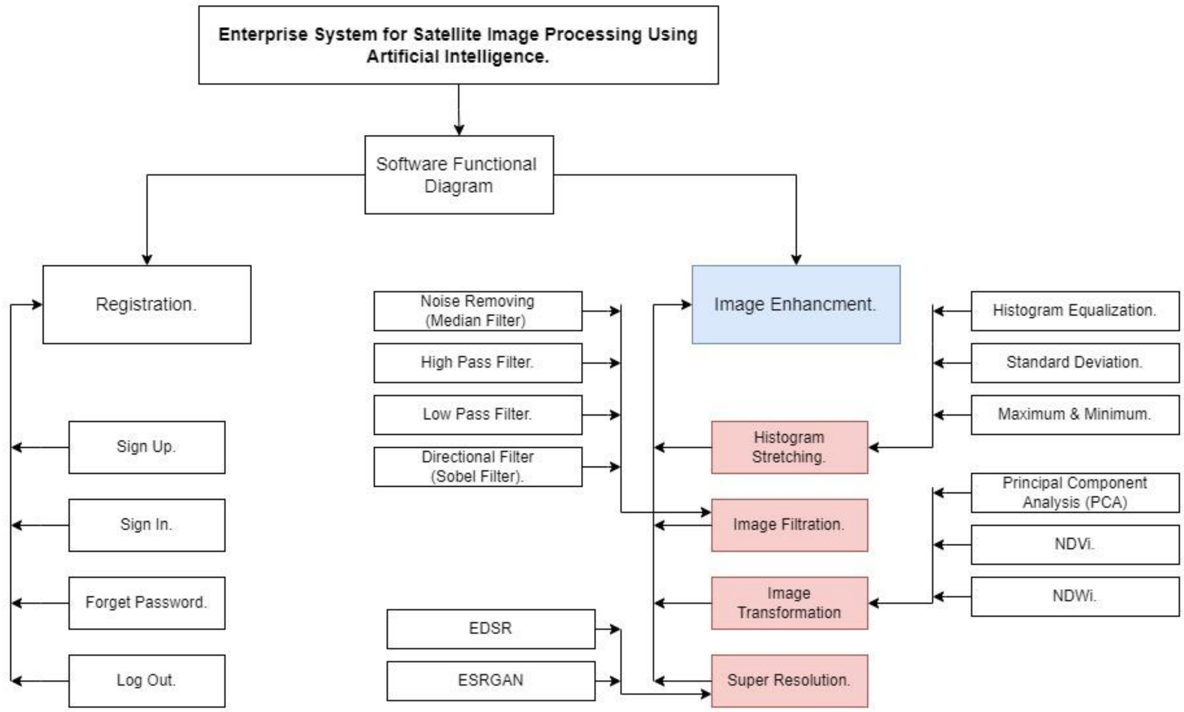


Figure 47 Project Functional Diagram.

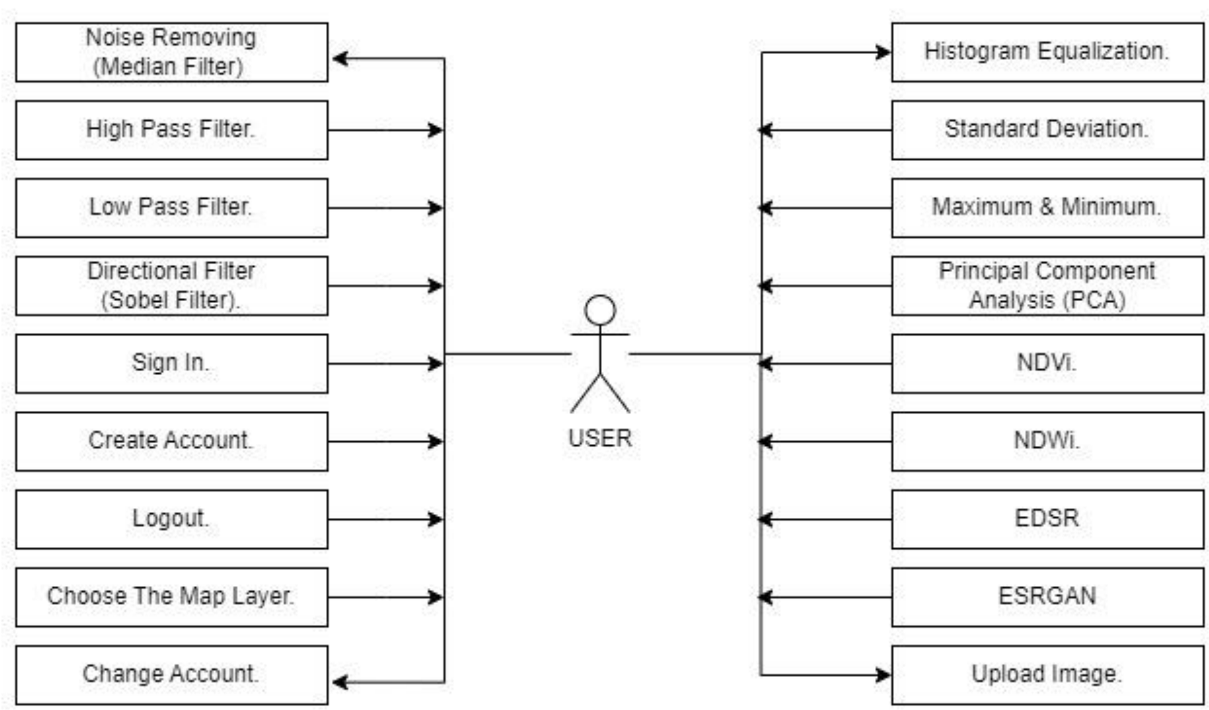


Figure 48 Use Case Diagram.

6-3 Non-Functional Requirements.

- **Usability.**

  Simple to use: the user will be able to surf the website easily with no difficulties facing him/her.

- **Reliability.**

  Crash safe. The application should be crash safe in 90% of its runtime.

- **Performance**

  Short response time: the loading time of the application must be less than 10 seconds. All other response times must be below 5 seconds.

6-4 System Models.

6-4-1 Actors.

The actors of the system are:

1. Web Application Developer.

Within a satellite image processing system, the Web Application Developer holds a pivotal role in crafting an intuitive and efficient platform accessible via web browsers. This platform enables users to seamlessly interact with and process satellite imagery. Here are the key responsibilities and tasks associated with this role:

- **Interface Design and Development:** Creating a user-friendly interface that facilitates easy upload, visualization, and manipulation of satellite images. This involves implementing interactive features, ensuring cross-browser and cross-device compatibility, and optimizing user experience.
- **Integration of Image Processing Algorithms:** Collaborating with image processing experts to seamlessly integrate intricate algorithms. These algorithms aim to enhance, filter, and analyze satellite images. The developer ensures these algorithms are effectively incorporated into the web application framework for efficiency.
- **User Authentication and Permissions:** Implementing secure user authentication methods and managing user roles and permissions.
- **Testing and Debugging:** Conducting comprehensive testing to ensure the application functions flawlessly across diverse scenarios. Debugging issues and promptly addressing errors or performance bottlenecks are part of this role.
- **Documentation and Support:** Creating comprehensive documentation for users and fellow developers. Providing support and troubleshooting assistance to users encountering difficulties while using the application.

2. Web Application User.

The user is the person who uses all the functions and the features of the system like Sign up, sign in, Reset Password, Edit Profile, Post A Contact Us Message, Make Image Filtering, Make a Histogram Stretching, Make an Image Transformation, Make a Super Resolution of Image, Upload Image, Change Account, Logout.

6-5 Scenarios.

6-5-1 Sign in.

If the User has already an account, he will sign by using his username and password to open his account, the username and the password will be sent to the server to check if they are valid or not, if they are valid the account will be opened directly, if the username or the password are invalid, a message will appear to the user that there is something incorrect.

6-5-2 Signup.

If the user needs to register to the system, a registration form appears to him, so he can fill the form by entering his full name, number, email, password, etc. and then click the button (Signup), if the user does not fill any of this information, a message will appear to the user that he has to fill all the wanted information to complete his registration.

6-5-3 Reset Password.

Reset Password allows users to regain access by submitting their email or username, triggering a validation process via email or token for setting a new password. Error messages guide users in case of incorrect or missing details, maintaining system security and user management efficiency.

6-5-4 Get in touch.

In our proposed system If the user needs to post a contact message, he will choose "Contact us "and then enters his email and write the message he wants to send and click send, then the system will send the developer an email with the message the user sent.

6-5-5 Choose The Map Layer.

The system allows users to select map layers for visualization. Users can choose specific map layers, such as satellite imagery, street maps, or terrain views, from a selection menu. This selection enables users to customize their viewing preferences and access the desired geographic information for analysis or visualization purposes.

6-5-6 Upload Image.

The system allows users to upload their satellite images for processing and analysis. Users can select and upload their image files through an intuitive interface. Once uploaded, these images become accessible within the system for applying various image enhancement, filtration, transformation, super-resolution, or other processing techniques available. This functionality enables users to work with their own datasets, empowering them to derive insights and perform analyses tailored to their specific needs and objectives.

6-5-7 Histogram Stretching.

The system offers histogram stretching methods like histogram equalization, standard deviation, minimum, and maximum. Histogram equalization redistributes pixel intensities to enhance image contrast. Standard deviation stretching adjusts pixel values based on their distance from the mean, widening or narrowing the histogram. Minimum and maximum stretching reassign pixel values to span the entire intensity range, enhancing image brightness and contrast. Users can choose these methods to optimize image appearance and improve visual analysis of satellite imagery.

6-5-8 Image Filtration.

The system facilitates image filtration through diverse methods such as noise removal via the Median Filter, accentuation of high-frequency components with the High Pass Filter, smoothing and detail retention via the Low Pass Filter, and edge detection through the Directional Filter like the Sobel Filter. These methods offer users varied ways to process satellite imagery, allowing noise reduction, feature enhancement, and edge detection, catering to different analytical needs, and improving image quality for effective interpretation.

6-5-9 Image Transformation.

The system supports image transformation techniques such as Principal Component Analysis (PCA), Normalized Difference Vegetation Index (NDVI), and Normalized Difference Water Index (NDWI). PCA reduces data dimensions by extracting essential features, aiding in image compression and analysis. NDVI highlights vegetation health by comparing visible and near-infrared light, while NDWI detects water content by contrasting near-infrared and green wavelengths. These transformation methods enable users to derive valuable insights related to vegetation health and water presence, facilitating informed decisions across agricultural, environmental, and land management applications.

6-5-10 Super Resolution.

The system incorporates super-resolution techniques such as EDSR (Enhanced Deep Super-Resolution) and ESRGAN (Enhanced Super-Resolution Generative Adversarial Networks). EDSR employs deep learning to reconstruct high-resolution images from low-resolution inputs, enhancing image quality and detail. ESRGAN, utilizing generative adversarial networks, refines image textures and sharpness, producing more realistic and visually appealing high-resolution outputs. These methods empower users to upscale satellite imagery, improving clarity and enabling finer analysis for diverse applications like remote sensing, urban planning, and infrastructure development.

6-5-11 Show Result.

The system enables users to view processed results after applying various image enhancement, filtration, transformation, and super-resolution techniques. Once users select and execute their preferred method, the system displays the enhanced, filtered, transformed, or super-resolved imagery for analysis and interpretation. This visual representation allows users to assess the effectiveness of the applied technique and derive insights for their specific applications in fields such as environmental monitoring, agriculture, urban planning, and more.

6-5-12 Logout.

The system provides users with a simple and secure way to log out of their accounts. Users can initiate the log-out process by accessing the designated log-out button or option within the interface. Upon clicking this button, the system ensures a secure session termination, effectively logging the user out of their account. This action helps maintain account security by ending the user's access to the system's functionalities and data, ensuring privacy, and preventing unauthorized access to sensitive information.

6-6 Specification Tables.

<table>
<tr><th colspan="3">Sign in.</th></tr>
<tr><td>Actor:</td><td colspan="2">User.</td></tr>
<tr><td>Precondition:</td><td colspan="2">Open the website on any browser and has account on it.</td></tr>
<tr><td rowspan="2">Flow of Events.</td><td>Actor Steps.</td><td>System Steps.</td></tr>
<tr><td>The user enters the username and password.
Click on “Sign in”</td><td>The system checks if the username and password are correct, if they are correct the account opens directly.
If one of the username/passwords are incorrect, the system sends a message to the doctor that there is something wrong, so tries again.</td></tr>
</table>

<table>
<tr><th colspan="3">Sign Up.</th></tr>
<tr><td>Actor:</td><td colspan="2">User.</td></tr>
<tr><td>Precondition:</td><td colspan="2">Open the website on any browser.</td></tr>
<tr><td rowspan="2">Flow of Events.</td><td>Actor Steps.</td><td>System Steps.</td></tr>
<tr><td>The user fills out the registration form.
Then click (Sign up).</td><td>The system checks if there is an account with the same number before in the database.
If not, the system checks if the user filled in all the wanted information.
If the two steps above are valid the system create username and adds the new account to the database, then opens the profile for him.</td></tr>
</table>

<table>
<tr><th colspan="3">Reset Password.</th></tr>
<tr><td>Actor:</td><td colspan="2">User.</td></tr>
<tr><td>Precondition:</td><td colspan="2">Open the website on any browser and has account on it.</td></tr>
<tr><td rowspan="2">Flow of Events.</td><td>Actor Steps.</td><td>System Steps.</td></tr>
<tr><td>• The user clicks on “Forget Password.”
• Then click (Sign up).</td><td>• The system checks if the username and email are correct, if they are correct the system sends mail with password.
• If the username/email is incorrect the system shows something wrong, so try again.</td></tr>
</table>

<table>
<tr><th colspan="3">Upload Image.</th></tr>
<tr><td>Actor:</td><td colspan="2">User.</td></tr>
<tr><td>Precondition:</td><td colspan="2">The user has an account on the website and is accessing it via any browser.</td></tr>
<tr><td rowspan="2">Flow of Events.</td><td>Actor Steps.</td><td>System Steps.</td></tr>
<tr><td>• The user clicks on the "Upload Image" button.
• The user selects the image file for upload.</td><td>• The system prompts the user to select the desired image file from their local device.
• The system validates the file format and size to ensure compatibility and feasibility for processing.</td></tr>
</table>

<table>
<tr><th colspan="3">Histogram Stretching.</th></tr>
<tr><td>Actor:</td><td colspan="2">User.</td></tr>
<tr><td>Precondition:</td><td colspan="2">The user has accessed the image processing system and has an uploaded image.</td></tr>
<tr><td rowspan="2">Flow of Events.</td><td>Actor Steps.</td><td>System Steps.</td></tr>
<tr><td>• The user chooses the "Histogram Stretching" option from the processing methods menu.

• The user selects the desired method among options like histogram equalization, standard deviation, minimum, or maximum stretching.</td><td>• The system processes the selected method for histogram stretching on the chosen image.

• The system validates the file format and size to ensure compatibility and feasibility for processing.

• The system processes the selected method for histogram stretching for Standard Deviation, Minimum and Maximum on the chosen image.

• The system displays the stretched image, showing the applied histogram stretching method's effect.</td></tr>
</table>

<table>
<tr><th colspan="3">Image Filtration.</th></tr>
<tr><td>Actor:</td><td colspan="2">User.</td></tr>
<tr><td>Precondition:</td><td colspan="2">The user has accessed the image processing system and has an uploaded image.</td></tr>
<tr><td rowspan="2">Flow of Events.</td><td>Actor Steps.</td><td>System Steps.</td></tr>
<tr><td>The user selects the uploaded image for image filtration.
The user chooses the "Image Filtration" option from the processing methods menu.
The user selects the desired filtration method among options like Median Filter (Noise Removing), High Pass Filter, Low Pass Filter, or Directional Filter (Sobel Filter).</td><td>The system processes the selected method for Image Filtration on the chosen image.
The system validates the file format and size to ensure compatibility and feasibility for processing.
The system processes the selected method for Image Filtration for Median Filter, High Pass Filter, Low Pass Filter, Sobel Filter on the chosen image.
The system displays the Filtered image.</td></tr>
</table>

Image Transformation.		
Actor:	User.	
Precondition:	The user has accessed the image processing system and has an uploaded image.	
	Actor Steps.	System Steps.
Flow of Events.	• The user selects the uploaded image for image Transformation. • The user chooses the "Image Transformation" option from the processing methods menu. • The user selects the desired transformation method among options like Principal Component Analysis (PCA), Normalized Difference Vegetation Index (NDVI), or Normalized Difference Water Index (NDWI).	• The system processes the selected method for Image Transformation on the chosen image. • The system validates the file format and size to ensure compatibility and feasibility for processing. • The system processes the selected method for Image Transformation for Principal Component Analysis (PCA), Normalized Difference Vegetation Index (NDVI), or Normalized Difference Water Index (NDWI). • The system displays the Transformed image.

<table>
<tr><th colspan="3">Super Resolution.</th></tr>
<tr><td>Actor:</td><td colspan="2">User.</td></tr>
<tr><td>Precondition:</td><td colspan="2">The user has accessed the image processing system and has an uploaded image.</td></tr>
<tr><td rowspan="2">Flow of Events.</td><td>Actor Steps.</td><td>System Steps.</td></tr>
<tr><td>• The user selects the uploaded image for Super Resolution.

• The user chooses the "Super Resolution" option from the processing methods menu.

• The user selects the desired super-resolution method among options like Enhanced Deep Super-Resolution (EDSR) or Enhanced Super-Resolution Generative Adversarial Networks (ESRGAN).</td><td>• The system processes the selected method for Image Super Resolution on the chosen image.

• The system validates the file format and size to ensure compatibility and feasibility for processing.

• The system processes the selected method for Image Transformation for the desired super-resolution method among options like Enhanced Deep Super-Resolution (EDSR) or Enhanced Super-Resolution Generative Adversarial Networks (ESRGAN).

• The system displays the Super Resolution image.</td></tr>
</table>

<table>
<tr><th colspan="3">Post A Contact Us Message</th></tr>
<tr><td>Actor:</td><td colspan="2">User.</td></tr>
<tr><td>Precondition:</td><td colspan="2">1. Open the website on any browser and have an account on it.
2. Goes to contact us page</td></tr>
<tr><td rowspan="2">Flow of Events.</td><td>Actor Steps.</td><td>System Steps.</td></tr>
<tr><td>• The user enters his username.

• Then type his message.

• Enters his email.</td><td>• The system checks for his username.

• And then sends the system an email with the user’s message.</td></tr>
</table>

Logout.		
Actor:	User.	
Precondition:	Open the website on any browser and has account on it.	
Flow of Events.	Actor Steps.	System Steps.
	• The user clicks on logout button.	• The system logs his out directly.

6-7 Activity Diagram.

Sign in Activity Diagram.

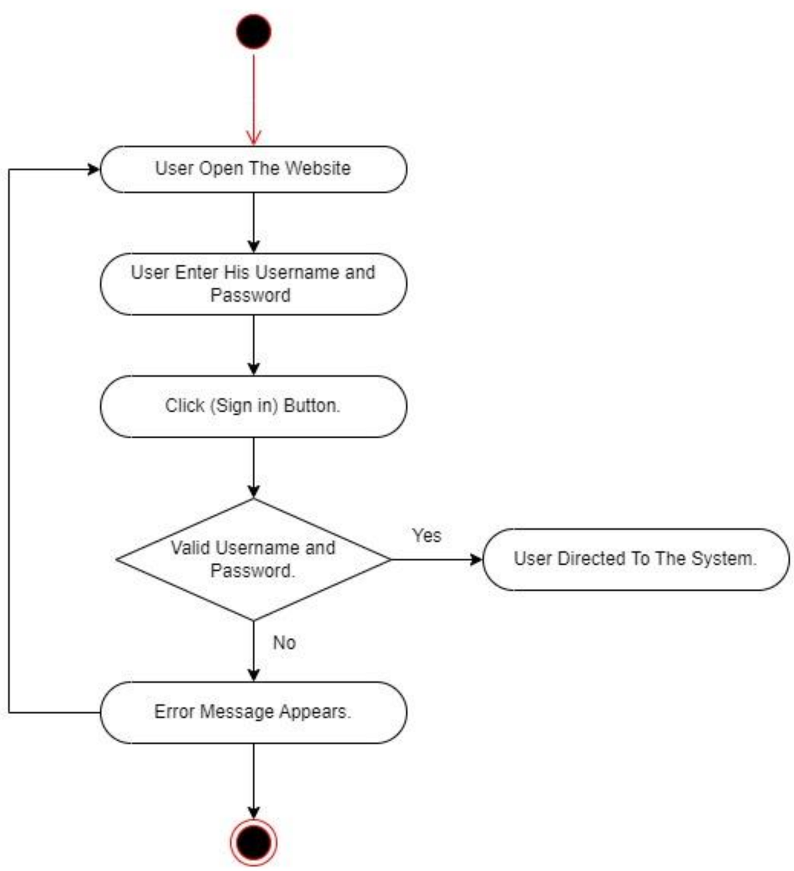


Figure 49 Sign in Activity Diagram.

Sign Up Activity Diagram.

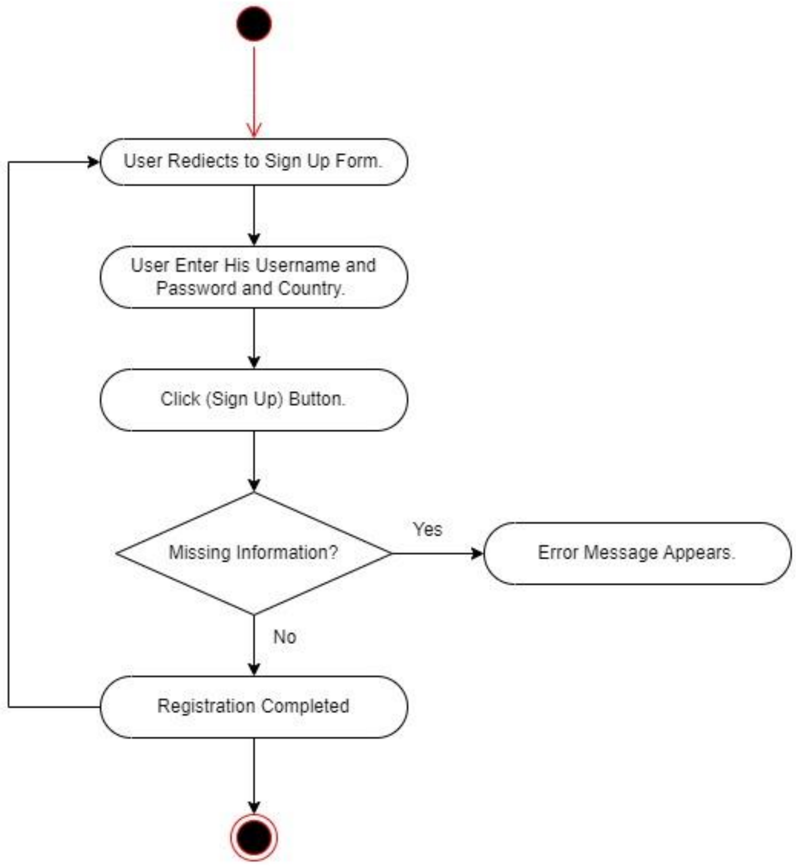


Figure 50 Sign Up Activity Diagram.

Forget Password Activity Diagram.

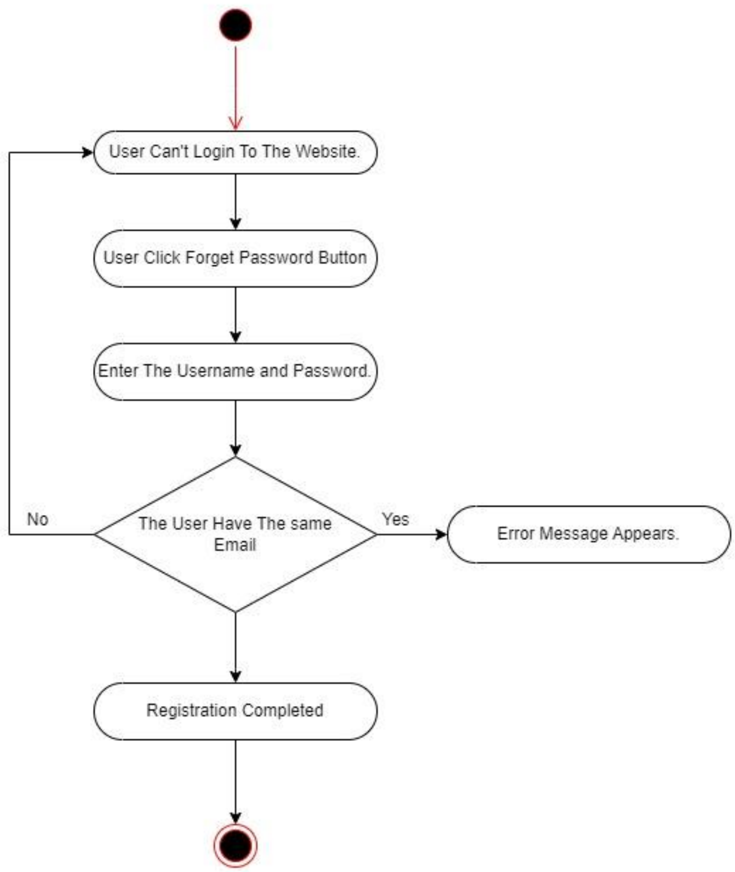


Figure 51 Forget Password Activity Diagram.

Functions and Methods Activity Diagram.

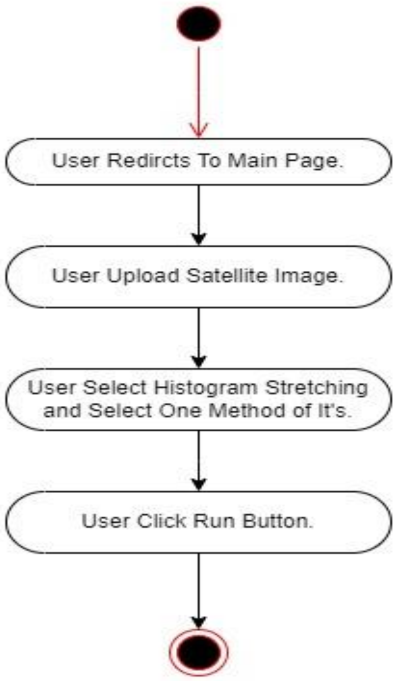


Figure 53 Histogram Stretching Activity Diagram.

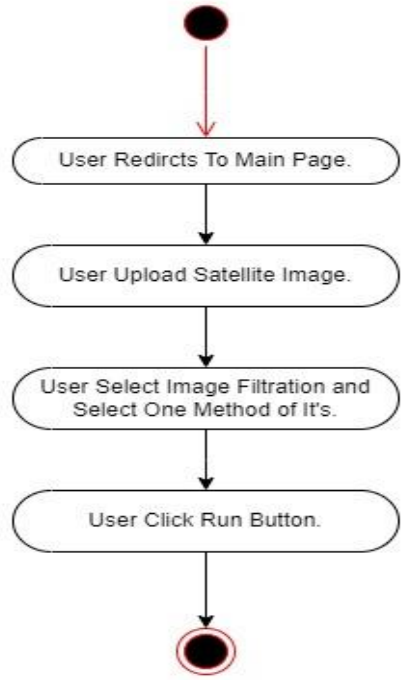


Figure 52 Image Filtration Activity Diagram.

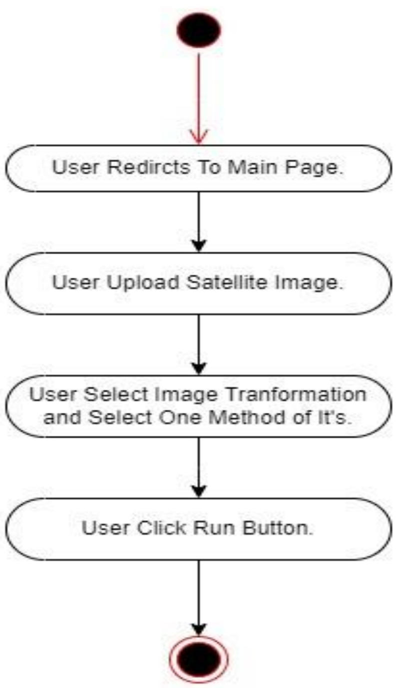


Figure 55 Image Transformation Activity Diagram.

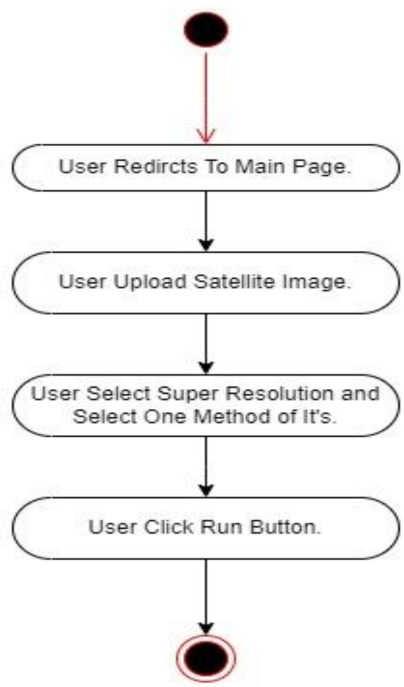


Figure 54 Super Resolution Activity Diagram

Contact Us Activity Diagram.

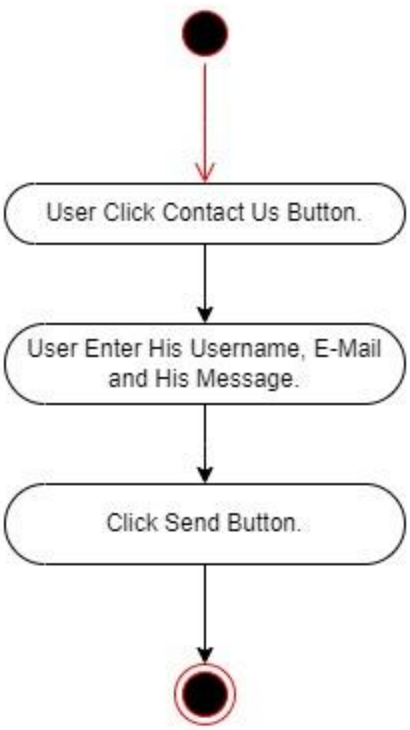


Figure 56 Contact Us Activity Diagram.

6-8 Database Physical Model.

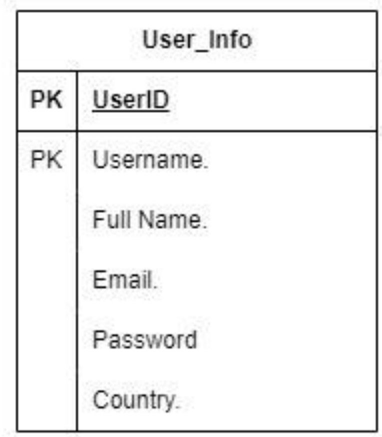


Figure 57 Database Schema

6-9 Database physical Model Description.

Attribute	Datatype	Size	Description.	Key
UserID	Int.	6	The User's ID.	Primary Key
Username	Varchar	15	Unique Sequence of Characters Used to Identify a User	Composite Primary Key.
Full Name.	Varchar	50	The User's Full Name.	
Email	Varchar	45	The User's Email	
Password.	Varchar	30	A Secret Word or Phrase That Must Be Used to Gain Admission to The App.	
Country.	Varchar	15	The User's Country.	

6-10 Database Entities Operations.

Attribute	Operations
Username	• Insert • Select
Password	• Insert • Select • update
Email	• Insert • Select • update
Full Name	• Insert • Select • update
Country	• Insert • Select • update

Implementation.

Chapter Seven.

7-1 Implementing System Modules.

7-1-1 Home Page.

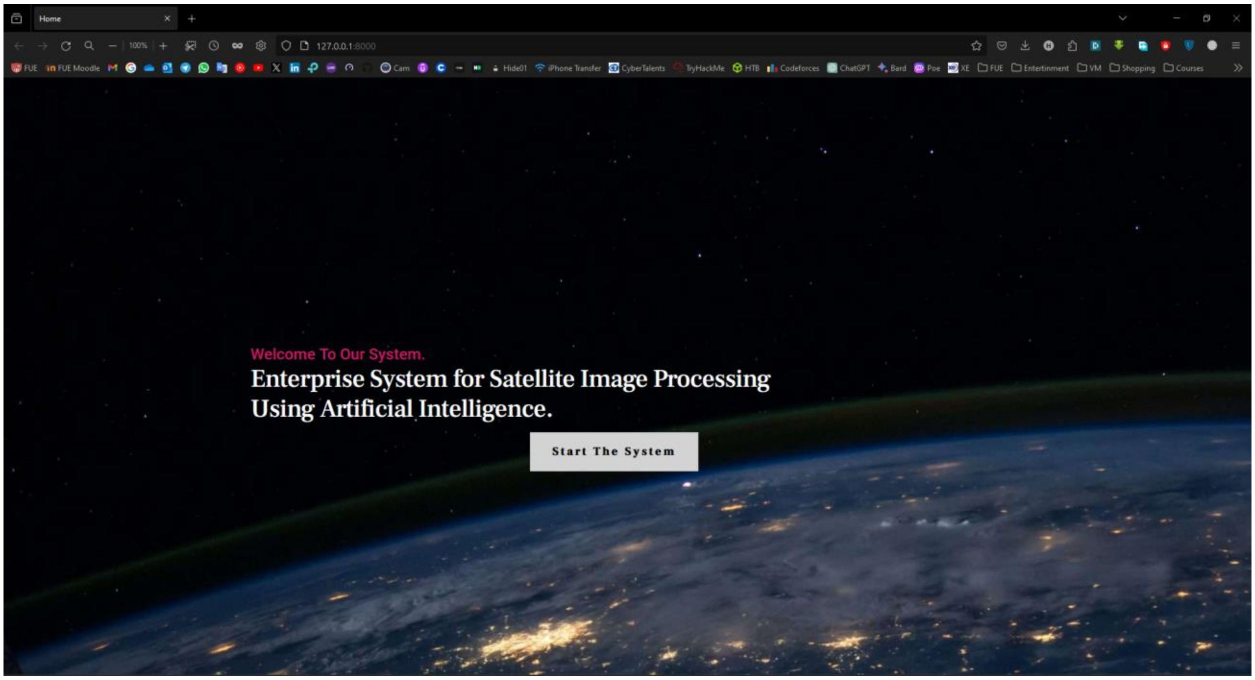


Figure 58 Home Page.

7-1-2 Team Members Page.

7-1-3 Create Account Page:

The user creates an account by filling in the form or Login as Guest.

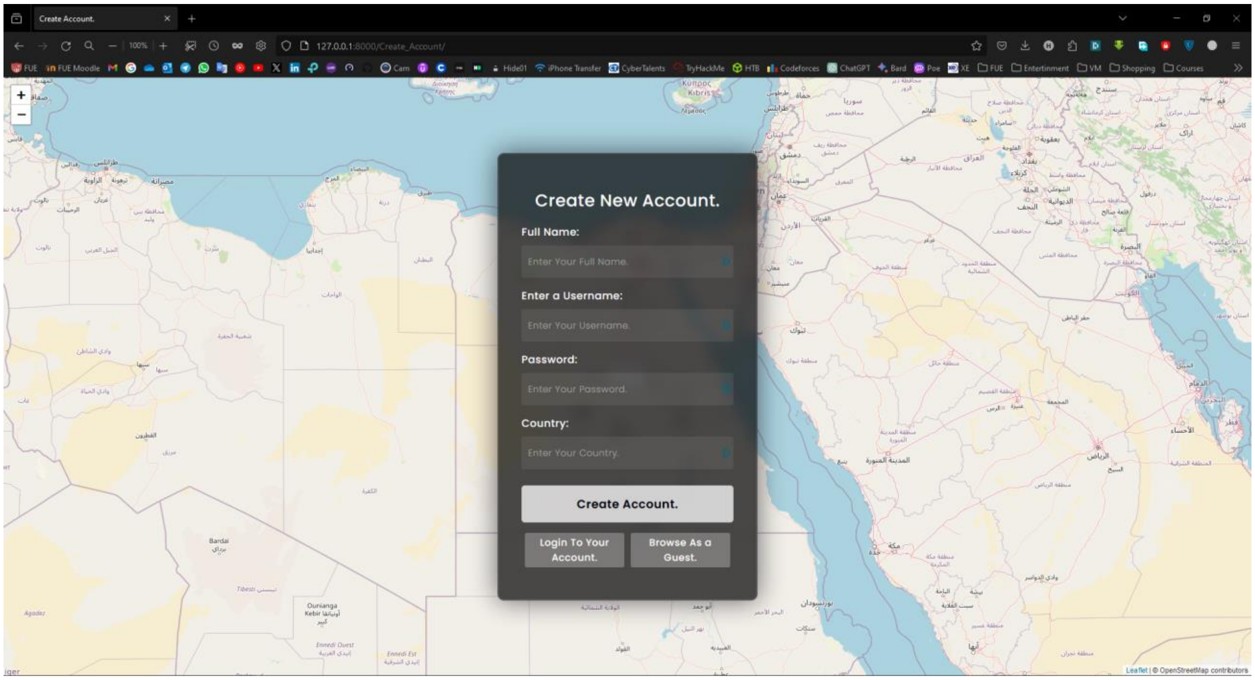


Figure 59 Create Account Page.

7-1-4 Login Page:

The user accesses his account by filling in the form then clicks the button (Login). And can also click (Forgotten Password) Button.

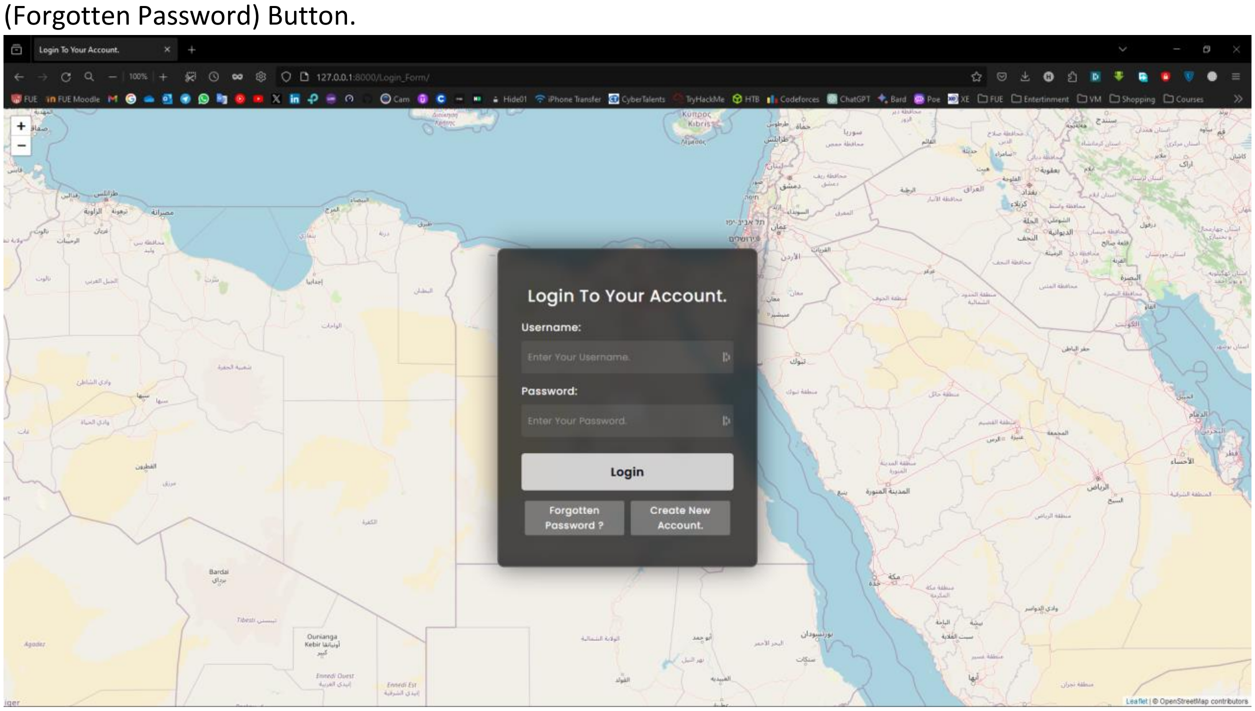


Figure 60 Login Page Page.

7-1-5 Reset Password Page.

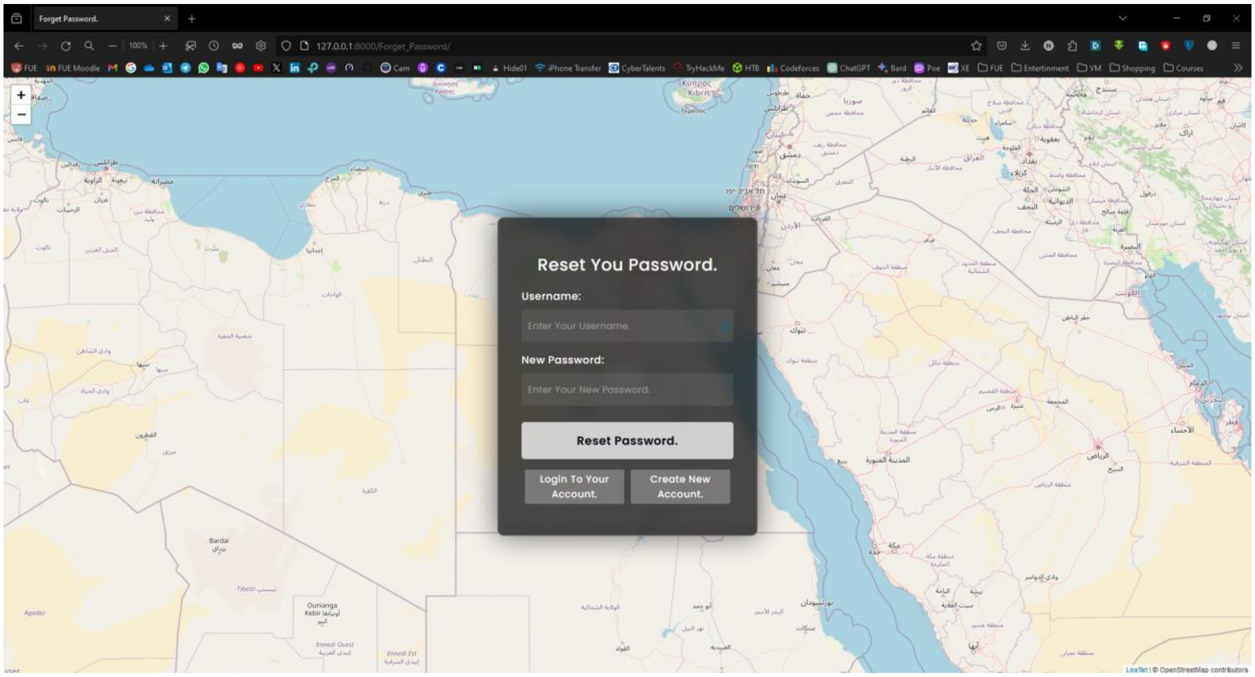


Figure 61 Reset Password Page.

7-1-6 Contact US.

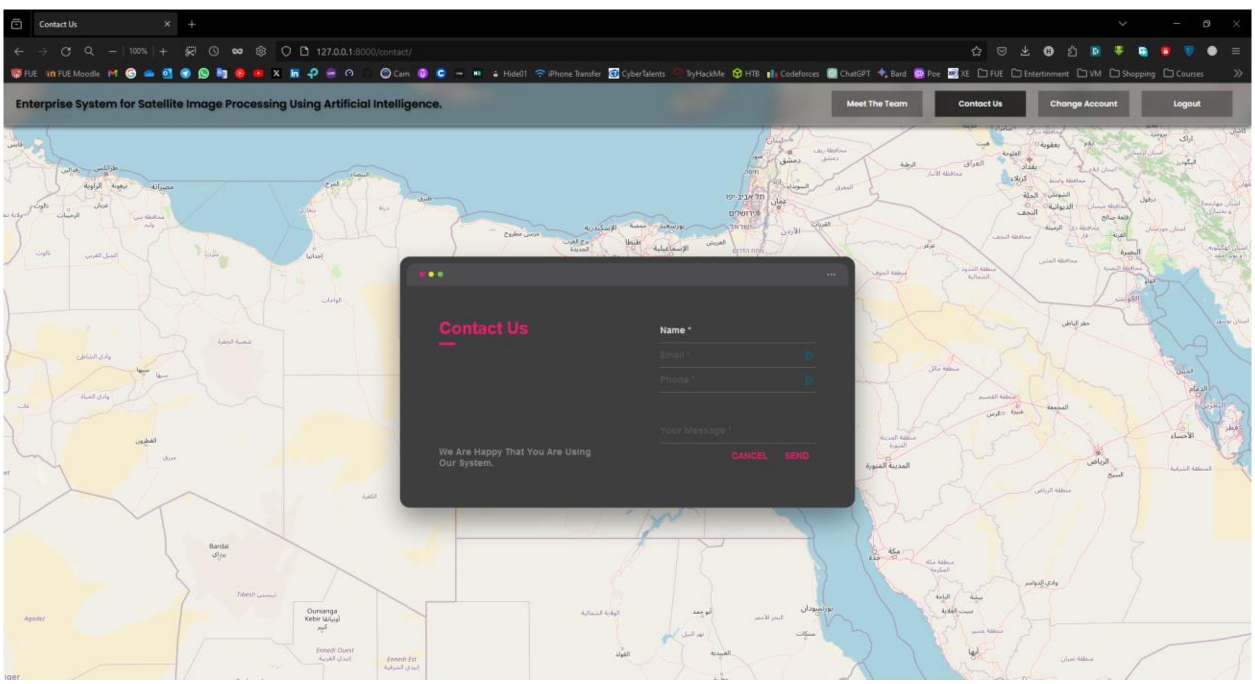


Figure 62 Contact US Page.

7-1-7 System View.

User after creates the account, he enters the view section where there is a map for him to choose the Map Based Layer.

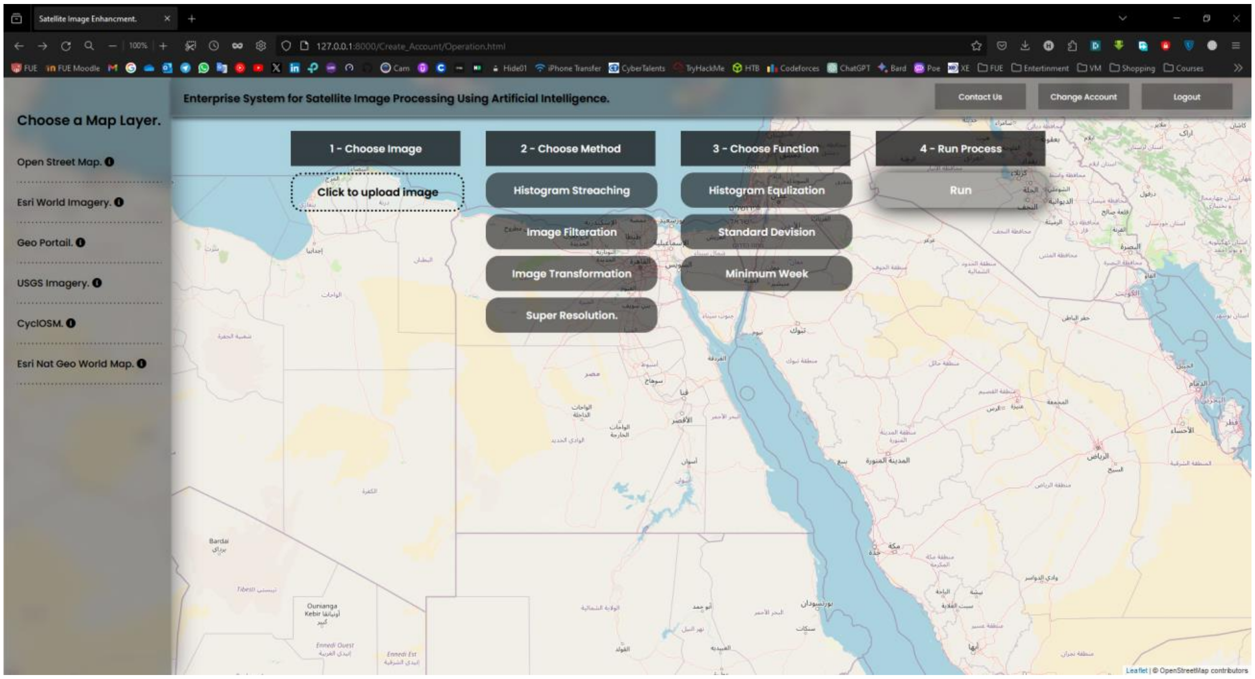


Figure 63 System View.

7-1-8 Map Layer Selection.

The user can choose a variety of 6 different map layers.

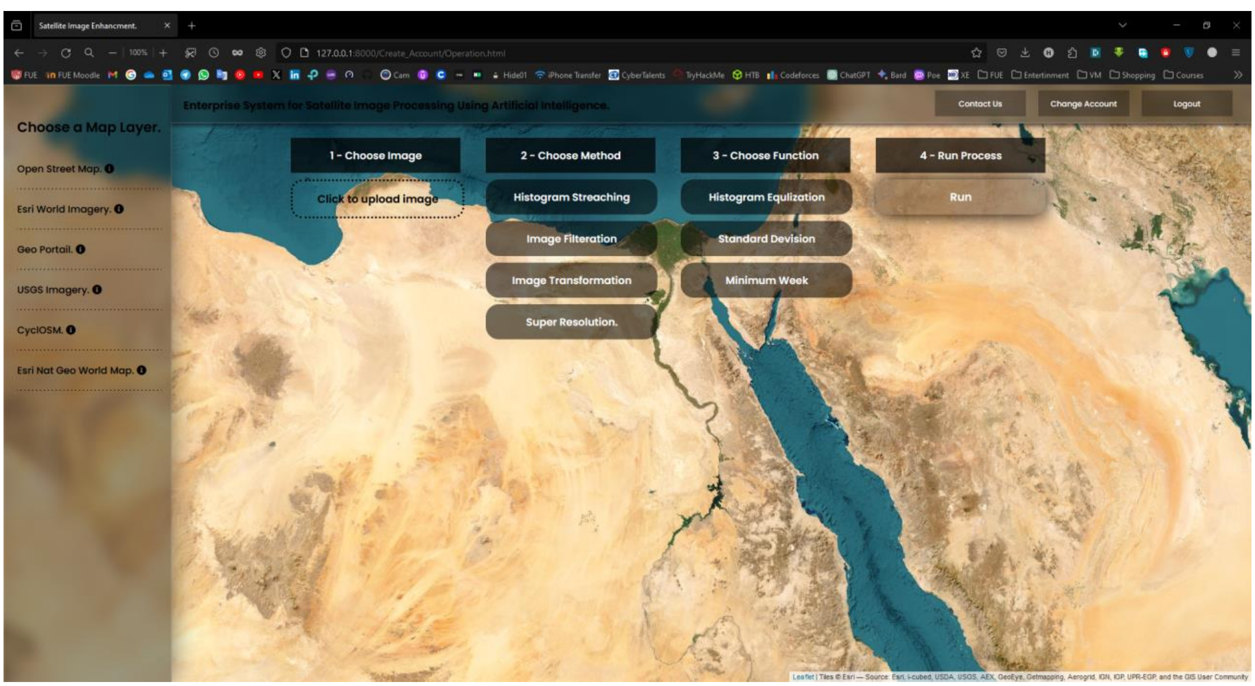


Figure 64 Map Layer Selection.

7-1-9 Map Layer Info.

When the user clicks on the (i) button next to the selected map, information about this map appears.

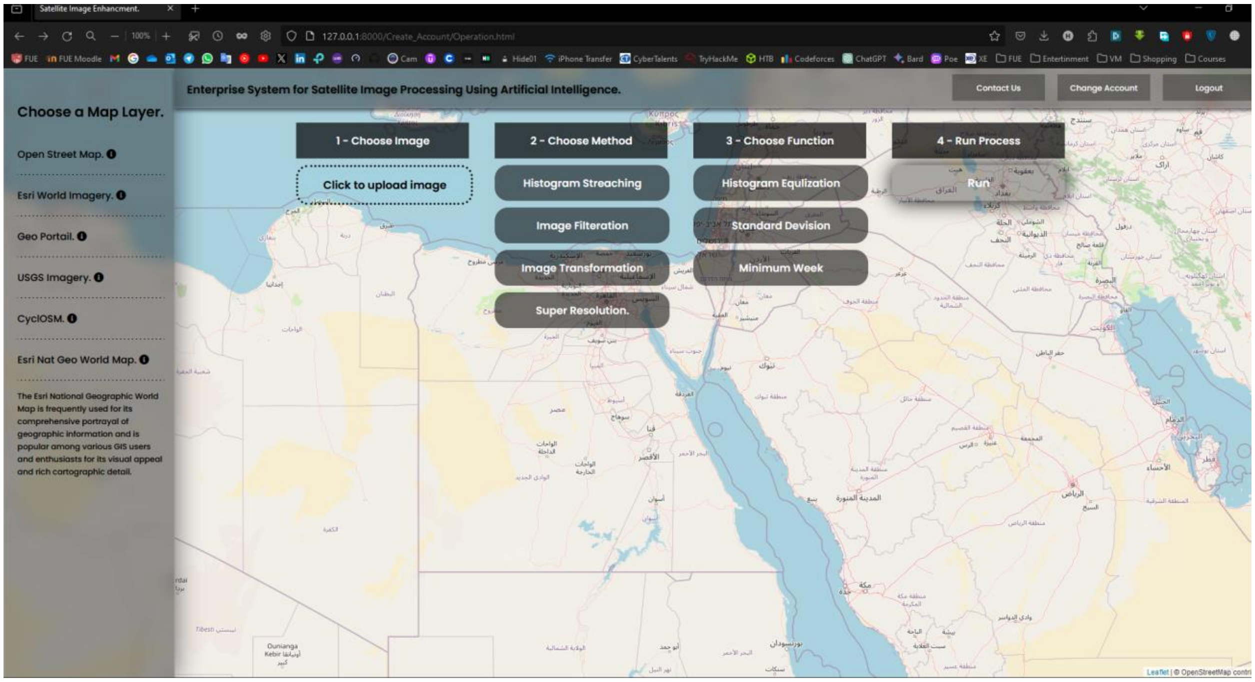


Figure 65 Map Layer Info.

7-1-10 Histogram Stretching View.

The Users can choose Histogram Stretching to User Their Functions.

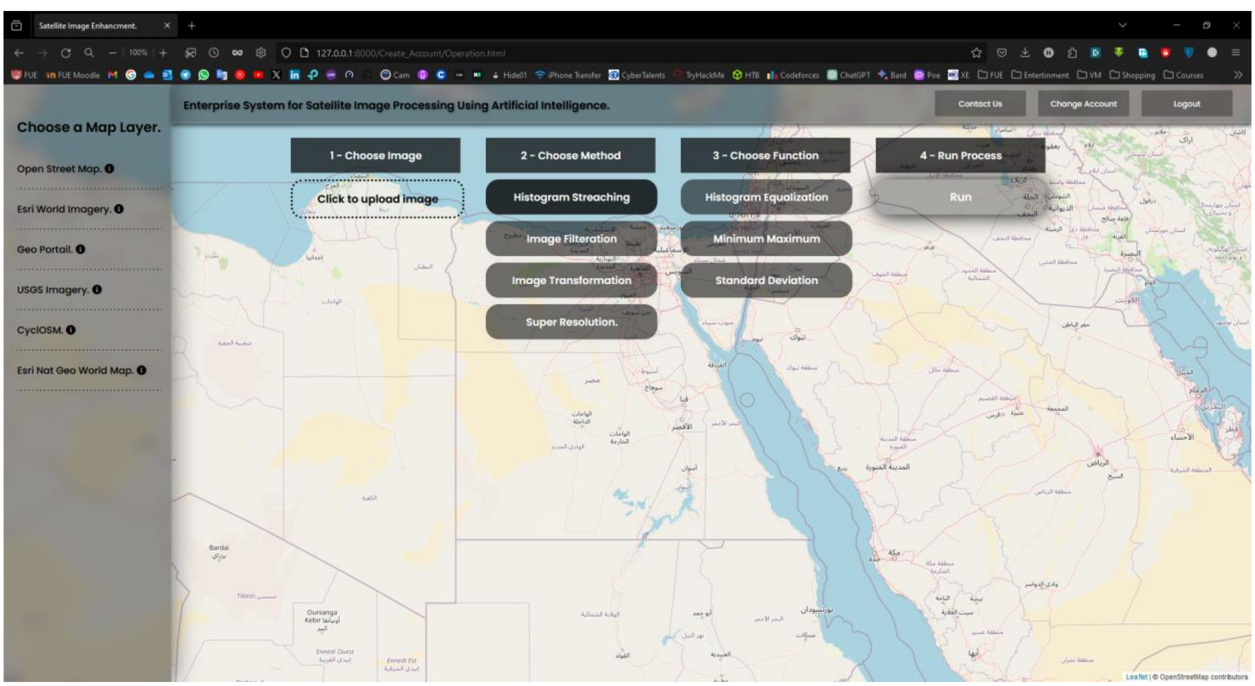


Figure 66 Histogram Stretching View.

7-1-11 Image Filtration View.

The Users can choose Image Filtration to User Their Functions.

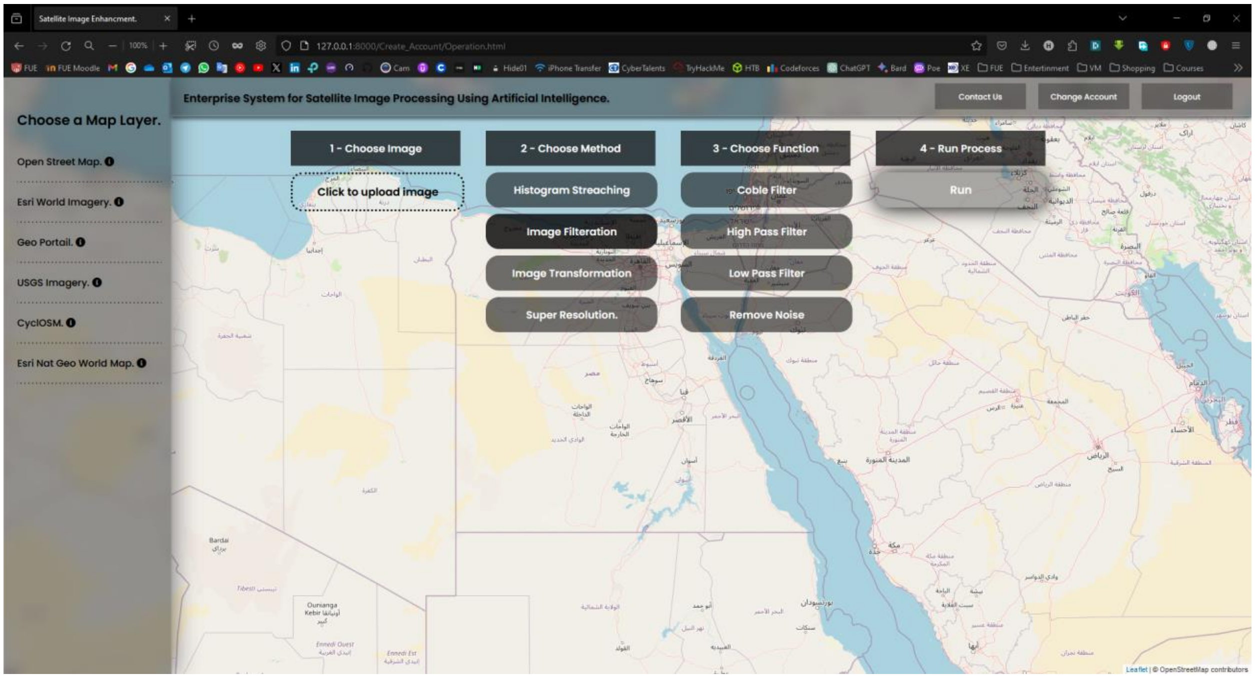


Figure 67 Image Filtration View.

7-1-12 Image Transformation View.

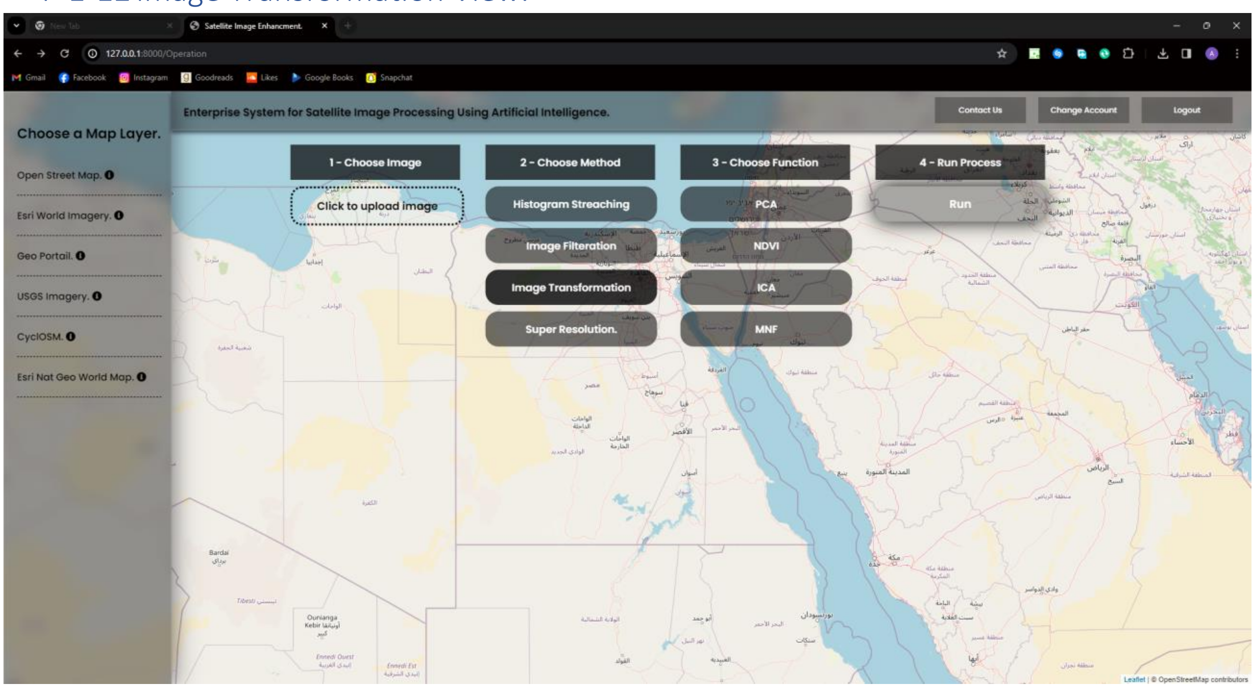


Figure 68 Image Transformation View.

7-1-13 Super Resolution View.

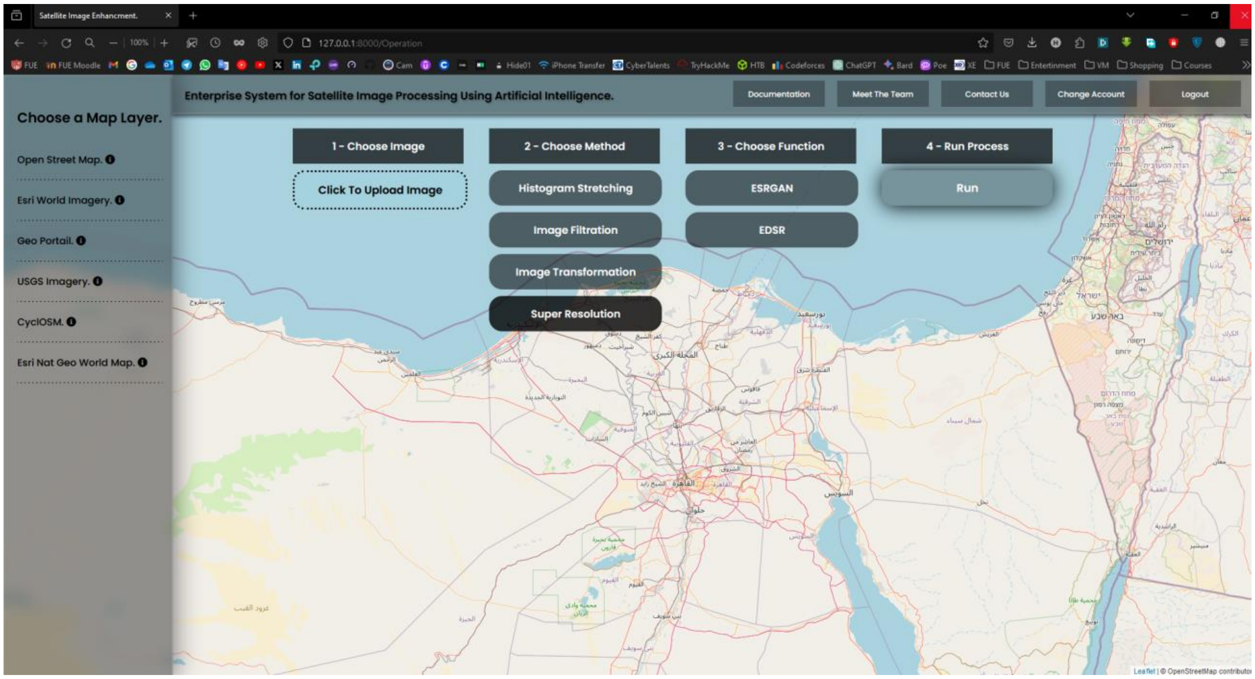


Figure 69 Super Resolution View.

7-1-14 Show Results on the Map Based

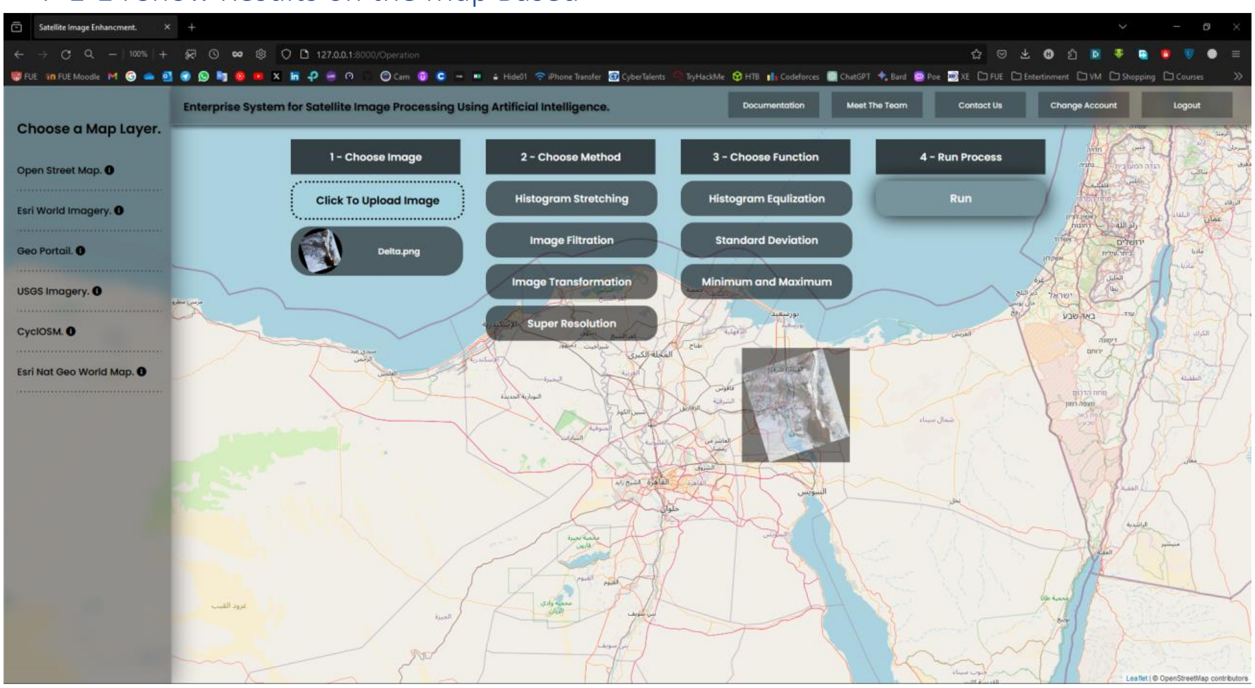


Figure 70 Overlap the Result on The Map Based

7-1-15 Histogram Stretching Method Algorithms.

7-1-15.1 Histogram Equalization Algorithm.

```
import cv2 as cv
import numpy as np
from matplotlib import pyplot as plt

path = "delta.png"
img = cv.imread(path)

cv.imshow('image',img)
cv.waitKey(0)
cv.destroyAllWindows()

hist,bins = np.histogram(img.flatten(),256,[0,256])
cdf = hist.cumsum()
cdf_normalized = cdf * float(hist.max()) / cdf.max()
plt.plot(cdf_normalized, color = 'b')
plt.hist(img.flatten(),256,[0,256], color = 'r')
plt.xlim([0,256])
plt.legend(('cdf','histogram'), loc = 'upper left')
plt.show()

equ = cv.equalizeHist(img)

cv.imshow('equ.png',equ)
cv.waitKey(0)
cv.destroyAllWindows()

```

Figure 71 Histogram Equalization Algorithm

7-1-15.2 Minimum and Maximum Algorithm.

```
import cv2
import numpy as np
import matplotlib.pyplot as pltt

img = cv2.imread('ddd.TIF', cv2.IMREAD_COLOR)

# Split the image into color channels
r, g, b = img[:,:,0], img[:,:,1], img[:,:,2]

# Plot the histograms for each channel
hist_r = np.zeros(256)
hist_g = np.zeros(256)
hist_b = np.zeros(256)

for i in range(img.shape[0]):
    for j in range(img.shape[1]):
      hist_r[r[i,j]] += 1
      hist_g[g[i,j]] += 1
      hist_b[b[i,j]] += 1

pltt.plot(hist_r, color='red', alpha=0.10)
pltt.plot(hist_g, color='green', alpha=0.10)
pltt.plot(hist_b, color='blue', alpha=0.10)
pltt.show()

# Stretch the contrast for each channel
min_r, max_r = np.min(r), np.max(r)
min_g, max_g = np.min(g), np.max(g)
min_b, max_b = np.min(b), np.max(b)
re_stretch = np.zeros((img.shape[0], img.shape[1]), dtype=np.uint8)
gr_stretch = np.zeros((img.shape[0], img.shape[1]), dtype=np.uint8)
bl_stretch = np.zeros((img.shape[0], img.shape[1]), dtype=np.uint8)

for i in range(img.shape[0]):
   for j in range(img.shape[1]):
      re_stretch[i,j] = int((r[i,j] - min_r) * 255 / (max_r - min_r))
      gr_stretch[i,j] = int((g[i,j] - min_g) * 255 / (max_g - min_g))
      bl_stretch[i,j] = int((b[i,j] - min_b) * 255 / (max_b - min_b))

# Merge the channels back together
img_stretch = cv2.merge((re_stretch, gr_stretch, bl_stretch))

# Display the original and stretched images side by side
pltt.subplot(121)
pltt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
pltt.title('Original Image')
```

Figure 72 Minimum and Maximum Algorithm.

7-1-15.3 Standard Deviation Algorithm.

```
import cv2
```

```
import numpy as np

# Load an image
image = cv2.imread('ddd.TIF', cv2.IMREAD_GRAYSCALE)

# Define the standard deviation for stretching
std_dev = 2.0  # You can adjust this value

# Calculate the mean and standard deviation of the image
mean, std = cv2.meanStdDev(image)

# Calculate lower and upper bounds based on standard deviation
lower_bound = int(mean - std_dev * std)
upper_bound = int(mean + std_dev * std)

# Clip pixel values to ensure they fall within the bounds
stretched_image = np.clip(image, lower_bound, upper_bound)

# Normalize the stretched image to the full 0-255 range
stretched_image = cv2.normalize(stretched_image, None, 0, 255,
cv2.NORM_MINMAX)

# Convert to uint8 (8-bit) data type
stretched_image = stretched_image.astype(np.uint8)

# Display the original and stretched images
cv2.imshow('Original Image', image)
cv2.imshow('Stretched Image', stretched_image)
cv2.waitKey(0)
cv2.destroyAllWindows()

# Save the stretched image if needed
cv2.imwrite('stretched_image.jpg', stretched_image)
```

Figure 73 Standard Deviation Algorithm

7-1-16 Image Filtration Method.

7-1-16.1 Soble Filter Algorithm.

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

# Load the image
img0 = cv2.imread('ddd.TIF')

# Resize the image to a smaller size
new_height, new_width = 800, 600  # Adjust the dimensions as needed
img0 = cv2.resize(img0, (new_width, new_height))

# Convert to grayscale
gray = cv2.cvtColor(img0, cv2.COLOR_BGR2GRAY)

# Remove noise using Gaussian blur
img = cv2.GaussianBlur(gray, (3, 3), 0)

# Convolute with proper kernels
laplacian = cv2.Laplacian(img, cv2.CV_64F)
sobelx = cv2.Sobel(img, cv2.CV_64F, 1, 0, ksize=5)  # x
sobely = cv2.Sobel(img, cv2.CV_64F, 0, 1, ksize=5)  # y

# Display the images
plt.subplot(2, 2, 1), plt.imshow(img, cmap='gray')
plt.title('Original'), plt.xticks([]), plt.yticks([])
plt.subplot(2, 2, 2), plt.imshow(laplacian, cmap='gray')
plt.title('Laplacian'), plt.xticks([]), plt.yticks([])
plt.subplot(2, 2, 3), plt.imshow(sobelx, cmap='gray')
plt.title('Sobel X'), plt.xticks([]), plt.yticks([])
plt.subplot(2, 2, 4), plt.imshow(sobely, cmap='gray')
plt.title('Sobel Y'), plt.xticks([]), plt.yticks([])
plt.show()
```

Figure 74 Soble Filter Algorithm.

7-1-16.2 High-Pass Filter Algorithm.

```
import cv2
import numpy as np

# Load the image
image = cv2.imread('Lenna.png', cv2.IMREAD_GRAYSCALE)

# Create a 5x5 kernel for the high-pass filter
kernel = np.array([[-1, -1, -1, -1, -1],
                   [-1,  1,  2,  1, -1],
                   [-1,  2,  4,  2, -1],
                   [-1,  1,  2,  1, -1],
                   [-1, -1, -1, -1, -1]])

# Apply the filter using convolution
high_pass_image = cv2.filter2D(image, -1, kernel)

# Ensure pixel values are in the valid range (0-255)
high_pass_image = np.clip(high_pass_image, 0, 255)

# Save the filtered image
cv2.imwrite('high_pass_filtered_image.jpg', high_pass_image)

```

Figure 75 High-Pass Filter Algorithm.

7-1-16.3 Low-Pass Filter Algorithm.

```
import cv2
import numpy as np

# Load the image
image = cv2.imread('ddd.TIF')

# Define the size and standard deviation of the Gaussian kernel
kernel_size = (5, 5)
sigma = 1.0

# Create the Gaussian kernel
gaussian_kernel = cv2.getGaussianKernel(kernel_size[0], sigma)
gaussian_kernel = gaussian_kernel * gaussian_kernel.T

# Apply the filter using convolution
low_pass_image = cv2.filter2D(image, -1, gaussian_kernel)

# Save the filtered image
cv2.imwrite('low_pass_filtered_image.jpg', low_pass_image)

```

Figure 76 Low-Pass Filter Algorithm.

7-1-16.4 Median Blur Remove Noise Filter Algorithm.

```python
import cv2

# Load the image
image = cv2.imread('Image With Noise.jpg')

# Remove noise using a median filter
filtered_image = cv2.medianBlur(image, 11)

# Save the image
cv2.imwrite('Median_Blur.jpg', filtered_image)
print("Filtered image saved as 'Median_Blur.jpg'")

```

Figure 77 Median Blur Remove Noise Filter Algorithm.

7-1-17 Image Transformation.

7-1-17.1 Independent Component Analysis (ICA) Algorithm.

```python
import numpy as np
import matplotlib.image as mpimg
from sklearn.decomposition import FastICA
import rasterio

def perform_ica(hyper_image, n_components):
    # Reshape the hyper-image into a 2D array (pixels as rows, bands as columns)
    num_pixels = hyper_image.shape[0] * hyper_image.shape[1]
    num_bands = hyper_image.shape[2]
    flattened_image = hyper_image.reshape(num_pixels, num_bands)

    # Perform ICA
    ica = FastICA(n_components=n_components, random_state=0)
    ica_result = ica.fit_transform(flattened_image)

    # Reshape ICA components back to the original image shape
    ica_image = ica_result.reshape(hyper_image.shape)
    return ica_image

# Load satellite image using rasterio (replace 'your_satellite_image.tif' with the path to your image)
with rasterio.open('Delta.png') as src:
    # Read the image data (bands)
    image_bands = src.read()
    metadata = src.meta

# Transpose the image bands to match the shape (bands, rows, columns)
image_bands = np.transpose(image_bands, (1, 2, 0))

# Number of independent components to extract
num_components = 3  # Adjust this as needed

# Perform ICA transformation
ica_result = perform_ica(image_bands, num_components)

# Save the Independent Components as separate image files
for i in range(num_components):
    component = ica_result[:, :, i]
    output_file = f'component_{i+1}.png'
    mpimg.imsave(output_file, component, cmap='gray')
```

Figure 78 Independent Component Analysis (ICA) Algorithm.

7-1-17.2 Principal Component Analysis (PCA) Algorithm.

```python
import cv2
import numpy as np
from sklearn.decomposition import PCA

img = cv2.imread('Delta_1_RGB.tif')  # Remove the color conversion, assume image is in BGR format
blue, green, red = cv2.split(img)

# Convert to float and normalize
df_blue = blue.astype(np.float32) / 255.0
df_green = green.astype(np.float32) / 255.0
df_red = red.astype(np.float32) / 255.0

# Reshape the data for PCA
df_blue_flat = df_blue.reshape(-1, df_blue.shape[-1])
df_green_flat = df_green.reshape(-1, df_green.shape[-1])
df_red_flat = df_red.reshape(-1, df_red.shape[-1])

# PCA for each channel
pca_b = PCA(n_components=50)
pca_b.fit(df_blue_flat)
trans_pca_b = pca_b.transform(df_blue_flat)
inv_pca_b = pca_b.inverse_transform(trans_pca_b).reshape(df_blue.shape)

pca_g = PCA(n_components=50)
pca_g.fit(df_green_flat)
trans_pca_g = pca_g.transform(df_green_flat)
inv_pca_g = pca_g.inverse_transform(trans_pca_g).reshape(df_green.shape)

pca_r = PCA(n_components=50)
pca_r.fit(df_red_flat)
trans_pca_r = pca_r.transform(df_red_flat)
inv_pca_r = pca_r.inverse_transform(trans_pca_r).reshape(df_red.shape)

# Scale back to 0-255 range
b_arr = np.clip(inv_pca_b * 255.0, 0, 255).astype(np.uint8)
g_arr = np.clip(inv_pca_g * 255.0, 0, 255).astype(np.uint8)
r_arr = np.clip(inv_pca_r * 255.0, 0, 255).astype(np.uint8)

# Merge the channels
img_reduced = cv2.merge((b_arr, g_arr, r_arr))

# Save the compressed image as a TIFF file
cv2.imwrite('PCAcompressed_image.tif', img_reduced)

```

Figure 79 Principal Component Analysis (ICA) Algorithm.

7-1-17.3 Normalized Difference Vegetation Index (NDVI) Algorithm.

```python
import numpy as np
from skimage import io

# Load an RGB image
image = io.imread('Delta.png')  # Replace with the path to your image

# Assuming the image is in the form of (height, width, channels)
if len(image.shape) == 3 and image.shape[2] >= 3:  # Ensure it's a color image with 3 channels
    # Extract individual bands (assuming the image is in RGB format)
    red = image[:, :, 0].astype(np.float32)
    nir = image[:, :, 2].astype(np.float32)

    # Calculate NDVI
    ndvi = (nir - red) / (nir + red + 1e-8)  # Adding a small value to prevent division by zero

    # Scale NDVI to 0-255 and convert to uint8 for image saving
    ndvi_scaled = ((ndvi + 1) * 127.5).astype(np.uint8)

    # Save NDVI as an image file (adjust the file format if needed)
    output_file = 'ndvi_output.tif'
    io.imsave(output_file, ndvi_scaled)

    print(f"NDVI saved as {output_file}")
else:
    print("The loaded image doesn't seem to have the expected shape for RGB color image.")

```

Figure 80 Normalized Difference Vegetation Index (NDVI) Algorithm.

7-1-17.4 Minimum Noise Fraction (MNF) Algorithm.

```python
import numpy as np
from sklearn.decomposition import PCA
import rasterio
import cv2

# Function to perform MNF transformation
def mnf_transform(hyper_image):
    num_pixels = hyper_image.shape[0] * hyper_image.shape[1]
    num_bands = hyper_image.shape[2]
    flattened_image = hyper_image.reshape(num_pixels, num_bands)

    pca = PCA(n_components=num_bands)
    pca.fit(flattened_image)

    mnf_components = pca.components_.T
    mnf_image = np.dot(flattened_image,mnf_components).reshape(hyper_image.shape)

    return mnf_image

with rasterio.open('ddd.TIF') as src:
    image_bands = src.read()

image_bands = np.transpose(image_bands, (1, 2, 0))

mnf_result = mnf_transform(image_bands)

# Remove singleton dimension if exists
mnf_result = np.squeeze(mnf_result)

# Save MNF-transformed image as a TIFF file
output_file = 'mnf_transformed_image.tif'
with rasterio.open(
    output_file,
    'w',
    driver='GTiff',
    height=mnf_result.shape[0],
    width=mnf_result.shape[1],
    count=mnf_result.shape[2] if len(mnf_result.shape) == 3 else 1,
    dtype=mnf_result.dtype
) as dst:
    dst.write(mnf_result, 1)  # Write the MNF-transformed image data to the file

print(f"MNF-transformed image saved as {output_file}")

# Load the saved image and compress it
with rasterio.open(output_file) as src:
    img = src.read(1)  # Read the first band assuming it's grayscale
```

```python

# Convert the image to 8-bit depth
img_8bit = cv2.normalize(img, None, 0, 255, cv2.NORM_MINMAX, dtype=cv2.CV_8U)

# Define the new dimensions for compression
new_width = 500  # Replace with desired width
new_height = 500  # Replace with desired height

# Check if the image is loaded successfully
if img_8bit is not None:
    # Compress the image
    compressed_img = cv2.resize(img_8bit, (new_width, new_height))

    # Save the compressed image
    compressed_output_file = 'compressed_mnf_transformed_image.jpg'
    cv2.imwrite(compressed_output_file, compressed_img)

    print(f"Compressed image saved as {compressed_output_file}")
else:
    print("Error loading the image")

```
Figure 81 Minimum Noise Fraction (MNF) Algorithm.

7-1-18 Super Resolution.

7-1-18.1 ESRGAN (2X).

```
import os
import time
from PIL import Image
import numpy as np
import tensorflow as tf
import tensorflow_hub as hub
import matplotlib.pyplot as plt
os.environ["TFHUB_DOWNLOAD_PROGRESS"] = "True"

# Declaring Constants
IMAGE_PATH = "Image With Noise.jpg"
SAVED_MODEL_PATH = "https://tfhub.dev/captain-pool/esrgan-tf2/1"

def preprocess_image(image_path):
  """ Loads image from path and preprocesses to make it model ready
      Args:
        image_path: Path to the image file
  """
  hr_image = tf.image.decode_image(tf.io.read_file(image_path))
  # If PNG, remove the alpha channel. The model only supports
  # images with 3 color channels.
  if hr_image.shape[-1] == 4:
    hr_image = hr_image[...,:-1]
  hr_size = (tf.convert_to_tensor(hr_image.shape[:-1]) // 4) * 4
  hr_image = tf.image.crop_to_bounding_box(hr_image, 0, 0, hr_size[0], hr_size[1])
  hr_image = tf.cast(hr_image, tf.float32)
  return tf.expand_dims(hr_image, 0)

def save_image(image, filename):
  """
    Saves unscaled Tensor Images.
    Args:
      image: 3D image tensor. [height, width, channels]
      filename: Name of the file to save.
  """
  if not isinstance(image, Image.Image):
    image = tf.clip_by_value(image, 0, 255)
    image = Image.fromarray(tf.cast(image, tf.uint8).numpy())
  image.save("%s.jpg" % filename)
  print("Saved as %s.jpg" % filename)

def plot_image(image, title=""):
  """
    Plots images from image tensors.
    Args:
      image: 3D image tensor. [height, width, channels].
      title: Title to display in the plot.
  """
```

```
    image = np.asarray(image)
    image = tf.clip_by_value(image, 0, 255)
    image = Image.fromarray(tf.cast(image, tf.uint8).numpy())
    plt.imshow(image)
    plt.axis("off")
    plt.title(title)

hr_image = preprocess_image(IMAGE_PATH)

# Plotting Original Resolution image
plot_image(tf.squeeze(hr_image), title="Original Image")
save_image(tf.squeeze(hr_image), filename="Original Image")
```

7-1-18.2 GAN (4X).

```
from flask import *
import uuid
import os
from gandeblur import *
from denoise import *
from denoise import denoisee
from keras import backend as K
import logging
import shutil

logger = logging.getLogger()
logger.disabled = False
ALLOWED_EXTENSIONS = set(['png', 'jpg', 'jpeg'])
UPLOAD_FOLDER=os.path.join('static','uploads')
app = Flask(__name__)
app.config['UPLOAD_FOLDER']=UPLOAD_FOLDER



@app.route('/')
def upload():
    return render_template("index.html")
@app.route('/den')
def den():
    return render_template("denoise.html")
@app.route('/process', methods = ['POST'])
def success():
    if request.method == 'POST':
        f = request.files['formfile']
        f_ext=f.filename.split('.')[1]
        if(f_ext in ALLOWED_EXTENSIONS):

            #print(type(f))
            unique_filename = str(uuid.uuid4())
            unique_file=unique_filename+'.'+f_ext
            f.save(os.path.join(app.config['UPLOAD_FOLDER'],unique_file))
            K.clear_session()
```

```
            deblur2('generator.h5',os.path.join(app.config['UPLOAD_FOLDER'],unique_file),'',unique_file)
            before_pic=os.path.join(app.config['UPLOAD_FOLDER'],unique_file)
            after_pic=os.path.join(app.config['UPLOAD_FOLDER'],"generated_"+unique_file)
            return render_template("responseblur.html", name = f.filename, before = before_pic, after = after_pic)
        else:
            return """File format has to be one of the following:
            <br/> 1 .png
            <br/> 2 .jpg
            <br/> 3 .jpeg
            <br/> The file uploaded was in <strong>{}</strong> format""".format(f_ext)

@app.route('/denoise', methods = ['POST'])
def denois():

    if request.method == 'POST':
        f=request.files['formfile']
        f_ext=f.filename.split('.')[1]
        if(f_ext in ALLOWED_EXTENSIONS):
            #print(type(f))
            unique_filename = str(uuid.uuid4())
            unique_file=unique_filename+'.'+f_ext
            f.save(os.path.join(app.config['UPLOAD_FOLDER'],unique_file))
            denoisee('denoise_model.h5',os.path.join(app.config['UPLOAD_FOLDER'],'',unique_file,))
            before_pic=os.path.join(app.config['UPLOAD_FOLDER'],unique_file)
            after_pic=os.path.join(app.config['UPLOAD_FOLDER'],"denoised_"+unique_file)
            return render_template("success.html", name = f.filename, before = before_pic, after = after_pic)

if __name__ == '__main__':
    app.run(host ='0.0.0.0',debug = True)

```

Testing and Evaluation.

Chapter Eight.

8-1 Test Cases.

8-1-1 Sign in.

Sign in Testing Case

Sign in.			
Input	Except	Actual	Result
Empty	Cannot Sign In	"Please Fill Out This Field" Message	True
All Fields	Successfully Sign In	“Sign In Successfully”	True

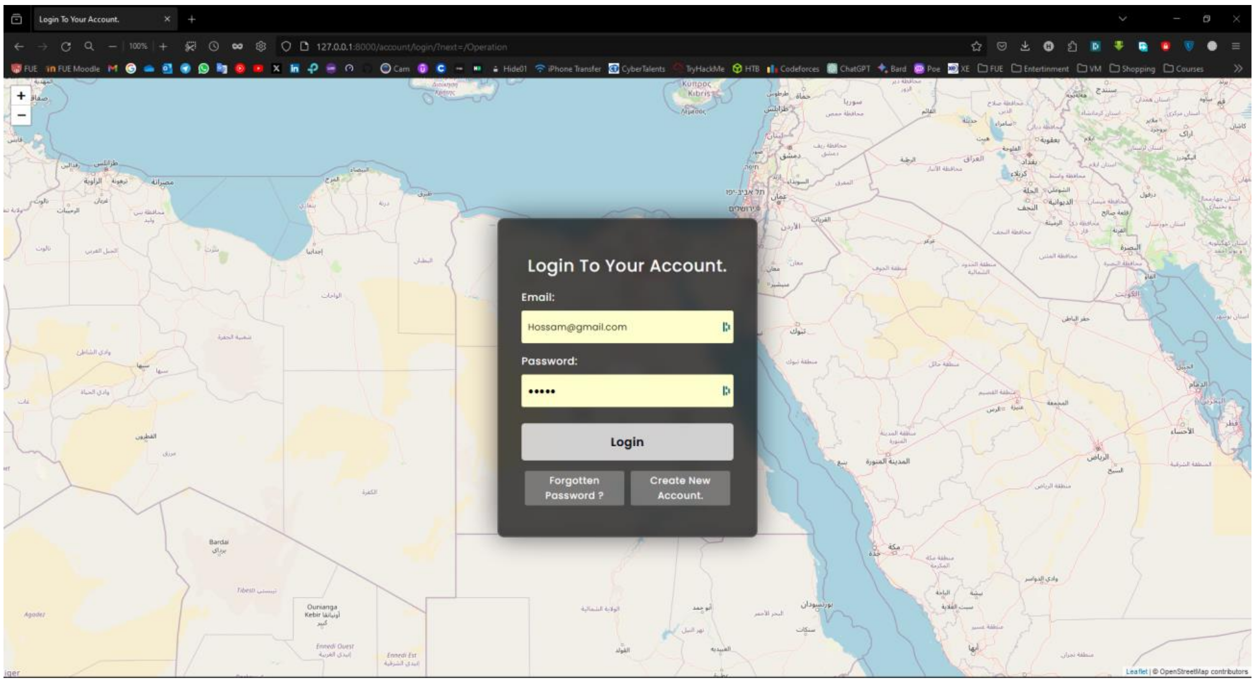


Figure 82 Sign in Successfully.

8-1-2 Sign Up.

Sign Up Testing Case.

Sign Up			
Input	Except	Actual	Result
Empty	Can't Signup	" The Field Is Required to Register " Message	True
All Fields	Successfully	"Signup P Successfully"	True

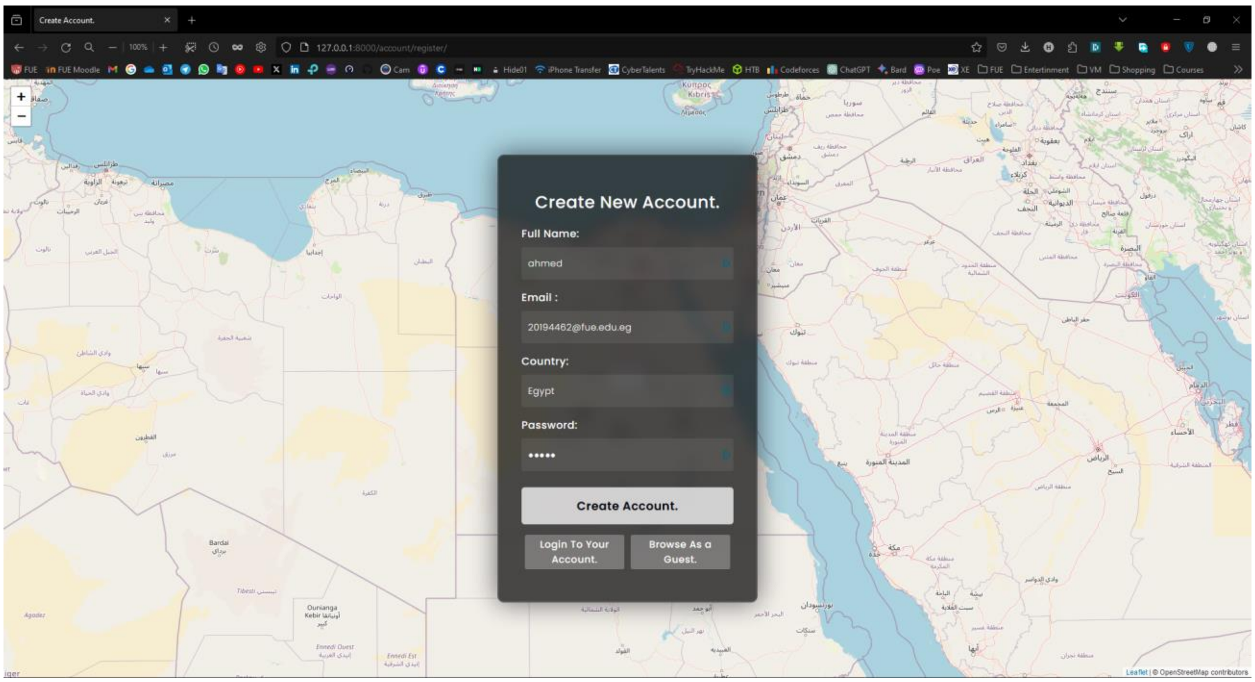


Figure 83 Sign Up Successfully.

8-1-3 Reset Password.

Edit Profile Testing Case.

Reset Password.			
Input	Except	Actual	Result
If “Profile” Button Is Pressed on And Update Your Password, Then Click On “Save”	Password Can Be Updated	The Password Will Be Updated	True

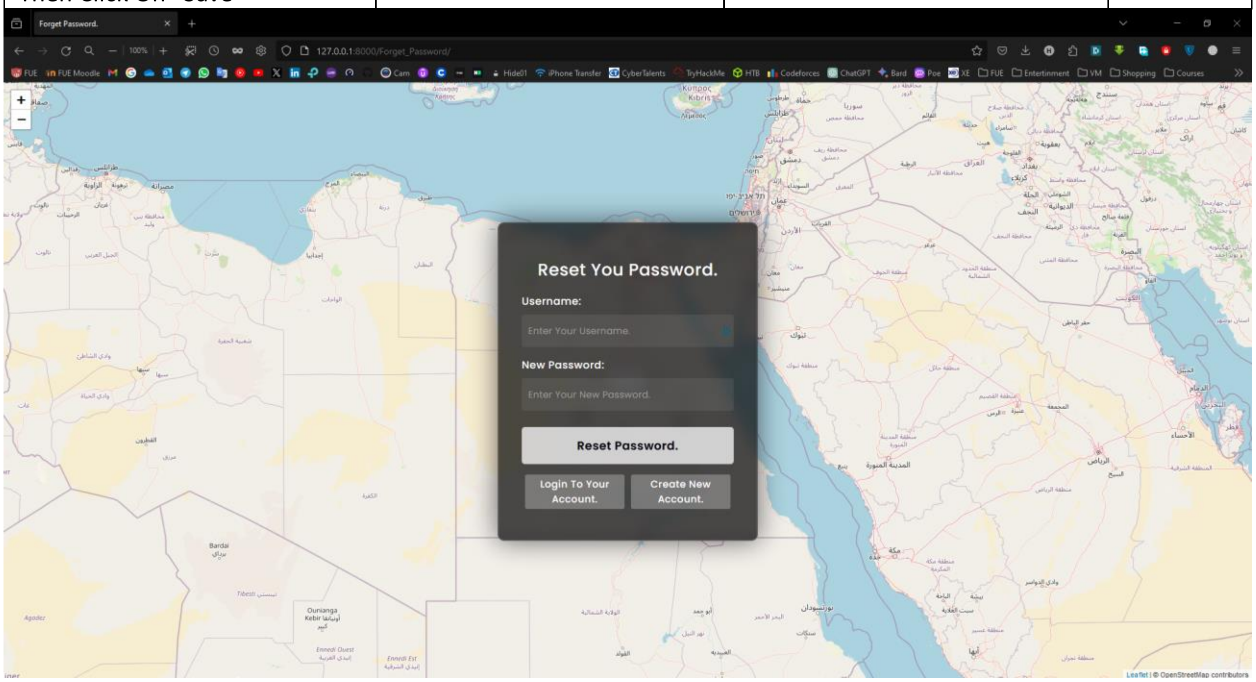


Figure 84 Reset Password Result.

8-1-4 Overlap The Input Photo to The Map Based.

Overlap The Output Photo to The Map Based.			
Input	Except	Actual	Result
The user initiates the "Overlap the Output Photo to The Map Based" feature by uploading a photo and providing specific location details on the map. The user then clicks the "Run" button to generate a combined output.	1 The user fails to upload a photo. 2 The user neglects to specify location details on the map. 3 The uploaded photo format is incompatible or unsupported. 4 There's an issue with the map data, such as invalid coordinates.	The system processes the user's input, validates the photo and location details, and overlays the uploaded photo onto the specified location on the map. It ensures proper alignment and proportionality.	True

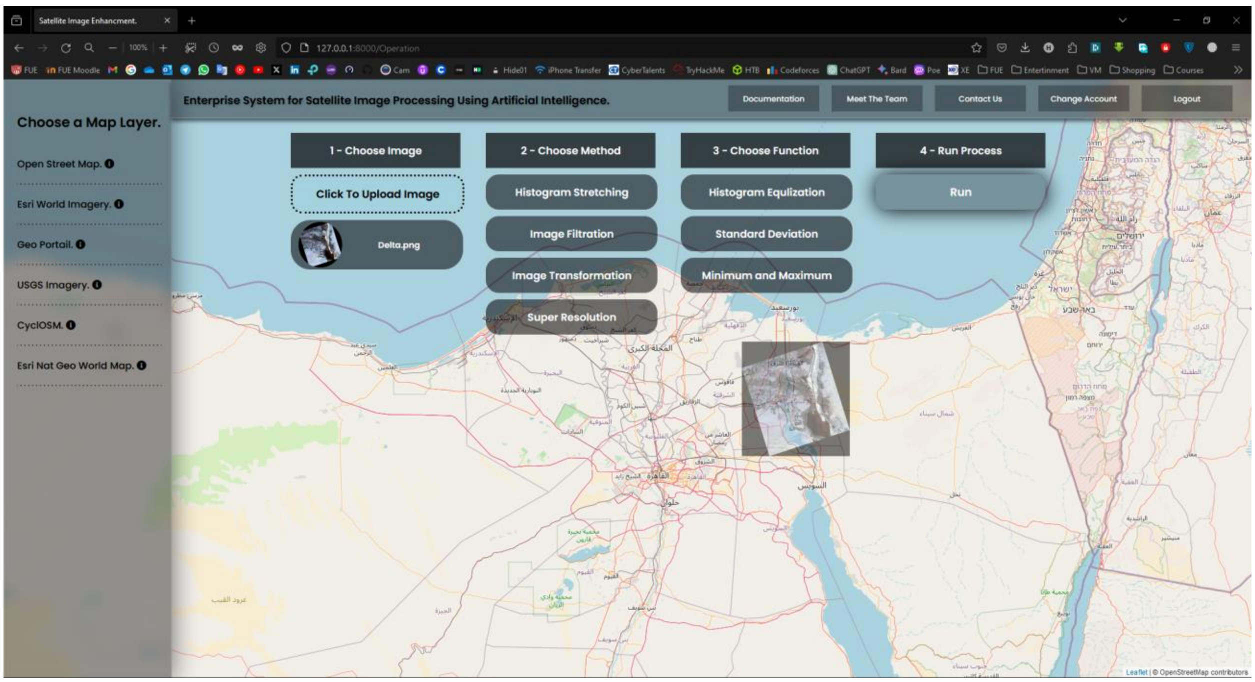


Figure 85 Overlap the Input Photo to Map Layer.

8-1-5 Overlap The Output Photo to The Map Based.

Overlap The Output Photo to The Map Based.			
Input	Except	Actual	Result
The user uploads a photo and specifies location details on the map, then clicks the "Overlap" button	The Metadata fails to provide location details, or the uploaded photo is incompatible.	The system overlays the uploaded photo onto the specified location on the map.	True

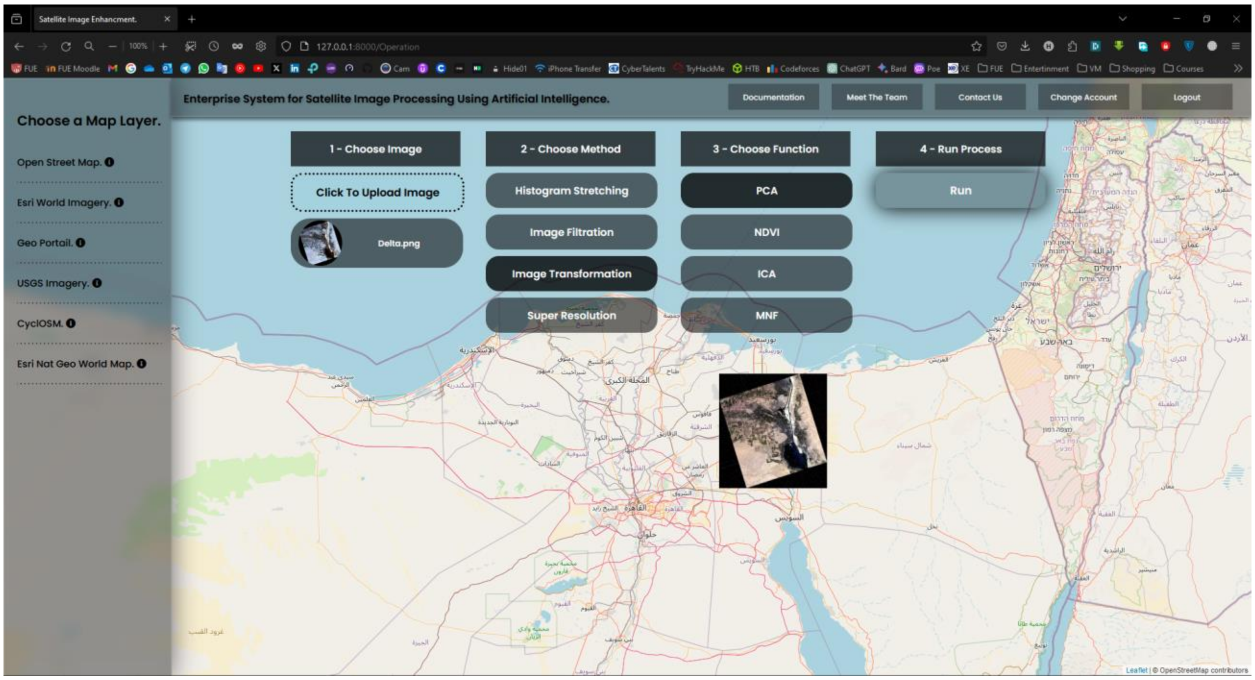


Figure 86 Overlap the Output Photo to Map Layer.

8-1-6 Contact Us.

Contact Us.			
Input	Except	Actual	Result
The user entered all the required fields and the message he wants to send and then clicked "send" button.	An e-mail is sent to the developer with the user's message	E-mail successfully sent	True

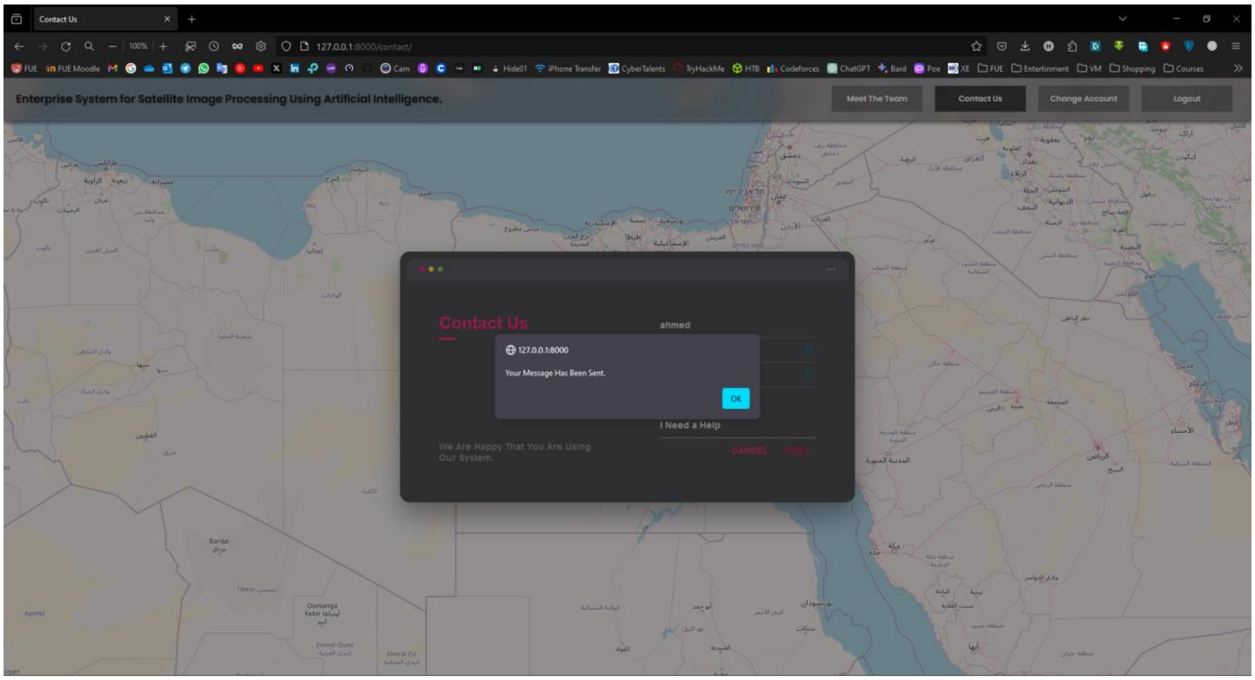


Figure 87 Contact Us Email Sent.

8-1-7 Logout.

Logout.			
Input	Except	Actual	Result
the user clicks on logout Button.	The system logs the user out	User successfully logs out	True

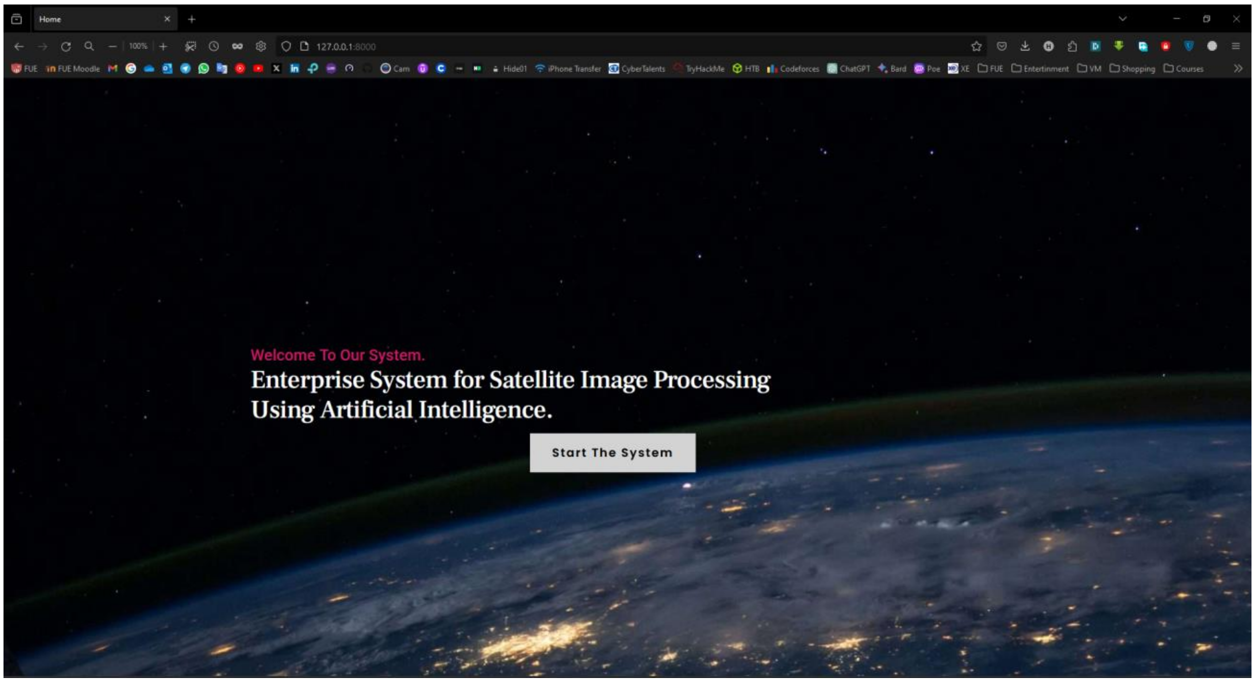


Figure 88 Logout Successfully.

Conclusions and Future Work.

Chapter Nine.

9-1 Conclusion.

In conclusion, the integration of artificial intelligence (AI) systems in satellite image processing has proven to be a transformative advancement, enhancing the analysis and interpretation of vast amounts of satellite data. Through the application of sophisticated methods such as histogram stretching, image filtering, image transformation, and super resolution, these AI-powered systems have demonstrated their efficacy in extracting meaningful information from satellite imagery.

In essence, the synergy between AI systems and satellite image processing methods has opened new frontiers in remote sensing and Earth observation. The amalgamation of these technologies not only expedites data analysis but also augments the accuracy and reliability of insights derived from satellite imagery. As we continue to advance in this field, the symbiotic relationship between AI and satellite image processing methods promises to propel our understanding of Earth's dynamics and contribute to a myriad of scientific, environmental, and societal applications.

9-2 Future Work.

- Optimize the system to become more user-friendly by implementing an API to capture the dataset from the map directly.
- Optimizing our system by minimizing the processing time by upgrading the system hardware (i.e., GPU) and utilizing parallel processing.
- Further modify the system according to unit testing, the expert system testing and the user testing.
- The system can be implemented to become real time object detection.

References.

Chapter Ten.

10 References.

1 Image Processing Techniques for Analysis of Satellite Images for Historical Maps Classification—An Overview by Anju Asokan, J. Anitha, Monica Ciobanu, Andrei Gabor, Antoanela Naaji, D. Jude Hemanth.

2 Remote Sensing Satellite Image Processing Techniques for Image Classification: A Comprehensive Survey by Sowmya D. R, P. Deepa Shenoy, Venugopal K. R. 11, March 2017.

3 Coastline Extraction using Satellite Imagery and Image Processing Techniques by Saeed AL-Mansoori* and Fatima AL-Marzouqi Vol.6, No.4 (Aug 2016).

4 Advancements in satellite image classification: methodologies, techniques, approaches, and applications by Guy Armel Fotso, Laurent Bitjoka, Tallha Akram, Mengue Mbom, 04 Jul 2021.

5 Satellite Image Processing for Land Use and Land Cover Mappin Ashoka Vanjare1+, S.N. Omkar1*, J.Senthilnath 2014, 10, 18-28.

6 High performance image processing of satellite images using graphics processing units, Michal Rumanek; Tomasz Danek; Andrzej Lesniak. 24-29 July 2011.

7 High Resolution Satellite Image Processing by Roshan Rajak; Deepu Raveendran; Maruthi Chandrasekhar Bh; Shanti Swarup Medasani 25-27 November 2015.

8 Multi-spectral Satellite Image Processing by M. Fleury, R. P. Self, and A. C. Downton

9 Satellite Imagery Cadastral Features Extractions using Image Processing Algorithms by Usman Babawuro, Zou Beiji 2, July 2012.

10 Image processing of Sentinel-1 SAR (S-1) and Sentinel-2 multispectral (MSI) (S-2) data for applications in landscape archaeology by Jan Haas, 2023-08-31.

11 "Deep Learning-based Satellite Image Classification for Urban Land Cover Mapping" by Sarah A. Thompson, John M. Sullivan, and Emily R. Foshee, IEEE Transactions on Geoscience and Remote Sensing, 2020.

12 "Semantic Segmentation of High-Resolution Satellite Images using Convolutional Neural Networks" by Maria P. Rodriguez, Carlos A. Gomez, and Juan M. Gomez, Remote Sensing Letters, 2018.

13 "A Review of Machine Learning Approaches for Hyperspectral Image Processing" by Liangpei Zhang, Qi Wang, and Chengquan Huang, IEEE Transactions on Geoscience and Remote Sensing, 2016.

14 "Satellite Image Super-Resolution using Generative Adversarial Networks" by Chen Chen, Qiang Zhang, and Jia Wang, Remote Sensing, 2021.

15 "Object Detection in Satellite Images using Convolutional Neural Networks" by Amanda L. Silva, Roberto M. Cesar Jr., and Claudio Rosito Jung, ISPRS Journal of Photogrammetry and Remote Sensing, 2018.

16 "Automated Cloud Detection in Satellite Imagery using Machine Learning" by Xiaoxiang Zhu, He Wang, and Lei Zhang, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2017.

17 "Enhancing Satellite Image Resolution using Generative Adversarial Networks with Attention Mechanism" by Yujia Xie, Jiaojiao Yang, and Yifan Zhang, Remote Sensing, 2022.

18 "Change Detection in Satellite Images using Convolutional Neural Networks and Recurrent Neural Networks" by Jun Chen, Lianru Gao, and Yanning Zhang, International Journal of Remote Sensing, 2017.

19 "Feature Learning for Hyperspectral Image Classification: A Comparative Review" by Shuai Li, Wei Wang, and Qingxi Tong, IEEE Transactions on Geoscience and Remote Sensing, 2014.

20 "Deep Feature Learning for Hyperspectral Image Classification: A Review" by Qian Du, Yushi Zhao, and Liangpei Zhang, Remote Sensing, 2019.

21 "Spectral-Spatial Classification of Hyperspectral Images using 3D Convolutional Neural Networks" by Wei Li, Hui Lin, and Yu Zhang, IEEE Transactions on Geoscience and Remote Sensing, 2017.

22 "Satellite Image Dehazing using Generative Adversarial Networks" by Meng Ding, Hongyang Lu, and Tianyu Hu, Remote Sensing, 2023.

23 "Machine Learning Approaches for Cloud Detection in Satellite Images" by Xiaotong Zhang, Yingchun Li, and Liangpei Zhang, International Journal of Remote Sensing, 2016.

Manuals and User Guide.

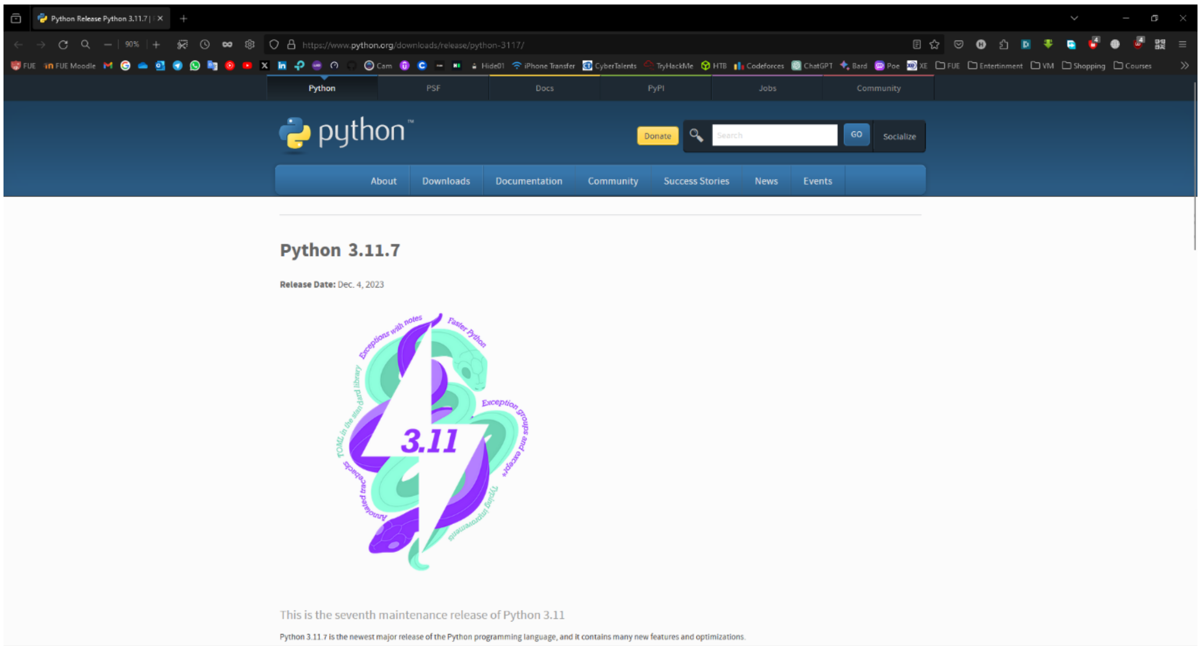
Chapter Eleven.

11 Environment Setup.

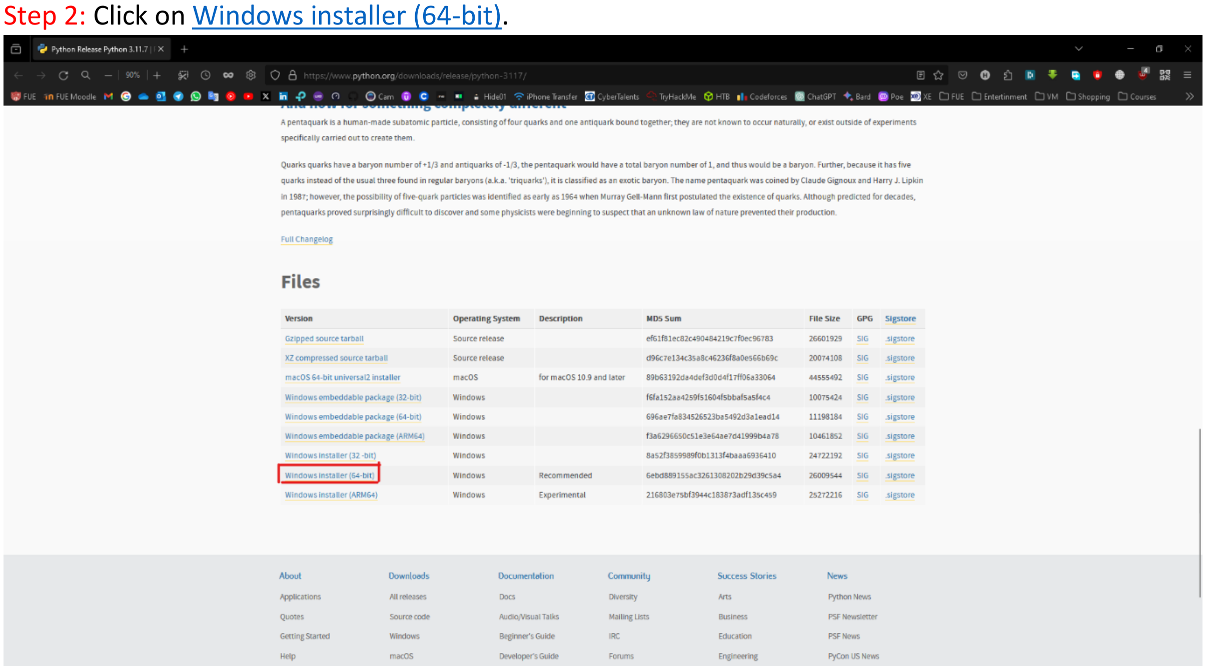
11-1 Python installation.

This section describes how to install Python 3.11.7 on the local system.

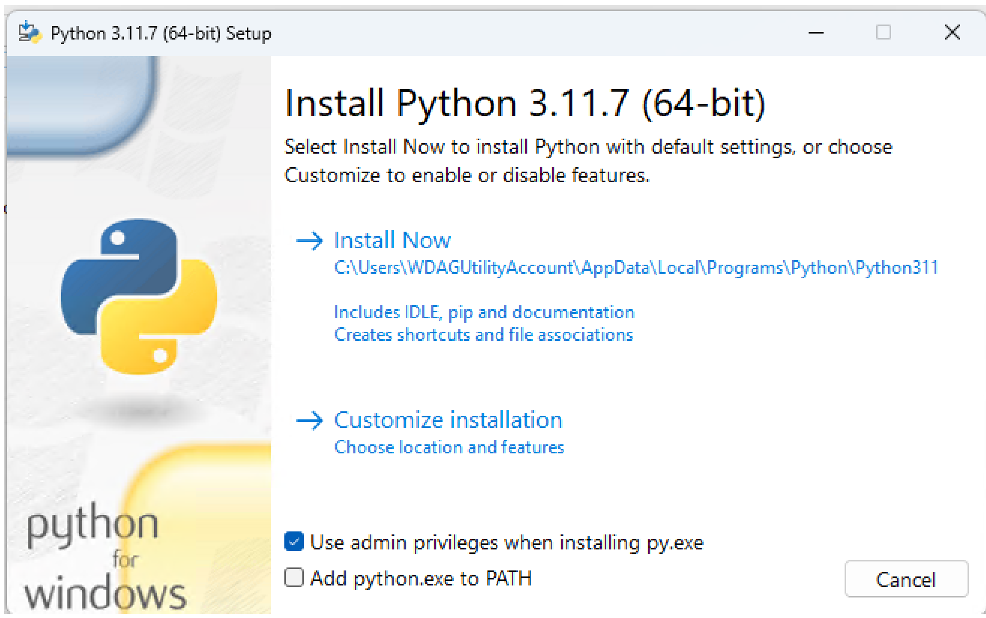
Step 1: Go to https://www.python.org/downloads/release/python-3117/ and then click on the button to download python 3.11.7.

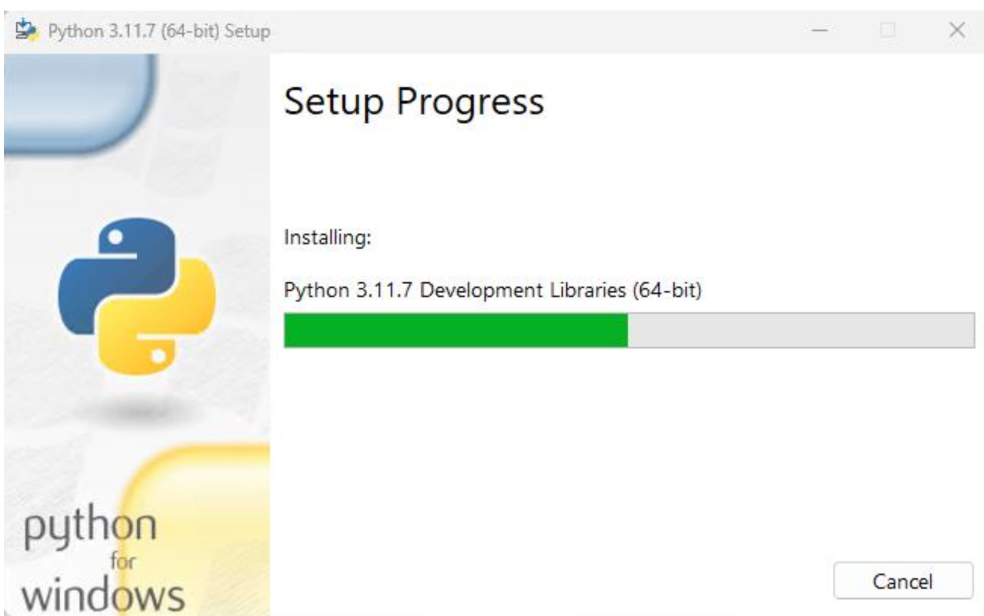


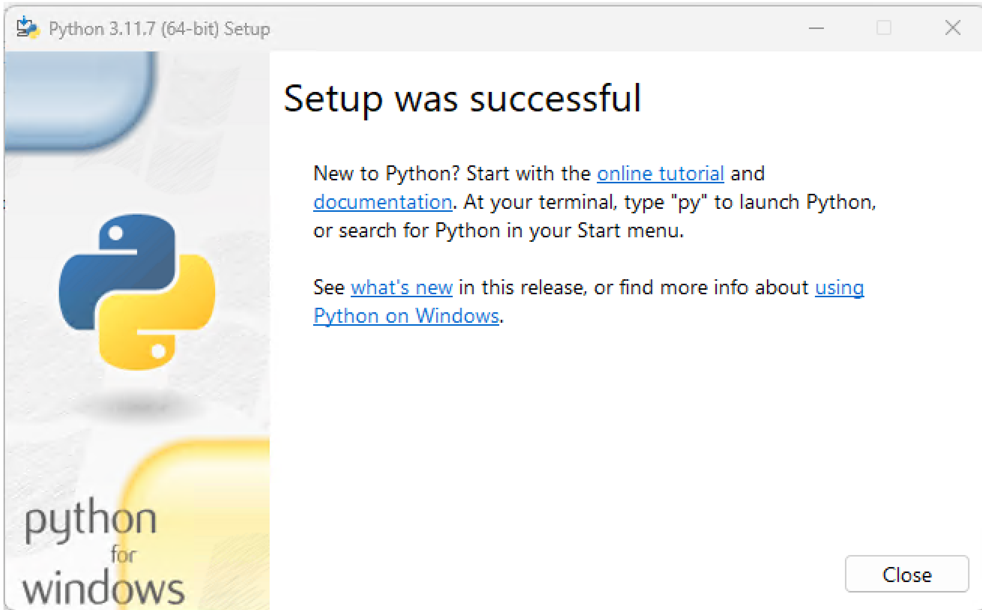
Step 2: Click on [Windows installer (64-bit)](#).



Step 3: Double-click the icon labeling the file python-3.11.7-a-amd64.exe.





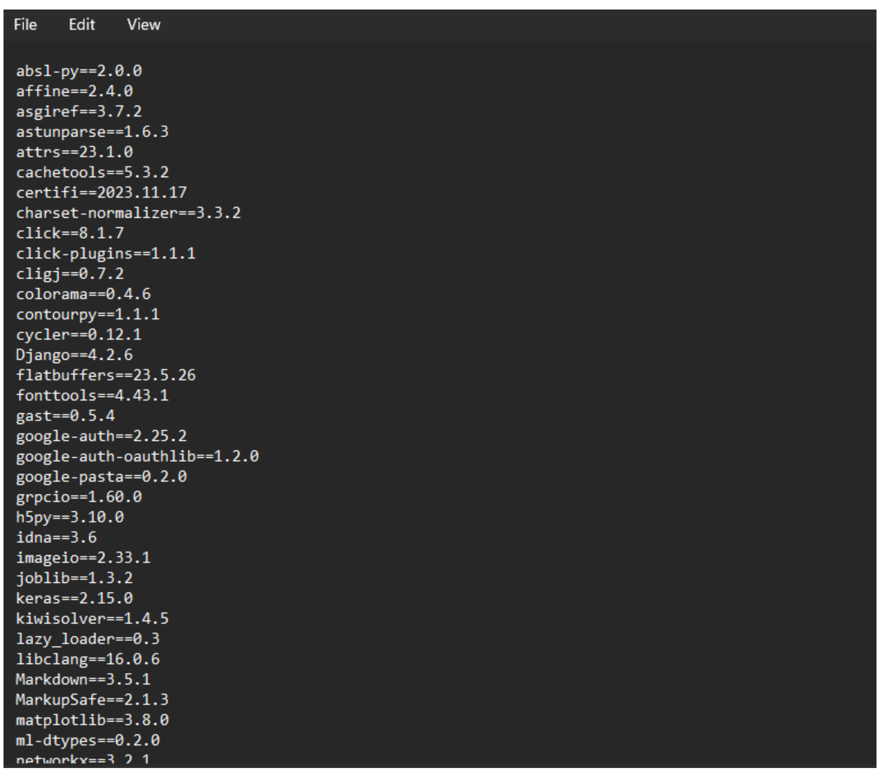


Step 4: After a short period of time, your setup will be completed, Click the Close button.

Congrats, you just installed Python on Windows!

Step 5: Now use python 3.11.7 to install these libraries using pip and pipwin command.

Step 6: Run → pip install -r requirements.



File Edit View

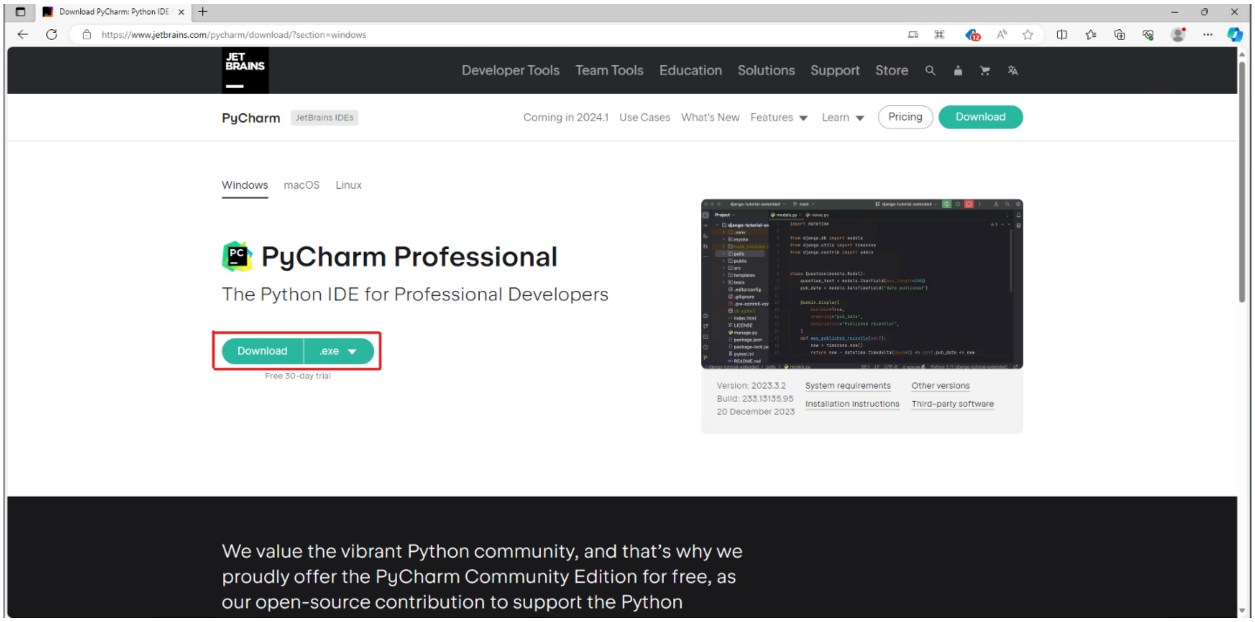
```
absl-py==2.0.0
affine==2.4.0
asgiref==3.7.2
astunparse==1.6.3
attrs==23.1.0
cachetools==5.3.2
certifi==2023.11.17
charset-normalizer==3.3.2
click==8.1.7
click-plugins==1.1.1
cligj==0.7.2
colorama==0.4.6
contourpy==1.1.1
cycler==0.12.1
Django==4.2.6
flatbuffers==23.5.26
fonttools==4.43.1
gast==0.5.4
google-auth==2.25.2
google-auth-oauthlib==1.2.0
google-pasta==0.2.0
grpcio==1.60.0
h5py==3.10.0
idna==3.6
imageio==2.33.1
joblib==1.3.2
keras==2.15.0
kiwisolver==1.4.5
lazy_loader==0.3
libclang==16.0.6
Markdown==3.5.1
MarkupSafe==2.1.3
matplotlib==3.8.0
ml-dtypes==0.2.0
networkx==3.2.1
```

Congratulations, now you can run The Project Environment.

11-2 PyCharm installation.

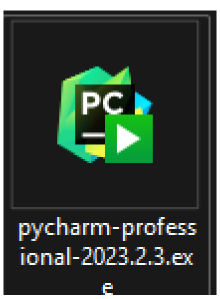
To install PyCharm on your windows computer you must follow these steps.

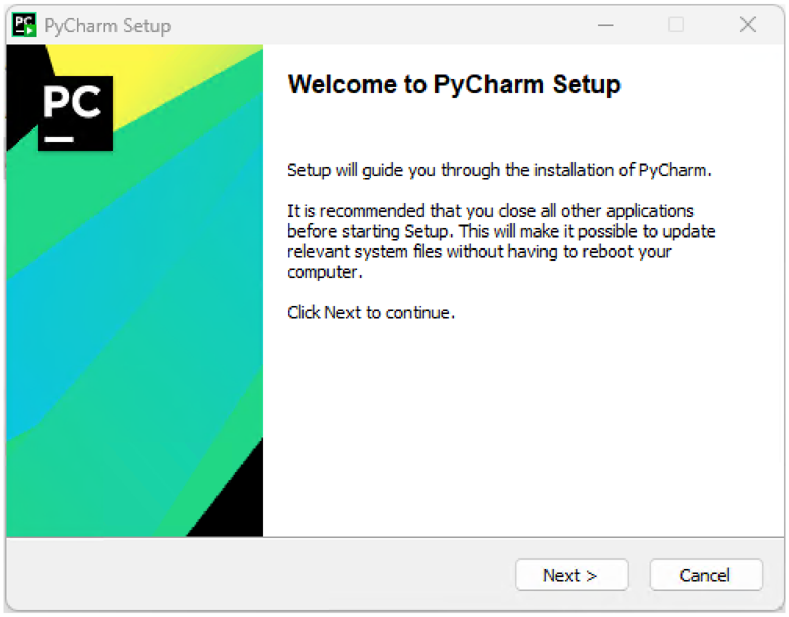
Step 1: Visit the site https://www.jetbrains.com/pycharm/download/?section=windows to download it.

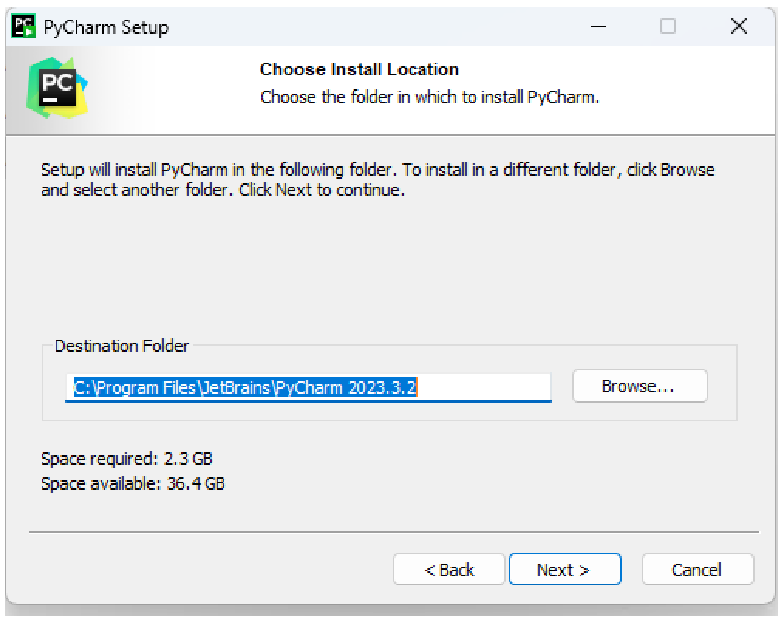


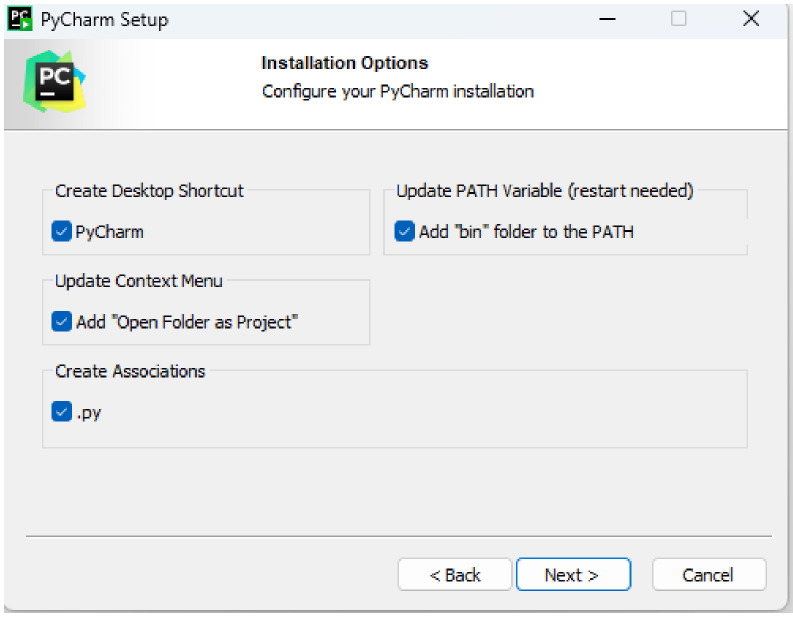
Step 2: When you click on the Download option, it will take you to the download page. You can download any package suitable for your computer, but for the sake of this tutorial, download the full version.

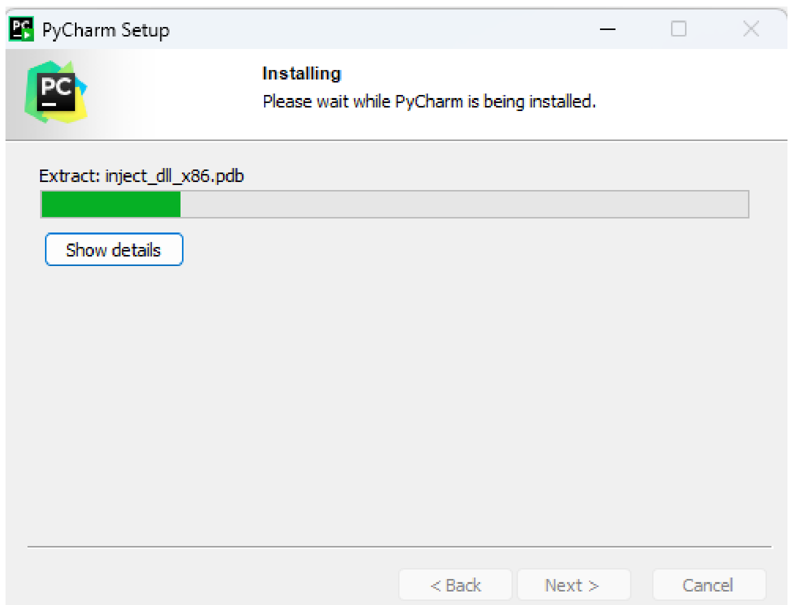
Step 3: Double-click the icon labeling the file pycharm-professional-2023.2.3.exe.




PyCharm Setup
PC
Welcome to PyCharm Setup
Setup will guide you through the installation of PyCharm.
It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.
Click Next to continue.
Next >
Cancel


PyCharm Setup
PC
Choose Install Location
Choose the folder in which to install PyCharm.
Setup will install PyCharm in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.
Destination Folder
C:\Program Files\JetBrains\PyCharm 2023.3.2
Browse...
Space required: 2.3 GB
Space available: 36.4 GB
< Back
Next >
Cancel

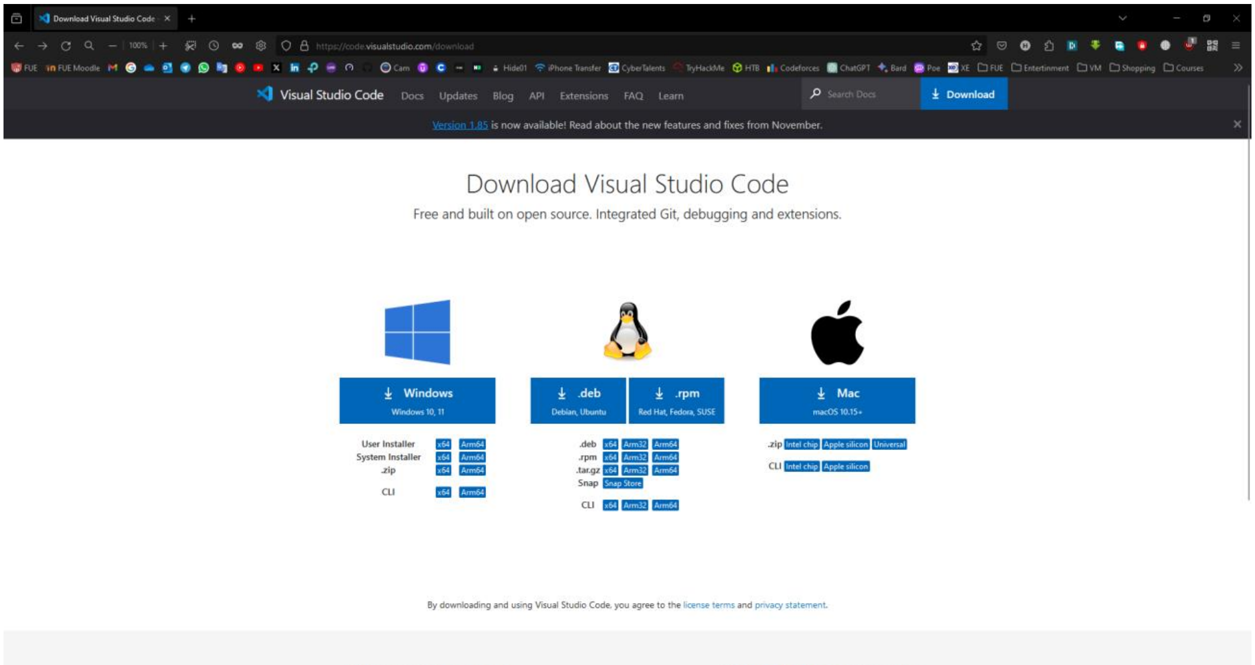




Now you have successfully installed NetBeans PyCharm on your computer.

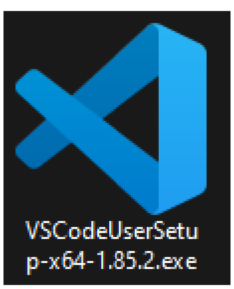
11-3 Visual Studio Code Installation.

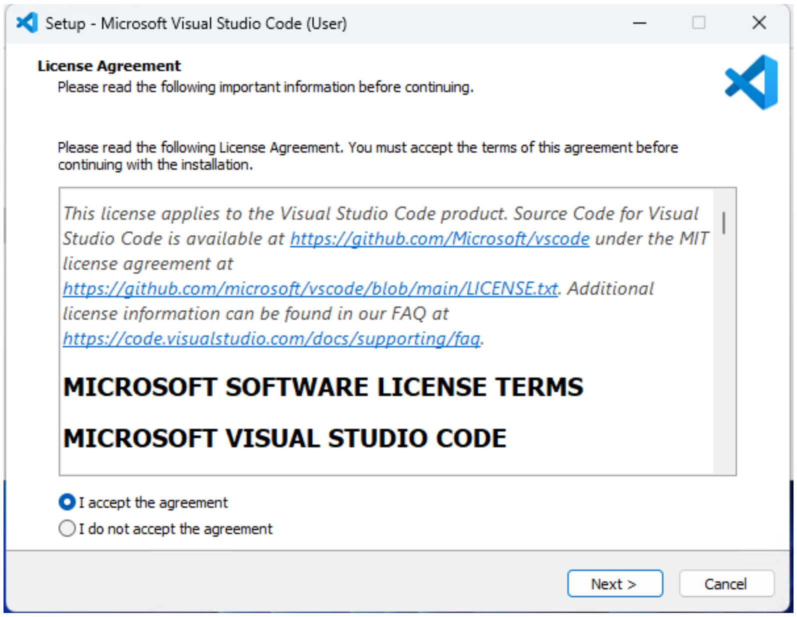
Step 1: Visit the site https://code.visualstudio.com/download to download it.

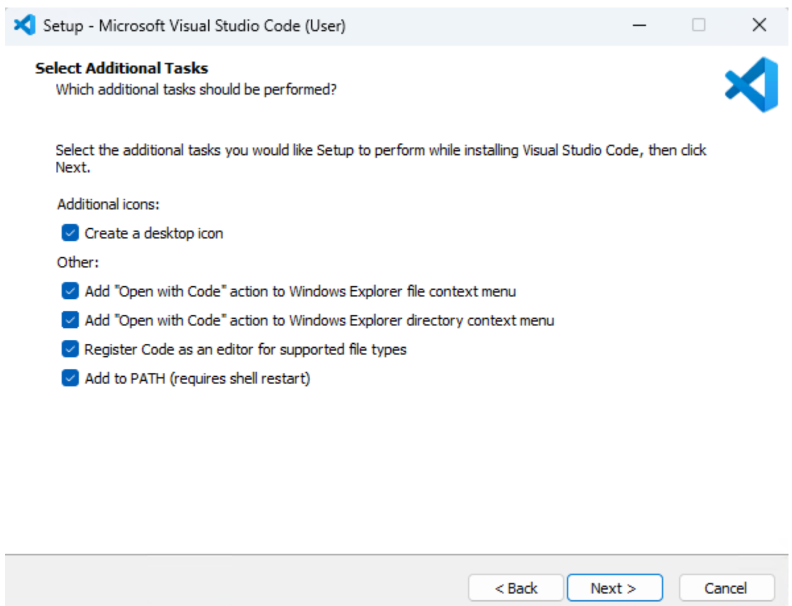


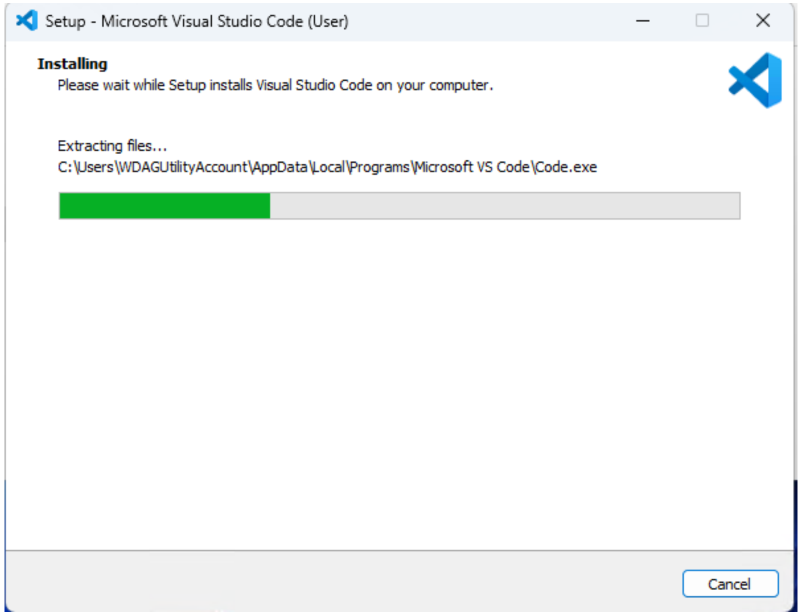
Step 2: When you click on the Windows option, it will take you to the download page. You can download any package suitable for your computer.

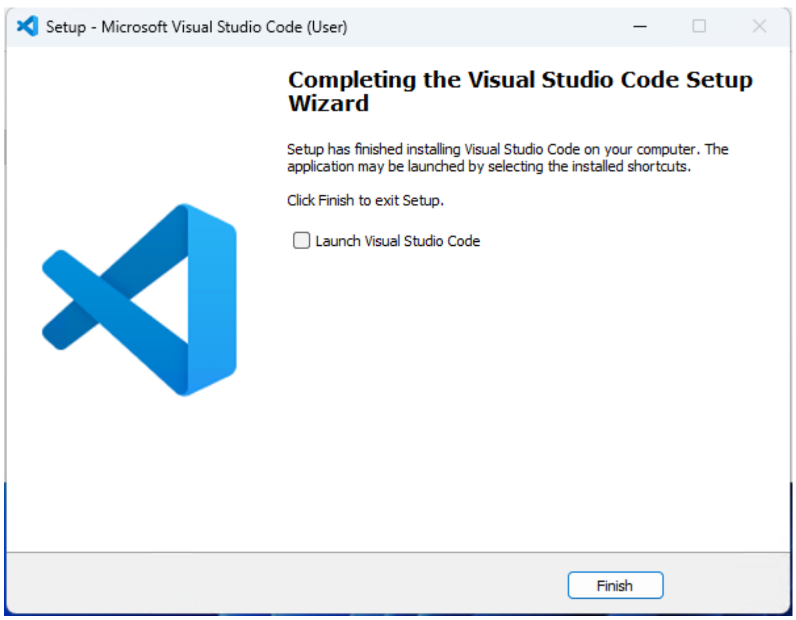
Step 3: Double-click the icon labeling the file VSCodeUserSetup-x64-1.85.2.exe.




Setup - Microsoft Visual Studio Code (User)
License Agreement
Please read the following important information before continuing.
Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.
This license applies to the Visual Studio Code product. Source Code for Visual Studio Code is available at https://github.com/Microsoft/vscode under the MIT license agreement at https://github.com/microsoft/vscode/blob/main/LICENSE.txt. Additional license information can be found in our FAQ at https://code.visualstudio.com/docs/supporting/faq.
MICROSOFT SOFTWARE LICENSE TERMS
MICROSOFT VISUAL STUDIO CODE
I accept the agreement
I do not accept the agreement
Next >
Cancel


Setup - Microsoft Visual Studio Code (User)
Select Additional Tasks
Which additional tasks should be performed?
Select the additional tasks you would like Setup to perform while installing Visual Studio Code, then click Next.
Additional icons:
Create a desktop icon
Other:
Add "Open with Code" action to Windows Explorer file context menu
Add "Open with Code" action to Windows Explorer directory context menu
Register Code as an editor for supported file types
Add to PATH (requires shell restart)
< Back
Next >
Cancel





Now you have successfully installed NetBeans Vs Code on your computer.