Detecting and Measuring Desolate Lands and Water Bodies from Satellite Images

A thesis
Submitted in partial fulfillment of the requirements for the Degree of
Bachelor of Science in Computer Science and Engineering

Submitted by

Md. Rafi	**190104041**
Jerin Ahasan Kheya	**190104043**
Asif Mamun Hridoy	**190104047**
Syeda Annan Asrafi	**190104050**

Supervised by

Prof. Dr. Md. Shahriar Mahbub

Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

Dhaka, Bangladesh

November 21, 2023

CANDIDATES' DECLARATION

We, hereby, declare that the thesispresented in this report is the outcome of the investigation performed by us under the supervision of Prof. Dr. Md. Shahriar Mahbub, Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh. The work was spread over two final year courses, CSE4100: Project and Thesis I and CSE4250: Project and Thesis II, in accordance with the course curriculum of the Department for the Bachelor of Science in Computer Science and Engineering program.

It is also declared that neither this thesis nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

Md. Rafi
190104041

Jerin Ahasan Kheya
190104043

Asif Mamun Hridoy
190104047

Syeda Annan Asrafi
190104050

CERTIFICATION

This thesis titled, **"Detecting and Measuring Desolate Lands and Water Bodies from Satellite Images"**, submitted by the group as mentioned below has been accepted as satisfactory in partial fulfillment of the requirements for the degree B.Sc. in Computer Science and Engineering in November 21, 2023.

Group Members:

Md. Rafi	**190104041**
Jerin Ahasan Kheya	**190104043**
Asif Mamun Hridoy	**190104047**
Syeda Annan Asrafi	**190104050**

Prof. Dr. Md. Shahriar Mahbub
Professor & Supervisor
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

Prof. Dr. Md. Shahriar Mahbub
Professor & Head
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

ACKNOWLEDGEMENT

At the very first, we would like to show our deepest gratitude to our supervisor Prof. Dr. Md. Shahriar Mahbub, Professor & Head, Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, who offered continuing support and constant encouragement, which empowered us to complete our thesis. His guidance, comments, suggestions, constructive criticism, enthusiastic and steady oversight all through our work were invaluable. We also thank all the faculty members of the Computer Science and Engineering department, Ahsanullah University of Science and Technology for helping us in many ways whenever needed. Finally, we would also like to express our gratitude to our beloved parents and family for their moral support and warm encouragement.

Dhaka
November 21, 2023

Md. Rafi

Jerin Ahasan Kheya

Asif Mamun Hridoy

Syeda Annan Asrafi

ABSTRACT

Locating and assessing desolate land and water body that is accessible in our nation is crucial to figuring out whether we have adequate resources to defy the impending famine. Due to the high availability of satellite images, it is easier to use satellite data to measure accessible land rather than undertaking human ground surveys. Due to the unavailability of high-resolution public satellite dataset of our country, we used medium-quality Apple Maps satellite images. We categorized the images into three classes (desolate land, water body and other) and masked the images manually to create our own dataset. Only locating these lands is not enough as measuring the area will give a more effective view of present land situation. Although there are various models and algorithms available that may be used to classify land portions, we employed deep learning U-Net architecture both with and without integrated Vision Transformer (ViT) to conduct semantic segmentation on our dataset. Both the U-Net and U-Net integrated with ViT perform better for smaller datasets. This is one of the reasons we used U-Net. Using U-Net architecture we achieved 88.1% accuracy (micro-average) with the test data. On the other hand, with U-Net with ViT we achieved 88.4% accuracy. Another popular metric for evaluating semantic segmentation is Jaccard Index, commonly known as the Intersection over Union (IoU). We attained IoU value (macro-average) of 0.697 and 0.704 for U-Net and U-Net with ViT respectively. Moreover, a website is implemented to demonstrate the the work more efficiently.

Contents

CANDIDATES' DECLARATION i

CERTIFICATION ii

ACKNOWLEDGEMENT iii

ABSTRACT iv

List of Figures vii

List of Tables viii

1 Introduction 1

2 State of the art 5

3 Dataset 10

4 Methodology 13

4.1 System Overview 13
4.2 Data Pre-processing 13
4.3 Introduction to U-Net 15
4.4 Introduction to ViT 17
4.5 Approach 1: U-Net 19
4.6 Approach 2: U-Net with ViT 21
4.7 Measuring the Detected Land 22
4.8 Project Management 24
4.9 Financial Expenditure 25

5 Result 26

5.1 Performance Metrics 26
5.2 Intersection over Union 27
5.3 Result Analysis 29
5.3.1 Approach 1: U-Net 29
5.3.2 Approach 2: U-Net with ViT 31

5.4 Comparision between Approach 1 and Approach 2 33
5.5 Prediction 34
5.6 Verdict 34
5.7 Demonstration of website 34

6 Future Work **39**

7 Conclusion **40**

References **42**

List of Figures

1.1 Acute Food Insecurity Situation for May - September 2023 2

3.1 Example of dataset 12

4.1 Flowchart of overall methodology 14
4.2 Example of Transformation 15
4.3 Flowchart of Data Pre-processing 16
4.4 U-Net Architecture 17
4.5 Vision transformer model overview 18
4.6 Illustration of image patches 18
4.7 Runway length measurement of DAC from- (i) Apple Maps, (ii) Google Maps 23
4.8 Process for measuring meters per pixel. (i) Original image downloaded from Apple Maps, (ii) Rotated image so that the length of the court is in vertical position, (iii) Count pixels of the court with using Adobe Illustrator 23
4.9 Sequence of processes in Gantt Chart 24

5.1 Original and mask image showing majority and minority classes 27
5.2 Visualizing IoU 28
5.3 Three different IoU values for three different cases 28
5.4 U-Net: Loss vs Epochs 29
5.5 U-Net: Accuracy vs Epochs 30
5.6 Confusion matrices of Approach 1 30
5.7 U-Net with ViT: Loss vs Epochs 31
5.8 U-Net with ViT: Accuracy vs Epochs 32
5.9 Confusion matrices of Approach 2 32
5.10 Model Prediction Examples (Approach 1). From left to right- original image, ground truth mask, predicted mask 35
5.11 Model Prediction Examples (Approach 2). From left to right- original image, ground truth mask, predicted mask 36
5.12 Homepage 37
5.13 Uploading image 37
5.14 Predicted output image 38

List of Tables

3.1 List of all three classes 11
3.2 Distribution of dataset 12

5.1 Performance metric of U-Net approach 31
5.2 Performance metric of U-Net with ViT approach 33
5.3 Class-wise comparison of metrics between Approach 1 and Approach 2 33
5.4 Overall comparison based on micro-average 34
5.5 Overall comparison based on macro-average 34

Chapter 1

Introduction

The recent conflict, economic shocks, climate extremes, and rising fertilizer prices create an enormous global hunger and malnutrition crisis, with more than 345 million people facing high levels of food insecurity in 2023 – more than double the number in 2020 [1]. Almost 24 percent of the analysed population from different parts of Bangladesh experienced high risk of acute food insecurity which is classified in IPC Phase 3 and 4 (IPC Phase 3: Acute Food and Livelihood Crisis, IPC Phase 4: Humanitarian Emergency) between harvest season where estimated 7.9 million people are in IPC Phase 3 and 1 million people are in IPC Phase 4. People under Phase 3 or higher is likely to increase 8 percent from the current period if no further actions are taken within time [2]. The acute food insecurity situation for May to September 2023 for Bangladesh is shown in Figure 1.1. Not only land but water body also provides a great deal in our food resource as to increase global food supplies on a sustainable basis water is considered as vital requirement for farmers [3]. Not to mention that fish are produced in the naturally-flowing water without disrupting the volumes or ultimate destinations of spring flows [4]. To meet the crisis, our prime minister advised the country's people to use every inch of land for food production [5]. To ascertain whether we have adequate resources to withstand the crisis, it is vital to locate and measure desolate lands in our country that can be used for food production.

The government has a statistical record of cultivated land and rivers but does not have a record of desolate land or any small water bodies. A field survey could be carried out to gather data about arid lands and water sources. However, these surveys are time-consuming, expensive, and heavily labor-intensive. Today satellite data is widely available to the public without any cost. The NASA/USGS Landsat Program is such a program. The program has satellite data of the whole of Bangladesh. The program also delivers complete coverage of the Earth's surface every eight days [6]. Thus, current data is also accessible. There are many more services that offer satellite data to the public like- SPOT, Google Earth, Apple Maps, etc. With the abundance of satellite data, conducting such surveys by hand, which

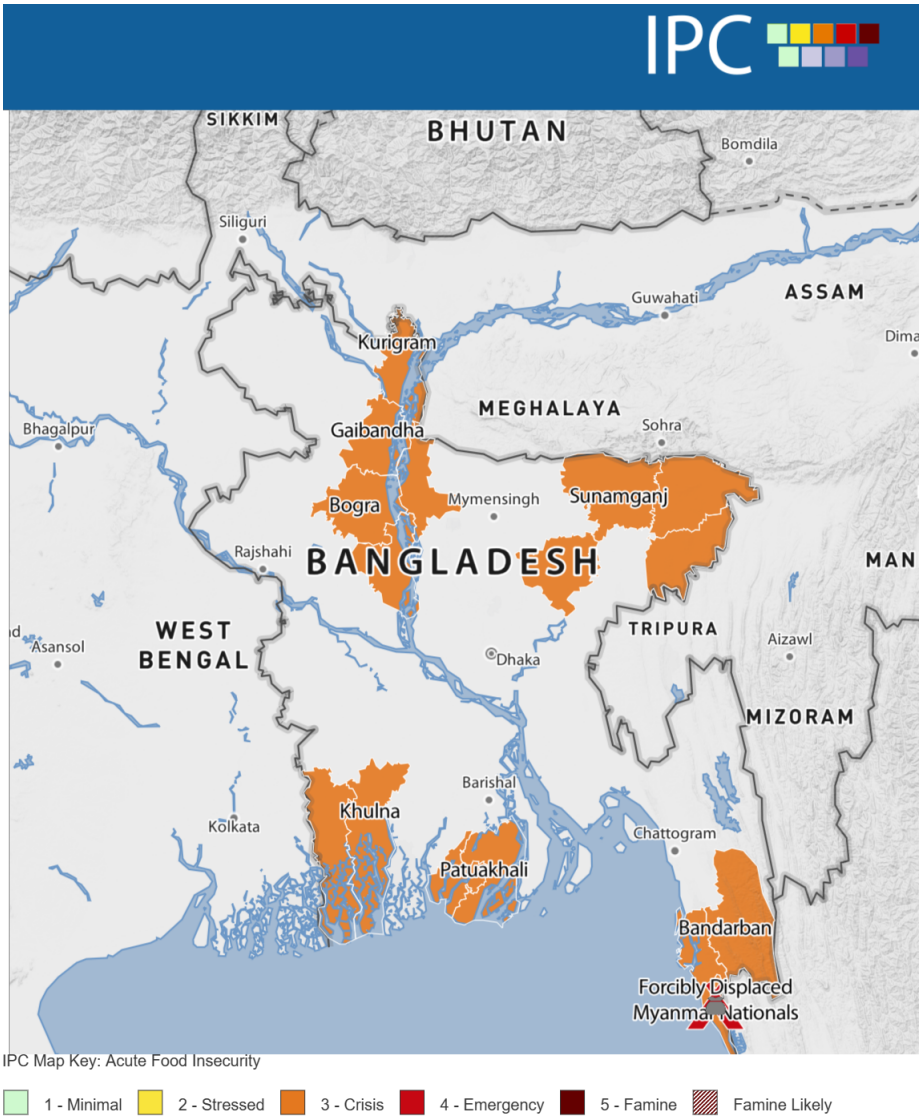


Figure 1.1: Acute Food Insecurity Situation for May - September 2023

entails a lot of labor-intensive labor, is not an effective strategy. Instead, the surveys can be completed automatically, considerably more quickly, and inexpensively using satellite data combined with different machine learning approaches.

The application of satellite data has increased enormously in recent years due to its high availability. Satellite imagery and data products collected from satellite instruments integrate the datasets used by experts and researchers for modeling decision support systems related to the environmental and agricultural domains. These systems are meant to offer specialized, high-value applications and services, such as anticipating crop diseases or yield production [7].

Our objective is to detect and measure desolate land and water bodies from satellite images. The key difficulties are automatic land and water delimitation, classifying and identifying the detected land and water bodies, and calculating its extent. Researchers have extensively investigated automatic land delimitation and have developed a number of methodologies and algorithms, including segmentation algorithms [8], clustering algorithms [7,9,10], and fuzzy algorithms [11].

Despite the abundance of satellite image datasets, high-resolution datasets are hard to get and in most cases, not available for our nation (e.g., SPOT-5, SPOT-6, etc.). Furthermore, it is quite rare to find datasets annotated with pixel-by-pixel for semantic segmentation purpose. A certain amount of annotated dataset is available for Dubai [12] but there is no such dataset available for our country. Therefore, we assembled satellite pictures from Apple Maps and manually annotated them to create our own dataset by masking within three classes (desolate lands, water bodies and other). Chapter 3 of this book contains further information about our dataset.

Although many research has been conducted on different types of methodologies for this similar kind, we used the U-Net architecture [13] with and without Vision Transformer [14] for the semantic segmentation of satellite images. Semantic segmentation of satellite images is the ability to segment an unknown satellite image and classify these segments into various categories (e.g., cropland, road, water, houses, etc.). After testing, we obtained a overall accuracy (micro-average) of 88.1% and 88.4% for using only U-Net architecture and using U-Net combined with Vision Transformer respectively for the same test data. Another popular metric for segmentation task is Jaccard Index [15, 16], also know as, Intersection over Union (IoU). We were able to achieve IoU value (macro-average) of 0.697 and 0.704 using only U-Net architecture and combined U-Net with Vision Transformer respectively. More about the results can be found in this book's Chapter 5.

Both the U-Net and modified U-Net with integrated Vision Transformer architecture achieved great success in fulfilling our objective of detecting and measuring desolate lands and water bodies. If we can increase our dataset including data from all around the world and explore

more state-of-the-art models, surely the models will be able to detect more accurately. We constructed a website to demonstrate our work more clearly and if the website is published in public domain, everyone around the world will be able to access it.

This book is structured as follows- state-of-the-art or related works are in the next chapter, description of the dataset is the chapter after that, then the following chapters are methodology, result, future works and conclusion.

Chapter 2

State of the art

Image segmentation has been one of the most challenging problems in computer vision for the past three decades [17]. Numerous works have been done on image segmentation. Researchers trying to optimize the cost and performance of the available segmentation method. U-Net is one of the most popular image segmentation architectures since 2015.

We have studied the works of many researchers and scholars about different methods of extracting different parts of images, like- segmentation, clustering, classification, etc. Summary of some of the papers is given below:

Arango et al. [7] use a partition clustering algorithm called Partitioning Around Medoids to perform automatic delimitation of cultivable land. They took advantage of the multispectral bands of remote sensing satellite images. While satellite images provide three channels R,G,B values for a place or a coordinate, multi-spectral bands provide more information like IR, temperature, reflections, UV, etc. This extra band information is particularly useful when confusion arises to categorize a part of the land. By using a combination of different bands the area can be easily identified. Arango et al. [7] performed clustering for different bands and compared the results. They used Landsat-8 satellite data of three different parts of Spain. They evaluate the performance of clustering by using a metric called quality index. Their work is to identify cultivable and non-cultivable land. Although they need two clusters for this problem, they take different numbers of clusters and ultimately merge all cultivable clusters into one cluster and all non-cultivable clusters into another cluster. This technique allowed them to compare the change of results for different combinations of bands and clusters. Their study found that initializing the number of clusters, $K = 2$ gives better results for each band. Increasing the number of clusters often reduces the quality index value. However, selecting the right band is more tricky. For different places, the same band gives different results.

In satellite images, pixel classification can be quite challenging because the test set's im-

ages might have different lighting conditions or even represent different regions from the training set's images. Ohleyer et al. [18] suggested a different kind of dataset with several geographic regions for the training set and the test set employing a more contemporary network architecture, the Mask R-CNN, and compared its results to those of two more traditional networks, existing Caffe implementation of the Fully Convolutional Network and its Multi-Layer Perceptron Extension, to determine how well it performed. In some low-training-time locations, this method performs better than baselines, but not in all of them. Two components make up their strategy. They begin by attempting to duplicate previously obtained baseline results. Then, they aspire to study a new Convolutional Neural Network (CNN) architecture that has proved its efficiency in instance segmentation tasks: the Mask R-CNN (Mask Region-based CNN). To be noted that an adequate dataset for working with satellite photos was provided by Inria, comprising training and test images covering several geographic regions. Additionally, utilizing a Fully Convolutional Network (FCN) and its extension with a Multi-Layer Perceptron (MLP), they provide some segmentation results. Two assessment metrics are taken into account to quantify performances: the accuracy, which is determined by the percentage of correctly classified pixels, and the intersection over union (IoU) of the expected class. On average their approach outperforms FCN and MLP to predict well on low dense cities. On the contrary, MLP predicts better than Mask R-CNN on very dense areas. They concluded on the mask R-CNN is a fascinating and promising network design for semantic segmentation that even enables instance segmentation on satellite photos. However, it takes more practice and improved hyperparameter tuning than they were able to provide.

Guérin et al. [19] present a method for automatically segmenting satellite images into six classes using the Swin Transformer architecture. The dataset used was created from IGN open data. The six classes include sparse forest, dense forest, moor, herbaceous formation, building, and road. The paper reports both quantitative and qualitative segmentation results on the dataset and discusses the strengths and limitations of the proposed method. The authors have made the dataset and trained model publicly available. They use Swin Transformer, a computer vision backbone that has shown to be efficient and state-of-the-art for tasks such as image classification, object detection, and semantic segmentation. The architecture of the Swin Transformer is based on Shifted WINdows, which makes it robust against scale variability while maintaining linear efficiency. The Shifted Windows concept involves shifting the window by half of its size to limit the self-attention computation to non-overlapping local windows while maintaining cross-window connections. They use an implementation based on mmsegmentation for its experiments. Lastly, they discussed the results of the EC-JRC project on "Crop area estimation with satellite images in Ukraine," which was carried out by the Ukrainian Space Research Institute of NASUNSAU in collaboration with the MARS team of the JRC. The best results were achieved using MLP classifiers and TM

images, which proved to be a cost-efficient way to improve estimators. MODIS images were also found to be cost-efficient. The study provided useful insights in the Geoland2-Crop CIS framework and updated the conclusions of the MARS project in the mid-90s, which had shown technical solidity but failed to reach the cost-efficiency threshold with the costs of 15 years ago.

Kussul et al. [20] discusses the use of satellite data and machine learning techniques for crop mapping and estimation of agricultural areas. The study was conducted in Ukraine, and the authors compared the performance of three different satellite datasets (MODIS, Landsat-5, and AWiFS) in classifying different crop types and estimating their areas. The authors used machine learning algorithms such as Multilayer Perceptron (MLP), Support Vector Machine (SVM), and Decision Tree (DT) to classify crop types based on satellite data. They found that MLP performed better than SVM and DT in terms of classification accuracy on the test data, showing better generalization ability. The MLP classifier on Landsat-5 data showed slightly better results compared to other satellite data in terms of classification accuracy for test datasets. The authors also used regression analysis to estimate crop areas, and they found that using satellite data as a covariate improved the efficiency of area estimation for major crops such as wheat, barley, maize, and soybeans. The study provides insights into the potential of using satellite data and machine learning algorithms for crop mapping and estimation of agricultural areas.

Mai et al. [21] discusses the challenges of segmenting synthetic aperture radar (SAR) images in remote sensing applications due to the influence of speckle noise. The authors propose a new Hierarchical Iterative Clustering Algorithm using Spatial and Spectral information (HICLASS) to increase the accuracy of image clustering. They compare the proposed algorithm with other methods such as FCM, SC, and ISC and evaluate the performance of the algorithms on the basis of several validity indexes such as MSE, IQI, DI, and CSI. The results show that the proposed algorithm outperforms the other methods in terms of noise reduction and oil stain classification. This paper also provides a review of other clustering techniques based on spectral clustering and their limitations. The authors conclude that the proposed algorithm is effective in segmenting SAR images and can be applied to other remote sensing applications.

Awad et al. [22] describes a new method for segmenting satellite images using an unsupervised artificial neural network called Kohonen's self-organizing map (SOM) and a threshold technique called T-Cluster. The SOM is used to organize pixels according to grey level values of multiple bands into groups, and then the T-Cluster is used to cluster the image into disjoint regions. The new method is called TSOM. The paper provides an overview of SOM and how it is implemented as part of the new segmentation method. It also presents the threshold clustering technique (T-Cluster) and explains the TSOM segmentation method. The paper covers the ISODATA algorithm and is devoted to experimental results. Finally,

the paper gives the conclusions. The stability, homogeneity, and efficiency (speed-wise) of the TSOM method are confirmed through experiments performed on two different satellite images. The stability and homogeneity of both methods are determined using a procedure selected from the functional model.

Xu et al. [23] present an innovative methodology aimed at identifying and classifying urban vacant land for sustainable city planning. Introducing a comprehensive typology that categorizes vacant land into four distinct classes—transportation-associated land, natural sites with unfavorable conditions, unattended areas or reserve parcels, and brownfield sites. The study integrates remote sensing images, GIS layers, and citizen science data within a data fusion framework. This approach addresses various challenges, including inconsistent terminology, inefficient registration systems, and high costs of data collection, which hindered prior efforts in mapping vacant land. The utilization of remote sensing images proves pivotal, enabling the identification of characteristic patterns associated with vacant land types. Instead of using the usual way of sorting images, the authors went for a different method called a data fusion framework. This new way is s available for different towns and cities to use freely and accessibly. This innovative approach not only enhances the accuracy of identifying vacant but it also encourages open and fair city planning, supporting sustainable development strategies in cities.

Wurm et al. [24] has conducted a research which is particularly relevant given the challenges posed by rapid urbanization, especially in the global south,and the need to address issues related to informal urban development and poverty. They want to provide better infrastructure and improving the quality of life for slum dwellers and this is the key goal and try to analyze the transfer learning capabilities of FCNs in mapping slums across various satellite images. In their work, they focuses on detecting cities through satellite imagery and using fully convolutional networks (FCNs) and transfer learning. Also, they use very high-resolution optical satellite imagery from QuickBird and other sources like Sentinel-2 and TerraSAR-X. They get Sentinel-2 data is free-of-charge but has lower resolution, posing challenges for slum mapping and TerraSAR-X data has higher resolution and is considered powerful for intra-urban structure analysis. But they faces so many problems when they try to transferring the model from optical to SAR data, emphasizing differences in image characteristics. QuickBird optical image achieves 86–88% positive prediction value and sensitivity for mapped slum. Sentinel-2 data shows a huge change in performance with transfer learning (from 38 to 55% for positive prediction value and from 79 to 85% for sensitivity). Lastly, transfer learning proves valuable in collecting information on small-scaled urban structures like slum patches, even in satellite images of lower resolution.

Bartalev et al. [25] introduces an innovative method that uses multi-year MODIS-derived phenological metrics to map the arable land across Russia. During 2005 to 2013, this technique created high-resolution annual maps, ensuring consistency and comparability

through automated data processing. Importantly, this method effectively significant changes in arable areas, supporting the trends observed in official agricultural statistics. When compared to reference data, the evaluation revealed an accuracy of over 80%, with the most productive regions achieving scores above 80%. However, differences in specific areas indicated potential issues with the training sets, emphasizing the need to fix these issues to improve accuracy. The research highlights the importance of fixing training set issues to better identify farming areas, highlighting how this innovative method could significantly change how we map agriculture in Russia.

Chapter 3

Dataset

In case of image segmentation, datasets must be annotated throughout to their pixel-level. The purpose of this is to get pixel-level classified or masked images which are manually labeled and segment various objects or regions within an image correctly. Most of the time, these kind of datasets are not found ready-made. Sometimes even if the dataset exists, it might not be of individual's interest. That's why datasets are made manually and requires substantial manual effort as annotators need to carefully outline and label the data and lastly these annotations are used as ground truth data for training and evaluating.

Masked images showing various classes of different segments are also necessary for supervised segmentation problems. There aren't many datasets of this kind available. Therefore, we developed our own dataset.

Segmentation problems require high-resolution data to work perfectly. However, accessing high-resolution data is not always possible. Furthermore, Bangladesh's very high-resolution satellite data are not available in any public dataset. After comparing the quality of the two most popular public satellite image datasets- Google Maps and Apple Maps, we noticed that Apple Maps' images have higher resolution than Google Maps.

We downloaded all the images used in our dataset from Apple Maps. By using Doodler, Adobe Illustrator and Paint all the downloaded images were masked according to their corresponding unmasked image where the required elements from those images are outlined carefully by their starting and ending point. Different colors are used to distinguish between elements within the images while masking. Two of our group members handled masking and another member validated the masked image manually by comparing both masked and unmasked images. During this process, a little amount of our data was removed from the overall dataset.

We segmented the images into three classes- desolate land, water and other where in the masked form the colors #3366CC (51, 102, 204), #DC3912 (220, 57, 18) and #FF9900

(255, 153, 0) represents desolate land, water and other respectively. Lands that are uncultivated and empty area, no matter the size are labeled as desolate lands. The small size and irregular shape of courtyard, lawns of houses etc. made the masking challenging. Paddy fields, cornfields, and other similar areas, any buildings or tin-shed homes, trees are labeled as other. Tin-shed houses are notoriously difficult to distinguish from adjacent portions because sunshine reflecting off the tin roof ruins the color of the pixels outside the roof border. As we collected images from rural areas of our country, ponds and other water bodies were very common to see. Any kind of water body were labeled as water. All other components that don't fit into desolate land or water categories are labeled as other. Some areas are very hard to identify because the low resolution makes the image fuzzy. Even we humans struggle to identify these regions, therefore these regions are also classified as other. Thus we collected dataset from both rural and urban areas of our country.

In the mask image, all three classes are annotated using three distinct colors. Each color stands for a single class and is distinct from the others. Table 3.1 below provides these colors' values.

Table 3.1: List of all three classes

Class	**Hex Code**	**Short Description**
Desolate land	#3366CC	Lands that are not cultivated yet; empty land
Water	#DC3912	Ponds, rivers, canals, etc.
Other	#FF9900	Everything else.

At this time, our original dataset contains 1552 images of 256×256 pixels each. Out of these 1552 images, 776 images are original satellite images collected from Apple Maps, and the other 776 images are their corresponding masked images. The dataset is doubled by applying data augmentation techniques. So, after augmentation, the dataset contains 1552 satellite images and 1552 corresponding masked images. Within these 1552 masked and unmasked images, 1088 images are used for training (which is 70% of overall dataset) and from the remaining 464 images (30% of overall dataset), half of the images which is 232 images (15% of overall dataset) are used for validation and other half are used for testing. All of these images are RGB color images and contains both urban and rural locations as well as different water bodies. Our dataset is released to the public so that others can use it for their projects. The dataset is available at www.kaggle.com/datasets/seralchemist/semantic-segmentation-bd-v3.

Distribution of dataset into train, test and validation sets are shown in Table 3.2.

Examples of the dataset are shown in the Figure 3.1

Table 3.2: Distribution of dataset

	Train	Validation	Test	Total
Proportion	70%	15%	15%	100%
Data	1088	232	232	1552

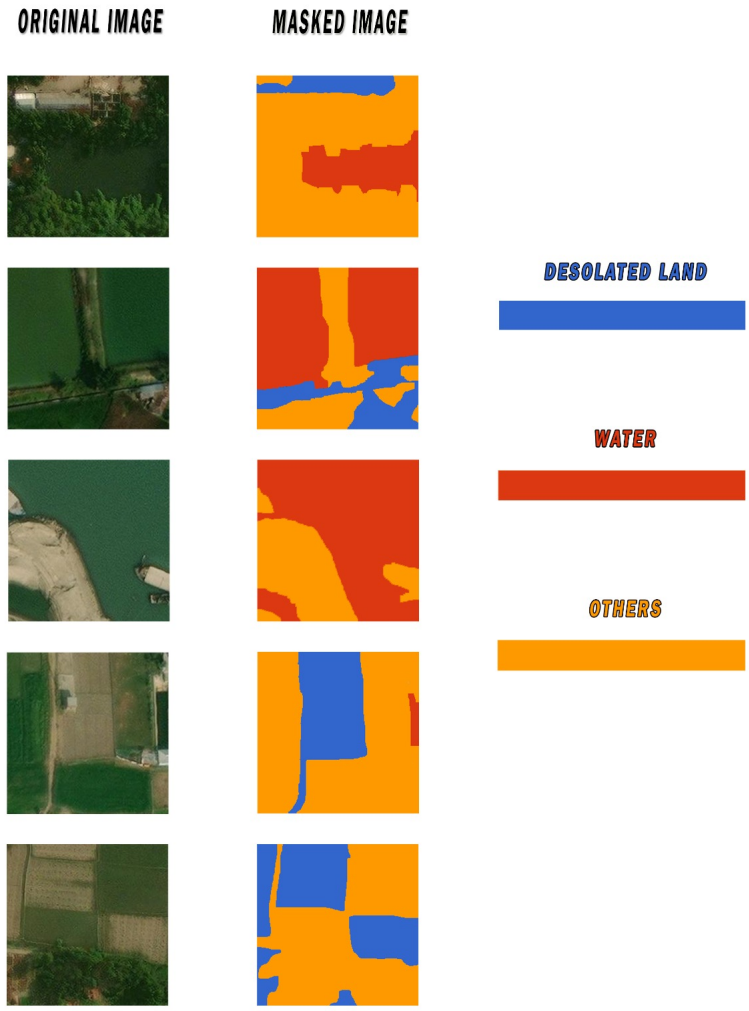


Figure 3.1: Example of dataset

Chapter 4

Methodology

4.1 System Overview

A primer on the current approach is provided here. Flowchart of overall methodology is shown in Figure 4.1.

Preparation and the process of manual labeling of the data are described in Chapter 3. Data pre-processing is described in the Data Pre-processing section, an introduction to U-Net is given in Introduction to U-Net section, introduction to vision transformer (ViT) is given in Introduction to ViT section, and the process of classifying a pixel by using only U-Net and by using U-Net with ViT is given in Approach 1: U-Net and Approach 2: U-Net with ViT section respectively. The following sections are Measuring the Detected Land, Project Management and Financial Expenditure.

4.2 Data Pre-processing

The input data is mainly satellite images and their corresponding mask images. After reading all the images, the pixel values of masks images is changed according to their classes. For example, all the pixels with value [51, 102, 204], [220, 57, 18], and [225, 153, 0] is changed into 0, 1, and 2 respectively. These process is also known as label encoding. The original images are normalized in the range between 0 and 1, and the masks images are converted into tensors.

The next step is transformation. This step is implemented to introduce more variation to the dataset so that the model can recognize and hence improve performance for test data. Transformation functions used in this work are:

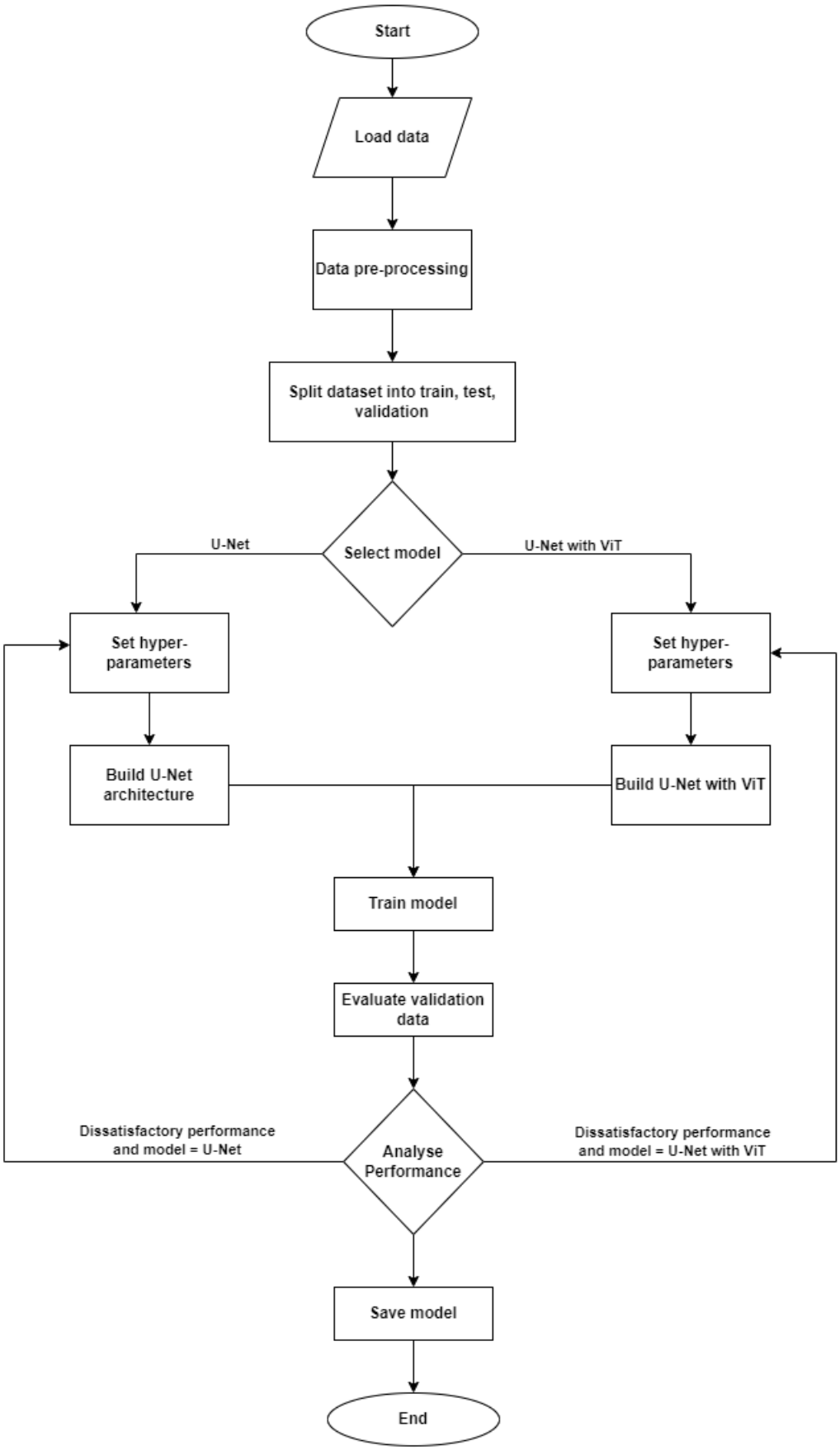


Figure 4.1: Flowchart of overall methodology

1. **Rotation:** Roughly 25% data with their masks are selected for rotation. Rotation is performed in 90°, 180° or 270° randomly.

2. **Vertical Flip:** Roughly 12.5% data with their masks are selected for vertical flip.

3. **Horizontal Flip:** Roughly 12.5% data with their masks are selected for horizontal flip.

4. **Color Jitter:** Only the original images go through random color adjustments. The brightness, contrast and hue of the images is adjusted by 10%. This adjustment is random. For example, the image will be 10% brighter or darker. 50% of the data goes through this step.

Example of performed transformation is shown in Figure 4.2.

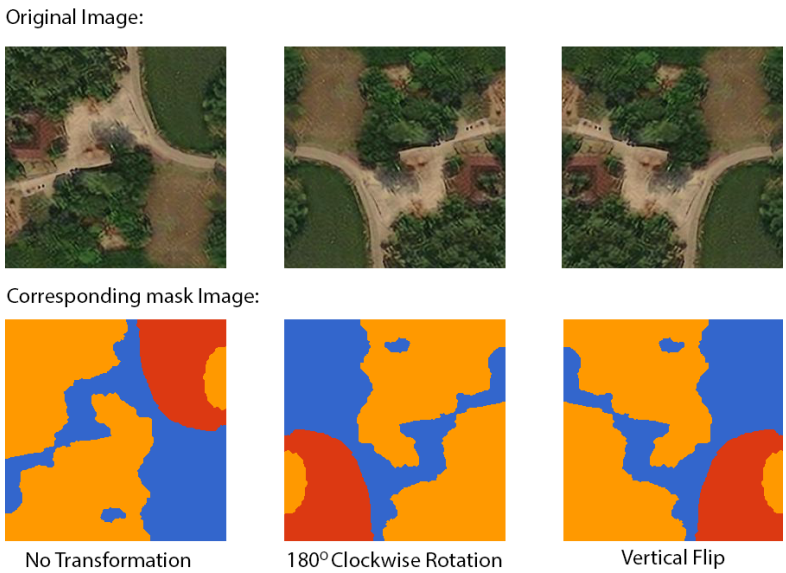


Figure 4.2: Example of Transformation

Flowchart of data preprocessing step is shown in Figure 4.3.

4.3 Introduction to U-Net

U-Net architecture is a fully convolutional neural network that was developed for biomedical image segmentation at the Computer Science Department of the University of Freiburg [13]. Though the model is initially proposed for the segmentation of medical images, its use is widely spread from object segmentation to satellite data segmentation.

What makes the U-Net unique is its ability to work with relatively small amounts of annotated data [26]. It excels in scenarios where there's a limited dataset available for training.

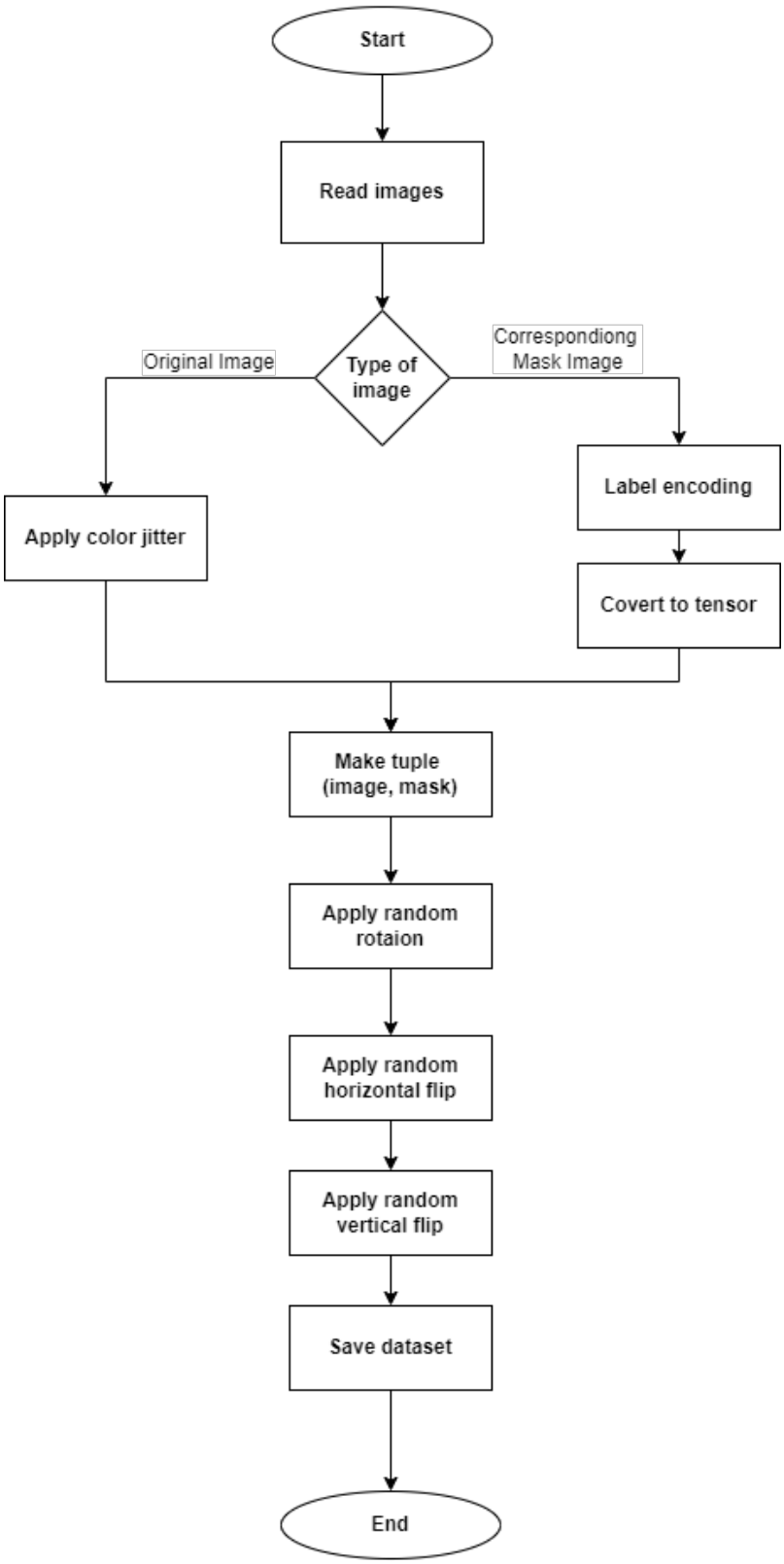


Figure 4.3: Flowchart of Data Pre-processing

Taking consideration of our dataset which is not so small and not so big, we chose to use U-Net architecture.

The model gets its name from its U-shape architecture. The left part of the U-shape architecture is the encoder part and the right side is the decoder part [27]. The encoder part extracts relevant features from image and the decoder part builds segmentation mask from features. The architecture of the model is shown in Figure 4.4 [13].

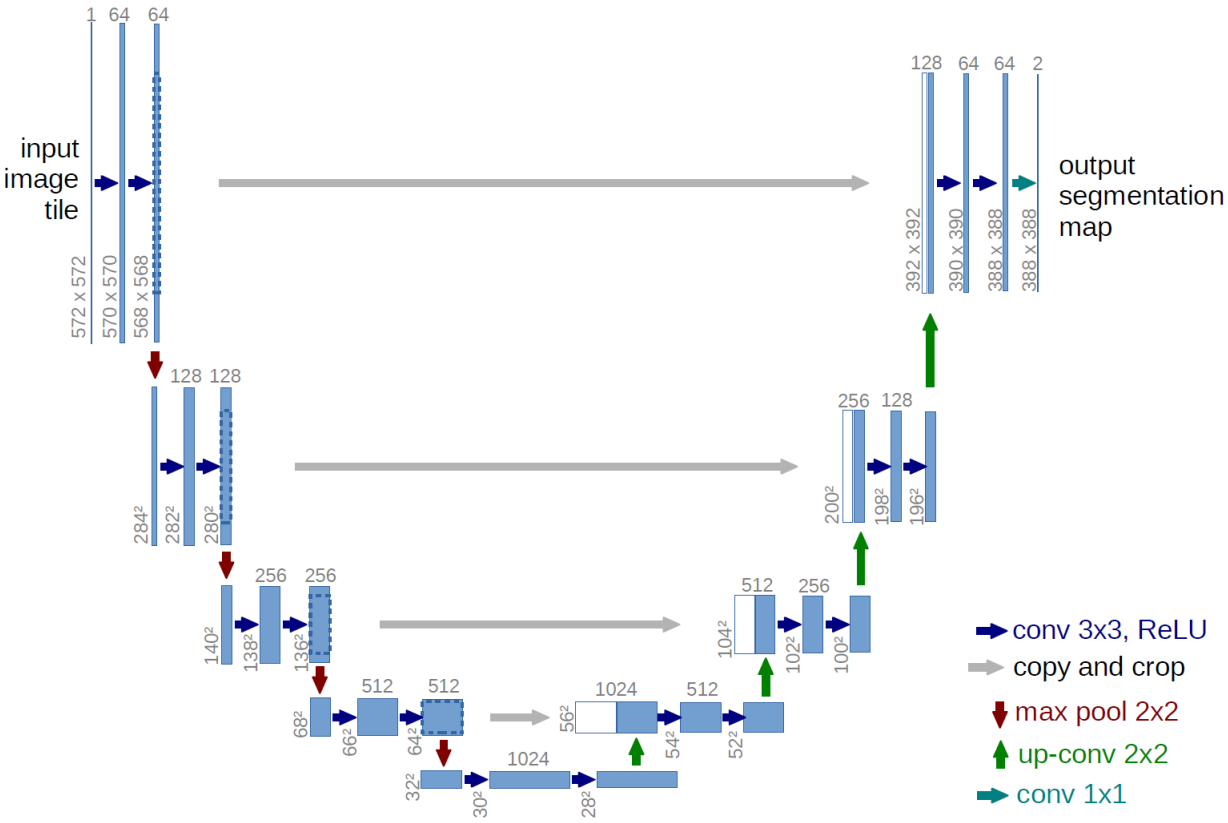


Figure 4.4: U-Net Architecture

U-Net take raw images as input and gives segmented images as output. U-Net consists of a large number of different operations illustrated by small arrows. The input image is fed into the network at the beginning. Then the data is propagated in the network along all the paths and the segmentation map comes out at the end.

4.4 Introduction to ViT

Vision transformer was introduced by Dosovitskiy et al. [14] in 2020. This is inspired by and similar to the transformer model proposed by Vaswani et al. [28], but instead of working with text data, vision transformer deals with images. The key difference between the traditional transformer and the vision transformer is that vision transformer does not have a decoder. It has transformer encoder and instead of decoder there is extra linear layer for

final classification called Multi-Layer Perceptron Head or MLP head for short. The model overview of the original vision transformer is shown in Figure 4.5 [14].

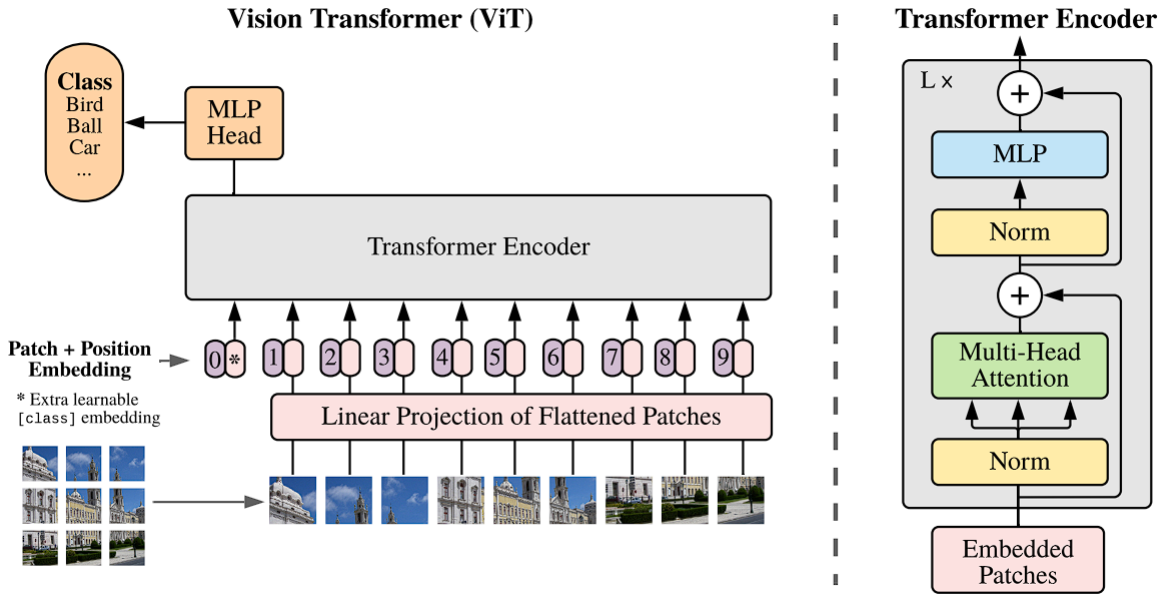


Figure 4.5: Vision transformer model overview

Vision transformer tries to represent images like words. Like word embedding, it generates embedding for images. But instead of generating embeddings for the whole image, it divided the image into multiple smaller patch images, and then generate embeddings for each individual patch image. For example, consider a image of 256 by 256 pixels. If we select patch size as 64 by 64 pixels, then it will generate 16 images each of size 64 by 64 pixels. An illustration of this is shown in Figure 4.6.

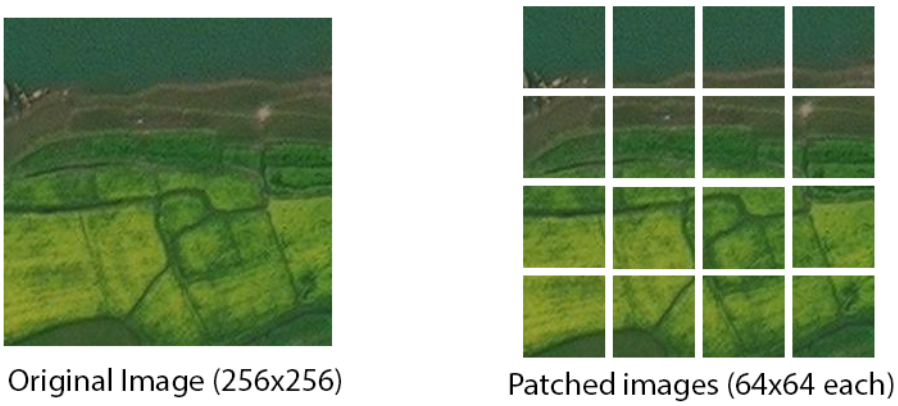


Figure 4.6: Illustration of image patches

For a RGB image, patched image is generated for each channel. After generating patched image, the images are then rearranged linearly (like flattening). This rearranged the 2D

patches into 1D vector. Each pixels of the patch is considered as a token. For example a patch with 16 by 16 pixels is flattened and has 256 tokens. The next step is linear projection of flattened patches. The linear projection involves two main steps- wight matrix multiplication and bias addition. So, this step multiply each sequence of the patches with weights and add bias. The weights and bias are learned during the training process. So, the result of this step is lower dimension vector i.e. it has less elements than previous step. By reducing the dimensionality, we can extract essential features and captures the most important information by discarding less important details.

Like the traditional transformer, the vision transformer also makes use of position embeddings. The reason for this is that the attention in the transformer is positioned independent. Position embedding is added to each image patch indicating each patch location in the image.

Then the position embedding is fed into the transformer encoder. The first layer of transformer encoder is the self-attention layer. Self-attention allows each patch to attend to and gather information from other patches. It captures dependencies between the patches and also enable the model to consider the global context. After self-attention layer, vision transformer has feed forward network. The output of each patch is passed through a feed forward neural network. This helps capture complex non-linear relationship within the patches. The final layer layer is a classification layer or the MLP head. It maps the output of the tansformer into the desired output format.

4.5 Approach 1: U-Net

In this approach only U-Net architecture is used. We followed the architecture given by Ronneberger et al. [13] Some hyper-parameters like kernel size, and activation function should not be changed to gain a better result. Other hyper-parameters are tuned to acquire better results. Hyper-parameters used for training is:

- **Epoch :** 100
- **Learning Rate :** 1e-4
- **Optimizer :** Adam
- **Loss Function :** Focal Loss

Steps of how we implemented the U-Net architecture are given below:

1. **Convolution 1:** We used 3 channel color image of size 256X256 pixels as input. For the first convolution, we chose 64 filters. Kernel size and activation function is defined by the U-Net architecture. So, we keep the kernel size as 3 and use the activation function ReLU. For 3X3 convolution 1 pixel border is lost. Therefore, to keep the output the same size as the input we use padding 'same'. According to the architecture, the output of the first convolution is the input for the next convolution. The parameters for the next convolution are the same as the previous one.

2. **Max-pooling 1:** The output of the second convolution operation is gone through the max-pooling operation. We followed the U-Net architecture and performed max-pooling with 2X2 filter.

3. **Convolution 2:** In this step, we performed convolution 2 times like the previous convolution step. But this time, the number of filter is doubled from the previous one as suggested by the U-Net architecture. So, here, number of filters is 128.

4. **Max-pooling 2:** Like step 2, max-pooling operation with 2X2 filter is performed here again.

5. **Convolution 3:** This step is the same as step 3 but the number of filters is doubled from 128 to 256.

6. **Max-pooling 3:** Max-pooling operation with 2X2 filter is performed here again.

7. **Convolution 4:** This step is the same as step 5 but the number of filters is doubled from 256 to 512.

8. **Max-pooling 4:** Max-pooling operation with 2X2 filter is performed here again.

9. **Convolution 5:** Convolution with 1024 filters, a 3X3 kernel, ReLU activation function, and 'same' padding. Then another convolution is performed with the same parameters.

10. **Up-convolution 1:** In this step, we performed transposed convolution with number of output filters halves of the previous step (512) and stride 2. This is an important step because here, the concatenation of the feature map with the final output of step 7 (convolution 4).

11. **Convolution 6:** In this step, we performed convolution, then drop out, then convolution again. The parameters of these two convolutions are the same as the parameters for the step 7.

12. **Up-convolution 2:** In this step, we performed the same operation as step 10 (up-convolution 1). The only difference is the number of filters is reduced to 256 (half of the previous step) and the output is concatenated with the output of Convolution 3.

13. **Convolution 7:** This step is the same as the Convolution 3 step.

14. **Up-convolution 3:** In this step, we performed the same operation as step 12 (up-convolution 2). The only difference is the number of filters is reduced to 128 (half of the previous step) and the output is concatenated with the output of Convolution 2.

15. **Convolution 8:** This step is the same as the Convolution 2 step.

16. **Up-convolution 4:** This step is similar to step 14. But the number of filters is reduced to 64 and the output is concatenated with the output of step 1.

17. **Convolution 9:** This step is the same as the Convolution 1 step.

18. **Output:** The segmented output is gained by performing convolution with number of filters as same as the number of segmented classes (for our dataset it is 3), kernel size 1X1 (as described in the architecture), 'softmax' as activation function and with no padding. The softmax activation function normalizes the output values from 0 to 1. This value indicates the probability of a pixel falling to a particular class.

4.6 Approach 2: U-Net with ViT

In this approach combination of U-Net architecture with vision transformer is used. We followed the encoding part of U-Net architecture like we followed before, after that vision transformer is implemented, and then the decoder part of U-Net architecture is implemented. Hyper-parameters like kernel size, and activation function are remained same as previous. Hyper-parameters used for training is:

- **Epoch :** 100
- **Learning Rate :** 1e-4
- **Optimizer :** Adam
- **Loss Function :** Focal Loss

Steps of how we implemented the U-Net architecture with vision transformer are given below:

1. **Encoder (Downsampling Path):** This is similar to the encoder part of previous U-Net architecture. Same configuration as previous are used.

2. **Vision Transformer Integration :** This steps initialize a vision transformer module with specific parameters like image size, patch size, dimension, depth, channels etc. Then the ViT module is applied to the output of the last downsampling layer, reshaping and processing the output to match the model's architecture. After that, additional layers is used to adjust the tensor dimensions to match the subsequent upsampling layers.

3. **Decoder (Upsampling Path):** This step sequentially upsample the feature maps using the up convolution layers, where the decoded information is merged with the corresponding feature maps from the encoder. This step is similar to the decoder part of the previously discussed U-Net architecture, and same configuration is used here also.

4. **Output Layer:** This step generate the final logits (raw predictions) after the decoder.

So, in summery this approach performs the following steps:

- Encodes the input through the downsampling path.
- Applies Vision Transformer processing to the encoded feature maps.
- Decodes the processed feature maps through the upsampling path.
- Generates the final output logits.

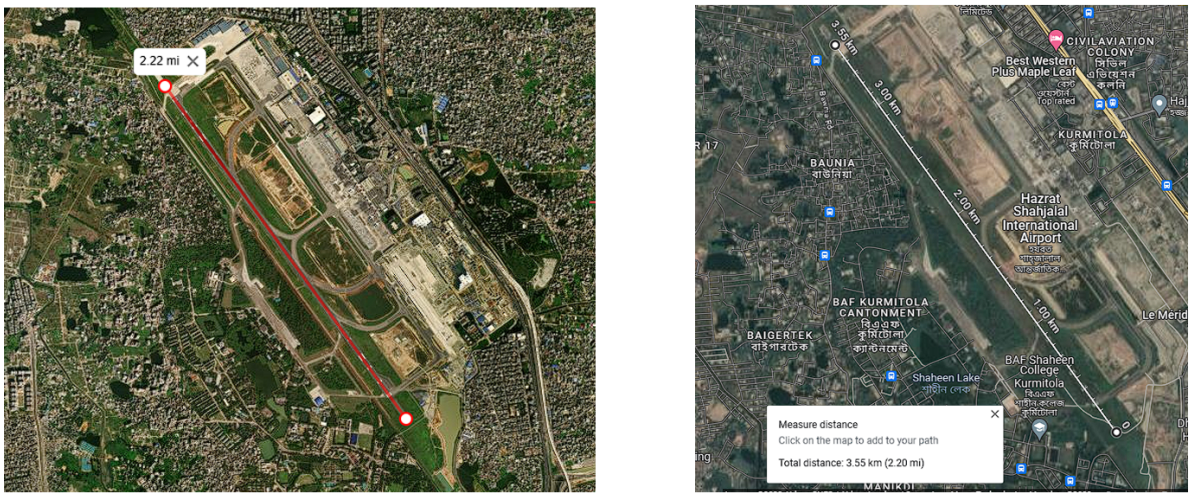
4.7 Measuring the Detected Land

Information about how much area one pixel of Apple Maps covered could not be found anywhere. This makes the whole measurement process harder. However, the solved this problem by counting how many pixels covers a place whose area/length is already known. The process is described in details below-

The length of the runway of Hazrat Shahjalal International Airport (DAC) is 3,505 meter [29]. We measured this distance from both Apple Maps and Google Maps to check how accurately the maps can measure distances. Both Apple Maps and Google Maps measured the runway length correctly as shown in Figure 4.7.

This process is repeated over other areas and in all cases both Apple Maps and Google Maps provide accurate distance measurements.

We take some places such as bridge, stadium etc. as references. On of the references was Dhanmondi Tennis Court. The length of a tennis court is 23.77 meter [30]. We save an image tile of highest zoom level from Apple Maps, rotates it, and then count how many



(i) (ii)

Figure 4.7: Runway length measurement of DAC from- (i) Apple Maps, (ii) Google Maps

pixels need to cover the length of the tennis court. Our findings shows that 90 pixels from Apple Maps is needed to cover the 23.77 meter length of the tennis court. This step is illustrated in Figure 4.8.

(i) (ii) (iii)

Figure 4.8: Process for measuring meters per pixel. (i) Original image downloaded from Apple Maps, (ii) Rotated image so that the length of the court is in vertical position, (iii) Count pixels of the court with using Adobe Illustrator

From the image, it can be seen that the length of the tennis court (23.77 m) takes 90 pixels in length. So, one pixel takes $(23.77 \div 90)$ m or $0.2641m$ approximately.

We repeated this procedure and measured distance/pixel for different areas. The calculation shows that on average each pixel is $0.262m$ in length and wide. Therefore, each pixels has approximate area of 0.068644 m^2.

For the measuring step, we simply counts how many pixels of an image is classified as one

particular class and then multiply the number of pixels by 0.068644 m^2. For example, if for a particular image, the model identified x pixels as desolated lands, then the area of desolate lands in that particular image is $x \times 0.068644m^2$.

4.8 Project Management

This thesis has conducted in a period of one year. Throughout this period, various task has accomplished in different times. The time and sequence in which this research is progressed is shown in the Gantt chart in Figure 4.9.

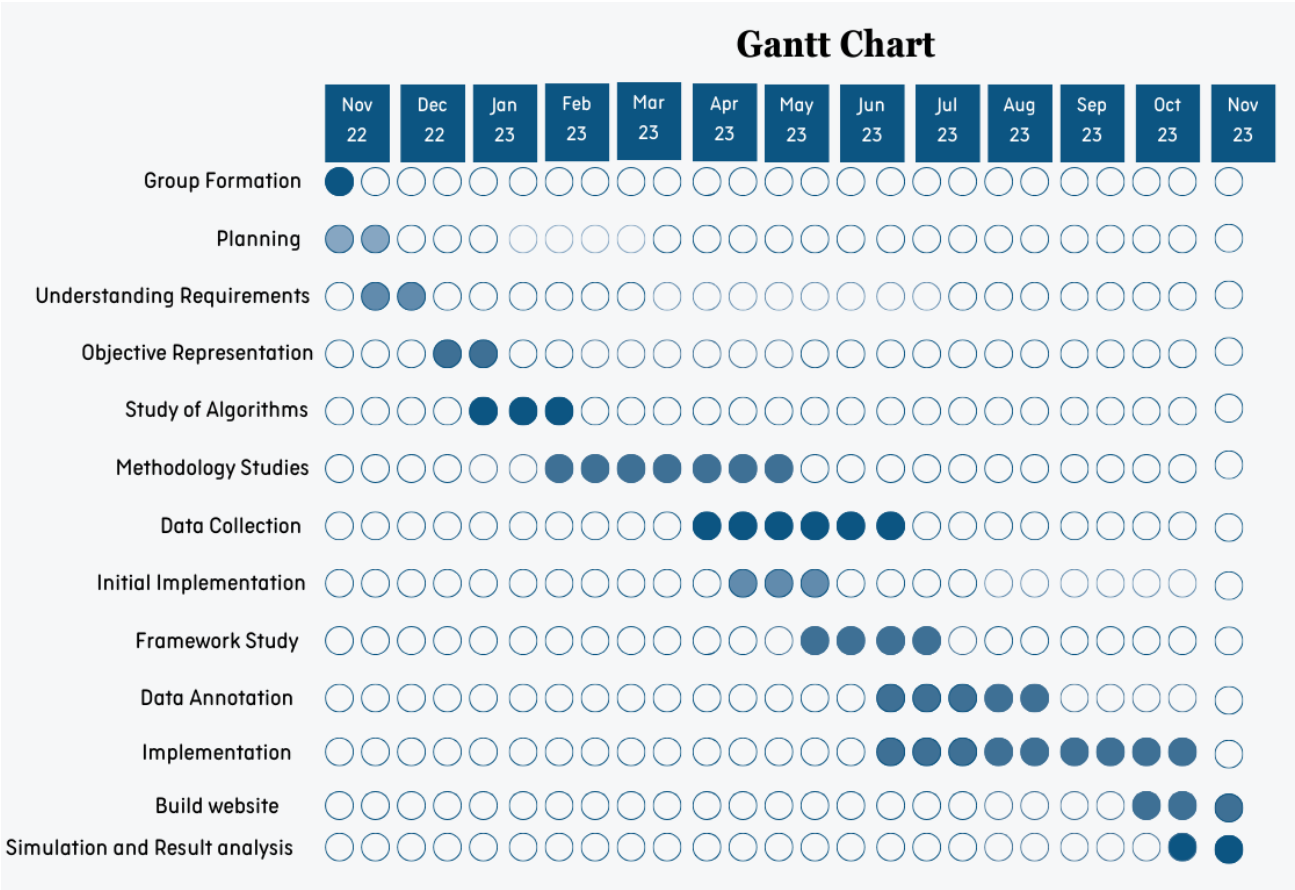


Figure 4.9: Sequence of processes in Gantt Chart

The process started in November 2022 with group formation. After that initial planning was done. Then the requirements for completing this thesis is studied and set the objectives. At that time, thesis topic has selected and we moved on studying different machine learning methods and algorithms required for this work. From February 2023, we started studying methodologies that are needed for this study. This process took upto May 2023. In the meantime, we collected satellite data from different sources and compared various factors among them such as- data qualities, time of taking the data and availability etc. Also, in

April 2023, we started initial coding and by the mid May 2023, we were able to generate preliminary results. From late May 2023 to July 2023, we studied various frameworks required in this thesis. Parallelly we annotated data in the period of June to August 2023. In June 2023, we started the main implementation and we have implemented two architectures to achieve semantic segmentation in our dataset. We also studied Django framework and build a website for demonstrating the work of this thesis. From late October to early November 2023, we simulate and analysed the results.

4.9 Financial Expenditure

All the tools (except Adobe Illustrator) and dataset used in this work are free of cost. One of the team members has purchased the Adobe Illustrator previously, and therefore we did not have to buy any licence for the product. We annotated dataset manually and no third party were involved in this work. So this work has accomplished without cost.

Chapter 5

Result

Several assessment metrics are taken into account to quantify performances such as- accuracy, precision, recall, F-1 score, and Jaccard index. Jaccard index is also known as intersection over union (IoU).

5.1 Performance Metrics

1. **Accuracy :** It is calculated as the ratio of correctly predicted instances to the total number of instances evaluated:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.1}$$

2. **Precision :** Precision measures the accuracy of positive predictions made by a model among all instances predicted as positive. Formula for precision is:

$$Precision = \frac{TP}{TP + FP} \tag{5.2}$$

3. **Recall :** Recall measures the ability of a model to correctly identify all relevant instances of a class among all true instances of that class in the dataset. Formula for recall is:

$$Recall = \frac{TP}{TP + FN} \tag{5.3}$$

4. **F-1 Score :** F1 score combines precision and recall into a single metric, providing a balance between both measures by calculating the harmonic mean of precision and recall. Formula for F-1 score is:

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{5.4}$$

5. **Jaccard Index:** Jaccard index also known as intersection over union (IoU) is a popular metric for evaluating segmentation task [16]. Details about this metric is discussed on the next section.

5.2 Intersection over Union

Accuracy is not a very good metric for evaluating semantic segmentation. The problem arises when most of the data is divided into some major classes. In our dataset, 'desolate lands' and 'other' are the two major classes. Figure 5.1 below shows an original and corresponding masked image of our dataset.

(a) Original image (b) Masked image

Figure 5.1: Original and mask image showing majority and minority classes

As we see most of the pixels here fall into 'other' classes. So if the model only predicts the major class i.e. 'other', the accuracy will improve. The problem is that when the model ignores the minor classes, it is still possible to get high accuracy. But in reality, the model is not good enough because it cannot segment minor classes. The inaccuracy of the minor classes is overshadowed by the accuracy of major classes. For this reason, IoU is used to evaluate performances for segmentation problems.

It is initially defined for the set operations. It is calculated by taking the intersection of two sets and dividing the result by the union of those two sets. In other words, IoU is the overlap between prediction and ground truth divided by the union between prediction and ground truth.

To visualize the concept of IoU let's consider the Figure 5.2 [31].

When the predicted truth doesn't intersect with the ground truth then the IoU is zero. For example, if there present some pixels of barren land in the upper left corner of the image but the model predicted barren land in the lower right corner IoU value will be zero. If the ground truth and predicted truth overlap somewhere then we will get some value for IoU. A good IoU means that the model predicted the pixels at the right position, as shown in

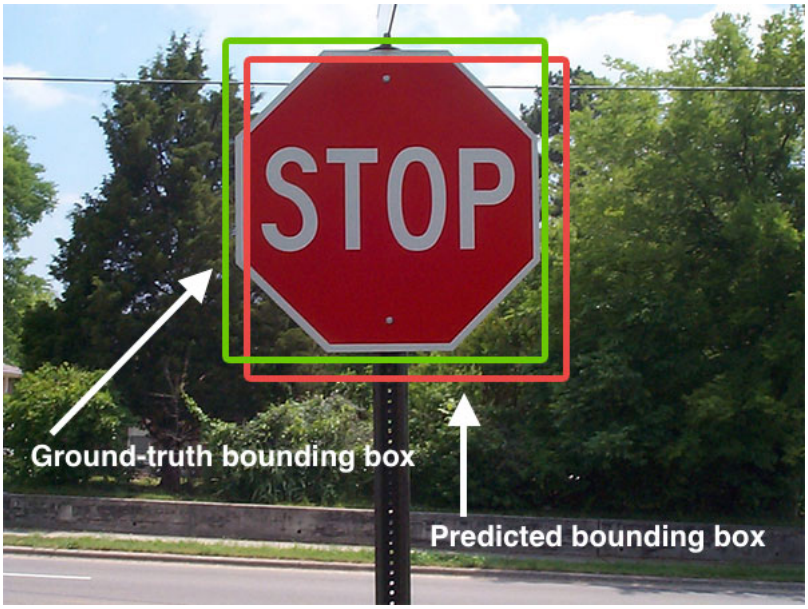


Figure 5.2: Visualizing IoU

Figure 5.3 [31]. Therefore, IoU is a popular metric to evaluate segmentation problems.

Figure 5.3: Three different IoU values for three different cases

Now, a problem arises when calculating the IoU. We cannot perform set union and intersection operations with our ground truth and predicted values. To solve this an approximation of IoU can be made using probabilities. This gives the equation for this approximation IoU' [32]:

$$IoU' = \frac{|T * P|}{|T + P - (T * P)|} \tag{5.5}$$

Note that, sometimes values of the numerator and denominator could be zeros. This could produce the exception 'divide by zero' in the program. So when implementing this formula in our code, we added 1 to both the numerator and denominator in order to prevent division by zero.

In this work for measuring IoU, Karas' definition [**?**] is followed which is equivalent to equation (5.5). IoU according to Keras' definition is:

$$IoU'' = \frac{TP}{TP + FP + FN} \tag{5.6}$$

5.3 Result Analysis

We have tested both Approach 1 and Approach 2 with same set of test data. The test dataset contains 232 images of 256 × 256 pixels each. So, both approaches has predicted 232 × 256 × 256 pixels or, 15,204,352 pixels each. In this experiment, all performance metrics are measured at pixel level.

5.3.1 Approach 1: U-Net

Training and validation loss vs epochs graph for sixty epoch is shown in the Figure 5.4:

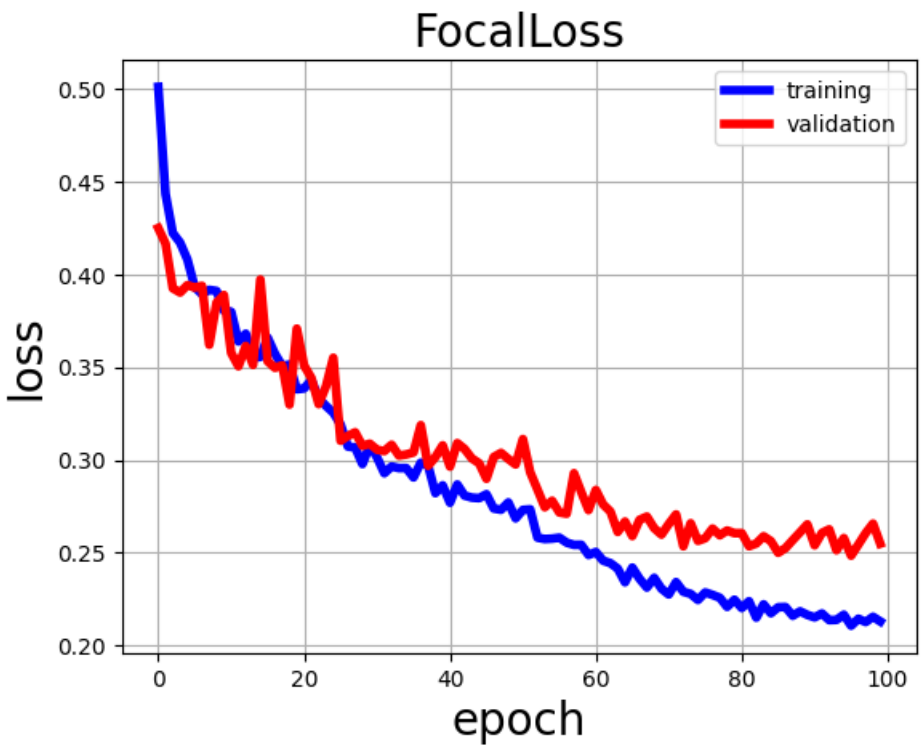


Figure 5.4: U-Net: Loss vs Epochs

Training and validation accuracy vs epochs graph for sixty epoch is shown in the Figure 5.5.

Confusion matrix for three class- 'Desolate lands', 'Water', and 'Other' using only U-Net architecture (Approach 1) are given in Figure 5.6.

We calculated accuracy, precision, recall, F-1 score, and IoU for all three classes by using equation (5.1), (5.2), (5.3), (5.4), (5.6) respectively. Performance score of these metrics

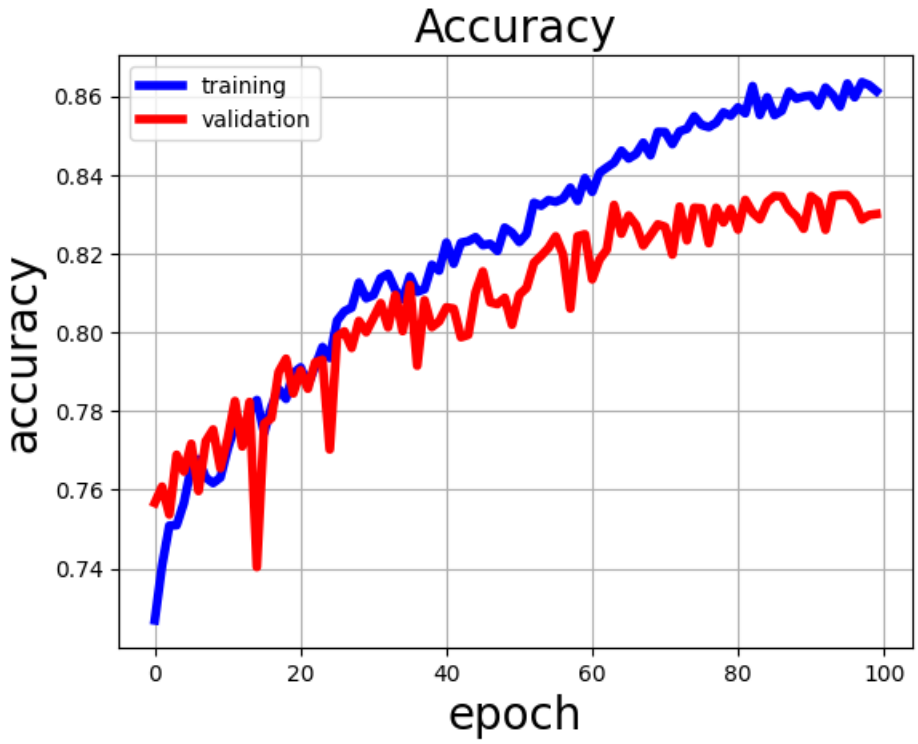
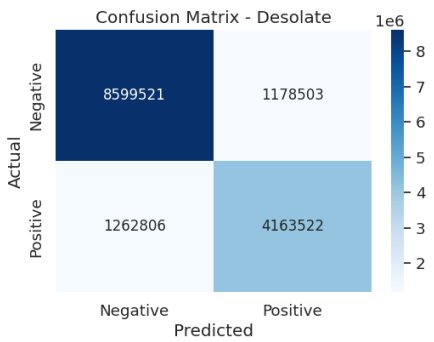
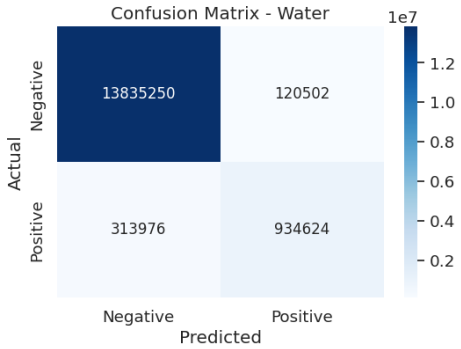


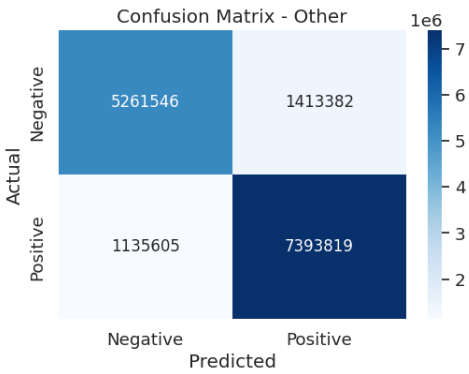
Figure 5.5: U-Net: Accuracy vs Epochs



(a) Desolate



(b) Water



(c) Other

Figure 5.6: Confusion matrices of Approach 1

Table 5.1: Performance metric of U-Net approach

	Desolate	Water	Other
Accuracy	0.839	**0.971**	0.832
Precision	0.779	**0.886**	0.840
Recall	0.767	0.749	**0.867**
F-1 Score	0.773	0.811	**0.853**
IoU	0.630	0.683	**0.744**

are given in Table 5.1.

5.3.2 Approach 2: U-Net with ViT

Training and validation loss vs epochs graph for one hundred epoch is shown in the Figure 5.7:

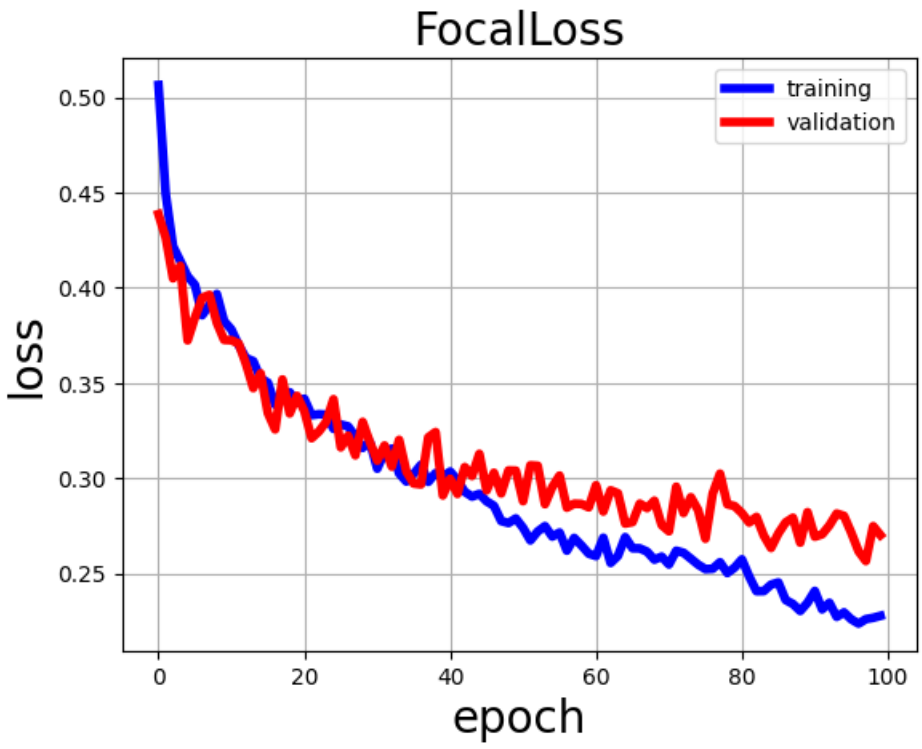


Figure 5.7: U-Net with ViT: Loss vs Epochs

Training and validation accuracy vs epochs graph for one hundred epoch is shown in the Figure 5.8:

Confusion matrix for three class- 'Desolate lands', 'Water', and 'Other' using modified U-Net with ViT architecture (Approach 2) are given in Figure 5.9.

We calculated accuracy, precision, recall, F-1 score, and IoU for all three classes by using equation (5.1), (5.2), (5.3), (5.4), (5.6) respectively. Performance score of these metrics

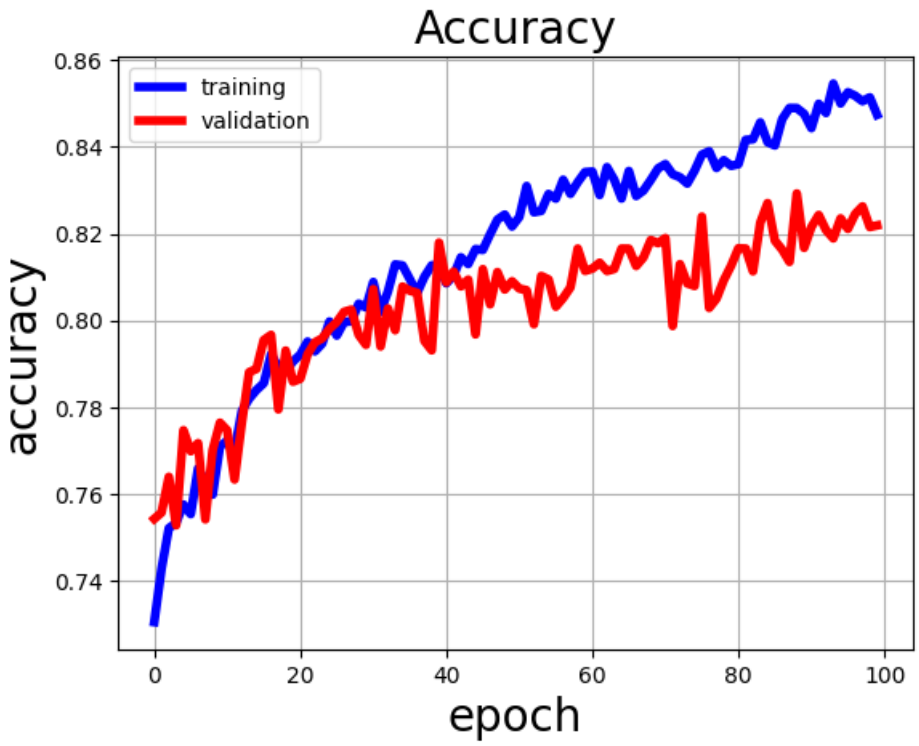
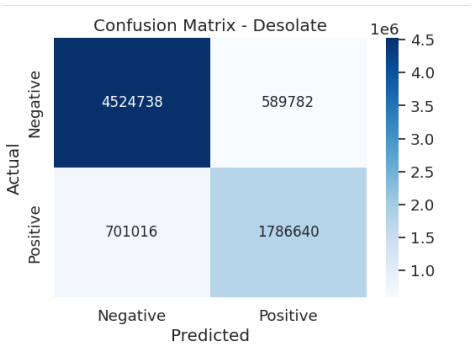
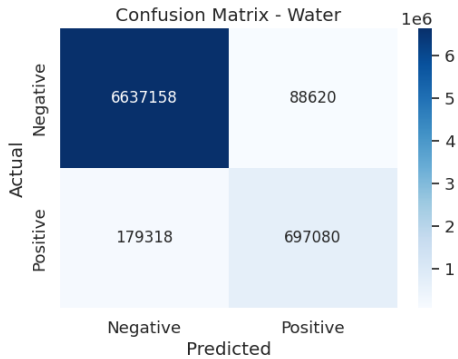


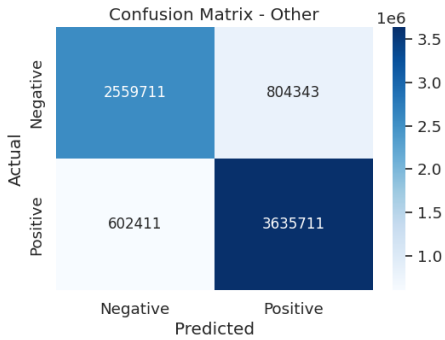
Figure 5.8: U-Net with ViT: Accuracy vs Epochs



(a) Desolate



(b) Water



(c) Other

Figure 5.9: Confusion matrices of Approach 2

Table 5.2: Performance metric of U-Net with ViT approach

	Desolate	**Water**	**Other**
Accuracy	0.846	**0.967**	0.840
Precision	0.800	0.822	**0.842**
Recall	0.757	0.767	**0.879**
F-1 Score	0.778	0.793	**0.860**
IoU	0.637	0.657	**0.755**

Table 5.3: Class-wise comparison of metrics between Approach 1 and Approach 2

	Desolate		**Water**		**Other**	
	U-Net	**U-Net with ViT**	**U-Net**	**U-Net with ViT**	**U-Net**	**U-Net with ViT**
Accuracy	0.839	**0.846**	**0.971**	0.967	0.832	**0.840**
Precision	0.779	**0.800**	**0.886**	0.822	0.840	**0.842**
Recall	**0.767**	0.757	0.749	**0.767**	0.867	**0.879**
F-1 Score	0.773	**0.778**	**0.811**	0.793	0.853	**0.860**
IoU	0.630	**0.637**	**0.683**	0.657	0.744	**0.755**

are given in Table 5.2.

5.4 Comparision between Approach 1 and Approach 2

Class-wise performance metric comparison can be found in Table 5.3.

From the Table 5.3 it can be seen that U-Net combined with ViT gives slightly better performance in 'Desolate' and 'Other' classes in almost every metrics than U-Net alone. However, in case of 'Water' class stand-alone U-Net architecture gives better result.

Table 5.4 and Table 5.5 give the comparison of overall performances between Approach 1 and Approach 2 based on micro-average and macro-average respectively.

Here, instead of class-wise performance, overall performance is considered. Overall performances can be measured by both micro-average and macro-average.

Micro-average compute metric globally by counting the total true positives, false negatives, true negatives, and false positives across all classes [33]. In another words, micro-average sums all the individual true positives, false negatives, true negatives, and false positives of all classes and compute metric using the sum up value.

Macro-averaging calculates each class's performance metric (e.g., precision, recall) and then takes the arithmetic mean across all classes. So, the macro-average gives equal weight to each class, regardless of the number of instances [33].

From the overall comparison, it can be seen that in all the performance metrics, U-Net

Table 5.4: Overall comparison based on micro-average

	Accuracy	**Precicion**	**Recall**	**F1-Score**	**IoU**
U-Net	0.881	0.822	0.822	0.822	0.697
U-Net with ViT	**0.884**	**0.826**	**0.826**	**0.826**	**0.704**

Table 5.5: Overall comparison based on macro-average

	Accuracy	**Precicion**	**Recall**	**F1-Score**	**IoU**
U-Net	0.881	**0.835**	0.794	**0.812**	**0.686**
U-Net with ViT	**0.884**	0.821	**0.801**	0.811	0.683

combined with ViT gives slightly better results than stand alone U-Net architecture when micro-average is considered. The scenario is not same when macro-average is considered.

5.5 Prediction

Figure 5.11 shows the examples of the model of approach 1 predicting on test data:

Figure 5.11 shows the examples of the model of approach 2 predicting on test data:

5.6 Verdict

Accuracy, precesion, recall, F-1 score and IoU of the model is satisfactory though there is a chance to improve the performance. Obtaining this kind of accuracy and IoU was possible because we used U-Net architecture combined with vision transformer. U-Net works better for a smaller dataset. One problem we noticed during the model prediction is that the however the model can predict majority classes correctly it suffers sometimes predicting minority classes. The reason for that is for minority classes, the model did not get enough data to train itself better. Enriching the dataset will solve this problem.

5.7 Demonstration of website

We have developed a website for the demonstration purpose of our works. The website is not available on public domain at this moment. The website takes input image, run our saved model on back-end and generate prediction. The prediction is displayed as output.

A step-by-step demonstration of the website is illustrated in this sub-section.

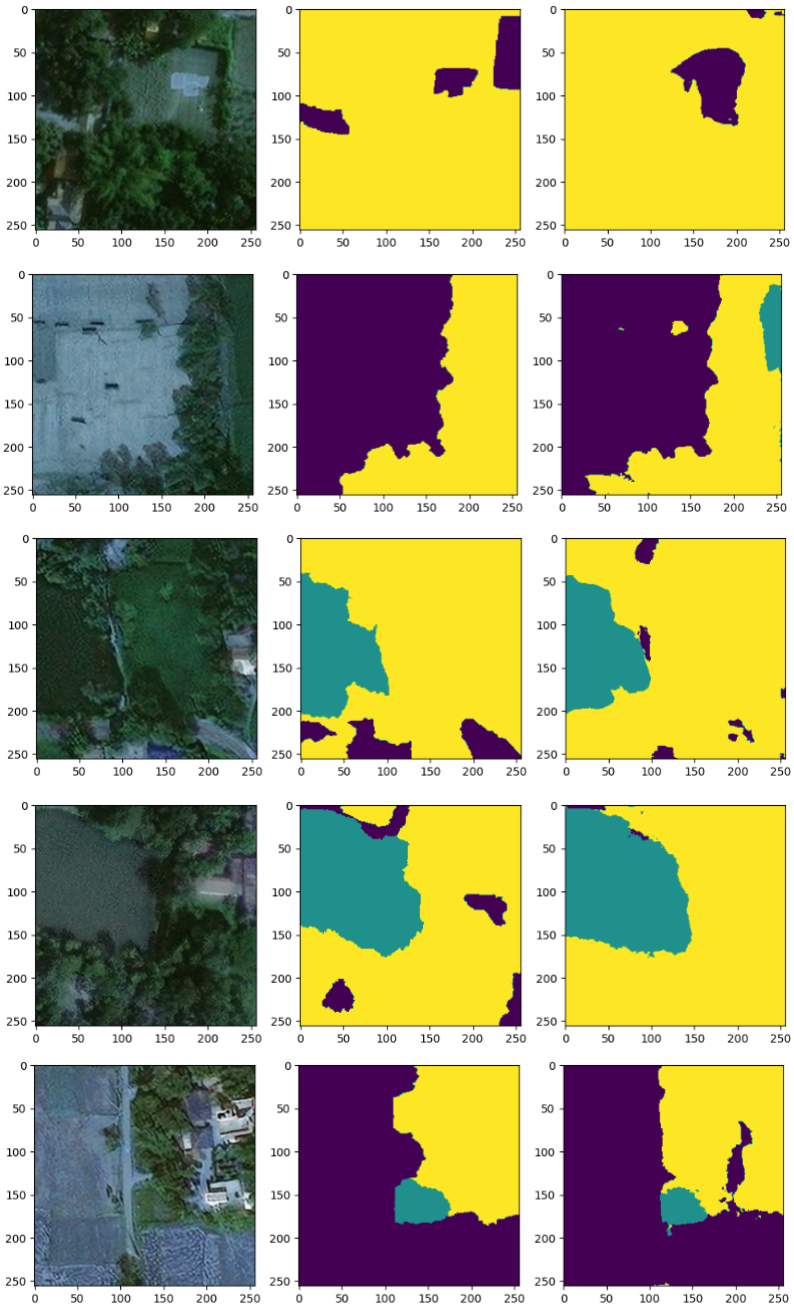


Figure 5.10: Model Prediction Examples (Approach 1). From left to right- original image, ground truth mask, predicted mask

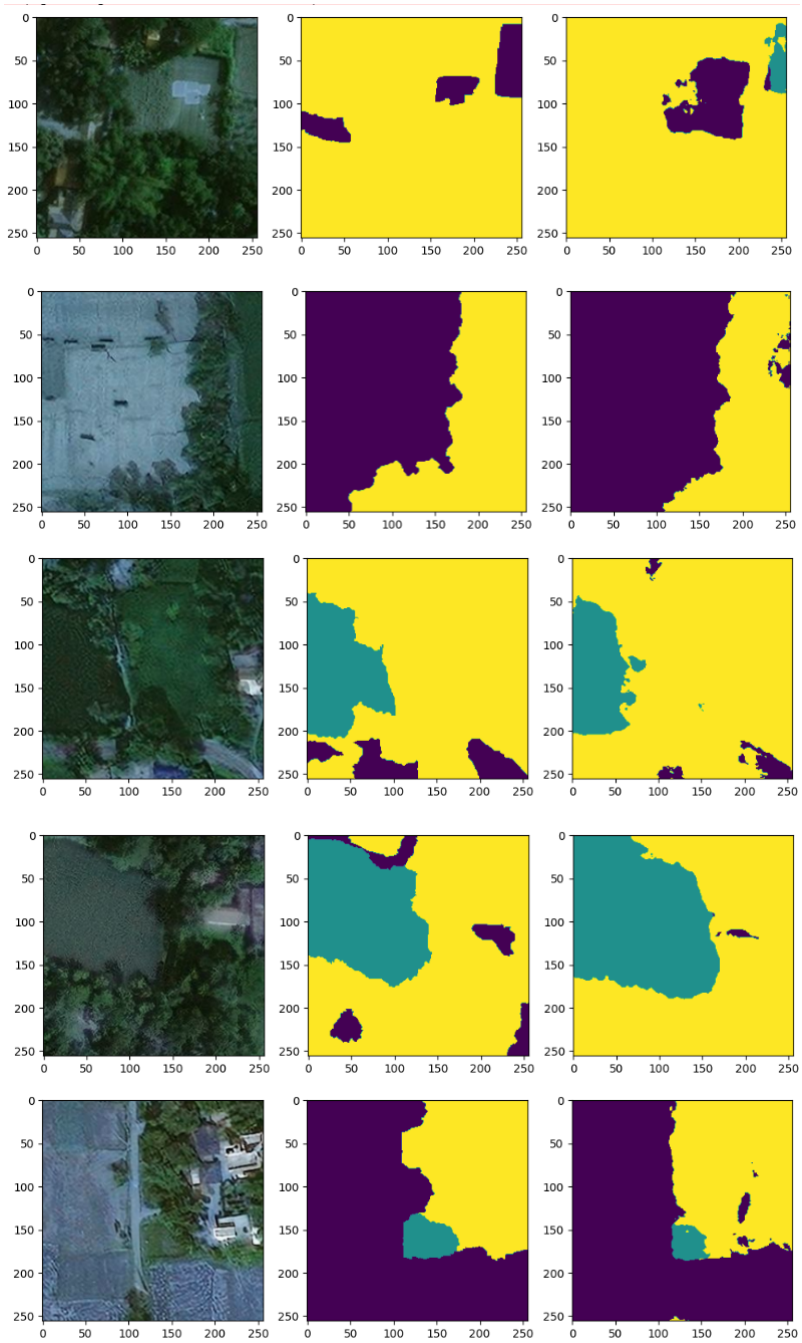


Figure 5.11: Model Prediction Examples (Approach 2). From left to right- original image, ground truth mask, predicted mask

1. **Homepage:**

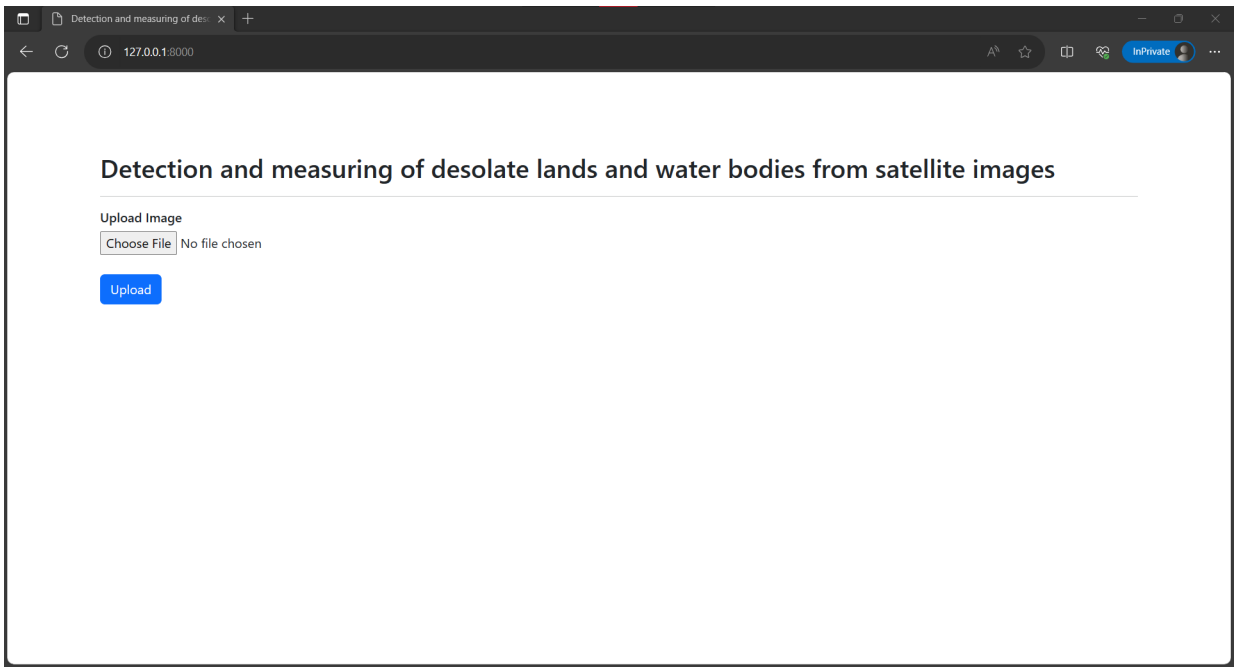


Figure 5.12: Homepage

2. **Upload image:**

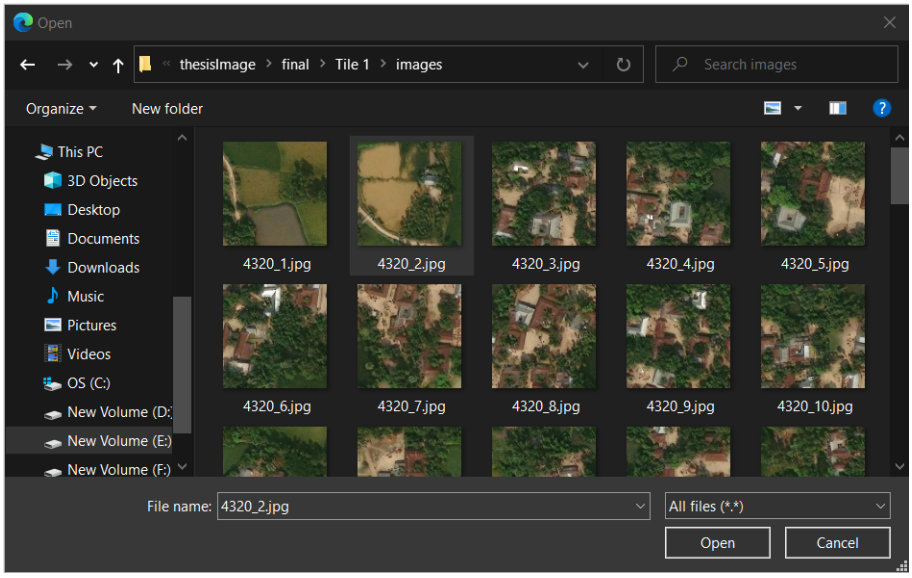


Figure 5.13: Uploading image

3. **Output:**

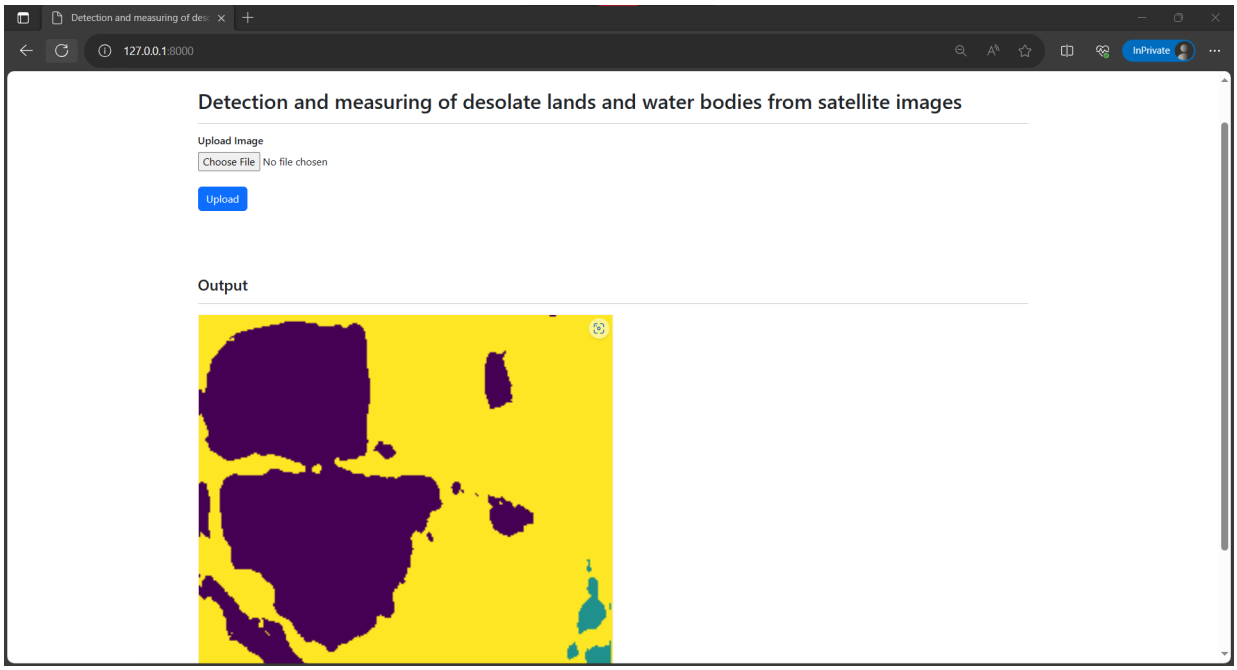


Figure 5.14: Predicted output image

Chapter 6

Future Work

Although the objective of our work is achieved, there are various scope to further expand the work in future. Due to the time constrains we were not able to explore some crucial factors. Our biggest obstacle was lack of dataset. Also annotating new dataset is too much time consuming. The current dataset can be increased in future as more the model is trained, more the results will improve. Moreover, including data from different district will bring out variation. If high resolution images are collected, it can increase the probability of classifying more accurately in future.

Again, appointing more members to validate the manually made annotation can be proved beneficial. In some extreme cases where human eyes are unable to identify certain areas, members may visit that place to get better idea of the surrounding in order to clarify confusions.

As it is already shown, we have implemented a website to get instant result of our work. The UI of the website can be updated and more options like batch upload can be implemented. Also, the website can be published in public domains so that everyone around the world can access it.

Furthermore, other models like- ResNet-50, DeepLab, HRNet etc. can be used and be compared among them based on various factors. Dataset can increased in such a way by gathering data from all around the world and training the model or advanced model to detect desolate land and water bodies from anywhere in the world. Also the model can be expanded to detect elements other than only desolate land or water body.

Chapter 7

Conclusion

In conclusion, the use of satellite data for measuring and locating desolate land and water body is a crucial step toward addressing the impending famine crisis. While field surveys are time consuming, expensive, and labor-intensive, satellite data provides a cost-effective and efficient alternative. In this study, we used medium-quality Apple Maps satellite images to create our own dataset and employed the U-Net deep learning architecture both with and without vision transformer for semantic segmentation.

As we discussed earlier, there were less dataset source available for our work, that's why we initiated with U-net architecture because it can generate excellent result even with small set of data. Being a fully convolutional network, the U-Net model can overcome limitations faced by traditional segmentation methods, such as loss of spatial information, insufficient boundary localization, and poor recognition of small objects within the image proving to be a highly efficient and accurate method for image segmentation tasks. The manually annotated datasets were fully utilized and appropriate training strategies have contributed greatly to the success of U-Net model.

During the course of this thesis, we observed that U-Net architecture efficiently captured and highlighted even the complex details and boundaries from the input images, thus leading to more accurate segmentation results. Various datasets and evaluation metrics have demonstrated the superiority of U-Net compared to traditional segmentation algorithms. From the methodology section, it's clear that the encoder part of U-Net is responsible for extracting higher-level features and reducing spatial dimension of input image and decoder layer combined with upsampling and skip connection helps retaining detailed spatial information.

Despite numerous advantages, U-Net still presents some challenges and limitations. This architecture highly relies on availability of labeled training data that can be time-consuming and expensive to acquire where domains are highly specialized or limited datasets. More-

over, U-Net may struggle segmenting overlapping objects or where boundaries are not clearly distinguishable.

Even though U-Net provided satisfactory results than other architecture, the modified U-Net with integrated Vision Transformer acquired a slightly better result than the previous. This architecture works same as U-Net but also utilizes Vision Transformer where the self-attention mechanism allows to capture global contextual information and multi-layer perceptron provide non-linear transformations to enhance the learned features. This architecture combines the benefits of both CNN and Transformers. The initial encoding and downsampling works as usual like U-Net. After that a Vision Transformer module is embedded and the rest structure follows as U-Net. This integration has potentially improved the overall performance and generalization ability of the network.

Lastly, it can be concluded that our work has demonstrated the effectiveness of the U-Net as well as U-Net with integrated Vision Transformer in image segmentation task. Our website will provide easy access to the model to any users. However, even though these two methods proved to be effective, further research is needed to enhance their ability to handle complex scenarios and improve the performance with limited labeled data.

References

[1] "A global food crisis." https://www.wfp.org/global-hunger-crisis. Accessed: 2023-05-15.

[2] I. F. S. P. Classification, "Bangladesh: Acute food insecurity situation march-april 2023 and projection for may-september 2023." https://www.ipcinfo.org/ipc-country-analysis/details-map/es/c/1156384/?iso3=BGD. Accessed: 10-Nov-2023.

[3] Food and A. O. of the United Nations, *Water for Sustainable Food and Agriculture*. FAO, 2017.

[4] C. E. Boyd, "Water use in aquaculture," *WORLD AQUACULTURE-BATON ROUGE-*, vol. 36, no. 3, p. 12, 2005.

[5] "Grow more food, save money to avert famine: PM." https://www.thedailystar.net/news/bangladesh/news/grow-more-food-save-money-avert-famine-pm-3141731. Accessed: 15-May-2023.

[6] "Landsat 9 image of bangladesh coast." https://www.usgs.gov/media/images/landsat-9-image-bangladesh-coast. Accessed: 15-May-2023.

[7] R. Arango, A. Campos, E. F. Combarro, E. Canas, and I. Díaz, "Mapping cultivable land from satellite imagery with clustering algorithms," *International Journal of Applied Earth Observation and Geoinformation*, vol. 49, pp. 99–106, 2016.

[8] M. Pedroso, J. Taylor, B. Tisseyre, B. Charnomordic, and S. Guillaume, "A segmentation algorithm for the delineation of agricultural management zones," *Computers and Electronics in Agriculture*, vol. 70, no. 1, pp. 199–208, 2010.

[9] J. Kumar, R. T. Mills, F. M. Hoffman, and W. W. Hargrove, "Parallel k-means clustering for quantitative ecoregion delineation using large data sets," *Procedia Computer Sci-*

ence, vol. 4, pp. 1602–1611, 2011. Proceedings of the International Conference on Computational Science, ICCS 2011.

[10] E. W. Schuster, S. Kumar, S. E. Sarma, J. L. Willers, and G. A. Milliken, "Infrastructure for data-driven agriculture: identifying management zones for cotton using statistical modeling and machine learning techniques," in *2011 8th International Conference Expo on Emerging Technologies for a Smarter World*, pp. 1–6, 2011.

[11] M. Liu and A. Samal, "A fuzzy clustering approach to delineate agroecozones," *Ecological Modelling*, vol. 149, no. 3, pp. 215–228, 2002.

[12] H. in the Loop, "Semantic segmentation of aerial imagery." https://www.kaggle.com/datasets/humansintheloop/semantic-segmentation-of-aerial-imagery. Accessed: 2023-11-10.

[13] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pp. 234–241, Springer, 2015.

[14] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[15] T. Eelbode, J. Bertels, M. Berman, D. Vandermeulen, F. Maes, R. Bisschops, and M. B. Blaschko, "Optimization for medical image segmentation: theory and practice when evaluating with dice score or jaccard index," *IEEE Transactions on Medical Imaging*, vol. 39, no. 11, pp. 3679–3690, 2020.

[16] J. Bertels, T. Eelbode, M. Berman, D. Vandermeulen, F. Maes, R. Bisschops, and M. B. Blaschko, "Optimizing the dice score and jaccard index for medical image segmentation: Theory and practice," in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part II 22*, pp. 92–100, Springer, 2019.

[17] Y. Guo, Y. Liu, T. Georgiou, and M. S. Lew, "A review of semantic segmentation using deep neural networks," *International journal of multimedia information retrieval*, vol. 7, pp. 87–93, 2018.

[18] S. Ohleyer and E. Paris-Saclay, "Building segmentation on satellite images," pp. 1–8, 2018.

[19] E. Guérin, K. Oechslin, C. Wolf, and B. Martinez, "Satellite image semantic segmentation," *arXiv preprint arXiv:2110.05812*, 2021.

[20] N. Kussul, S. Skakun, A. Shelestov, O. Kravchenko, J. F. Gallego, and O. Kussul, "Crop area estimation in ukraine using satellite data within the mars project," in *2012 IEEE international geoscience and remote sensing symposium*, pp. 3756–3759, IEEE, 2012.

[21] S. D. Mai, L. T. Ngo, and H. Le Trinh, "Satellite image classification based spatial-spectral fuzzy clustering algorithm," in *Intelligent Information and Database Systems: 10th Asian Conference, ACIIDS 2018, Dong Hoi City, Vietnam, March 19-21, 2018, Proceedings, Part II 10*, pp. 505–518, Springer, 2018.

[22] M. Awad, "An unsupervised artificial neural network method for satellite image segmentation.," *Int. Arab J. Inf. Technol.*, vol. 7, no. 2, pp. 199–205, 2010.

[23] S. Xu and M. Ehlers, "Automatic detection of urban vacant land: An open-source approach for sustainable cities," *Computers, environment and urban systems*, vol. 91, p. 101729, 2022.

[24] M. Wurm, T. Stark, X. X. Zhu, M. Weigand, and H. Taubenböck, "Semantic segmentation of slums in satellite images using transfer learning on fully convolutional neural networks," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 150, pp. 59–69, 2019.

[25] D. E. P. Sergey A. Bartalev and E. A. Loupian, "Mapping of arable land in russia using multi-year time series of modis data and the lagma classification technique," *Remote Sensing Letters*, vol. 7, no. 3, pp. 269–278, 2016.

[26] A. Ramadurai, "Understanding u-net architecture for image segmentation." Available online, 2018. Accessed on November 18, 2023.

[27] M. Tran, "Understanding u-net." Available online, 2022. Accessed on November 18, 2023.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[29] Wikipedia contributors, "Hazrat shahjalal international airport," 2023. [Online; accessed 12-November-2023].

[30] Wikipedia contributors, "Tennis court," 2023. [Online; accessed 12-November-2023].

[31] "Jaccard index." https://en.wikipedia.org/wiki/Jaccard_index. Accessed: 2023-05-15.

[32] F. van Beers, *Using intersection over union loss to improve binary image segmentation*. PhD thesis, 2018.

[33] Evidently AI Team, "Accuracy, precision, and recall in multi-class classification," 2023. [Online; accessed 12-November-2023].

Generated using Undegraduate Thesis LaTeX Template, Version 1.4. Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh.

This thesis was generated on Monday 27th November, 2023 at 5:34am.