Unsupervised Semantic Segmentation for Localization of WetLand Area Fluctuations

by

Anika Tahsin
23141058
Maisha Fairooz
23141060
Gazi Rehan Rabbi
20101080

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
School of Data and Sciences
Brac University
May 2024

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Anika Tahsin
23141058

Maisha Fairooz
23141060

Gazi Rehan Rabbi
20101080

Approval

The thesis titled "Unsupervised Semantic Segmentation for Localization of WetLand Area Fluctuations" submitted by

1. Anika Tahsin(23141058)
2. Maisha Fairooz(23141060)
3. Gazi Rehan Rabbi(20101080)

Of Spring, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on May 27, 2024.

Examining Committee:

Supervisor: (Member)

Dr. Md. Golam Rabiul Alam
Professor
Department of Computer Science and Engineering
School of Data and Sciences
Brac University

Program Coordinator: (Member)

Dr. Md. Golam Rabiul Alam
Professor
Department of Computer Science and Engineering
School of Data and Sciences
Brac University

Head of Department: (Chairperson)

Dr. Sadia Hamid Kazi
Chairperson and Associate Professor
Department of Computer Science and Engineering
School of Data and Sciences
Brac University

Ethics Statement

This research paper is done and contributed by all the group members, and it's plagiarism-free.

Abstract

This research delves deeply into the intricate dynamics of wetlands in Bangladesh, with a particular focus on the haors, utilizing continuous monitoring to grasp the nuanced temporal changes that occur. It introduces an innovative unsupervised semantic segmentation methodology tailored for analyzing the yearly fluctuations in wetlands. Leveraging the rich dataset provided by multi-temporal satellite imagery and cutting-edge unsupervised learning algorithms, this approach stands poised to revolutionize our understanding of wetland dynamics. At the heart of our methodology lies the strategic application of feature extraction and advanced clustering techniques, with a notable inclusion being the decoder model. These techniques enable the segmentation of wetland regions based on discernible patterns of expansion and contraction. Moreover, our research extends beyond mere segmentation, incorporating time series methods to forecast wetland fluctuations. By integrating predictive analytics into our framework, we strive to provide not just a snapshot of wetland conditions but also insights into their future trajectories. To validate the efficacy of our approach, rigorous comparative analyses with actual data are conducted. This empirical validation serves to enrich our comprehension of river system dynamics and lends support to ongoing wildlife preservation initiatives. Our methodology represents a significant advancement in unsupervised learning methods, adept at adapting to dynamic conditions without the constraints of labeled training data. Furthermore, the incorporation of advanced clustering techniques enhances our ability to pinpoint regions undergoing substantial changes, thereby facilitating targeted conservation efforts. Crucially, the journey continues after segmentation and prediction. Post-processing of segmentation results allows for meticulous accuracy assessment, ensuring the reliability of our findings. Through a series of meticulously designed experiments, we showcase the robustness and effectiveness of our methodology and model. By pushing the boundaries of unsupervised semantic segmentation and environmental research, we aspire to make meaningful contributions to the broader scientific community and pave the way for informed conservation strategies.

Keywords: Wetland, Semantic Segmentation, Unsupervised, Computer Vision, Image Clustering, RAM, Grounding DINO, SAM, ARIMA, Gaussian Hidden Markov Model

Dedication

This research is a dedication to our wetlands, recognizing their vital role in our ecosystem and our commitment to their preservation and sustainable management.

Acknowledgement

First of all, all the praise to the Great Allah, for whom we could complete our thesis without any major setbacks. Moreover, we want to show our gratitude and respect to our Supervisor, Dr. Md. Golam Rabiul Alam, sir. His guidance and feedback were the core of our thesis. Lastly, we are also grateful to our friends, teachers, parents, and mentors for their time and support.

Table of Contents

List of Figures

List of Tables

List of Algorithms

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

ACG Adaptive Concept Generator

$ACSeg$ Adaptive Conceptual Segmentation

ADC Associative Deep Clustering

AHE Adaptive Histogram Equalization

$ARIMA$ AutoRegressive Integrated Moving Average

$ASPP$ Atrous spatial pyramid poolingn

$ASTER$ Advanced Spaceborne Thermal Emission and Reflection Radiometer

$AWEI$ Automated Water Extraction Index

$CLAHE$ Contrast Limited Adaptive Histogram Equalization

CNN Convolutional Neural Network

CRF Conditional Random Field

DAC Deep Adaptive Clustering

$DCNN$ Deep Convolutional Neural Networks

DEM Digital Elevation Model

$DINO$ Deeper Into Neural Networks

DPN Deep Parsing Network

$ENet$ Efficient Neural Network

$FCCRF$ Fully Connected Conditional Random Field

$FCNs$ Fully Convolutional Networks

$FLOPS$ Floating Point Operations Per Second

GAP Global Average Pooling

$GHMM$ Gaussian Hidden Markov Model

GIS Geographic Information System

$GLCF$ Global Land Cover Facility

$GNNs$ Graph Neural Networks

GPU Graphical Processing Unit

HDR High Dynamic Range

HP Hidden Positives

$HRWI$ High Resolution Water Index

IIC Invariant Information Clustering

IoU Inter-section Over Union

$LGED$ Local Government and Engineering Department

$LISA$ Large Language Instructed Segmentation Assistant

$LLaVa$ Large Language and Vision Assistant

LLM Large Language Model

$LoRA$ Low-Rank Adaptation of Large Language Models

$LSTM$ Long Short Term Memory

$LUSS$ Large-Scale Unstructured Semantic Search

MAE Masked Autoencoder

mAP mean average precision

$MAPE$ Mean Absolute Percentage Error

MF Mean Field

MIM Mutual Information Maximization

$MINE$ Mutual Information Neural Estimation

$mIoU$ mean Inter-section Over Union

MIV Mutual Information Volume

$MNDWI$ Modified Normalized Difference Water Index

MRF Markov Random Field

$NDWI$ Normalized Difference Water Index

NIR Near Infrared

NLP Natural Language Processing

PA Pixel Accuracy

PIC Power Iteration Clustering

$PiCIE$ Pixel-level feature Clustering using Invariance and Equivariance

RAM Recognize Anything Model

$RMSE$ Root Mean Squared Error

SAM Segment Anything Model

SAR Synthetic Aperture Radar

$SOTA$ State-of-the-art

$SPNN$ Scale Invariant Probabilistic Neural Network

SSD Sum of Squared Difference

$STEGO$ Self-supervised Transformer with Energy-based Graph Optimization

$Swin-T$ Swin Transformer

$USFWS$ U.S. Fish and Wildlife Service

ViT Vision Transformer

VLM Vision Language Model

Chapter 1

Introduction

Wetlands are mainly considered as the area of inland or coastal land partly saturated and covered by water bodies. Wetlands are regions where water either consistently covers the soil or is present either at or near the soil's surface throughout the year, including throughout the time of the growing seasons [35]. It encompasses a variety of ecosystems, such as swamps, forested wetlands, bogs, wet prairies, prairie potholes, mangroves, and different types of marshes (salt, brackish, intermediate, and fresh), as well as vernal pools.

Figure 1.1: Wetland

Wetlands differ from rivers, lakes, etc, because of the differences in topology, soil water chemistry, climate, and vegetation. This wetland acts as a water filter, giving food to fish and wildlife. The large wetland may consist of several small wetlands being vulnerable to many environmental factors. Some factors are climate change and land use dynamics, which cause the wetlands to expand or shrink over time. Many wetlands get dried around some time of the year, which can be disastrous for living beings. For this reason, conservation efforts need to be made.

1.1 Identifying Wetlands: Understanding Their Characteristics

Wetlands are characterized in different ways and can influence different species of animals and plants. Therefore, it is important for us to understand the importance of its conservation as well as its identification.

Wetlands are primarily characterized by the presence of hydric soil, which remains saturated or flooded for extended periods during the growing season, which creates anaerobic conditions. This type of soil is typically dark and rich in organic material, making the water have a darker color than usual, and also provides a fertile environment for the growth of specialized vegetation. The water levels in wetlands can fluctuate due to seasonal changes and groundwater growth. Water regime differs from marshes, which are often changed to bogs and swamps to have a stable water level. So, we can see that biodiversity is created within the wetland itself.

On the other hand, rivers, lakes, and other water bodies have different hydrology properties. Unlike wetlands, rivers and lakes can maintain a stable presence of water and do not have the presence of hydric soils. They may have a variety of aquatic plants.

(a) A Wetland (b) A Lake

Figure 1.2: Identification of Wetlands. Here, we can see there is a difference of color in the wetland and lake because of their soil properties.

Wetland plays a major role in the survival of both people and animals, being a dynamic ecosystem that can undergo significant changes over time. These changes: Shrinkage may indicate environmental stress or degradation, and Expansion could be a signal recovery in water regimes. They also play a crucial role in water purification, flood control, and groundwater recharge, which maintains the ecological balance, biodiversity, and overall health of our planet. Also, human beings' localization can be detected by the state of the wetland. Knowing the precise locations

of shrinking or expanding the water resources can be managed more effectively, ensuring ecological services are maintained.

(a) Shrunk State of Tanguar Haor, Bangladesh in 1997

(b) Expansion State of Tanguar Haor, Bangladesh in 1999

Figure 1.3: Fluctuation of Tanguar Haor. In Figure (a), in the year 1997, Tanguar Haor had a state of shrinkage, indicating environmental stress. and in Figure (b), the year 1999 Tanguar Haor had improved and expanded.

In addition to water management, wetlands act as carbon sinks, storing large amounts of carbon that help to mitigate climate change. Localizing the changes in wetland areas assists in assessing their role in carbon sequestration and developing strategies to enhance their capacity to combat climate change.

This ecosystem provides a habitat for thousands of species, both aquatic living and terrestrial. It also helps to fight climate change. Wetland saves us from many natural hazards. For example, it incepts high tide, spreads the force of water that is incoming, and also, when heavy rain occurs, it absorbs the water into the porous ground which is beneath the wetland surface. Unfortunately, wetlands are being destroyed in various ways. Because of the climate, it is destroyed, and sometimes pesticides also cause the wetland to be distorted. Pesticides and fertilizers can migrate to wetlands, causing the wetlands to be distorted.

In Bangladesh, the Teesta River Basin (TRB) is crucial for agriculture in the downstream area. However, water shortages have been created in recent years, which challenges the local farmers. A study [48] was held to measure water shortages and identify crop-related problems and their impact on agriculture. The severity of the water shortage in the Teesta River Basin was starkly evident in the data collected from four villages in the Nilphamari district in April 2015. Over the past 15 years, there has been a significant decrease in the river's water levels. This shrinkage has led to higher irrigation costs, making farming more expensive and less profitable. The farmers' reports of increased irrigation expenses underscore the urgency of the

situation. Due to water scarcity, many farmers are growing maize and tobacco instead of rice. They are also facing soil contamination due to decreased soil fertility. The cost of fertilizers also increased. Shrinkage affected crop production and caused a problem in the sustainability of agriculture in the region.

A recent study conducted by the U.S. Fish and Wildlife Service (USFWS) has revealed that agricultural pesticides are causing substantial harm to wild ducklings in the prairie pothole region of the United States [1]. The insecticides are either "acutely toxic to waterfowl, to the aquatic vertebrates on which the adult and juvenile waterfowl depend for food, or both" [1].
From 1970 to 2015, approximately 35% of the world's wetlands vanished, and this decline is still speeding up. In fact, wetlands are disappearing three times quicker than forests [2], [41].

With this concern, to effectively segment wetlands, we focus on its key identity feature, hydric soil, which can be recognized by its dark color or organic richness through water bodies. By identifying the element in unsupervised data, we can accurately map and analyze wetland areas and aid in their conservation and management. Our research gives a detailed explanation of methodology by mixing spatial and spectral information. It also incorporates advanced clustering techniques and temporal analysis.

Figure 1.4: Wetland Segmentation

Unsupervised learning is beneficial, as it ensures the scalability of the approach, which can be implemented in various geographical locations. We are using unsupervised semantic segmentation as it automatically categorizes and identifies objects and regions without labels, making the process more convenient. Unlike supervised segmentation, which requires annotated and labeled regions, this method does not need a specific training dataset as they are designed to be like this. This convenience is particularly beneficial for the dynamic and diverse nature of wetlands, where variations in water level and types of vegetation are common. This method allows the

capture of uncommon spectral signatures of various wetland features and also allows a relationship among pixels, which improves the accuracy of the boundaries of wetlands. Also, it allows temporal analysis, which observes the place or object for a long time to detect the changes that it has undergone throughout time. We can learn a lot of advanced clustering techniques through this research as well, for example, k-means clustering or hierarchical clustering which are used in unsupervised semantic segmentation. It supports semantic segmentation by grouping the pixels with the same characteristics. Unsupervised semantic segmentation helps in scalability, as well as dealing with extensive wetland areas. By using unsupervised semantic segmentation, we can gain a proper insight into the changes in wetlands. With the powerful tool of unsupervised semantic segmentation, we can identify the wetland status and determine the shrinkage and expansion of the wetland. This knowledge can be instrumental in saving the ecosystem and making people aware of the situation. We can actively contribute to the restoration and conservation of wetlands, a cause that is crucial for our environment and future.

Semantic segmentation is a challenging task in automatic image processing, so implementing unsupervised semantic segmentation can be tricky. Semantic segmentation is a process by which an object from an image can be identified at a much finer granularity than the classification. It reduces the difficulty of labeling the training data as it does not require labeled data. To put it simply, segmenting an image can require over 100 times more effort for a human annotator compared to simply defining or drawing bounding boxes [17]. Moreover, in intricate fields like biology, astrophysics, or medicine, the ground-truth segmentation labels may be unclear, unknown, or demand specialized expertise to establish [16].

Figure 1.5: Wetland Segmentation with PaliGemma Vision Language Model

Incorporating unsupervised semantic segmentation with time series analysis prediction, we can better understand the pattern and predict the fluctuation over time. As we move forward, we will integrate sophisticated models and real-time monitoring to ensure the sustainability and resilience of wetland ecosystems, ultimately contributing to our planet's health and future generations' well-being.

1.2 Research Problem

Wetlands are not static. They change their shape and sizes over the inter- and intra-annual timescales, which affect aquatic life; the ecosystem also affects localization. There are people whose livelihood depends on these rivers and lakes, etc., for which they choose to stay near the wetland. The expansion and shrinkage of the wetland affect their living, too.

In Dhaka, the spatial and temporal shift dynamics in wetlands were measured by analyzing four Landsat images. They employed a supervised classification algorithm and post-classification change detection technique within the Geographic Information System (GIS) environment. The results of accuracy of the wetland maps generated from Landsat data ranged from 87% to 92.5% [4]. The research uncovered a notable decline in the area of wetlands, rivers, and lakes within Dhaka city over the past 30 years, with reductions of 76.67% and 18.72%, respectively [4]. As a result, every year, city residents endure severe waterlogging issues during the rainy season. In a report assessing the environmental strategy of Dhaka, waterlogging stands out as a significant issue causing suffering to the city [4]. This affected the drainage system in the city which created the water-logging. Therefore, we can see that the shrinkage and expansion also affect city life. The wetland is decreasing faster than the forest, which is very concerning. This is a global crisis and a nightmare for living, breathing beings.

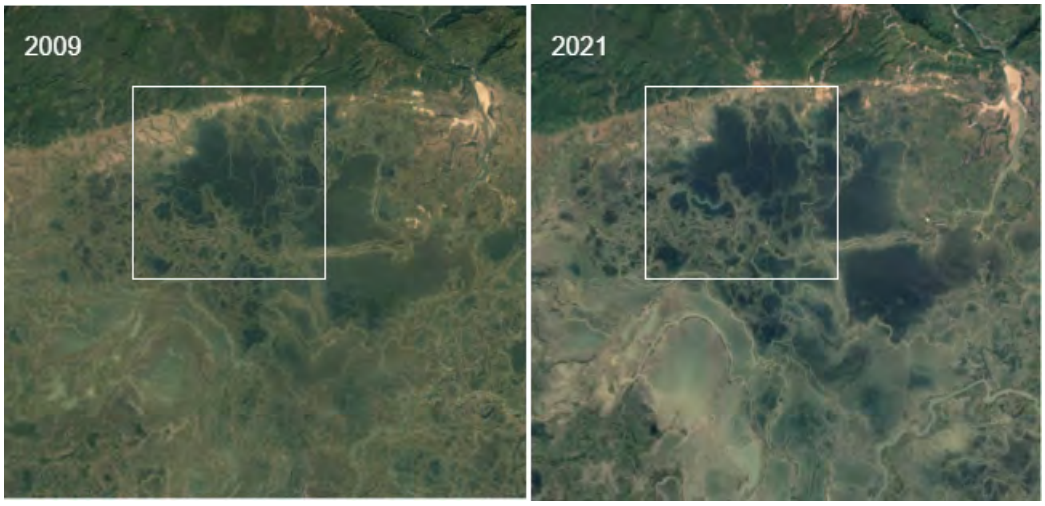


Figure 1.6: Wetland Segmentation Progress

Recent technologies, which include remote sensing technology and artificial intelligence, have opened a new view in our journey to determine the dynamics of wetlands [33]. The precise mapping and constant monitoring of wetlands require advanced remote sensing techniques and image analysis methodologies. One of the main tasks in wetland segmentation is semantic segmentation, which involves separating the wetland area into different classes, for example, open water, bare soil, vegetation, and many more. The traditional supervised methods for this wetland segmenta-

tion rely on labeled training data sets. This labeled trained data can be expensive, limited in scope, and time-consuming. However, this problem can be solved using unsupervised semantic segmentation, which is an alternative to this method.

The primary goal of this unsupervised semantic segmentation research is to uncover and identify semantically meaningful categories within image collections, all without any form of annotations. The algorithm needs to generate features that have clear semantic meaning and are concise enough to create separate clusters for each pixel. Various approaches have been introduced in semantic segmentation systems to acquire knowledge and learn from less precise forms of labels like bounding boxes, scribbles, point annotations, classes, or tags [23]. However, only a handful of studies have set forth into the realm of semantic segmentation without any human supervision or reliance on motion cues. Approaches like DeepAqua [42], Invariant Information Clustering (IIC) [24], and PiCIE [25] strive to acquire semantically meaningful features by incorporating transformation equivariance coupled with a clustering step to enhance feature compactness. Due to the absence of previous knowledge of the task in computer vision, the models are required to be trained to obtain our desired results, but with the help of this unsupervised semantic method, it is now much easier to obtain the desired result as it does not require any labeled data..

Now, some questions in hand can be noted with this research topic,

How can we leverage state-of-the-art (SOTA) machine learning techniques to predict future fluctuations in wetland areas? What methods can we use to segment wetland areas and analyze and forecast the deformation of wetlands over time?

We contribute to solving the above problems with the help of our research. This method of unsupervised semantic segmentation is more flexible and versatile as it does not rely on the annotated images for training.

This thesis embarks on a pioneering exploration into the world of unsupervised semantic segmentation with an amazing computational technique that promises to explore the intricate puzzle of wetland shrinkage and expansion. With the help of this model, we can identify the underlying semantic information within the wetland image, which will help us classify many things in the ecosystem and delineate the boundaries of the wetland without labeled training data made by humans. We aim to develop an accurate and robust methodology that helps to identify and classify the objects of wetlands in Bangladesh. The main goal is to provide land managers, environmental scientists, and policymakers to understand wetland dynamics better and make wise decisions for the restoration and conservation of these wetlands in our country.

1.3 Research Contributions

In this research, we developed an advanced to identify and map wetlands by using high-dynamic images from satellites without pre-labeled data. Specifically, we propose a combined model designed and trained to enhance the accuracy of wetland segmentation and track fluctuations over time. The main contributions of our work are summarized as follows:

- We created a unique dataset by extracting image data from Google Earth, focusing on Bangladesh's wetlands, including Tanguar Haor, Hakaluki Haor, Khorchar Haor, Hail Haor, and such. This effort provided a diverse and representative sample of these ecosystems, spanning from 1983 to 2022, and fills a gap as such datasets are not readily available elsewhere, marking a significant contribution to ecological and environmental research.

- We built an inference that integrates the power of three distinct models—Recognize Anything Model (RAM), Grounding DINO, and Segment Anything Model (SAM). This cohesive approach boosts and leverages the strengths of each model, enhancing the overall accuracy and efficiency of wetland segmentation and forecasting.

- We enhance wetland segmentation by fine-tuning with the Segment Anything Model (SAM) and leveraging its zero-shot generalization design and decoder-only model. This process utilizes prompts from Grounding DINO, bounding boxes, and wetland descriptions to produce binary masks, augmenting segmentation accuracy for diverse object categories and advancing ecological research.

- We made substantial modifications to the existing probabilistic set-level approach so that it suits our model. These modifications included refining the probabilistic algorithms to improve their precision and reliability in finding the total area of the segmented portion and predicting wetland fluctuation.

- We used time series approaches after segmentation with our proposed model to forecast wetland area fluctuations in the year 2025 in Bangladesh. This approach addresses a previously unattempted challenge, providing valuable insights and predictions that are critical for wetland conservation and management.

- We compiled and curated a comprehensive dataset of Bangla tags tailored for this research. This involved gathering relevant tags that are commonly associated with wetland features and phenomena, thus adding a new language to the existing tagging resources.

This detailed contributions section highlights each major achievement of our research, showcasing the innovations and improvements made to existing methods and how they specifically apply to wetlands in Bangladesh and demonstrating the potential of state-of-the-art machine learning techniques in environmental monitoring.

1.4 Research Organization

In the upcoming chapters, we will delve into the methodology, theoretical functions, and practical applications. We begin with a discussion of some related works on wetland self-supervised data (Section 2.1). Followed by reviewing existing papers on unsupervised semantic segmentation along with some self-supervised segmentation papers (Section 2.2). Afterward, we show our working plan in a top-view block diagram throughout the research in Chapter 3 to Chapter 5.

Chapter 2

Literature Review

Supervised semantic segmentation, which involves human interaction with labeling training data manually. This whole process is time-consuming and expensive, and there can be human error because of errors that will result in incorrect predictions. Supervised segmentation methods rely on these human-labeled data to learn and classify pixels or areas to predict accurately on the test data. Meanwhile, the unsupervised segmentation approach resolves this issue without relying on pre-labeled or human-annotated data. The unsupervised semantic segmentation method has emerged as a promising avenue that offers adaptability, automation, etc. The literature review helps us to find the key research findings and methods in the field of unsupervised semantic segmentation by highlighting the challenges and opportunities. Through going through the papers, we got to see the perspective of the methods in a different way; they underscore the potential of the methods to provide accuracy in addressing the challenges.

2.1 Related Works

We now move to the wetland localization and found a fact that water extention is a particular issue to look at. Water extension varies over time and space, which results in multiple annotations for the exact same area [42].

With that in mind, Pena et al. introduced us to a model called DEEPAQUA that uses cross-modal knowledge distillation. The model adopts the Normalized Difference Water Index (NDWI) to train a Convolutional Neural Network (CNN) for water segmentation from Synthetic Aperture Radar (SAR) image data [42]. As a student model, they used U-Net, and for the teacher model, NDWI. Their goal was to detect vegetated water without manually annotated data. For the qualitative results of their method, they used Pixel Accuracy (PA) and Intersection Over Union (IOU) on their test set. Both metrics outperformed the best competitor model, Otsu. The results were on Otsu 0.895, and 0.646 for PA and IOU respectively, and for DEEP AQUA [42] it was a PA of 0.971 and an IOU of 0.890. This exhibits the effectiveness of their approach in cross-modal knowledge distillation with unsupervised data.

Hu et al. explain a deep learning classification approach formed on CHRIS hyperspectral image in their paper, where they also mention the presence of fully

connected methods to conduct their test on the Huanghe River Estuary coastal wetland data [20]. The data categories included Reed, Tamarix, Spartina, Water, Tidal Flat, and Farmland. They determined that performing K-L transformation, subsequently a spectral-only feature, followed by a combination of spectral and texture features would result in a higher accuracy. Using a Deep Convolutional Neural network (CNN), their model outperformed all other approaches by 4.15% [20].

Here in the paper, Hassan et al. use a digital elevation model (DEM) to determine the slope of the area and the area where vegetation is needed. The research results indicate the utilization of Landsat 5 (TM) images from 1989 and 2013 [5]. They selected four channels, including RED, NIR, Blue, and Green, to extract spectral information for surface water and vegetation in the study area where the Landsat data were obtained freely from the GLCF (Global Land Cover Facility). On the flip side, they obtained an ASTER (Advanced Spaceborne Thermal Emission and Reflection Radiometer) Digital Elevation Model (DEM) through a collaborative effort between NASA and the Japan Space System [5]. In this research, DEM data was employed to calculate slope and drainage networks. To delineate the study area within the Landsat image, a vector polygon map of the Sirajganj district was acquired from the Bangladesh Local Government and Engineering Department (LGED). To achieve the objective, they employed the NDWI method, which proved to be the most effective in extracting surface water. This process included identifying both the primary water channels and smaller water bodies, resulting in the classification of approximately 60,000 hectares, equivalent to 24% of the total study area. Additionally, they generated slope and drainage density maps using DEM data. Among these, the slope map with values less than 20% was selected as the most suitable topographic factor for ecological restoration.

Mark Hamilton et al. affirmed that modern self-supervised visual backbones can produce state-of-the-art results in Semantic segmentation without supervision [32]. Their architecture is inspired by demonstrated correlations between deep features and ground truth label co-occurrence. Their approach utilizes an unsupervised learning signal by introducing us to a contrasting loss that highlights feature correspondences. Their system, STEGO, creates accurate semantic segmentation predictions from low-rank representations. They demonstrate that STEGO's loss is equivalent to MLE in Potts, linking it to CRF inference. Models across our entire dataset of pixels. They also show that STEGO significantly outperforms previous state-of-the-art models in semantic segmentation, achieving a +14 mIoU improvement for CocoStuff and a +9 mIoU improvement for Cityscapes. The architectural decisions of STEGO are supported by an ablation study on the CocoStuff dataset. The authors provide a concise overview of their primary findings pertaining to the 27 categories of CocoStuff.

The STEGO method outperforms the previous leading technique, PiCIE, by a significant margin [25], in terms of both linear probe and clustering metrics in unsupervised learning. The STEGO method shows significant improvements when compared to competing methods like PiCIE and DINO. Specifically, it achieves a +14 boost in unsupervised mean Intersection over Union (mIoU), a +6.9 increase

in unsupervised accuracy, a +26 improvement in linear probe mIoU, and a +21 increase in linear probe accuracy. Additionally, it demonstrates a substantial +8.7 enhancement in unsupervised mIoU and a +7.7 improvement in unsupervised accuracy on the Cityscapes validation dataset. The results of these two experiments illustrate that, despite the absence of fine-tuning the backbone for the datasets in question, the self-supervised weights of DINO on ImageNet [3] are sufficient to effectively address both scenarios simultaneously. Furthermore, STEGO outperforms plain feature clustering from unmodified DINO, MoCoV2, and ImageNet-trained ResNet50 backbones. This study highlights the benefits of including a segmentation component during training to improve feature matching.

2.2 Background of Unsupervised and Self-Supervised Learning

Mathilde et al. proposed a method called DeepCluser for convnets that works with any clustering algorithm like k-means and Power Iteration Clustering (PIC) [15]. They focused on k-means clustering where their approach requires minimal additional steps. Their method was to alternate between clustering of images and updating the weights of convents by predicting cluster assignments that get labeled as 'pseudo-labels' to optimize previous clustering. They show their work for training convents from scratch to image classification. They trained DeepCluster on a training set of ImageNet, and to measure the impact, they used the YFCC100M dataset [11], [12] for the pre-training. The model outperforms previous unsupervised methods on 3 tasks. For classification, it outperforms 73.7%, detection 55.4%, and segmentation 45.1%. The largest improvement was 7.5% over the state-of-the-art segmentation task. Also, DeepCluster [15] performs slightly better than other unsupervised methods in detecting.

Like Mathilde et al., many authors have combined clustering algorithms with deep learning. However, combining clustering with learning methods can often lead to debased solutions [15]. To work on this issue, Ji et al. introduce a new scalable clustering method, Invariant Information Clustering (IIC), for unsupervised learning of convents [21]. They performed their experiments on large datasets, i.e., STL, CIFAR, and COCO-Stuff, with results of 59.6%, 61.7%, and 72.3%, respectively, beating the closest competitors like ADC[19], DAC [13] (53.0%, 52.2%, 54.0%). The IIC model outperformed DeepCluster in unsupervised segmentation on the COCO-Stuff-3 dataset with 72.3%, whereas DeepCluster's result was 41.6%. Overall they outperformed all previous methods by 18.3% for the COCO-Stuff-3 dataset.

Cho et al. introduced an innovative framework for semantic segmentation, leveraging invariance and equivariance within clustering [25]. They introduced a technique that incorporates geometric consistency as an inherent bias, enabling the model to grasp both invariance and equivariance principles for photometric and geometric variations. Through their learning objective, their framework becomes proficient in

capturing high-level semantic concepts. Their approach, PiCIE (Pixel-level feature Clustering using Invariance and Equivariance) [25], stands out as the first method that can segment both 'stuffs' and 'things' categories without the need for hyperparameter tuning or task-specific pre-processing. PiCIE significantly outperforms current benchmarks on COCO [8] and Cityscapes [10], achieving a remarkable +17.5% increase in accuracy and a +4.5% improvement in mean Intersection Over Union (mIoU). They showed that PiCIE gives a good standard of training and better performance.

Shanghua et al. introduce new challenges in the domain of sizeable unsupervised learning [36]. Their objective was to build up the performance of semantic segmentation in real-world settings by manipulating a wide range of diverse and extensive data. This study suggests a benchmark for Large-Scale Unstructured Semantic Search (LUSS) that deals with a wide range of data showcases significant diversity, defines a well-defined task objective, and includes a broad evaluation framework. Moreover, the research states that they introduced a great approach to Labeling Unsupervised Semantic Segmentation (LUSS) [36]. They label pixels by learning category and shape features from a big dataset, all without needing human annotations. Their method uses improved learning and pixel-level labeling with pixel attention. The method is evaluated in their study. They used different testing methods to see how well LUSS performs in pixel-level tasks like semantic segmentation. They also test it with unsupervised learning and partially supervised segmentation techniques. Again, they provide an overview of the obstacles and potential avenues for future research in the field of LUSS. The observed improvements in performance attained by their method, compared to SwAV and PixelPro, are respectively 1.7% and 1.2% in mean average precision (mAP) for object detection.

The paper IIC [21] by Ji et al., was based on Mutual Information, restricting the prediction field to only patches instead of the entire image which might affect the clustering. Robert Harb and Patrick KnÖbelreiter recognized this limitation and introduced a model called InfoSeg, which incorporates global high-level features across the entire image [25]. To suppress local noise and encourage to encoding of high-level information, they use Local Deep InfoMax. From their dataset, they take the input and send it for feature representation then they compute the MIV. Afterward, they choose each spatial position of the MIV and pass it to the segmentation. To maximize the MIV they perform Mutual Information Neural Estimation (MINE) to maximize the lower bounds parametrized by Deep Neural Networks. Robert Harb and Patrick KnÖbelreiter provide a quantitative comparison with their method and a few others, i.e., IIC, Isola, InMARS, and K-Means with COCO-Stuff, Potsdam datasets [25]. Their model InfoSeg outperformed the mentioned methods, resulting in COCO-Stuff-3 and Potsdam-3, 73.8 and 71.6 Pixel-Accuracy (PA), respectively.

Seong et al. proposed a method, Contrastive learning by discovering hidden positives that learn pixel-level semantic clusters without limitations [44]. They determine their method to be free of limitation that relies solely on a predetermined

backbone, mostly limited, not for segmentation tasks. Their method ensures contextual consistency along the patches with the same semantics. They train their model with contrastive learning with two types of hidden positives: i) global (discovered from samples in mini-batch) and ii) local (discovered from subsets of surrounding patches). In contrast to existing state-of-the-art methods, their model outperformed. The results were compared among COCO-Stuff, Cityscapes, and Potsdam-3 datasets. Hidden Positives (HP) improved over previous models in almost all cases with a 56.1% Accuracy and 23.2% mean Intersection Over Union (mIoU) in unsupervised COCO-Stuff data. In the unsupervised Potsdam-3 dataset, their model HP gave an accuracy of 82.4% [44].

On the other hand, Chieh Chen et al. took a different approach, using DeepLabv2, and made three key contributions with significant practical benefits [14]. First, they employed unsampled filters, also known as atrous convolution, as a potent technique for dense prediction tasks. This technique enabled them to expand the filters' field of view, capturing a broader context without adding more parameters or computational load. They proposed a method called atrous spatial pyramid pooling (ASPP) to better detect objects of different sizes. ASPP uses filters at various sampling rates and viewing angles, helping to capture both objects and the bigger picture in the image. Finally, by combining the method of DCNN and probabilistic graphical models, they improved the localization of the object boundaries. The combination of downsampling and max pooling in the DCNNs archive has gained invariance but it created a problem with localization accuracy. They overcame this problem by combining the responses from the final layer of the deep convolutional neural network (DCNN) with a fully connected Conditional Random Field (FCCRF). This approach was demonstrated to enhance localization performance both qualitatively and quantitatively. They introduced the 'DeepLab' dataset during the PASCAL VOC-2012 semantic image segmentation task, achieving a 79.7% mean intersection over union (mIOU) on the test set. They also improved results on three other datasets: PASCAL-Context [6], PASCAL-Person-Part [7], and Cityscapes [10].

A paper by M. Schmitt et al. utilizes the SEN12MS dataset and data from the IEEE-GRSS 2020 Data Fusion Contest to address the challenge of developing semantic segmentation models for global land cover mapping, even in the presence of imperfect and imprecise labels [24]. While standard shallow and deep learning approaches have shown promising mapping capabilities, the current results fall short of being deemed suitable for practical, off-the-shelf solutions. Therefore, they assert that the development of particular models within the world of weakly supervised machine learning is imperative. They predict that these models will significantly enhance the efficacy of a comprehensive and entirely automated satellite-based system for monitoring global land cover. It is mentionable that three classes, namely Shrublands, Wetlands, and Barren, consistently show poor metrics across all classification methods. Among the classes in the SEN12MS dataset, these three are the least frequent, with the exception of the understandably rare Snow/Ice class. In contrast, it is observed that the DFC2020 validation set shows a significant overrepresentation of Wetlands.

Eliasof et al. proposed an innovative approach that harnesses recent developments in unsupervised learning by integrating Mutual Information Maximization (MIM), Neural Superpixel Segmentation, and Graph Neural Networks (GNNs) into an end-to-end framework [30], [31]. In order to learn semantically meaningful image restoration, they combined compact representations of superpixels and GNNs. They demonstrated that enhancing their GNN-based approach enabled the model to capture interactions between distant pixels in the image, serving as a robust prior compared to existing CNNs. When comparing their approach to current methods across four popular datasets, their experiments exhibit both qualitative and quantitative advantages. They immediately saw an accuracy improvement of 4.6%, showing the significance of superpixel information. Finally, by considering the full model includes both the SPNN and GNN components, a further accuracy gain of 2.6% is obtained.

In Chaurasia et al. paper, they proposed a novel deep neural network architecture known as ENet (efficient neural network) [31], which is designed for tasks that require quick processing. ENet is up to 18x faster, needs 75x fewer FLOPs for less computation, has 7x fewer parameters, and provides similar or 9x better results to existing methods. They conducted tests using two datasets, namely CamVid and Cityscapes [10], as well as SUN datasets. They compared their approach with current state-of-the-art methods, weighing the trade-offs between network accuracy and processing speed. The research also includes performance measurements of their architecture on embedded systems and offers recommendations for potential software enhancements to further improve the speed of ENet [31]. It was built to run faster achieving over 10 frames per second (fps) on the NVIDIA TX1 board using an input image size of 640x360, making it suitable for real-world road scene parsing analysis. This can prove its usefulness in data-center applications when especially processing with a large set of high quality images.

Liu et al. propose a method for semantic image segmentation that integrates a variety of information into the Markov Random Field (MRF) [9], including high-order relationships and a mix of label contexts. They addressed the MRF problem by introducing the Deep Parsing Network (DPN), a Convolutional Neural Network (CNN). This innovation allows for deterministic end-to-end computation in a single forward pass, departing from earlier methods that relied on iterative algorithms for optimizing MRFs. DPN builds upon a modern CNN architecture to handle Unary terms and incorporates carefully designed additional layers to approximate the mean field algorithm for pairwise terms, yielding several advantageous properties. Unlike recent approaches that merge MRF and CNN, which demand numerous MF algorithm iterations during backpropagation for each training image, DPN stands out by delivering improved performance with just one MF algorithm iteration [9]. Furthermore, DPN encompasses various pairwise term representations, encompassing many prior works as specific instances. Finally, DPN simplifies the parallelization of MF, resulting in improved speed on a Graphical Processing Unit (GPU). We exten-

sively tested DPN using the PASCAL VOC 2012 dataset, achieving a segmentation accuracy of 77.5% with just one DPN model [9].

In the paper of Chevitarese et al., they introduced a deep neural network architecture designed for segmentation, demonstrating promising results on seismic data. This architecture builds upon the foundation of existing work on Fully Convolutional Networks (FCNs) [18]. In their paper, they introduced a newly discovered deep neural network architecture modified for the semantic segmentation of seismic images, requiring minimal training data. To achieve this, they innovatively employed a transposed residual unit in place of the conventional dilated convolution for the decode block. Instead of relying on predefined shapes for upscaling, their network learns the entire process of feature upscaling from the encoder. They conducted training using the Penobscot 3D dataset, an authentic seismic dataset acquired off the coast of Nova Scotia, Canada. Their approach was benchmarked against two established deep neural network architectures: the Fully Convolutional Network and U-Net. In their conducted experiments, it is demonstrated that their method consistently attains a mean intersection over Union (mIoU) metric of over 99%, outperforming existing models. Furthermore, the qualitative results indicate that the model produces masks closely resembling human interpretation with minimal discontinuity. By conducting a comparative analysis of two tables, it was determined that Danet-FCN2 emerges as the model striking the optimal equilibrium between performance and efficiency. This model boasts the least number of operations and nearly five times fewer parameters than Unet, all while achieving mIOU values surpassing 89% across all scenarios with 80 x 120 tiles.

According to Li et al., They reached higher segmentation performance without retraining and achieved performance on the PASCAL VOC 2012 dataset, highlighting the excellence of ACSeg through its adaptive conceptualization approach [38]. In line with this, they introduced ACSeg, which can be seen as a transition from self-supervised image-level models to dense prediction tasks. This approach leverages the pre-trained models' extracted representations while also acquiring new ones. It is seen that on PASCAL VOC, COCO, dataset ACG outperforms IIC [21], PiCIE [25], ImageNet, etc. by 47.1 and 16.4 in mIoU. Also in the case of speed, comparing it with k-means, spectral, AP, Agglomerative, ACG performs better which is by 149.2 per second.

Chapter 3

Segmentation Techniques and Tools

The first stepping stone of our whole research project is to get familiar with the concept of Segmentation techniques. This will also help us to be more certain about the importance of our research. To do that, we are going through related methods, specifically on unsupervised semantic segmentation, along with methods that are similar to our field of research. While studying the approaches, we summarize the methods and models that can be used for image segmentation.

There are many methods for unsupervised semantic segmentation through which image classification and segmentation can be done. Some basis models are vanilla CNN (convolution neural network) (Section 3.1), Inception (Section 3.2), and K-means clustering (Section 3.3). Along with some Vision architectures like LLaVa (Large Language and Vision Assistant) (Section 3.4), LISA (Large Language Instructed Segmentation Assistant) (Section 3.5), and DeepAqua (Section 3.6).

3.1 Convolutional Neural network (CNN)

Convolutional Neural network processes structured grid data, it is considered as a deep learning model. It gives effective results in computer vision tasks, image segmentation, and object detection. It automatically and adaptively learns high-level representations of the visual data. There are many key components of CNN, which include convolutional layers, fully connected layers, and pooling layers. It acts as the fundamental building block of the CNN.

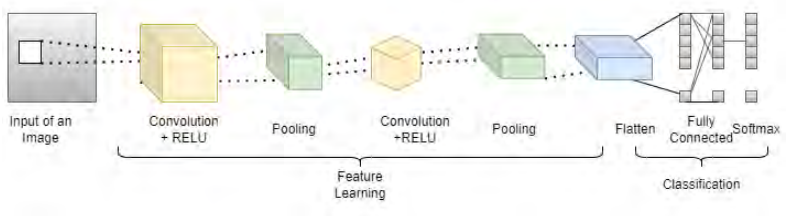


Figure 3.1: Convolutional Neural Network (CNN)

A layer of convolution operations is applied to input data using kernels or filters,

which helps produce a feature map. It captures spatial hierarchies in the data. It is used for object detection, medical image analysis, facial recognition, etc.

Another component is the Activation function, which is usually applied after the convolution operations. A nonlinearity is introduced to the network. One of the activation functions is ReLu (including Rectified Linear Unit). It replaces negative values with zeros, which helps the network to learn complex relationships among the data. The pooling layer is also another component that downsamples the dimension of spatial in feature maps. As a result, it enhances the translation invariance of the network. Commonly used methods are max pooling and average pooling. Another component is fully connected layers (Figure 3.2). By connecting every neuron to another neuron of the next layer it gives us a final prediction. Previous layers extract the high-level features.

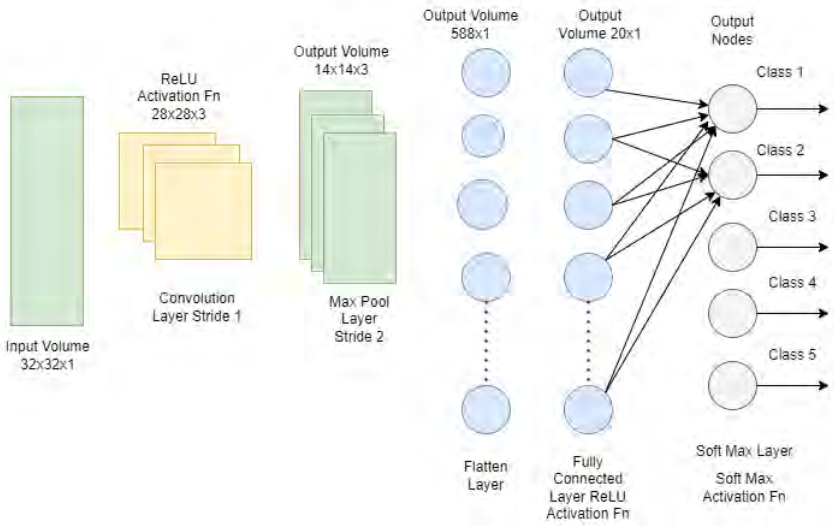


Figure 3.2: Fully Connected Layer (FCNN)

Then, the feature maps are flattened into one dimension vector. It helps in formatting the neural network suitable for traditional. The output feature map $\mathbf{F}$ of a convolutional layer can be expressed as:

$$\mathbf{F}_{ij}^{(k)} = \sigma\left(\sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{c=1}^{C}\mathbf{W}_{mn}^{(k,c)}\mathbf{X}_{i+m-1,j+n-1}^{(c)} + b^{(k)}\right) \tag{3.1}$$

Where:

- $\mathbf{F}_{ij}^{(k)}$ is the value of the k-th feature map at position (i, j).
- σ is the activation function (e.g., ReLU).
- M and N are the height and width of the convolution kernel.
- C is the number of input channels.
- $\mathbf{W}_{mn}^{(k,c)}$ is the weight of the kernel at position (m, n) for the c-th channel and k-th feature map.

- $\mathbf{X}^{(c)}_{i+m-1,j+n-1}$ is the value of the input at position $(i+m-1, j+n-1)$ for the c-th channel.
- $b^{(k)}$ is the bias term for the k-th feature map.

The CNNs use backpropagation which helps in training the dataset using labels. It adjusts the parameter to lessen the comparison between predicted and actual results. This works like a stochastic gradient descent which helps in optimizing a loss function. CNN gives an exceptional performance in a huge range of fields. It makes them well suited for the tasks because of their ability.

3.2 InceptionV3

Inception is a deep CNN architecture, which is also known as GoogLeNet, which is developed by Google researchers. This model took inspiration from V2, V3, and V4 of the Deep Learning method. It allows the capture of multi-scale features in the network efficiently. They also have some key features, including,

1. `Inception module:` This module incorporates filters that differ in sizes (1x1, 3x3, 5x5). It also pools operation continuously. It gives us a more effective representation of input data.

2. `1x1 convolution:` This network uses 1x1 convolutions (Figure 3.3). On the other hand, spatial features are captured by 5x5 and 3x3 convolutions. This plays an important role in reducing dimensionality and aggregation of channel-wise features and also helps in parameter reduction.

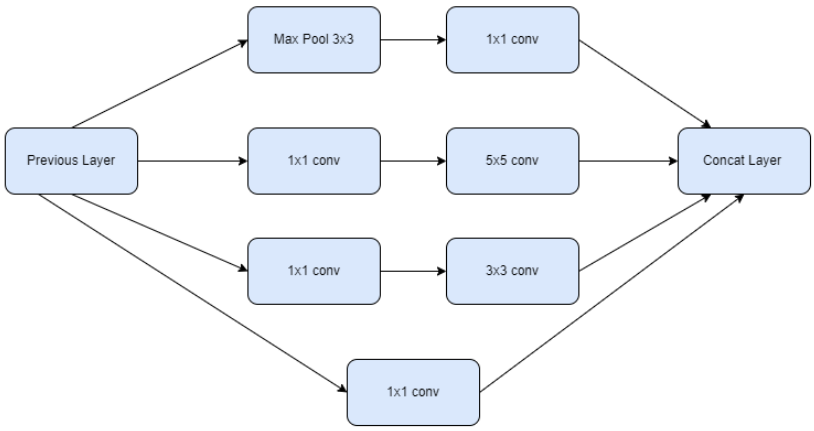


Figure 3.3: 1x1 Convolution

3. `Global Average pooling:` Inception implies global average pooling instead of fully connected layers at the end of the network. It helps in averaging the feature map also helps in dimension reduction, also helps in reducing overfitting, which is very effective.

$$\text{GAP}(X) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X_{ijk} \tag{3.2}$$

where:

- X is the input tensor with dimensions $H \times W \times C$.
- H and W are the height and width of the input tensor.
- X_{ijk} is the value at position (i, j) in the k-th channel.

4. `Auxiliary Classifiers:` This works in the intermediate layers of training data. It helps in vanishing the gradient problem. It also adds additional supervision to the lower layers.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{main}} + \alpha \mathcal{L}_{\text{aux}} \tag{3.3}$$

Where:

- $\mathcal{L}_{\text{main}}$ is the loss from the main classifier.
- $\mathcal{L}_{\text{aux}}$ is the loss from the auxiliary classifier.
- α is a weight factor that determines the contribution of the auxiliary loss.

5. `Batch normalization:` This helps to accelerate training and also helps in stabilizing by normalizing. It normalizes the input values in mini-batches. This results in converging faster.

$$\hat{x}_i = \frac{x_i - \mu_{\text{batch}}}{\sqrt{\sigma^2_{\text{batch}} + \epsilon}} \tag{3.4}$$

$$y_i = \gamma \hat{x}_i + \beta \tag{3.5}$$

Where:

- x_i $\leftarrow$input value.
- μ_{batch} $\rightarrow$mean of the batch.
- σ^2_{batch} $\rightarrow$variance of the batch.
- ϵ $\rightarrow$a small constant added for numerical stability.
- γ and β $\rightarrow$learnable parameters that scale and shift the normalized value.

3.3 K-means

K-means algorithms are used in clustering to partition a dataset into distinct, non-overlapping subsets. It groups data of similar data points and then comprises distinct clusters. It represents a centroid. The algorithm follows a straightforward iterative process: Initialization, Assignment, Update, Repeat. It first uses the initialization technique. Then, each data is assigned to a cluster that has the closest centroid distance. Mainly, the Euclidean distance is used. After this assigning the algorithm updates the centroid.

The centroid is updated by using by calculating the mean of data points that are present in the cluster. The steps are repeated until convergence. It iterates until it gets a suitable clustering solution. Then, it determines the optimal numbers of clusters by using some methods, for example, The elbow method, Silhouette score,

etc.

The algorithm aims to minimize the within-cluster sum of squares, which is the sum of the squared distances between each data point and its assigned centroid. This objective can be mathematically expressed as:

$$j = \arg\min_{S} \sum_{i=1}^{k} \sum_{x \in S_i} \|x_i^j - \mu_i\|^2 \tag{3.6}$$

Where:

- j →objective,
- k →the number of clusters,
- S_i →set of points in cluster i,
- x →a data point,
- μ_i →the centroid of cluster i.

K-means is good for scalability and efficiency as it is suitable for a large number of datasets. This algorithm runs multiple times using different initializations and then chooses the best result. So we can say that k means gives us effective results.

3.4 LLaVa: Large Language and Vision Assistant

LLaVa [39] is an AI system that combines the strengths of computer vision and natural language processing (NLP). As an advanced multimodal model, LLAVA excels at understanding images and answering questions about them. This end-to-end trained Large Multimodal Model (LMM) processes both text prompts and images containing rules or instructions. This integration allows LLaVa to perform complex visual reasoning tasks, making it a robust tool for applications requiring both visual and textual understanding.

LLaVa combines the CLIP [29] visual encoder with the Vicuna chatbot to form an end-to-end multimodal pipeline that yields state-of-the-art results. Here, the network architecture utilizes the strengths of a pre-trained large language model (LLM) and the visual Vicuna model as the LLM because of its exceptional ability to follow instructions. This integration not only boosts the system's ability to interpret visual and textual information but also guarantees that it delivers precise and relevant responses to user queries.

The CLIP (Contrastive Language-Image Pre-Training) (Figure 3.4) model bridges the gap between visual and textual understanding, leveraging large-scale datasets of images and their corresponding textual descriptions to learn a unified representation of both modalities. Unlike traditional models that require extensive labeled data, CLIP is trained using a contrastive learning technique where it learns to match images with their corresponding textual descriptions.

This innovative model can understand and generate descriptions for a wide variety of images by aligning the features of images and text in a shared latent space. The training process maximizes the similarity between the correct image-text pairs and minimizes the similarity between incorrect pairs. As a result, CLIP excels at zero-shot learning, meaning it can recognize and categorize new images without needing specific training for each new task.

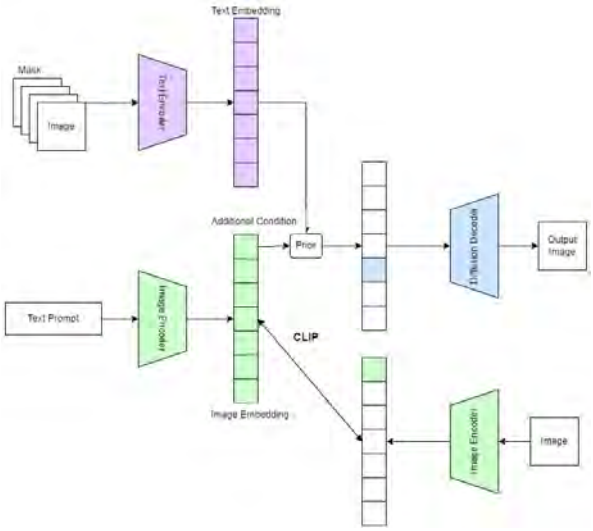


Figure 3.4: CLIP Approach

Vicuna chatbot's one of the key features is its ability to follow complex instructions and engage in meaningful dialogue, which is driven by its underlying architecture incorporating powerful language models that have been fine-tuned on vast amounts of conversational data. As a result, Vicuna (Figure 3.5) can handle nuanced queries, provide detailed explanations, and even exhibit a degree of contextual awareness that makes interactions more fluid and engaging.

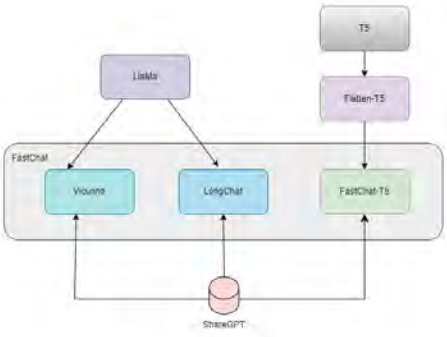


Figure 3.5: Vicuna Approach

The integration of the Vicuna chatbot within the LLaVa system exemplifies its strengths. By combining the CLIP model's visual understanding capabilities with Vicuna's conversational prowess, the system can interpret visual cues and respond to related questions effectively. This synergy not only enhances the user experience but also ensures that responses are both accurate and contextually relevant.

The Multimodal Instruction Following the Data Creation process enhances the model's ability to follow complex instructions across different modalities. The data used to train LLaVa is based on three different types of instructions:
1. A brief description of the image content; 2. A long, detailed explanation of the image, and 3. Logical reasoning about the image content.
The model architecture involves users interacting by inputting text prompts and images. The language model tokenizer processes the text prompt, while the vision encoder tokenizes the image.

For input images visual features extraction, the pre-trained CLIP visual encoder ViT-L14 is used, which is then converted into language embedding tokens through a fine-tuned trainable projection matrix (W).

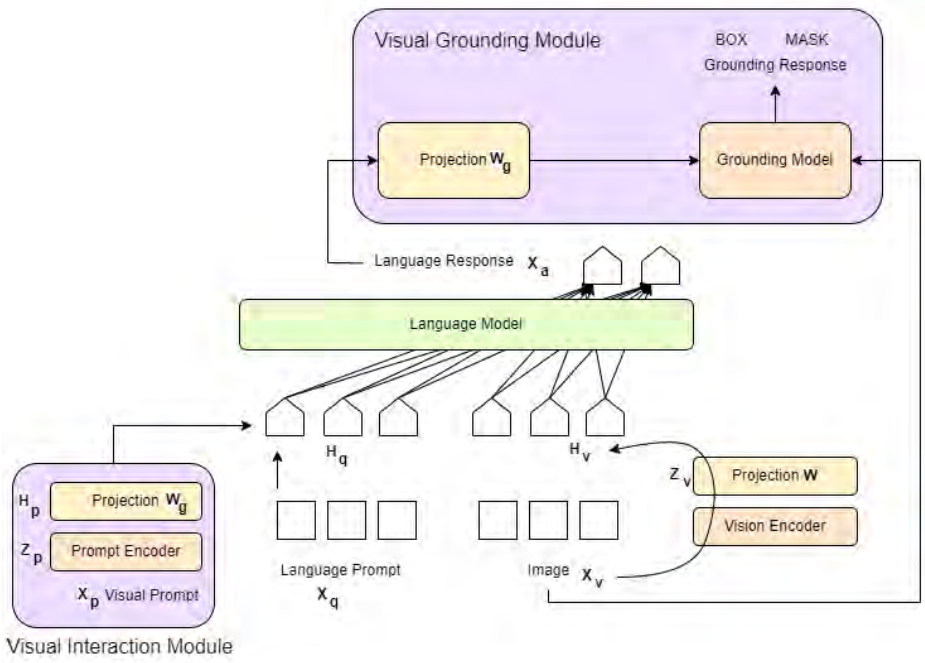


Figure 3.6: Network Architecture of LLaVA

Training the model involves generating multi-turn conversations for each input image, including responses and a series of prompts. Initially, both vision and text inputs are contained in the first prompt, followed by text inputs in subsequent prompts, training the model to respond to text prompts about the image. A simple linear layer is used to connect image features to word embedding space that allows for quick data-centric experiments.

3.5 LISA: Large Language Instructed Segmentation Assistant

To overcome some of the LLaVa challenges, a Reasoning Segmentation via Large Language Model LISA [47], built an approach to semantic segmentation that uses the capabilities of multi-modal large language models (LLMs) (Figure 3.7) to under-

stand and analyze visual scenes which allows to segment objects in an image based on instructions given in natural language.

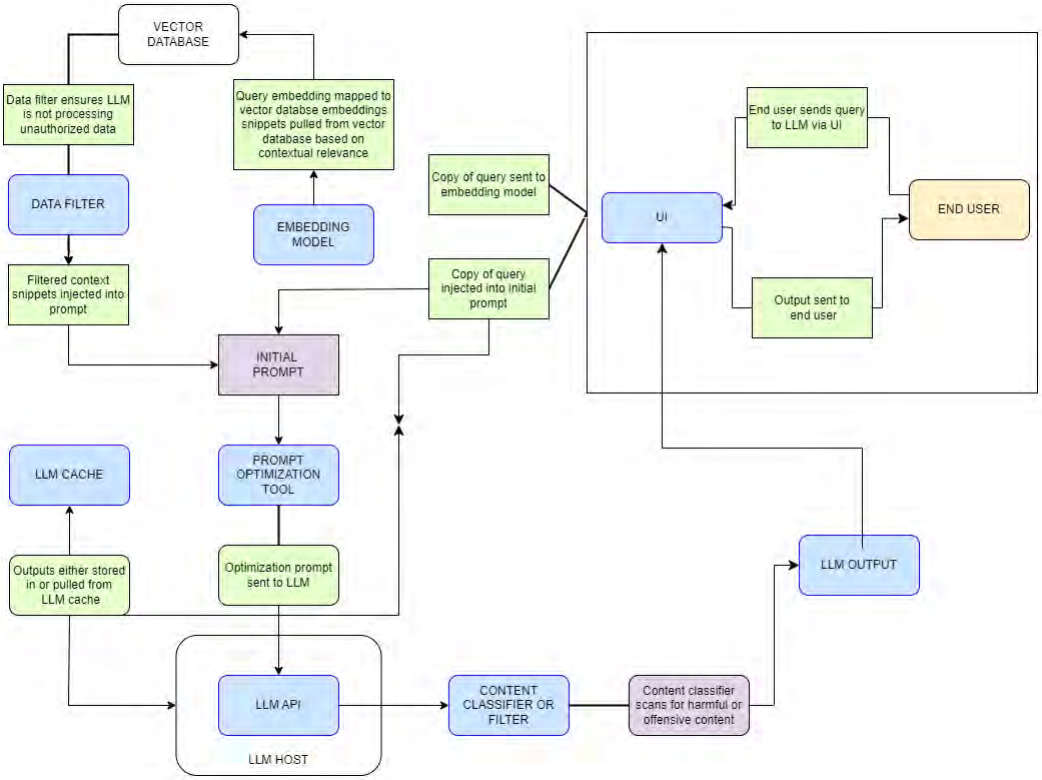


Figure 3.7: LLM Application Diagram

The model performs a reasoning segmentation task that involves understanding and interpreting implicit human instructions. It shows strong zero-shot performance on the reasoning segmentation task, even when trained exclusively on datasets that do not involve reasoning. LISA follows a two-stage process: (1) visual feature extraction and (2) language-guided reasoning and segmentation.

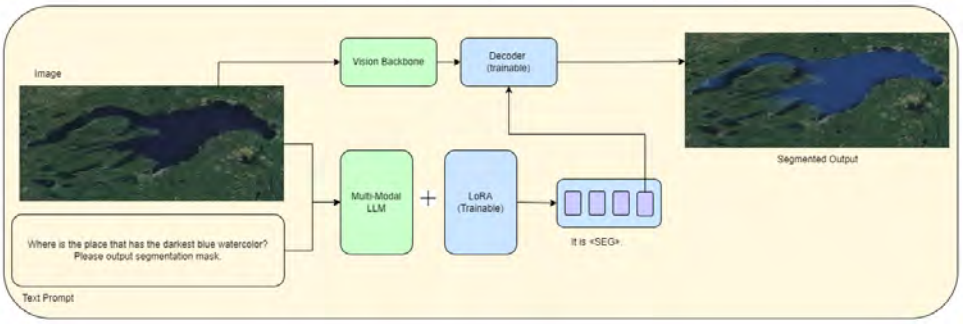


Figure 3.8: LISA Architecture

The model takes both an image and user input texts as inputs. The image is processed through the Vision Backbone model and a Multi-modal Large Language Model (LLM). Simultaneously, the text input is processed through the same Multi-

modal LLM, enhanced with LoRA (Low-Rank Adaptation of Large Language Models).

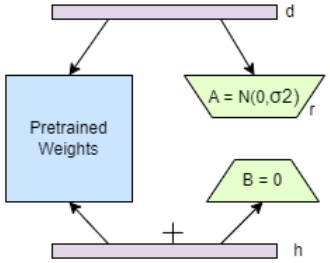


Figure 3.9: LoRA Reparametrization [28]

The outputs from the Vision encoder and the LLM are then combined and used to decode and generate a segmented image.

3.6 DeepAqua: Self-Supervised Semantic Segmentation of Wetland Surface

The framework of DeepAqua [43] operates using a dual model having a teacher model and a student model where the teacher model, knowledge distillation architecture, functions as a thresholding model, and the student model employs a U-Net architecture.

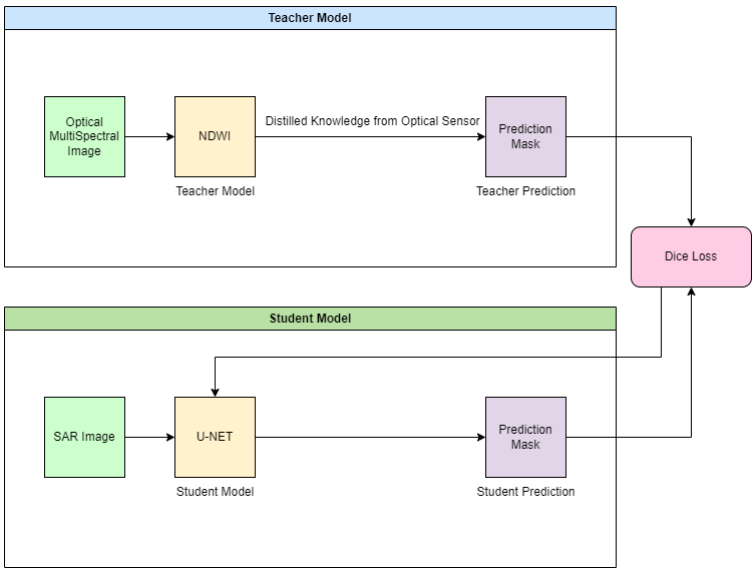


Figure 3.10: DeepAqua Model Architecture

The teacher model creates water masks from optical images using the Normalized Difference Water Index (NDWI), while the student model generates segmentation masks from Synthetic Aperture Radar (SAR) images.

Both models are jointly trained by minimizing the Dice loss between their outputs. They utilize different data types: the teacher model extracts water surface information from optical images to produce segmented images, which the student model then tries to replicate using radar imagery.

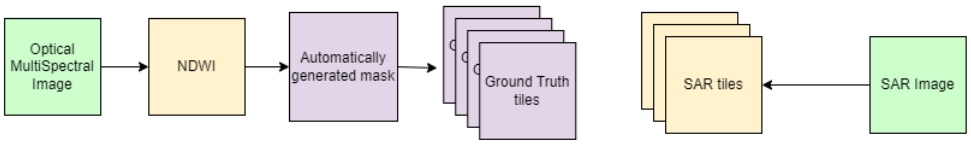


Figure 3.11: Training Process

The core of the model is the cross-modal knowledge distillation process, where knowledge, the detailed NDWI water mask from the teacher model, is transferred to the student model to create a segmentation mask. For backpropagation, Dice Loss is calculated to update the weights of the student model based on the results while training. The goal of the cross-modal knowledge distillation is to minimize the Dice loss between the NDWI mask and the segmentation mask which ensures that both of the masks closely resemble each other.

It is important to note that advanced tools such as the Segment Anything Model (SAM) and the Recognize Anything Model (RAM) have demonstrated superior performance in various applications. Recognizing their potential, we have integrated these models into our workflow. The detailed process of incorporating these and their specific contributions to our research are thoroughly explained in the methodology section that follows.

Chapter 4

Methodology

In the following section, we present a detailed top-level view block diagram that outlines the workflow of our Wetland fluctuation Localization research, which provides a visual representation of each step involved in the process, from data pre-processing to analysis and final evaluation.

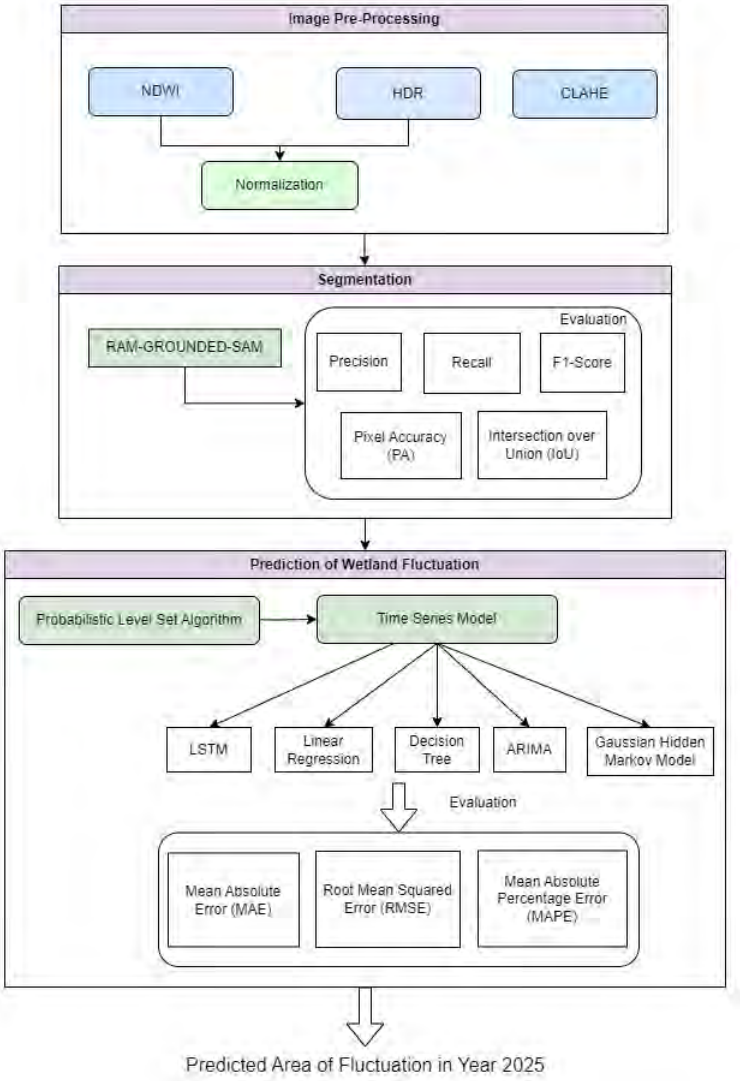
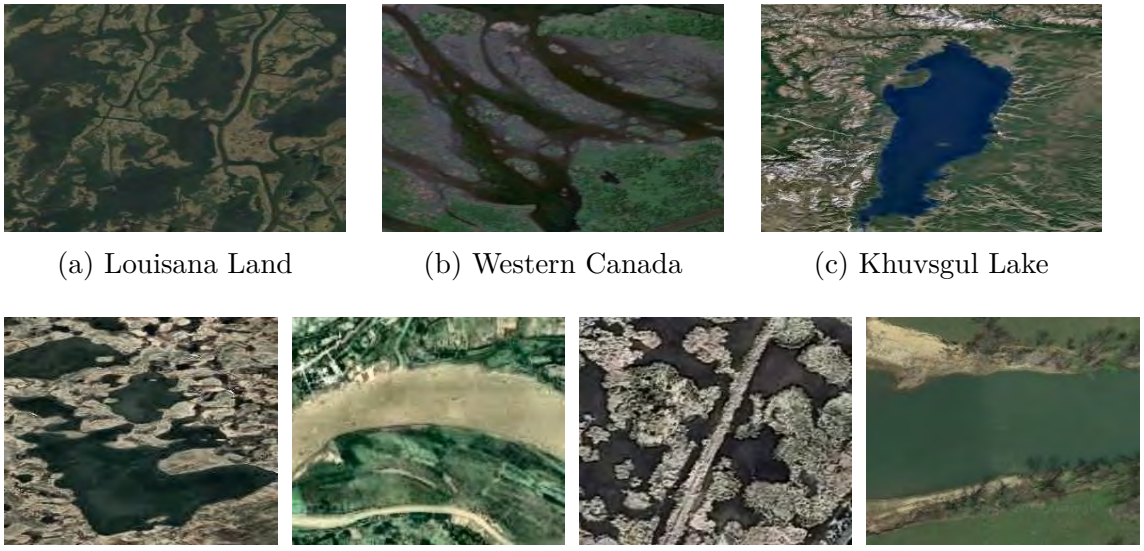


Figure 4.1: Top Level Block Layout of Proposed RAM-Grounded-SAM

The Top Level Block Layout (Figure 4.1) outlines the detailed process for predicting wetland area fluctuations by 2025. Starting with image pre-processing using NDWI, HDR, and CLAHE techniques for enhancing quality, followed by normalization to maintain dataset consistency. Next, the enhanced images undergo segmentation with the RAM-GROUNDED-SAM method, which is then evaluated through metrics like Precision, Recall, F1-Score, Pixel Accuracy (PA), and Intersection over Union (IoU). The segmentation results are used in a Probabilistic Level Set Algorithm, applying various time series models and assessed using error metrics, and these models produce the final output predicting wetland area fluctuations for 2025.

4.1 Dataset Description

We are using **MLRSNet** Dataset to localize wetlands as our primary source, which is publically available in Mendeley Data [22]. This dataset is composed of high spatial resolution optical satellite images comprising more than 100000 remote sensing images. The images have a fixed size of 256×256 pixels, which has various pixel resolutions (10m to 0.1m). These are the images we are using for image segmentation.



(a) Louisana Land (b) Western Canada (c) Khuvsgul Lake

(d) Sen River (e) Ganga (f) Nile Lake (g) Haigmill Lake

Figure 4.2: MLRSNet Dataset of Satellite Wetland Images. Figures (a), (b), (c), (d), (e), (f), and (g) illustrate samples from the MLRSNet dataset, which we use to apply pre-processing techniques and localize wetlands.

Additionally, for our research, we collected our dataset (Figure 4.3) by extracting image data from Google Earth [49], explicitly focusing on the wetland areas in Bangladesh. Our dataset includes images from various haors, such as Tanguar Haor, Hakaluki Haor, Hail Haor, and other significant wetland regions (Figure 4.3). The images range from 15 meters per pixel for older and broader coverage areas to 30 centimeters per pixel for more recent and detailed satellite images. High-resolution satellite images, such as those from the WorldView-3 satellite, have resolutions as satisfactory as 31 centimeters per pixel.

(a) Hakaluki Haor, Sylhet (b) Dekhar Haor, Gazinagar (c) Korchar Haor, Bishwamvarpur (d) Tanguar Haor, Sunamganj

Figure 4.3: Hoars of different districts in Bangladesh

These datasets are not readily available elsewhere, underscoring our work's uniqueness and value. By creating a custom dataset, we have made a major contribution to ecological and environmental research. Utilizing the timelapse feature of Google Earth, we compiled a comprehensive dataset spanning the years 1983 to today. This extensive timeline enables us to analyze long-term changes and patterns in these critical wetland ecosystems, providing insights crucial for understanding their dynamics and informing conservation efforts.

This meticulous data collection effort has allowed us to capture a diverse and representative sample of these vital ecosystems. The ability to observe and analyze over four decades of environmental changes offers an unprecedented perspective that is vital for addressing contemporary environmental challenges.

Our dataset, therefore, fills a significant gap in the availability of long-term ecological data for Bangladesh's wetlands, and it serves as a valuable resource for researchers, policymakers, and conservationists who are working to protect these essential habitats. This pioneering effort enhances our understanding of wetland dynamics and sets a foundation for future research and informed decision-making in wetland management and conservation.

4.2 Dataset Pre-processing

With our unsupervised satellite images, we saw that a few images from MLRSNet were a bit white-washed and blurry; hence, to handle these, we experimented with some pre-processing techniques before feeding them to our model. We use different methods to test out which one of the pre-processing techniques will work better for our case.

4.2.1 Histogram Equalization

First, we applied a histogram. We used sharpening, Adaptive Histogram Equalization (AHE), and Contrast Limited Adaptive Histogram Equalization (CLAHE). But we saw that for our case, CLAHE (Figure 4.4) is better as it does not destroy the resolution and gives us more details in the image. It is an effective tool for improving the visibility of details in such images.

The 256x256 images are first divided into small blocks of 8x8 tiles with a batch size of 32 using OpenCV2, and each of the tiles is histogram equalized with an added contrast limit of 2.0. When any tile histogram bin crosses the contrast limit, also called clip limit=2.0, OpenCV2 clips those pixels and distributes them uniformly to other bins, and then performs Histogram equalization. A clipping limiting factor is applied as it maintains a more neutral appearance. Then, bilinear interpolation is performed to remove any artifacts found in tile borders. The interpolation methods are used to smooth out these transitions.

CLAHE operates by applying histogram equalization to contextual regions in the image. The contrast in each region is enhanced so that the histogram of the output region approximately matches the histogram specified by the contrast limit. The process is given by:

$$p_{output}(i,j) = T(p_{input}(i,j)) = \frac{\text{CDF}(p_{input}(i,j)) - \text{CDF}_{min}}{1 - \text{CDF}_{min}} \cdot (L-1) \tag{4.1}$$

Where:

- $p_{output}(i,j)$ is the output pixel intensity at location (i,j),
- $p_{input}(i,j)$ is the input pixel intensity at location (i,j),
- T is the transformation function,
- $\text{CDF}(p_{input}(i,j))$ is the cumulative distribution function of the input pixel intensity within the local region,
- CDF_{min} is the minimum non-zero value of the cumulative distribution function,
- L is the number of possible intensity levels in the image.

As for the limiting factor, the amplification by clipping the histogram at a predefined value (clip limit) redistributes the excess pixels uniformly across the histogram bins.

The clipped histogram H_{clip} is computed as:

$$H_{clip}(k) = \begin{cases} H(k) & \text{if } H(k) \leq \text{ClipLimit} \\ \text{ClipLimit} & \text{if } H(k) > \text{ClipLimit} \end{cases} \tag{4.2}$$

Where:

- $H(k)$ is the histogram value for bin k,
- ClipLimit=2.0 is the maximum allowed value for the histogram bins.

Then the excess pixels clipped are redistributed equally among all histogram bins:

$$H_{redistribute}(k) = H_{clip}(k) + \frac{\text{TotalExcess}}{N} \tag{4.3}$$

Where:

- $H_{redistribute}(k)$ →the redistributed histogram value for bin k,
- TotalExcess →the total number of clipped pixels,
- N →number of histogram bins.

This is how we implemented the CLAHE for our images,

Algorithm 1 CLAHE Implementation

Require: *image*
Ensure: *image_clahe*
  procedure APPLY_CLAHE_COLOR(*image*, *clip_limit*, *grid_size*)
    lab_image ← convert_color(*image*, BGR_TO_LAB)
    (*l_channel*, *a_channel*, *b_channel*) ← split_channels(*lab_image*)
    clahe ← create_clahe(*clipLimit*, *tileGridSize*)
    l_channel_clahe ← apply_clahe(*clahe*, *l_channel*)
    lab_image_clahe ← merge_channels(*l_channel_clahe*, *a_channel*, *b_channel*)
    image_clahe ← convert_color(*lab_image_clahe*, LAB_TO_BGR)
    return *image_clahe*

Corresponding images after using CLAHE,

(a) Louisana Land (b) Western Canada (c) Khuvsgul Lake

(d) Sen River (e) Ganga (f) Nile Lake (g) Haigmill Lake

Figure 4.4: Images after applying CLAHE. Figures (a), (b), (c), (d), (e), (f), and (g) are the outputs after using CLAHE; we can see that it enhances the contrast of images, which helps us to differentiate between different features and areas among wetlands images of MLRSNet data.

4.2.2 Normalized Difference Water Index (NDWI)

The NDWI (Normalized Difference Water Index) is a tool that is used to track changes in water content in lakes, rivers, reservoirs, and other water bodies.

Water bodies absorb a lot of light, especially in the visible and infrared parts of the spectrum; hence, NDWI leverages this characteristic by using data from the green and near-infrared (NIR) bands of satellite images to make water bodies stand out more clearly, which helps to easily identifying and monitoring water bodies in satellite images. NDWI values usually range from -1 to 1. Here, positive values indicate water bodies, while negative values represent non-water areas like soil and vegetation.

$$NDWI = \frac{(Band)}{(Band + NIR)} \tag{4.4}$$

where,

- Band is Green Band of Wetland Image,
- NIR is the Near Infrared band.

For our unsupervised images, first of all, green and NIR bands are read. Then, we converted the Green channel and NIR bands to float32 when it performs arithmetic operations to ensure precise calculation. Afterward, NDWI (Figure 4.6) is computed using error handling. NDWI values were normalized to the range [0,255] for visualization.

Algorithm 2 Computing NDWI

Require: $green\_band, nir\_band$
Ensure: $ndwi\_contrast$
  procedure COMPUTE_NDWI($green\_band, nir\_band$)
    $ndwi \leftarrow$ equation
    $ndwi\_normalized \leftarrow ((ndwi + 1) \times 127.5).astype(np.uint8)$
    $ndwi\_contrast \leftarrow$ cv2.normalize($ndwi\_normalized$, None, $alpha = 0, beta = 255, norm\_type, dtype$)
    return $ndwi\_contrast$

Original Images,

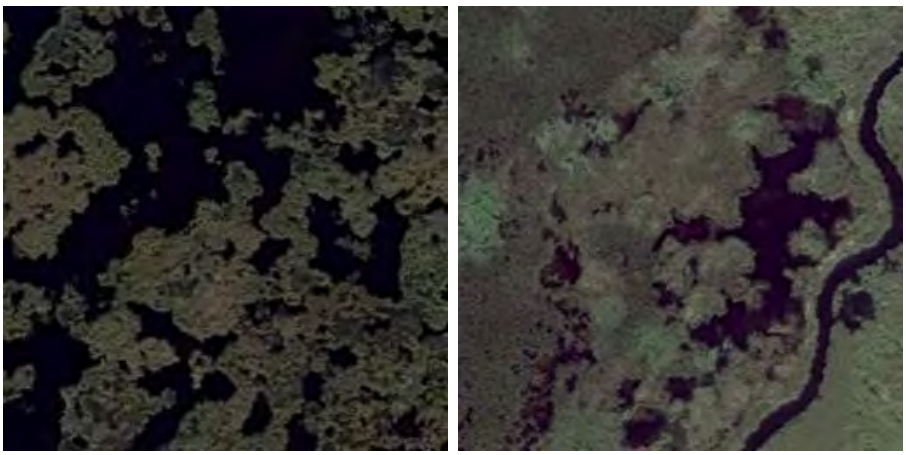


Figure 4.5: Images before using NDWI

Corresponding output after NDWI,

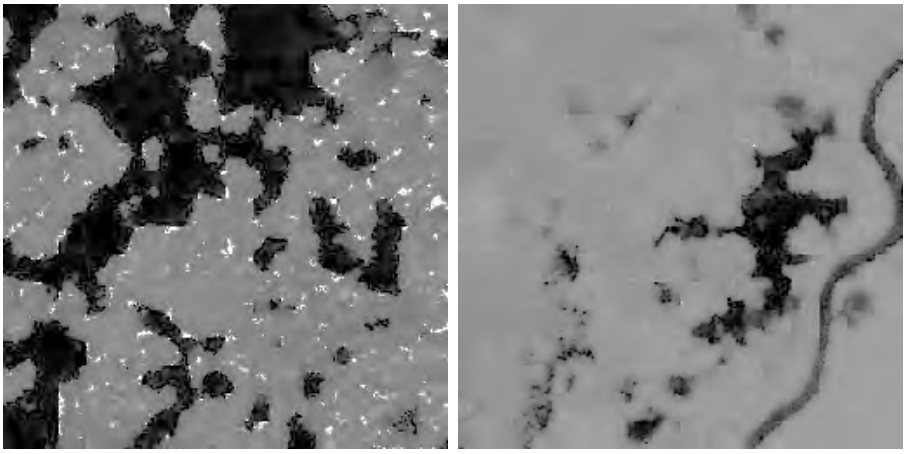


Figure 4.6: Images after using NDWI

4.2.3 High Dynamic Range (HDR)

In this pre-processing technique, the range of luminosity is increased. Here, we simulated HDR through a tone mapping algorithm, Reinhard Algorithm, to enhance the dynamic range of the image.

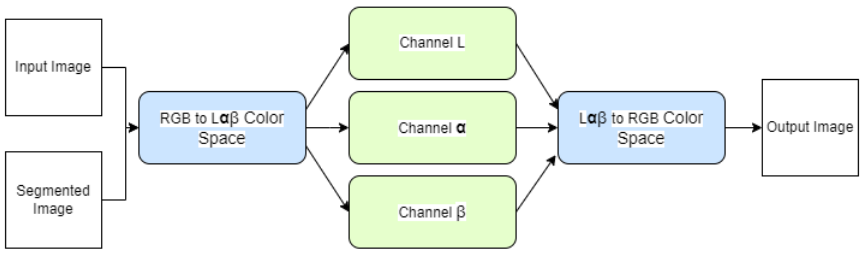


Figure 4.7: Reinhard Tone Mapping

The transformations for the three channels (L, α, and β) are given by the following equations:

Channel L,

$$O_L = \frac{\sigma_t^L}{\sigma_c^L}(c^L - \mu(c^L)) + \mu(t^L) \tag{4.5}$$

Channel α,

$$O_\alpha = \frac{\sigma_t^\alpha}{\sigma_c^\alpha}(c^\alpha - \mu(c^\alpha)) + \mu(t^\alpha) \tag{4.6}$$

Channel β,

$$O_\beta = \frac{\sigma_t^\beta}{\sigma_c^\beta}(c^\beta - \mu(c^\beta)) + \mu(t^\beta) \tag{4.7}$$

Where:

- O represents the output channel value

- c represents the content image channel value
- t represents the target image channel value
- μ denotes the mean
- σ denotes the standard deviation

This is how we implemented HDR pre-processing,

Algorithm 3 HDR Implementation

Require: *image*
Ensure: *hdr_image*
  procedure APPLY_CLAHE_COLOR(*image*, *clip_limit*, *grid_size*)
    display_image(*hdr_image*)
    return *hdr_image*
  if *image* is not None **then**
    *exposure*1 ← convert_to_float32(*image*)/255.0
    tonemap ← create_tonemap_reinhard()
    hdr_image ← process_tonemap(*tonemap*, *exposure*1)
    hdr_image ← convert_to_uint8(*hdr_image* ∗ 255)
    display_image(hdr_image)

We converted the image to its float32 format and normalized it by dividing it by 255. The Reinhard tone mapping is applied to simulate an HDR effect (Figure 4.9) to adjust the contrast and brightness. This makes the image to have more details in both shadows and highlights. Then, we convert the processed image back to an 8-bit format and display it.

Original Images,

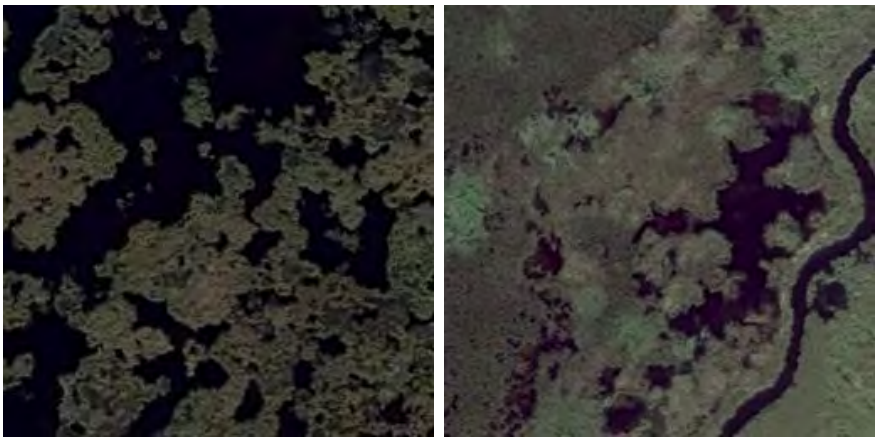


Figure 4.8: Images before using HDR

Corresponding output after HDR,

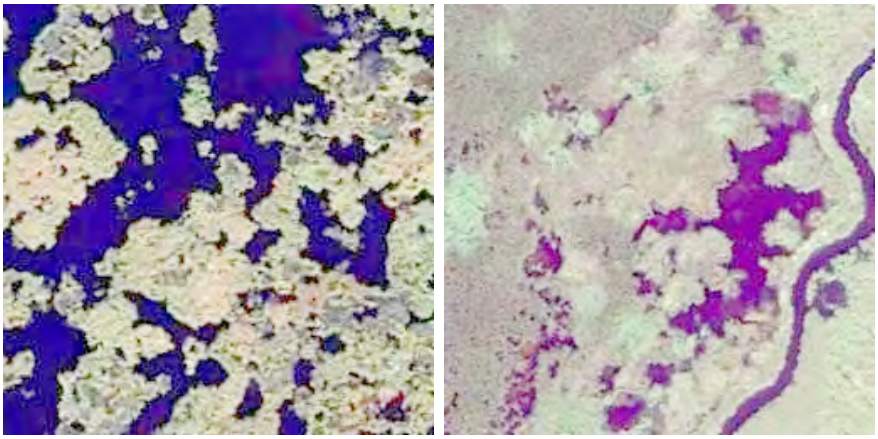


Figure 4.9: Images after using HDR

The goal of performing HDR is to capture the details in both darker and lighter areas of the images.

4.3 Model Specification

Our research focuses on using unsupervised satellite image data to localize the expansion and shrinkage of wetlands. In pursuit of doing that, we came across challenges led by existing frameworks that are primarily operated on supervised data. Some of the models were very good with segmenting, but they needed annotation to segment. To overcome this hurdle, we are taking a step-by-step approach by integrating three models, the Recognize Anything Model (RAM) [46], Grounding DINO (GD) [40], and Segment Anything Model (SAM) [37], calling it RAM-Grounded-SAM, each having its own purpose of execution to enhance the abilities of SAM to mask our targeted perimeter making it possible to work with unsupervised data.

4.3.1 Segment Anything Model (SAM)

The Segment Anything Model (SAM) [37] is a transformer-based model that leverages the power of self-attention mechanisms to segment objects in a highly efficient and flexible manner.

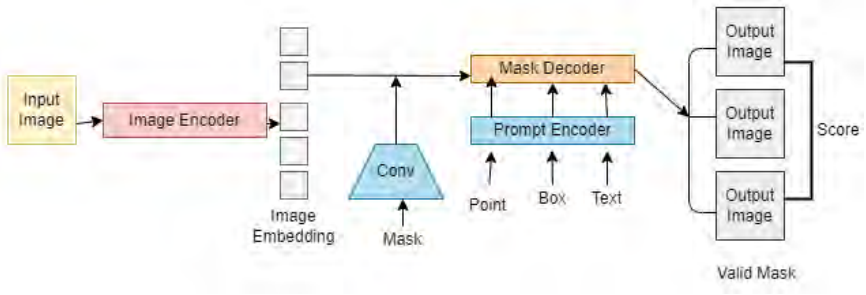


Figure 4.10: Segment Anything Model (SAM) Architecture

It intertwines the principles of both computer vision and natural language processing (NLP) by representing a significant stride to tackle the promotable segmentation task.

The notion of prompting techniques of NLP inspires the model task, revolving around generating a valid segmentation mask based on a given prompt. These prompts are multimodal and embrace various forms of information such as foreground, background points, rough bounding boxes, and free-form texts about which part of the input image should be segmented. SAM uses zero-shot learning with prompting instead of re-training with three components structured into an image encoder, flexible prompt encoder, and fast mask decoder.

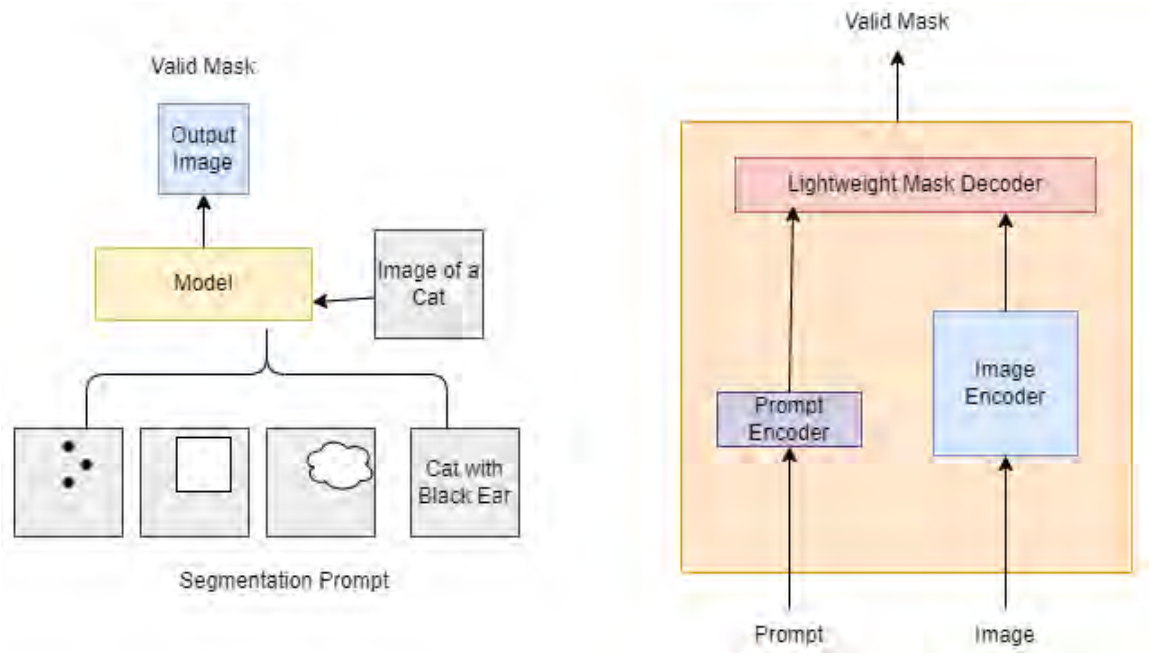


Figure 4.11: Overview of Segment Anything Model (SAM)

The image encoder is used to generate embedding for the segmented input image. This portion was motivated by flexibility in processing high-resolution inputs and consisted of a masked autoencoder (MAE) pre-trained Vision Transformer (ViT). Meanwhile, the prompt encoder is used to generate embedding for two sets of prompts. One represents sparse, including points, boxes by positional encodings, and texts using a text encoder from CLIP [29]. Another represents the Dense prompt, which is masks having spatial correspondence with the image, which is summed up with convolutions.

The last component, the Mask Decoder, is made by modifying the Transformer decoder block that updates all embeddings by using self-attention and cross-attention in two directions. Image and unmask embedding are fused using element-wise summation and put through a mask decoder. As for output, to avoid ambiguity, three output scores are shown rather than one.

4.3.2 Grounding DINO

Grounding DINO [40] is an open set object detector with a dual-encoder single-decoder architecture by performing vision-language-modality fusion that outputs multiple object boxes and noun phrases for a given image or text pair.

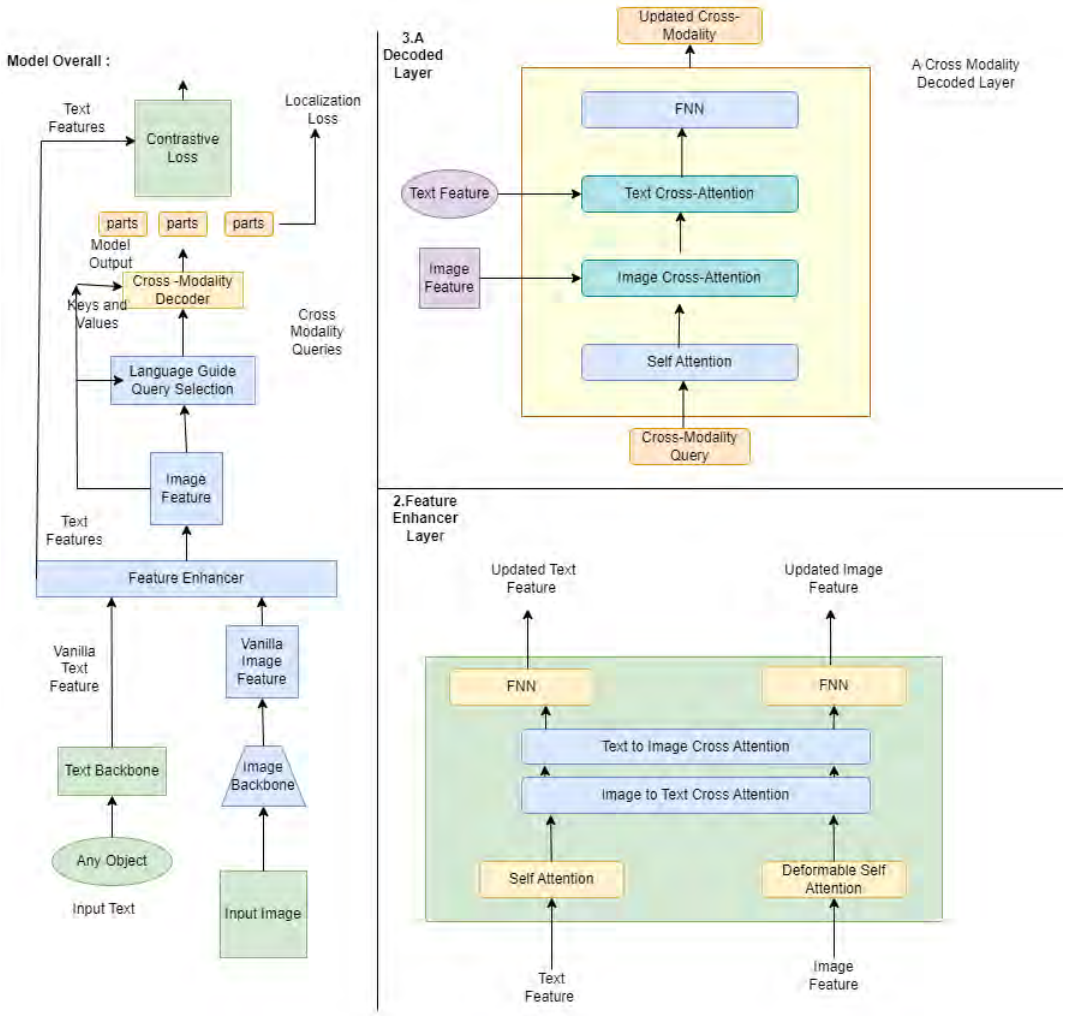


Figure 4.12: Architecture of Grounding DINO

The architecture is sectioned into a Feature encoder, feature enhancer, language-guided query selection, and cross-modality decoder. First of all, with the feature encoder, image and text features are extracted with Swin-T and BERT, respectively. Then, these extracted features are fed into the feature enhancer layer to perform cross-modality feature fusion, which includes multiple feature enhancer layers. As Grounding Dino takes images and detects specified input text, the authors have built a language-guided query selection module. Then, the cross-modality decoder is used to combine image and text modality features, and each cross-modality is fed into a self-attention layer and into the image cross-attention layer and text attention layer for combining image and text, respectively. Auxiliary loss is computed after each decoder layer and encoder outputs, and final losses between ground truths and matched predictions are calculated. Finally, the outputs of the last layer are used to predict the object boxes to extract phrases.

4.3.3 Recognize Anything Model (RAM)

Recognize Anything Model (RAM) [46] is a strong foundational model for image tagging that can identify important tags better than other models without overlooking crucial details. It enables generalization to previously unseen categories by incorporating semantic information into label queries.

The image tags are extracted through text semantic parsing without manual annotations. Similar to the text2tag model, RAM has three components: image encoder, image-tag recognition decoder, and text generation encoder-decoder.

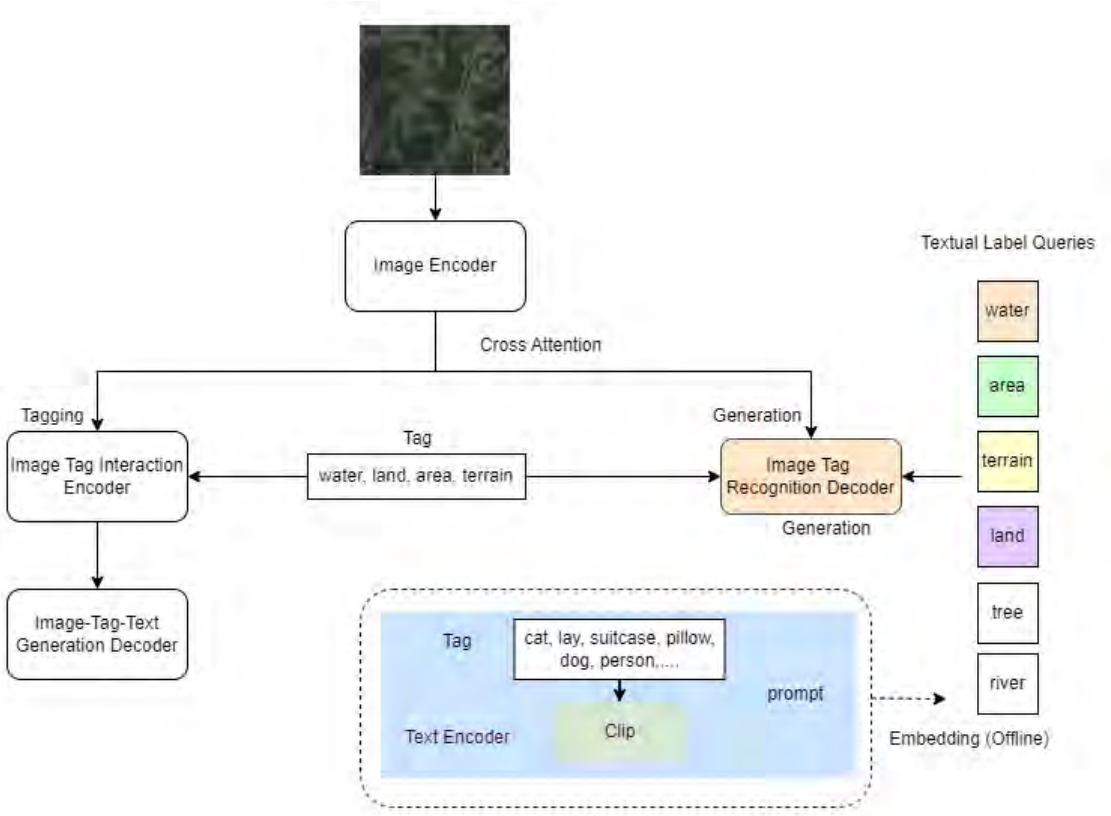


Figure 4.13: Recognize Anything Model (RAM) Architecture

In the earlier stage, for feature extraction, cross-attention layers are passed through an image-tag interaction encoder with generation and through an image-tag recognition decoder with tags. As the image encoder, Swin Transformer (Swin-T) is used over ViT. To perform prompt ensembling, CLIP is used as a text encoder and also as an image encoder to distill image features to improve recognition ability on unseen categories, unlike the text2tag model. During the training phase, the recognition head focuses on understanding and predicting tags that are extracted from text. Then, it takes on a dual role in the real-world application or inference phase. At first, it acts as a crucial bridge between images and tags that helps to transform them into meaningful labels, which further enhances the process of generating image captions with detailed semantic guidance through the tags that were predicted.

4.4 Proposed Model: RAM-GROUNDED-SAM

In our comprehensive research study, the main focus and goal is to segment wetland areas with the primary objective of effectively localizing and thoroughly analyzing the patterns of shrinkage and expansion over time. Here, we feed our model satellite image data collected from MLRSNet and Zoom Earth [52]. Then, we test it out on the Google Earth [49] satellite HD images from the year 1983 to 2023.

In order to apply that, we started off with the Segment Anything Model (SAM). While studying the model and testing it on our data, we found that it uses Zero-Shot Transfer [34] and that it needs labels or classes for masking and captioning. Since we are working on Unsupervised data and found that the Recognize Anything Model (RAM) does not necessarily need annotated data, we decided to combine them. In addition, we incorporated Grounding DINO to build boxes on our segmented mask accurately. Therefore, our proposed model architecture is a fusion of 3 models with three stages along which we build an inference, where each of the models has its own specified functions and strengths.

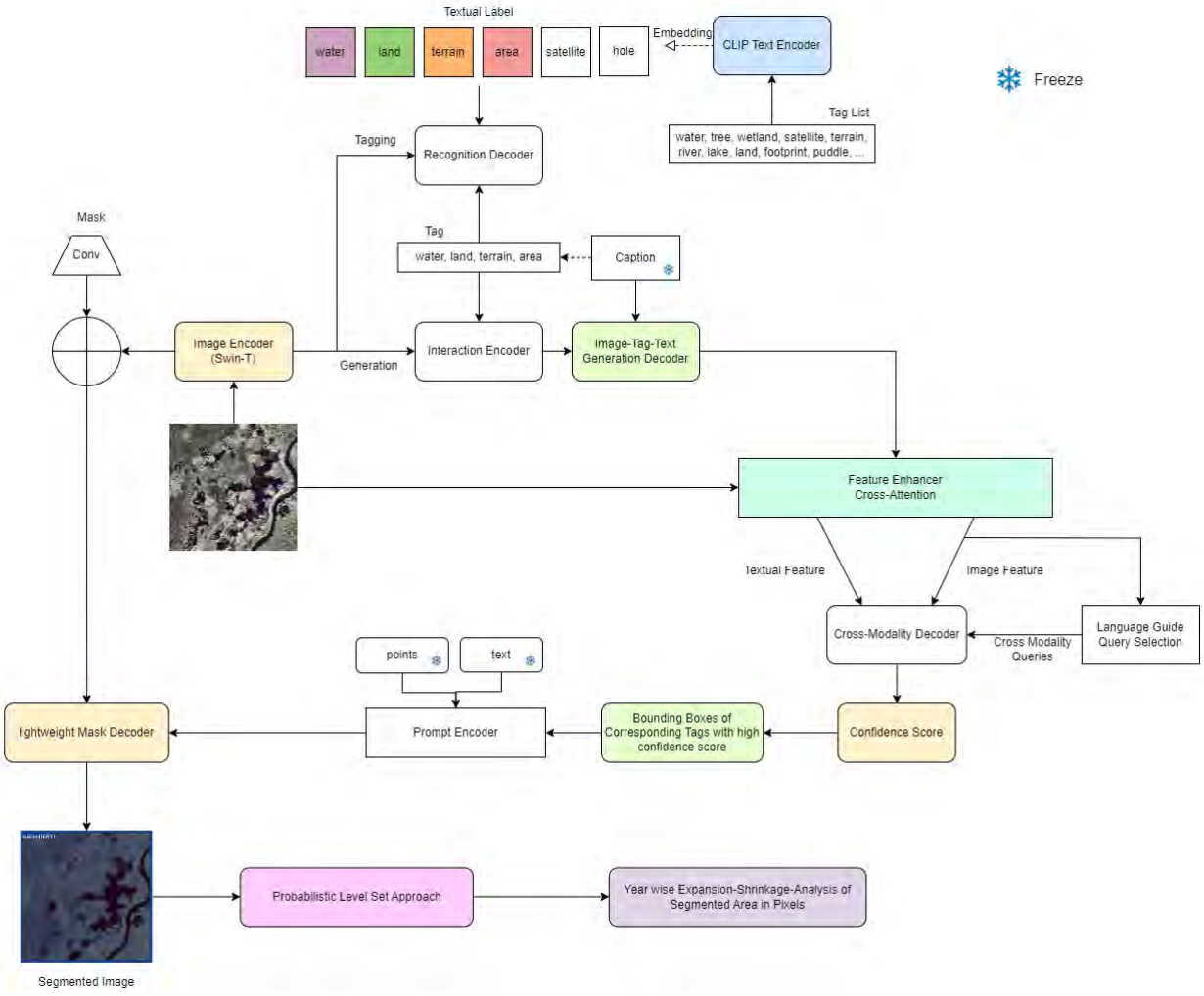


Figure 4.14: Proposed Model Architecture

In the first stage, we use the Recognize Anything Model (RAM), passing the image through an image encoder, Swin-T, a transformer model, for feature extraction. Here, visual and language features are extracted using 12-layer transformers for text generation and two 2-layer transformers for tag decoder. These layers collaborate to extract higher-level features from the input image incrementally. Initially, in the early layers, low-level features like edges and textures are detected, and gradually, as the layers progress, the features build up to more complex features, such as object parts and shapes in the later layers.

After the features are extracted from the image, the output of the encoder, which is a feature map, is split into 2 sections: Generation and Tagging, using cross-attention where generation goes into the Image-Tag Interaction Encoder, and Tagging goes into the Image-tag Recognition Decoder. The Image-Tag-Text Generation Decoder puts the tag in textual form for captioning. On the other hand, the CLIP text encoder, prompt engineering, is employed to provide the model with a set of textual prompts for the object of interest in the image (Algorithm 4). In our case, these prompts can be "water," "land," "footprint," "puddle," "tree," and so on.

CLIP text encoder is responsible for embedding the tag list (textual prompts) into a high-dimensional vector space, which creates Textual label queries for the image to be put through the Image-Tag Recognition Decoder. We utilized two types of tag lists: one in English and the other in Bengali, which we created ourselves. The embeddings are combined with the visual features extracted by the encoder with corresponding textual prompts to effectively learn recognition and generate the tags to be segmented later on.

Algorithm 4 RAM and Tagging Model Inference

Require: *image*, *tagging_model*, *tagging_model_type*, *specified_tags*, *do_det_seg* ▷ representing image, tagging_model, tagging_model_type, specified_tags, do_det_seg as a, b, c, d, e respectively
Ensure: *tags_result*, *caption_result*, *det_seg_result*
  procedure INFERENCE_TAGGING_MODEL(a, b, c, d, e)
    if c is "RAM" **then**
      $res \leftarrow$ inference_ram(a, b)
      $tags \leftarrow$ replace first index of res
      print("Tags: ", tags)
    else
      $res \leftarrow$ inference_tag2text(a, b, d)
      $tags \leftarrow$ replace first index of res
      $caption \leftarrow$ third index of res
      print(format("Tags: tags"))
      print(format("Caption: caption"))
    if not e **then**
      if c is "RAM" **then**
        return replace(tags, ", ", " | "), caption, None

After the tagging and captioning part, the second stage is to build object segmentation boxes (Algorithm 5) for each tag generated by RAM. In this stage, Grounding DINO takes the input image and tags from the previous stage (RAM) instead of relying on its pixel-level labels to identify and describe objects within the image.

Here, the model gets the predictions of confidence scores and bounding boxes to detect the tag. These predictions are filtered by applying a threshold to remove the low confidence scores and keep only the high confidence scores to enhance the reliability of the features retained for further processing.

$$c_i = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}} \tag{4.8}$$

where,

- c_i is the confidence score for tag i.
- z_i is the raw output (logits) for predicted tags from RAM.
- N is the total number of tags.

Then, the captions are tokenized, and the model generates phrases. The image and text features are passed through the Cross-Modality decoder to detect objects by comparing the predictions to these tokens.

Algorithm 5 GroundingDINO Inference and Box Transformation

Require: *tags*, *image*, *device*, *grounding_dino_model*, *box_threshold*, *text_threshold*, *iou_threshold*, *sam_model* ▷ representing these as a, b, c, d, e, f, g, h respectively
Ensure: *all_boxes*, *all_pred_phrases*
  procedure PROCESS_TAGS(a, b, c, d, e, f, g, h)
    for each *tag* **in** split(a, ',') **do**
      $image\_tensor \leftarrow$ convert_to_tensor($b, dtype =$ float32)$to(b)$
      $(boxes\_filt, scores, pred\_phrases) \leftarrow$ GDO($d, image\_tensor,$ strip(tag)$, e, f, b$)
  ▷ get_grounding_output as GDO, image_tensor as iTen
      $nms\_idx \leftarrow$ nms($boxes\_filt, scores, g$).numpy().tolist()
      $boxes\_filt \leftarrow boxes\_filt[nms\_idx]$
      $pred\_phrases \leftarrow [pred\_phrases[idx]$ for idx in $nms\_idx]$
      $transformed\_boxes \leftarrow$ apply_boxes_torch($h, boxes\_filt,$ shape(b)[: 2])$to(b)$
      extend(all_boxes, boxes_filt to CPU)
       extend(all_pred_phrases, pred_phrases)−

From the last decoder layer, a set of bounding boxes, confidence scores, and descriptive phrases is used to predict object boxes that highlight and describe the objects in the image, effectively grounding the textual description in the visual data. Here, the tags with high confidence scores get selected to be passed forward for setting bounding boxes.

So far, we have obtained the tags and bounding boxes. Moreover, we now move on to the final stage, where we generate the segmentation mask. In this stage, we use the Segment Anything Model (SAM) to segment and draw a mask on the specific element to annotate images with masks visually. The masks highlight the area of interest by drawing points over all non-zero pixels in a mask, using either a specified color or a random color if we specify it.

Algorithm 6 SAM Model Mask Prediction

Require: sam_model, $transformed\_boxes$, $device$, $size$ ▷ represent as a, b, c, d respectively
Ensure: out_image
  procedure PREDICT_AND_DRAW_MASKS(a, b, c, d)
    $(masks, \_, \_) \leftarrow$ predict_torch(a, point_coords = None, point_labels = None, boxes = to(b, c), multimask_output = True)

    $mask\_image \leftarrow$ create_new_image($'RGBA', size$, color = $(0, 0, 0, 0)$)
    $mask\_draw \leftarrow$ create_image_draw($mask\_image$)
    for each $mask$ **in** $masks$ **do**
      $draw\_mask$(convert_to_numpy(cpu($mask[0]$)), $mask\_draw$, random_color = True)
    $out\_image \leftarrow$ convert_to_RGBA(raw_image)
    alpha_composite(out_image, mask_image) **return**(out_image)

The masking process uses a transformer decoder, which has a unique ability to focus on important parts of the image using its self-attention mechanism. This means that it can zero in on the visual features that matter, guided by the prompts that it got from the RAM stage. This allows us to highlight the relevant objects while ignoring the background noise. The decoder works in steps and continuously improves its understanding of the objects along the way by looking at different parts of the image and refining its segmentation. This step-by-step refinement enables the model to accurately capture the complex shapes and boundaries of the objects it identifies, resulting in precise and well-defined segmentation masks.

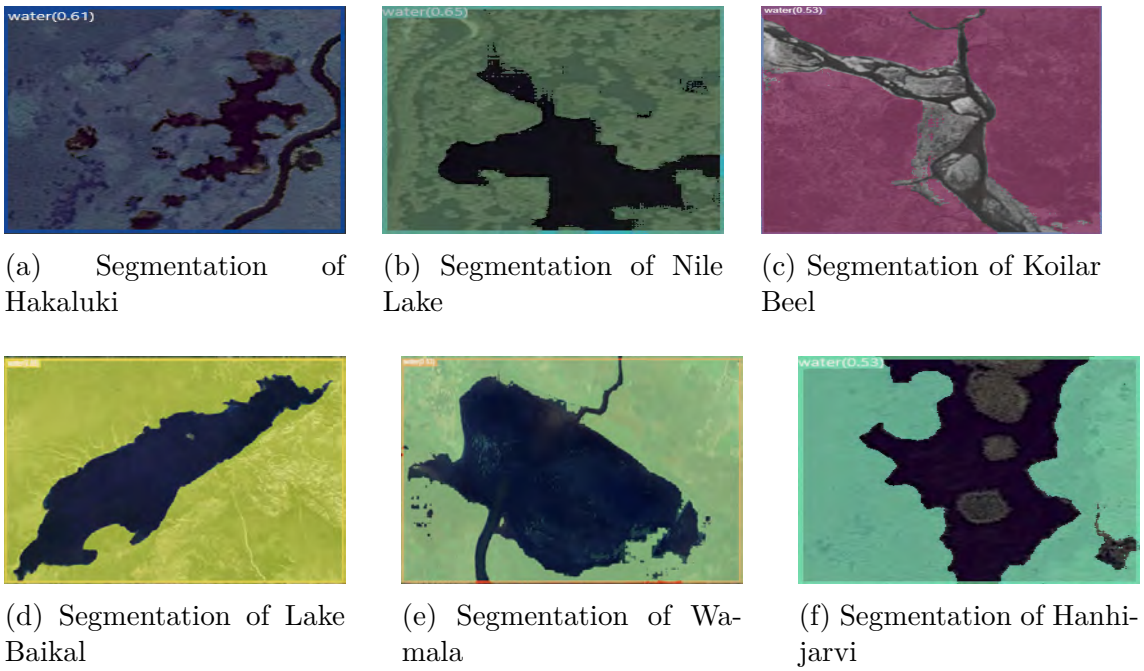
To integrate the three models effectively, we developed an inference mechanism that acts as a bridge between the models and freezes the sections that are not necessary for our specific task.

For instance, in the image encoder section, we decided to use the Swin-T [50] architecture from RAM, as it performs better for our purposes than the Masked Autoencoder (MAE) [27] pre-trained Vision Transformer (ViT) [26], [51] used in SAM. So, we utilize the Swin-T encoder for our dataset because it allows us to process the data more effectively compared to using ViT.

Another thing to note is that, after obtaining the extracted tags with the RAM, if we were to feed the output directly from stage 1 to stage 3, which is SAM for masking, we might segment all the tags extracted in the first stage. However, our inference strategy ensures that only the relevant features and sections, in our case, 'water,'

'wetland,' 'river,' 'lake,' and any water and wetland-related tags, are passed along, maintaining the precision and efficiency of the segmentation process. By carefully coordinating the models, we can use each one's strengths while avoiding unnecessary computations and redundancy, which results in more accurate and reliable segmentation outcomes.

Segmentation output with our proposed Model,



(a) Segmentation of Hakaluki

(b) Segmentation of Nile Lake

(c) Segmentation of Koilar Beel

(d) Segmentation of Lake Baikal

(e) Segmentation of Wamala

(f) Segmentation of Hanhijarvi

Figure 4.15: Wetland Segmentation with our Proposed Model. The output tags are area, image, land, sea, satellite, terrain, and water. Before performing the confidence matrix, the NMS box was 7, and water had the highest confidence score with 0.61, 0.65, 0.53, 0.65, and 0.53, respectively; the NMS box became 1. After the RAM phase, Grounding DINO passed the water box to SAM, and the water body was masked. The segmented images are mix of the MLRSNet, and our dataset.

Chapter 5

Model Implementation and Result Analysis

In this chapter, we start analyzing the theoretical concepts and put them into practice with Zoom Earth [52] and Google Earth [49] data while detailing how we implemented various models, algorithms, and methods discussed earlier. Here, we thoroughly look at the process of building inference for combining the three models, training after fine-tuning with additional layers, and testing our models using Zoom Earth and Google Earth of the years 1983-2024. Additionally, we implement probabilistic methods on the segmented images generated by our model to find the fluctuation of the areas. Through a detailed examination and experiment, we aim to validate our hypotheses, address research questions, and make significant contributions to our field of study.

5.1 Model Implementation and Experiments

To implement the model, we used specified pre-train checkpoints: 1) Ram and tag2text use Swin Large 14m Transformer [49], 2) Grounding DINO uses SwinT_OGC, and 3) SAM uses MAE pre-trained ViT-H [49]. RAM-Grounded-SAM was trained on the MLRSNet and a few Zoom Earth satellite image Datasets.

5.1.1 Training Phase

For our training phase, we prepared a custom function to load our data efficiently, which is designed to handle specific formats while preprocessing the data. While in the preprocessing stage, we applied histogram, Normalized Difference Water Index, and High Dynamic Range and chose Contrast Limited Adaptive Histogram Equalization, which is integrated into our custom function to apply CLAHE on the images and send it for the training phase.

In the RAM recognize section, it takes the output of CLAHE and starts feature extracting where we did not freeze anything and let it run.

During the fine-tuning process with the Segment Anything Model (SAM), we contributed to generating binary masking using prompt engineering that distinguishes the object of interest (wetland) from the background, which SAM was not used to

doing. For each epoch SAM takes the prompts based on the previous steps, converts to Embedding vector space feed to model and predicts the pixels of interest, and draws a mask on it, finally giving us the segmented output. As SAM is designed primarily for zero-shot generalization and a decoder-only model, it can proficiently create segmentation masks using prompts it got from Grounding DINO, bounding boxes, and text descriptions specifying wetland property.

Algorithm 7 Binary Mask Generation with SAM

Require: *sam_model*, *image*, *prompt* ▷ SAM model, input image, segmentation prompt
Ensure: *binary_mask*
  procedure GENERATE_BINARY_MASK(*sam_model*, *image*, *prompt*)
    segmentation_mask ← *sam_model.segment*(*image*, *prompt*) ▷ Generate segmentation mask
    binary_mask ← convert_to_binary(*segmentation_mask*) ▷ Convert to binary format
    return *binary_mask*

This algorithm takes the SAM model, input image, and segmentation prompt from Grounding DINO as input and returns a binary mask. It first generates a segmentation mask using the SAM model and then converts this mask into a binary format to distinguish between the wetland property and the background.

In essence, our training phase established a solid foundation for data preprocessing and model preparation. The integration of SAM during fine-tuning presents an avenue to enhance segmentation accuracy tailored to distinct object categories.

5.2 Model Comparison

In this portion of our research, we conduct an in-depth comparison between the segmentation capabilities of the PaliGemma [53] and our proposed model, RAM-Grounded-SAM.

5.2.1 PaliGemma vs RAM-Grounded-SAM (Proposed Approach)

PaliGemma, which leverages the SigLIP [45] vision and Gemma language model inspired by PaLI-3, is a lightweight open vision-language model (VLM) designed for various segmentation tasks. Our objective is to evaluate the performance of both models in segmenting water bodies from different types of imagery.

To start with, we test and analyze PaliGemma's segmentation efficiency using a Google Earth image of a river. Below, the first image on the top shows the original Google Earth image, while the second image on the bottom demonstrates the segmentation output produced by PaliGemma.

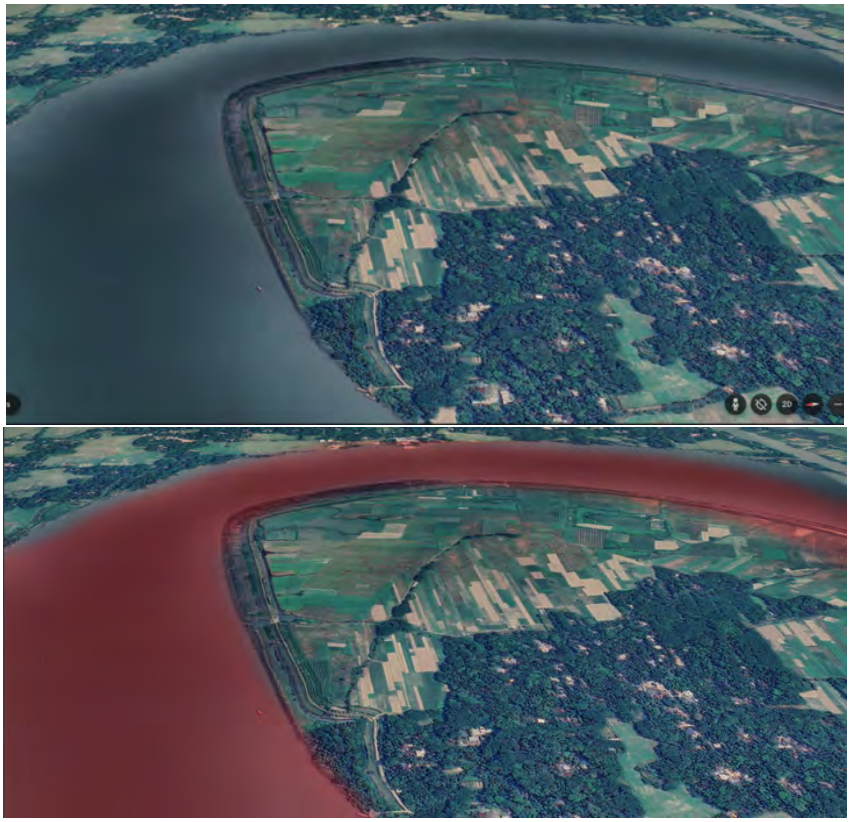


Figure 5.1: Segmentation of PaliGemma on Google Map Image. The prompt was given 'segment the river' and a paligemma-3b-mix-224 model with greedy decoding.

As we can see, PaliGemma can segment the river ideally because it recognizes the water body pattern in the Google map-like view, which indicates that the model effectively utilizes its prior knowledge from seen images data to recognize and segment rivers in Google Earth imagery.

Next, we assess the performance of PaliGemma segmentation on a satellite image from our dataset. The following images showcase the original satellite image on the left and the segmentation output by PaliGemma on the right.

Figure 5.2: Segmentation of PaliGemma on Satellite Image. The prompt was given 'segment the wetland'. The model used was paligemma-3b-mix-224 with greedy decoding.

In this case, the model struggles significantly when given the prompt to segment wetlands or rivers, masking the entire image. The challenges faced while segmenting satellite imagery could be due to the different characteristics and complexities of such images compared to the Google Earth images it was trained on or familiar with. When we pass the Google Earth image through our model, it showcases proficient segmentation capabilities and effectively maps out various features within the image. Additionally, our model outperforms PaliGemma when applied to satellite imagery, demonstrating superior segmentation accuracy and refinement.

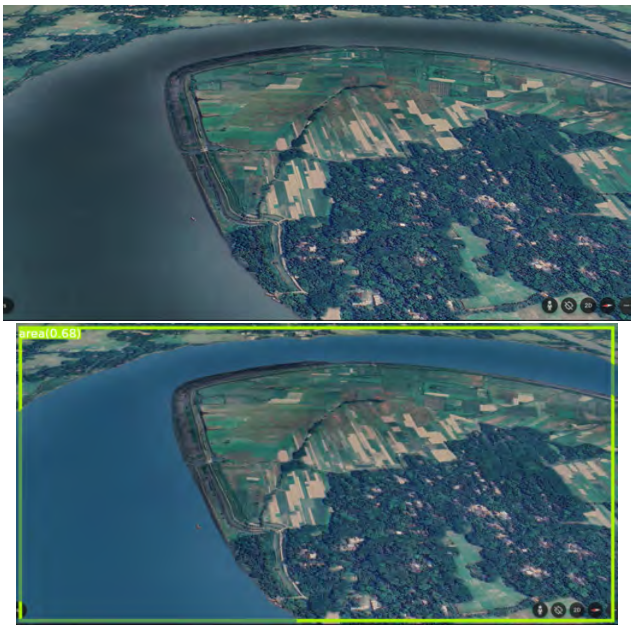


Figure 5.3: Segmentation of RAM-Grounded-SAM on Google Earth Map-like Image

Through this comparative analysis, we highlight the strengths and limitations of PaliGemma, emphasizing the necessity for a robust model like RAM-Grounded-SAM, which we propose to overcome these challenges. As Paligemma is totally new, it needs very curated data, but our model can segment on satellite or drone-capture data, which is why it achieves more accurate and reliable segmentation across various types of imagery.

5.3 Forecast of Wetland Fluctuations With Probabilistic Method

In this research stage, we used Time Series methods with the Probabilistic Level Set Approach following the segmentation to accurately determine the localized regions. This step helps us to identify the total area that is segmented from the image. We ran the Probabilistic Level Set method on our test image data from Google Earth [49] within the year 1983 to 2022 that was segmented by our model.

5.3.1 Probabilistic Level Set Approach

Probabilistic Level Set methods integrate statistical information from image data into their framework, which helps to enhance the robustness and accuracy of the segmentation process. The level set function is calculated over the region of interest to determine the area of a segmented object. In these approaches, object boundaries are defined by the zero-level set of a higher-dimensional function. The Chan-Vese segmentation algorithm from Probabilistic Level Set methods aims to divide images into regions based on criteria like intensity uniformity. The segmented area refers to the count of pixels within the identified region the algorithm has distinguished from the rest of the image. We modified the approach for our research since we had already segmented the image with our proposed model. We freeze the Chan-Vase method and incorporate our segmented image into the area calculation approach with 'cv2.countNonZero', which finds the clustered pixels (Algorithm 8). Also, the algorithm has determined to be part of the image's object or region of interest, in our case, the segmented water area.

Algorithm 8 Calculating Area of Segmented Objects

Require: *segmentation_mask*
Ensure: *areas*
  procedure CALCULATE_AREA_LEVEL_SET(*segmentation_mask*)
    areas ← create_empty_list()
    threshold ← 135
    max_value ← max(*segmentation_mask*)
    for each *i* **from** *threshold* **to** (*max_value* + 1) **do**
      object_mask ← (*segmentation_mask* == *i*).astype(uint8)
      area ← count_non_zero(*object_mask*)
      append(areas, area)−
       return (areas)−

5.3.2 Wetland Fluctuation Forecast

Once we have successfully retrieved the total area for each segmented image, the next critical step in our research process is to integrate this data into time series models. This involves a systematic approach to converting the spatial information derived from the segmented images into a temporal dataset that allows us to analyze and forecast fluctuations and patterns over time. We explored and experimented with different time series methods to choose what was best for our approach.

Linear Regression (LiR)

Linear regression models the relationship between a dependent variable and 1 or more independent variables to find the best-fitting straight line through the data points. We used it as a baseline model to establish a reference point to gauge the added values of AutoRegressive Integrated Moving Average (ARIMA), Gaussian Hidden Markov Model, or Long Short-Term Memory (LSTM) while comparing. The equation linear regression uses,

$$y = \beta_0 + \beta_1 x + \epsilon \tag{5.1}$$

Here, ϵ is the error term. And β_0, β_1 are estimated using the least squares method to minimize the sum of the squared differences between the observed and predicted values.

$$\text{Minimize} \sum_{i=1}^{n} (observed\_values - \hat{observed\_values})^2 \tag{5.2}$$

where $\hat{observed\_values}$ represents the predicted values.

Linear Regression predicts a wetland fluctuation of 76292.9709639954 on our test time series dataset in June 2025.

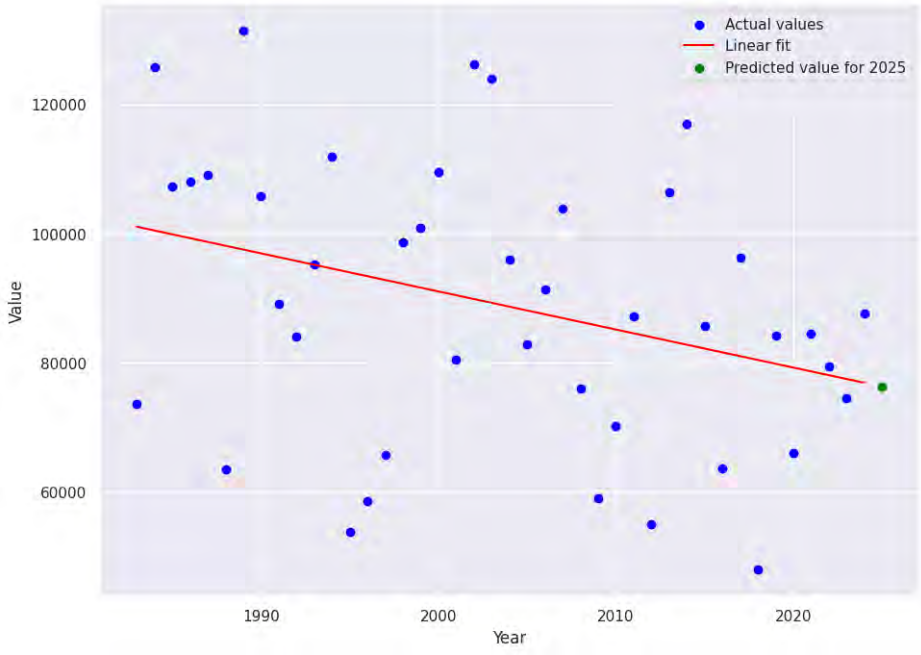


Figure 5.4: Linear Regression Prediction Plot

Decision Tree Regressor (DTR)

Decision Trees (DT) split the data into subsets based on the value of input features, wetlands segmented total area of each image, which is done recursively to form a tree-like structure of decisions and then to the prediction of the target variable. The splitting process continues until it reaches certain conditions, like reaching a maximum tree depth, having a minimum number of samples to split a node, or when there's no further improvement in information gain. The equation to the Decision Tree,

$$\text{Gini impurity} = 1 - \sum_{i=1}^{n} p_i^2 \tag{5.3}$$

In the case of the regression decision tree, the prediction is made by averaging the values of the training examples in the leaf node. Decision Trees predicts a wetland fluctuation of 87592.0 on our test time series dataset in June 2025.

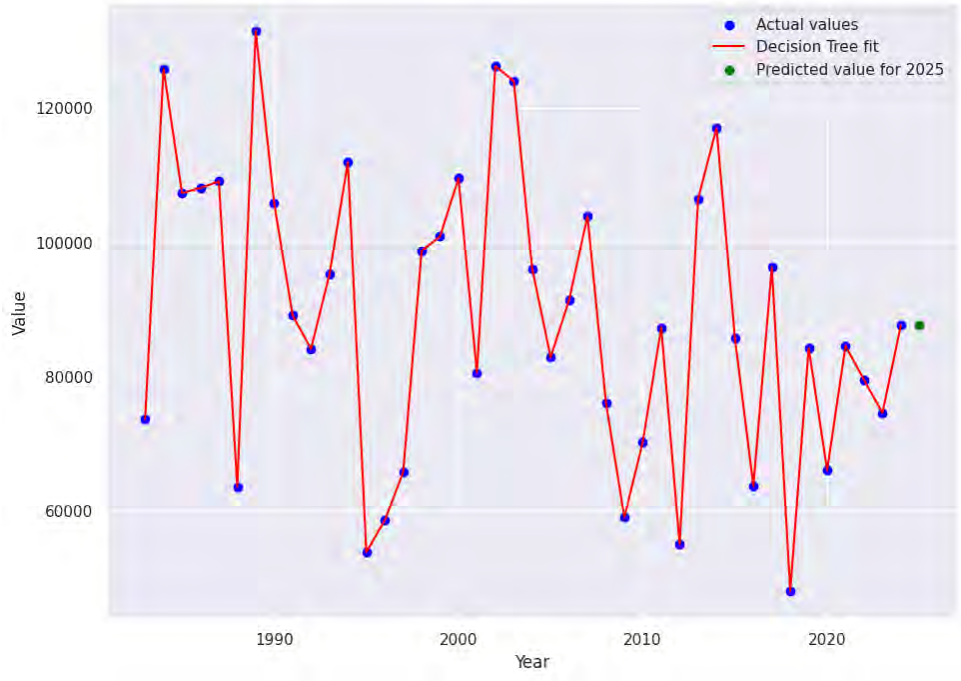


Figure 5.5: Decision Trees Prediction Plot

Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM), a type of recurrent neural network (RNN), can learn long-term dependencies, which makes them suitable for tasks where the previous context is important. As we want to predict the fluctuation of the next year and which depends on the fluctuation of the previous years, we used LSTM.

LSTMs use 3 gates: the input gate, the forget gate, and the output gate, to regulate the addition or removal of information from the cell state that allows the network to retain information over long periods.

The equation for Forget Gate is 5.1.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{5.4}$$

The equations for the Input Gate are the following: 5.2 and 5.3.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{5.5}$$

$$\tilde{C}t = \tanh(W_C \cdot [ht - 1, x_t] + b_C) \tag{5.6}$$

The equation for the Cell State Update is 5.4.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{5.7}$$

The equations for the Output Gate are the following: 5.5 and 5.6.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5.8}$$

$$h_t = o_t \cdot \tanh(C_t) \tag{5.9}$$

Linear Regression predicts a wetland fluctuation of 83751.9 on our test time series dataset in June 2025.

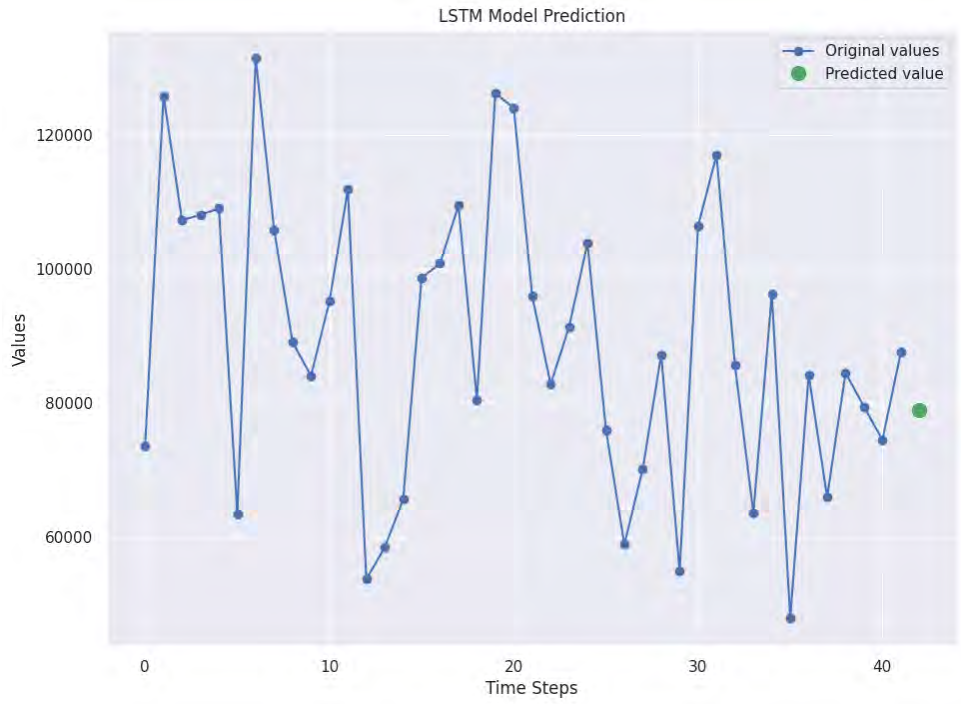


Figure 5.6: Long Short-Term Memory (LSTM) Prediction Plot

AutoRegressive Integrated Moving Average (ARIMA)

ARIMA, AutoRegressive Integrated Moving Average, is a statistical method for time series regression analysis and forecasting. Maximum likelihood estimation or least squares algorithms are used to estimate the coefficients of ARIMA. It combines three components: 1) AutoRegressive (AR), 2) Integrated (I), and 3) Moving Average (MA) to model a wide range of time series data and to forecast future points.

Starting with AutoRegressive (AR) involves regressing the time series on previous values, and the current value of the series is expressed as a linear function,

$$X_t = c + \sum_{i=1}^{p} \phi_i X_{t-i} + \epsilon_t \tag{5.10}$$

Here,

- X_t is the total area of a segmented image,
- c is a constant,
- p is the order of the AR model (the number of lagged values included),
- ϕ_i are the coefficients,
- ϵ_t is white noise.

To make the time series stationary, integrate part (I) is fused with a constant mean and variance over time, which is a requirement for many time series models. Differencing involving subtraction of the previous observation from the current observation can be applied one or more times until stationarity is achieved if the original series is not stationary.

$$Y_t = X_t - X_{t-1} \tag{5.11}$$

Here, Y_t is the differenced series. The number of times differencing is applied is denoted by the value of d in the order of integration. If d is 1, it means differencing is applied once, and If d is 2, differencing is applied twice, and so on.

The final component, the Moving Average (MA) Component, models the error term as a linear combination of past error terms. Here, the current value of the series is expressed as a linear function of previous white noise terms.

$$X_t = c + \epsilon_t + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} \tag{5.12}$$

Here,

- q is the order of the number of lagged forecast errors included,
- θ_i are the coefficients,
- ϵ_t is white noise.

These 3 components, AutoRegressive, Integrate, and Moving Average, are combined to work as a single framework, An ARIMA model which can be expressed as,

$$x'_{time} = I + \alpha_1 x'_{t-1} + \alpha_2 x'_{t-2} + \cdots + \alpha_p x'_{time-p} + e_time + \theta_1 e_{time-1} + \theta_2 e_{time-2} + \cdots + \theta_q e_{time-q} \tag{5.13}$$

- x'_t represents the value of the time series at time,
- I →constant,
- α_i →coefficients of the autoregressive component,
- x'_{t-i} →the lagged values of the time series,
- e_time is the error term with the given time,
- e_{time-i} →lagged value of the error term,
- θ_q →coefficients of the moving average component.

ARIMA predicts a wetland fluctuation of 80877.87 on our test dataset in June 2025.

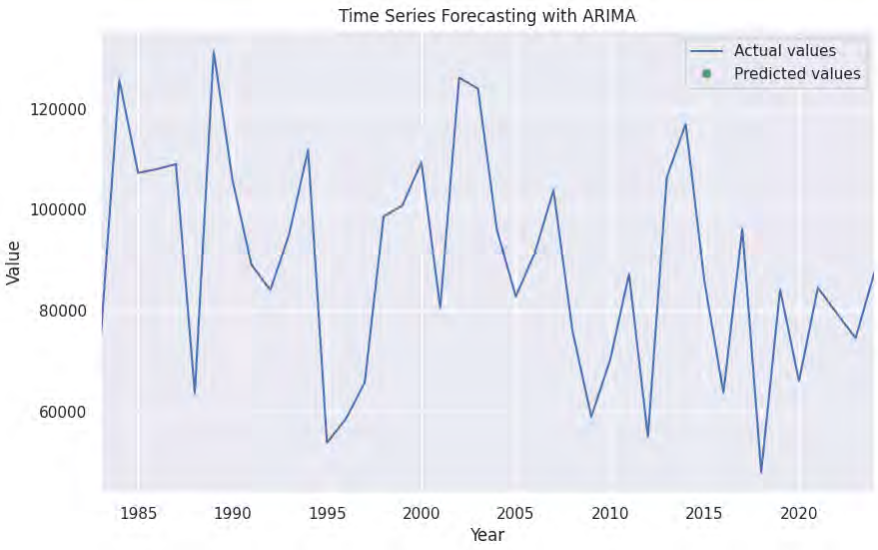


Figure 5.7: AutoRegressive Integrated Moving Average (ARIMA) Prediction Plot

Gaussian Hidden Markov Model (GHMM)

Gaussian Hidden Markov Model, a statistical tool, models sequences, and time series data, and the system being modeled are assumed to be a Markov process with hidden states. In our research, we are excited to harness the Gaussian Hidden Markov Model (GHMM) capabilities to delve into patterns within segmented data. By doing so, we anticipate gaining valuable insights into ecological dynamics, which will greatly inform decision-making processes in conservation efforts.

A Hidden Markov Model (HMM) combines two key components: 1) a Markov chain to model the hidden states, which transitions probabilistically, and 2) a probability

distribution to model the observations given the hidden states. In Gaussian HMM, the observations are modeled using Gaussian distributions.

The Markov chain begins with a set of hidden states, and the transition probabilities between these states adhere to the Markov property. This means that the probability of moving to the next state depends solely on the current state, without considering the sequence of previous states.

$$P(s_{t+1} = s_j \mid s_t = s_i) = a_{ij} \tag{5.14}$$

Here, a_{ij} is the transition probability from state s_i to state s_j, and the sum of transition probabilities from any state to all possible next states is,

$$\sum_{j=1}^{N} a_{ij} = 1 \tag{5.15}$$

In addition to the transition probabilities, there is an initial state distribution, which gives the probability of the system starting in each state,

$$P(s_1 = s_i) = \pi_i \tag{5.16}$$

Next, the observation model, which, in the case of GHMM, assumes that the observations are generated from Gaussian distributions. For each hidden state, there is an associated Gaussian distribution with mean and variance. Combining these components, a Gaussian HMM is characterized by,

$$P(x_t \mid s_t = s_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(x_t - \mu_i)^2}{2\sigma_i^2}\right) \tag{5.17}$$

where,

- N Number of hidden states,
- a_{ij} state transition probability matrix,
- π $\rightarrow$initial state distribution,
- μ_i is the mean of the Gaussian distributions for each state,
- σ_i^2 is the variance of the Gaussian distributions for each state.

This formulation enables the GHMM to capture complex patterns and fluctuations within the observed data.

Using a Gaussian Hidden Markov Model, we predict a wetland fluctuation of 81059.44024 on our test time series dataset in June 2025.

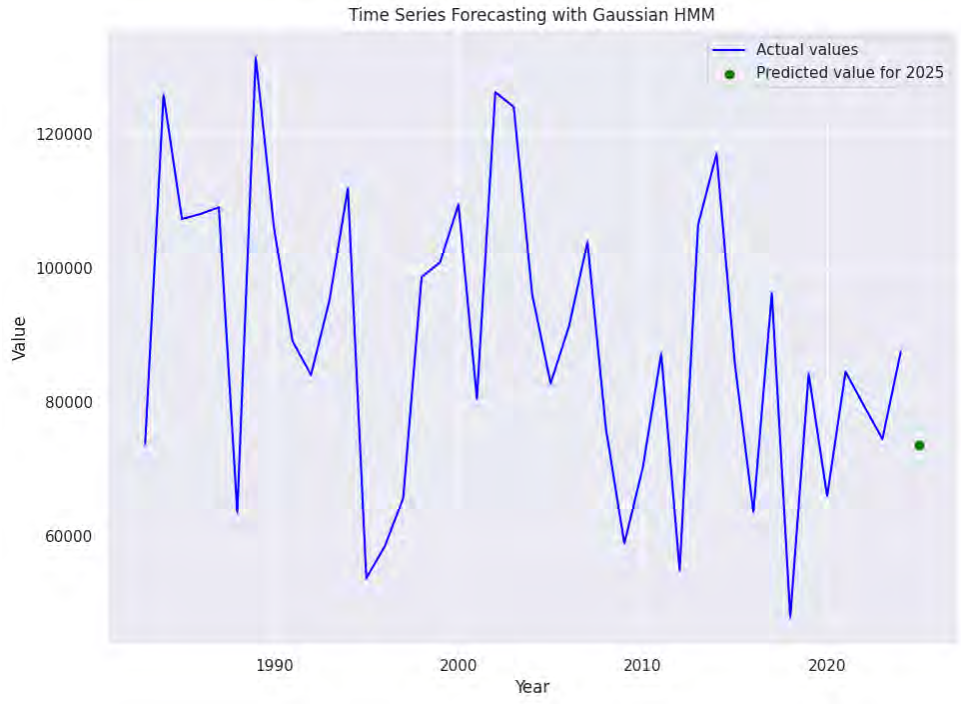


Figure 5.8: Gaussian Hidden Markov Model (GHMM) Prediction Plot

We can see in the time series graph of LSTM (Figure 5.6) and GHMM (Figure 5.8) that there will be a shrinkage in the next year, 2025, and ARIMA (Figure 5.7) predicts an expansion, where the decision tree remains unbiased, predicting that there will be no change. We chose linear regression as our base model, and we could say that most of the prediction models align with our base model. So we get a prediction of shrinkage in the following year. Also, out of all the models, GHMM gave us the best results of predictions because of its flexibility, robustness, and ability to handle unpredictability. Below, we show the results of all the methods we used for Fluctuation Prediction,

Table 5.1: Time Series Predicted value for June Year 2025 in Wetland Fluctuation. The best result is highlighted in bold.

Model	Prediction
Linear Regression (LiR)	76292.97
Decision Tree Regressor (DTR)	87592.0
Long Short-Term Memory (LSTM)	83751.95
AutoRegressive Integrated Moving Average (ARIMA)	80877.87
Gaussian Hidden Markov Model (GHMM)	**73654.86**

According to the assessment outcomes depicted in Table 5.1, it's evident that each model yields distinct forecasts for the time series data related to wetland fluctuations. Among the models evaluated, GHMM stands out with the lowest predicted value of 73654.86, indicating that it has effectively captured the underlying patterns

and dynamics within the wetland data, resulting in a more accurate prediction compared to other models. While the LiR predicts 76292.97 and DTR predicts 87592.0, these models may oversimplify the complex relationships inherent in the data, leading to less precise forecasts. On the other hand, LSTM and ARIMA offer predictions of 83751.95 and 80877.87, respectively. While these models demonstrate competitive performance, they may need to fully capture the intricate temporal dependencies present in the wetland data.

Therefore, the Gaussian Hidden Markov Model (GHMM) emerges as the most promising choice for predicting wetland fluctuations in June 2025 for our test dataset, offering the highest level of accuracy among the models considered. It is noteworthy that although the performance of GHMM in predicting values for 2022, 2023, and 2024 was not as effective as other models, it outperformed all others in the normal prediction of 2025.

5.3.3 Predicting Future Values and Validation with Actual Data

In this section, we try another approach to assess our models' performance. We set our models to predict wetland area values for the upcoming years 2022, 2023, and 2024.

To assess the accuracy of our predictions, we will validate the forecasted values for 2022 against observed data. Leveraging information up to the year 2021, our initial step involves predicting the wetland area value for 2022. Subsequently, this forecasted value will serve as input for predicting the wetland area values for 2023 and 2024 iteratively. By comparing the predicted values with actual data for 2022, we aim to gain valuable insights into the precision and reliability of our model, thereby ensuring a robust evaluation of its predictive capabilities.

Linear Regression prediction plot for years 2022, 2023, and 2024.

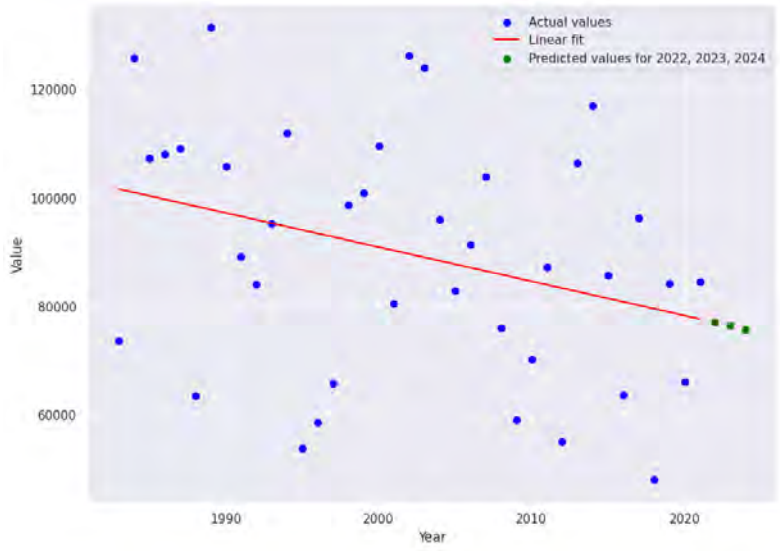


Figure 5.9: Linear Regression Prediction Plot

Decision Tree prediction plot for years 2022, 2023, and 2024.

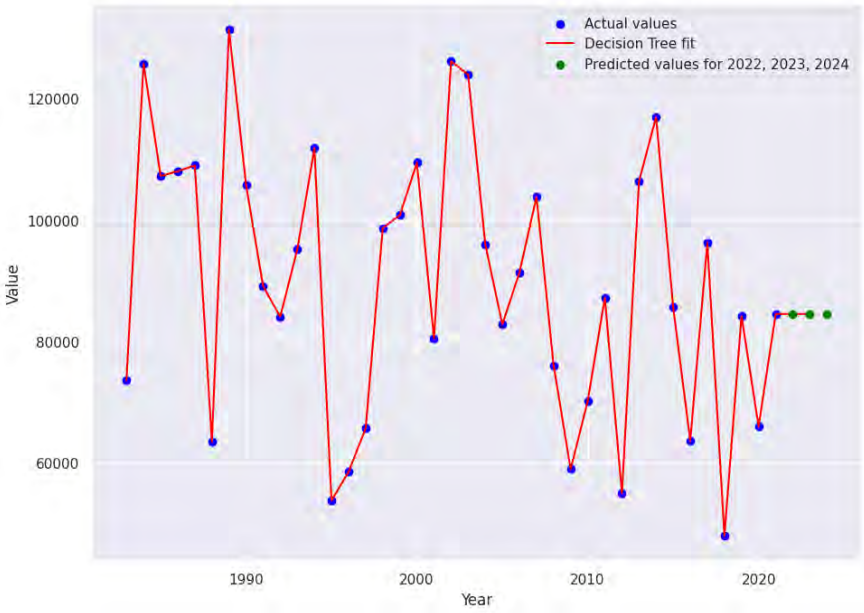


Figure 5.10: Decision Trees Prediction Plot

Long Short-Term Memory (LSTM) prediction plot for years 2022, 2023, and 2024.

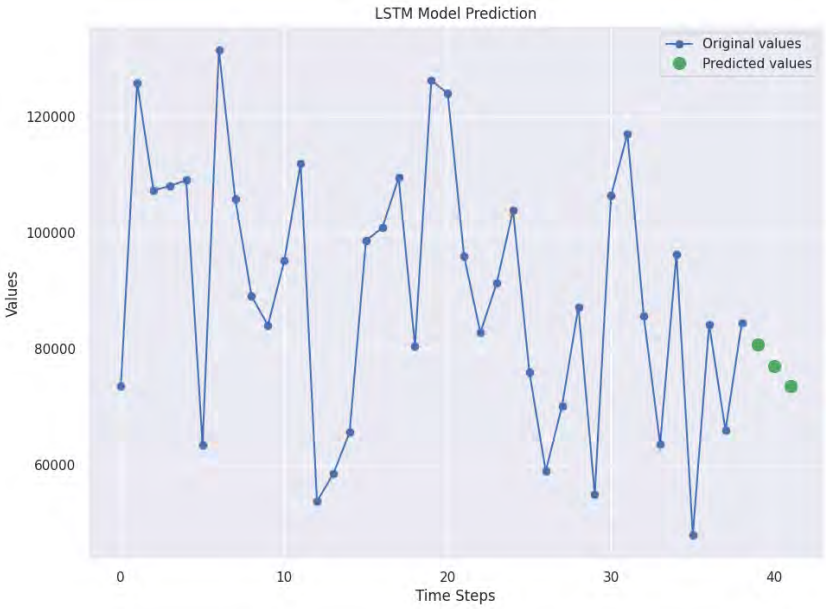


Figure 5.11: Long Short-Term Memory (LSTM) Prediction Plot

AutoRegressive Integrated Moving Average (ARIMA) prediction for years 2022, 2023, and 2024.

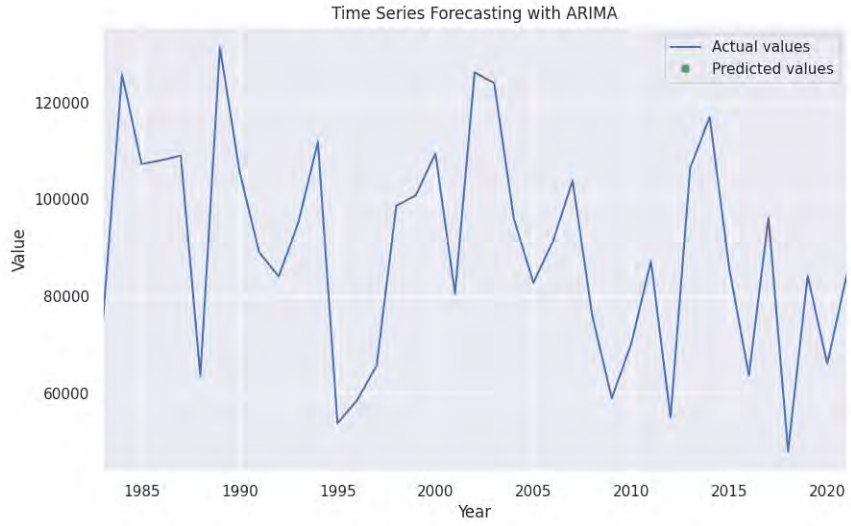


Figure 5.12: AutoRegressive Integrated Moving Average (ARIMA) Prediction Plot

Gaussian Hidden Markov Model prediction for years 2022, 2023, and 2024.

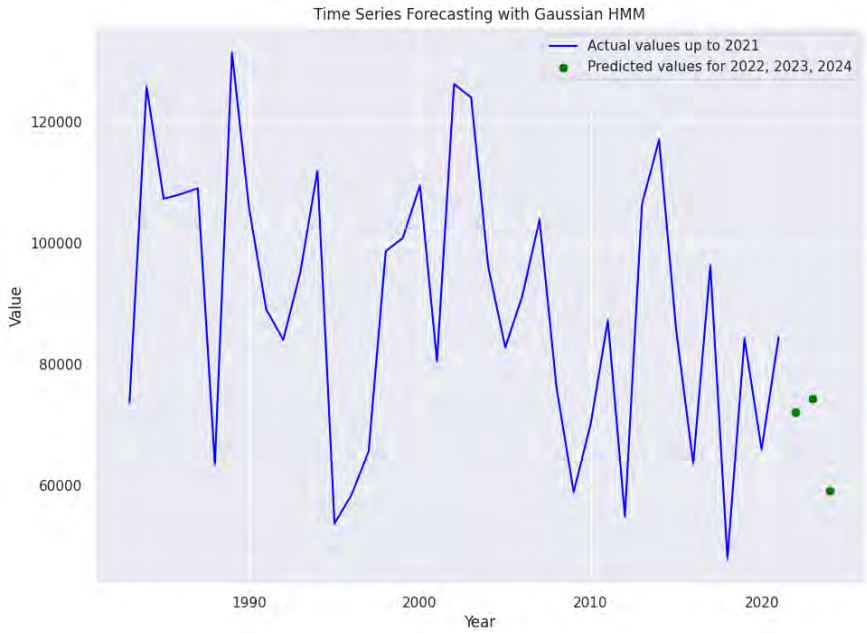


Figure 5.13: Gaussian Hidden Markov Model Plot

Predictions for years 2022, 2023, and 2024 are shown in the tables below.

In 2022, there is a discernible variance between the predicted and actual values across all models.

Table 5.2: Time Series Predicted Value vs Actual Value for June Year 2022 in Wetland Fluctuation. The closest predicted value of the Actual Value is highlighted in bold.

Model	Prediction of 2022	Actual of 2022
LiR	77011.20677	79448
DTR	87522.0	79448
LSTM	**80193.2890**	79448
ARIMA	74234.4677	79448
GHMM	72146.0201	79448

Notably, in Table 5.2, LSTM provides a close estimate, projecting 80,193 units compared to the observed 79,448 units. Similarly, linear regression offers a relatively precise approximation at 77,011 units. However, Decision Tree, ARIMA, and GHMM exhibit slight deviations from the actual value, with forecasts ranging from 87,522, 72,146 to 72146.0201 units.

Table 5.3: Time Series Predicted Value vs Actual Value for June Year 2023 in Wetland Fluctuation. The closest predicted value of the Actual Value is highlighted in bold.

Model	Prediction of 2023	Actual of 2023
LiR	76380.84	74528
DTR	87522.0	74528
LSTM	83751.95	74528
ARIMA	78207.48	74528
GHMM	**74350.18**	74528

Moving forward to 2023 (Table 5.3), the GHMM model demonstrates the closest prediction, estimating 74350.18 units against the observed 74,528 units, performing slightly closer to the actual value than the previous year. Likewise, linear regression and ARIMA provide a reasonably accurate forecast of 76,380 and 78,207 units, respectively. However, both Decision Tree and LSTM show slight disparities from the actual value, predicting values between 87522.0 and 83,752 units.

Table 5.4: Time Series Predicted Value vs Actual Value for June Year 2024 in Wetland Fluctuation. The closest predicted value of the Actual Value is highlighted in bold.

Model	Prediction of 2024	Actual of 2024
LiR	75750.48	87592
DTR	**87522.0**	87592
LSTM	83751.95	87592
ARIMA	72447.32	87592
GHMM	59211.81	87592

In the subsequent year, 2024 (Table 5.4), the Decision Tree model maintains its consistency by providing the closest prediction at 87,522 units. Conversely, both linear regression and ARIMA demonstrate deviations from the actual value, with forecasts ranging from 75,750 to 72,447 units. The LSTM model also deviates slightly, estimating 83,752 units. Notably, GHMM presents a significant variance from the actual value, forecasting 59,212 units.

It is noteworthy that although the performance of GHMM in predicting values for 2022, 2023, and 2024 was not as effective as other models, it outperformed all others in the normal prediction of 2025.

5.4 Result Analysis with Evaluation Metrices

In this section of our chapter, we delve into the various evaluation metrics utilized in unsupervised semantic segmentation of wetland area fluctuations. For the segmentation evaluation, we compared it with DEEPAqua in both qualitative and quantitative aspects.

5.4.1 Segmentation Quantitative Results

We explored many options to choose the right evaluation metrics for our case. For instance, in image segmentation tasks, metrics like Intersection over Union (IoU) and Dice Coefficient provide insights into the spatial accuracy of segmentations. At the same time, precision and recall balance the understanding of false positives and negatives. These metrics help identify how well the segmented outputs align with the actual wetland regions, capturing the extent and the intricacies of their boundaries.

Precision

Precision measures the accuracy of the positive predictions made by our proposed model, which tells us what proportion of the items the model labeled as positive are actually positive.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{5.18}$$

A high precision means when our model predicts a positive water pixels, it is very likely to be correct.

Recall

Recall, sensitivity, or the true positive rate, gauges the model's capacity to accurately detect all pertinent instances within the image dataset. It indicates the proportion of actual positives that the model successfully identified.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{5.19}$$

A high recall indicates that the model is successfully capturing most of the positive instances to find water pixels in ground truth.

F1-Score

The F1 score serves as the harmonic mean of precision and recall, offering a unified metric that weighs both aspects equally. Therefore, it proves particularly beneficial in situations where we need to give equal importance to both precision and recall.

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5.20}$$

Ranges from 0 to 1, with 1 being the best possible score. It helps to assess how well the model balances the trade-off between precision and recall, especially when one metric is low and the other is high.

Pixel Accuracy (PA)

Pixel Accuracy (PA) measures the proportion of correctly classified pixels out of the total pixels in our wetland images.

$$\text{Pixel Accuracy} = \frac{\text{Number of Correctly Classified Pixels}}{\text{Total Number of Pixels}} \tag{5.21}$$

High pixel accuracy indicates that a large proportion of the wetland image pixels are correctly segmented.

Intersection over Union (IoU)

Intersection over Union (IoU), the Jaccard Index, calculates the ratio of the intersection to the union of the predicted and ground truth areas. This provides a more nuanced understanding of the overlap between the predicted and actual segments.

$$\text{IoU} = \frac{\text{Intersection Area}}{\text{Union Area}} \tag{5.22}$$

A high IoU indicates a strong overlap between the predicted and ground truth segments, meaning the segmentation is accurate.

To assess the performance of our methods, we have compiled the quantitative evaluation results with Precision, Recall, F1 score, Pixel Accuracy (PA), and Intersection over Union (IoU) into a comprehensive Table 5.5. This table highlights the effectiveness of our approaches across various metrics, providing a clear and concise summary of the results.

Table 5.5: Semantic Segmentation Results of Various Models over Area in Bangladesh. The last row is the performance of our Proposed Model.

Model	Precision	Recall	F1 Score	PA	IoU
DeepAqua-NDWI	0.97	0.88	0.98	0.90	0.93
DeepAqua-MNDWI	0.97	0.85	0.95	0.89	0.92
DeepAqua-AWEI	0.97	0.84	0.98	0.85	0.91
DeepAqua-HRWI	0.97	0.86	0.97	0.88	0.92
RAM-Grounded-SAM	0.99	0.89	0.94	0.98	0.95

Table 5.5, where we have gathered all the data and put it together, assesses how well our segmentation model performs. We have compared our results with those of the

DeepAqua model to get a clear picture of how practical our approach is, especially when it comes to identifying different features in Bangladesh's landscapes.

The DeepAqua model shines in terms of precision, recall, and F1 scores across different indices. It can pinpoint water bodies using various spectral indices like NDWI (Normalized Difference Water Index), MNDWI (Modified Normalized Difference Water Index), AWEI (Automated Water Extraction Index), and HRWI (High-Resolution Water Index). It consistently achieves high precision scores, around 0.97, and recall scores above 0.85 for each index, showing it is great at accurately spotting water bodies. Moreover, its F1 scores, which measure the balance between precision and recall, are mostly above 0.95, indicating solid performance.

In comparison, our proposed combined model, RAM-Grounded-SAM, exhibits (in Table 5.5) superior performance with a precision of 0.99, a recall of 0.89, an F1 score of 0.94, a pixel accuracy (PA) of 0.98, and an Intersection over Union (IoU) of 0.95. Our model has attention mechanisms that allow dynamic focus on specific parts of the image that are most relevant to the task at hand instead of processing the entire image uniformly. We focus on IoU because it gives a fine-grained eval between predicted and ground truth.

5.4.2 Segmentation Qualitative Results

For the qualitative results, we tested our model on some images that DeepAqua used and got a segmentation of wetlands closer to the ground truth.

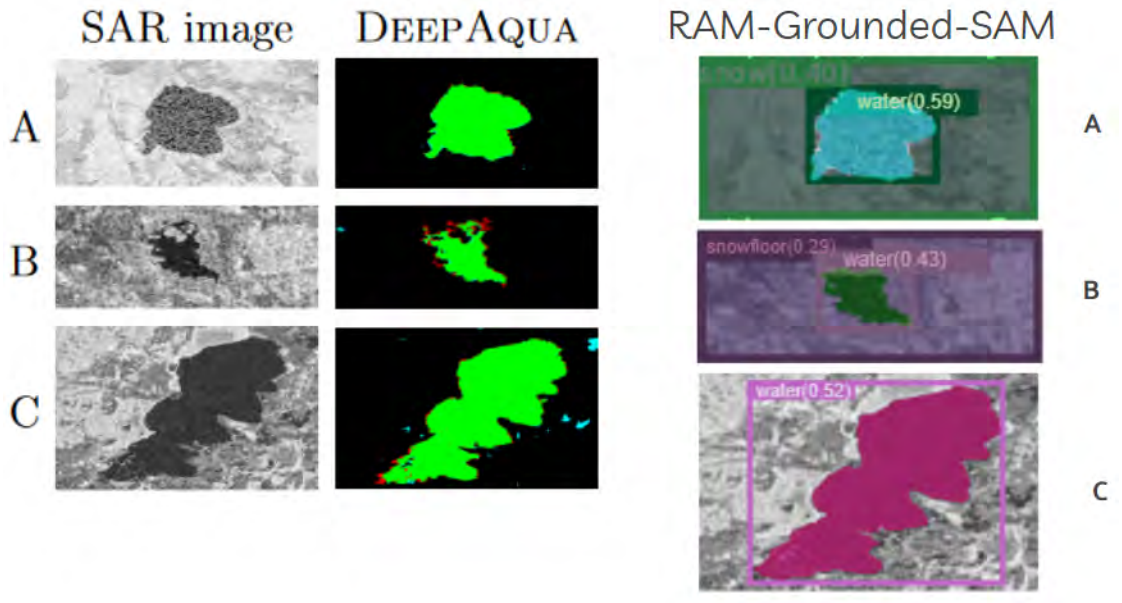


Figure 5.14: Qualitative Results of RAM-Grounded-SAM. The SAR image and DeepAqua segmentation results were achieved from the DeepAqua paper [42].

These results demonstrate that our model matches and surpasses DeepAqua in several key metrics, particularly intersection over union and segmentation.

Overall, while the DeepAqua model showcases impressive capabilities in semantic segmentation tasks, our RAM-Grounded-SAM model provides a more accurate and efficient solution. This superior performance highlights the potential of our model to offer more detailed and reliable insights into wetland dynamics, making a significant contribution to wetland conservation and management efforts in Bangladesh.

5.4.3 Time Series Evaluation Metrices

In the realm of time series analysis, particularly for forecasting wetland area fluctuations, metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE) are crucial. These metrics assess how closely the model's predictions align with observed data, highlighting the accuracy of the forecasts.

Mean Absolute Error (MAE)

Mean Absolute Error (MAE) measures the average magnitude of errors between predicted and observed values, calculating the absolute difference between each predicted value and its corresponding observed value.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|x_i - \hat{x}_i| \tag{5.23}$$

A lower MAE indicates better model performance, as it reflects smaller deviations between predicted and observed values.

Root Mean Squared Error (RMSE)

Root Mean Squared Error (RMSE) emphasizes larger errors by taking the square root of the average of the squared differences between predicted and observed values.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2} \tag{5.24}$$

Mean Absolute Percentage Error (MAPE)

Mean Absolute Percentage Error (MAPE) measures the average percentage difference between predicted and observed values, calculating the absolute percentage difference between each predicted value and its corresponding observed value, then averages these differences.

$$\text{MAPE} = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{5.25}$$

We have compiled the quantitative evaluation results of the time series evaluation into a comprehensive table. The table highlights the effectiveness of our approaches across various metrics, providing a clear and concise summary of the results.

Table 5.6: Time Series Prediction Evaluation Results of Various Methods. The best results are highlighted.

Model	Mean Absolute Error (MAE)	Root Mean Squared Error (RMSE)	Mean Absolute Percentage Error (MAPE)
LiR	20503.91	24901.10	22.24%
DTR	**17739.42**	**21477.54**	21.76%
LSTM	18067.34	22059.11	**21.19%**
ARIMA	18842.89	22909.65	21.39%
GHMM	21665.26	26342.08	22.99%

Looking at the results (Table 5.6), it is evident that each method has its strengths and weaknesses. The LSTM model, for example, shows a Mean Absolute Error (MAE) of 18067.34 and a Root Mean Squared Error (RMSE) of 22059.11, with a Mean Absolute Percentage Error (MAPE) of 21.19%. ARIMA shows a Mean Absolute Error (MAE) of 18842.89 and a Root Mean Squared Error (RMSE) of 22909.65, with a Mean Absolute Percentage Error (MAPE) of 21.39%. Similarly, the Decision Tree Regressor and Linear regression models also demonstrate competitive performance, 21.76% and 22.24%, respectively. With all the evaluation calculations, we get about 80 percent accurate results.

Interestingly, the Gaussian Hidden Markov Model (GHMM) has slightly higher error metrics than the other models. However, it is worth noting that GHMM previously provided a promising prediction for next year's fluctuation, indicating its potential to capture subtle patterns and nuances in the data that other models might miss.

Considering all factors, including the GHMM's previous success in prediction, we can confidently say that the Gaussian Hidden Markov Model gave us a prediction result of 73654.86 and is the best choice for our data. This indicates that in Tanguar Haor, a fluctuation will decrease in June 2025. Despite its marginally higher error metrics in this evaluation, its ability to capture complex temporal dependencies and provide accurate predictions for future fluctuations makes it a valuable tool for our analysis.

Chapter 6

Conclusion and Future Work

Using unsupervised semantic segmentation helps identify the wetland's condition and assess its shrinkage and expansion. Our research is a contribution to the restoration and conservation of the wetland.

The primary objective of this unsupervised semantic segmentation research is to identify and pinpoint meaningful objects within collections of images without using any annotations. The algorithm must generate features for each pixel that take control of semantic significance and are sufficiently concise to establish detectable clusters. Several studies have introduced semantic segmentation systems that can learn from less precise label forms, including classes, tags, bounding boxes, scribbles, or point annotations [17]. Nevertheless, a limited number of studies address the task of semantic segmentation without relying on human supervision or motion cues. Efforts such as Independent Information Clustering (IIC) [16] and PiCIE [23] strive to acquire semantically significant features by ensuring transformation equivariance. Additionally, these approaches incorporate a clustering process to enhance the efficiency of the acquired features. The absence of prior knowledge in computer vision necessitates the training of models in order to achieve desired outcomes. However, using unsupervised semantic methods has facilitated attaining desired results without needing labeled data. This method does not necessitate manual annotation as it automatically assigns it.

This research begins with the initiative to explore the world of unsupervised semantic segmentation using a remarkable computational technique that promises to figure out the complex puzzle of wetland contraction and expansion. We can learn and understand the natural semantic information within the wetland image by utilizing this model.

Our objective is to prepare and formulate a precise and resilient methodology that facilitates the identification and categorization of wetland objects in our nation. The main objective is to enhance the understanding of wetland dynamics among land managers, environmental scientists, and policymakers, helping them make informed decisions regarding the restoration and conservation of wetlands within our nation.

6.1 Future Work

We aim to further enhance our combined model approach with further modification to increase its capability to detect wetland areas.

Firstly, we plan to incorporate more diverse datasets around Bangladesh and worldwide to improve our model's generalizability in different wetland environments. We also intend to integrate real-time data feeds from satellite imagery to enable dynamic monitoring and forecasting of wetland changes.

In addition to these improvements, we will fine-tune the RAM Tag2Text model to function with a specially designed Bangla tag list. This fine-tuning process will involve adjusting and training the model parameters on Bangla-specific data and transforming visual data into text descriptions in tags and captions. This will enhance the model's ability to generate accurate and contextually relevant descriptions for the local ecosystem.

Lastly, we will focus on publishing our findings in scientific journals and presenting our work at relevant conferences to contribute to the broader scientific community. Through collaboration with environmental agencies and research institutions, we hope to apply our research in practical conservation and management efforts, ultimately aiding in preserving critical wetland ecosystems.

Bibliography

[1] F. G. Hayden, "Wetlands provisions in the 1985 and 1990 farm bills," *Journal of Economic Issues*, vol. 24, no. 2, pp. 575–587, 1990, ISSN: 00213624. [Online]. Available: http://www.jstor.org/stable/4226296 (visited on 09/18/2023).

[2] D. Hinrichsen, *The environmental impact of wetland destruction and deforestation*, 1999. [Online]. Available: https://www.123helpme.com/essay/The-Environmental-Impact-of-Wetland-Destruction-and-26916.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[4] U. Habiba, F. Haider, A. Ishtiaque, M. S. Mahmud, and A. Masrur, "Remote sensing amp;amp; gis based spatio-temporal change analysis of wetland in dhaka city, bangladesh," *Journal of Water Resource and Protection*, vol. 03, no. 11, pp. 781–787, Oct. 2011. DOI: 10.4236/jwarp.2011.311088.

[5] M. S. Hassan and S. Mahmud-ul-islam, *Identification of Wetland Restoration Areas of Chalan Beel in Sirajganj District, Bangladesh using Integrated GIS and Remote Sensing*, Nov. 2014. DOI: 10.7537/j.0711.1554-0200.

[6] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille, "The role of context for object detection and semantic segmentation in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[7] X. C. R. M. X. L. S. F. R. U. A. Yuille, "Pascal-part dataset," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. [Online]. Available: https://paperswithcode.com/dataset/pascal-person-part.

[8] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, *Microsoft coco: Common objects in context*, 2015. arXiv: 1405.0312 [cs.CV].

[9] Z. Liu, X. Li, P. Luo, C.-C. Loy, and X. Tang, "Semantic image segmentation via deep parsing network," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1377–1385.

[10] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[11] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li, 2016. [Online]. Available: https://webscope.sandbox.yahoo.com/catalog.php?datatype=i&did=67.

[12] ——, "Yfcc100m," *Communications of the ACM*, vol. 59, no. 2, pp. 64–73, 2016. DOI: 10.1145/2812802.

[13] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan, "Deep adaptive image clustering," in *Proceedings of the IEEE International Conference on computer vision*, 2017, pp. 5879–5887.

[14] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[15] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," *Computer Vision – ECCV 2018*, pp. 139–156, Jul. 2018. DOI: 10.1007/978-3-030-01264-9_9.

[16] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2018.

[17] A. Zlateski, R. Jaroensri, P. Sharma, and F. Durand, "On the importance of label quality for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2018.

[18] D. Civitarese, D. Szwarcman, E. V. Brazil, and B. Zadrozny, "Semantic segmentation of seismic images," *arXiv preprint arXiv:1905.04307*, 2019.

[19] P. Haeusser, J. Plapp, V. Golkov, E. Aljalbout, and D. Cremers, "Associative deep clustering: Training a classification network with no labels," in *Pattern Recognition: 40th German Conference, GCPR 2018, Stuttgart, Germany, October 9-12, 2018, Proceedings 40*, Springer, 2019, pp. 18–32.

[20] Y. Hu, J. Zhang, Y. Ma, X. Li, Q. Sun, and J. An, "Deep learning classification of coastal wetland hyperspectral image combined spectra and texture features: A case study of huanghe (yellow) river estuary wetland," *Acta Oceanologica Sinica*, vol. 38, pp. 142–150, 2019.

[21] X. Ji, J. F. Henriques, and A. Vedaldi, *Invariant information clustering for unsupervised image classification and segmentation*, 2019. arXiv: 1807.06653 [cs.CV].

[22] X. Qi, *Mlrsnet: A multi-label high spatial resolution remote sensing dataset for semantic scene understanding*, Jul. 2020. [Online]. Available: https://data.mendeley.com/datasets/7j9bv9vwsx/2.

[23] Z. Ren, Z. Yu, X. Yang, M.-Y. Liu, A. G. Schwing, and J. Kautz, *Ufo^2: A unified framework towards omni-supervised object detection*, 2020. arXiv: 2010.10804 [cs.CV].

[24] M. Schmitt, J. Prexl, P. Ebel, L. Liebel, and X. X. Zhu, "Weakly supervised semantic segmentation of satellite images for land cover mapping–challenges and opportunities," *arXiv preprint arXiv:2002.08254*, 2020.

[25] J. H. Cho, U. Mall, K. Bala, and B. Hariharan, *Picie: Unsupervised semantic segmentation using invariance and equivariance in clustering*, 2021. arXiv: 2103.17070 [cs.CV].

[26] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, *An image is worth 16x16 words: Transformers for image recognition at scale*, 2021. arXiv: 2010.11929 [cs.CV].

[27] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, *Masked autoencoders are scalable vision learners*, 2021. arXiv: 2111.06377 [cs.CV].

[28] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, *Lora: Low-rank adaptation of large language models*, 2021. arXiv: 2106.09685 [cs.CL].

[29] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, *Learning transferable visual models from natural language supervision*, 2021. arXiv: 2103.00020 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2103.00020.

[30] M. Eliasof, N. B. Zikri, and E. Treister, "Unsupervised image semantic segmentation through superpixels and graph neural networks," *arXiv preprint arXiv:2210.11810*, 2022.

[31] ——, "Unsupervised image semantic segmentation through superpixels and graph neural networks," *arXiv preprint arXiv:2210.11810*, 2022.

[32] M. Hamilton, Z. Zhang, B. Hariharan, N. Snavely, and W. T. Freeman, "Unsupervised semantic segmentation by distilling feature correspondences," *arXiv preprint arXiv:2203.08414*, 2022.

[33] H. Jafarzadeh, M. Mahdianpari, E. W. Gill, B. Brisco, and F. Mohammadimanesh, "Remote sensing and machine learning tools to support wetland monitoring: A meta-analysis of three decades of research," *Remote Sensing*, vol. 14, no. 23, 2022, ISSN: 2072-4292. DOI: 10.3390/rs14236104. [Online]. Available: https://www.mdpi.com/2072-4292/14/23/6104.

[34] S. F. Bhat, R. Birkl, D. Wofk, P. Wonka, and M. Müller, *Zoedepth: Zero-shot transfer by combining relative and metric depth*, 2023. arXiv: 2302.12288 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2302.12288.

[35] U. EPA, *What is a wetland? / us epa*, May 2023. [Online]. Available: https://www.epa.gov/wetlands/what-wetland.

[36] S. Gao, Z.-Y. Li, M.-H. Yang, M.-M. Cheng, J. Han, and P. Torr, "Large-scale unsupervised semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 6, pp. 7457–7476, Jun. 2023. DOI: 10.1109/tpami.2022.3218275. [Online]. Available: https://doi.org/10.1109%2Ftpami.2022.3218275.

[37] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, *Segment anything*, 2023. arXiv: 2304.02643 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2304.02643.

[38] K. Li, Z. Wang, Z. Cheng, R. Yu, Y. Zhao, G. Song, C. Liu, L. Yuan, and J. Chen, "Acseg: Adaptive conceptualization for unsupervised semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7162–7172.

[39] H. Liu, C. Li, Q. Wu, and Y. J. Lee, *Visual instruction tuning*, 2023. arXiv: 2304.08485 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2304.08485.

[40] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, and L. Zhang, *Grounding dino: Marrying dino with grounded pre-training for open-set object detection*, 2023. arXiv: 2303.05499 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2303.05499.

[41] Z. N/A, *Why are wetlands important to people and planet?* Feb. 2023. [Online]. Available: https://www.zurich.com/en/media/magazine/2022/why-we-should-care-about-and-protect-our-wetlands.

[42] F. J. Peña, C. Hübinger, A. H. Payberah, and F. Jaramillo, "Deepaqua: Self-supervised semantic segmentation of wetlands from sar images using knowledge distillation," *arXiv preprint arXiv:2305.01698*, 2023.

[43] ——, *Deepaqua: Self-supervised semantic segmentation of wetland surface water extent with sar images using knowledge distillation*, 2023. arXiv: 2305.01698 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2305.01698.

[44] H. S. Seong, W. Moon, S. Lee, and J.-P. Heo, "Leveraging hidden positives for unsupervised semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 19 540–19 549.

[45] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, *Sigmoid loss for language image pre-training*, 2023. arXiv: 2303.15343 [cs.CV].

[46] Y. Zhang, X. Huang, J. Ma, Z. Li, Z. Luo, Y. Xie, Y. Qin, T. Luo, Y. Li, S. Liu, Y. Guo, and L. Zhang, *Recognize anything: A strong image tagging model*, 2023. arXiv: 2306.03514 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2306.03514.

[47] X. Lai, Z. Tian, Y. Chen, Y. Li, Y. Yuan, S. Liu, and J. Jia, *Lisa: Reasoning segmentation via large language model*, 2024. arXiv: 2308.00692 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2308.00692.

[48] [Online]. Available: https://www.ecohubmap.com/hot-spot/drying-up-of-teesta-river-bangladesh/113jt3klkgqg190.

[49] *Google earth*. [Online]. Available: https://earth.google.com/web/.

[50] HuggingFace. [Online]. Available: https://huggingface.co/docs/transformers/en/model_doc/swin.

[51] ——, [Online]. Available: https://huggingface.co/docs/transformers/en/model_doc/vit.

[52] N. Interactive, *Live weather map hurricane tracker*. [Online]. Available: https://zoom.earth/.

[53] *Paligemma demo - a hugging face space by big-vision*. [Online]. Available: https://huggingface.co/spaces/big-vision/paligemma.