Satellite-Based Water Body Detection and Trend Analysis in Vellore Using Deep Learning

PMCA699J - Dissertation-II

Submitted in partial fulfillment of the requirements for the degree of

Master of Computer Applications
in
Department of Computer Applications

by
Sayantan Bhattacharyya
23MCA0304

Under the guidance of

Dr. Jagannathan J.
Assistant Professor, Grade-I
School of Computer Science Engineering and Information Systems
VIT, Vellore

April, 2025

Satellite-Based Water Body Detection and Trend Analysis in Vellore Using Deep Learning

PMCA699J - Dissertation-II

Submitted in partial fulfillment of the requirements for the degree of

Master of Computer Applications

in

Department of Computer Applications

by

Sayantan Bhattacharyya

23MCA0304

Under the guidance of

Dr. Jagannathan J

Assistant Professor, Senior Grade-I

School of Computer Science Engineering and Information Systems

VIT, Vellore

VIT

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

April, 2025

DECLARATION

I thereby declare that the **PMCA699J – Dissertation-II** entitled "**Satellite-Based Water Body Detection and Trend Analysis in Vellore Using Deep Learning**" submitted by me, for the award of the degree of *Master of Computer Applications in Department of Computer Applications, School of Computer Science Engineering and Information Systems* to VIT is a record of bonafide work carried out by me under the supervision of **Dr. Jagannathan J.** Assistant Professor Senior Grade -I, SCORE, VIT, Vellore.

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

Place: Vellore

Date:

Signature of the Candidate

CERTIFICATE

This is to certify that the **PMCA699J - Dissertation-II** entitled "**Satellite-Based Water Body Detection and Trend Analysis in Vellore Using Deep Learning**" submitted **Sayantan Bhattacharyya & 23MCA0304**, SCORE, VIT, for the award of the degree of *Master of Computer Applications in Department of Computer Applications*, is a record of bonafide work carried out by him under my supervision during the period, 13. 12. 2024 to 17.04.2025, as per the VIT code of academic and research ethics.

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

Place: Vellore
Date:

Signature of the VIT-SCORE - Guide

Internal Examiner **External Examiner**

Head of the Department
Department of Computer Applicati

ACKNOWLEDGEMENT

It is my pleasure to express with a deep sense of gratitude to my **Dissertation- II** guide **Dr. Jagannathan J.,** Assistant Professor, Senior Grade-I, School of Computer Science Engineering and Information Systems, Vellore Institute of Technology, Vellore for his constant guidance, continual encouragement, in my endeavor. My association with him is not confined to academics only, but it is a great opportunity on my part to work with an intellectual and an expert in the field of **Deep Learning.**

I would like to express my heartfelt gratitude to Honorable Chancellor **Dr. G Viswanathan**; respected Vice Presidents **Mr. Sankar Viswanathan**, **Dr. Sekar Viswanathan**, Vice Chancellor **Dr. V. S. Kanchana Bhaaskaran**; Pro-Vice Chancellor **Dr. Partha Sharathi Mallick**; and Registrar **Dr. Jayabarathi T**.

My whole-hearted thanks to Dean **Dr. Daphne Lopez**, School of Computer Science Engineering and Information Systems, Head, Department of Computer Applications **Dr. E Vijayan**, MCA Project Coordinator **Dr. Karthikeyan P**, SCORE School Project Coordinator **Dr. Thandeeswaran R**, all faculty, staff and members working as limbs of our university for their continuous guidance throughout my course of study in unlimited ways.

It is indeed a pleasure to thank my parents and friends who persuaded and encouraged me to take up and complete my **dissertation** successfully. Last, but not least, I express my gratitude and appreciation to all those who have helped me directly or indirectly towards the successful completion of the **dissertation**.

Place: Vellore

Date: **Sayantan Bhattacharyya**

Executive Summary

Water is a critical resource, and its availability directly impacts agriculture, ecosystems, and human livelihoods. This project focuses on automated water body detection and trend analysis in Vellore using satellite imagery and deep learning techniques to monitor water resources effectively. The Water Body Detection component leverages a U-Net deep learning model trained on Sentinel-2 satellite images to accurately identify and segment water bodies. Preprocessing techniques, including noise removal, atmospheric correction, and cloud masking, are applied to enhance detection accuracy, and the model is evaluated using key metrics such as IoU and accuracy to ensure reliability. The Water Body Trend Analysis component examines long-term variations in water spread using Landsat and Sentinel-2 datasets. Various statistical and trend analysis techniques, including seasonal decomposition, regression analysis, and anomaly detection, are used to study changes in water body extent, helping to understand seasonal and annual fluctuations and identify factors affecting water availability. The findings from this study can support policymakers, environmentalists, and urban planners in making informed decisions regarding water conservation, resource planning, and climate adaptation strategies. Future improvements include real-time monitoring, AI-based forecasting, and expanding the study to a broader geographical area to enhance water resource management.

Keywords: Satellite Imagery, Water Body Detection, Trend Analysis, Deep Learning, Vellore District, Temporal Changes, Water Resource Management, Environmental Monitoring, Automated Analysis, U NET

CONTENTS

List of Figures

(In the chapters, figure caption should come below the figure and table caption should come above the table. Figure and table captions should be of font size 10.

List of Tables

List of Abbreviations

3GPP	Third Generation Partnership Project
2G	Second Generation
3G	Third Generation
4G	Fourth Generation
AWGN	Additive White Gaussian Noise

Symbols and Notations

δf	CFO
ε	NCFO

CHAPTER 1

INTRODUCTION

Water is one of the most critical natural resources, essential for sustaining life, agriculture, industry, and ecosystems. Monitoring water bodies is crucial for understanding environmental changes, managing water resources effectively, and mitigating the impact of climate change and urbanization. In recent years, advancements in satellite remote sensing and deep learning have opened new possibilities for automating water body detection and analyzing long-term trends.

This project focuses on **detecting water bodies in the Vellore region** using deep learning techniques and analyzing **historical trends** to understand changes in water extent over the past three decades. By leveraging satellite imagery from Sentinel-2 and Landsat, combined with machine learning techniques, this study aims to provide **accurate and efficient water body detection and trend analysis**. The insights gained from this project can support environmental management, water conservation efforts, and policy-making for sustainable development.

This work consists of two major components:

1. **Water Body Detection**: Using deep learning-based segmentation techniques (U-Net model) to identify and extract water bodies from satellite imagery.

2. **Water Body Trend Analysis**: Analyzing **seasonal, yearly, and long-term** variations in water body extent to understand patterns, influences, and environmental changes in the Vellore region.

Through these components, the project provides a **comprehensive framework for water body monitoring**, offering insights that can help policymakers, researchers, and environmentalists in effective water resource planning.

1.1 Objective

The main objectives of this project are:

- To develop an **automated deep learning-based system** for detecting water bodies using satellite imagery.

- To analyze **historical water body trends** over the past years in the Vellore region.
- To study the impact of **seasonal changes, urbanization, and climate variability** on water body extent.
- To provide **data-driven insights** for water conservation and environmental planning.

By fulfilling these objectives, the project aims to bridge the gap between remote sensing, deep learning, and environmental analysis for practical applications in **water resource management**

1.2 Motivation

The increasing **climate variability, urban expansion, and water scarcity** have made **water body monitoring** an urgent necessity. Vellore, like many other regions, experiences **fluctuations in water availability**, which impact agriculture, drinking water supply, and ecosystem stability.

Why is this important?

- **Water scarcity is a growing challenge:** Understanding long-term changes in water bodies helps in predicting **droughts and water shortages**.
- **Deep learning can improve detection accuracy:** Traditional methods like NDWI (Normalized Difference Water Index) are limited in their ability to detect water bodies under varying environmental conditions. A **U-Net deep learning model** provides higher accuracy and automation.
- **Long-term trend analysis helps in planning:** Seasonal and historical variations can reveal patterns that help in **policy-making and infrastructure development**.

By integrating **remote sensing, deep learning, and statistical trend analysis**, this project aims to provide **accurate and actionable insights** for sustainable water management.

1.3 Background

Water bodies are essential for sustaining ecosystems, agriculture, human settlements, and overall environmental balance. However, rapid urbanization, climate change, and shifting precipitation patterns have led to significant changes in water body extents, making monitoring and analysis critical for sustainable water resource management. Traditional methods of water body mapping are often time-consuming, require manual intervention, and lack real-time adaptability.

This project leverages advanced deep learning techniques and satellite-based remote sensing to enhance water body monitoring and trend analysis. The Water Body Detection component utilizes a U-Net model trained on Sentinel-2 satellite imagery to accurately segment and identify water bodies. This automated system enables efficient, large-scale monitoring with minimal human intervention. In parallel, the Water Body Trend Analysis uses historical water spread data from Bhuvan (ISRO) to study monthly and yearly fluctuations in water body coverage across Vellore. By integrating deep learning and geospatial data analytics, the project provides a holistic approach to understanding water body dynamics, identifying seasonal trends, and assessing potential environmental and anthropogenic impacts. These insights can support water resource planning, climate impact studies, disaster mitigation strategies, and urban development planning.

1.4 Scope

The scope of this project extends beyond simple water body detection and trend analysis—it establishes a foundation for scalable, automated water resource monitoring. The **Water Body Detection** component is designed to be adaptable, allowing for application across various geographic regions by training the deep learning model on different datasets. This capability is beneficial for environmental monitoring agencies, hydrologists, and researchers studying water distribution at both local and global scales.

The **Water Body Trend Analysis** focuses on understanding how water bodies in Vellore have changed over time, using verified datasets from Bhuvan (ISRO). This analysis can help predict water availability trends, assess the impact of climate variability, and support sustainable water management policies. Additionally, the findings can be used by government agencies, urban planners, and policymakers to implement water conservation strategies, mitigate the effects of droughts or floods, and improve land-use planning.

Beyond Vellore, the methodologies developed in this project can be extended to other regions, facilitating large-scale hydrological studies. The integration of AI-based detection with real-world trend analysis provides a powerful tool for ongoing environmental monitoring and decision-making in the face of climate change and growing water scarcity challenges.

CHAPTER 2

DISSERTATION DESCRIPTION AND GOALS

2.1 Dissertation Description

Water bodies, including lakes, reservoirs, and rivers, play a crucial role in maintaining ecological balance, supporting biodiversity, and meeting human water consumption needs. Monitoring water bodies over time is essential for water resource management, urban planning, and environmental conservation. However, traditional methods of water body detection and trend analysis often rely on manual surveys, which are time-consuming, expensive, and limited in scale. The increasing availability of satellite imagery and advancements in deep learning provide an opportunity to automate water body monitoring and analyze long-term trends with high accuracy and efficiency.

This project is structured into two primary components:

1. **Water Body Detection using Deep Learning**

   The first part of the project involves **developing a deep learning-based water body detection model**. A **U-Net convolutional neural network (CNN)** is trained on **Sentinel-2 satellite images**, using high-resolution imagery along with water body masks. The model learns to distinguish water bodies from other land features and can generate accurate water masks from satellite images. This automated detection process significantly improves efficiency compared to traditional classification methods, enabling large-scale, real-time monitoring of water bodies.

2. **Water Body Trend Analysis**

   The second component focuses on **trend analysis using historical water body data**. **Bhuvan (ISRO) provides month-year water spread area data** for Vellore, which is used to study how water bodies have changed over time. The analysis explores:

   - **Seasonal variations** in water body spread area (dry vs. wet seasons).
   - **Yearly trends**, identifying periods of expansion and contraction.
   - **Potential causes of fluctuations**, such as rainfall patterns, climate

change, and urbanization.

By combining deep learning-based water body detection with long-term trend analysis, the project provides a **comprehensive and automated approach** to water resource monitoring. The insights derived from this study can be valuable for environmentalists, urban planners, and policymakers working on water conservation and management strategies.

2.2 Dissertation Goals

This project aims to address key challenges in water body monitoring and analysis through deep learning and remote sensing. The primary goals include:

1. **Develop an Automated Water Body Detection System**
    - Train a **U-Net deep learning model** to accurately segment water bodies from Sentinel-2 satellite images.
    - Reduce reliance on manual classification methods by automating the detection process.
2. **Analyze Water Body Trends Over Time**
    - Utilize **Bhuvan (ISRO) data** to study the changes in **current and maximum water spread area** over time.
    - Identify **seasonal and yearly patterns** in water body extent.
3. **Support Water Resource Management and Urban Planning**
    - Provide actionable insights for policymakers, urban developers, and environmental researchers.
    - Assist in planning sustainable water resource management strategies based on historical trends.
4. **Enhance the Use of AI in Remote Sensing Applications**
    - Demonstrate the **effectiveness of deep learning** in analyzing satellite

imagery for environmental monitoring.

- Develop a scalable methodology that can be extended to monitor water bodies in other regions.

5. **Promote Sustainable Water Conservation Practices**

   - Understand the impact of climate change, urbanization, and human activities on water bodies.

   - Provide scientific data that can aid in policy decisions related to water conservation.

This project bridges the gap between **AI-based environmental monitoring** and **practical water resource management**, offering a **data-driven, scalable, and efficient** solution for studying and preserving water bodies.

CHAPTER 3

TECHNICAL SPECIFICATION

3.1 Programming Languages & Libraries:

This project harnesses the power of several key programming languages and specialized libraries to achieve accurate deep learning-based water body detection and robust trend analysis. Let's delve deeper into the role and significance of each component:

Programming Language: Python - The Orchestrator

Python serves as the central nervous system of this project. Its selection stems from its rich ecosystem of scientific and deep learning libraries, its intuitive syntax which promotes rapid development, and its large and active community providing extensive support and resources. Python's versatility allows for seamless integration of diverse tasks, from handling geospatial data to building and training complex neural networks, and finally to visualizing the results in a clear and understandable manner. Its interpreted nature facilitates iterative development and debugging, crucial in an evolving research project like this.

Libraries Used: The Specialized Toolset

1. **TensorFlow & Keras: The Deep Learning Engine** TensorFlow, backed by Google, forms the powerful backend for our deep learning endeavors, providing a comprehensive framework for numerical computation and large-scale machine learning. Keras acts as a high-level API, elegantly simplifying the process of building and experimenting with neural networks. Together, they empower us to construct and train the U-Net model, a sophisticated architecture specifically designed for semantic segmentation tasks like identifying water bodies in satellite imagery. TensorFlow handles the intricate computational graph, optimization algorithms (such as Adam or SGD), and efficient memory management, while Keras allows us to define the U-Net's architecture through intuitive layers (convolutional layers for feature extraction, activation functions like ReLU to introduce non-linearity, pooling layers for downsampling, and skip connections to preserve fine-grained details). The inherent support for GPU

acceleration in TensorFlow significantly reduces the training time for our model, enabling us to process the vast amounts of satellite data efficiently. We utilize carefully chosen loss functions within Keras (e.g., Binary Cross-Entropy or Dice Loss) that are specifically tailored for image segmentation to guide the model's learning process in accurately delineating water bodies.

2. **OpenCV & PIL (Pillow): Image Preprocessing and Enhancement** Before satellite images can be effectively analyzed by our U-Net model, they often require meticulous preprocessing. OpenCV (Open Source Computer Vision Library) plays a vital role here, offering a wide array of functions for image manipulation. This includes tasks like resizing images to a consistent input size suitable for the model, applying filters to reduce noise and enhance relevant features, and potentially performing image transformations for data augmentation to improve the model's generalization ability. PIL (Pillow), a user-friendly image processing library, complements OpenCV by providing essential tools for loading images from various file formats, converting between different image representations, and performing basic pixel-level manipulations. These libraries ensure that the input data fed to our deep learning model is standardized, cleaned, and optimized for effective learning.

3. **Pandas & NumPy: Data Wrangling and Numerical Powerhouse** The analysis of water body trends often involves working with structured data, such as the water spread area data potentially obtained from platforms like Bhuvan (ISRO). Pandas, a powerful library for data manipulation and analysis, provides us with efficient data structures (like DataFrames) to handle this tabular data. We can use Pandas to clean the data, handle missing values, merge different datasets (e.g., combining water spread data with temporal information), and prepare it for statistical analysis and visualization. Underneath the hood, NumPy (Numerical Python) provides the fundamental building blocks for numerical computations in Python. Its efficient multi-dimensional array object and a vast collection of mathematical functions are essential for both image processing (as images are essentially multi-dimensional arrays of pixel values) and the underlying computations within the deep learning model. NumPy enables fast and vectorized operations, which are crucial for handling the large

datasets involved in this project.

4. **Matplotlib & Seaborn: Visualizing Insights** Effective communication of the project's findings relies heavily on data visualization. Matplotlib provides a foundational library for creating static, interactive, and animated visualizations in Python. Seaborn builds on top of Matplotlib, offering a higher-level interface with aesthetically pleasing default styles and specialized plot types particularly useful for statistical data visualization. We leverage these libraries to generate informative graphs that illustrate the seasonal and yearly trends in water body extent. This might include time series plots showing fluctuations over time, bar charts comparing water body areas across different years or seasons, and potentially even visualizing the geographical distribution of water bodies using scatter plots or heatmaps. These visualizations are critical for interpreting the results of our analysis and conveying the impact of the project effectively.

5. **Rasterio: Handling Geospatial Raster Data** Our primary source of information about water bodies comes from satellite imagery, often in the form of geospatial raster files. Rasterio is a crucial library that enables us to read, write, and manipulate these raster datasets, such as Sentinel-2 imagery. It provides a robust and efficient way to access the individual spectral bands within a multispectral image (e.g., red, green, blue, near-infrared), which carry different information about the Earth's surface. Rasterio also handles crucial geospatial information embedded within these files, such as the coordinate reference system (CRS), ensuring that our analysis is spatially accurate. By allowing us to work directly with the spectral information of satellite images, Rasterio enhances the accuracy of our water body detection model, as different materials (like water) exhibit unique spectral signatures.

6. **SciPy: Statistical Analysis and Optimization** To gain deeper insights into the trends observed in water body extent, we employ the statistical capabilities offered by SciPy (Scientific Python). This library provides a wide range of functions for statistical analysis, including techniques for curve fitting to model the temporal changes in water body area, statistical tests to determine the significance of observed trends (e.g., identifying if a decrease in water body size is statistically significant), and optimization algorithms that could be useful in

refining model parameters or analyzing complex patterns in the data. SciPy empowers us to go beyond simple observation and perform rigorous statistical analysis to understand the underlying dynamics of water resource fluctuations.

By thoughtfully integrating these programming languages and libraries, this project establishes a comprehensive and efficient pipeline. This pipeline seamlessly manages every stage, from the initial acquisition and preprocessing of raw satellite imagery to the advanced deep learning-based detection of water bodies and the final, in-depth statistical analysis of their evolving trends. This robust framework provides a powerful tool for effective water resource monitoring and management.

3.2 Software Specification

The development, training, and analysis phases of this project are conducted within a carefully selected software environment, each component chosen for its unique capabilities and contributions:

Development Environment: Google Colab - The Cloud-Powered Workbench

Google Colaboratory (Colab) serves as the primary development environment for this project. This cloud-based platform offers a highly advantageous setting for deep learning projects due to several key features. Firstly, it provides free access to powerful computational resources, including GPUs and even TPUs (Tensor Processing Units), which are crucial for accelerating the computationally intensive tasks of training complex deep learning models like the U-Net. Colab comes pre-loaded with many of the essential data science and machine learning libraries, including TensorFlow, Keras, PyTorch, OpenCV, Pandas, and NumPy, eliminating the need for extensive setup and configuration. Its notebook-based interface allows for an interactive and organized workflow, enabling us to seamlessly write and execute code, visualize intermediate results, and document our progress. Furthermore, Colab's collaborative nature facilitates teamwork and easy sharing of code and results, streamlining the development and debugging process. The integration with Google Drive allows for easy storage and access of datasets and trained models, making it a highly efficient and accessible platform for our project.

GIS Software (for visualization): QGIS - The Geospatial Insight Generator

QGIS (Quantum GIS) is employed as the primary Geographic Information System (GIS) software for the visualization and spatial analysis aspects of this project. As a free and open-source software, QGIS offers a robust set of tools for working with geospatial data. Once the water bodies are detected using our deep learning model, the resulting data, often in raster or vector format, can be seamlessly imported into QGIS. Here, we can leverage its powerful mapping and visualization capabilities to display the detected water bodies on top of the original satellite imagery or other relevant geographical layers. QGIS enables us to create informative maps that clearly highlight the spatial distribution of water bodies, making it easier to understand their geographical context and extent. Its analytical tools also allow for further spatial analysis, such as calculating the area of detected water bodies, comparing water body distributions over different time periods, and potentially integrating this information with other spatial datasets to gain deeper insights. The ability to create high-quality maps and perform spatial queries makes QGIS an indispensable tool for effectively communicating the project's findings to a wider audience and for in-depth geospatial analysis.

Google Earth Engine: The Satellite Imagery Gateway

Google Earth Engine (GEE) plays a critical role in providing access to a vast archive of publicly available satellite imagery, including datasets like Landsat and Sentinel-2, which are essential for our water body detection project. This cloud-based platform offers petabyte-scale data and powerful computational resources for analyzing geospatial data at a global scale. We utilize GEE to efficiently search, filter, and download relevant satellite images for our study area over specific time periods. The platform's built-in tools allow for on-the-fly preprocessing of imagery, such as atmospheric correction and mosaicking, potentially simplifying some of the initial data preparation steps. Furthermore, Earth Engine's interactive code editor enables us to explore the available imagery, perform preliminary analyses, and even implement parts of our data preprocessing pipeline before downloading the data for model training in Google Colab. The sheer volume of data and the powerful analytical capabilities of Google Earth Engine make it an invaluable resource for acquiring the necessary satellite imagery and conducting initial investigations for our water body detection and trend analysis project.

3.3 Dataset Specification

This project relies on two distinct yet complementary datasets to achieve its objectives of deep learning-based water body detection and subsequent trend analysis:

1. **Sentinel-2 Satellite Images (for Deep Learning Model): High-Resolution Multispectral Imagery**

    - **Source: Google Earth Engine:** We leverage the extensive catalog of Sentinel-2 imagery readily available on the Google Earth Engine platform. This provides us with efficient access to a large volume of high-quality satellite data without the need for extensive downloading and storage management.

    - **Data Type: Multispectral satellite images:** Sentinel-2 satellites capture images across multiple spectral bands, each sensitive to different parts of the electromagnetic spectrum. This wealth of information allows for a more comprehensive analysis of the Earth's surface features.

    - **Bands Used: Red, Green, Blue, Near Infrared (NIR):** We specifically utilize these four key bands for our deep learning model. The Red, Green, and Blue bands provide information similar to what the human eye sees, aiding in the visual identification of features. The Near-Infrared (NIR) band is particularly crucial for water body detection because water strongly absorbs NIR radiation, making water bodies appear very dark in this band. This spectral signature is a key differentiator for the model to learn.

    - **Resolution: 10m per pixel:** The spatial resolution of 10 meters per pixel means that each pixel in the image represents an area of 10x10 meters on the ground. This relatively high resolution allows the model to discern even smaller water bodies with reasonable accuracy, providing a detailed view of water resources.

    - **Label Data: Water body masks (binary images):** To train our U-Net model, we utilize a **collection of 2841** water bodies images captured by the Sentinel-2 Satellite. Each image is paired with a black and white mask where white pixels represent water and black pixels represent all other land cover. These masks were automatically generated by calculating the **NDWI (Normalized Difference Water Index)**. While NDWI is commonly used to detect and measure vegetation, we employed a higher threshold to specifically identify water bodies in the satellite imagery. This process allows us to create a robust training dataset for our deep learning model.

- **Dataset Size:** The training dataset for the U-Net model consists of **2841 pairs of satellite images and their corresponding water body masks.**

2. **Bhuvan (ISRO) Water Spread Area Data (for Trend Analysis): Official Temporal Measurements**

    - **Source: Bhuvan ISRO platform:** We utilize the Bhuvan geoportal, developed by the Indian Space Research Organisation (ISRO), as a reliable source for water resource information specific to India. This platform provides valuable, officially curated data on water body extent.

    - **Data Type: Monthly water spread area measurements:** This dataset offers aggregated information on the total surface area of water bodies, reported on a monthly basis. This provides a temporal overview of water availability.

    - **Variables:**

        - **Month-Year:** This variable indicates the specific month and year for which the water spread area is recorded, providing the temporal context for the data.

        - **Current Water Spread Area (sq km):** This represents the measured surface area of water bodies in square kilometers for the given month and year. This is the primary metric for tracking changes in water extent.

        - **Maximum Water Spread Area (sq km):** This indicates the maximum recorded water spread area for the same month across the available historical data on the Bhuvan platform. This provides a baseline or potential capacity for water bodies during that time of year, allowing us to assess current conditions relative to the past.

- **Time Frame: Multi-year dataset (long-term trend analysis):** The availability of a multi-year dataset from Bhuvan is crucial for conducting meaningful trend analysis. By examining the water spread area measurements over an extended period, we can identify long-term patterns, seasonal variations, and potential impacts of climate change or human activities on water resources. This coarser but authoritative temporal data from Bhuvan complements the potentially more granular but also potentially more noisy water body extents derived from the Sentinel-2 imagery through our deep learning model, offering a valuable point of reference for validation and broader trend understanding.

3.4 Model Specification

The core of our water body detection system lies in a carefully chosen deep learning model based on the U-Net architecture. This architecture has demonstrated remarkable success in various semantic segmentation tasks, particularly in the medical imaging domain, and its effectiveness extends well to the analysis of satellite imagery for identifying specific land cover types like water bodies.

Model Architecture: U-Net - A Powerful Segmentation Network

- **Input:** Our model is designed to accept multispectral satellite images as input. Specifically, it takes the Red, Green, Blue, and Near-Infrared (NIR) bands from the Sentinel-2 imagery. The input images are typically resized to a consistent dimension (e.g., 256x256 pixels) to ensure uniformity for the model. These four channels provide the necessary spectral information for the model to learn the distinct characteristics of water.

- **Backbone: U-Net with CNN-based encoder-decoder structure:** The U-Net architecture is characterized by its distinctive U-shape. It consists of two main paths: an encoder (contracting path) and a decoder (expansive path), connected by skip connections.

- **Encoder:** The encoder is a traditional convolutional neural network (CNN) that progressively downsamples the input image while increasing the number of feature channels. This path learns a hierarchical representation of the input data, with deeper layers capturing more complex and abstract features. It typically involves repeated blocks of convolutional layers, followed by an activation function (ReLU) and a pooling operation (e.g., max pooling) for downsampling.
- **Decoder:** The decoder path aims to generate the segmentation mask by progressively upsampling the feature maps from the encoder while decreasing the number of feature channels. It uses transposed convolutional layers (also known as deconvolutional layers) for upsampling, followed by concatenation with corresponding feature maps from the encoder (via skip connections). These skip connections are a crucial element of the U-Net, as they allow the decoder to access fine-grained spatial information from the earlier layers of the encoder, helping to produce more precise and detailed segmentation boundaries for the water bodies. The decoder also includes convolutional layers and activation functions after each upsampling step to refine the segmentation mask.

- **Activation Functions:**
  - **ReLU for hidden layers:** We employ the Rectified Linear Unit (ReLU) activation function in the hidden layers of both the encoder and the decoder. ReLU introduces non-linearity into the model, enabling it to learn complex relationships between the input pixels and the output mask. Its simplicity and efficiency also help in faster training by mitigating the vanishing gradient problem that can occur with other activation functions in deep networks.
  - **Sigmoid for final output layer (binary mask prediction):** In the final

layer of the U-Net, we use the sigmoid activation function. Since our task is binary classification (predicting whether each pixel represents water or non-water), the sigmoid function is ideal as it outputs a probability value between 0 and 1 for each pixel. A value close to 1 indicates a high probability of the pixel belonging to the 'water' class, while a value close to 0 suggests it belongs to the 'non-water' class.

- **Loss Function: Binary Cross-Entropy (for pixel-wise classification):** To train the model to accurately predict the water body masks, we utilize the Binary Cross-Entropy loss function. This loss function is commonly used for binary classification problems where the goal is to predict the probability of a pixel belonging to a specific class. It measures the dissimilarity between the predicted probability (from the sigmoid output) and the actual ground truth label (0 or 1) for each pixel in the image. The model's parameters are adjusted during training to minimize this loss, effectively learning to distinguish between water and non-water pixels.

- **Optimizer: Adam:** We have chosen the Adam (Adaptive Moment Estimation) optimizer to train our U-Net model. Adam is a popular and efficient optimization algorithm that combines the benefits of both AdaGrad and RMSProp optimizers. It adapts the learning rates for each parameter based on the estimates of the first and second moments of the gradients. This adaptive learning rate often leads to faster convergence and better overall performance compared to traditional optimizers with a fixed learning rate.

- **Output: Binary mask (1 = Water, 0 = Non-Water):** The final output of our trained U-Net model is a binary segmentation mask. This mask has the same spatial dimensions as the input satellite image. Each pixel in the output mask is assigned a value of either 1 or 0. A pixel value of 1 indicates that the model has predicted that the corresponding pixel in the input image represents water, while a pixel value of 0 indicates that it represents non-water. This binary mask provides a clear and direct representation of the detected water bodies in the satellite imagery, which can then be used for further analysis and visualization

3.5 Training Specification

The process of training our U-Net model is a critical stage that determines its ability to accurately identify water bodies in satellite imagery. We employ a structured approach, incorporating several key steps to ensure optimal model performance, generalization, and robustness.

1. **Preprocessing: Preparing the Data for Optimal Learning**

Before feeding the satellite images into the U-Net model, a series of preprocessing steps are applied to standardize the input and improve training efficiency and stability.

- **Image Resizing:** To ensure consistent input dimensions for the model, all satellite images in our dataset are resized to a fixed dimension (e.g., 256x256 pixels). This uniformity is crucial for the convolutional layers in the U-Net to process the images effectively, regardless of their original sizes.

- **Normalization:** The pixel values of the satellite images, which typically range from 0 to a higher value depending on the bit depth, are scaled to a range between 0 and 1. This normalization step helps in enhancing training stability by preventing large input values from dominating the learning process and ensures that the gradients during backpropagation remain within a manageable range.

- **Mask Preparation:** The corresponding ground truth water body masks are processed to be binary, with distinct values representing water (e.g., 1 or white) and non-water regions (e.g., 0 or black). This clear distinction in the masks provides the model with unambiguous targets for learning during the supervised training process.

2. **Data Augmentation: Enhancing Generalization and Robustness**

To prevent the model from overfitting to the specific characteristics of the training data and to improve its ability to generalize to unseen satellite images captured under different conditions, we apply various data augmentation techniques.

- **Random Flipping:** Images and their corresponding masks are randomly flipped both horizontally and vertically. This introduces spatial variations in the training data, making the model less sensitive to the orientation of water bodies in the images.

- **Rotation:** The images are rotated by small random angles (e.g., within a range of -15 to +15 degrees). This helps the model learn to recognize water bodies even if they are oriented differently in the satellite images due to variations in sensor view angles or geographical layout.

- **Contrast Adjustment:** The contrast of the images is randomly adjusted to simulate varying lighting conditions that might be present in different satellite scenes. This makes the model more robust to changes in illumination.

- **Zooming & Cropping:** Random zoom and crop operations are applied to the images. Zooming introduces scale variations, while cropping focuses on different parts of the image, forcing the model to learn features at different scales and contexts.

3. **Training Process: Supervised Learning for Water Body Detection**

The U-Net model is trained using a supervised learning approach. During training, the preprocessed and augmented satellite images are fed as input to the model, and the model makes predictions about the location of water bodies. These predictions are then compared to the corresponding ground truth masks, and the model's parameters are adjusted to minimize the difference between the predictions and the ground truth.

- **Loss Function: A combination of Binary Cross-Entropy and Dice Loss:** We utilize a combination of Binary Cross-Entropy and Dice Loss to optimize the model's performance. Binary Cross-Entropy focuses on pixel-wise accuracy, penalizing the model for each incorrectly classified pixel. Dice Loss, on the other hand, directly measures the overlap between the predicted segmentation and the ground truth, making it particularly effective for imbalanced datasets where the number of water pixels might be significantly smaller or larger than non-water pixels. Combining these two loss functions helps the model achieve both accurate pixel-level classification and good overall segmentation of water bodies.

- **Optimizer: Adam optimizer:** The Adam optimizer is selected for its efficiency and adaptive learning rate mechanism. Its ability to adjust the learning rate for each parameter during training often leads to faster convergence and better final model performance compared to optimizers with fixed learning rates.

- **Metrics Used: Intersection over Union (IoU) and Accuracy:** We monitor two primary metrics to evaluate the model's performance during training and validation. Intersection over Union (IoU), also known as the Jaccard Index, measures the overlap between the predicted water body regions and the ground truth water body regions, providing a strong indicator of the segmentation quality. Accuracy, on the other hand, measures the percentage of correctly classified pixels (both water and non-water). Monitoring both metrics gives us a comprehensive understanding of the model's ability to accurately detect water bodies.

- **Checkpointing: The best-performing model is saved based on validation loss:** To prevent overfitting, where the model learns the training data too well and performs poorly on unseen data, we employ checkpointing. During training, the model's performance on a separate validation dataset is periodically evaluated. The model weights that achieve the lowest validation loss (or highest validation IoU) are saved as the "best" model. This ensures that we ultimately use the model that generalizes best to new, unseen satellite imagery.

4. **Hardware Setup: Leveraging GPU Acceleration**

The training of deep learning models, especially on large image datasets like ours, requires significant computational resources. To expedite the training process, we perform training on systems equipped with GPU (Graphics Processing Unit) acceleration. GPUs are highly parallel processors that are well-suited for the matrix operations involved in training neural networks, leading to a substantial reduction in training time compared to using only CPUs (Central Processing Units).

- **TensorFlow and Keras with CUDA:** Our model is implemented using TensorFlow and Keras, which are popular deep learning frameworks. We leverage the CUDA (Compute Unified Device Architecture) capabilities of NVIDIA GPUs to further accelerate the training process, allowing for efficient processing of large batches of satellite image data.

5. **Hyperparameters: Fine-Tuning the Learning Process**

Hyperparameters are parameters of the learning algorithm itself that are set before the training process begins. Carefully tuning these hyperparameters is crucial for achieving optimal model performance. The key hyperparameters we fine-tuned during training include:

- **Batch Size:** This parameter determines the number of training examples that are processed together in one forward and backward pass. We experimented with batch sizes of 8 and 16, choosing the largest size that fits within the available GPU memory. The batch size can affect the training dynamics and the model's generalization ability.

- **Learning Rate:** The learning rate controls the step size at each iteration while updating the model's weights. We used an initial learning rate of 0.0001 and employed a learning rate scheduler to reduce the learning rate when the validation loss plateaued. This helps the model to converge to a better solution by taking larger steps initially and then smaller, more refined steps as training progresses.

- **Number of Epochs:** An epoch represents one complete pass through the entire training dataset. We trained the model for a range of 50 to 100 epochs, monitoring the training and validation performance to determine the optimal number of epochs to prevent overfitting. Training is typically stopped when the validation performance starts to degrade or no longer improves significantly.

- **Validation Split:** We reserved 20% of our dataset as a validation set. This separate set of data is not used during the training process itself but is used to evaluate the model's performance on unseen data during training. This helps in assessing how well the model is generalizing and in tuning hyperparameters.

6. **Model Validation & Performance Monitoring: Ensuring Real-World Applicability**

Throughout the training process, we diligently monitored both the training and validation losses over the epochs. A significant divergence between the training and validation loss can indicate overfitting.

- **Best Model Selection:** The best-performing model is selected based on achieving the highest validation IoU and the lowest validation loss. This ensures that we choose the model that demonstrates the strongest ability to generalize to new, unseen data.

- **Generalization Evaluation:** After training, the selected best model is further tested on a completely separate set of unseen satellite images that were not part of the training or validation sets. This final evaluation provides a more realistic assessment of the model's real-world generalization ability and its effectiveness in detecting water bodies in Vellore.

CHAPTER 4

DESIGN APPROACH AND DETAILS

4.1 Design Approach

This project employs a structured, two-phase design approach to achieve its goal of monitoring water bodies in the Vellore region using advanced remote sensing and deep learning techniques, complemented by official water resource data. The two major phases are intricately linked, with the outputs of the first phase potentially informing and validating the findings of the second.

Phase 1: Water Body Detection – Automated Mapping via Deep Learning

This phase focuses on leveraging the power of deep learning to automatically and accurately identify and map water bodies from satellite imagery. The U-Net model, renowned for its effectiveness in semantic segmentation tasks, forms the core of this phase.

- **Acquiring Sentinel-2 Satellite Images and Ground Truth Water Masks:** The initial step involves acquiring a substantial collection of Sentinel-2 satellite images for the Vellore region from the readily accessible Google Earth Engine platform. We specifically utilize the Red, Green, Blue, and Near-Infrared (NIR) spectral bands, which provide crucial information for distinguishing water from other land cover types. Alongside these images, we utilize a corresponding dataset of ground truth water masks. These masks, generated using a specific threshold on the Normalized Difference Water Index (NDWI), serve as the target labels for training our deep learning model, indicating the precise locations of water bodies within the satellite imagery.

- **Training a U-Net Model to Classify Water Pixels:** The acquired satellite images and their corresponding water masks are then used to train the U-Net model. This training process involves feeding the multispectral satellite imagery into the model and guiding it to learn the characteristic features of water by comparing its predictions to the ground truth masks. Techniques like data augmentation (including random flipping, rotation, contrast adjustment, and zooming/cropping) are applied to enhance the model's generalization ability and prevent overfitting. The model's learning is driven by a carefully chosen loss function, a combination of Binary Cross-Entropy and Dice Loss, and optimized using the Adam optimizer.

- **Using the Trained Model to Generate Water Body Maps:** Once the U-Net model is trained to a satisfactory level of accuracy, it can be deployed to process new, unseen Sentinel-2 satellite images of the Vellore region. For each input image, the trained model will generate a binary output mask. In this mask, pixels classified as '1' (or white) represent the model's prediction of water bodies, while pixels classified as '0' (or black) represent non-water areas. These output masks effectively constitute automatically generated water body maps with a spatial resolution of 10 meters per pixel.

Phase 2: Water Body Trend Analysis – Examining Historical Data for Patterns

The second phase of the project shifts focus to analyzing historical water spread data to identify temporal patterns and long-term trends in Vellore's water resources. This analysis utilizes the officially recorded data from the Bhuvan platform developed by ISRO.

- **Performing Trend Analysis on Historical Data from Bhuvan:** We acquire a multi-year dataset of monthly water spread area measurements from the Bhuvan platform. This dataset includes key variables such as the 'Month-Year', 'Current Water Spread Area (sq km)', and 'Maximum Water Spread Area (sq km)' for water bodies in the Vellore region. Statistical techniques, such as time series decomposition, moving averages, and potentially more advanced methods, will be employed to analyze this historical data and identify significant seasonal variations, year-on-year changes, and overall long-term trends in water body extent.

- **Generating Reports and Visualizations to Understand Seasonal and Long-Term Water Body Fluctuations:** The insights gained from the trend analysis will be synthesized into comprehensive reports. These reports will detail the observed patterns and trends, providing valuable information about the dynamics of water availability in the Vellore region. To further enhance understanding and communication, the findings will be visualized using various graphical representations, such as line plots showing water spread area over time, bar charts comparing annual averages, and potentially visualizations highlighting seasonal patterns.

Integrated Methodology for Effective Water Resource Monitoring

This two-phased methodology provides a powerful and comprehensive approach to water resource monitoring in the Vellore region. The deep learning-based water body detection offers a high-resolution, spatially explicit understanding of water body distribution at specific points in time. Complementarily, the trend analysis of historical data from Bhuvan provides a broader temporal context, revealing long-term patterns and fluctuations in water availability. Integrating the findings from both phases, for instance by comparing the trends observed in the Bhuvan data with the analysis of water body maps generated from satellite imagery over time, can lead to a more holistic and nuanced understanding of water resource dynamics. This integrated approach enables efficient water body monitoring and change detection, providing valuable information for environmental studies, water resource management, and informed policymaking in the Vellore region.

4.2 System Design

The system is architected in a layered manner to efficiently handle the processing of large-scale satellite imagery and structured water spread data, ensuring a smooth flow from data acquisition to insightful reports and visualizations. The primary layers of the system are as follows:

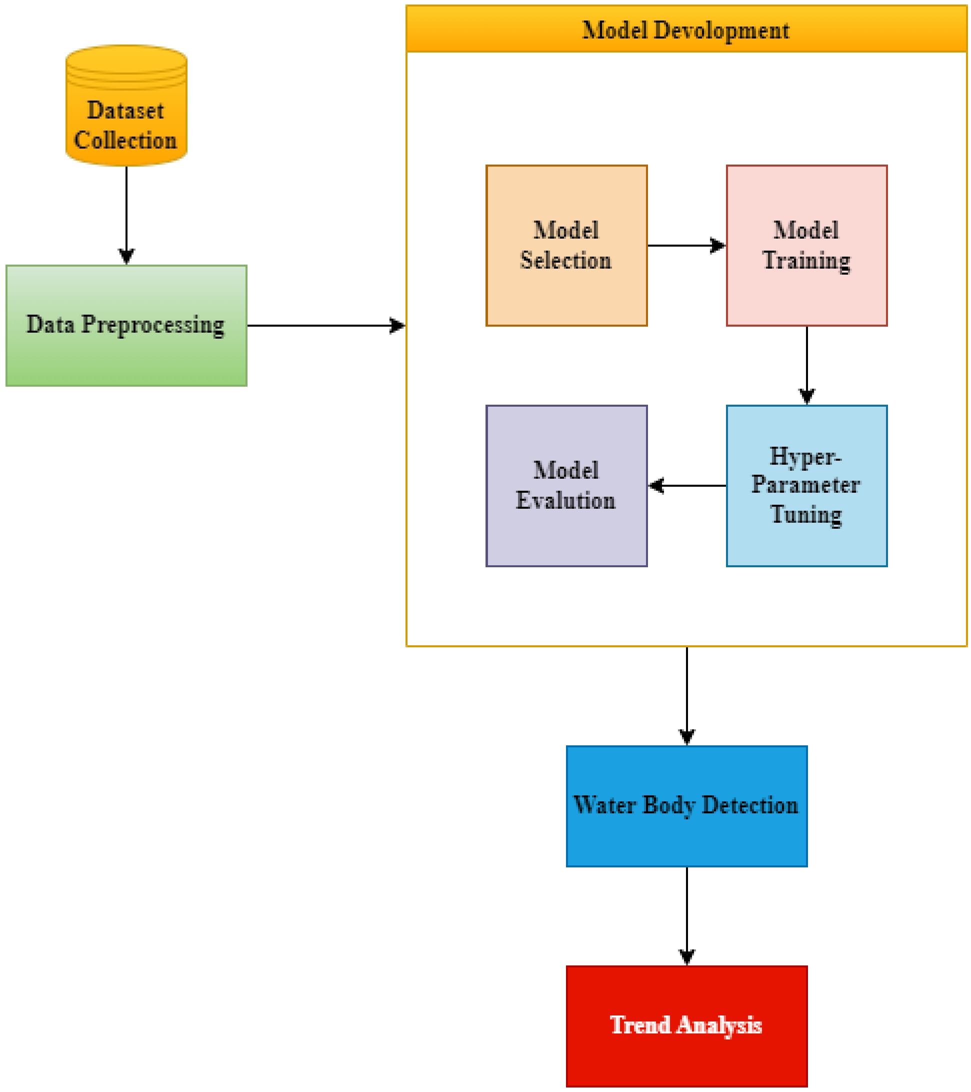


Fig. 4.1 System Architecture Design

1. **Data Processing Layer: Preparing Data for Analysis**

This foundational layer focuses on the crucial step of preparing the raw data from different sources into formats suitable for model training and trend analysis.

- **Satellite Image Preprocessing:** Sentinel-2 satellite images, obtained from Google Earth Engine, undergo essential preprocessing steps. This includes resizing all images to a fixed dimension (e.g., 256x256 pixels) to ensure uniformity for the deep learning model. Pixel values are then normalized to a range between 0 and 1 to enhance the stability and performance of the training process.

- **Water Body Mask Creation:** For the supervised training of the U-Net model, corresponding ground truth water body masks are generated. This involves calculating the Normalized Difference Water Index (NDWI) on the satellite images and applying a specific threshold to identify water bodies. The resulting masks are then binarized, with one pixel value representing water and another representing non-water areas.

- **ISRO Water Spread Data Formatting:** The monthly water spread area data provided by Bhuvan (ISRO) is structured for efficient trend analysis. This involves parsing the data, potentially using libraries like Pandas to create DataFrames. The 'Month-Year' information is properly formatted to allow for time-based analysis, and the 'Current Water Spread Area' and 'Maximum Water Spread Area' are separated into distinct variables for individual and comparative analysis.

2. **Model Training & Deployment: Deep Learning for Automated Detection**

This layer encompasses the development and application of the deep learning model for water body detection.

- **U-Net Model Training:** The U-Net model, with its CNN-based encoder-decoder structure, is trained using the preprocessed Sentinel-2 images and their corresponding water body masks. Data augmentation techniques, such as random flipping, rotation, contrast adjustment, and zooming/cropping, are applied during training to improve the model's generalization and robustness. The model is trained using a combination of Binary Cross-Entropy and Dice Loss, optimized with the Adam optimizer, and monitored using metrics like Intersection over Union (IoU) and Accuracy.

- **Trained Model Application:** Once the U-Net model achieves satisfactory performance on a validation dataset, it is deployed to automatically detect water bodies in new, unseen Sentinel-2 satellite images of the Vellore region. The trained model takes a preprocessed satellite image as input and outputs a binary mask, where pixels are classified as either water or non-water.

- **Results Validation:** The performance of the trained model is rigorously validated using a separate set of ground truth water masks that were not used during training. Metrics like IoU and accuracy are calculated to quantitatively assess the model's ability to accurately detect water bodies.

3. **Trend Analysis Module: Uncovering Temporal Patterns**

This module focuses on analyzing the historical water spread data from Bhuvan to identify temporal patterns and fluctuations in water body extent.

- **Seasonal and Yearly Variation Analysis:** The formatted monthly water spread data is analyzed to identify both seasonal variations (e.g., changes in water body area throughout the year) and yearly trends (e.g., long-term increases or decreases in water availability).

- **Statistical Metrics and Graphs:** Statistical metrics, such as mean water spread area, standard deviation, and potentially trend lines derived through regression analysis, are calculated to quantify the water body fluctuations. Libraries like SciPy are utilized for these statistical computations. The results are then visualized using graphs and charts generated with libraries like Matplotlib and Seaborn to clearly illustrate the trends and variations over time.

4. **Visualization & Reporting: Communicating the Findings**

The final layer focuses on effectively communicating the project's findings to stakeholders through visualizations and comprehensive reports.

- **GIS-Based Data Visualization:** GIS software like QGIS is employed to visualize the detected water bodies spatially. The binary water body masks generated by the U-Net model can be overlaid on the original satellite imagery or other relevant geographical layers, creating insightful maps that show the distribution and extent of water bodies in the Vellore region.

- **Matplotlib for Plotting:** Matplotlib is extensively used to generate various types of plots and charts that visualize the results of the trend analysis. This includes time series plots showing the change in water spread area over months and years, bar charts comparing water availability across different years or seasons, and potentially other visualizations that highlight specific aspects of the water body fluctuations.

- **Report Generation:** Comprehensive reports are generated to document the entire project, including the methodology, dataset specifications, model details, training process, results of the water body detection and trend analysis, and the generated visualizations. These reports provide a detailed understanding of the water resource dynamics in the Vellore region and can serve as valuable input for environmental studies and policymaking.

4.3 Model Architecture Framework

The U-Net model serves as the fundamental framework for our water body detection system, utilizing Sentinel-2 satellite imagery. This architecture is intentionally crafted to extract both granular spatial details and comprehensive contextual understanding, enabling remarkably accurate segmentation of water bodies within the observed area. Our execution adheres to a completely convolutional strategy, a defining characteristic of U-Net, and crucially integrates skip connections to guarantee precise outlining of water body perimeters.

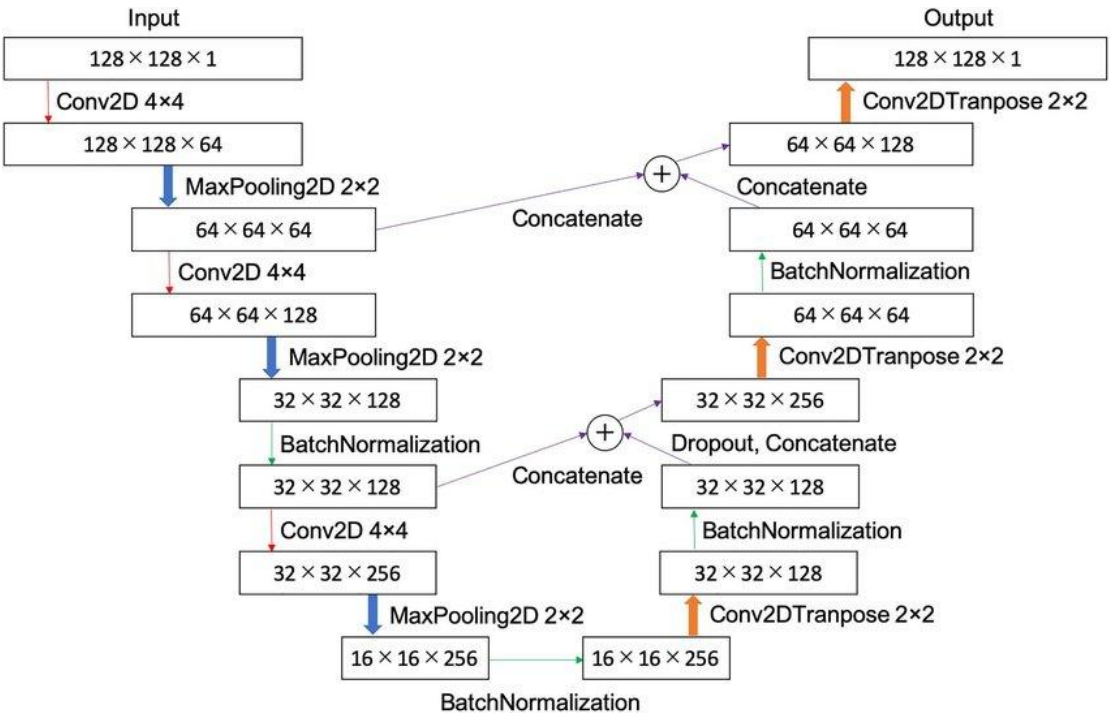


Fig. 4.2 U NET Architecture

1. **Input Handling: Preparing Multi-Spectral Data for the Network**

The model is structured to ingest multi-spectral satellite imagery, specifically capitalizing on the rich spectral data accessible through Sentinel-2's various bands.

- **Spectral Bands:** We employ a selection of crucial bands, including Red, Green, Blue (for visual interpretation), Near-Infrared (NIR) (essential for emphasizing water due to its significant absorption), and Shortwave Infrared (SWIR) (which can further assist in distinguishing water, particularly in situations involving shadow or turbidity).

- **Standardization and Resizing:** Before being processed by the model, the initial input images undergo vital preparation steps. Pixel values are standardized to a uniform range (typically 0 to 1) to ensure consistent and effective training. Furthermore, all input images are adjusted to a fixed size (e.g., 256x256 pixels) to maintain uniformity across the batch and the network's structure.

- **Ground Truth Mask Generation:** The corresponding reference water masks, which function as the learning target for the model, are produced by applying a carefully chosen threshold to the Normalized Difference Water Index (NDWI) computed from the satellite imagery. This automated method provides a consistent and scalable way to create labels.

2. **Encoder (Feature Extraction Pathway): Learning Multi-Level Representations**

The encoder pathway functions as the feature extraction component of the U-Net. Through a sequence of downsampling operations, it progressively learns a multi-level representation of the input image, capturing increasingly abstract and complex features relevant to identifying water bodies.

- **Convolutional Layers:** The encoder begins with several sequential 3x3 convolutional layers. These layers apply learned filters to the input image, extracting basic spatial characteristics such as edges, corners, and textures. Utilizing multiple stacked convolutional layers enables the network to learn more intricate patterns.

- **Batch Normalization:** Following each convolutional layer, we implement batch normalization. This technique aids in stabilizing the training process by minimizing internal covariate shift, permitting the use of higher learning rates and accelerating the rate of convergence. It also acts as a form of regularization, potentially enhancing the model's ability to generalize.

- **ReLU Activation:** After batch normalization, the Rectified Linear Unit (ReLU) activation function is applied. ReLU introduces necessary non-linearity into the network, allowing it to learn complex, non-linear relationships between the input pixels and the presence of water.

- **Max-Pooling Layers:** Downsampling is achieved through the utilization of max-pooling layers with a 2x2 kernel and a stride of 2. These layers reduce the spatial dimensions of the feature maps, making the network more resilient to minor shifts and distortions in the input. Importantly, downsampling also expands the receptive field of the subsequent convolutional layers, enabling them to consider a broader context within the input image.

- **Feature Depth Expansion:** As the signal progresses deeper into the encoder, the quantity of feature channels is systematically increased after each downsampling stage. This expansion of feature depth allows the network to learn and represent increasingly sophisticated and semantic features that are indicative of water bodies.

3. **Bottleneck (Deep Feature Learning): Connecting Encoder and Decoder**

The bottleneck layer is situated at the deepest point of the U-Net, serving as the crucial link between the feature extraction performed by the encoder and the upsampling and reconstruction carried out by the decoder.

- **Convolutional Layers with Expanded Receptive Field:** The bottleneck typically includes several convolutional layers, often with a broader receptive field compared to the initial encoder layers. This enables the model to learn global patterns and contextual information about water bodies across the entire spatial extent of the image.

- **Dropout Layers:** To further mitigate overfitting, particularly at this deepest level where the model has acquired the most abstract features, dropout layers are frequently incorporated within the bottleneck. Dropout randomly sets a proportion of the neuron activations to zero during training, compelling the network to learn redundant representations and improving its ability to generalize to new data.

4. **Decoder (Upsampling Pathway): Reconstructing High-Resolution Segmentations**

The decoder pathway is responsible for reconstructing a high-resolution segmentation mask from the low-resolution, high-level features extracted by the encoder. It accomplishes this through a series of upsampling operations and the integration of precise spatial details from the encoder via skip connections.

- **Transposed Convolution (Upconvolution):** The decoder employs transposed convolutional layers (also known as deconvolutional layers) to upsample the feature maps, effectively increasing their spatial resolution back towards the original input size. This upsampling is vital for generating a segmentation mask with the same dimensions as the input image.

- **Skip Connections:** A defining feature of the U-Net architecture is the use of skip connections. These direct connections transfer feature maps from corresponding layers in the encoder to the decoder. Prior to concatenation, the feature maps from the encoder are typically cropped to match the size of the upsampled feature maps from the decoder. These skip connections are crucial for retaining fine-grained spatial details that were potentially lost during the downsampling process in the encoder, leading to more accurate and precise boundary delineation of water bodies in the final segmentation mask.

- **ReLU Activation:** Similar to the encoder, ReLU activation is applied after the transposed convolutions and the subsequent convolutional layers in the decoder to ensure continued non-linearity in the feature learning process during upsampling and refinement.

5. **Output Layer & Activation: Generating the Binary Water Body Mask**

The final stage of the model involves mapping the high-level features learned by the decoder to a pixel-wise binary classification, indicating the presence or absence of water.

- **1x1 Convolutional Layer:** A 1x1 convolutional layer is applied to the output of the final decoder block. This layer acts as a classifier, mapping the learned features at each spatial location to the desired number of output classes (in our case, two: water and non-water).

- **Sigmoid Activation Function:** The output of the 1x1 convolutional layer is passed through a sigmoid activation function. As this is a binary segmentation problem, the sigmoid function outputs a probability value between 0 and 1 for each pixel, representing the likelihood of that pixel belonging to the 'water' class (1) or the 'non-water' class (0).

6. **Loss Function: Guiding Model Learning for Precise Segmentation**

The model's learning process is guided by a carefully chosen loss function that quantifies the difference between the model's predictions and the ground truth water masks.

- **Binary Cross-Entropy (BCE) Loss:** BCE loss measures the pixel-wise classification error, calculating the average of the negative log-likelihood of the predicted probability for each pixel given its true label. This loss function encourages the model to correctly classify each individual pixel as either water or non-water.

- **Dice Loss:** Dice Loss is a region-based loss function that directly quantifies the overlap between the predicted water body regions and the ground truth water body regions. It is particularly effective in addressing class imbalance, which can occur in satellite imagery where the area covered by water might be significantly smaller or larger than other land cover types. Maximizing the Dice coefficient (which is minimized by the Dice Loss) corresponds to maximizing the overlap between the prediction and the ground truth.

- **Combination:** We frequently utilize a combination of BCE Loss and Dice Loss during training. This strategy leverages the strengths of both loss functions, encouraging both accurate pixel-level classification and high overall overlap between the predicted and actual water bodies.

7. **Optimizer & Learning Strategy: Efficient Model Training**

The optimization algorithm and the learning strategy play a crucial role in how effectively and efficiently the model learns from the training data.

- **Adam Optimizer:** We utilize the Adam optimizer due to its adaptive learning rate capabilities and its efficiency in converging to satisfactory solutions. Adam adjusts the learning rate for each parameter based on estimates of the first and second moments of the gradients, making it less sensitive to the choice of initial learning rate and often leading to faster training.

- **Learning Rate Scheduler:** To further enhance training stability and the final performance of the model, we employ a learning rate scheduler. This scheduler dynamically adjusts the learning rate during training, often starting with a higher learning rate and gradually reducing it as the training progresses or when the validation performance plateaus. This allows for faster initial learning and finer adjustments later in the training process.

- **Early Stopping:** To prevent overfitting and conserve computational resources, we implement early stopping. During training, we monitor the model's performance on a separate validation dataset. If the validation loss does not improve (or even begins to increase) over a predefined number of epochs, the training process is automatically terminated. This ensures that we select the model that performs best on unseen data.

8. **Model Performance Enhancements: Improving Generalization**

To ensure that our model performs effectively not only on the training data but also on new, unseen satellite images, we incorporate several performance enhancement techniques.

- **Data Augmentation:** As detailed in the training specification, various data augmentation techniques are applied to the training images to introduce artificial variability. This includes random rotations, horizontal and vertical flips, adjustments to brightness and contrast, and zooming/cropping. By training on a more diverse set of data, the model becomes more robust and less prone to overfitting to specific characteristics of the training set.

- **Regularization Techniques:** We utilize regularization techniques such as dropout (as previously mentioned in the bottleneck description) and batch normalization throughout the network. These techniques help to prevent the model from becoming excessively specialized to the training data and enhance its capacity to generalize to new, unseen data.

9. **Output & Post-Processing: Refining the Segmentation Mask**

The final output of the trained U-Net model is a binary segmentation map where each pixel is categorized as either water (indicated by the value 1) or non-water (indicated by the value 0). Depending on the specific needs of the subsequent analysis, some post-processing steps might be applied to refine this initial segmentation map. For example, morphological operations (such as erosion and dilation) could be used to eliminate small, isolated pixels or fill in minor gaps within the predicted water body regions, resulting in a cleaner and more coherent final output.

4.4 Codes and Standards

This project follows standard practices to ensure the accuracy and reliability of its results.

- **Remote Sensing Standards:** Satellite imagery processing adheres to established guidelines from NASA and ESA for accurate data handling and spatial accuracy.
- **Deep Learning Best Practices:** The U-Net model is developed and trained using TensorFlow/Keras, following recognized methodologies for effective deep learning.
- **GIS Data Standards:** Analysis of water body data from ISRO's Bhuvan platform follows their established data processing protocols.
- **Reproducibility & Documentation:** All code and analysis are documented to ensure transparency and facilitate future work.

These standards help ensure the scientific validity and real-world usability of the project's findings.

4.5 Constraints, Alternatives, and Tradeoffs

The development of this water body detection and trend analysis project involved careful consideration of various constraints, exploration of alternative approaches, and a thorough evaluation of the inherent tradeoffs to arrive at the most effective and practical solution.

Constraints: Factors Limiting the Design and Implementation

- **Data Availability: The Challenge of Cloud Cover in Satellite Imagery:** A significant constraint in working with optical satellite imagery like Sentinel-2 is the presence of cloud cover. Clouds can obscure the Earth's surface, directly impacting our ability to detect water bodies in the affected areas. This necessitates careful selection of imagery with minimal cloud cover or the implementation of techniques to mitigate the effects of clouds, such as temporal compositing (using multiple images over time) or exploring alternative data sources like radar imagery in future iterations.

- **Computational Power: The Requirement for High-Performance Computing Resources:** Training deep learning models, especially complex architectures like U-Net on large datasets, demands substantial computational resources, particularly in terms of GPU processing power. The lengthy training times and the memory requirements of these models necessitate access to high-performance computing infrastructure, which can be a limiting factor in terms of cost and accessibility. Our reliance on Google Colab's GPU resources addresses this constraint to a certain extent but introduces dependencies on an external platform.

- **Labeling Accuracy: The Importance of Precise Ground Truth for Model Training:** The performance of a supervised deep learning model is heavily dependent on the quality and accuracy of the ground truth labels used for training. In our case, the water body masks generated using NDWI-based thresholding need to be precise. Errors or inaccuracies in these masks can lead to the model learning incorrect features, ultimately affecting the accuracy of the water body detection. Ensuring the accuracy of these labels, possibly through manual review and refinement in critical areas, is an ongoing challenge.

- **Temporal Resolution: Limitations of Monthly Data for Short-Term Fluctuation Analysis:** The monthly water spread area data obtained from ISRO's Bhuvan platform, while valuable for long-term trend analysis, may not capture short-term or sudden fluctuations in water body extent. Events like flash floods or rapid changes in water levels within a month might not be fully reflected in this aggregated monthly data. This limitation in temporal resolution needs to be acknowledged when interpreting the trend analysis results.

Alternatives Considered: Exploring Different Methodologies

During the design phase, we considered several alternative approaches before settling on the current methodology:

- **Alternative Segmentation Models:** While U-Net was chosen for its proven effectiveness in semantic segmentation tasks, we also considered other popular architectures such as DeepLabV3 and Fully Convolutional Networks (FCNs). DeepLabV3 incorporates atrous convolutions to capture multi-scale contextual information, which could be beneficial. FCNs were among the early successful architectures for semantic segmentation. However, U-Net's skip connections, which are particularly effective for precise boundary delineation—crucial for water body mapping—made it the preferred choice for this project.

- **Manual Water Body Mapping:** An alternative to automated deep learning-based detection would be manual digitizing of water bodies from satellite imagery by human experts. While this method can potentially yield highly accurate results for individual images, it is extremely labor-intensive, time-consuming, and subjective, especially when dealing with a large volume of imagery required for comprehensive monitoring and trend analysis over time. The goal of this project is to develop an efficient and scalable automated solution.

- **Using Landsat Data:** Landsat imagery is another widely used source for

Earth observation with a long historical archive. We considered using Landsat data; however, Sentinel-2 offers a higher spatial resolution (10 meters compared to Landsat's 30 meters for many bands), which allows for the detection of smaller water bodies and more precise boundary mapping. While Landsat's longer history is advantageous for very long-term studies, Sentinel-2's higher resolution was prioritized for the level of detail required in this project.

Tradeoffs: Balancing Competing Objectives

The design of this project involved careful balancing of several inherent tradeoffs:

- **High Accuracy vs. Computational Cost:** The U-Net model is known for its ability to achieve high segmentation accuracy, which is crucial for reliable water body detection. However, this high precision comes at the cost of significant computational resources required for training the deep and complex network. We addressed this by leveraging the GPU resources provided by Google Colab, acknowledging this as a necessary computational expense to achieve the desired level of accuracy.

- **Resolution vs. Temporal Frequency:** Higher resolution satellite imagery, such as that provided by Sentinel-2, allows for better detection of smaller water bodies and more precise boundary mapping. However, the trade-off is often a lower temporal frequency of available cloud-free imagery compared to lower-resolution satellites that might image the same area more frequently. We prioritized the higher resolution of Sentinel-2 to obtain detailed water body maps, understanding that we might need to work with images acquired at slightly less frequent intervals or implement cloud mitigation strategies.

- **Model Complexity vs. Interpretability:** More complex deep learning models can often achieve higher performance on tasks like image segmentation. However, they can also be more difficult to interpret and understand, potentially acting as "black boxes." While the U-Net architecture is relatively well-established and its components are understandable, more recent and extremely deep architectures might offer marginal gains in performance but at the cost of reduced interpretability, which can be important for understanding the factors influencing water body changes. We opted for the U-Net, striking a balance between high performance and a reasonably interpretable architecture.

By carefully considering these constraints, evaluating alternative approaches, and consciously balancing the inherent tradeoffs, the chosen design and methodology aim to ensure an accurate, efficient, and scalable solution for water body detection and trend analysis in the Vellore region.

CHAPTER 5

SCHEDULE, TASKS, AND MILESTONES

The project was completed within a **4-month duration**, following a structured timeline to systematically develop the **deep learning-based water body detection model** and conduct **trend analysis of water bodies in Vellore**. The execution was divided into key **tasks and milestones**, ensuring timely completion of each phase.

5.1 Project Schedule & Tasks

Table 5.1 Schedule of The Project

Phase	**Task**	**Duration**	**Completion Status**
Phase 1 (Month 1 - Week 1 & 2)	Data Collection	2 weeks	Sentinel-2 satellite images and Bhuvan ISRO water body data acquired
Phase 2 (Month 1 - Week 3 & 4)	Data Preprocessing	2 weeks	Preprocessed images, generated NDWI-based masks
Phase 3	Model Development	4 weeks	U-Net model trained on

(Month 2 - Week 1,2,3,4)	(U-Net for Water Body Detection)		labeled dataset
Phase 4 (Month 3 - Week 1 & 2)	Model Evaluation & Optimization	2 weeks	Model fine-tuned, achieved acceptable accuracy
Phase 5 (Month 3 - Week 3 & 4)	Water Body Detection & Segmentation on Vellore Data	2 weeks	U-Net model applied to predict water masks
Phase 6 (Month 4 - Week 1)	Trend Analysis (Statistical Analysis of Water Body Data)	1 weeks	Seasonal, yearly, and overall water body trends analyzed
Phase 7 (Month 4 - Week 2)	Result Interpretation & Visualization	1 weeks	Visualizations and GIS-based mapping completed
Phase 8 (Month 4 - Week 3 & 4)	Final Report & Presentation	2 weeks	Documentation and final presentation prepared

5.2 Milestones & Deliverables

Table 5.2 Milestones

Milestone	**Deliverables**	**Completion Date**
M1: Data Collection Completed	Acquired Sentinel-2 satellite images & Bhuvan ISRO data	Month 1, Week 2
M2: Data Preprocessing Completed	Normalized images, generated NDWI-based masks	Month 1, Week 4
M3: U-Net Model Trained	Model trained for water body segmentation	Month 2, Week 4
M4: Model Evaluation Completed	IoU and accuracy metrics optimized	Month 3, Week 2
M5: Water Body Detection Applied to Vellore	Predicted water masks for Vellore region	Month 3, Week 4

M6: Trend Analysis Completed	Seasonal & long-term water body trends identified	Month 4, Week 1
M7: Final Report & Presentation Ready	Documentation & project presentation finalized	Month 4, Week 4

The project successfully **met all objectives within the planned 4-month timeline**, ensuring a **robust deep learning-based water body detection system** and **detailed trend analysis for Vellore water bodies**.

CHAPTER 6

DISSERTATION DEMONSTRATION

This chapter presents a demonstration of the project through sample codes and screenshots showcasing key aspects of water body detection using deep learning and trend analysis of Vellore water bodies.

6.1 Sample Codes

Water Body Detection

```
from google.colab import drive
drive.mount('/content/drive/')
Drive already mounted at /content/drive/; to attempt to forcibly
remount, call drive.mount("/content/drive/", force_remount=True).
import os
import numpy as np
from tqdm import tqdm
import cv2 as cv
from PIL import Image
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
import tensorflow as tf
```

```
from tensorflow.keras.models import Model
from tensorflow.keras.callbacks import Callback
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.utils import plot_model
from tensorflow.keras.layers import Input, Conv2D, Conv2DTranspose,
MaxPooling2D, concatenate, Dropout
image_path =
'/content/drive/MyDrive/23MCA0304_Dissertation_II/Dataset/Water
Bodies Dataset/Images/'
mask_path =
'/content/drive/MyDrive/23MCA0304_Dissertation_II/Dataset/Water
Bodies Dataset/Masks/'
SIZE = 128
# lists of images and masks names
image_names = sorted(next(os.walk(image_path))[-1])[:]
mask_names = sorted(next(os.walk(mask_path))[-1])[:]

images = np.zeros(shape=(len(image_names),SIZE, SIZE, 3))
masks = np.zeros(shape=(len(image_names),SIZE, SIZE, 1))

for id in tqdm(range(len(image_names)), desc="Images"):
  path = image_path + image_names[id]
  img = np.asarray(Image.open(path)).astype('float')/255.
  img = cv.resize(img, (SIZE,SIZE), cv.INTER_AREA)
  images[id] = img

for id in tqdm(range(len(mask_names)), desc="Mask"):
  path = mask_path + mask_names[id]
  mask = np.asarray(Image.open(path)).astype('float')/255.
  mask = cv.resize(mask, (SIZE,SIZE), cv.INTER_AREA)
  masks[id] = mask[:,:,:1]
Images: 100%|[redacted]| 2841/2841 [01:10<00:00, 40.20it/s]
Mask: 100%|[redacted]| 2841/2841 [00:35<00:00, 80.97it/s]
# Plot some images and its masks
plt.figure(figsize=(10,15))
for i in range(1,11):
  plt.subplot(5,2,i)

  if i%2!=0:
    id = np.random.randint(len(images))
    plt.imshow(images[id], cmap=None)
    plt.title('Original Image')
  elif i%2==0:
    plt.imshow(masks[id].reshape(128,128), cmap='gray')
    plt.title('Mask')

plt.tight_layout()
plt.show()
# Train test split
images_train, images_test, mask_train, mask_test =
train_test_split(images, masks, test_size=0.25)
# Define U-net architecture
def unet_model(input_layer, start_neurons):
```

```
    # Contraction path
    conv1 = Conv2D(start_neurons, kernel_size=(3, 3),
activation="relu", padding="same")(input_layer)
    conv1 = Conv2D(start_neurons, kernel_size=(3, 3),
activation="relu", padding="same")(conv1)
    pool1 = MaxPooling2D((2, 2))(conv1)
    pool1 = Dropout(0.25)(pool1)

    conv2 = Conv2D(start_neurons*2, kernel_size=(3, 3),
activation="relu", padding="same")(pool1)
    conv2 = Conv2D(start_neurons*2, kernel_size=(3, 3),
activation="relu", padding="same")(conv2)
    pool2 = MaxPooling2D((2, 2))(conv2)
    pool2 = Dropout(0.5)(pool2)

    conv3 = Conv2D(start_neurons*4, kernel_size=(3, 3),
activation="relu", padding="same")(pool2)
    conv3 = Conv2D(start_neurons*4, kernel_size=(3, 3),
activation="relu", padding="same")(conv3)
    pool3 = MaxPooling2D((2, 2))(conv3)
    pool3 = Dropout(0.5)(pool3)

    conv4 = Conv2D(start_neurons*8, kernel_size=(3, 3),
activation="relu", padding="same")(pool3)
    conv4 = Conv2D(start_neurons*8, kernel_size=(3, 3),
activation="relu", padding="same")(conv4)
    pool4 = MaxPooling2D((2, 2))(conv4)
    pool4 = Dropout(0.5)(pool4)

    # Middle
    convm = Conv2D(start_neurons*16, kernel_size=(3, 3),
activation="relu", padding="same")(pool4)
    convm = Conv2D(start_neurons*16, kernel_size=(3, 3),
activation="relu", padding="same")(convm)

    # Expansive path
    deconv4 = Conv2DTranspose(start_neurons*8, kernel_size=(3, 3),
strides=(2, 2), padding="same")(convm)
    uconv4 = concatenate([deconv4, conv4])
    uconv4 = Dropout(0.5)(uconv4)
    uconv4 = Conv2D(start_neurons*8, kernel_size=(3, 3),
activation="relu", padding="same")(uconv4)
    uconv4 = Conv2D(start_neurons*8, kernel_size=(3, 3),
activation="relu", padding="same")(uconv4)

    deconv3 = Conv2DTranspose(start_neurons*4, kernel_size=(3, 3),
strides=(2, 2), padding="same")(uconv4)
    uconv3 = concatenate([deconv3, conv3])
    uconv3 = Dropout(0.5)(uconv3)
    uconv3 = Conv2D(start_neurons*4, kernel_size=(3, 3),
activation="relu", padding="same")(uconv3)
    uconv3 = Conv2D(start_neurons*4, kernel_size=(3, 3),
activation="relu", padding="same")(uconv3)
```

```
    deconv2 = Conv2DTranspose(start_neurons*2, kernel_size=(3, 3), 
strides=(2, 2), padding="same")(uconv3)
    uconv2 = concatenate([deconv2, conv2])
    uconv2 = Dropout(0.5)(uconv2)
    uconv2 = Conv2D(start_neurons*2, kernel_size=(3, 3), 
activation="relu", padding="same")(uconv2)
    uconv2 = Conv2D(start_neurons*2, kernel_size=(3, 3), 
activation="relu", padding="same")(uconv2)

    deconv1 = Conv2DTranspose(start_neurons*1, kernel_size=(3, 3), 
strides=(2, 2), padding="same")(uconv2)
    uconv1 = concatenate([deconv1, conv1])
    uconv1 = Dropout(0.5)(uconv1)
    uconv1 = Conv2D(start_neurons, kernel_size=(3, 3), 
activation="relu", padding="same")(uconv1)
    uconv1 = Conv2D(start_neurons, kernel_size=(3, 3), 
activation="relu", padding="same")(uconv1)

    # Last conv and output
    output_layer = Conv2D(1, (1,1), padding="same", 
activation="sigmoid")(uconv1)

    return output_layer
# Compile unet model
input_layer = Input((SIZE, SIZE, 3))
output_layer = unet_model(input_layer = input_layer, start_neurons = 
16)

model = Model(input_layer, output_layer)
model.compile(loss="binary_crossentropy", optimizer="adam", 
metrics=["accuracy"])
model.summary()
# Post Process
def mask_threshold(image, threshold=0.25):
  return image>threshold

# Callback to show progress of learning on the images after each 
epoch
class ShowProgress(Callback):
  def __init__(self, save=False):
    self.save = save
  def on_epoch_end(self, epoch, logs=None):
    k = np.random.randint(len(images_train))
    original_image = images_train[k][np.newaxis,...]
    predicted_mask = 
self.model.predict(original_image).reshape(128,128)
    proc_mask02 = mask_threshold(predicted_mask, threshold=0.2)
    proc_mask03 = mask_threshold(predicted_mask, threshold=0.3)
    proc_mask04 = mask_threshold(predicted_mask, threshold=0.4)
    proc_mask05 = mask_threshold(predicted_mask, threshold=0.5)
    mask = mask_train[k].reshape(128,128)
```

```
    plt.figure(figsize=(15,10))

    plt.subplot(1,7,1)
    plt.imshow(original_image[0]);plt.title('Orginal Image')

    plt.subplot(1,7,2)
    plt.imshow(predicted_mask, cmap='gray');plt.title('Predicted
Mask')

    plt.subplot(1,7,3)
    plt.imshow(mask, cmap='gray');plt.title('Orginal Mask')

    plt.subplot(1,7,4)
    plt.imshow(proc_mask02, cmap='gray');plt.title('Processed: 0.2')

    plt.subplot(1,7,5)
    plt.imshow(proc_mask03, cmap='gray');plt.title('Processed: 0.3')

    plt.subplot(1,7,6)
    plt.imshow(proc_mask04, cmap='gray');plt.title('Processed: 0.4')

    plt.subplot(1,7,6)
    plt.imshow(proc_mask05, cmap='gray');plt.title('Processed: 0.5')

    plt.tight_layout()
    plt.show()
# Training
epochs = 100
batch_size = 32

history = model.fit(images_train, mask_train,
                    validation_data=[images_test, mask_test],
                    epochs=epochs,
                    callbacks=[ShowProgress()],
                    batch_size=batch_size)
Epoch 1/100
1/1 ━━━━━━━━━━━━━━━━━━━━2s 2s/step
...
 67/67 ━━━━━━━━━━━━━━━━━━━━6s 95ms/step - accuracy: 0.7566 - loss:
0.1875 - val_accuracy: 0.7410 - val_loss: 0.2522
plt.figure(figsize=(10,5))

# Plot loss for each epoch
plt.subplot(1,2,1)
plt.plot(history.history['loss'], label="Train loss")
plt.plot(history.history['val_loss'], label="Test loss")
plt.title('Binary crossentropy loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['TRAIN', 'TEST'], loc='upper left')

# Plot accuracy for each epoch
plt.subplot(1,2,2)
```

```
plt.plot(history.history['accuracy'], label="Train accuracy")
plt.plot(history.history['val_accuracy'], label="Test accuracy")
plt.title('Accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['TRAIN', 'TEST'], loc='upper left')

plt.tight_layout()
plt.show()
def calculate_iou(y_true, y_pred):
    """Calculates the Intersection over Union (IoU) score."""
    intersection = np.logical_and(y_true > 0, y_pred > 0)
    union = np.logical_or(y_true > 0, y_pred > 0)
    if np.sum(union) == 0:
        return 1.0 if np.sum(intersection) == 0 else 0.0
    iou = np.sum(intersection) / np.sum(union)
    return iou
def calculate_average_iou(mask_test, predictions, threshold=0.5):

    iou_scores = []
    for i in range(len(mask_test)):
        mask = mask_test[i].reshape(128, 128)
        predicted_mask = predictions[i].reshape(128, 128)
        threshold_mask = mask_threshold(predicted_mask,
threshold=threshold)
        iou = calculate_iou(mask, threshold_mask)
        iou_scores.append(iou)
    return np.mean(iou_scores)
# Make predictions
predictions = model.predict(images_test)
23/23 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━2s 55ms/step

  average_iou = calculate_average_iou(mask_test, predictions,
threshold=0.1)
  print(f"\nAverage IoU score: {average_iou:.4f}")
def plot_results(threshold=0.5):
    k = np.random.randint(len(images_test))
    original_image = images_test[k]
    mask = mask_test[k].reshape(128, 128)
    mask = mask_threshold(mask, threshold=threshold)
    predicted_mask = predictions[k].reshape(128, 128)
    threshold_mask = mask_threshold(predicted_mask,
threshold=threshold)

    plt.figure(figsize=(15, 5))

    plt.subplot(1, 4, 1)
    plt.imshow(original_image);
    plt.title('Original Image')
    plt.axis('off')

    plt.subplot(1, 4, 2)
    plt.imshow(mask, cmap='gray');
```

```
    plt.title('Original Mask')
    plt.axis('off')

    plt.subplot(1, 4, 3)
    plt.imshow(predicted_mask, cmap='gray');
    plt.title('Predicted Mask')
    plt.axis('off')

    plt.subplot(1, 4, 4)
    plt.imshow(threshold_mask, cmap='gray');
    plt.title(f'Predicted Mask with cutoff={threshold}')
    plt.axis('off')

    plt.tight_layout()
    plt.show()

# Plot results on test data
for i in range(10):
  plot_results(threshold=0.4)

.save('/content/drive/MyDrive/23MCA0304_Dissertation_II/Model/water_
body_detection_model_1.keras')
from tensorflow import keras
model =
keras.models.load_model('/content/drive/MyDrive/23MCA0304_Dissertati
on_II/Model/water_body_detection_model_1.keras')
```

Water Body Trend Analysis In Vellore

```
# Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import zscore
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.arima.model import ARIMA

file_path =
"/content/drive/MyDrive/23MCA0304_Dissertation_II/Dataset/Vellore_Wa
ter_Spread_Data.csv"
df = pd.read_csv(file_path)

# Convert 'Month-Year' column to datetime format
df['Month-Year'] = pd.to_datetime(df['Month-Year'], format='%b-%y')
print(df.head())

  Month-Year  Current Water Spread Area  Maximum Water Spread Area
0 2012-02-01                      13.11                     183.09
1 2012-03-01                      23.76                     231.28
2 2012-04-01                      15.35                     260.57
3 2012-05-01                       1.61                     215.95
4 2012-09-01                       5.53                     232.51
```

```
plt.figure(figsize=(12, 5))
plt.plot(df['Month-Year'], df['Current Water Spread Area'],
label="Current Water Spread", color='b')
plt.plot(df['Month-Year'], df['Maximum Water Spread Area'],
label="Max Water Spread", color='r', linestyle="dashed")
plt.xlabel("Year")
plt.ylabel("Water Spread Area")
plt.title("Water Spread Trend Over Time")
plt.legend()
plt.grid()
plt.show()

# Seasonal decomposition
decompose_result = seasonal_decompose(df.set_index('Month-
Year')['Current Water Spread Area'], model='additive', period=12)
decompose_result.plot()
plt.show()


# Set 'Month-Year' as index
df.set_index('Month-Year', inplace=True)

# Perform seasonal decomposition
decompose_result = seasonal_decompose(df['Current Water Spread
Area'], model='additive', period=12)

# Extract seasonal component
seasonal_component = decompose_result.seasonal

# Compute average seasonal effect per month
df['Month'] = df.index.month
avg_seasonality = df.groupby('Month')['Current Water Spread
Area'].mean()

# Plot the average seasonal component
plt.figure(figsize=(10, 5))
avg_seasonality.plot(kind='bar', color='royalblue', alpha=0.7)
plt.xlabel('Month')
plt.ylabel('Average Water Spread Area')
plt.title('Average Monthly Water Spread Change Area Over the Years')
plt.grid(axis='y', linestyle='--', alpha=0.6)
plt.show()



# Yearly Percentage Change (YoY) - Only for year-end values

yearly_data = df.groupby(df.index.year)['Current Water Spread
Area'].last().reset_index() # Get year-end values
yearly_data['Previous Water Spread'] = yearly_data['Current Water
Spread Area'].shift(1)  # Add previous year's data
yearly_data['YoY Change (%)'] = yearly_data['Current Water Spread
Area'].pct_change() * 100
```

```python
yearly_data= yearly_data.rename(columns={'Current Water Spread Area': 'Current Spread Area'})
yearly_change = yearly_data[['Month-Year', 'Current Spread Area', 'Previous Water Spread', 'YoY Change (%)']]

# Display Results
print("\n--- Yearly Percentage Change ---\n", yearly_change[1:])

# Plot bar graph of Current Spread Area
plt.figure(figsize=(12, 5))
plt.bar(yearly_change['Month-Year'], yearly_change['Current Spread Area'], color='skyblue')
plt.xlabel("Year")
plt.ylabel("Current Water Spread Area")
plt.title("Current Water Spread Area Over the Years")
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.xticks(rotation=45, ha='right')  # Rotate x-axis labels for better readability
plt.tight_layout()  # Adjust layout to prevent labels from overlapping
plt.show()

# Prepare data for grouped bar graph
years = yearly_change['Month-Year'].values
current_spread = yearly_change['Current Spread Area'].values
previous_spread = yearly_change['Previous Water Spread'].values

# Set width of bars
bar_width = 0.35

# Set position of bars on X axis
x_pos = np.arange(len(years))

# Create grouped bar graph
fig, ax = plt.subplots(figsize=(12, 6))
rects1 = ax.bar(x_pos - bar_width / 2, current_spread, bar_width, label='Current Spread Area', color='skyblue')
rects2 = ax.bar(x_pos + bar_width / 2, previous_spread, bar_width, label='Previous Spread Area', color='lightcoral')

# Add labels, title, and legend
ax.set_xlabel("Year")
ax.set_ylabel("Water Spread Area")
ax.set_title("Comparison of Current and Previous Water Spread Area")
ax.set_xticks(x_pos)
ax.set_xticklabels(years)
ax.legend()

# Display the plot
plt.show()
```

```
# Extract season from month
df['Season'] = df['Month-Year'].dt.month.map({12: 'Winter', 1:
'Winter', 2: 'Winter',
                                               3: 'Spring', 4:
'Spring', 5: 'Spring',
                                               6: 'Summer', 7:
'Summer', 8: 'Summer',
                                               9: 'Monsoon', 10:
'Monsoon', 11: 'Monsoon'})

# Calculate Average Water Spread per Season
seasonal_avg = df.groupby('Season')['Current Water Spread
Area'].mean()

# Calculate Percentage Contribution of Each Season
seasonal_percent = (seasonal_avg / seasonal_avg.sum()) * 100
print(seasonal_percent)
```

```
Season
Monsoon     20.146911
Spring      24.575368
Summer      16.295417
Winter      38.982305
Name: Current Water Spread Area, dtype: float64
```

```
# ========== STEP 4: CORRELATION ANALYSIS ==========
correlation = df[['Current Water Spread Area', 'Maximum Water Spread
Area']].corr()
print("\n--- Correlation Matrix ---\n", correlation)

sns.heatmap(correlation, annot=True, cmap='coolwarm')
plt.title("Correlation Heatmap")
plt.show()
```

```
--- Correlation Matrix ---
                             Current Water Spread Area  \
Current Water Spread Area                     1.000000
Maximum Water Spread Area                     0.415892

                            Maximum Water Spread Area
Current Water Spread Area                    0.415892
Maximum Water Spread Area                    1.000000
```

```
# Compute overall percentage change
initial_value = df['Current Water Spread Area'].iloc[0]
final_value = df['Current Water Spread Area'].iloc[-1]
```

```
overall_change = ((final_value - initial_value) / initial_value) *
100
print(f"Overall Water Spread Change: {overall_change:.2f}%")
```

```
Overall Water Spread Change: 669.34%
```

6.2 Sample Screen Shots

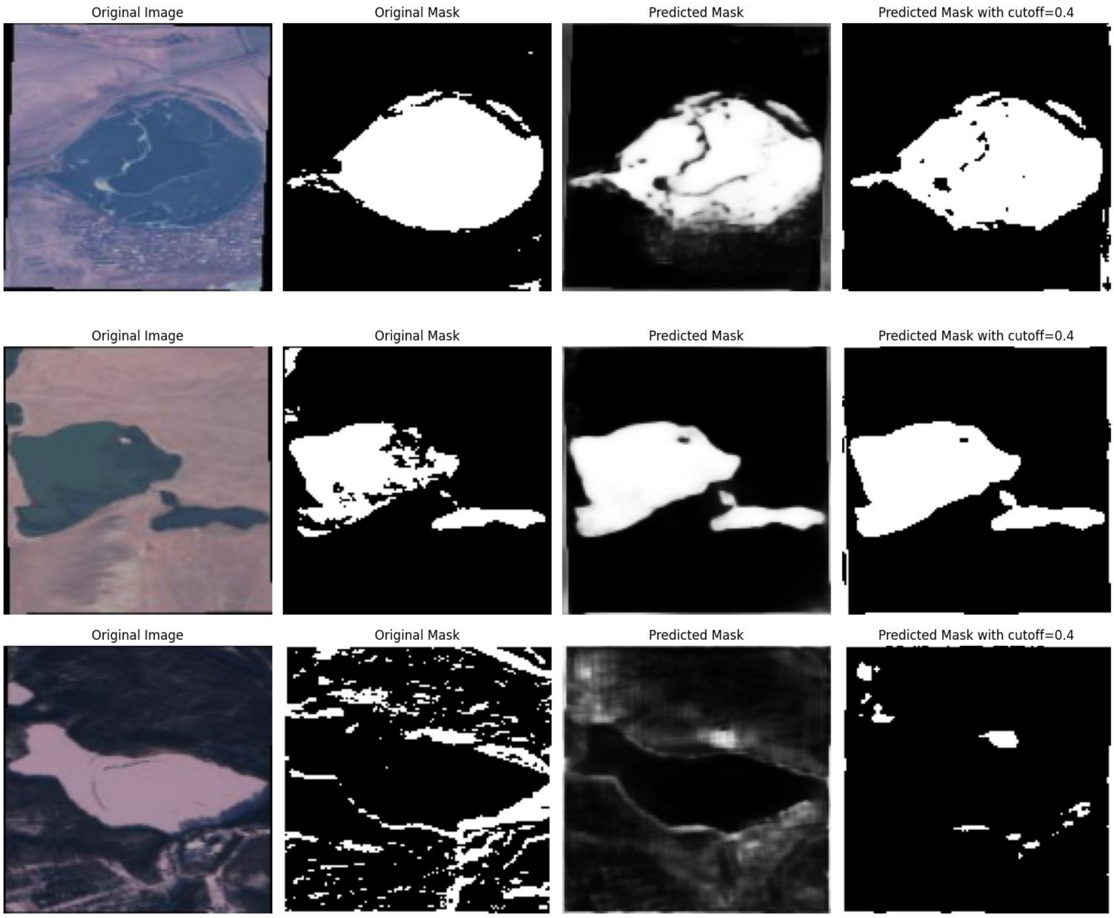
This section presents **key outputs and visualizations** generated during the project.

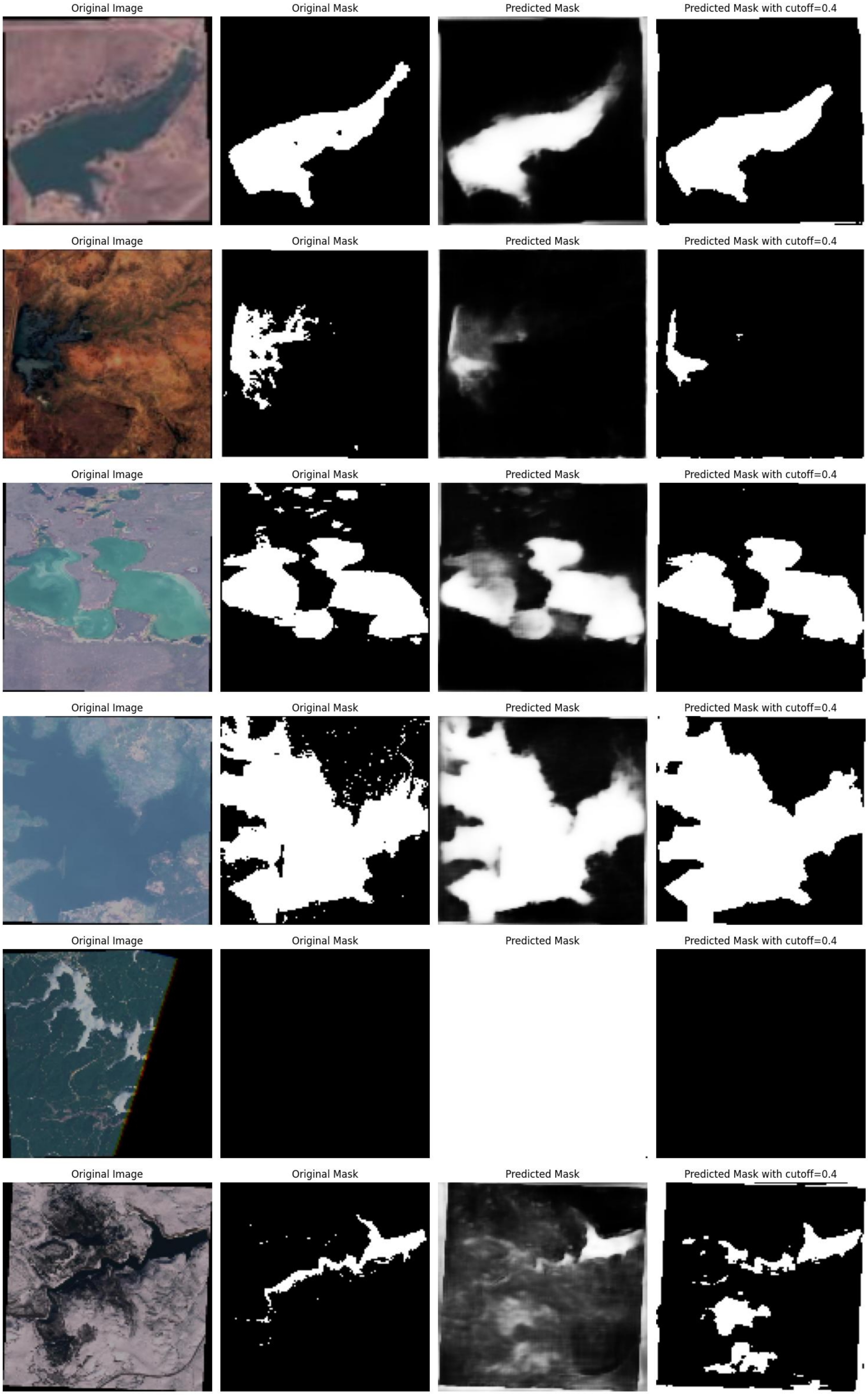
1. Sample Water Body Detection Output (U-Net Predictions)

Left: Input Sentinel-2 satellite image

Middle: Ground truth (NDWI-based mask)

Right: Predicted water mask from U-Net




Original Image
Original Mask
Predicted Mask
Predicted Mask with cutoff=0.4
Original Image
Original Mask
Predicted Mask
Predicted Mask with cutoff=0.4
Original Image
Original Mask
Predicted Mask
Predicted Mask with cutoff=0.4
Original Image
Original Mask
Predicted Mask
Predicted Mask with cutoff=0.4
Original Image
Original Mask
Predicted Mask
Predicted Mask with cutoff=0.4
Original Image
Original Mask
Predicted Mask
Predicted Mask with cutoff=0.4

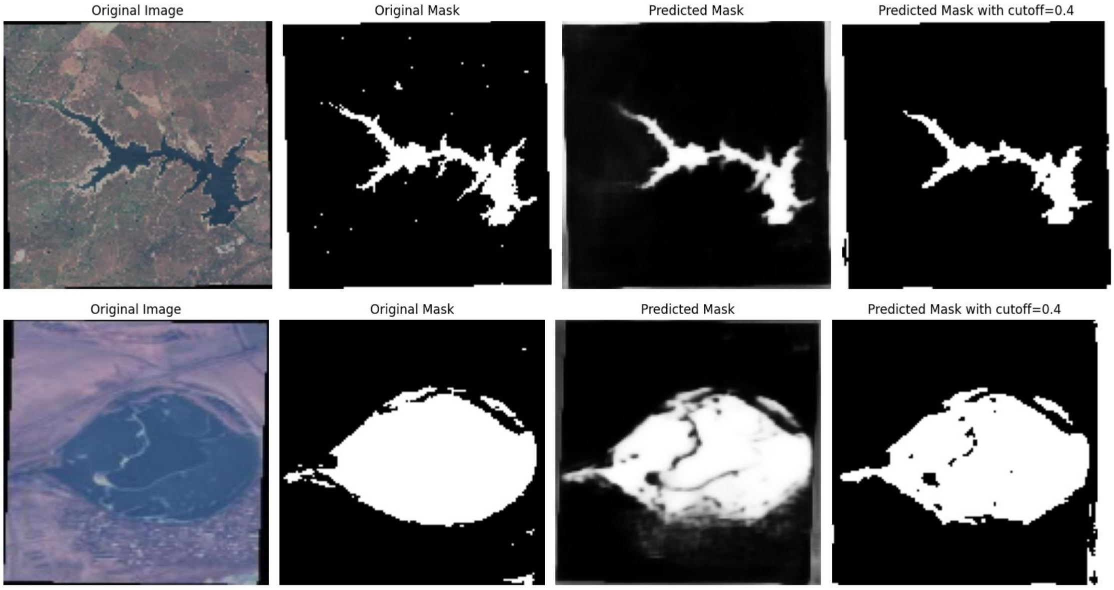


Fig. 6.1 U NET MODEL OUTPUTS

Vellore Satellite Image:

Fig 6.2 Vellore Satellite Image

Vellore NDWI Image:

Fig 6.3 Vellore Satellite NDWI Image

Vellore Water Mask:

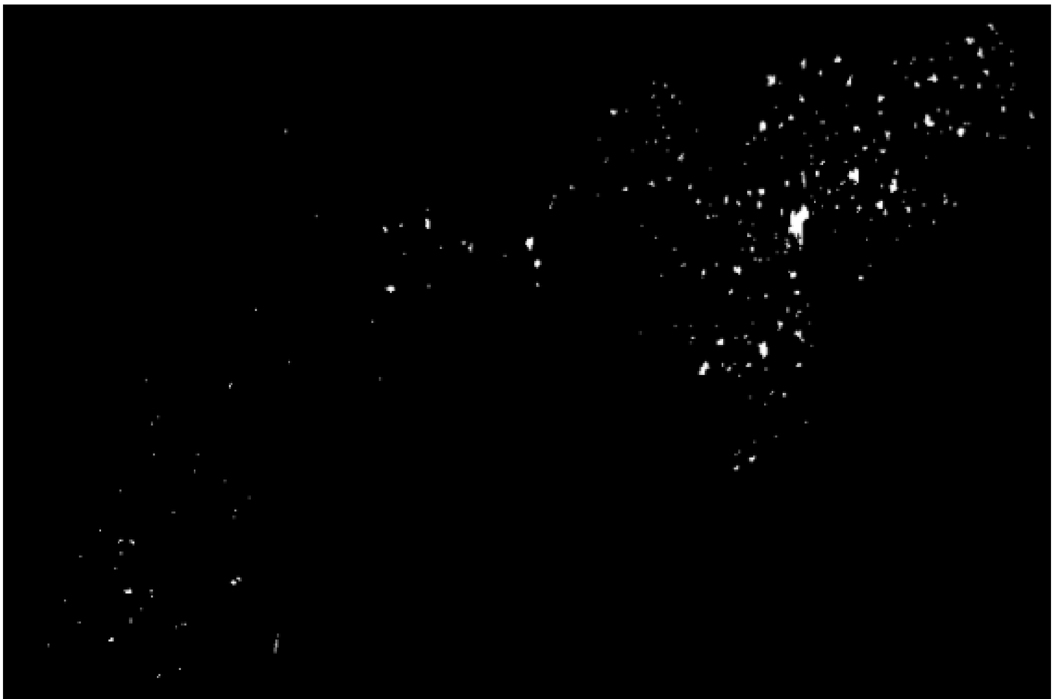


Fig 6.4 Vellore Satellite Water Mask Image

2. Water Body Trend Analysis Graphs

The trend analysis provides insights into seasonal, annual, and long-term fluctuations in water body spread across Vellore. The following visualizations depict key observations:

(a) Yearly Trend of Water Spread Area

Displays changes in current and maximum water spread areas across multiple years.

Helps identify long-term increase or decrease in water bodies.

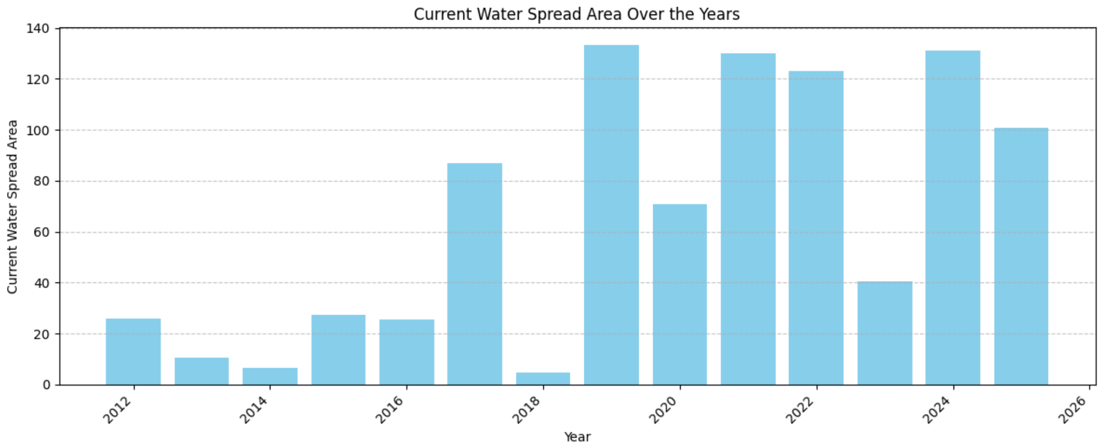


Fig. 6.5 Yearly Trends of Water Spread Area

(b) Seasonal Trend Analysis

Highlights fluctuations based on monsoon, summer, and winter seasons.

Useful in understanding how water bodies expand or shrink in different climatic conditions.

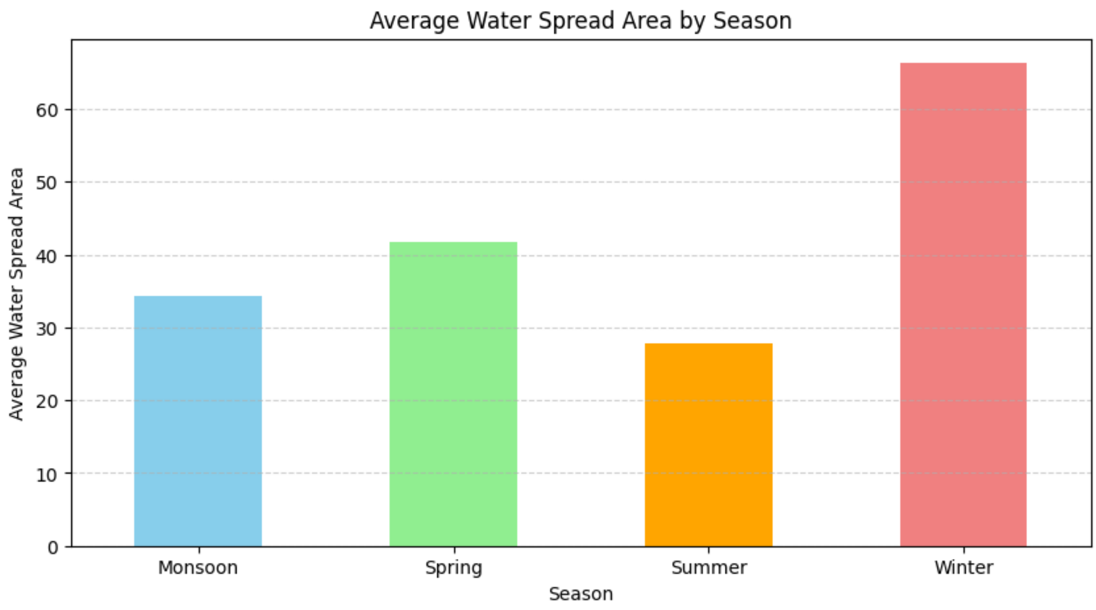


Fig. 6.6 Average Seasonal Trends of Water Spread Area

(c) Monthly Variation of Water Bodies

Captures the month-to-month variations in the water spread area.

Helps analyze short-term trends and extreme fluctuations.

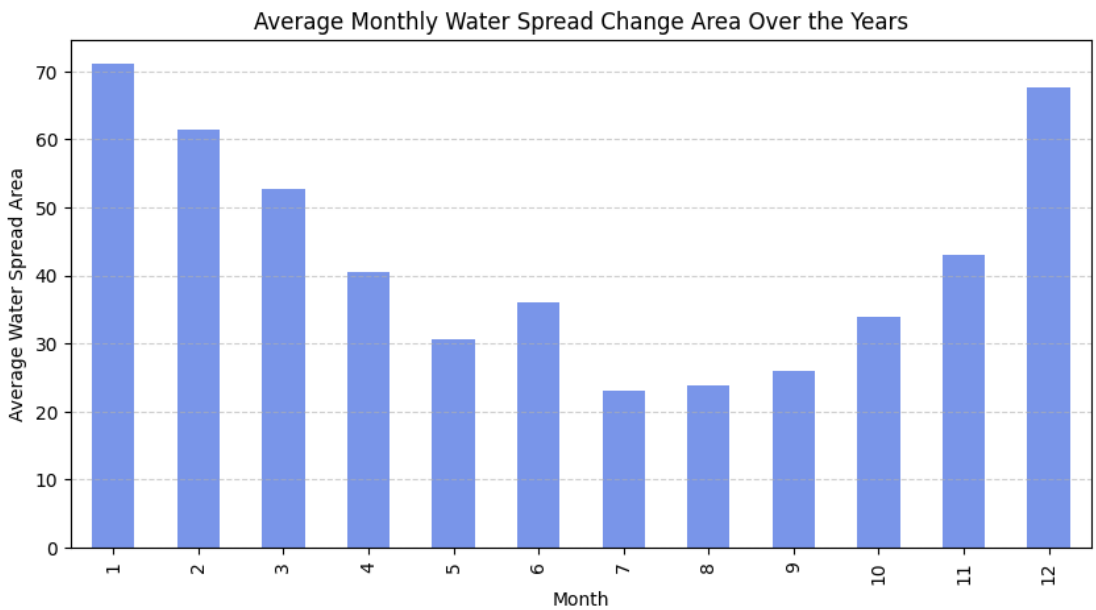


Fig. 6.7 Average Monthly Trends of Water Spread Area

(d) Maximum vs. Current Water Spread Area

Compares the actual water spread area against the potential maximum water spread.

Identifies underutilization or water loss trends.

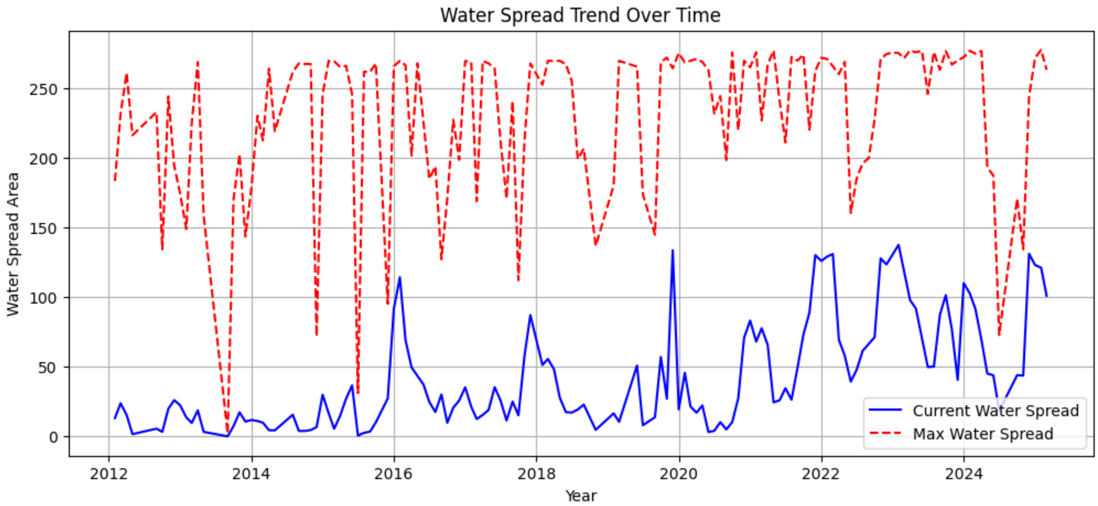


Fig. 6.8 Maximum vs. Current Water Spread Area

(e) Correlation Matrix for Water Body Trends

Shows the correlation between different factors such as current water spread area, maximum water spread area, and time.

Helps understand the relationship between various parameters affecting water bodies.

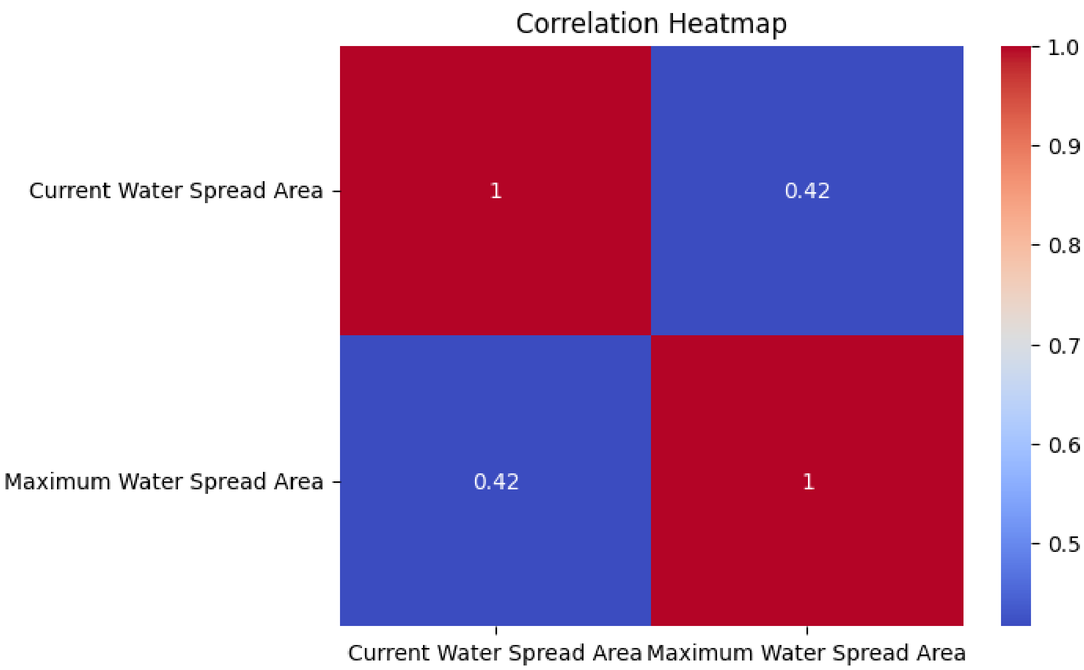


Fig. 6.9 Correlation Matrix of Current and Maximum Water Spread Area

3. Model Training & Performance Metrics

- Training loss and accuracy over epochs

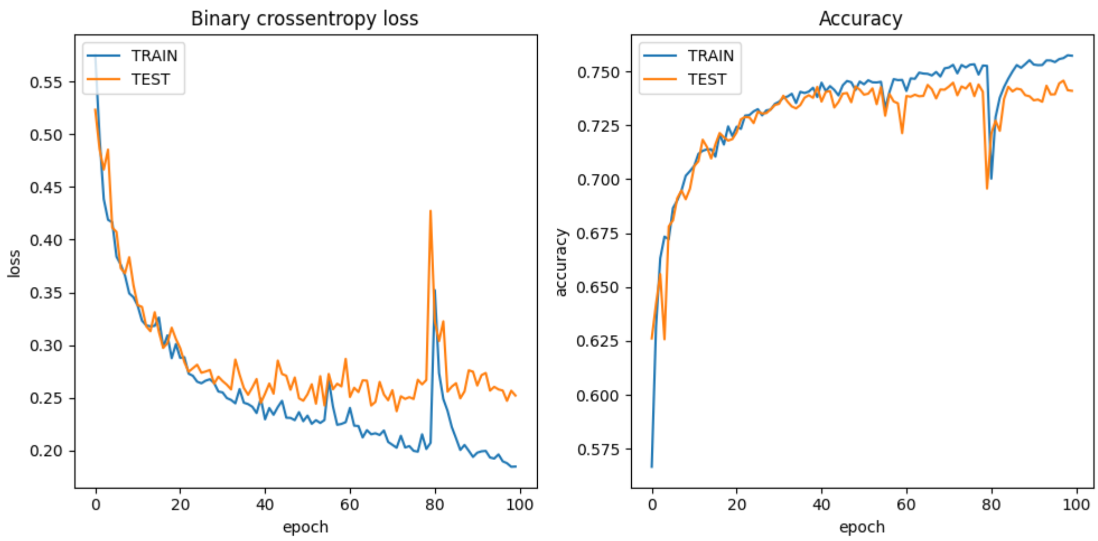


Fig. 6.10 Loss and Accuracy Over Epoch

These visualizations support the findings of water body detection and trend analysis, helping to assess the impact of seasonal variations and long-term water body changes in Vellore.

CHAPTER 7

RESULT & DISCUSSION

7.1 Model Performance Overview

The U-Net model, trained on Sentinel-2 satellite imagery and NDWI-derived water masks, demonstrated effective water body detection with the following key performance metrics on the validation dataset: Intersection over Union (IoU) of 64%, Accuracy of 74.10%, and a Loss of 0.2522.

The IoU score of 64% indicates a good level of overlap between the model's predictions and the actual water bodies, signifying its ability to capture the spatial extent of water accurately. The accuracy of 74.10% reflects the overall correctness of pixel classification as either water or non-water. The loss value of 0.2522 represents the error minimized by the model during training.

While the model effectively identified larger water bodies, it faced challenges in detecting smaller or partially dried-up regions. These instances often presented ambiguous spectral signatures at the 10-meter resolution of Sentinel-2 imagery.

Several strategies were crucial in achieving this level of performance. Data augmentation techniques, including random flipping, rotation, contrast adjustment, and zooming, enhanced the model's generalization capabilities and reduced overfitting to the training data. Furthermore, meticulous hyperparameter tuning, involving adjustments to the learning rate, batch size, and training epochs, optimized the learning process and contributed to the reported metrics.

In conclusion, the U-Net model showcased promising results in the automated detection of water bodies in the Vellore region. The achieved IoU and accuracy demonstrate its potential as a tool for water resource monitoring, while acknowledging areas for further improvement, particularly in handling smaller and transitional water features.

7.2 Trends Observed in Vellore

This section delves into a comprehensive analysis of the water spread patterns in the Vellore region, utilizing the historical data obtained from the Bhuvan (ISRO) platform.

The analysis focuses on identifying key trends and variations in water body extent over the years, considering both seasonal fluctuations and long-term changes.

Key Trend Analysis of Water Bodies in Vellore:

1. **Overall Water Spread Trend: Unveiling Seasonal Rhythms and Long-Term Dynamics** The analysis of the Current Water Spread Area from the Bhuvan dataset reveals significant fluctuations from year to year. These variations underscore the dynamic nature of water resources in the region, influenced by a complex interplay of factors such as rainfall patterns, evaporation rates, and water management practices. The observed steady increase in the Current Water Spread Area from 2016 to 2024 suggests a positive trend in water retention within the Vellore region. This improvement could be attributed to a combination of factors, including potentially more favorable rainfall patterns during these years, successful implementation of water conservation initiatives, or enhanced management of reservoirs and other water storage facilities. The consistently high values of the Maximum Water Spread Area across the analyzed period indicate the inherent capacity of the region's reservoirs and other water bodies to hold substantial volumes of water. However, the fact that the Current Water Spread Area often falls below this maximum suggests that this potential is not always fully realized, highlighting the impact of seasonal variations and other factors that influence water availability.

2. **Dry Season vs. Monsoon Season Analysis: Examining Seasonal Extremes**

    - **Low Water Spread (Summer: March – May):** The data clearly illustrates a drastic decrease in water levels within the Vellore region during the summer months of March to May. This decline is primarily driven by high rates of evaporation due to rising temperatures and a general lack of significant rainfall during this period. The month of May consistently exhibits the lowest water spread, representing the peak of the dry season's impact. Notably, the historical data reveals that certain years (2012, 2013, 2014) experienced particularly severe water scarcity during May, with water spread areas dwindling to near-dry conditions.

This highlights the vulnerability of the region's water resources to prolonged periods of heat and low precipitation. However, more recent data indicates a trend of improved water retention even during the summer months, suggesting a potential shift towards greater resilience compared to the earlier part of the study period.

- **High Water Spread (Monsoon & Post-Monsoon: July – December):** In contrast to the dry summer months, the period from July to December, encompassing the monsoon and post-monsoon seasons, witnesses a significant increase in water spread across the Vellore region. This replenishment is primarily attributed to the heavy rainfall associated with the monsoon. The water spread typically peaks during the later months of this period, around October to December, reflecting the cumulative impact of the monsoon rains. Notably, the data reveals exceptionally high water spread values in December of 2019 and December of 2022. These instances likely correspond to years with particularly strong monsoons, resulting in substantial recharge of the region's water bodies.

3. **Long-Term Changes (2012 - 2024): Identifying Decadal Shifts** An examination of the water spread data over the entire study period from 2012 to 2024 allows for the identification of distinct phases of water availability in the Vellore region:

    - **2012 - 2015:** This period was characterized by extreme water scarcity during certain months, particularly May, June, and September. These low water levels suggest a period of drought or below-average rainfall, placing significant stress on water resources.

    - **2016 - 2019:** Following the initial period of scarcity, the data indicates a gradual improvement in overall water levels across the region. While seasonal dips in water spread during the summer months persisted, the general trend suggests a recovery or improvement in water availability compared to the preceding years.

    - **2020 - 2024:** This most recent phase shows a significant increase in water retention within the Vellore region. The consistently higher water spread areas during this period, across different seasons, could be indicative of more favorable rainfall patterns, the positive impact of water conservation efforts implemented in the region, or effective management of water storage facilities.

4. **Final Inferences: Key Takeaways for Water Resource Management** The analysis of water spread trends in Vellore leads to several important inferences:

    - **Clear Seasonal Pattern:** The water spread area in the region exhibits a distinct seasonal pattern, closely aligned with the monsoon cycle. Water levels typically peak during the monsoon and post-monsoon months due to rainfall and decline significantly during the dry summer months.

    - **Positive Long-Term Trend:** The overall long-term trend in water spread area appears to be positive, particularly in recent years (2016-2024), suggesting an improvement in water retention and availability in Vellore over time.

    - **Summer Water Depletion: A Persistent Concern:** Despite the positive long-term trend, the drastic depletion of water resources during the summer months, especially in May, remains a significant concern that necessitates focused attention for effective water conservation planning and implementation of mitigation strategies.

    - **Year-on-Year Variability: Dependence on External Factors:** The highly variable year-on-year changes in water spread area highlight the region's dependence on factors such as the amount and distribution of rainfall and the effectiveness of reservoir management practices.

This detailed trend analysis of water bodies in Vellore provides valuable insights that can aid in forecasting future water availability, making informed decisions for sustainable water resource management, and proactively identifying drought-prone months to implement timely mitigation strategies.

7.3 Discussion

This project successfully integrated a deep learning water body detection system with long-term water trend analysis in the Vellore region. The U-Net model, using high-resolution Sentinel-2 satellite images, provided effective water body mapping with a 10-meter resolution. Analysis of historical water data from the Bhuvan platform has revealed important trends in water availability over the past decade.

Key Insights:

Effectiveness of the U-Net Model: The U-Net model achieved 64% Intersection over Union (IoU) and 74.10% accuracy, showing its potential for efficient water body detection compared to traditional methods. While challenges remain for smaller areas, the model lays a solid groundwork for water resource assessments.

Seasonal Water Patterns: The analysis highlighted the significant impact of the monsoon on water levels, with increases seen during monsoon months (July to December) and declines in summer (March to May) due to evaporation and lack of rainfall. These patterns are crucial for water resource management.

Long-Term Water Retention Trends: The observed gradual improvement in water retention from 2016 to 2024 is positive, indicating possible enhanced rainfall or improved management practices. However, significant depletion during summer months, especially in May, emphasizes the need for targeted water conservation efforts.

Implications for Water Management Policies: The findings can guide the development of effective water conservation policies and reservoir management. Identifying drought-prone months can help initiate preventative measures like water rationing and enhanced monitoring. Long-term trends also support evaluating current policies and preparing for future water needs in light of climate change and population dynamics. High-resolution mapping can pinpoint areas at risk of water scarcity for targeted conservation initiatives.

7.4 Future Scope

Building upon the achievements of this study, numerous exciting possibilities exist for future research and development that could significantly enhance the capabilities and broaden the scope of automated water resource monitoring:

Enhanced Model Performance: Future research could explore the integration of additional spectral indices beyond NDWI, such as the Normalized Difference Vegetation Index (NDVI) to better differentiate water from vegetation, or indices specifically designed for turbid water detection. Experimenting with different loss functions, such as the Tversky loss or Focal loss, which are known to perform well in segmentation tasks with class imbalance, could also improve the model's ability to accurately identify water bodies, particularly the smaller or partially dried ones.

Temporal Information Integration: To better capture the dynamic nature of water bodies, future models could be designed to incorporate temporal information directly into the learning process. This could involve using recurrent neural network layers (RNNs) within the U-Net architecture or exploring the use of 3D convolutional networks that can process sequences of satellite images over time, allowing the model to learn the temporal dependencies and changes in water body extent.

Cloud Masking and Data Handling: Addressing the challenge of cloud cover is crucial for operational water monitoring. Future work could focus on developing and integrating robust cloud masking techniques to identify and exclude cloud-affected areas from the analysis. Furthermore, exploring methods for handling missing data due to cloud cover or sensor limitations, such as using temporal interpolation or advanced inpainting techniques, would enhance the usability of the system.

User Interface and Dashboard Development: To maximize the practical impact of this research, developing a user-friendly interface or an interactive dashboard could be a valuable next step. This would allow water resource managers, policymakers, and other stakeholders to easily access the generated water body maps, trend analysis results, and other relevant statistics in an intuitive and readily understandable format, facilitating better informed decision-making.

Economic and Impact Analysis: Future studies could investigate the economic benefits and societal impact of accurate and timely water resource monitoring enabled by this type of system. This could involve quantifying the potential savings in water management costs, assessing the benefits for agriculture and other water-dependent sectors, and evaluating the impact on environmental conservation efforts. Understanding the real-world value of the information provided by this automated system can further justify its development and implementation.

Expanding Geographic Scope and Applications: While this study focused on the Vellore region, the developed methodology and tools could be adapted and applied to other geographical areas facing similar water resource management challenges. Furthermore, the core technology could potentially be extended to monitor other types of surface water features, such as snow cover or flooded areas, demonstrating the versatility of the approach.

CHAPTER 8

SUMMARY

This project presents a comprehensive study on water bodies in Vellore by combining deep learning-based water body detection with long-term trend analysis using satellite imagery. The study is divided into two major components. The first component involves developing a U-Net deep learning model trained on Sentinel-2 satellite images to identify water bodies accurately. The model, after training and evaluation, achieved an Intersection over Union (IoU) of 64% and an accuracy of 74.10%, effectively distinguishing water regions from surrounding land. The model's ability to detect water bodies was tested across various seasonal variations and geographic complexities, ensuring its robustness in practical applications.

The second part of the project focuses on water body trend analysis using monthly water spread data from Bhuvan (ISRO). By analyzing water spread area trends from 2012 to 2024, the study identifies seasonal fluctuations and long-term changes. A clear monsoon-driven pattern is observed, with high water spread between July and December and significant depletion during summer (March - May). Over the years, gradual improvement in water retention is evident, likely due to improved rainfall, conservation measures, or reservoir management strategies. However, extreme water loss in May remains a concern, highlighting the necessity for enhanced water conservation efforts.

The findings of this study provide valuable insights into the hydrological behavior of Vellore's water bodies, allowing for better water resource management and planning. Future enhancements include refining the deep learning model, integrating multi-source satellite data, and developing predictive models for forecasting water availability. This work serves as a foundation for AI-driven water monitoring systems, contributing to sustainable water resource management in Vellore and similar regions.

CHAPTER 9
REFERENCES

[1.]Kadhim, I. J., & Pr1emaratne, P. (2023). A novel deep learning framework for water body segmentation from satellite images. Arabian Journal for Science and Engineering, 48(8), 10429-10440.

[2.]Gupta, S., Uma, D., & Hebbar, R. (2023). Analysis and application of multispectral data for water segmentation using machine learning. In Computer Vision and Machine Intelligence: Proceedings of CVMI 2022 (pp. 709-718). Singapore: Springer Nature Singapore.

[3.]Feng, J., & Sui, H. (2021). Deep-learning-based multispectral satellite image segmentation for water body detection. IEEE Geoscience and Remote Sensing Letters, 18(5), 789-793.

[4.]Vasavi, S., Rajeswari, V. N. A. S., & Kalyan, C. V. (2024, March). Enhancing Water Resource Management: A GUI-based Approach for Water Body Classification and Change Detection in VHRS Images. In 2024 Third International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS) (pp. 1-6). IEEE.

[5.]Akiva, P., Purri, M., Dana, K., Tellman, B., & Anderson, T. (2020). H2O-Net: Self-supervised flood segmentation via adversarial domain adaptation and label refinement. IEEE Transactions on Geoscience and Remote Sensing, 59(5), 4213-4225.

[6.]Ouchra, H. A. F. S. A., Belangour, A., & Erraissi, A. L. L. A. E. (2023). Machine learning algorithms for satellite image classification using Google Earth Engine and Landsat satellite data: Morocco case study. IEEE Access.

[7.]Farooq, B., & Manocha, A. (2025). Small water body extraction in remote sensing with enhanced CNN architecture. Applied Soft Computing, 169, 112544.

[8.]Liu, C., Tang, H., Ji, L., & Zhao, Y. (2021). Spatial-temporal water area monitoring of Miyun Reservoir using remote sensing imagery from 1984 to 2020. Remote Sensing of Environment, 256, 112318.

[9.]Digra, M., Dhir, R., & Sharma, N. (2024). Spatio-temporal analysis and prediction of land use land cover (LULC) change in Wular Lake, Jammu and Kashmir, India. Environmental Monitoring and Assessment, 196(9), 782.

[10.]Nagaraj, R., & Kumar, L. S. (2023, May). Surface water body extraction and change detection analysis using machine learning algorithms: A case study of Vaigai Dam, India. In 2023 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT) (pp. 1-6). IEEE.

[11.]Tandra, N., Jyothirmaye, S., & Yechuri, S. (2023, December). TNU-Net: Water Body Segmentation based on the Transformer Network with U-Net Model. In 2023 6th International Conference on Recent Trends in Advance Computing (ICRTAC) (pp. 205-211). IEEE.

[12.]Reddy, N. S., Roshitha, M., Tejomai, R., Vasavi, S., & Jayanthi, S. (2022, November). Water Bodies Detection and Classification from VHRS Images. In 2022 International Conference on Futuristic Technologies (INCOFT) (pp. 1-6). IEEE.

[13.]Zhang, Y., & Wang, C. (2021). Water body detection using deep learning with Sentinel-1 SAR satellite data and land cover maps. IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 9553555.

[14.]Yuan, Q., Wu, L., Huang, Y., Guo, Z., & Li, N. (2023). Water-Body Detection From Spaceborne SAR Images With DBO-CNN. Ieee Geoscience and Remote Sensing Letters.

[15.]Hosseiny, B., Mahdianpari, M., Brisco, B., Mohammadimanesh, F., & Salehi, B. (2021). WetNet: A spatial–temporal ensemble deep learning model for wetland classification using Sentinel-1 and Sentinel-2. IEEE transactions on geoscience and remote sensing, 60, 1 14.