School of Computer Science Engineering and Information Systems

Winter Semester 2024-2025

Department of Computer Applications

PMCA699J – Dissertation -II

Review -2

Date: 20.3.2025

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

23MCA0304 – Sayantan Bhattacharyya

Under the Guidance of

Dr Jagannathan J

Assistant Professor, Senior Grade - I

SCORE

Guide Signature with date

Dr Jagannathan J

Internal Examiner -1

Signature

Internal Examiner-2

Signature

ABSTRACT:

The study of water body detection and trend analysis is critical for sustainable water resource management, particularly in regions facing water scarcity and climatic challenges. This project focuses on detecting and analyzing water bodies in the Vellore district using satellite imagery combined with advanced deep learning techniques. By leveraging high-resolution satellite data, this study aims to identify water bodies with precision, track their temporal changes, and provide insights into seasonal and long-term trends. The analysis will help uncover patterns in water body dynamics, influenced by factors such as monsoonal variations, urbanization, and environmental changes. The project's outcomes have the potential to assist policymakers, urban planners, and environmentalists in devising strategies for better water resource management, planning, and conservation efforts. Additionally, the use of deep learning ensures an automated and scalable approach, making the methodology adaptable for other regions with similar challenges. This research also highlights the importance of integrating technology and environmental studies to address real-world problems. The findings are expected to contribute to a deeper understanding of the region's water dynamics while providing a foundation for further studies in satellite-based environmental monitoring and analysis.

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

[1.] INTRODUCTION:

Water is a crucial resource for human survival, agriculture, and industrial activities. With the growing population and climate change effects, sustainable water management has become a pressing issue. Understanding the distribution, extent, and trends of water bodies is essential to mitigate water scarcity and support effective planning. Remote sensing and satellite imagery have emerged as powerful tools for monitoring and analyzing surface water dynamics over time.

Vellore district, known for its semi-arid climate, experiences significant variations in water availability due to monsoonal fluctuations, urban expansion, and environmental changes. Identifying water bodies and analyzing their trends is essential for water conservation and sustainable management. Traditional methods of water body monitoring rely on manual field surveys, which are time-consuming, expensive, and limited in spatial and temporal coverage.

This project leverages high-resolution satellite imagery and deep learning techniques to automate the detection and analysis of water bodies in Vellore. By applying advanced image processing and deep learning models, the system will provide insights into seasonal and long-term variations in water body dynamics. The outcomes of this research will aid policymakers, urban planners, and environmentalists in making informed decisions regarding water resource management. Additionally, the methodology developed in this study can be adapted for similar regions facing water-related challenges.

[2.] PROBLEM STATEMENT:

Water scarcity is an increasing concern in many parts of the world, including the Vellore district. The availability of surface water bodies fluctuates due to natural and anthropogenic factors such as seasonal rainfall variations, urban encroachment, deforestation, and climate change. Traditional methods of water body mapping and monitoring are labor-intensive, time-consuming, and prone to human errors, making them inefficient for large-scale and real-time applications.

Satellite-based remote sensing combined with deep learning presents an opportunity to address these challenges by automating water body detection and trend analysis. However, existing approaches often suffer from limitations such as low spatial resolution, inconsistent detection accuracy due to cloud cover or terrain variations, and the lack of temporal analysis for long-term water management insights.

This project aims to bridge these gaps by developing an automated, high-accuracy system for detecting water bodies and analyzing their changes over time in Vellore. The study will employ deep learning algorithms trained on high-resolution satellite imagery to enhance precision and scalability. The insights generated will provide valuable information on seasonal variations, long-term trends, and environmental impacts affecting water resources. This research will contribute to more efficient water resource management, conservation strategies, and policy-making effort

[3.] OBJECTIVES:

The primary objectives of this project are:

- **To detect and map water bodies** in Vellore district using high-resolution satellite imagery.
- **To analyze temporal changes** in water bodies by examining seasonal and long-term variations.
- **To evaluate factors influencing water body dynamics**, such as monsoonal patterns, urbanization, and environmental changes.
- **To assist policymakers, urban planners, and environmentalists** by providing actionable insights for better water resource management and conservation strategies.
- **To contribute to the field of satellite-based environmental monitoring**, establishing a framework that can be adapted to other regions facing similar challenges.

[4.] SCOPE OF THE PROJECT:

The scope of this project extends across various domains, including remote sensing, deep learning, environmental analysis, and urban planning. It focuses on:

- **Geographical Coverage:** The study area is Vellore district, which includes a range of water bodies such as lakes, reservoirs, and seasonal streams.
- **Data Acquisition:** High-resolution satellite images from sources like Sentinel, Landsat, or commercial satellite providers will be used for analysis.

- **Technology Utilization:** Deep learning models, including CNN-based architectures and transformer networks, will be employed for automated water body detection.
- **Temporal Analysis:** The project will examine historical satellite imagery to track water body changes over time, considering seasonal and climatic influences.
- **Application and Impact:** The findings will support local authorities, policymakers, and researchers in formulating strategies for sustainable water management, urban planning, and climate resilience.
- **Scalability:** The proposed system will be adaptable for analyzing water bodies in other regions experiencing similar water scarcity and climatic challenges.

[5.] PROPOSED SYSTEM:

The proposed system aims to develop an advanced satellite-based water body detection and trend analysis framework using deep learning techniques. The system will consist of the following key components:

1. **Data Collection:** Satellite images will be acquired from reliable sources such as Sentinel-1, Sentinel-2, Landsat, or commercial high-resolution imagery providers. Pre-processing techniques will be applied to remove noise, correct distortions, and enhance image quality.
2. **Deep Learning Model Development:** A deep learning architecture, such as CNN-based models or transformer networks (e.g., TNU-Net), will be employed to detect and segment water bodies from satellite images. The model will be trained on labeled datasets to improve accuracy.
3. **Temporal Trend Analysis:** Historical satellite images will be analyzed to identify seasonal variations and long-term trends in water body distribution. Machine learning techniques will be applied to detect patterns and correlations between environmental factors and water body changes.
4. **Visualization and Interpretation:** The system will generate visual reports and statistical insights on water body trends, which will be presented using GIS-based mapping tools. The results will be used to assess water resource sustainability and policy interventions.
5. **Deployment and Scalability:** The model will be designed for scalability, allowing it to be applied in other regions with similar water scarcity and environmental challenges.

Additionally, it will be optimized for real-time or periodic updates based on newly available satellite imagery.

This proposed system will ensure high accuracy, automation, and efficiency in water body detection and trend analysis, providing crucial insights for environmental monitoring and sustainable water resource management.

[6.]LITERATURE SURVEY:

S.NO	TITLE	MERITS	DEMERITS
1	A Novel Deep Learning Framework for Water Body Segmentation from Satellite Images.	Proposes a deep learning-based architecture that effectively detects water bodies using low to medium-resolution RGB images, eliminating the need for high-resolution multichannel data.	The model's performance may be limited when applied to high-resolution images or in scenarios requiring detailed water body delineation.
2	Analysis and Application of Multispectral Data for Water Segmentation Using Machine Learning	Demonstrates that shortwave infrared bands (B11 and B12) are most effective for water segmentation and introduces 'BandNet', a neural network achieving high accuracy with reduced computational resources.	The study focuses on single-band analysis, which may not capture the full spectral complexity of water bodies in diverse environments.
3	Deep-Learning-Based Multispectral Satellite Image Segmentation for Water Body Detection	Utilizes deep learning techniques to enhance water body detection accuracy in multispectral satellite images.	The reliance on multispectral data may limit applicability in regions where such data is unavailable or of low quality.
4	Enhancing Water Resource Management: A GUI-based Approach for Water Body Classification and Change Detection in VHRS Images	Introduces a user-friendly GUI for water body classification and change detection, facilitating practical applications in water resource management.	The approach may require high-resolution imagery, which can be resource-intensive to obtain and process.

5	H2O-Net: Self-Supervised Flood Segmentation via Adversarial Domain Adaptation and Label Refinement	Presents a self-supervised model that effectively segments flood areas, reducing the need for extensive labeled data.	The model's focus on flood scenarios may limit its applicability to other types of water body segmentation tasks.
6	Machine Learning Algorithms for Satellite Image Classification Using Google Earth Engine and Landsat Satellite Data: Morocco Case Study	Applies machine learning algorithms within the Google Earth Engine platform for efficient satellite image classification, demonstrating practical utility in a specific regional context.	The case study's regional focus may limit the generalizability of the findings to other geographic areas.
7	Small Water Body Extraction in Remote Sensing with Enhanced CNN Architecture	Develops an enhanced CNN architecture tailored for extracting small water bodies from remote sensing data, improving detection accuracy.	The specialized focus on small water bodies may reduce effectiveness when applied to larger or more complex water bodies.
8	Spatial-Temporal Water Area Monitoring of Miyun Reservoir Using Remote Sensing Imagery from 1984 to 2020	Provides a comprehensive temporal analysis of water area changes over several decades, offering valuable insights into long-term water resource trends.	The study is specific to the Miyun Reservoir, which may limit the applicability of the findings to other regions or water bodies.
9.	Spatio-Temporal Analysis and Prediction of Land Use Land Cover (LULC) Change in Wular Lake, Jammu and Kashmir, India	Combines spatio-temporal analysis with predictive modeling to assess land use and land cover changes, aiding in environmental planning.	The focus on a specific geographic area may limit the generalizability of the methodologies to other regions.
10	Surface Water Body Extraction and Change Detection Analysis Using Machine Learning Algorithms: A Case Study of	Employs machine learning algorithms for effective water body extraction and change detection, providing a practical case study application.	The case study approach may limit the applicability of the methods to other contexts without further validation.

	Vaigai Dam, India		
11	TNU-Net: Water Body Segmentation Based on the Transformer Network with U-Net Model	Integrates transformer networks with U-Net architectures to enhance water body segmentation performance.	The complexity of the model may lead to increased computational requirements and potential overfitting in limited data scenarios.
12	Water Bodies Detection and Classification from VHRS Images	Focuses on detecting and classifying water bodies using very high-resolution satellite images, improving accuracy in detailed analyses.	The reliance on very high-resolution images may limit applicability in regions where such data is not readily available.
13	Water Body Detection Using Deep Learning with Sentinel-1 SAR Satellite Data and Land Cover Maps	Leverages deep learning techniques with Sentinel-1 SAR data and land cover maps to improve water body detection accuracy.	The dependence on SAR data and land cover maps may limit applicability in areas lacking such datasets.
14	Water-Body Detection from Spaceborne SAR Images with DBO-CNN	Introduces the DBO-CNN model for effective water body detection using spaceborne SAR images.	The model's performance may be affected by the quality and resolution of the SAR images used.
15	WetNet: A Spatial–Temporal Ensemble Deep Learning Model for Wetland Classification Using Sentinel-1 and Sentinel-2	Proposes an ensemble deep learning model that effectively classifies wetlands by integrating data from multiple satellite sources.	The model's complexity may result in higher computational demands and may require extensive training data.

[6.1] FINDINGS IN LITERATURE SURVEY:

The review of existing research on water body detection and trend analysis using satellite imagery and deep learning has yielded several key insights:

1. **Effectiveness of Deep Learning for Water Segmentation:**
   Multiple studies have demonstrated the superiority of deep learning models such as

CNNs, U-Net, and transformers for water body segmentation compared to traditional machine learning techniques. Models like **TNU-Net** and **DBO-CNN** have shown promising results in enhancing segmentation accuracy.

2. **Significance of Multispectral and SAR Data:**
   Research highlights the effectiveness of **shortwave infrared (SWIR) bands** (e.g., B11 and B12) for water detection, as they enhance the contrast between water and land. Additionally, the use of **Sentinel-1 SAR data** has improved water body detection in challenging conditions such as cloud cover or varying land types.
3. **Limitations of Resolution and Data Availability:**
   Studies relying on **low to medium-resolution** images may struggle with precise water body delineation. Methods that use **very high-resolution (VHRS) imagery** offer greater accuracy but are computationally expensive and not always accessible.
4. **Temporal Analysis is Underexplored:**
   While some studies provide **spatio-temporal analysis**, long-term monitoring using **historical satellite images** remains underexplored. More research is needed to integrate **trend analysis with predictive modeling** for better forecasting of water body changes.
5. **Challenges in Generalization Across Regions:**
   Several studies focus on specific geographic regions, such as **Miyun Reservoir (China), Wular Lake (India), and Morocco**. This highlights the need for models that are adaptable to diverse climatic and geographical conditions.
6. **Automated vs. User-Guided Approaches:**
   Some studies propose **GUI-based classification tools** for practical applications in water resource management, while others emphasize **fully automated deep learning models** for scalability.

[7.] METHODOLOGY:

- **Data Collection:** Gathering stunning high-resolution satellite images from the Sentinel-1, Sentinel-2, and Landsat missions, ensuring a detailed and expansive view of the study area, capturing the intricate dynamics of our planet's water bodies.
- **Preprocessing:** Employing advanced noise removal techniques, atmospheric correction processes, and image enhancement methods to refine the data, thereby elevating its clarity and overall quality to a level suitable for precise analysis.
- **Model Development:** Crafting and fine-tuning cutting-edge convolutional neural

network (CNN) architectures or sophisticated transformer-based models, such as TNU-Net, dedicated to the meticulous segmentation of water bodies, revealing their shapes and boundaries with remarkable accuracy.

- **Training and Evaluation:** Harnessing meticulously curated labeled datasets of water bodies to educate and validate our models. This rigorous process involves comparing model predictions against ground truth data, ensuring the utmost precision in the results obtained.
- **Temporal Analysis:** Conducting an insightful comparative analysis of both historical and contemporary satellite imagery, allowing us to unveil the seasonal variations and persistent transformations occurring within our precious water resources over time.

[8.] SOFTWARE REQUIREMENTS:

Functional Requirements:

Detection of Water Bodies: The system must employ advanced algorithms to automatically identify and delineate water bodies within satellite imagery or aerial photography. This feature should be capable of detecting various types of water bodies, including lakes, rivers, and wetlands, while distinguishing them from surrounding land areas.

Analysis of Temporal Trends: The application should facilitate the examination of changes in water body characteristics over time. This includes analyzing variations in size, distribution, and water quality indicators across different seasons and years, enabling stakeholders to understand long-term ecological and environmental trends.

Data Preprocessing and Enhancement: The platform should include features for preprocessing raw data, such as noise reduction and image correction. Additionally, it should enhance data quality through normalization or enrichment to ensure that subsequent analyses yield accurate and reliable results.

Non-Functional Requirements:

- **Scalability and Adaptability:** The system must be designed to handle increasing volumes of data and adapt to new technologies and methodologies. It should accommodate additional features and scaling options to meet evolving user needs while maintaining optimal performance levels.

- **High Computational Efficiency:** The application must utilize computational resources efficiently, ensuring fast processing and analysis of large datasets without excessive resource consumption. This will facilitate timely retrieval of results, especially in scenarios requiring real-time data processing.
- **Accuracy and Reliability:** It is critical that the algorithms employed in water body detection and analysis yield high accuracy levels. The system must be rigorously tested and validated against ground truth data to guarantee dependable outputs for stakeholders.

[9.] SYSTEM ARCHITECTURE:

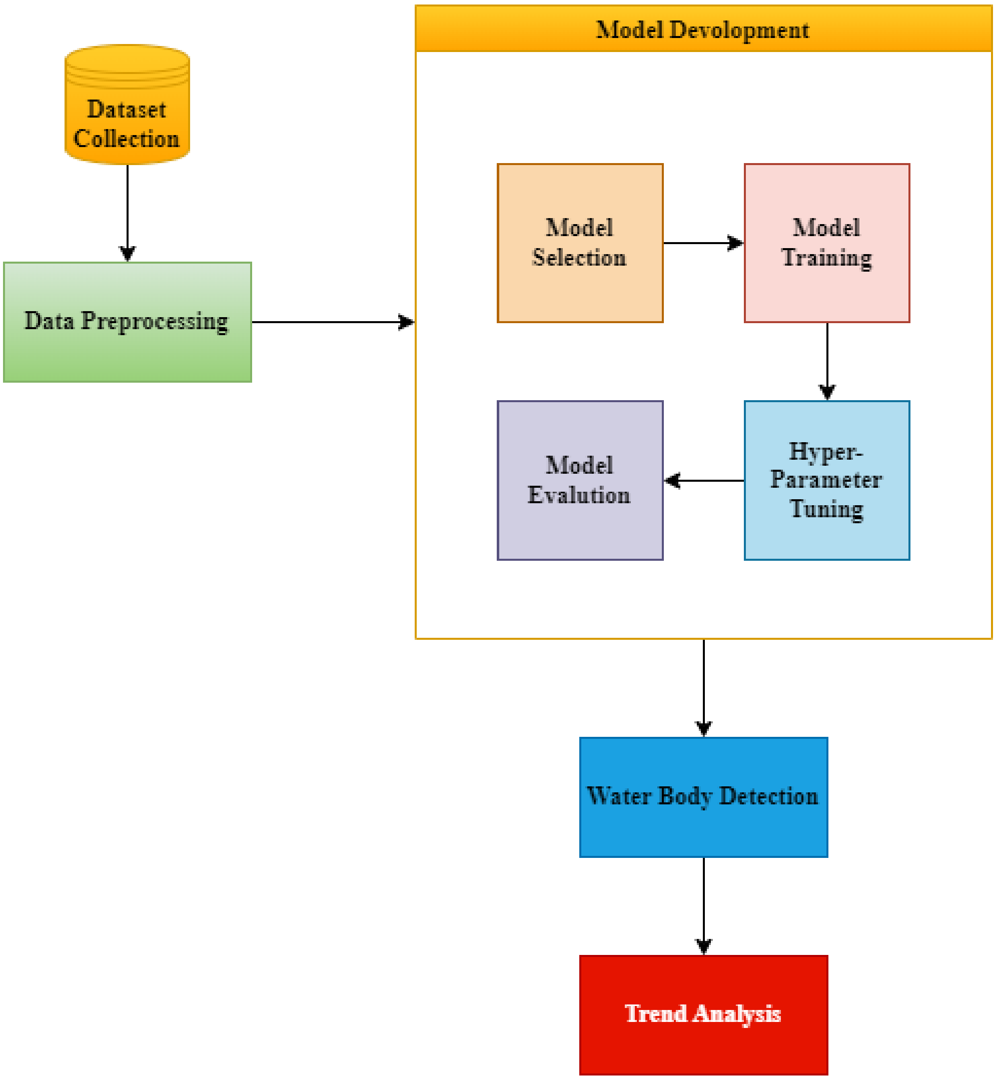

Fig. 9.1 System Architecture

[10]. UML DIAGRAMS:

Use Case Diagram:

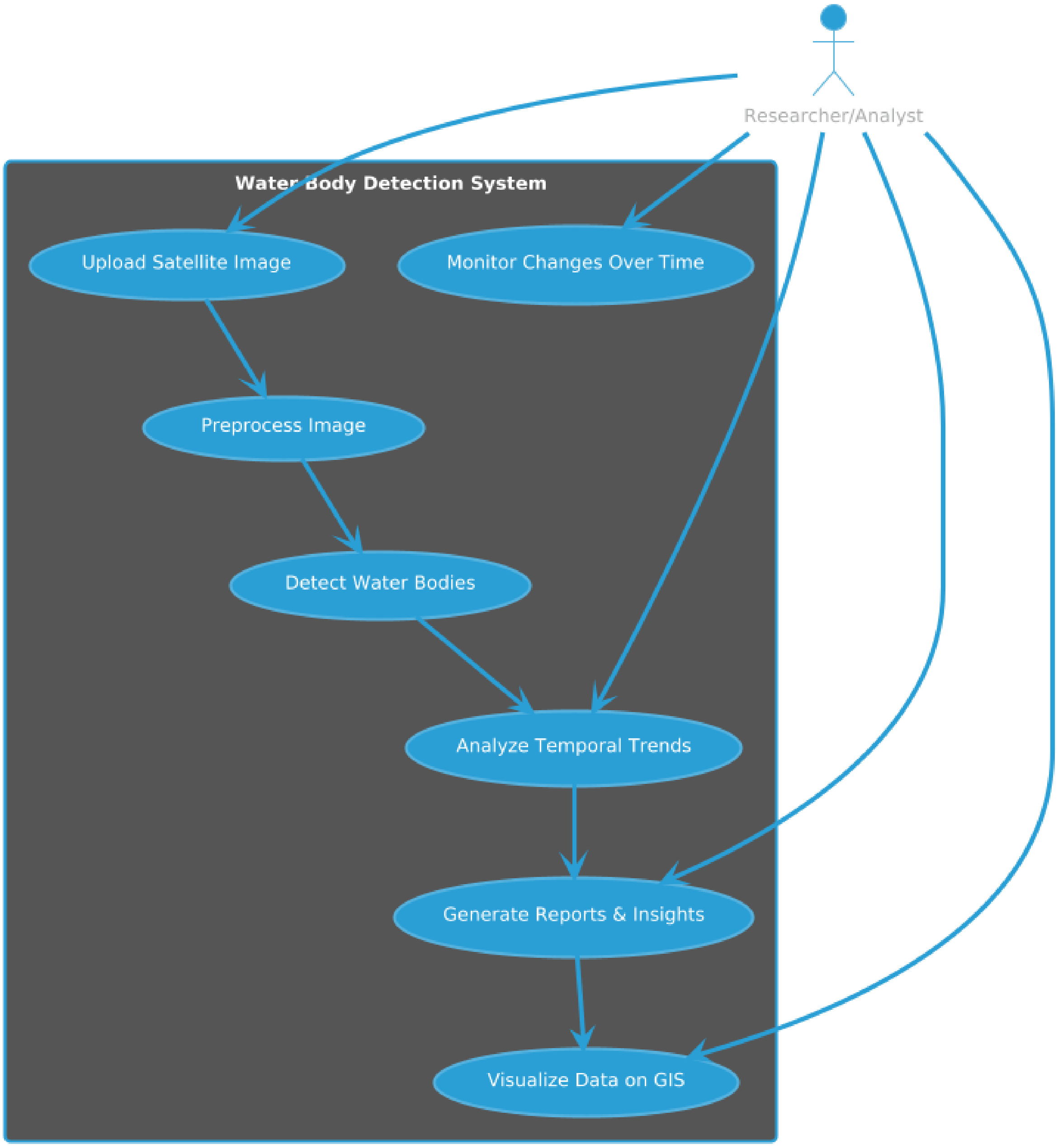

Fig. 10.1 Use Case Diagram

Class Diagram:

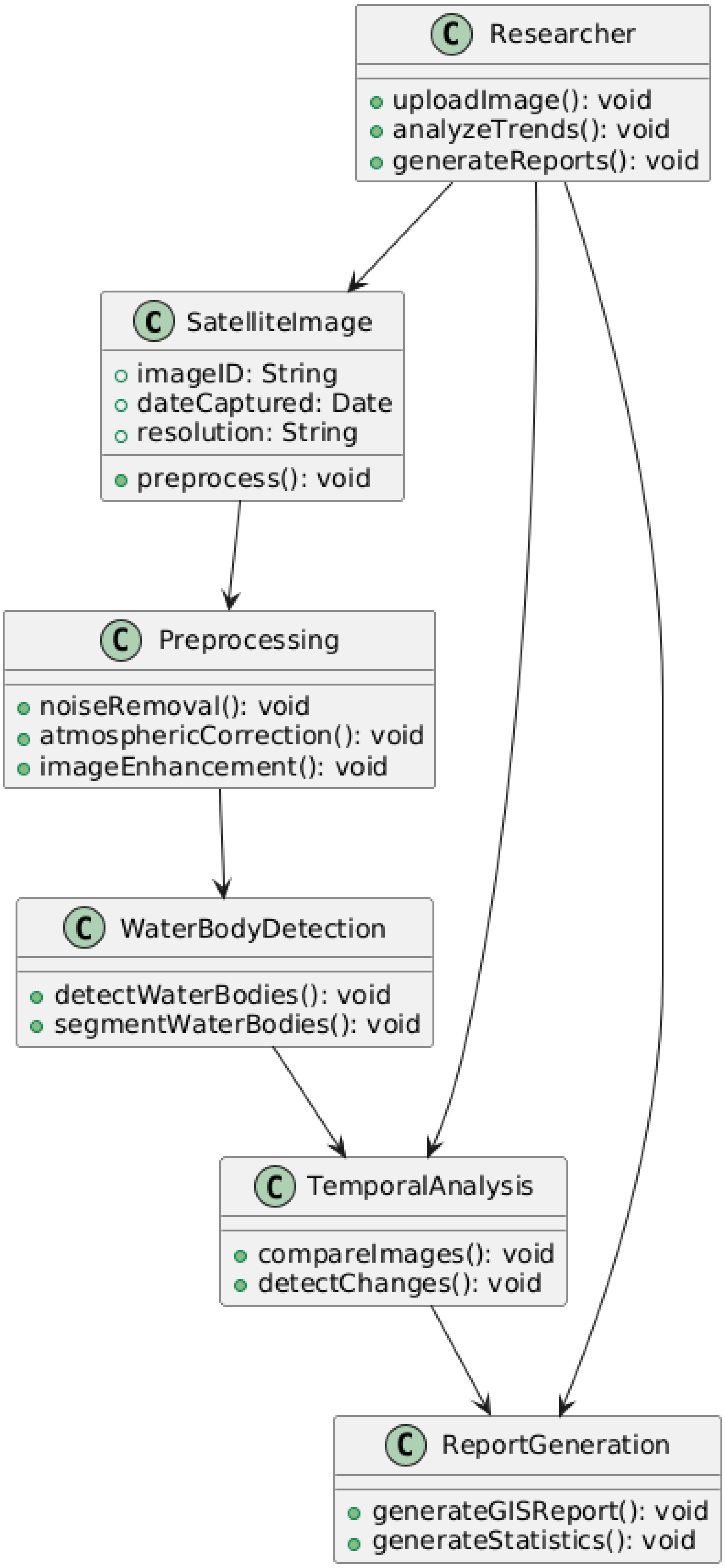

Fig. 10.2 Class Diagram

[11]. IMPLEMENTATION:

11.1. PROJECT PLAN

The project is divided into multiple phases to ensure systematic development and accurate results:

Phase 1: Data Collection and Preprocessing

- Acquire satellite images from Sentinel-2 and Landsat covering the Vellore region over a 30-year period.
- Apply preprocessing techniques, including noise removal, atmospheric correction, and image enhancement.

Phase 2: Model Training and Evaluation

- Train a U-Net model using 2841 high-resolution satellite images and corresponding water body masks.
- Evaluate model performance using metrics such as IoU (Intersection over Union), accuracy, and F1-score.

Phase 3: Water Body Detection and Mask Generation

- Apply the trained U-Net model to generate water masks from the acquired dataset.
- Verify the accuracy of detected water bodies using ground truth data.

Phase 4: Trend Analysis and Temporal Changes

- Analyze seasonal, yearly, and long-term trends by examining changes in water body extent over the study period.
- Identify patterns and factors influencing water body dynamics.

Phase 5: Report Generation and Visualization

- Create comprehensive reports that visualize water body trends and highlight key insights.
- Use GIS-based mapping tools for effective visualization and presentation of findings.

Each phase is critical to ensuring the accuracy and relevance of the analysis, with the ultimate goal of aiding policymakers and environmentalists in water management.

11.2 SAMPLE CODE

```
from google.colab import drive
drive.mount('/content/drive/')


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

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]

# 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
```

```python
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)
```

```
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()

# Make predictions
predictions = model.predict(images_test)

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)
    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('Orginal Image')

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

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

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

    plt.tight_layout()
    plt.show()
```

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

11.3 SAMPLE SCREEN SHOTS

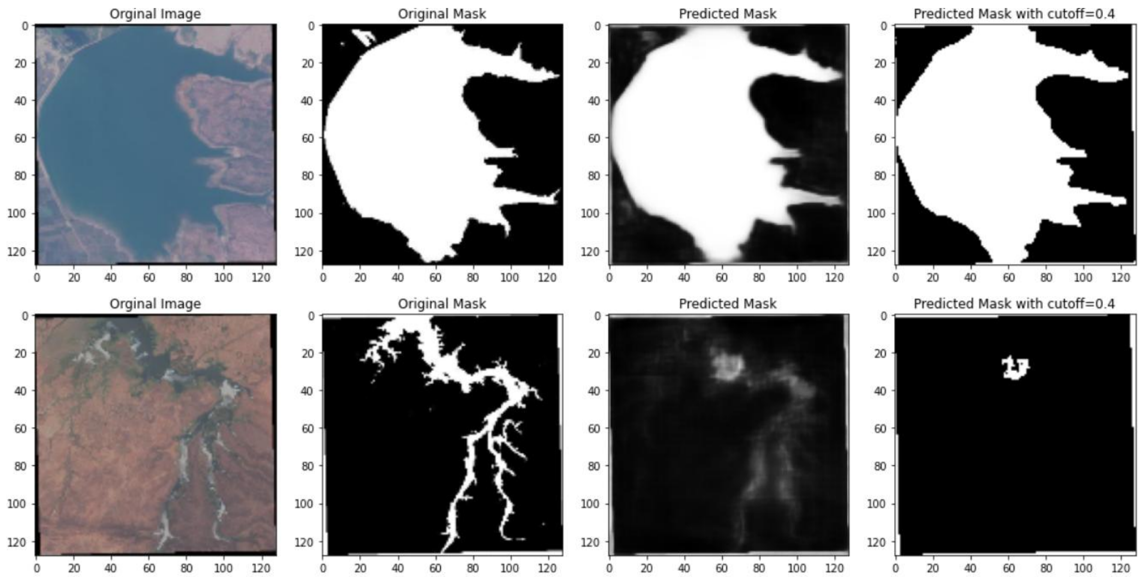

Fig 11.1 Model Output

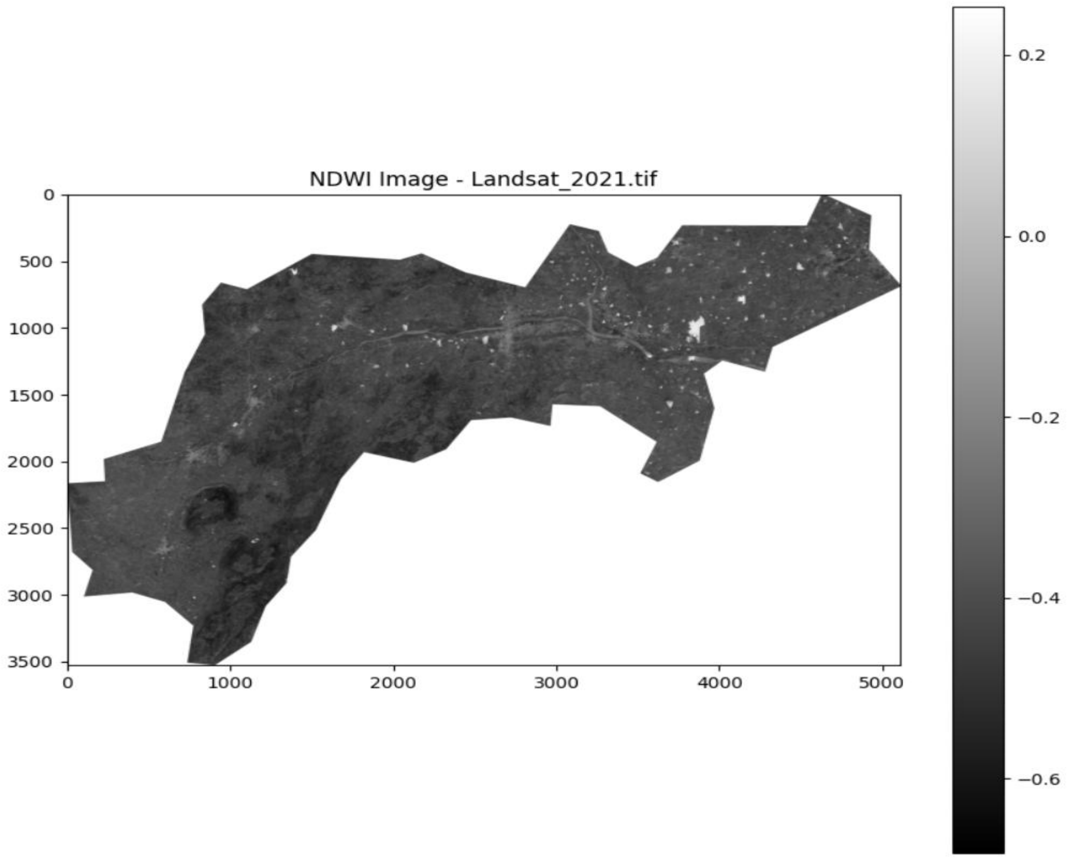

Fig 11.2 Vellore NDWI Image

e

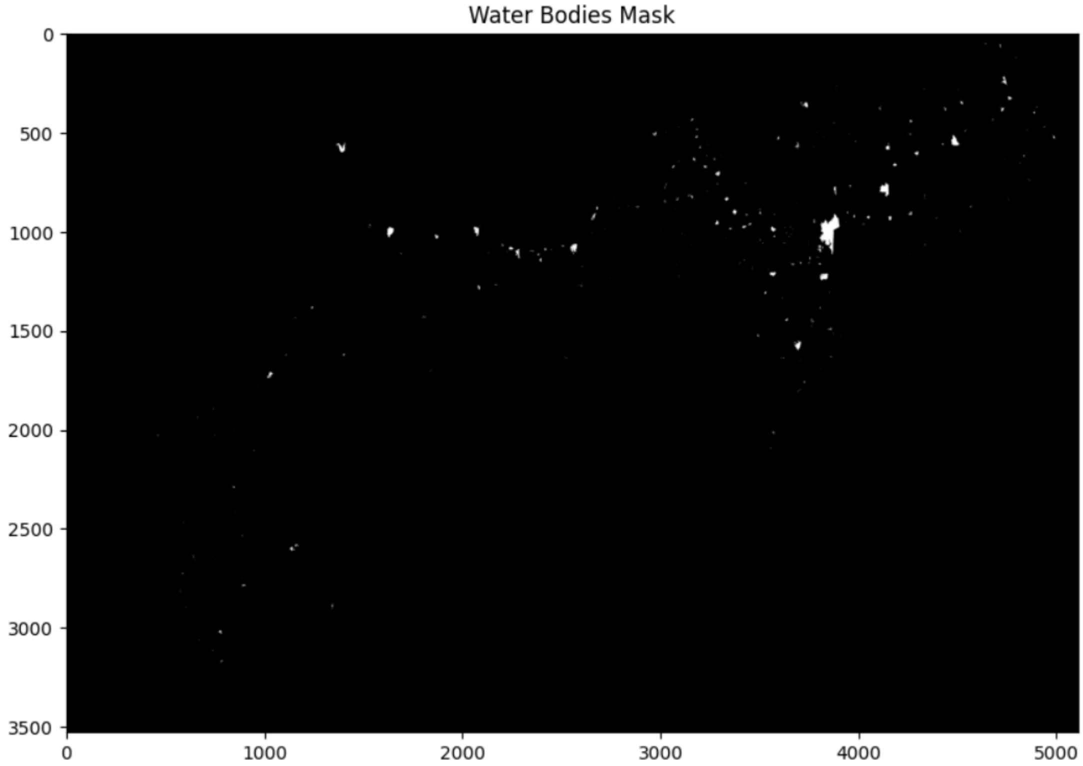

Fig 11.3 Vellore Water Mask

[12]. SUMMARY:

This project aims to detect water bodies and analyze temporal trends in Vellore by utilizing advanced satellite imagery and cutting-edge deep learning algorithms. Focusing on high accuracy, scalability, and real-time monitoring, the proposed methodology seeks to provide critical insights that empower policymakers to make informed decisions regarding sustainable water resource management and environmental conservation efforts. By integrating machine learning with geospatial analysis, the approach will facilitate the identification of changes in water bodies over time, supporting targeted interventions and effective resource allocation. Additionally, the innovative techniques developed in this project can be adapted and applied to other regions experiencing similar challenges in water resource management and sustainability.

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.