**SIDDAGANGA INSTITUTE OF TECHNOLOGY, TUMAKURU-572103**

**(An Autonomous Institute under Visvesvaraya Technological University, Belagavi)**

# Project Report on

# "Water Body Detection Using Deep Learning From Remote Sensing"

submitted in partial fulfillment of the requirement for the award of the degree of

**BACHELOR OF ENGINEERING**

**in**

**INFORMATION SCIENCE & ENGINEERING**

**Submitted by**

| | |
|---|---|
| JEEVAN V S | (1SI20IS017) |
| KARTHIK | (1SI20IS020) |
| MITHILESH K | (1SI20IS029) |
| GANGADHAR S | (1SI21IS403) |

under the guidance of

**Dr. Nagaratna B. Chittaragi**

Assistant Professor

Department of ISE,

SIT, Tumakuru-03

**DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING**

**2023-24**

**SIDDAGANGA INSTITUTE OF TECHNOLOGY, TUMAKURU-572103**

(An Autonomous Institute under Visvesvaraya Technological University, Belagavi)

**DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING**

# CERTIFICATE

Certified that the project work entitled "Water Body Detection Using Deep Learning From Remote Sensing" is a bonafide work carried out by JEEVAN V S (1SI20IS017), KARTHIK (1SI20IS020), MITHILESH K (1SI20IS029) and GANGADHAR S (1SI21IS403) in partial fulfillment for the award of degree of Bachelor of Engineering in Information Science & Engineering from Siddaganga Institute of Technology, an autonomous institute under Visvesvaraya Technological University, Belagavi during the academic year 2023-24. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report deposited in the department library. The Project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the Bachelor of Engineering degree.

| | | |
|---|---|---|
| Dr. Nagaratna B. Chittaragi | Dr. R. Aparna | Dr. S. V. Dinesh |
| Assistant Professor | Prof. and Head | The Principal |
| Dept. of ISE | Dept. of ISE | SIT, Tumakuru-03 |
| SIT, Tumakuru-03 | SIT, Tumakuru-03 | |

**External viva:**

**Names of the Examiners** **Signature with date**

**1.**

**2.**

# ACKNOWLEDGEMENT

We offer pranams at the lotus feet of **His** Holiness, **Dr. Sree Sree Sivakumara Swamigalu**, Founder President and **His** Holiness, **Sree Sree Siddalinga Swamigalu**, President, Sree Siddaganga Education Society, Sree Siddaganga Math for bestowing upon their blessings.

We deem it as a privilege to thank **Dr. M N Channabasappa**, Director, SIT, Tumakuru, **Dr. Shivakumaraiah**, CEO, SIT, Tumakuru, and **Dr. S V Dinesh**, Principal of SIT, Tumakuru for fostering an excellent academic environment in this institution, which made this endeavor fruitful.

We would like to express our sincere gratitude to **Dr. R Aparna**, Professor and Head of the Department of ISE, SIT, Tumakuru for her encouragement and valuable suggestions.

We thank our guide **Dr. Nagaratna B Chittaragi**, Assistant Professor, Department of Information Science & Engineering, SIT, Tumakuru for the valuable guidance, advice and encouragement.

| | |
|---|---|
| Jeevan V S | (1SI20IS017) |
| Karthik | (1SI20IS020) |
| Mithilesh K | (1SI20IS029) |
| Gangadhar S | (1SI21IS403) |

# Abstract

Water body detection from satellite imagery plays a vital role in environmental monitoring such as fluctuations and seasonal variations on water bodies. Water body detection system assists in flood Management for Identification of water bodies to assess flood risks, predict flood events, plan mitigation strategies, flood defenses, and drainage systems. It also assists in natural Resource Management to map water bodies for water allocation such as irrigation, drinking, and industry etc. This study investigates the application of deep learning approaches for automatic water body detection. We have explored the limitations of traditional models like K-Nearest neighbor (KNN) and Bayes algorithm with respect to water body detection tasks as they in capturing semantic information. We then evaluate the performance of various deep learning architectures, including Convolutional Neural Networks (CNNs) and ensemble based Random Forest technique. Our findings demonstrate that U-Net, a CNN architecture specifically designed for segmentation, achieves a moderate accuracy of 80%. However, a Random Forest model surpasses this performance by achieving a significantly higher accuracy of 93%. Hyperparameter tuning, further improves the performance of Random Forest to 94%, emphasizing the effectiveness of ensemble learning approaches in accurate water body detection from satellite imagery. Further, employing a transfer learning technique namely ResNet50, which is also optimized for image classification, results in considerably lower accuracy. This study has highlighted the importance of selecting appropriate architectures for semantic segmentation tasks with respect to water body detection.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

India is a large country with diverse landscapes and climates, resulting a wide range of water bodies, including lakes, rivers, pond, and wetlands. These water bodies plays a biggest role in the country's economy, agriculture, and ecosystem. They are also under increasing pressure from human activities, such as urbanization, agriculture, and industrial pollution. Remote sensing techniques have emerged as a powerful technique for monitoring and managing water resources.Remote sensing data can provide a view and cost-effective way to map and monitor water bodies over large areas.

Water body detection from satellite images using deep learning is a crucial task with diverse applications in environmental monitoring, urban planning, and natural resource management. Satellites, such as Landsat and Sentinel, provide valuable data for this purpose, offering a comprehensive view of the Earth's surface, including water bodies like lakes, rivers, and reservoirs. However, the sheer volume and complexity of satellite imagery require sophisticated techniques for analysis, making deep learning an ideal candidate for automated water body detection [2]. The process starts with acquiring satellite imagery from different sources. Preprocessing steps like radiometric and geometric corrections ensure image quality. Then, contrast stretching and histogram equalization techniques enhance the visibility of water bodies within the imagery. A critical step involves creating labeled training datasets. Each pixel is meticulously labeled as "water" or "non-water" through manual or semi-automated tools.

Convolutional Neural Networks (CNNs) are particularly adept at this task due to their ability to learn spatial features from images. Architectures like U-Net and Mask R-CNN have shown success in semantic segmentation tasks, enabling the accurate identification and delineation of water bodies. The model is then trained by optimizing its parameters using algorithms like SGD or Adam to learn from the labeled data. Finally, a separate test dataset is used to evaluate the model's performance with metrics like Intersection over Union (IoU) and F1 score. Post-processing techniques like removing small detections and smoothing boundaries further refine the final output. Challenges remain due to variations in water characteristics (color, turbidity) and interference from clouds and shadows in the imagery. Future research will focus on improving detection accuracy and efficiency, integrating additional data sources like radar imagery, and developing methods for long-term water body monitoring. Analyzing these spatial and temporal changes in surface water holds significant value for water resource management, biodiversity studies, and emergency response efforts [3].

Understanding how surface water changes geographically (spatially) and over time (temporally) is critical for managing water resources, protecting biodiversity, responding to emergencies like floods and droughts, and studying global climate change. Remote sensing offers a powerful and convenient tool to identify surface water across vast areas using data from satellites and airplanes. Over the past few decades, scientists have developed numerous methods to extract water bodies from this remote sensing data. These methods fall into five main categories: Threshold Method: Simple but Flawed: This easy-to-use approach struggles to distinguish between water and shadows on the Earth's surface. Spectral Index Method: Popular but Not Perfect: The most common method, it's effective but can't remove shadows, ice, or snow from the image. Object-Oriented Method: High-Resolution, High-Effort: Ideal for high-resolution images, this method requires significant time and processing power to segment the image [**?**].

Water bodies, as one of the fundamental elements of the earth, are not only essential to the natural ecological cycle, but closely geared to human life as well, such as health, irrigation, electric power generation, and so on Water bodies mainly include rivers, canals, ponds, lakes, and seas. It should be noted that for water bodies that change with seasonal climate, for example, seasonal rivers that are covered by water in the rainy season but sand in the summer. When images are acquired during the rainy season, they are naturally identified as water bodies. Thanks to the earth observation capability of the above mentioned sensors and advanced image processing techniques, it is possible to extract water bodies from RS imagery. For optical RS images, the methods based on handcrafted features appear and demonstrate fine performance [4].

## 1.1 Motivation

The motivation behind this project lies in the critical importance of monitoring and managing water resources, particularly water bodies such as lakes, rivers, and reservoirs. Water bodies are vital natural resources that support various ecological, social, and economic functions. However, their sustainable management requires accurate and timely information about their extent, dynamics, and condition.Traditional methods of water body detection and monitoring are often labor-intensive, time-consuming, and costly. Additionally, they may be limited in their spatial and temporal coverage. By leveraging deep learning techniques on satellite image datasets, this project aims to overcome these limitations and offer a more efficient, scalable, and cost-effective solution for water body detection.

The use of deep learning enables the automated extraction of water bodies from satellite imagery with high accuracy and reliability. This approach not only reduces the manual effort required but also allows for the analysis of large-scale and multi-temporal datasets, facilitating comprehensive monitoring and assessment of water resources.Furthermore, accurate detection of water bodies using satellite imagery can contribute to various applications, including water resource management, environmental monitoring, disaster response, urban planning, and climate change studies. By providing timely and precise information about the distribution and dynamics of water bodies, this project ultimately aims to support informed decision-making and sustainable development efforts.

## 1.2 Objectives

In this project, the development of a robust deep learning model is paramount for the automated detection and delineation of water bodies in satellite imagery. Leveraging state-of-the-art deep learning techniques, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), the aim is to achieve high accuracy in identifying water bodies across various types of satellite imagery.

To enhance the accuracy of the detection algorithms, the project will explore innovative approaches such as transfer learning, ensemble methods, and semantic segmentation. Transfer learning involves leveraging pre-trained models on large datasets to bootstrap the training process, while ensemble methods combine multiple models to improve overall performance. Semantic segmentation techniques allow for pixel-level classification, enabling precise delineation of water bodies and accurate boundary detection.

Efforts will also be dedicated to optimizing the efficiency of the detection process to enable real-time or near-real-time monitoring of water bodies. This involves optimizing model architectures, utilizing parallel processing techniques, and streamlining data preprocessing pipelines to reduce computational overhead and latency.

## 1.3 Organisation of the report

The report is divided into six chapters and they are as follows:-

**Chapter 1: Introduction:** A detailed overview of the motivation, background and objectives of the proposed work is presented in this chapter.

**Chapter 2: Literature Survey:** Discusses about the literature review about the prior works related to water body detection and the machine learning models incorporated for this task. This chapter also given an emphasis on needs of the proposed work by highlighting the limitations of the existing models in the literature.

**Chapter 3: Requirement Specifications:** This chapter gives information regarding the hardware and software requirements of the proposed work also highlights on the various tools and techniques used for implementation of the current work.

**Chapter 4: System Design:** This chapter presents the overall system design, along with work flow diagrams.

**Chapter 5: Implementation Methods:** This chapter highlights on the details of implementation of the system along with experimental setup information.

**Chapter 6: Results And Discussion:** Detailed analysis and discussions of the achieved results from current system is presented along with discussion of the overall summary.

**Chapter 7: Conclusion And Future work:** Walks through the conclusion and future works of the proposed system.

# Chapter 2

# Literature Survey

Extracting information on water regions from satellite images is acrucial for various applications like flood monitoring, resources management and studying environmental changes. This chapter presents brief overview of few of the works proposed for detection of the water bodies from satellite images. This chapter also given information about the few deep learning based works. The proposed system focuses on the detection of waterbodies using satellite imagery, employing advanced semantic segmentation techniques [5]. A correct suitable deep learning model for semantic segmentation, such as U-Net or DeepLab, is selected and trained using annotated datasets to distinguish between water and non-water regions effectively. The trained model undergoes rigorous validation and fine-tuning processes to optimize its performance, minimizing false positives and false negatives [6]. Finally, the model's effectiveness is evaluated using quantitative metrics, and output semantic segmentation maps are generated, providing valuable insights into the distribution and extent of waterbodies within the study area.Multispectral and hyperspectral remote sensing images offer rich data for studying surface features like water bodies.

Research in this remote sensing has been ongoing for years due to the importance of water conservation. Various digital image processing techniques have been employed to develop effective systems for quantifying changes in water bodies over time. These methods, categorized into single-band, spectral index, machine learning, and spectral mixture analysis approaches, have been extensively explored and experimented with using data from variety of remote sensing sensors and satellites [7]. A brief review of the few of the existing models which are proposed with respect to water body detection.

A study has been proposed which aimed to assess the accuracy of mapping river line water bodies using image processing techniques with Landsat 5 TM data. The study quantified the classification accuracy of single-band density slicing compared to a six-band maximum likelihood classification. Results demonstrated that Landsat TM data can accurately delineate water bodies, with density slicing achieving an overall accuracy of 96.9 percentage, a producer's accuracy of 81.7 percentage, and a user's accuracy of 64.5 percentage, comparable to multispectral classification methods [8].

A paper addresses the significance of water body classification from high-resolution optical remote sensing (RS) images, highlighting its practical applications and ongoing advancements. Through a comprehensive review of approximately 200 papers, it identifies challenges and opportunities, particularly focusing on deep learning techniques. The paper categorizes existing methods, presents practical applications, and evaluates representative approaches on widely used datasets. Additionally, it proposes future research directions, aiming to provide insights and inspiration for further exploration in this field [9].

Another paper highlights the significance of monitoring water resources through remote sensing, focusing on spatial and temporal distributions. It discusses the use of satellite data for extracting key features of water bodies, emphasizing efficiency and safety. Methods for water information extraction, including reflection characteristics and challenges, are reviewed, with future applications forecasted [10].

In this study, application of Fully Convolutional Networks (FCN) for extracting water bodies from Very High spatial Resolutions (VHR) optical images with limited training samples. Through extensive validation using GaoFen-2 images, the study identifies four key factors affecting FCN performance: input features, training data, transfer learning, and data augmentation. Results demonstrate FCN's robustness and cost-effectiveness compared to other methods like NDWI, SVM, and SM, particularly in urban areas with mixed water and shadows. Despite the influence of key factors on FCN performance, selecting suitable settings is manageable, offering insights applicable to extracting various land covers from VHR imagery [11].

A study introduces a methodology for estimating water body extent using UAV-collected remotely sensed data. Three sensors—RGB camera, thermal infrared camera, and laser scanner—were tested on an Aibot X6 platform across various water body types. Processing yielded 2.5-D and 2-D geospatial products used for water body extent estimation via three image processing strategies. The most effective product was identified as a four bands RGB+TIR ortho mosaic, achieving a kappa coefficient above 0.9 for water body identification and comparable planar accuracy to similar studies [12].

A paper in the literature addresses the limitations of traditional deep semantic segmentation networks (DSSNs) in water-body detection by introducing a novel multiscale residual network (MSResNet) utilizing self-supervised learning (SSL). The MSResNet effectively captures multiscale and multishape characteristics of water bodies while leveraging unlabeled data for enhanced training. Experiments on two public datasets show that MSResNet outperforms existing deep learning backbones, with SSL further improving detection performance [13].

A few of the authors have introduced a two-level machine-learning framework for identifying urban water types from high-resolution remote-sensing images, addressing the urgent need for water resource management in rapidly urbanizing areas. The framework first extracts water bodies at the pixel level and then identifies water types at the object level using a combination of indexes and features. Validation on GeoEye-1 and WorldView-2 images in Wuhan and Shenzhen demonstrates satisfactory accuracies for both water extraction and water type classification in complex urban environments [14].

An article has introduced MC-WBDN, a novel DCNN model designed for automated water body detection from satellite imagery, crucial for urban hydrological studies. MC-WBDN integrates innovative components such as a multichannel fusion module and Enhanced Atrous Spatial Pyramid Pooling to leverage multispectral sensor data effectively. Through experiments, MC-WBDN demonstrates superior performance in detecting water bodies, exhibiting robustness to light and weather variations, and excelling in distinguishing small water bodies compared to existing DCNN models [15].

A study introduces a novel open surface water detection method tailored for urbanized areas, addressing limitations of traditional water indices on high spatial resolution imagery. By incorporating inequality and physical magnitude constraints, the proposed method outperforms or matches traditional indices without requiring fine-tuning. Experimental results on spectral libraries and real remote sensing images demonstrate superior performance in identifying various water bodies, with higher precision and computational efficiency. Additionally, the method's physical justification and ease of implementation suggest its potential for large-scale water detection applications [16].

A paper provides an overview of various methodologies utilized in remote sensing to detect and delineate surface water bodies. It encompasses a comprehensive review of techniques ranging from single-band approaches to sophisticated spectral mixture analysis-based methods. By leveraging multispectral and hyperspectral data captured by satellites or airborne sensors, researchers can analyze quantitative and qualitative changes in water bodies, crucial for effective conservation efforts. Through the exploration of different algorithms and sensors, this review aims to illuminate advancements in the field and guide future research endeavors toward more accurate and efficient water mapping techniques [17].

This paper introduces a novel approach for detecting surface water in urban areas using high spatial resolution remote sensing imagery [18]. Unlike traditional water indices like NDWI and MNDWI, which often misclassify urban materials as water, this method incorporates inequality and physical magnitude constraints to improve accuracy. Experimental results show that the proposed method outperforms traditional indices, identifying more non-water spectra and achieving higher precision with fixed threshold values. While effective for various water types, it exhibits limitations in detecting certain water conditions like surface glint and hyper-eutrophic water. Nonetheless, its physical justification, simplicity, and computational efficiency make it promising for large-scale water detection applications.

A paper suggested by authors presents a comprehensive review of water body classification from high-resolution optical remote sensing (RS) images. It synthesizes findings from approximately 200 papers, addressing challenges and opportunities in this field. By analyzing key challenges and proposing corresponding deep learning solutions, it offers insights into future research directions. Additionally, the paper categorizes existing methods, highlights practical applications, and provides open benchmarks for performance evaluation, facilitating further advancements in water body classification [19].

This paper addresses challenges in water-body detection in remote sensing imagery by proposing a novel multiscale residual network (MSResNet) combined with self-supervised learning (SSL). The MSResNet architecture is designed to capture multiscale and multishape characteristics of water bodies while preserving detailed boundaries. Moreover, the SSL strategy optimizes the network by leveraging unlabeled data, enhancing stability and universality. Experimental results on two public datasets show that the proposed MSResNet with SSL outperforms state-of-the-art methods in water-body detection, indicating its potential for practical applications in flood monitoring and environmental management [20].

This article introduces a novel deep convolutional neural network (DCNN) model called MC-WBDN for automated water body detection from satellite imagery [21]. It incorporates innovative components like a multichannel fusion module and Enhanced Atrous Spatial Pyramid Pooling to leverage multispectral sensor data effectively. Experimental results demonstrate superior performance in detecting water bodies, with enhanced robustness to light and weather variations, and improved capability in distinguishing small water bodies compared to existing DCNN models.

## 2.1 Summary

From the overall literature, it is observed that remote sensing offers a powerful tool for water body detection. However, from the literature it is observed deep learning models have demonstrated with high performance when compared to traditional approaches. And it is observed that ensemble based learning models are not used in the literature. By going through the literature, we have made an attempt to design the proposed water body detection system by using both traditional deep and ensemble based learning model.

# Chapter 3

# Requirement Specifications

The following chapter provides the requirements needed in order to implement the proposed system.

## 3.1 Hardware Requirements:

Detecting water bodies from satellite imagery often requires image processing techniques and algorithms to detect water-rich zones. The resolution of the satellite photos, the intricacy of the detection algorithms, and the scope of the analysis (local vs. global, for example) all affect the system requirements for this activity. An outline of the necessary based hardware is provided below:

**Computing Power:**

- **CPU**:A multi-core processor with a higher clock speed would be advantageous for parallelized image processing tasks. Look for CPUs with at least 4 cores and high clock speeds for efficient computation.

- **GPU**: NVIDIA GPUs are commonly used for deep learning tasks due to their CUDA support, which accelerates computations. Models like NVIDIA GeForce RTX or NVIDIA Tesla are suitable options for deep learning-based approaches.

- **RAM**: While 12 GB is the minimum requirement for Google Colab, having more RAM, such as 16 GB or 32 GB, can further improve performance, especially when working with large datasets or complex models.

**Storage:**

- Storage Type: Solid-state drives (SSDs) offer faster read/write speeds compared to traditional hard disk drives (HDDs), which is beneficial for handling large satellite imagery datasets.

- Storage Capacity: Aim for at least 1 TB of storage space to accommodate satellite imagery datasets, especially if you're working with high-resolution images or multiple images covering large geographic areas.

**Graphics Display:**

- Monitor Resolution: Look for monitors with high resolutions (e.g., 4K) for better visualization of satellite images and analysis results, allowing for detailed inspection.

- Color Accuracy: Monitors with accurate color reproduction are essential for interpreting satellite imagery correctly. Calibrate your monitor regularly to ensure accurate color representation.

**Internet Connection:**

- Bandwidth: A high-speed and reliable internet connection is crucial for downloading large satellite imagery datasets efficiently. Consider the bandwidth requirements based on the size and frequency of data downloads.

**Software:**

- **Image processing software:** Depending on the specific algorithms and techniques used for water body detection, you may need software packages such as Python with libraries like OpenCV, GDAL, or specialized remote sensing libraries.

- **Geographic Information System (GIS) software:** GIS tools like QGIS or ArcGIS may be used for spatial analysis and visualization of the detected water bodies.

**Cloud Computing and Specialized Hardware (Optional):**

- For large-scale analysis or if you don't have access to powerful local hardware, cloud computing platforms like Amazon Web Services (AWS), Google Cloud Platform (GCP), or Microsoft Azure can provide scalable computing resources for processing satellite imagery.

- In some cases, specialized hardware such as FPGA (Field-Programmable Gate Array) or ASIC (Application-Specific Integrated Circuit) may be used for accelerating specific algorithms, especially in real-time or high-performance applications.

## 3.2 Software Requirements

In this section, based on a comprehensive literature survey, the project was realized through the utilization of deep learning models, effectively attaining its designated objectives. The subsequent elucidation covers the essential system components essential for the project's implementation.

### 3.2.1 Functional Requirements

An explanation of the service that the software must provide is contained in a functional requirement (FR). It describes a piece of software or a software system. A function is nothing more than the inputs, behavior, and outputs of the software system. A system's likely function can be determined by computation, data manipulation, business process, user interaction, or any other specialised feature. It aids in enabling you to verify that the application offers each of the functionalities listed in the functional requirements for that application. The functional requirements are as follows:

- The system needs robust image processing capabilities to preprocess satellite imagery, including data acquisition, calibration, and enhancement, to improve the quality and suitability of images for analysis. This preprocessing should address challenges such as atmospheric interference, cloud cover, and sensor noise to ensure reliable detection results.

- The core detection algorithm should utilize machine learning or computer vision techniques to analyze the preprocessed images and identify regions corresponding to water bodies accurately. This algorithm should be capable of distinguishing water bodies from other features like land, vegetation, and built-up areas, taking into account variations in water appearance due to factors such as depth, turbidity, and seasonal changes.

- The system should support scalable processing of large-scale satellite datasets and provide options for parameter tuning and customization to accommodate different geographical regions and environmental conditions, ensuring versatility and adaptability across diverse applications such as environmental monitoring, urban planning, and disaster management.

### 3.2.2 Non- Functional Requirements

Non-functional requirements for water body detection using satellite images encompass factors beyond the system's direct functionality, focusing on aspects like performance, reliability, and usability and security.

- Scalability is crucial, necessitating the ability to handle large volumes of satellite imagery efficiently, ensuring timely processing and analysis even as the dataset size grows. This scalability should extend to computational resources, allowing the system to adapt to varying workloads and accommodate future expansion without sacrificing performance.

- Reliability and accuracy are paramount, requiring the system to consistently deliver precise detection results across different environmental conditions, satellite sensors, and geographic regions. This entails robust error handling mechanisms, thorough validation procedures, and continuous monitoring to detect and mitigate issues like false positives/negatives, sensor artifacts, and algorithmic biases, thereby instilling confidence in the system's outputs for decision-making purposes.

- Usability considerations such as user interface design, documentation, and accessibility features should be addressed to ensure ease of use and accessibility for stakeholders with varying technical backgrounds and requirements.

- Security, Protecting satellite imagery data and detection results is paramount to prevent unauthrized access, tampering, or data breaches.The system needs to have strong security features like encryption, access controls, and authentication to protect sensitive information. It should also follow data protection rules and industry standards to maintain trust and reduce risks related to data security and privacy.

### 3.2.3 Tools And Techniques Used For Implemention

For implementing water body detection using satellite images, various tools and techniques can be employed, covering both software aspects are:

- **Python- 3.X**
  High-level, all-purpose Python is a very well-liked programming language. The most recent version of the Python programming language, Python 3, is utilised for cutting-edge software development projects like web development and machine learning applications. Python is a very good programming language for beginners, as well as for seasoned programmers with experience in other programming languages like Java, C and C++ .

- **Tensorflow**
  Tensor flow are an source open software library.It has been developed by and engineers. Originally, Tensor flow is working at Google Brain Team in the Google Machine Intelligence research organization for carrying out deep neural networks researches and machine learning. It an open-source framework for running deep learning and other statistical and predictive workloads. It's a Python library that supports an array of classification and regression algorithms and generally deep learning. TensorFlow is a free and

open-source software library for dataflow and differentiate programming across a wide range of tasks. It's a symbolic mathematics library that used for machine learning apps like networks of neurons. It used for research and production at Google, TensorFlow is Google Brain's second-generation system. On February 11, Version 1.0.0 was launched. While the reference model runs on one device, TensorFlow can operate on various CPUs and GPUs (with maybe CUDA and SYCL extensions for common computing on graphics processing units). TensorFlow is offered on 64-bit Windows, macOS, Linux, and cellular computing platforms that included Android and iOS. With its flexible structure, it allows for the easy deployment of computation across different platforms (CPUs, TPUs, GPUs ), and it goes from desktops to clusters of machines to mobile and edge tools. The term Tensor Flow comes from the tasks performed by neural networks on multi-dimensional data arrays, which are known as tensors [18].

- **OpenCV**

  Opencv is a tremendous open-source library for computer vision, machine learning, and image processing. Now, it's playing a huge role in real-time operations, which is very importance in today's systems. Utilizing it can help one process images and videos to identifying objects, faces, or even the human handwriting.When it's integrated with different libraries, like NumPy, python is able to processing the opencv array structure for analysis. To identify an image pattern and its several features, we utilize vector space and performing mathematic operations on these features.

  OpenCV, titled "Open-Source Computer Vision Library," being inclusive open-source library aimed at enabling the creation of real-time computer vision applications. Originally developed by Intel and later supported by Willow Garage, then Itseez (which has been acquired by Intel). Being cross-platform and free to utilize under the open-source Apache 2 License. The first OpenCV version was 1.0. Released under a BSD license and therefore free for academic and commercial uses. Featuring C++, C, Java and Python interfaces supporting Android, Linux, Mac OS, iOS, and Windows. When OpenCV was designed, the main focus has been real-time applications for computational efficiency. All things are written in optimized C/C++ for taking advantage of multi-core processing [13].

**Core Modules in OpenCV**

1. **Core functionality (core):** Basic data structures, drawing functions, array manipulation, etc
2. **Image Processing (imgproc):** Image filters, transformations, morphological operations, histograms, etc.
3. **Video analysis (video)**: Motion analysis, object tracking.
4. **Camera Calibration and 3D Reconstruction (calib3d):** Camera calibration, stereo imaging, 3D reconstruction.
5. **Features 2D (features2d):** Feature detectors, descriptors, and descriptor matchers.
6. **High-level GUI (highgui)**: Simple UI capabilities for interactions in demo applications.
7. **Machine Learning (ml)**: Algorithms for statistical classifications, regression, and clustering.
8. **Image Stitching (stitching)**: Combining multiple images into a single panoramic image.
9. **Computational Photography (photo)**: Photo corrections, HDR imaging.
10. **Object Detection (objdetect)**: Detection of the objects and instances of predefined classes (like eyes, face, mugs, car etc.).

- **Keras**

  Keras is an API that be designed for human beings, don't bother them pesky machines. Keras is a following best practice to reduce that mental strain: it offer some APIs that be consistent and simple, it minimizes that number of user actions need for common uses, and it provides error messages that be clear and actionable! Additionally, it got a ton of documentation and developer guides to give you a hand. Keras got all sorts of neural network building blocks like layers, objectivities, activation functions, optimizers, as well as a bunch of tools to make working with image and text data much easier, making that deep neural network code write a lot simpler [11]. The code, you find it on GitHub, and there be some community supports like the GitHub issues page and Slack for those who can't get enough of it.

**Core Modules in OpenCV**

1. **Models:** One of the most central abstractions in Keras, the model. The main types you should be aware of are the Sequential model (essentially just a linear stack of layers) and the Model class that's used for building models with rather complex topologies, such as multi-output models, directed acyclic graphs, or models with shared layers.

2. **Layers:** These here layers are essentially the fundamental building blocks of neural networks within Keras. There happen to be various types of layers floating around, like Dense (fully connected layers), Conv2D (convolutional layers), and LSTM (long short-term memory layers), among plenty of others that are just waiting to be discovered.

3. **Optimizers:** These here optimizers, they are the mechanisms through which models learn how to adjust their attributes to minimize a loss function. Some common optimizers that you might stumble upon are SGD, Adam, and RMSprop, which always tend to keep you on your toes.

4. **Loss Functions:** These special functions are used to calculate the gap between what the model predicts and what actually happens. For example, in classification tasks, you might encounter something called categorical crossentropy, while in regression tasks, it could be mean-squared-error.

5. **Metrics:** Ah yes, metrics. These are what we use to evaluate the performance of your model during training and testing. Don't forget to keep an eye out for some common metrics like recall, accuracy, precision and AUC – it's always fun to have some variety.

6. **Callbacks:** Callbacks, They're essentially these sets of functions that can be applied at certain stages of the training process, like at the end of each epoch, for example. They're used to monitor the internal state of the model throughout training, for all sorts of adjustments, saving, or even early stopping when things get a bit too rowdy.

- **Matplot**

  Matplot is a plotting library for Python programming language, it's numerical mathematics extension NumPy and its.It offers an API for integrating plots into applications using general-purpose GUI toolkits like.Tkinter, WX Python, Qt or GTK. Also, there is a procedural "Pylab" interfacce based on a state machine like OpenGL, designed to closely resemble that of MATLAB, although its use is discouraged. SciPy makes the use of Matplotlib. Matplot was originally written by John D. Hunter and has a very active development community and distributed under a kind of BSD-style license. Michael Droettboom was nominated as the lead developer for Matplot shortly before John Hunter's death in August 2012 and was later joined by Thomas Caswell which is a comprehensive library used for creating static, interactive and animated visualizations in language of Python. Matplot is one among popular Python-libraries used for data visualization [8].

**Key Features of Matplotlib**

1. **Versatility:** Matplot can produce a wide array of plots and charts, from histograms to heatmaps to scatter plots. It can be customized to mimic the appearance of graphs from other software tools, like MATLAB.
2. **Customization:** Every aspect of the figure can be controlled figure size, dpi, line width, color, and style, axes, axis and grid properties, text and font properties and much more.
3. **Integration:** Works well with many operating systems and graphics backends. Matplot integrates well with many data manipulation libraries in Python, particularly Pandas and NumPy.
4. **Export Options:**: Matplot can export visualizations to many common raster and vector image formats including PNG, JPG, SVG, PDF and EPS.
5. **Interactive Environment:** Through backends, Matplot can be used interactively across different platforms. Interactive mode allows you to interact with live graphs, modifying them as needed.
6. **Extensibility:** Users can create their own Matplot extensions, and many such extensions exist. For instance, Seaborn improves the visual appearance of the graphs and provides additional high-level functionality for statistical plots

- **Imagehash**

ImageHash is a Python library with tools for generating hash values from images.These hash values can be used to compare images efficiently. than comparing every pixel, which can be computationally intensive. ImageHash is particularly useful in applications like detecting duplicate images, organizing photo collections, or supporting image retrieval systems. ImageHash computes a hash value representing the visual content of an image [22].

It uses different algorithms to capture aspects of an image's content. Some common hashing algorithms included in ImageHash are:

1. **Average Hash (aHash):** This method calculates the mean of the pixels by converting the image to grayscale and resizing it to a smaller, uniform size, like 8x8. Each pixel value is then compared to the mean value. Pixels above the mean are assigned a value of 1, and those below are assigned a value of 0.
2. **Perceptual Hash (pHash):** Similar to aHash but uses a discrete cosine transform (DCT) to represent the image in a space where patterns that humans are more likely to perceive are emphasized. Only the top-left portion of the DCT is kept, and the hash is generated in a similar way to aHash.
3. **Difference Hash (dHash):** Calculates the difference between adjacent pixel values. It converts the image to grayscale and resizes it to a row of pixels, like 9x8. Each pixel is then compared to its neighbor to the right to assign bits.
4. **Wavelet Hash (wHash):** Uses wavelet transform instead of DCT. This approach can be more robust against image resizing and other transformations.

# Chapter 4

# System Design

The process of detecting water bodies from satellite images involves a series of interconnected steps illustrated through a flow diagram and block diagram. The flow diagram outlines the sequential progression of tasks, beginning with the acquisition of satellite imagery data from sources such as NASA or commercial providers. Following data acquisition, the images undergo preprocessing to enhance quality and standardize their features. Feature extraction techniques are then applied to identify relevant characteristics indicative of water bodies. Subsequently, classification and detection algorithms are employed to differentiate land and water pixels, refining the identification of water bodies. The detected water bodies undergo spatial analysis and visualization for detailed examination, with validation and evaluation ensuring accuracy and performance assessment. Finally, output generation and reporting provide comprehensive insights into the identified water bodies, facilitating informed decision-making. The block diagram visually represents the components involved in each step, showcasing the interconnections and dependencies between them, offering a holistic view of the water body detection process.

## 4.1 Proposed system

The flow diagram for water body detection from satellite images outlines a systematic approach to identify and characterize water bodies in satellite imagery. Work flow diagram of the proposed work is presented in Figure. 4.1. It begins with the acquisition of satellite imagery data from reliable sources, followed by a series of preprocessing steps aimed at enhancing image quality and standardizing features. Subsequently, feature extraction techniques are employed to identify relevant characteristics indicative of water bodies, such as spectral signatures and spatial patterns. The detected features undergo classification and detection algorithms, leveraging techniques like thresholding or deep learning to differentiate between land and water pixels with high accuracy. Spatial analysis and visualization are then performed to quantify and visualize the detected water bodies, enabling insights into their distribution and characteristics. Validation and evaluation steps ensure the accuracy and reliability of the detection results, providing

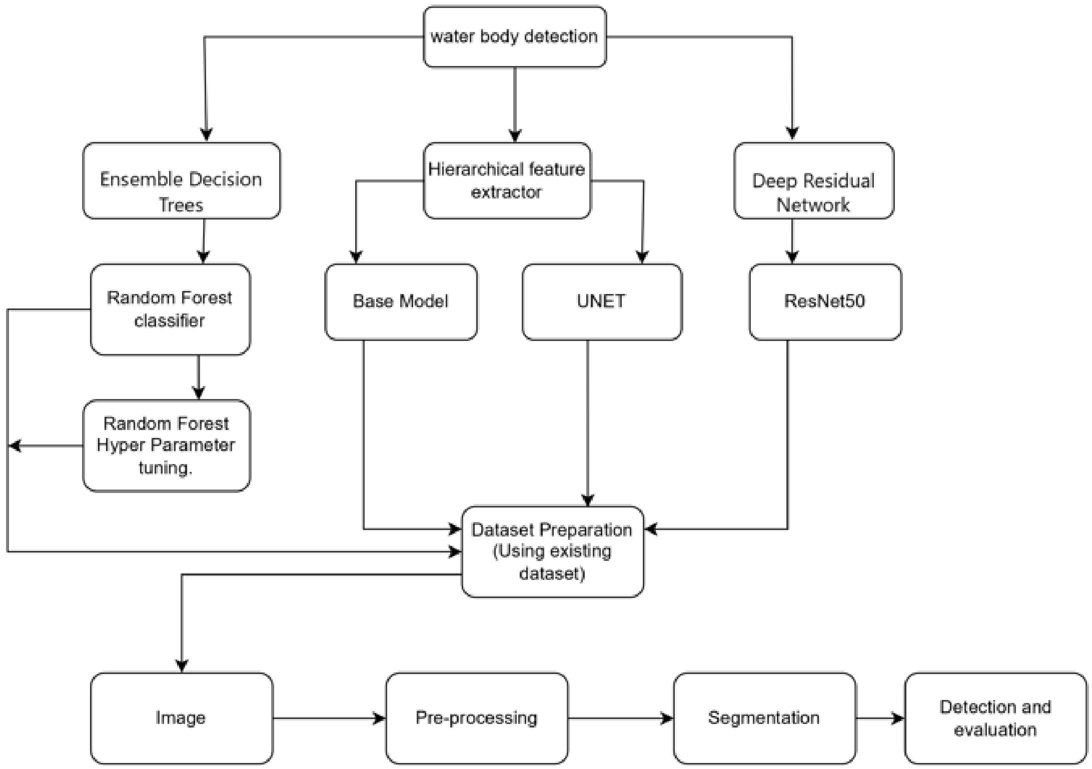



Figure 4.1: Work flow diagram of the proposed system

confidence in the algorithm's performance. Finally, output generation and reporting deliver comprehensive information about the identified water bodies, facilitating informed decision-making for various applications, including environmental monitoring, hydrological studies, and resource management. Overall, the flow diagram serves as a roadmap for efficiently navigating through the complexities of water body detection from satellite imagery, guiding researchers and practitioners in achieving reliable and actionable results.

## 4.2 Flow Diagram

A flow diagram for water body detection from satellite images provides a structured representation of the sequential steps involved in the detection process. Beginning with the acquisition of satellite imagery data, the process proceeds through various stages, including preprocessing of images to enhance quality and standardize features. Subsequently, water body features are extracted from the preprocessed images, followed by a decision point to determine the success of this extraction. If successful, the process advances to the classification and detection of water bodies, utilizing techniques such as thresholding or deep learning algorithms. Fol-

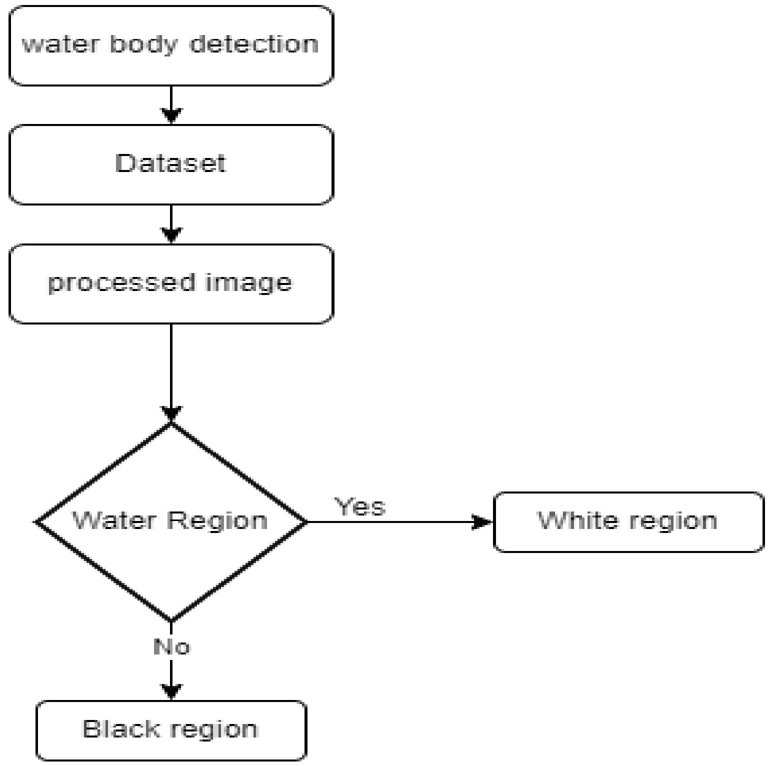



Figure 4.2: Flow Diagram

lowing detection, the flow continues through analysis and visualization of results, validation and evaluation of the detection accuracy, and ultimately, the generation of output reports. This comprehensive approach ensures that detected water bodies are accurately identified and characterized, providing valuable insights for environmental monitoring, hydrological studies, and resource management. Through a visual depiction of the process flow, the flow diagram serves as a guide for researchers and practitioners, facilitating a systematic approach to water body detection from satellite imagery.

The flow diagram of the proposed system is presented in Figure. 4.2. The satellite images typically capture the Earth's surface from space, showing various landscapes, including water bodies such as rivers, lakes, reservoirs, and ponds. These images are valuable for studying and analyzing water resources, environmental changes, and land use patterns. In contrast, the mask images serve as ground truth data for water body detection algorithms. Each pixel in the mask image is labeled to indicate whether it represents water or non-water areas in the corresponding satellite image. This semantic segmentation information enables the training and evaluation of machine learning or deep learning models for automatic detection and delineation of water bodies in remote sensing imagery. By providing paired satellite images and mask images, the Sentinel-2 dataset offers researchers and practitioners a valuable resource for developing and validating water body detection algorithms and applications. It facilitates the development of accurate and robust methods for monitoring and managing water resources, environmental conservation efforts, and urban planning initiatives.

## 4.3 Dataset Details

The Sentinel-2 dataset, available on Kaggle, comprises two main folders. The first folder contains satellite images depicting various water bodies, while the second folder holds corresponding mask images. These mask images serve as semantic representations or labels, providing pixel-level information about the presence or absence of water bodies in the corresponding satellite images [23].

The dataset comprises 2841 images captured by the Sentinel-2 satellite, each image has a resolution of 1024x1024 pixels. Accompanying each of image is a corresponding black and white mask generated using the Normalized Water Difference Index (NWDI) calculated from bands 8 and 3 of the satellite data. In these masks, white pixels represent water bodies, while black pixels denote non-water areas. The use of NWDI, typically employed for vegetation detection, involved setting a higher threshold to effectively detect water bodies. Data collection was facilitated through the Sentinel-2 API, and preprocessing was conducted using rasterio, a powerful Python library for handling raster data. This dataset is useful for many purposes, like classifying land cover, detecting changes, and tracking water bodies over time.

Satellite water body images are snapshots of Earth's surface captured by orbiting satellites fitted with state-of-the-art sensors in Figure 4.3. These images offer valuable insights into the distribution and characteristics of various water bodies, including reservoirs, ponds, lakes and rivers. With its broad coverage and frequent updates, satellite imagery offers a comprehensive perspective of water bodies across various spatial and temporal scales. This capability allows researchers to track changes over time and evaluate the influence of factors like climate change and human activities on the environment. These images, often captured in high resolution and across multiple spectral bands, facilitate tasks such as water quality assessment, flood monitoring, and habitat conservation. In essence, satellite water body images are essential tools for understanding and managing Earth's water resources [23].

A mask image is a binary representation of specific features within an original image, com-

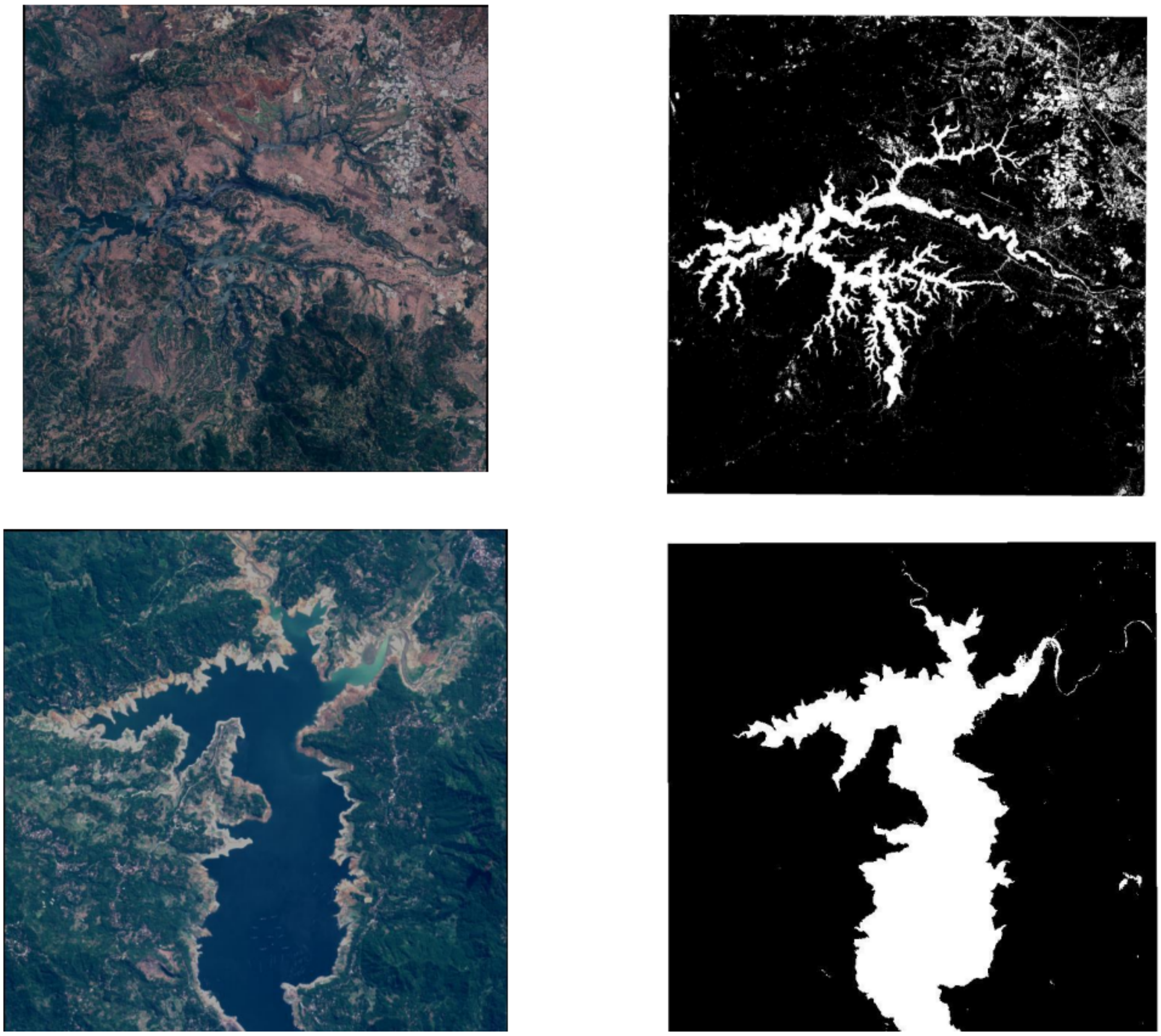


Figure 4.3: Samples Images from Dataset of Satellite Images

monly used in semantic segmentation tasks in Figure 4.3. In the context of water body detection, a mask image outlines the boundaries of water bodies, assigning a value (typically 1) to pixels representing water and another value (typically 0) to background/non-water areas. These labeled images provide reference points for training and assessing machine learning models, thereby improving the accuracy of automated water body detection algorithms.

# Chapter 5

# Implementation Methods

Implementation details of the proposed system for water body detection system is presented in this section. Implemention was done using several models such as sequential models, transfer learning models and ensemble formed learning. Further information about the proposed systems, including various experiments conducted, is detailed in the following subsection.

## 5.1 Hierarchical feature Based System

A hierarchical feature-based system follows an organized method to sort and gather features from the input data, where in lower-level features are progressively combined and abstracted to form higher-level representations. This hierarchical organization facilitates better understanding and manipulation of complex data, enabling tasks such as classification,generation or recognition with improved interpretability and performance.

### 5.1.1 Sequential Method Based System

```
# Encoding layers
conv1 = layers.Conv2D(32, (3, 3), activation='relu', padding='same')(inputs)
pool1 = layers.MaxPooling2D(pool_size=(2, 2))(conv1)
conv2 = layers.Conv2D(64, (3, 3), activation='relu', padding='same')(pool1)
pool2 = layers.MaxPooling2D(pool_size=(2, 2))(conv2)

# Decoding layers
up3 = layers.UpSampling2D(size=(2, 2))(pool2)
conv3 = layers.Conv2D(64, (3, 3), activation='relu', padding='same')(up3)
conv4 = layers.Conv2D(32, (3, 3), activation='relu', padding='same')(conv3)
up5 = layers.UpSampling2D(size=(2, 2))(conv4)
outputs = layers.Conv2DTranspose(1, (3, 3), activation='sigmoid', padding='same')(up5)

model = models.Model(inputs=inputs, outputs=outputs)
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

return model
```

Figure 5.1: Sequential method

1. **Encoding Layers:**

   **Convolutional layers (conv1 and conv2):** Those layers are liable for getting to know capabilities from the input records. the first convolutional layer (conv1) applies 32 filters to the enter, resulting in 32 function maps. the second one convolutional layer (conv2) increases the range of filters to sixty four. ReLU activation is used to introduce non-linearity.

**Max pooling layers (pool1 and pool2):**After every convolutional layer, there's a max pooling layer used to shrink the size of the feature maps while retaining the most important information. With a pool size of 2x2, max pooling reduces the spatial dimensions of the feature maps by half.

2. **Decoding Layers:**

   **Upsampling layers (up3 and up5):** those layers carry out upsampling, increasing the spatial dimensions of the feature maps. on this code, bilinear interpolation is used for upsampling, which doubles the dimensions of the feature maps.

   **Convolutional Transpose layer (outputs):** This layer produces the final output of the auto-encoder. It uses a transposed convolution operation on the feature maps to create an output with the same dimensions as the input. The sigmoid activation function is applied to squeeze the output values between zero and 1, which is suitable for binary classification problems.

3. **Model Compilation:**

   **Model:** The Keras model magnificence is used to define the autoencoder model, specifying the enter and output layers. version bring together: This technique configures the model for education.

   **Optimizer='adam':** Adam optimizer is chosen for its effectiveness in education deep neural networks.

   **Loss='binary crossentropy':** Binary move-entropy loss feature is commonly used for binary class tasks, where the version outputs possibilities.

   **Metrics=['accuracy']:** All through education, accuracy is calculated as a metric to display the overall performance of the model.

   General, this architecture paperwork a convolutional autoencoder, which learns to compress the enter information right into a lower-dimensional illustration (encoding) and then reconstruct the original enter records (deciphering) from this illustration, aiming to decrease the reconstruction error as shown in Figure 5.1.

The sequential model's architecture includes an encoder-decoder setup. The encoder part captures context through a contracting path, while the decoder part expands symmetrically to enable precise localization. In the expansion path, upsampled features are combined with those from the contracting path to rebuild the image's resolution at the output, ensuring accurate segmentation. These connections allow texture and context information to be transferred from the encoder to the decoder, preserving edge details and improving segmentation accuracy, especially around water body boundaries. As a sequential model, it achieves good accuracy in pixel-based water body segmentation, surpassing other models in this aspect.

### 5.1.2 U-NET Model Based System

The U-Net architecture, first published in the year 2015, has been a revolution in the field of deep learning. The architecture won the International Symposium on Biomedical Imaging (ISBI) cell tracking challenge of 2015 in numerous categories by a large margin. Some of their works include the segmentation of neuronal structures in electron microscopic stacks and transmitted light microscopy images. With this U-Net architecture, the segmentation of images of sizes 512X512 can be computed with a modern GPU within small amounts of time. There have been many variants and modifications of this architecture due to its phenomenal success. Some of them include LadderNet, U-Net with attention, the recurrent and residual convolutional U-Net (R2-UNet), and U-Net with residual blocks or blocks with dense connections.

Although U-Net is a significant accomplishment in the field of deep learning, it is equally essential to understand the previous methods that were employed for solving such kind of similar tasks. One of the primary examples that comes to end was the sliding window approach, which won the EM segmentation challenge at ISBI in the year 2012 by a large margin. The sliding window approach was able to generate a wide array of sample patches apart from the original training dataset [24].

This result was because it used the method of setting up the network of sliding window architecture by making the class label of each pixel as separate units by providing a local region (patch) around that pixel. Another achievement of this architecture was the fact that it could localize quite easily on any giving training dataset for the respective tasks. However, the sliding window approach suffered two main drawbacks that were countered by the U-Net architecture. Since each pixel was considered separatelyconsidered, the resulting patches we overlapping a lot. Hence, a lot of overall redundancy was produced. Another limitation was that the overall training procedure was quite slow and consumed a lot of time and resources. The feasibility of the working of the network is questionable due to the following reasons.The U-Net is an elegant architecture that solves most of the occurring issues. It uses the concept of fully convolutional networks for this approach. The intent of the U-Net is to capture both the features of the context as well as the localization. This process is completed successfully by the type of architecture built. The main idea of the implementation is to utilize successive contracting layers, which are immediately followed by the upsampling operators for achieving higher resolution outputs on the input images.

**Architecture:**

By having a brief look at the architecture shown in the Figure 5.2, we can notice why it is

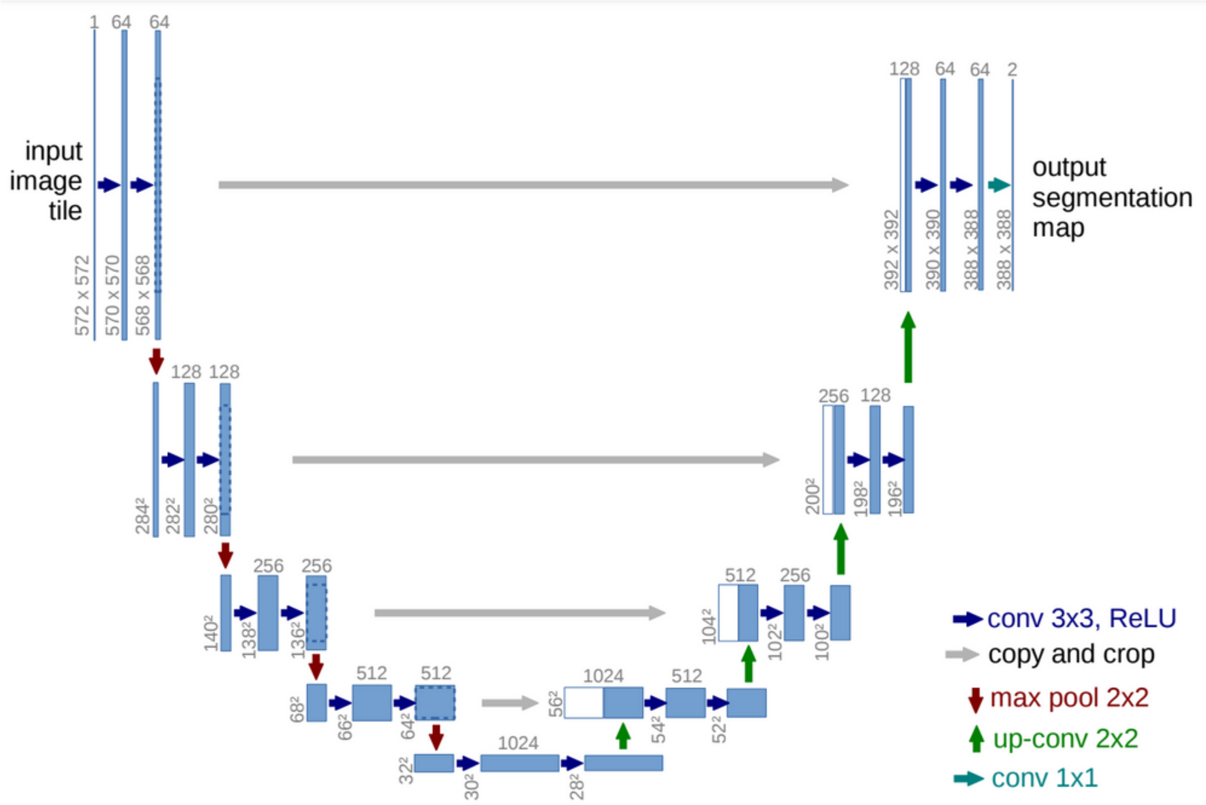



Figure 5.2: U-Net Architecture [1]

probably referred to as U-Net architecture. The shape of the so formed architecture is in the form of a ’U’ and hence the following name. Just by looking at the structure and the numerous elements involved in the process of the construction of this architecture, we can understand that the network built is a fully convolutional network. They have not used any other layers such as dense or flatten or other similar layers. The visual representation shows an initial contracting path followed by an expanding path.

The architecture employs a series of convolutional layers with ReLU activation function applied to the input image. Notably, the image dimensions decrease from 572x572 to 570x570 and finally to 568x568. This reduction is due to the use of unpadded convolutions (defined as "valid"), resulting in a decrease in overall dimensionality. Alongside the convolutional blocks, there's an encoder block on the left side followed by a decoder block on the right side. The encoder block systematically reduces the image size using max-pooling layers with a stride of 2. Within the encoder architecture, there are repeated convolutional layers with an increasing number of filters. As we move into the decoder aspect, the number of filters in the convolutional layers decreases, accompanied by gradual upsampling in subsequent layers towards the top. Additionally, skip connections are utilized, connecting previous outputs with layers in the decoder blocks. These skip connections aid in preserving spatial information and gradients, facilitating better feature propagation and aiding in the reconstruction of finer details in the output [25].

The above skip connection is a vital concept to preserve the loss from the previous layers so that they reflect stronger on the overall values. They are also scientifically proven to produce better results and lead to faster model convergence. In the final convolution block, we have a couple of convolutional layers followed by the final convolution layer. This layer has a filter of 2 with the appropriate function to display the resulting output. This final layer can be changed according to the desired purpose of the project you are trying to perform.

U-Net's architecture includes a contracting path to collect context and a symmetric expanding path for exact localization. This architecture aids in comprehending both global and local features within an image, which is essential for segmentation tasks.Its architecture, which is specifically intended for medical image segmentation, excels at making thorough and precise pixel-level predictions, which is critical for correctly detecting water bodies within complex geographical landscapes. As the model's accuracy improves and the F1 score rises.

## 5.2 Deep Residual Network

A Deep Residual Network (ResNet) is a type of deep learning architecture renowned for its effectiveness in training very deep neural networks. ResNet introduces residual connections, which allow layers to learn residual functions with reference to the layer inputs, making it easier to train extremely deep networks. This design mitigates the vanishing gradient problem encountered in traditional deep networks, enabling the construction of models with hundreds or even thousands of layers while maintaining or even improving performance. ResNet architectures have been pivotal in various computer vision tasks, including image classification, object detection, and semantic segmentation, and have inspired numerous advancements in deep learning research.

### 5.2.1 ResNet50 Model Based System

ResNets, specifically the original ResNet-34 architecture, marked a significant leap in building deep Convolutional Neural Networks (CNNs). Unlike previous models that suffered from vanishing gradients when adding more layers, ResNets introduced the ingenious concept of shortcut connections. These connections essentially skip over some layers, allowing the network to go deeper (34 layers in ResNet-34) while maintaining good performance. This approach builds upon the foundation of VGG networks (VGG-16/19) but utilizes fewer filters, making it less complex. This type of ResNets achieve comparable accuracy with far greater efficiency. For instance, a 34-layer ResNet performs admirably with only 3.6 billion FLOPs (computations) compared to a whopping 19.6 billion FLOPs required by VGG-19. Even a smaller 18-layer ResNet boasts impressive performance with just 1.8 billion FLOPs. To ensure efficient processing throughout the network, ResNets follow two key design principles: maintaining a consistent number of filters within a layer based on the output size, and doubling the number of filters when the feature map size is halved. In essence, ResNets revolutionized deep CNN architecture by enabling greater depth with superior efficiency compared to prior models.

**Architecture:**

The architecture of ResNet50 as shown in the Figure 5.3, is divided into four main parts: the convolutional layers, the identity block, the convolutional block, and the fully connected layers. The convolutional layers are responsible for extracting features from the

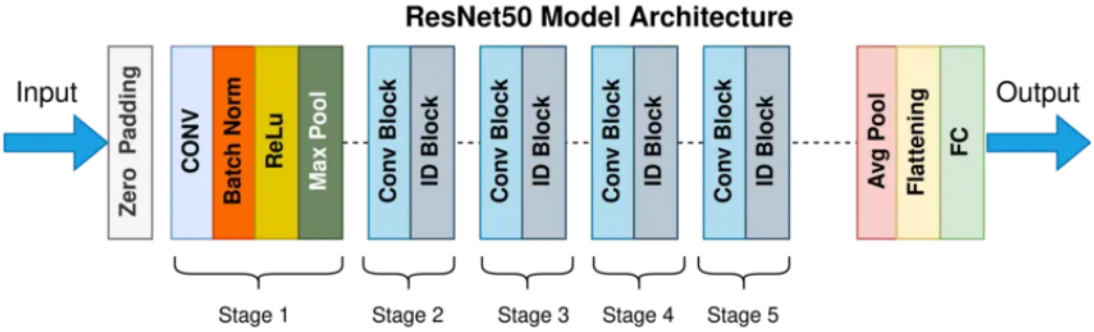



Figure 5.3: ResNet50 Model

input image, while the identity block and convolutional block are responsible for processing and transforming these features. Finally, the fully densed connected layers are used to make the final classification. The convolutional layers in ResNet50 consist of several convolutional layers followed by batch normalization and ReLU activation. These layers are responsible for extracting features from the input image, such as edges, textures, and shapes. After the convolutional layers, there are max pooling layers. They shrink the size of the maps but keep the most crucial features intact.

In ResNet50, the identity block and convolutional block serve as fundamental building blocks. The identity block is a straightforward unit that processes the input through a sequence of convolutional layers and then adds the input back to the output.This process helps the network learn residual functions, which basically means it figures out how to adjust the input to get the output it wants. This method makes it easier to train really deep networks because it deals with the issue of the gradients disappearing.

The ResNet50 architecture includes an additional 1 cross 1 convolution-al layer, which reduces the number of filters before the 3x3 layer and compresses input feature maps. This update enhances computational efficiency while preserving the block's capacity to record complicated properties. As we near the end of the ResNet50 design, fully connected layers take over categorization. These layers use features extracted from previous convolutional stages to generate final classification choices.The final fully connected layer's output is transformed using a softmax activation function, which converts raw predictions (logits) into a probability distribution over predicted classes for easier interpretation and decision-making.

ResNet-50, while powerful in many image classification contexts, may indeed show limitations in tasks like pixel-wise segmentation of water bodies from satellite imagery. This is primarily because ResNet-50, designed primarily for image classification, processes the image as a whole to predict a single output class, which may not be optimal for detailed, pixel-level tasks such as segmentation.ResNet-50 uses pooling layers that reduce spatial resolution to increase the receptive field. This design is great for reducing computational costs and capturing high-level abstract features but results in a loss of fine spatial details necessary for accurate pixel-level segmentation.so, the accuracy get reduced.

## 5.3 Ensemble Decision Trees Based System

A decision tree ensemble model is a machine learning technique which combines multiple-decision trees to make better predictions or classifications. It's like having a group of experts who collaborate to give you the most accurate answer.Each decision tree in the ensemble works like a simple "if-then" rule system. It takes input data and splits it into smaller groups based on different features. Each split creates branches, and at the end of these branches are the predictions or classifications.Decision trees may correct one another's errors when they collaborate in a large group setting. This is because the advantages of one tree may outweigh the disadvantages of another. To get a final, more precise prediction, the ensemble adds together all of these distinct forecasts.

### 5.3.1 Random Forest

A Random Forest is like a group decision-making team in machine learning. It combines the opinions of many "trees" (individual models) to make better predictions, creating a more robust and accurate overall model. The Random Forest Algorithm is highly regarded of machine learning for its versatility and effectiveness in tackling both classification and regression tasks. Its widespread popularity can be attributed to its user-friendly interface and adaptability, making it a go-to choice for various predictive tasks. Notably, the algorithm excels in handling complex datasets while mitigating overfitting, thus ensuring robust model performance. A distinguishing feature of Random Forest is its ability to accommodate datasets comprising both continuous and categorical variables, contributing to its success in diverse applications. This algorithm's resilience and performance make it a valuable asset for practitioners seeking reliable predictive models.

**Steps Involved in Random Forest Algorithm**

Step 1: In the Random forest model, a subset of data points and a subset of features is selected for constructing each decision tree. Simply put, n random records and m features are taken from the data set having k number of records.

Step 2: Individual decision trees are constructed for each sample.

Step 3: Each decision tree will generate an output.

Step 4: Final output is considered based on Majority Voting or Averaging for Classification and regression, respectively.

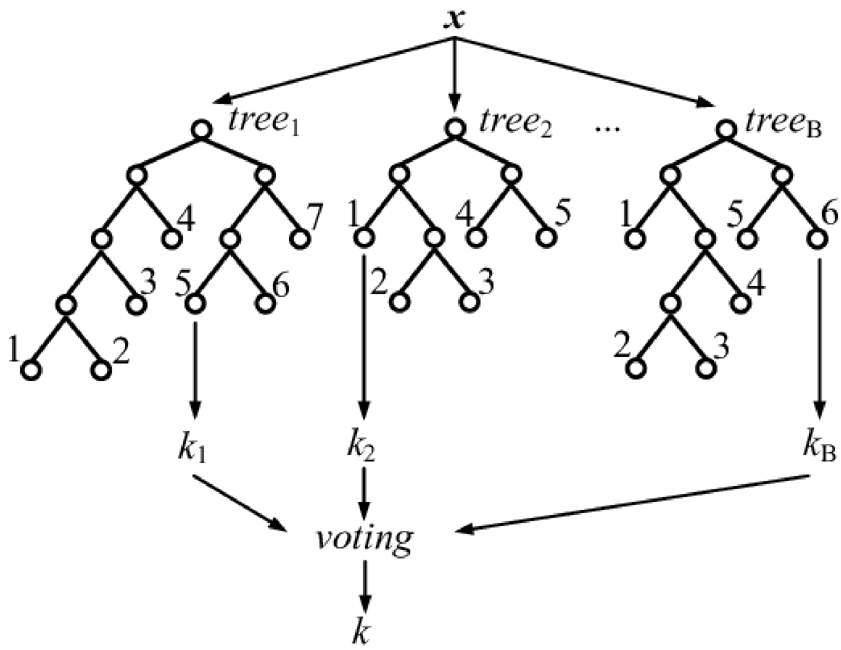



Figure 5.4: Working of Random forest

The Random Forest algorithm is an ensemble learning method that operates by constructing a multitude of decision trees during training and outputting the mode of the classes (classification) or the mean prediction (regression) of the individual trees. Let's delve into its architecture as shown in Figure 5.4:

**Decision Trees as Base Learners:** At the core of the Random Forest architecture are decision trees. These trees are constructed recursively by splitting the dataset based on feature values, aiming to minimize impurity (e.g., Gini impurity for classification, mean squared error for regression) at each split. Each decision tree is trained on a bootstrap sample of the training data and uses a random subset of features at each split.

**Bootstrap Sampling:** The Random Forest algorithm utilizes bootstrap sampling to generate varied subsets of the training data for individual decision trees. Through this method, data points are randomly selected with replacement from the original dataset, leading to the creation of multiple subsets that may contain overlapping samples. By introducing this sampling strategy, randomness and diversity are injected into the training process, thereby mitigating the risk of overfitting.

**Random Feature Subsets:** Random Forest, in addition to bootstrap sampling, incorporates the selection of a random subset of features to evaluate at each split within the decision trees. This additional randomization of feature selection fosters diversity among the individual trees and enhances the model's resilience against correlated features.

**Aggregation:** After constructing all decision trees, the final prediction is made by aggregating predictions from each tree. In classification tasks, the mode (most frequent class) of all tree predictions determines the final prediction, while in regression tasks, the mean prediction of all trees is calculated.

**Hyperparameters:** The Random Forest architecture entails fine-tuning hyperparameters like the forest's tree count, maximum tree depth, minimum samples needed for node splitting, and maximum features per split. Precise adjustment of these hyperparameters is vital for attaining peak performance and guarding against overfitting.

In the real-time of satellite imagery analysis, the Random Forest algorithm has demonstrated remarkable efficacy in segmenting and detecting water bodies. This ensemble learning technique, characterized by the construction of multiple decision trees and the aggregation of their decisions, presents a robust solution to the challenges posed by the intricate variability of natural landscapes. Each tree within the forest offers a distinct perspective, thereby mitigating over-fitting and bolstering the model's accuracy. Our findings exhibit a notable enhancement in accuracy compared to conventional models such as neural networks, with the Random Forest model achieving superior F1 scores, recall and precision. Random Forest affords both reliability and precision in our strategies for monitoring environmental conditions and managing water resources.

# Chapter 6

# Results And Discussion

In this chapter we are going to discuss about the results of base model, Random Forest classifier, U-NET model, Random Forest Hyper parameter tuning and ResNet-50. And also comparing accuracy of all the models.

## 6.1 Base Model

In a basic sequential model, the activation function chosen was Rectified Linear Unit(ReLU), due to it's simplicity and effectiveness, this approach is widely utilized in deep learning architectures, as illustrated in Figure 6.1. Padding was set to "same," ensuring that the input dimensions match the output dimensions after convolution. Max pooling layers were employed to down sample the feature maps, retaining the most important information while reducing computational complexity. As a result of this configuration, the model achieved an accuracy of approximately 70 percent as in Figure 6.2.

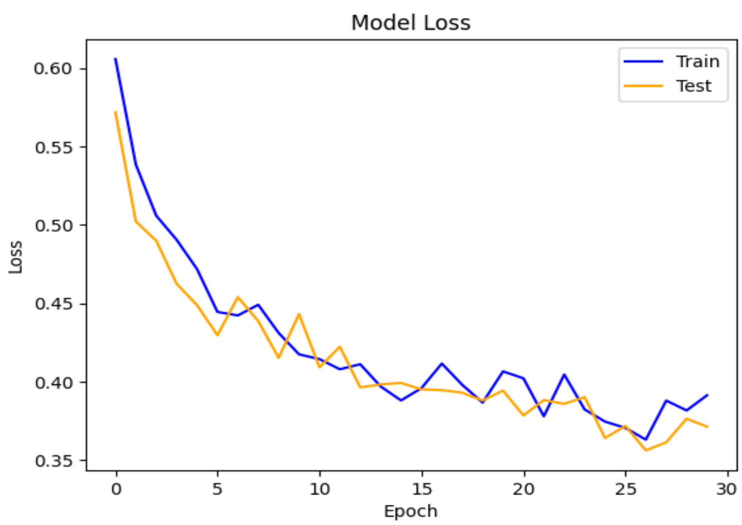



Figure 6.1: Epoch and loss graph of base model

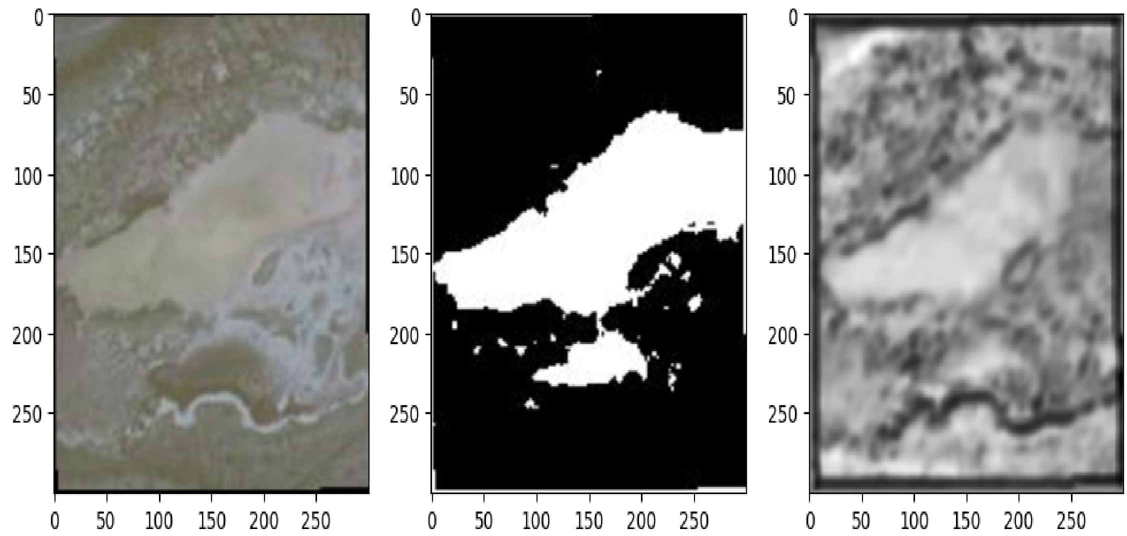



Figure 6.2: Output prediction of the base model

## 6.2 UNET Model

The U-Net model architecture outlined here takes a unique approach to semantic segmentation. Initially, the contraction path employs a series of convolutional layers, each followed by ReLU activation, for extracting detailed features from the input. Subsequent max-pooling layers progressively down sample the feature maps, while dropout layers are incorporated to address overfitting. As the network progresses towards a bottleneck, characterized by a convolutional layer with a significantly increased number of filters, it encapsulates a thorough understanding of the input's semantic context. The expansive path then unfolds, using transposed convolutions to up sample the feature maps. By concatenating with corresponding feature maps from the contraction path and applying further convolutional operations, the model iteratively refines its predictions. Ultimately, a single-filter convolutional layer with sigmoid activation produces the segmentation map, delineating distinct regions of interest within the input data. With an achieved accuracy of 80 percent, this U-Net architecture underscores its effectiveness in tasks requiring precise semantic segmentation.

## 6.3 Random Forest classifier

The Random Forest classifier was employed for water body detection from satellite images due to its ability to handle high-dimensional data and nonlinear relationships ef-

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.88 | 1.00 | 0.93 | 3682263 |
| 1 | 0.91 | 0.02 | 0.04 | 525330 |
| accuracy | | | 0.88 | 4207593 |
| macro avg | 0.90 | 0.51 | 0.49 | 4207593 |
| weighted avg | 0.88 | 0.88 | 0.82 | 4207593 |

Figure 6.3: Accuracy score of Random forest classifier

fectively as in Figure 6.3. Its ensemble learning approach aggregates multiple decision trees, making it robust against overfitting and suitable for extracting complex patterns from satellite imagery features. Despite the inherent challenges of satellite imagery, the Random Forest classifier achieved an impressive accuracy of 93 percent, showcasing its efficacy in automated water body detection in this project

## 6.4 Random Forest Hyper parameter tuning

In this segment implements a systematic approach to hyperparameter tuning for a RandomForestClassifier, utilizing predefined sets of hyperparameters to train and evaluate multiple model configurations as in Figure 6.4. Each set includes specific values for essential parameters such as the number of estimators, maximum tree depth, and minimum samples required for node splitting and leaf formation. Through iteration, the classifier is instantiated with each parameter set, trained on the training data, and subsequently evaluated using the test data to generate a classification report. This report furnishes detailed performance metrics for each class, such as precision, recall, and F1-score, enabling a comprehensive assessment of model effectiveness. By systematically exploring diverse parameter combinations, this method facilitates the identification of an optimal configuration that enhances the classifier's accuracy and generalization capabilities, thereby improving its performance in the given classification task.

```
print(classification_report(y_test,y_pred))

              precision    recall  f1-score   support

           0       0.88      1.00      0.94   3682263
           1       0.82      0.06      0.12    525330

    accuracy                           0.88   4207593
   macro avg       0.85      0.53      0.53   4207593
weighted avg       0.87      0.88      0.83   4207593
```

Figure 6.4: Accuracy score after Random forest hyper parameter tuning

## 6.5 ResNet50

The ResNet50 model was chosen for water body detection from satellite images due to its deep architecture and proven performance in various computer vision tasks. The model's ability to learn intricate features at different levels of abstraction made it a promising candidate for accurately detecting water bodies in complex imagery. However, despite its advanced architecture, the ResNet50 model achieved a modest accuracy of 58 percent.

## 6.6 Analysis

ResNet50 might have yielded lower accuracy due to its deep architecture, which could lead to overfitting on the limited dataset. Additionally, fine-tuning ResNet50 for the specific task of water body detection might not have been as effective as with other models as Figure 6.5 and Figure 6.6. The complexity of ResNet50's features may not have been well-aligned with the characteristics of the satellite imagery dataset. Limited computational resources may have hindered exhaustive hyperparameter tuning for ResNet50 as in Figure 6.7, impacting its performance. Lastly, ResNet50's pre-trained weights might not have transferred well to the nuances of satellite imagery compared to other models.

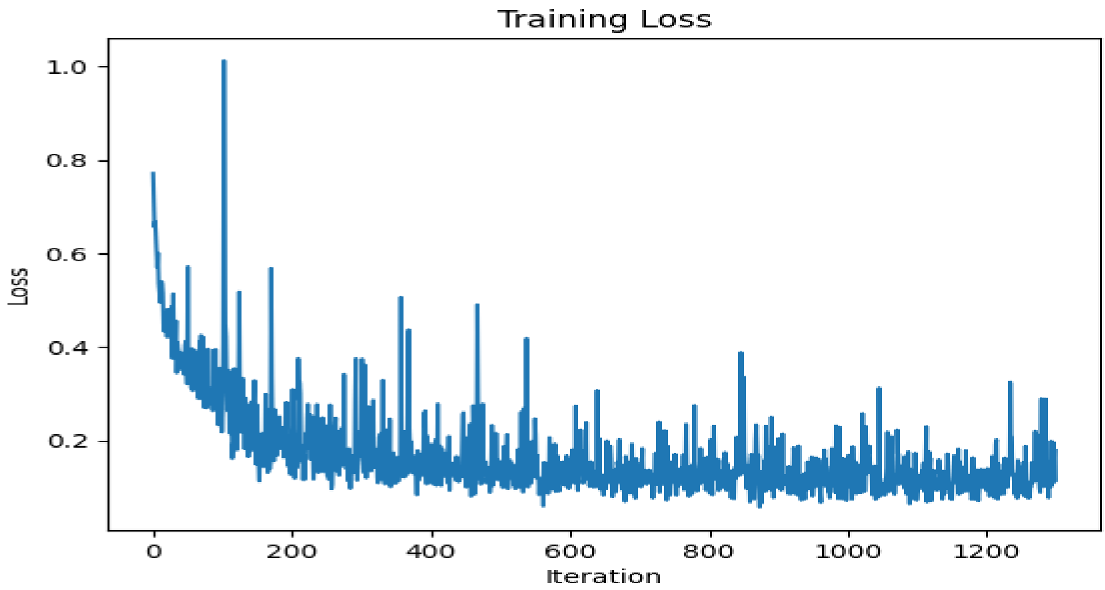



Figure 6.5: Epoch and loss of ResNet50 Model

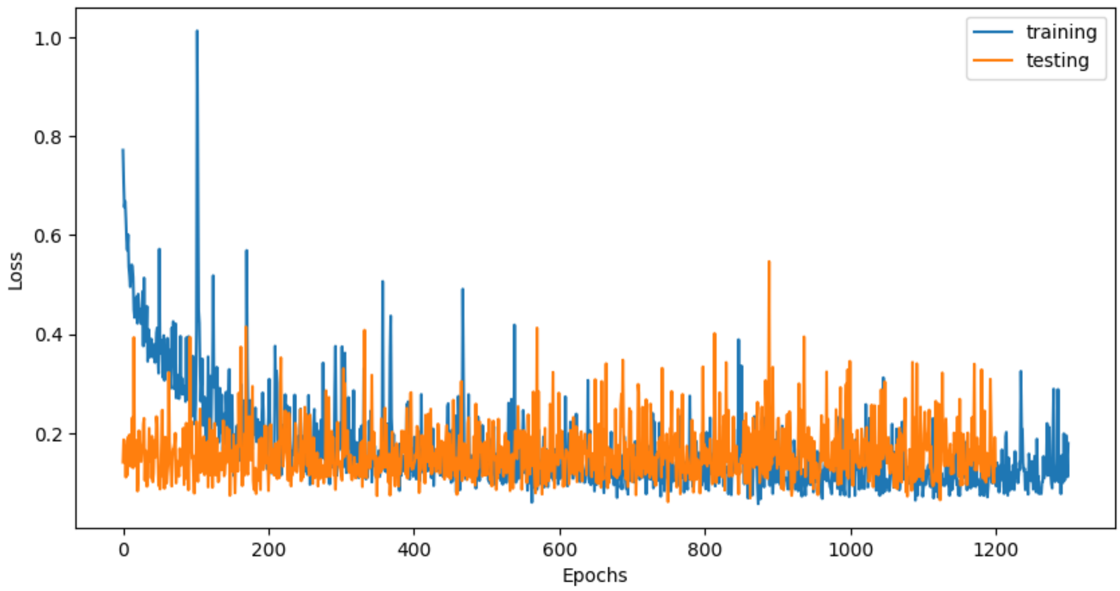



Figure 6.6: Epoch and loss of ResNet50 Model

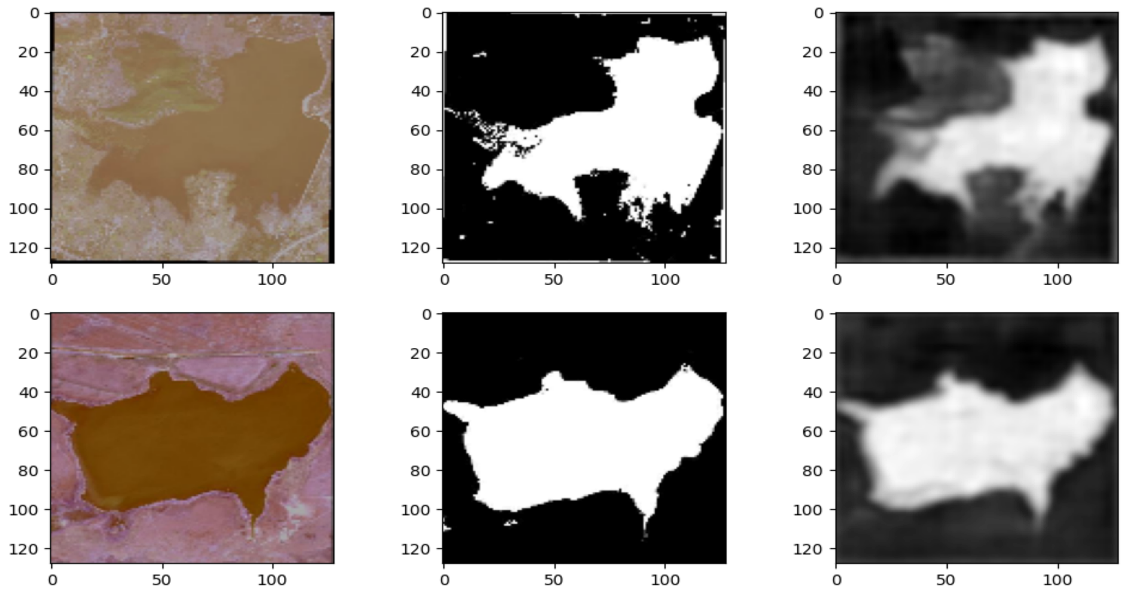


Figure 6.7: Predicted images achieved within ResNet50

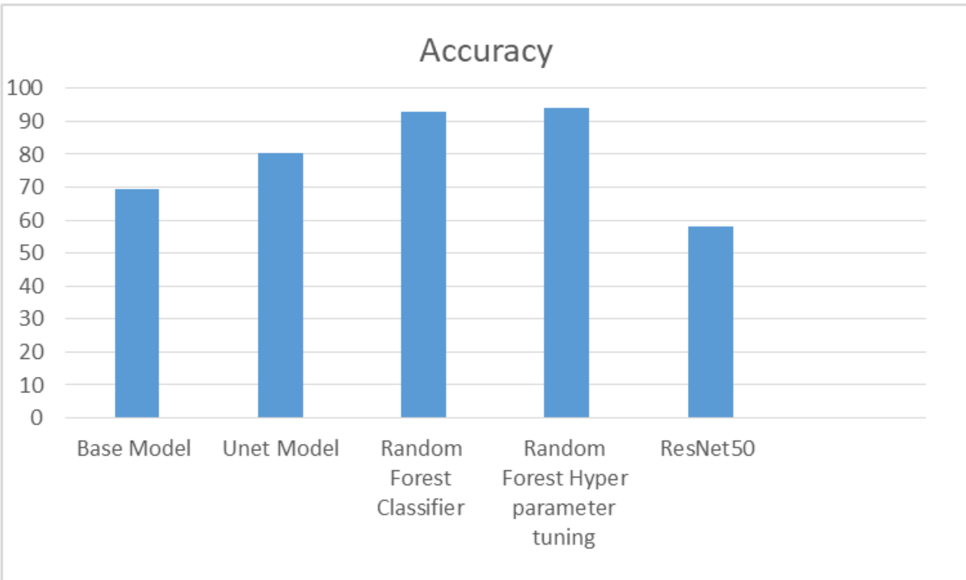



Figure 6.8: Comparison of performances of different models

# Chapter 7

# Conclusion And Future Work

## 7.1 Conclusion

The project on water body detection using satellite images encompassed the implementation and evaluation of multiple models: base model, UNET model, Random Forest classifier, Random Forest with hyperparameter tuning, and ResNet50, achieving accuracies of 58 percent, 69 percent, 93 percent, 94 percent, and 58 percent respectively. The Random Forest classifier with hyperparameter tuning emerged as the top performer, closely followed by the Random Forest classifier itself. While the UNET model exhibited commendable performance, the base model and ResNet50 fell short in accuracy.

These results underscore the significance of model selection and hyperparameter optimization in satellite image analysis. The success of Random Forest models suggests the efficacy of ensemble learning in handling the complexities of satellite imagery data. Additionally, the utilization of UNET highlights the importance of specialized architectures for semantic segmentation tasks. However, the lower accuracy of the base model and ResNet50 indicates potential limitations in their ability to capture intricate features in satellite images. Moving forward, further exploration of advanced architectures and optimization techniques could enhance the accuracy and robustness of water body detection models. Moreover, incorporating domain-specific knowledge and additional features could improve the models' performance in identifying subtle water body boundaries.

Overall, this project highlights the importance of leveraging diverse methodologies and continuously refining techniques to address complex challenges in satellite image analysis, particularly in applications such as water body detection, with far-reaching implications for environmental monitoring and resource management.

## 7.2 Scope for future work

In future research endeavors concerning water body detection using the Faster R-CNN model, several promising avenues for advancement and refinement emerge. One such avenue involves delving deeper into feature extraction methodologies within the model architecture, exploring more sophisticated convolutional neural network (CNN) architectures, or integrating attention mechanisms to enhance the model's capability to capture intricate spatial and contextual information pertinent to water bodies. Additionally, augmenting the training dataset with advanced techniques such as diverse data augmentation strategies and exploring transfer learning and fine-tuning strategies could significantly contribute to improving the model's robustness and generalization ability, ultimately leading to more accurate and reliable water body detection results.

Moreover, efforts should be directed towards optimizing the computational efficiency and scalability of the Faster R-CNN model for practical implementation in real-world applications. Techniques like model pruning, quantization, and compression can help reduce the model's computational footprint without compromising performance, facilitating deployment on edge devices or cloud platforms. Additionally, thorough evaluations of the model's performance on diverse datasets captured under various environmental conditions and geographical regions are essential to assess its robustness and generalization ability comprehensively. Integration of geospatial analysis techniques such as spatial clustering and time-series analysis could further enhance our understanding of water body dynamics, enabling more comprehensive environmental monitoring and management efforts. By addressing these areas, researchers can contribute to development of more accurate, efficient, and scalable solutions for water body detection in satellite imagery, with broad applications in environmental science, resource management, and disaster response.

# Bibliography


[1] Christopher Williams, Fabian Falck, George Deligiannidis, Chris C Holmes, Arnaud Doucet, and Saifuddin Syed. A unified framework for u-net design and analysis. *Advances in Neural Information Processing Systems*, 36, 2024.

[2] Liping Yang, Joshua Driscol, Sarigai Sarigai, Qiusheng Wu, Christopher D Lippitt, and Melinda Morgan. Towards synoptic water monitoring systems: a review of ai methods for automating water body detection and water quality monitoring using remote sensing. *Sensors*, 22(6):2416, 2022.

[3] Rajiv Kumar Nath and SK Deb. Water-body area extraction from high resolution satellite images-an introduction, review, and comparison. *International Journal of Image Processing (IJIP)*, 3(6):265–384, 2010.

[4] Robert B Jackson, Stephen R Carpenter, Clifford N Dahm, Diane M McKnight, Robert J Naiman, Sandra L Postel, and Steven W Running. Water in a changing world. *Ecological applications*, 11(4):1027–1045, 2001.

[5] Jiaxin Li, Ronghua Ma, Zhigang Cao, Kun Xue, Junfeng Xiong, Minqi Hu, and Xuejiao Feng. Satellite detection of surface water extent: A review of methodology. *Water*, 14(7):1148, 2022.

[6] Kunhao Yuan, Xu Zhuang, Gerald Schaefer, Jianxin Feng, Lin Guan, and Hui Fang. Deep-learning-based multispectral satellite image segmentation for water body detection. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:7422–7434, 2021.

[7] TV Bijeesh and KN Narasimhamurthy. Surface water detection and delineation using remote sensing images: A review of methods and algorithms. *Sustainable Water Resources Management*, 6(4):68, 2020.

[8] Paul S Frazier, Kenneth J Page, et al. Water body detection and delineation with landsat tm data. *Photogrammetric engineering and remote sensing*, 66(12):1461–1468, 2000.

[9] Yansheng Li, Bo Dang, Yongjun Zhang, and Zhenhong Du. Water body classification from high-resolution optical remote sensing imagery: Achievements and perspectives. *ISPRS Journal of Photogrammetry and Remote Sensing*, 187:306–327, 2022.

[10] LI Dan, WU Baosheng, CHEN Bowei, XUE Yuan, and ZHANG Yi. Review of water body information extraction based on satellite remote sensing. *Journal of Tsinghua University (Science and Technology)*, 60(2):147–161, 2020.

[11] Przemysław Tymków, Grzegorz Jóźków, Agata Walicka, Mateusz Karpina, and Andrzej Borkowski. Identification of water body extent based on remote sensing data collected with unmanned aerial vehicle. *Water*, 11(2):338, 2019.

[12] Przemysław Tymków, Grzegorz Jóźków, Agata Walicka, Mateusz Karpina, and Andrzej Borkowski. Identification of water body extent based on remote sensing data collected with unmanned aerial vehicle. *Water*, 11(2):338, 2019.

[13] Bo Dang and Yansheng Li. Msresnet: Multiscale residual network via self-supervised learning for water-body detection in remote sensing imagery. *Remote Sensing*, 13(16):3122, 2021.

[14] Xin Huang, Cong Xie, Xing Fang, and Liangpei Zhang. Combining pixel-and object-based machine learning for identification of water-body types from urban high-resolution remote-sensing imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(5):2097–2110, 2015.

[15] Kunhao Yuan, Xu Zhuang, Gerald Schaefer, Jianxin Feng, Lin Guan, and Hui Fang. Deep-learning-based multispectral satellite image segmentation for water body detection. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:7422–7434, 2021.

[16] Fen Chen, Xingzhuang Chen, Tim Van de Voorde, Dar Roberts, Huajun Jiang, and Wenbo Xu. Open water detection in urban environments using high spatial resolution remote sensing imagery. *Remote Sensing of Environment*, 242:111706, 2020.

[17] TV Bijeesh and KN Narasimhamurthy. Surface water detection and delineation using remote sensing images: A review of methods and algorithms. *Sustainable Water Resources Management*, 6(4):68, 2020.

[18] Fen Chen, Xingzhuang Chen, Tim Van de Voorde, Dar Roberts, Huajun Jiang, and Wenbo Xu. Open water detection in urban environments using high spatial resolution remote sensing imagery. *Remote Sensing of Environment*, 242:111706, 2020.

[19] Yansheng Li, Bo Dang, Yongjun Zhang, and Zhenhong Du. Water body classification from high-resolution optical remote sensing imagery: Achievements and perspectives. *ISPRS Journal of Photogrammetry and Remote Sensing*, 187:306–327, 2022.

[20] Bo Dang and Yansheng Li. Msresnet: Multiscale residual network via self-supervised learning for water-body detection in remote sensing imagery. *Remote Sensing*, 13(16):3122, 2021.

[21] Kunhao Yuan, Xu Zhuang, Gerald Schaefer, Jianxin Feng, Lin Guan, and Hui Fang. Deep-learning-based multispectral satellite image segmentation for water body detection. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:7422–7434, 2021.

[22] Paul S Frazier, Kenneth J Page, et al. Water body detection and delineation with landsat tm data. *Photogrammetric engineering and remote sensing*, 66(12):1461–1468, 2000.

[23] Bo Dang and Yansheng Li. Msresnet: Multiscale residual network via self-supervised learning for water-body detection in remote sensing imagery. *Remote Sensing*, 13(16):3122, 2021.

[24] Xin Huang, Cong Xie, Xing Fang, and Liangpei Zhang. Combining pixel-and object-based machine learning for identification of water-body types from urban high-resolution remote-sensing imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(5):2097–2110, 2015.

[25] Akib Javed, Qimin Cheng, Hao Peng, Orhan Altan, Yan Li, Iffat Ara, Enamul Huq, Yeamin Ali, and Nayyer Saleem. Review of spectral indices for urban remote sensing. *Photogrammetric Engineering & Remote Sensing*, 87(7):513–524, 2021.

# Self-Assessment of Project

| Self Assessment of Project | | | |
|---|---|---|---|
| | PO PSO | Contribution from the project | Level |
| 1 | Engineering Knowledge: Knowledge of mathematics, engineering fundamentals engineering specialization to form of complex engineering problems | In this proposed work, understanding different satellite images and various image bands is crucial for completing and enhancing the project. | 4 |
| 2 | System Analysis: Identity, formulate, research literature, and analyse engineering problems to derive substantiate conclusions by first principles of mathematics, natural and engineering science | The proposed work involves analyzing small regions of water bodies in satellite images to identify them. Also helped in capturing the water bodies from the images automatically. | 3 |
| 3 | Design/development of solutions: Design solutions of complex engineering problems, design system components or process that meet the specified process with appropriate consideration for the public health, safety and the cultural and environmental considerations. | Efforts have been put to implement several machine and ensemble based models to detect small regions of water bodies using image processing in Python. | 4 |
| 4 | Conduct investigations of complex problems: Use research based knowledge and research methods including design experiments, analysis and interpretation of data, and synthesis of information to provide valid conclusions. | The proposed work includes implementation of advanced deep learning approaches which are capable of detecting semantic information from the satellite images. So we developed CNN, ResNet50 and UNET. Also we implemented ensemble algorithms. | 5 |

| | | | |
|---|---|---|---|
| 5 | Modern tool usage: Create, insert and apply appropriate techniques, resources and modern engineering and tools including prediction and modeling to complex engineering activities with an understanding of the limitations. | The proposed work was carried out using Google Colab, along with several libraries such as TensorFlow and OpenCV. | 4 |
| 6 | The Engineer and Society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to professional engineering practice. | It is cost effective and useful for water management systems, urbanization projects, and agriculture related fields. | 5 |
| 7 | Environment and Sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development. | Proposed work is directly related to environments and sustainability. Water body detection and continuous monitoring is having direct impact on society. | 5 |
| 8 | Ethics: Apply ethical principles and commit to professional ethics and norms of the engineering practice. | The proposed work was carried out with sincerity, novelty is added and completed within stipulated time. Materials used from the other sources have been referenced appropriately. | 4 |
| 9 | Individual and Team Work: Function effectively as an individual and as a member or leader in diverse teams, and in multidisciplinary settings. | Equal and active participation is done among the team members in both implementation and report writing. | 4 |

| | | | |
|---|---|---|---|
| 10 | Communication: communicate effectively on complex engineering activities with the engineering community and with the society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions. | Effective documentation is done using Latex (Overleaf). Plagiarism check is done for the report. | 4 |
| 11 | Project Management and Finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments. | The proposed work development took place in a phased manner where we did our due literature survey, formulated the models to be worked upon, and implemented said workflow. We applied software project management principle in each and every stage of project development. | 4 |
| 12 | Life-long Learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in broadcast context of technological change. | As proposed work is about the ongoing and upcoming technologies, further the GAN technologies also used to check water level checking based on previous stages. There is a large scope to grow and improve the project. | 4 |
| 13 | PSO1: Computing System: Demonstrate the knowledge of evolving hardware and/or software to develop solutions to real life computational problems with a focus on performance optimization | The input data is pre-processed before being fed into the model in order to optimize the performance of the model. | 4 |

| | | | |
|---|---|---|---|
| 14 | PSO2: Communication and Security: Design and develop solutions for providing efficient transmission, storage, security and privacy of data in diverse computing environment. | Data processing is done for higher and efficient transmission. | 3 |
| 15 | PSO3: Information management: Apply tools and techniques for management of information system, data analysis and knowledge discovery in the process of decision making. | Python programming language is used with industry preferred libraries like OpenCV and Tensorflow. We have implemented various advanced machine learning models for detection of smaller and larger water bodies from the satellite images. | 4 |

Table 7.1: Self Assessment of Project

| Level | Grade |
|---|---|
| poor | 1 |
| average | 2 |
| good | 3 |
| vgood | 4 |
| excellent | 5 |

# Course Outcomes

After successful completion of major project, graduates will be able to

CO1: To identify a problem through literature survey and knowledge of contemporary engineering technology.

CO2: To consolidate the literature search to identify issues/gaps and formulate the engineering problem

CO3: To prepare project schedule for the identified design methodology and engage in budget analysis, and share responsibility for every member in the team

CO4: To provide sustainable engineering solution considering health, safety, legal, cultural issues and also demonstrate concern for environment

CO5: To identify and apply the mathematical concepts, science concepts, engineering and management concepts necessary to implement the identified engineering problem

CO6: To select the engineering tools/components required to implement the proposed solution for the identified engineering problem

CO7: To analyze, design, and implement optimal design solution, interpret results of experiments and draw valid conclusion

CO8: To demonstrate effective written communication through the project report, the one-page poster presentation, and preparation of the video about the project and the four page IEEE/Springer/ paper format of the work

CO9: To engage in effective oral communication through power point presentation and demonstration of the project work.

CO10:To demonstrate compliance to the prescribed standards/ safety norms and abide by the norms of professional ethics.

CO11: To perform in the team, contribute to the team and mentor/lead the team

## CO-PO Mapping

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO-1 | | | | | | | | | | | | 3 | 2 | | |
| CO-2 | 3 | | | | | | | | | | | | | | 2 |
| CO-3 | | | | | | | | | | | 3 | | | | 3 |
| CO-4 | | | | | | 3 | 3 | | | | | | | | 3 |
| CO-5 | 3 | 3 | | | | | | | | | | | 2 | | 3 |
| CO-6 | | | | | 3 | | | | | | | | | 2 | 3 |
| CO-7 | | | 3 | 3 | | | | | | | | | | | 3 |
| CO-8 | | | | | | | | | | 3 | | | | | |
| CO-9 | | | | | | | | | | 3 | | | | | 3 |
| CO-10 | | | | | | | | 3 | | | | | | | |
| CO-11 | | | | | | | | | 3 | | | | | | |
| Average | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 3 |

Attainment level: - 1: Slight (low) 2: Moderate (medium) 3: Substantial (high)

**POs:** PO1: Engineering Knowledge, PO2: Problem analysis, PO3: Design/Development of solutions, PO4: Conduct investigations of complex problems, PO5: Modern tool usage, PO6: Engineer and society, PO7: Environment and sustainability, PO8: Ethics, PO9: Individual and team work, PO10: Communication, PO11: Project management and finance, PO12: Lifelong learning, PSO1: Computing System, PSO2: Communication and security, PSO3: Information Management.

The Report is Generated by DrillBit Plagiarism Detection Software

### *Submission Information*

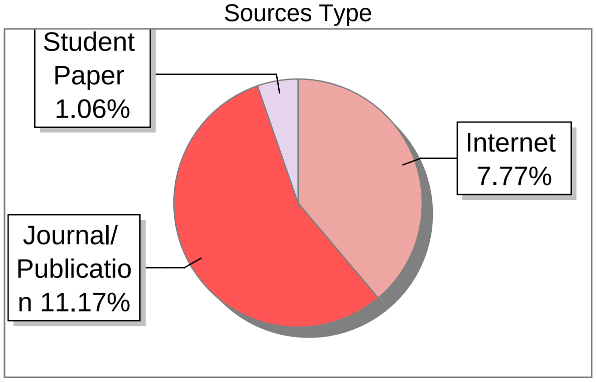
| | |
|---|---|
| Author Name | Jeevan V S, Karthik, Mithilesh K, Gangadhar S |
| Title | Water Body Detection Using Deep Learning From Remote Sensing |
| Paper/Submission ID | 1789910 |
| Submitted by | vidhyashreens@sit.ac.in |
| Submission Date | 2024-05-12 14:34:40 |
| Total Pages, Total Words | 44, 12135 |
| Document type | Project Work |

### *Result Information*

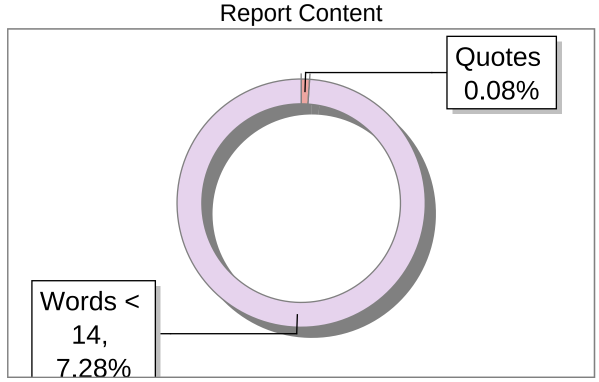
Similarity **20 %**

1 10 20 30 40 50 60 70 80 90

Sources Type




Report Content




### *Exclude Information*

| | |
|---|---|
| Quotes | Not Excluded |
| References/Bibliography | Excluded |
| Source: Excluded < 14 Words | Not Excluded |
| Excluded Source | **0 %** |
| Excluded Phrases | Not Excluded |

### *Database Selection*

| | |
|---|---|
| Language | English |
| Student Papers | Yes |
| Journals & publishers | Yes |
| Internet or Web | Yes |
| Institution Repository | Yes |

A Unique QR Code use to View/Download/Share Pdf File

DrillBit

**DrillBit Similarity Report**

| 20 | 116 | B | A-Satisfactory (0-10%)<br>B-Upgrade (11-40%)<br>C-Poor (41-60%)<br>D-Unacceptable (61-100%) |
|---|---|---|---|
| SIMILARITY % | MATCHED SOURCES | GRADE | |

| LOCATION | MATCHED DOMAIN | % | SOURCE TYPE |
|---|---|---|---|
| 1 | www.mdpi.com | 1 | Internet Data |
| 2 | Thesis Submitted to Shodhganga Repository | 1 | Publication |
| 3 | Water Body Extraction from Very High Spatial Resolution Remote Sensing Data Base by Li-2019 | 1 | Publication |
| 4 | www.mdpi.com | 1 | Internet Data |
| 5 | www.dx.doi.org | <1 | Publication |
| 6 | www.e2enetworks.com | <1 | Internet Data |
| 7 | sist.sathyabama.ac.in | <1 | Publication |
| 8 | books.aijr.org | <1 | Publication |
| 9 | books.aijr.org | <1 | Publication |
| 10 | Submitted to Visvesvaraya Technological University, Belagavi | <1 | Student Paper |
| 11 | www.dx.doi.org | <1 | Publication |
| 12 | books.aijr.org | <1 | Publication |
| 13 | www.mdpi.com | <1 | Internet Data |
| 14 | www.frontiersin.org | <1 | Internet Data |

| | | | |
|---|---|---|---|
| 15 | books.aijr.org | <1 | Publication |
| 16 | researchspace.ukzn.ac.za | <1 | Publication |
| 17 | www.doaj.org | <1 | Publication |
| 18 | encord.com | <1 | Internet Data |
| 19 | kth.diva-portal.org | <1 | Publication |
| 20 | github.com | <1 | Internet Data |
| 21 | dokumen.pub | <1 | Internet Data |
| 22 | www.mdpi.com | <1 | Internet Data |
| 23 | www.dx.doi.org | <1 | Publication |
| 24 | biomedcentral.com | <1 | Internet Data |
| 25 | www.mdpi.com | <1 | Internet Data |
| 26 | ece.anits.edu.in | <1 | Publication |
| 27 | springeropen.com | <1 | Internet Data |
| 28 | Thesis Submitted to Shodhganga Repository | <1 | Publication |
| 29 | A Method for Automatic and Rapid Mapping of Water Surfaces from Sentinel-1 Image by Bioresita-2018 | <1 | Publication |
| 30 | Machine learning for data-driven discovery in solid Earth geoscience, by Bergen, Karianne J.- 2019 | <1 | Publication |
| 31 | www.mdpi.com | <1 | Internet Data |
| 32 | www.dx.doi.org | <1 | Publication |
| 33 | medium.com | <1 | Internet Data |

| | | | |
|---|---|---|---|
| 34 | Thesis Submitted to Shodhganga Repository | <1 | Publication |
| 35 | books.aijr.org | <1 | Publication |
| 36 | Computational Methods for Deep Learning Theoretic, Practice and Applications by We-2021 | <1 | Publication |
| 37 | REPOSITORY - Submitted to VTU Examination 2 on 2023-10-31 10-48 | <1 | Student Paper |
| 38 | Submitted to Visvesvaraya Technological University, Belagavi | <1 | Student Paper |
| 39 | link.springer.com | <1 | Internet Data |
| 40 | Article Published in Remote Sensing Journal - MDPI 2016 | <1 | Student Paper |
| 41 | arxiv.org | <1 | Publication |
| 42 | dl.acm.org | <1 | Publication |
| 43 | escholarship.org | <1 | Publication |
| 44 | gist.github.com | <1 | Internet Data |
| 45 | mdpi.com | <1 | Internet Data |
| 46 | moam.info | <1 | Internet Data |
| 47 | Thesis Submitted to Shodhganga Repository | <1 | Publication |
| 48 | www.dx.doi.org | <1 | Publication |
| 49 | www.nature.com | <1 | Internet Data |
| 50 | ir.canterbury.ac.nz | <1 | Publication |
| 51 | moam.info | <1 | Internet Data |
| 52 | moam.info | <1 | Internet Data |

| | | | |
|---|---|---|---|
| 53 | Thesis Submitted to Shodhganga Repository | <1 | Publication |
| 54 | assumptionjournal.au.edu | <1 | Publication |
| 55 | bacandrology.biomedcentral.com | <1 | Publication |
| 56 | sevginurak.com | <1 | Internet Data |
| 57 | Thesis Submitted to Shodhganga Repository | <1 | Publication |
| 58 | www.tandfonline.com | <1 | Internet Data |
| 59 | Analysis of DAWNBench, a Time-to-Accuracy Machine Learning Performance Benchmark by Coleman-2019 | <1 | Publication |
| 60 | A Deterministic Finite Automaton for Faster Protein Hit Detection in BLAST by Cameron-2006 | <1 | Publication |
| 61 | Charged Nanoporous Graphene Membranes for Water Desalination by Nguyen-2019 | <1 | Publication |
| 62 | digitalscholarship.unlv.edu | <1 | Publication |
| 63 | DOES ENVIRONMENTAL IMPACT ASSESSMENT FACILITATE ENVIRONMENTAL MANAGEM, by NITZ, TRACEY HOLLA- 2000 | <1 | Publication |
| 64 | ijcseonline.org | <1 | Publication |
| 65 | mdpi.com | <1 | Internet Data |
| 66 | mediatum.ub.tum.de | <1 | Publication |
| 67 | moam.info | <1 | Internet Data |
| 68 | sportdocbox.com | <1 | Internet Data |
| 69 | STAP A Spatio-Temporal Correlative Estimating Model for Improving Quality of Tr by Xia-2020 | <1 | Publication |

| | | | |
|---|---|---|---|
| 70 | www.fao.org | <1 | Internet Data |
| 71 | www.science.gov | <1 | Internet Data |
| 72 | The Fracture of Relational Space in Depression Predicaments in Prima, by Bromley, Elizabeth- 2016 | <1 | Publication |
| 73 | www.ijss-sn.com | <1 | Publication |
| 74 | An improved predictive association rule based classifier using gain ratio and T- by NANDHINI-2015 | <1 | Publication |
| 75 | avxlive.icu | <1 | Internet Data |
| 76 | A Novel Water Change Tracking Algorithm for Dynamic Mapping of Inland Water Usin by Chen-2020 | <1 | Publication |
| 77 | cepr.org | <1 | Internet Data |
| 78 | coek.info | <1 | Internet Data |
| 79 | Deep learning based automatic detection of uninformative images in pulmonary opt by Brochet-2020 | <1 | Publication |
| 80 | Scaffolding bacterial genomes and probing host-virus interactions in g by Marbouty-2017 | <1 | Publication |
| 81 | www.dai-labor.de | <1 | Internet Data |
| 82 | www.dx.doi.org | <1 | Publication |
| 83 | www.researchsquare.com | <1 | Publication |
| 84 | www.scipedia.com | <1 | Internet Data |
| 85 | ajet.org.au | <1 | Publication |
| 86 | article.ajtas.org | <1 | Publication |

| | | | |
|---|---|---|---|
| 87 | arxiv.org | <1 | Publication |
| 88 | arxiv.org | <1 | Publication |
| 89 | asbmr.onlinelibrary.wiley.com | <1 | Internet Data |
| 90 | asbmr.onlinelibrary.wiley.com | <1 | Internet Data |
| 91 | biomedcentral.com | <1 | Internet Data |
| 92 | Design of a BR-ABC Algorithm-Based Fuzzy Model for Glucose Detection by Gupta-2020 | <1 | Publication |
| 93 | docplayer.net | <1 | Internet Data |
| 94 | energy5.com | <1 | Internet Data |
| 95 | Height of Right and Left Ethmoid Roofs Aspects of Laterality in 644 Patients by Rei-2011 | <1 | Publication |
| 96 | LANDSLIDE DETECTION FROM SATELLITE IMAGES USING MACHINE LEARNING BY 18QE1D1120 YR 2020, JNTUH | <1 | Student Paper |
| 97 | Language Model-Driven Topic Clustering and Summarization for News Arti by Yang-2019 | <1 | Publication |
| 98 | Mapping Landslide Hazards in Western Nepal Comparing Qualitative and by REGMI-2010 | <1 | Publication |
| 99 | mdpi.com | <1 | Internet Data |
| 100 | MR angiography of the portal venous system by Han-1996 | <1 | Publication |
| 101 | scholarworks.umass.edu | <1 | Publication |
| 102 | solidrockglasgow.com | <1 | Internet Data |
| 103 | stephypublishers.com | <1 | Internet Data |

| | | | |
|---|---|---|---|
| 104 | Submitted to Visvesvaraya Technological University, Belagavi | <1 | Student Paper |
| 105 | Submitted to Visvesvaraya Technological University, Belagavi | <1 | Student Paper |
| 106 | Using colleague ratings to evaluate the faculty members contribution to instruc by Donal-1982 | <1 | Publication |
| 107 | www.adv-geosci.net | <1 | Publication |
| 108 | www.bits-goa.ac.in | <1 | Publication |
| 109 | www.freepatentsonline.com | <1 | Internet Data |
| 110 | www.frontiersin.org | <1 | Publication |
| 111 | www.intechopen.com | <1 | Publication |
| 112 | www.jstage.jst.go.jp | <1 | Publication |
| 113 | www.linkedin.com | <1 | Internet Data |
| 114 | www.mdpi.com | <1 | Internet Data |
| 115 | www.mdpi.com | <1 | Internet Data |
| 116 | www.readbag.com | <1 | Internet Data |