# Classification of Water Bodies Using Ensemble of U-Net and Random Forest Algorithm

Vasavi S *, Venkata Kalyan Chintalapudi, and Akhila Sree Rajeswari Vuppuluri

Velagapudi Ramakrishna Siddhartha Engineering College, Andhra Pradesh, India
Email: vasavi.movva@gmail.com (V.S.); 208w1a0511@vrsec.ac.in (V.K.C.); 208w1a0560@vrsec.ac.in (A.S.R.V.)
*Corresponding author

***Abstract*—Water bodies classification using remote sensing and deep learning techniques plays a pivotal role in the effective management of water resources. This study aims to address the challenge of accurate water body detection and classification, which is essential for understanding their distribution and characteristics, ultimately informing water usage and conservation efforts. Current methodologies predominantly rely on Support Vector Machines (SVM) and pixel-based approaches, resulting in suboptimal accuracy. In response, this paper proposes an ensemble model that combines the U-Net neural network and the Random Forest algorithm for enhanced water body detection and classification. The research commences by obtaining high-resolution satellite images with a resolution of 0.5 m. The U-Net model is employed to segment water bodies, and contour analysis is subsequently applied to extract shape features. The Random Forest Classifier is then utilized to classify the segmented water bodies into distinct categories, including rivers, ponds, lakes, canals, and other water bodies. Following the U-Net segmentation, the rasterized segments are converted into vector format. These vector data are leveraged to update Geographic Information System (GIS) maps, contributing to more accurate cartographic representations. The proposed approach is rigorously evaluated using a dataset from urban areas in Kolkata, West Bengal, India. The achieved accuracy rate stands at 67.01%.**

***Keywords*—water body, shape features, U-Net, random forest algorithm, semantic segmentation, Geographic Information System (GIS) map**

## I. Introduction

Water bodies are crucial for the effective management and monitoring of water resources. The motivation for precise water body classification stems from the growing global population and the necessity for efficient water resource management to address future challenges and preserve natural resources. Accurate detection and classification of water bodies provide valuable insights into their distribution and characteristics, supporting water usage and conservation efforts [1]. However, traditional methods for water body classification, such as Support Vector Machines (SVM) and pixel-based/object-based approaches, often fall short in achieving high accuracy [2]. To tackle this, the study employs high-resolution satellite imagery and combines deep learning techniques—specifically, the U-Net and Random Forest algorithms. The U-Net detects water bodies, while the Random Forest algorithm classifies them into five categories based on shape features, such as rivers, lakes, canals, ponds, and other water bodies. This approach provides a practical solution to improve water body classification, essential for the conservation and management of water resources.

The ensemble model proposed in this paper uses the U-Net and Random Forest algorithm to detect and classify water bodies. The water bodies are segmented using U-Net [3], while the Random Forest algorithm utilizes shape features to classify the water body. This approach has shown promising results in various image classification tasks. Very high resolution Satellite images with 0.5 m resolution are taken from SAS planet, providing detailed information for accurate water body classification. The U-Net is a Convolutional neural network with a distinctive "U" shape, widely used in image segmentation, featuring both an encoder and a decoder block [3]. Skip connections facilitate spatial data transfer between the encoder and decoder, enhancing segmentation accuracy and preserving spatial information [3]. The proposed U-Net model effectively segments the water bodies by leveraging its architecture designed for semantic segmentation tasks [4].

Once the water bodies are segmented, the next step is to classify them into specific categories such as rivers, lakes, ponds, canals and other water bodies. This is achieved using the Random Forest Classifier, which considers shape features extracted of the segmented water bodies. Shape features are calculated using contour analysis, which captures geometric properties such as area, perimeter etc., Contour analysis plays a crucial role in extracting shape features from water bodies for classification purposes. Contours are the boundaries of connected regions in an image and can be used to capture the shape characteristics of water bodies accurately [5].

These features are utilized to train a Random Forest Classifier, which classifies the water bodies into categories such as rivers, lakes, ponds, canals and other water bodies. Random Forest is a machine learning approach that combines numerous decision trees to generate a stronger and more reliable classification

Manuscript received August 22, 2023; revised October 8, 2023; accepted October 17, 2023; published March 8, 2024.

model [6, 7]. It is ideal for classifying water bodies based on shape features because it performs particularly well in classification problems requiring high-dimensional feature spaces.

After the classification process, geo-referencing and change detection techniques are employed to analyze and understand the changes occurring in the water bodies over time. Geo-referencing is the process of assigning geographic coordinates to a satellite image or raster dataset, allowing it to be accurately positioned on the Earth's surface. It involves selecting reference data, identifying Ground Control Points (GCPs), and applying transformation models to align the image with known geographic coordinates [8]. The predicted mask image or the classified raster image is aligned with the corresponding spatial reference system, enabling spatial analysis and integration with other geospatial data.

After geo-referencing a raster image, the next step is raster to vector conversion. Raster to vector conversion involves transforming a geo-referenced raster image into a vector format, such as GeoJSON [9]. This process converts the pixel-based representation of the image into vector-based geometries, such as points, lines, and polygons, which can be represented as coordinates and attributes. Converting raster data to vector format enables more versatile analysis, feature extraction, and compatibility with various Geographic Information System (GIS) software [9]. GeoJSON files store geographic data in JSON (JavaScript Object Notation) format, providing a convenient and interoperable way to exchange and work with geospatial information [10].

Change detection is a critical step in analyzing the differences between past and present water body data. By comparing the GeoJSON files representing different time periods, we can identify the water bodies that are newly water bodies that have emerged, existing water bodies that have disappeared, and water bodies that are unchanged. This information is valuable for understanding the dynamics of water resources and supporting decision-making processes related to water management and conservation efforts.

The proposed system was evaluated on the Indian dataset, specifically the urban areas of Kolkata, West Bengal, to assess its performance in real-world scenarios. The classification will be limited to differentiating between rivers, lakes, canals, ponds and other water bodies, which are common water bodies found in the urban landscape.

The objectives of the study as follows: 1) Develop an exclusive dataset tailored to Kolkata, India. 2) Construct an ensemble model incorporating U-Net for segmentation and Random Forest for classification. 3) Conduct Change Detection analysis specifically focused on water bodies. 4) Revise GIS maps by integrating the identified changes through the Change Detection process.

This paper is organized as follows: The literature review of numerous works on water body detection and classification is covered in the first section of Section II. The next section, Section III, which comes next, goes into further detail on the suggested strategy and design. The interpretations and outcomes are discussed in Section IV.

## II. Related Works

For tropical wetland mapping, Liu *et al.* [11] discusses using Sentinel-2 imagery and the Google Earth Engine platform. It entails separating inland and coastal wetlands using Tide Height and Difference Threshold values, masking non-wetland regions using NDVI, NDBI, and NDWI values, segmenting different wetland clusters using the IF and SNIC algorithms, and using a Phenological classifier to separate vegetation and water body mixed wetlands. Random Forest is used for classification.

*Advantages*: Their method maps ten types of wetlands with 82.07%. The generated water cover map is valuable for monitoring, protecting, and managing tropical wetlands.

*Disadvantages*: The Sentinel-2 satellite, released in 2015, has limited historical observations for studying long-term changes in tropical wetlands. To address this difficulty, further satellite data must be included.

According to the work discussed in [12], the images are pre-processed to remove noise and enhance the contrast of water bodies. Next, a two-level pixel-based method is applied to preprocessed images. The water bodies are then extracted using a Water Index (WI) that identifies pixels with high water content. In the second step, the batch processing framework is used to automate the water extraction process. The framework allows for the simultaneous processing of multiple images, making it suitable for large-scale water information extraction. Finally, the extracted water bodies are validated using ground truth data or other available water body datasets.

*Advantages*: The object detection approach reduced salt and pepper noise, reduced the water pixels which are misclassified successfully, boosting the accuracy of water body extraction. It also ensured that ponds and other bodies of water were well represented in the retrieved photos.

*Disadvantages*: Because of the high resolution of the dataset, computations take longer and consume more memory.

Tang *et al.* [13] created a water dataset using JRC GSW. Following that the image compositing is accomplished monthly by combining Landsat-7/8 and Sentinel-1 images, then followed by the data merging by Digital Elevation Models. Classification features like spectrum reflectance features, spectral indices and radar backscattering features are considered. Then, using the sample points and classification features, the RF classifier is trained in each cell consisting of a 5×5 geographical grid to extract surface water. By comparing surface water distribution data from the same time period with JRC GSW data and the water body borders that are derived by higher resolution image interpretation, the accuracy of water body extraction findings was finally assessed.

*Advantages*: The suggested approach would accurately and consistently extract water bodies on a large scale, even with varying land cover and lake sizes. The results

show that the method is suitable for mapping surface water on a broad scale with high spatial and temporal resolution.

*Disadvantages*: The impervious surface dataset and land glacier data were used. The approach may have problems removing ice, snow, and the buildings interference information, which can compromise surface water mapping accuracy.

The NDWI is sharpened in the 1st stage to binarize the water pixels using Otsu approach. In the second stage, a four-neighbor connectivity technique is employed to generate water objects by joining independent water pixels. In the third stage, the water object's form and water quality attributes are computed and given to the classifier for classification [14].

*Advantages*: Large-scale aquaculture ponds can be identified using a combination of geometry and water quality indicators. Under the influence of water quality factors, MF has the capacity to appropriately categorize mixed water objects.

*Disadvantages*: Aquaculture ponds are another term for some natural lakes. The sample data set has a significant impact on the Mobile Financial Services (MFS) categorization as well.

According to the methodology in [15], The automated approach consists of: calculating surface water maps and flood frequency maps, classification based on shape features, classification based on phenology, and then calculating the accuracy are all available. The Random Forest (RF) method is used to do shape-based categorization. They applied this method on time series Sentinel-1 and Sentinel-2 photographs (received from Google Earth Engine in 2020) which build a water cover map of china with high spatial resolution.

*Advantages*: The method's ability to obtain high spatial-temporal resolution data on all sorts of full water cover at vast sizes was proved by having the comparisons with 7 other data products relating to water. This approach may be used to generate long-term comprehensive water cover maps at the global and national scales, as well as a water cover classification map for existing datasets.

*Disadvantages*: There were less rice fields in the CWaC due to Sentinel-1's imaging technique and Sentinel-2's restricted effective observation. Several unique situations, where some lakes are fed by rivers, cannot be handled by the feature-based approach. The lakes and river were manually edited apart, which required extensive post-processing.

Work described in [16] continues with extra trees classifier and opensource segmentation using the training data that is automatically generated from freely available data. Throughout the study period, vegetated water accounted for 70% of total water was mapped, highlights the importance of mapping open and vegetated water bodies for surface water mapping. They devised solutions that could be beneficial to public health in malaria-prone areas which have more river flooding in Africa. This research highlights the significance of L-band SAR which is widely available.

*Advantages*: This categorization system can automatically generate training data for open water and flooded vegetation.

*Disadvantages*: Although publicly available, Sentinel-1 C-band radar is limited in its ability to produce a signal from vegetated water bodies until grasses are 20 cm tall with suitable gaps between grass tussocks or clumps.

A test site's wetland area was defined using single polarized TerraSAR-X data and single and full-polarized RADARSAT-2 data [17]. The researchers intended to investigate how well several feature types, such as polarimetric scattering, intensity, and interferometric coherence performed in various categorization scenarios for wetland mapping. Classifier influence is accessed by using Random Forest (RF), SVM and an ensemble classifier. The research looked at how classification accuracy of both SVM and RF classifiers is affected by adding all feature types, as well as the classification accuracy of each feature type separately. To optimise the number of inputs and type, the researchers found that to integrate both Spearman's rank-order correlation and RF variable importance to get better results.

*Advantages*: The enhanced discriminating ability resulting from the synergistic application of several input characteristics. Combine the relevance of RF variables with analysis of Spearman's rank-order correlation for optimizing both the kind and amount of input characteristics. An ensemble classifier, RF, outperforms SVM, when employing an object-based classification strategy.

*Disadvantages*: Land cover types other than wetlands must be classified.

According to methodology in [18], They sharpened the 20 m SWIR bands by using NDWI of 10 m resolution image, which was focused on urban area water extraction. Then, the water maps were retrieved using object-level MNDWI mapping and a minimal bottom adjustment threshold. Their method was compared to existing techniques. According to the results, the suggested NDWI-based MNDWI images has greater separability and is more successful than current techniques for both classification-level and boundary-level water maps.

*Advantages*: Comparing with MNDWI (20 m) and NDWI (10 m), the MNDWI results are more accurate. In downtown Beijing, water maps extracted using MNDWI enhanced the Kappa coefficient about 0.3 and the boundary index by more than 90%.

*Disadvantages*: Normal NDWI provides adequate boundary position and classification accuracy for the coastal Yantai research location, where MNDWI approaches produce less accuracy, particularly for boundary index.

The following two areas have major contributions in the work reported in [19]. In terms of application, the extraction of urban water bodies (like rivers, ponds, lakes, and canals) from satellite images, which gotten little attention in the existing literature, is being investigated. In terms of strategy, a novel machine learning method is proposed, which includes pixel-level water-area detection followed by object-level water-type categorization. At

pixel level, NDWI, MSI, and NDVI information indexes are used to extract water zones. The shadow index (MSI) can help to reduce shadow false alarms greatly.

*Advantages*: The Shadow Index (MSI) has the potential to significantly reduce shadow false alarms, which are major errors in urban water bodies extraction. The suggested framework's dual-level design may feature information extraction from the pixel and object levels for detection and classification, respectively.

*Disadvantages*: Detection and identification of type of water body was critical criteria for urban water monitoring and management, are only possible in the defined scope zone, and water extraction requires further automation. Due to the time constraints of the sample approach, a more intelligent sampling strategy for water extraction and categorization is necessary.

*Research Gaps*

The field of water body mapping and GIS data management faces significant research gaps. Outdated GIS maps and water body data hinder resource management, while the complexity of India's diverse water bodies challenges existing systems. The absence of standardized datasets for training and testing, coupled with the variation in water bodies classification into lakes, rivers, ponds, and canals, pose challenges to developing robust and generalizable models. Furthermore, improving the accuracy of current algorithms remains a priority in this context. Addressing these gaps is vital for more accurate mapping and better-informed urban planning, disaster management, and environmental monitoring.

## III. Methods

An outline of the proposed system architecture, methodology, algorithms, and dataset for the designed system are presented here.

### A. Architecture

Fig. 1 depicts the proposed U-Net model for detecting the water bodies.

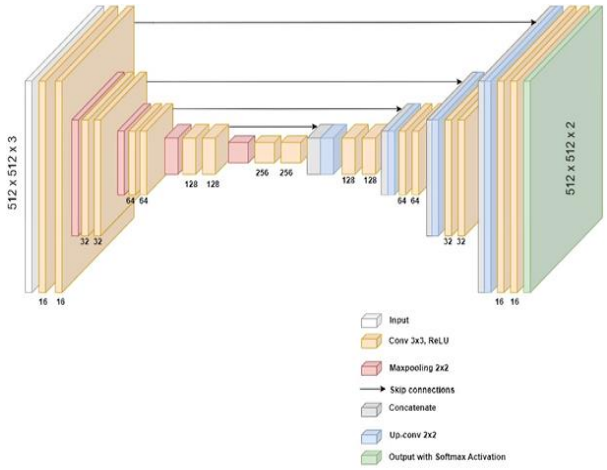



Fig. 1. Proposed U-Net architecture.

**Convolutional block:** A Convolutional block consists of two consecutive Convolutional layers with ReLU activation. Each layer applies a 3x3 convolution operation with a specified number of filters.

**Encoder block:** An encoder block consists of a Convolutional block followed by max pooling. The Convolutional block applies convolutions to the input, while the max pooling operation reduces the spatial dimensions by a factor of 2.

**Decoder block:** A decoder block consists of a transposed convolution layer, concatenation with skip features, and a Convolutional block. The transposed convolution layer performs up-sampling by applying a 2×2 convolution operation with a stride of 2. The skip features are the outputs from the corresponding encoder block and are concatenated with the up-sampled tensor. The concatenated tensor is then passed through a Convolutional block.

The above U-Net model starts by creating an input layer with a shape of (512, 512, 3). The Contraction path of the model is then defined, consisting of four Convolutional blocks with 16, 32, 64, and 128 filters, respectively. The central layer of the U-Net model is then defined, which consists of two Convolutional layers with 256 filters. Expansive path of the model is defined next, consisting of four decoder blocks having 128, 64, 32, and 16 filters, respectively. Finally, an output layer with 2 filters and *Softmax* activation function is defined, and the obtained model is returned.

$$softmax(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \quad (1)$$

where $x_i$ represents the *i-th* element of the input vector $x$ and $n$ is the total number of elements in the vector. The *Softmax* function returns a probability distribution across the input vector, with each output vector element representing the likelihood of that class [20]. Table I shows the comparison of existing Unet model with the proposed Unet model.

TABLE I. Modified U-Net Model Details

| Parameters | Existing Model [3] | Modified Model |
|---|---|---|
| Channels in input image | 1 | 3 |
| Shape of input image | (572,572,1) | (512,512,3) |
| Strides | 1 | 2 |
| Input Kernel size | 3×3 | 3×3 |
| Initial No.of Filters | 64 | 64 |
| Parameters | 1.9 M | 1.91 M |
| Pooling type | Max Pooling | Max Pooling |
| Size of Max Pooling at every layer | 2×2 | 2×2 |
| Number of layers | 23 | 32 |
| Number of channels in output image | 1 | 1 |

After the water body has been segmented, shape features such as area, perimeter, and compactness can be extracted using contours. Twelve shape features are calculated for each water body which is used to classify them into 4 classes. Using the Random Forest Classifier, we will classify the water bodies based on shape features into rivers, lakes, ponds and canals. The decision trees

used in Random Forest are constructed using the CART algorithm, which is a binary recursive partitioning algorithm that operates by splitting the data into subsets based on the values of one input variable at a time, until a stopping criterion is reached. The splitting criterion used by CART is the Gini impurity, which measures the quality of a split by the degree of impurity of the subsets, with impurity being defined as the probability of misclassifying a randomly chosen element in the subset.

$$Gini = 1 - \sum_{i=1}^{n} p_i^2 \quad (2)$$

where $n$ denotes number of classes, and $p_i$ is the proportion of samples belonging to class $i$.

Here the Random Forest model contains 300 decision trees, where each tree is limited to a maximum depth of 15 nodes. Fig. 2 depicts the Random Forest Classifier's structure. It shows how 12 shape features are used to classify water body segments. Multiple decision trees are trained on subsets of the data and make independent predictions, with the final classification of a given sample into rivers, ponds, lakes, canals and other water bodies will be the majority vote of the individual decision trees.

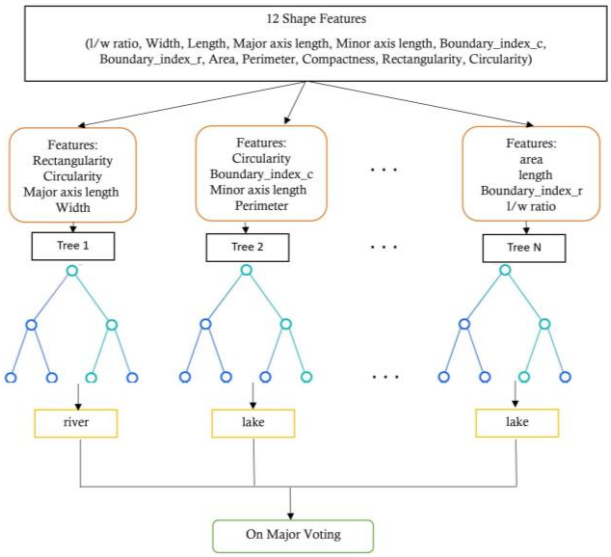



Fig. 2. Random forest classifier.

## B. Methodology

Fig. 3 presents the proposed system flowchart that included preprocessing, water body segmentation, water body classification, Post-processing, Geo-Referencing, Raster to vector conversion, Change Detection and Updation of GIS maps.

*1) Preprocessing*

Images can be effected by noises such as salt and pepper noise, speckle noise and Gaussian noise [21]. Techniques like Median Filtering and Gaussian Filtering (see Algorithm 1) can remove salt and pepper noise, but Bilateral Filtering can successfully reduce Gaussian and salt & pepper noise [22].

**Algorithm 1: Bilateral Filtering**

**Step 1:** Initialize an empty filtered image J with the same size as the input image I.

**Step 2:** For each pixel (x, y) in I do the following:

i. Initialize the filtered pixel value J(x, y) to zero.

ii. Initialize the weight sum W x, y) to zero.

iii. For each pixel (i, j) in a K × K window centered at (x, y) do the following:

- Compute the spatial distance between (x, y) and (i, j) as

$$d = \sqrt{(x-i)^2 + (y-j)^2} \quad (3)$$

- Compute the intensity difference between I(x, y) and I(i, j) as

$$r = |I(x,y) - I(i,j)| \quad (4)$$

- Compute the bilateral filter weight w as

$$W = e^{\left(-\left(\frac{d^2}{2\sigma_s^2}\right)-\left(\frac{r^2}{2\sigma_r^2}\right)\right)} \quad (5)$$

where $\sigma_s$ and $\sigma_r$ are the standard deviations of the spatial and intensity domains, respectively.

- Add the weighted pixel value I(i, j)*w to the filtered pixel value J(x, y).
- Add the weight w to the weight sum W(x, y).

iv. Set the filtered pixel value J(x, y) to J(x, y)/W(x, y).

**Step 3:** Return the filtered image J.

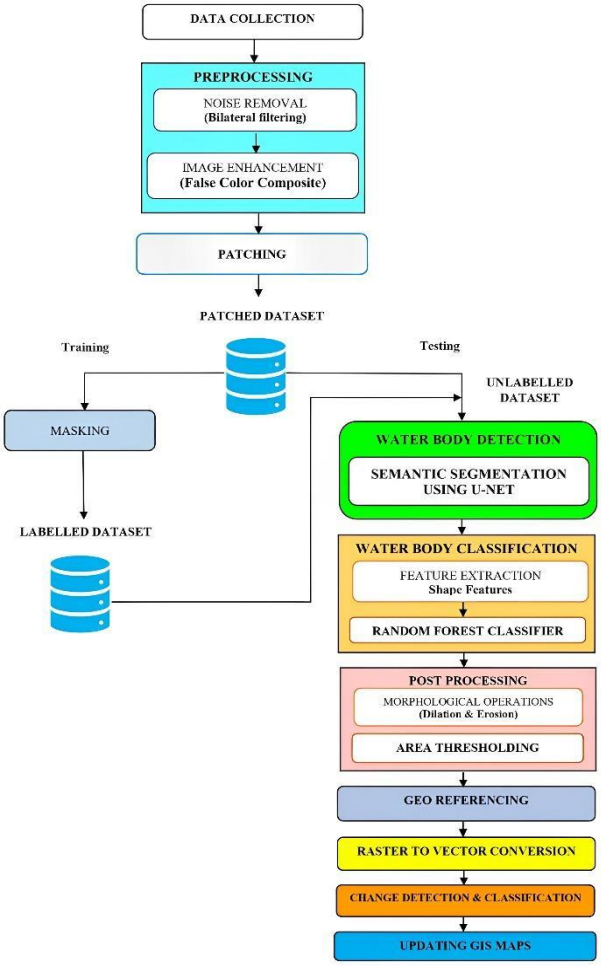



Fig. 3. Training and testing flow chart.

Noise removal techniques are used to enhance image quality. Image enhancing techniques such as False Color Composite (see Algorithm 2), on the other hand, can be used to recover lost data [23].

**Algorithm 2: False Color Composition**

**Step 1:** Import the multispectral image and its corresponding metadata.
**Step 2:** Choose which bands to use for each color channel in the FCC.
**Step 3:** Normalize the pixel values in each band to a common scale, such as between 0 and 255.
**Step 4:** Create a new image with the same dimensions as the input image, with three channels for red, green, and blue, respectively.
**Step 5:** Assign the pixel values from the selected bands to the corresponding color channels in the new image.

*2) Water body segmentation*

The preprocessed images are divided into patches of size 512×512 pixels (see Algorithm 3) and useful patches (see Algorithm 4) are taken to undergo the segmentation process.

**Algorithm 3: Patching**

**Step 1:** Read the image in TIFF format.
**Step 2:** Crop the image from the top left corner to the nearest size of the image that is divisible by the 512.
**Step 3:** Extract patches from the image using patchify() function.
**Step 4:** Loop through the patches and save each patch as a separate TIFF image.
**Step 5:** Repeat the same process for mask images.

**Algorithm 4: Extracting Useful Patches**

**Step 1:** Read the image and mask.
**Step 2:** Calculate the number of white pixels in the mask.
**Step 3:** If the percentage of white pixels is greater than 5 percent, Save the image and mask.
**Step 4:** Define the central layer of U-Net with 2 Convolutional layers with 256 filters, followed by no dropout layer.

The next step involves using the U-Net for semantic segmentation [24] (see Algorithm 5).

**Algorithm 5: Semantic Segmentation Using U-Net**

**Step 1:** Split the dataset into training (80 %) and testing sets (20 %).
**Step 2:** Define an input layer with shape (512,512,3) and assign it to the variable inputs.
**Step 3:** I Define Contraction path of the U-Net model, which consists of 4 Convolutional layers with 16, 32, 64, and 128 filters, respectively, a Convolutional layer with the same number of filters, and a max pooling layer.
**Step 4:** Repeat the above steps for all patched images and masks.
**Step 5:** The Expansive path consists of 4 transposed Convolutional layers with decreasing filter sizes, concatenated with corresponding Contraction path layers. It also includes 2 Convolutional layers with the same number of filters.
**Step 6:** Define an output layer with 2 filters and Softmax activation function.
**Step 7**: After training the U-Net model on a 512×512 image, now we will predict the output for an image of size 12289×6874.
**Step 8**: Split the image into individual patches of size 512×512.
**Step 9**: Apply the model to each square patch and predict the outputs.
**Step 10**: Merge the predicted outputs smoothly and save the predicted mask image.

*3) Water body classification*

Feature Extraction: Contours can be used to derive shape characteristics from segmented water bodies. Twelve shape characteristics are calculated and their respective formulas are mentioned in the below Algorithm 6.

**Algorithm 6: Calculating Shape Features Of Water Bodies**

**Step 1**: Load grayscale image and apply binary threshold.
**Step 2**: Find contours using thresholding method and convert grayscale image to color image.
**Step 3**: For each contour, draw it on color image and calculate shape features:

a. Calculate Major and minor axis length of ellipse using fitEllipse() function.
b. Calculate Length, width, and length-to-width ratio of bounding box using boundingRect() function.
c. Calculate area and perimeter of contour using moments.
d. Calculate Rectangularity, circularity, boundary index (circular objects), boundary index (rectangular objects), and compactness of contour

$$rectangularity = \frac{area}{length * width} \quad (6)$$

$$circularity = \frac{4 * \pi * area}{perimeter^2} \quad (7)$$

$$boundaryindex(C) = \frac{perimeter}{(2*\pi*majoraxis)} \quad (8)$$

$$boundaryindex(R) = \frac{perimeter}{(2*(majoraxis+minoraxis))} \quad (9)$$

$$compactness = \frac{area}{(\pi*majoraxis*minoraxis)} \quad (10)$$

*e.* Append calculated shape features to a list

**Step 4:** Create DataFrame from the list with appropriate column names and save DataFrame as CSV file.

*4) Random forest classification*

Random Forest (RF) Classifier (see Algorithm 7) is trained using a csv file containing the estimated shape features and class labels. We have considered 4 class labels. The trained RF model classifies the water bodies into 4 classes (rivers, lakes, canals, ponds). Finally, the trained RF model is used to label the expected mask image of size 12289×6874 with a specific color representing its projected label.

**Algorithm 7: Random Forest Classifier**

**Step 1:** Load the dataset from the shape features CSV file using the pandas library.
**Step 2:** Train the RandomForestClassifier with n_estimators = 300 and max depth = 15.
**Step 3:** Classify the water bodies into rivers, lakes, ponds and canals.
**Step 4:** Calculate shape features for the predicted mask image, a list of contours found in the image, and a color image.
**Step 5:** Use trained random forest model to predict labels for the image and store them in a list of labels.
**Step 6:** Apply area thresholding such that the area of the water body segment is less than 500,00, change the predicted label to other water bodies.
**Step 7:** Define a dictionary that maps each label to an RGB color, i.e, river: red, pond: orange, canal: violet, lake: blue and other water: black.
**Step 8:** For each contour, find the corresponding predicted label from the labels list, and use the label's color from the label dictionary to draw and fill the contour on the image.

*5) Post processing*

The output of the U-Net model may contain certain imperfections, such as small isolated regions, fragmented boundaries, or noisy predictions. To avoid those, we perform morphological operations like dilation and erosion. Dilation expands the boundaries, while erosion shrinks them. Later we have applied area thresholding for further improvement of accuracy. By applying appropriate combinations of these operations, the imperfections are consolidated (see Algorithm 8).

**Algorithm 8: Post Processing**

**Step 1:** Read the predicted mask image where 255 represents the foreground (water bodies).
**Step 2:** Define the structuring element for dilation and erosion.
**Step 3:** Perform erosion on the image using the defined kernel and a specified number of iterations.

a) Place the center of the kernel on each pixel of the input image, one by one.
b) Compare the pixel values of the input image that are overlapped by the kernel with the corresponding kernel values.
c) If all the corresponding pixel values in the kernel are non-zero (white), the center pixel of the kernel is part of the foreground. Otherwise, it is part of the background.
d) Set the output pixel value (in the eroded image) to the minimum value of the foreground pixels covered by the kernel. This operation effectively erodes or shrinks the boundaries of the foreground objects.
e) Repeat steps a-d for every pixel in the input image.
f) After processing all pixels, the resulting image is the eroded image.

**Step 4:** Set an area threshold to filter out small regions.

a) Perform connected component labeling using connectedComponentsWithStats() function on the eroded image to identify separate regions.
b) Iterate over the connected components and check their area against the threshold 50000.
c) Assign black color (0) to regions below the threshold by modifying the labels.
d) Convert the modified labels back to a binary image.

**Step 5:** Perform dilation on the area thresholded image using the same kernel and a specified number of iterations.

a) Place the center of the kernel on each pixel of the input image, one by one.
b) Compare the pixel values of the input image that are overlapped by the kernel with the corresponding kernel values.
c) If at least one corresponding pixel value in the kernel is non-zero (white), the center pixel of the kernel is part of the foreground. Otherwise, it is part of the background.
d) Set the output pixel value (in the dilated image) to the maximum value of the foreground pixels covered by the kernel. This operation effectively expands or thickens the boundaries of the foreground objects.
e) Repeat steps a-d for every pixel in the input image.
f) After processing all pixels, the resulting image is the dilated image.

**Step 6:** Save the modified image.

*6) Geo referencing*

The output obtained from the water body segmentation does not contain geographical information in it, i.e., coordinate information is lost. To restore the geographical information, Geo-Referencing is performed (see Algorithm 9).

**Algorithm 9: Geo-Referencing**

**Step 1:** Read the mask image and extract its dimensions.
**Step 2:** Read the reference GeoTIFF image and retrieve its geo-referencing information.
**Step 3:** Create a new GeoTIFF dataset for the output image.
**Step 4:** Set the geo-transform and projection of the output dataset.
**Step 5:** Read the pixel data from the mask image and write it to the output.

*7) Raster to vector conversion*

Following the process of geo-referencing, the output obtained is typically a GeoTIFF file (raster image). However, GIS maps primarily rely on vector formats. Therefore, in order to update GIS maps, the GeoTIFF file (raster image) needs to be converted into the GeoJSON format (vector image) (see Algorithm 10).

**Algorithm 10: Raster to Vector Conversion**

**Step 1:** Open the Geo-referenced mask file.
**Step 2:** Read the image data, transformation, and coordinate reference system (CRS) from the opened mask file.
**Step 3:** Create a binary mask where white pixels are True and black pixels are False.
**Step 4:** Extract vector shapes from the binary mask.
**Step 5:** Create Shapely geometries from the extracted shapes.
**Step 6:** Create a GeoDataFrame from the Shapely geometries, using the extracted CRS.
**Step 7:** Save the GeoDataFrame as a GeoJSON file using the defined output path and specifying the driver as 'GeoJSON'.

*8) Change detection and classification*

Using set operations on bi-temporal vector data, we can detect and classify changes in water bodies as new (see Algorithm 11), Extinct, or unchanged. This process helps monitor urban development and analyze spatial dynamics over time.

**Algorithm 11: Change Detection of Water Bodies**

**Step 1:** Predicted masks of image of year 2020(I1) and 2023(I2) are converted to vector format (.geojson ).
**Step 2:** Perform intersection operation, which results in unchanged water bodies.
**Step 3:** Perform set difference operation on I1 and intersection results in extinct water bodies.
**Step 4:** Perform set difference operation on I2 and intersection results in new water bodies.

*9) Updation of GIS maps*

To keep the GIS maps up to date and reflect the latest changes on the ground, the identified changes from the change detection analysis using geojson files are integrated into the GIS map layers (see Algorithm 12).

**Algorithm 12: Updation of GIS Maps**

**Step 1:** Collect and import updated vector data into QGIS software.
**Step 2:** Edit the data to accurately represent real-world features.
**Step 3:** Merge updated data with existing GIS map data.
**Step 4:** Validate the updated map for accuracy and completeness.
**Step 5**: Publish the updated map online or in print.

## *C. Study Area and Dataset Collection*

For the water bodies classification, the city of Kolkata, located in the state of West Bengal, India, situated at 22°34' N latitude and 88°22' E longitude is considered. The water demand in Kolkata has been increasing rapidly due to the growing population and urbanization. The water supply in Kolkata mainly comes from surface water sources such as the Hooghly River, Tolly's Nullah and other smaller water bodies, as well as underground sources. To meet the water demand of the population in 2021, Kolkata would require around 1,000 MLD (220 MGD) of water. However, the available supply is only around 750 MLD (165 MGD), leading to a shortage of about 250 MLD (55 MGD).

This shortage is mainly due to factors such as over-extraction of groundwater, contamination of surface water sources, and inadequate infrastructure for water supply and distribution. Fig. 4 presents the study area considered for this research.

Name of the Dataset: SasPlanet Images. Description: The VHRS images of size 12289×6874 are collected from SasPlanet which are about 0.5 m resolution. This database can be used for the segmentation of water bodies. The particularities of this setting are that each image depicts different water body images categorized into two classes such as water bodies and non-water bodies. Also, field data is collected for effective classification.

Classes: 2
Number of Images: 1333
Train set size: 1066
Test set size: 267
Image resolution: 512×512 pixels

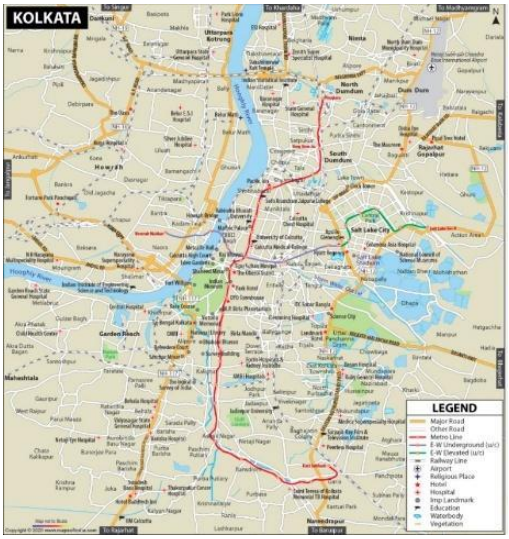



Fig. 4. Study area.

For validation purposes, we have taken cartosat-3 images which are of 0.5 m resolution. Fig. 5 shows the sample cartosat-3 image.

Name of the Dataset: Cartosat-3 Images.
Description: The VHRS images of size 512×512 are collected from Cartosat-3 which are about 0.5 m resolution. This database can be used for the validation purpose.
Number of Images: 268
Image resolution: 512×512 pixels

Fig. 5. Validation dataset specimen.

Masking is done using QGIS software. Water bodies are masked with white color. Fig. 6 shows the satellite image and the corresponding mask image.

Fig. 6. Original image and its mask image.

Name of the Dataset: Shape Features dataset (.csv file) Description: This dataset can be used for the classification of water bodies. The dataset contains shape

features of different water bodies. These shape features of water bodies are calculated from satellite images. The particularities of this set- ting are that each record depicts different water bodies classified into five classes such as lakes, rivers, canals, ponds and other water bodies.

Classes:5

Number of water bodies: 152

Train set size: 122

Test set size: 30

The individual water body shape features present in the mask image are calculated using Contours and stored in the .csv file. Fig. 7 presents the detected water bodies using contours and Fig. 8 presents a screenshot of the .csv file created with the shape features.

Fig. 7. Water bodies detected in mask images using contours.

Manually we label the water bodies as rivers, lakes, ponds, canals and other water bodies. Sample shape features in .csv file are shown in Fig. 8.

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Object | l/w | area | Rectangularity | Major axis | Minor axis | Perimeter | Circularity | Length | Width | boundary_index_c | boundary_index_r | Compactness | class |
| 2 | 1 | 1.993243 | 50172 | 0.296422199 | 1321.354 | 128.0948 | 1744.569 | 0.207155 | 590 | 296 | 0.210130702 | 0.601804309 | 0.094354196 | canal |
| 3 | 2 | 2.47619 | 131998 | 0.63727272 | 812.7755 | 254.8422 | 2682.711 | 0.230478 | 832 | 336 | 0.525319746 | 1.25640037 | 0.202850378 | pond |
| 4 | 3 | 3.269341 | 202606.5 | 0.004628861 | 37891.47 | 1155.148 | 2961.355 | 0.290323 | 1141 | 349 | 0.012438544 | 0.037920763 | 0.001473413 | pond |
| 5 | 4 | 2 | 300 | 0.887573964 | 0 | 0 | 74 | 0.688442 | 26 | 13 | 0.452979836 | 0.948717949 | 0.282523806 | pond |
| 6 | 5 | 1.11157 | 35312 | 0.755869596 | 297.1685 | 157.2073 | 1029.941 | 0.418318 | 269 | 242 | 0.55160742 | 1.133358268 | 0.240600968 | pond |
| 7 | 6 | 1.91018 | 440544 | 0.388304561 | 2895.419 | 391.8369 | 9414.441 | 0.062461 | 2871 | 1503 | 0.517491877 | 1.43196024 | 0.123601285 | canal |
| 8 | 7 | 1.416149 | 13687 | 0.676382752 | 270.677 | 74.75916 | 764.6274 | 0.294183 | 228 | 161 | 0.449592447 | 1.106756488 | 0.215299499 | canal |
| 9 | 8 | 2.107438 | 10067 | 0.364934203 | 434.4762 | 63.49207 | 740.9706 | 0.230413 | 255 | 121 | 0.271428493 | 0.743993712 | 0.116162263 | canal |
| 10 | 9 | 1.144743 | 3503966 | 0.300876954 | 5164.198 | 2255.112 | 9389.061 | 0.499488 | 2428 | 2121 | 0.289360864 | 0.632744946 | 0.09577219 | pond |
| 11 | 10 | 2.026633 | 5165338 | 0.878086683 | 4196.086 | 1401.9 | 14805.75 | 0.296106 | 4870 | 2403 | 0.561573511 | 1.322417935 | 0.279503908 | pond |

Fig. 8. Sample shape features in dataset.

### D. Evaluation Metrics

The evaluation metric considered for assessing the proposed U-Net model is Mean Intersection over Union (*IoU*), which is a commonly used evaluation metric in image segmentation tasks.

$$IOU = \frac{TP}{(TP+FP+FN)} \tag{11}$$

The *IoU* value ranges from 0 to 1, with a larger value indicating greater overlap between the expected and ground truth masks. An *IoU* of 1 indicates a perfect match, whereas an *IoU* of 0 indicates no overlap at all.

$$Mean\ IOU = \frac{\sum_{i=1}^{N} IOU_i}{N} \tag{12}$$

where $IOU_1$, $IOU_2$, ..., $IOU_n$ are the *IoU* values for each class, and *N* is the total number of classes.

The classification metrics for the Random Forest Classifier are Accuracy, Precision, Recall and F1 Score.

Accuracy measures the overall correctness of the classifier predictions.

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \tag{13}$$

Precision is defined as the percentage of accurately predicted positive samples among all anticipated positive samples.

$$Precision = \frac{TP}{(TP+FP)} \tag{14}$$

Recall is the percentage of accurately predicted positive samples among all actual positive samples.

$$Recall = \frac{TP}{(TP+FN)} \tag{15}$$

The F1 score is the harmonic mean of precision and recall, providing a balanced measure between the two.

$$F1-score = \frac{2\times precision\times recall}{(precision+recall)} \tag{16}$$

where TP: True Positive, TN: True Negative, FP: False Positive, FN: False Negative.

False Positive Rate (FPR) is a metric represents the proportion of actual negative instances (non-water bodies) that are incorrectly classified as positive instances (water bodies).

$$False\ Positive\ Rate = \frac{FP}{(FP+TN)} \tag{17}$$

### E. Software Requirements

Software requirements include:

- Python libraries including PIL, OS, Numpy, Scikit-learn and TensorFlow.
- Computing platforms like Google collab or Jupyter notebook.
- QGIS, a geographical information system tool.

### F. Hardware Requirements

- ×86 64-bit CPU
- 8 Gigabytes of RAM
- 4 Gigabytes of GPU
- Working platform of Windows 10.11 or above

## IV. Results and Discussion

In this part, the outcomes of the proposed system are discussed. The outcomes are attained after the proposed approaches are successfully implemented.

### A. Results

For salt and pepper noise and Gaussian noise removal, bilateral filter is used. Bilateral filter is a nonlinear edge-preserving smoothing filter that is used to eliminate noise from images while keeping their edges [25]. This filter is effective for reducing Gaussian noise and salt and pepper noise. Fig. 9 shows the bilateral filtered image output with an input of noisy image.

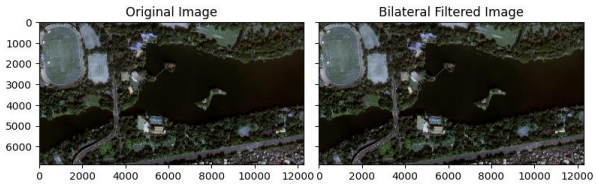



Fig. 9. Original image and bilateral filtered image.

False color composite technique is used to enhance the image. This technique is used to display satellite images that assign different colors to different bands of the electromagnetic spectrum, creating a composite image that emphasizes certain features or properties of the scene [23]. The output in Fig. 10 shows the False color composite image output with an input of the original image.

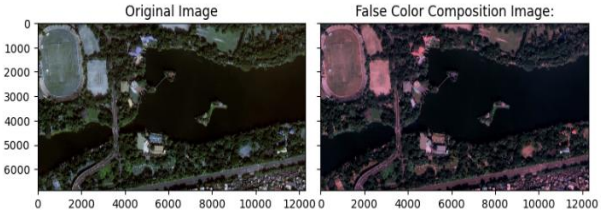
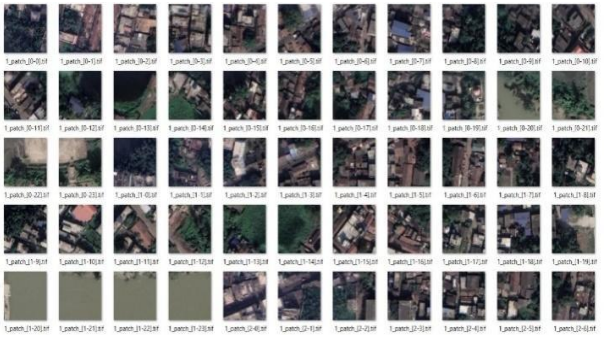
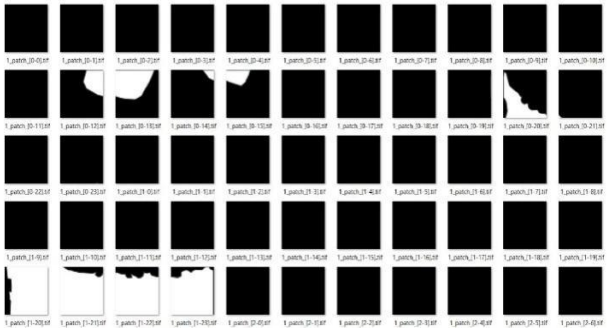



Fig. 10. Original image and false color composite image.

When training any deep learning algorithm, small images produce better results allowing more accuracy without loss of information. Here the images and masks are patched from 12289×6874 to 512×512 pixels using splitting through the patchify method as shown in Fig. 11.



(a)



(b)

Fig. 11. (a) Splitted Images (512×512) and (b) Splitted Masks (512×512).

For water body segmentation, the inputs into the training U-Net model include both labeled data and unlabeled data. After successful execution of the inputs through the model, it produces outputs which include masked water bodies in the images. Fig. 12(a), (b) and (c) shows the test input Image, testing label and predicted label for test image. False positive rate for segmentation of water bodies was around 2.22%.

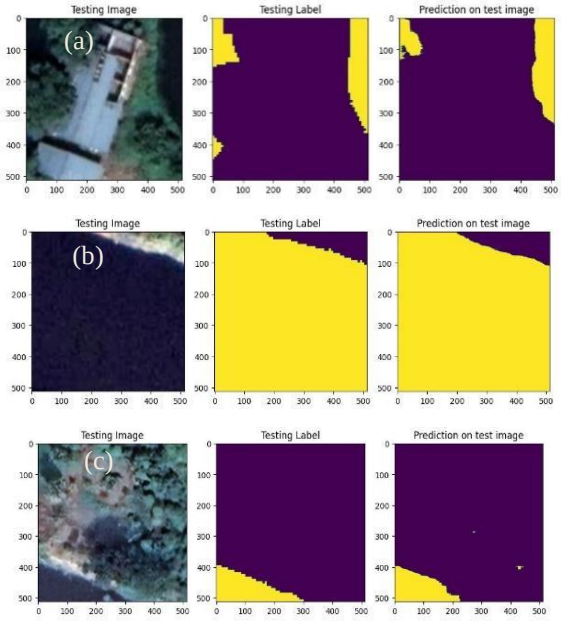



Fig. 12. (a) Testing image, (b) Testing mask, and (c) Predicted mask.

The model is now tested with mean intersection over union metric mean IOU.

$$Mean\ IOU = 0.9150$$

$$Accuracy = 96.44\%$$

After predicting the output for images of size 512×512, now we will predict the output for large images of size 12289×6874. Initially we split the image into overlapping patches of size (512×512). Apply the model to square patches of the image, and overlap the predictions to merge them. Finally, we get a predicted Output image of size 12289×6874. Fig. 13 represents the predicted output for large images.

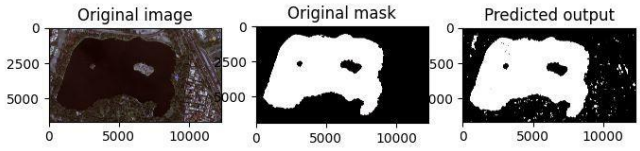
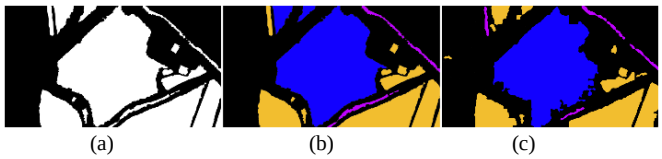



Fig. 13. Original image, original mask, Predicted Output for large image.

After the water body has been segmented,12 shape features are extracted using contours. Using the trained Random Forest model, water bodies are classified into 5 classes i.e., rivers, lakes, ponds, canals and other water bodies. Based on the input calculated features, the RF classifier labels each water body. Fig. 14(a), (b), (c) represents the predicted large image, Expected Output and predicted output.



(a) (b) (c)

Fig. 14. (a) Predicted large image, (b) Expected output, and (c) Predicted output after classification.

A confusion matrix is a performance evaluation tool that compares actual and expected values to determine the accuracy of a classification model. Table II represents the Confusion matrix of 152 water bodies collected from the study area.

TABLE II. CONFUSION MATRIX FOR 152 WATER BODIES IN OUR STUDY AREA

| Predicted | Actual | | | | |
|---|---|---|---|---|---|
| | River | Pond | Canal | lake | Other |
| **River** | 1 | 0 | 0 | 0 | 0 |
| **Pond** | 0 | 70 | 10 | 5 | 0 |
| **Canal** | 0 | 3 | 10 | 0 | 0 |
| **Lake** | 1 | 0 | 0 | 8 | 0 |
| **Other** | 0 | 30 | 1 | 0 | 13 |
| **Total** | 2 | 103 | 21 | 13 | 13 |

Average False Positive rate for classification of water bodies is 11.18%.

The misclassifications in the water bodies are mainly due to two reasons:

- Some ponds were inaccurately detected due to segmentation errors and had an area less than the threshold for pond classification. As a result, they are misclassified to other water bodies. So the false positive rate for other water bodies is high.
- Due to bridges that divide the canal into two sections, canals are misclassified as ponds. As a result, the area of canals is reduced, so they are misclassified as ponds. As a result, the false positive rate for ponds is major.

During the training and testing phase, we provided the shape features csv file, which were calculated from the original mask images, as input. During the validation phase, we entered shape features generated from the expected mask image of the U-Net model. Table III presents the performance of the proposed model during training, testing and validation phases.

TABLE III. COMPARISON OF PERFORMANCE IN DIFFERENT PHASES

| Metric | Training | Testing | Validation |
|---|---|---|---|
| Accuracy | 100% | 84% | 67.1% |
| Micro-averaged Precision | 100% | 84% | 67.1% |
| Micro-averaged Recall | 100% | 84% | 67.1% |
| Micro-averaged F1-score | 100% | 84% | 67.1% |
| Macro-averaged Precision | 100% | 89% | 75.5% |
| Macro-averaged Recall | 100% | 74% | 65.4% |
| Macro-averaged F1-score | 100% | 79% | 63.6% |

Fig. 15 presents an image from the 2020 dataset along with the predicted binary mask generated after post processing. Some segmentation errors can be found in the output of the U-Net model. To avoid them, we perform post-processing. It involves morphological operations such as Erosion, Dilation and area thresholding. We will first use erosion to shrink the boundary of segmented water bodies. Then, we will use area thresholding to rectify the mistakenly classifying non-water bodies as water bodies. Finally, we will use dilation to expand the boundary of the segmented water bodies.

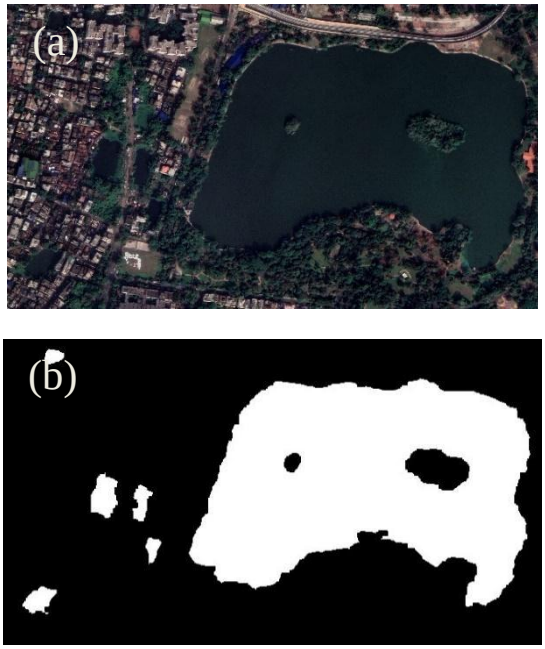



Fig. 15. (a) image from 2020 and (b) binary mask of water bodies after post-processing.

Now, we have taken cartosat-3 images for validation purpose. Fig. 16 presents a cartosat-3 image from the 2023 dataset along with the predicted binary mask generated. This output made it easier to perform change detection from the bi-temporal images.

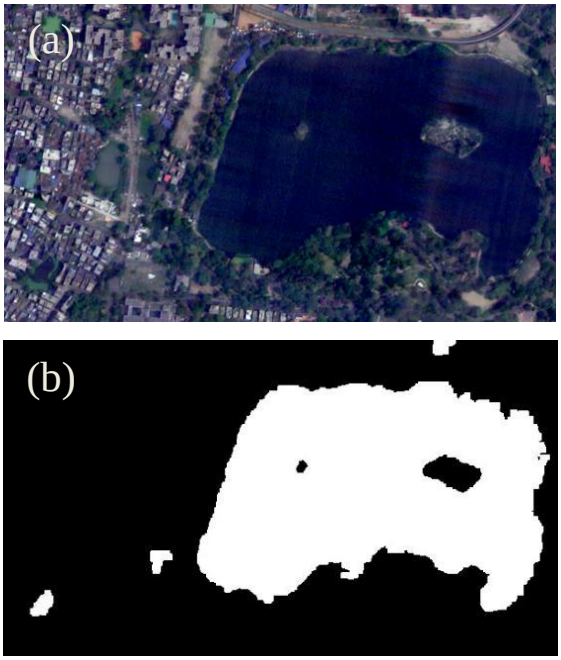



Fig. 16. (a) cartosat-3 image from 2023, (b) binary mask of water bodies after post-processing.

Fig. 17 shows three images, each as a GeoJSON layer. (1) indicates the water bodies that have remained unmodified during the specified time period. (2) indicates water bodies that have recently evolved throughout the historical period, while (3) symbolizes water bodies that have been extinct.

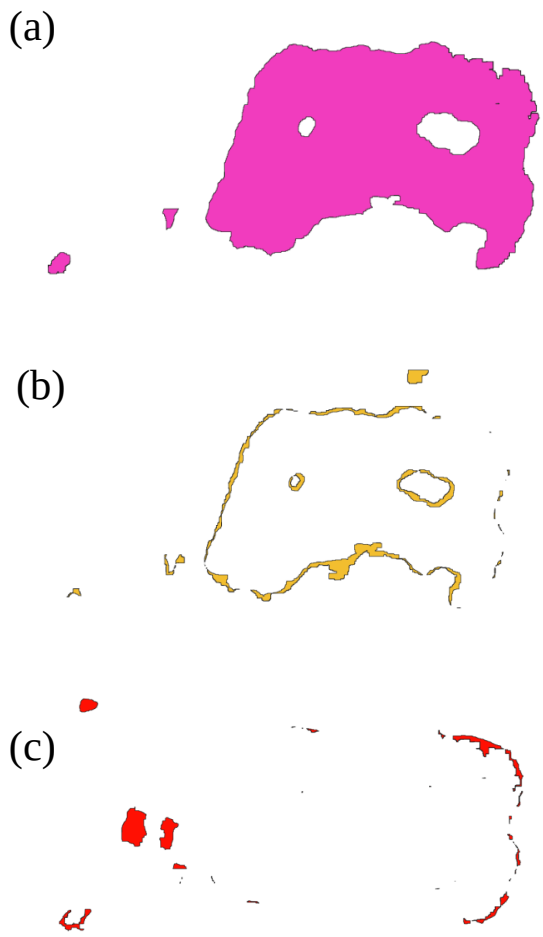



Fig. 17. (a) Unchanged Water bodies, (b) Newly emerged water bodies, and (c) Extinct water bodies.

Fig. 18 depicts the updating of GIS maps in QGIS software. To identify changes in water bodies, several set operations are done on the vector water body segmented layers 2020 GEF and 2023 GEF. The colour 'yellow' indicates freshly formed water bodies, 'pink' represents stable water bodies, and 'red' represents extinct water bodies.

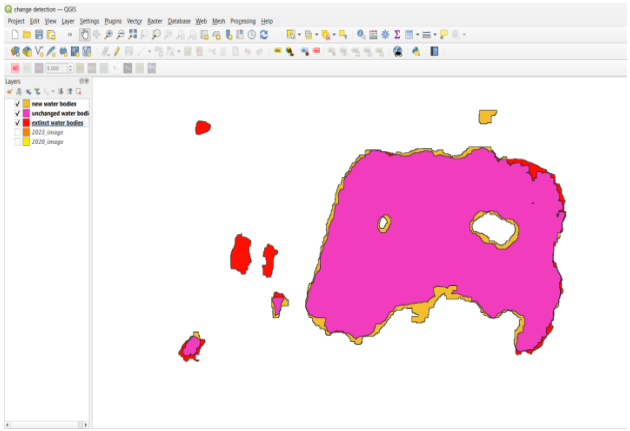


Fig. 18. Updated GIS map in QGIS.

Fig. 19 depicts the appearance of certain misclassification in the water bodies. This is due to the usage of VHRS Images, which necessitates a large amount of processing resources to train the model. Better results would be obtained by using greater computer power and training the model over longer epochs.

This graph represents the model accuracy graph for training and validation data using ReLU activation. The U-Net model was executed with ReLU activation as hidden activation along with Sigmoid as output activation. The model accuracy improved very well with the increase in the number of epochs.

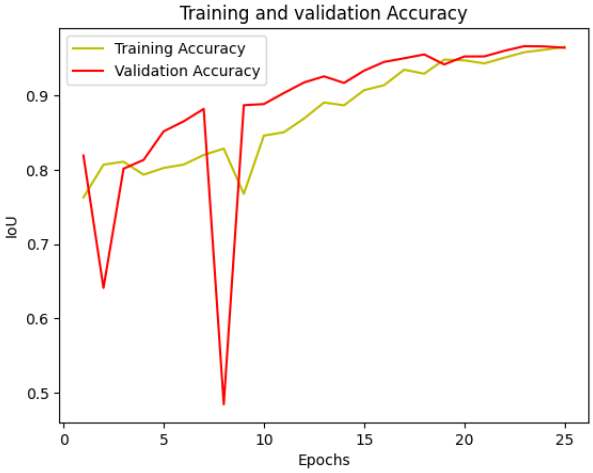



Fig. 19. Model accuracy vs Epochs.

Fig. 20 represents the model loss graph for training and validation data using ReLU activation. The loss also kept on minimizing with the epochs which started giving accurate results. The training and validation loss is almost the same after 15 epochs means that the model has stopped learning and has reached a state of convergence.

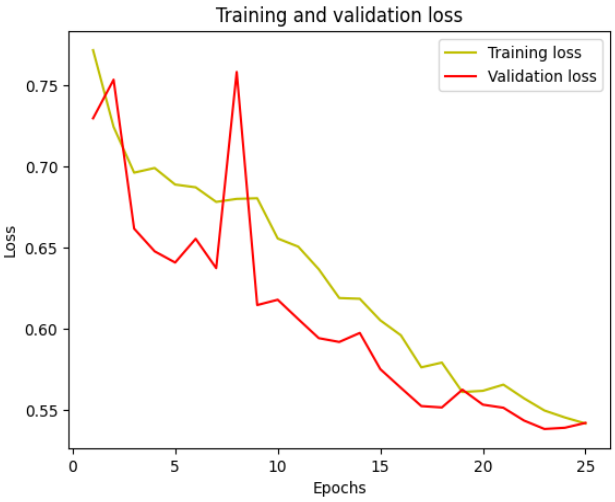



Fig. 20. Model loss vs Epochs.

Table IV represents the comparison with various existing other models for detection and classification of water bodies on different datasets.

TABLE IV. COMPARISON WITH OTHER MODELS

| Study | Methodology | Accuracy Test | Accuracy | Dataset |
|---|---|---|---|---|
| [11] | Random Forest | Accuracy | 82% | Vietnam dataset |
| [15] | Random Forest | F1 Score | 86% | China dataset |
| [26] | Unet with tensor flow | Accuracy | 88% | Chandigarh dataset |
| Proposed Model | Modified Unet model | Accuracy | 96.44% | Kolkata dataset |
| Proposed model | Random Forest | Accuracy | 67.1% | Kolkata dataset |

The proposed model performs better than the model proposed in the previous study [26] that used the U-Net architecture on an Indian dataset specific to Chandigarh. Our analysis demonstrates higher accuracy and relevance for Indian water body segmentation, showing the superiority of their approach. In particular, the proposed U-Net model consistently achieved higher accuracy compared to the model in [26].

## V. CONCLUSION AND FUTURE WORK

Natural resource conservation is not only a component of our daily lives; it is also a responsibility we have to the world and future generations. This work involves the detection and classification of water bodies using a U-Net model and a Random Forest model. Through the proposed model water bodies are extracted from satellite images for a dataset which has a resolution of 0.5 meter/pixel. The region chosen for this work was Kolkata, West Bengal. A total of 1333 images of 512×512 size are collected from the study area. The U-Net model was trained on preprocessed data which includes bilateral filtering for noise removal, false color composition for image enhancement. The accuracy of the U-Net model is around 96.44%. After segmentation of water bodies, shape features of individual water body are calculated using contours. The shape features are given as input to the Random Forest Classifier. The RF model classifies the water bodies into rivers, ponds, lakes, canals and other water bodies. The accuracy of Random Forest model is around 67.1%. This misclassification is due to segmentation errors where some of ponds are misclassified into other water bodies. Canals are misclassified into ponds due to bridges that divide the canals. Some of the morphological operations such as Dilation, Erosion and Area thresholding has been applied on the predicted mask as part of post-processing which improved the overall accuracy of the model. Change detection techniques are employed to compare the GeoJSON files representing different time periods, allowing the identification of newly emerged water bodies, extinct water bodies, and unchanged water bodies.

Future work mainly consists of improving the accuracy of the U-Net model and reducing the misclassification of water bodies.

## CONFLICT OF INTEREST

The authors declare no conflict of interest.

## AUTHOR CONTRIBUTIONS

S. Vasavi conducted conceptualization, methodology, validation, reviewing and editing; Ch Venkata Kalyan did original draft preparation, visualization, validation. V Akhila Sree Rajeswari prepared original draft, and did visualization, validation. All authors had approved the final version.

## REFERENCES


[1] R. Li *et al.*, "Multiattention network for semantic segmentation of fine-resolution remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–13, 2021.

[2] A. Dabija *et al.*, "Comparison of support vector machines and random forests for corine land cover mapping," *Remote Sensing*, vol. 13, no. 4, 777, 2021.

[3] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," arXiv reprint, arXiv:1505.04597, 2015.

[4] J. Li *et al.*, "Satellite detection of surface water extent: A review of methodology," *Water*, vol. 14, no. 7, 1148, 2022.

[5] C. Cui *et al.*, "Detection and classification of multiple power quality disturbances using stockwell transform and deep learning," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–12, 2022.

[6] T. M. Mitchell, *Machine Learning*, 1st ed. McGraw Hill Education, 1997.

[7] J. Zhang *et al.*, "Robotic grasp detection based on image processing and random forest," *Multimedia Tools and Applications*, vol. 79, pp. 2427–2446, 2020.

[8] Y. Liu, P. Dong, J. Ma, P. Li, and R. Liu, "Automatic water area extraction and shape feature analysis from remote sensing images," *Remote Sensing*, vol. 11, no. 3, 282, 2019.

[9] P. H. M. Ananias *et al.*, "ABD: A machine intelligent-based algal bloom detector for remote sensing images," *Software Impacts*, vol. 15, 100482, 2023.

[10] C. Wang *et al.*, "Data acquisition for urban building energy modeling: A review," *Building and Environment*, vol. 217, 109056, 2022.

[11] Y. Liu *et al.*, "Vietnam wetland cover map: Using hydro-periods Sentinel-2 images and Google Earth Engine to explore the mapping method of tropical wetland," *International Journal of Applied Earth Observation and Geoinformation*, vol. 115, 103122, 2022.

[12] Z. Ji *et al.*, "Large-scale extraction and mapping of small surface water bodies based on very high-spatial-resolution satellite images: A case study in Beijing, China," *Water*, vol. 14, no. 18, 2889, 2022.

[13] H. Tang *et al.*, "Large-scale surface water mapping based on landsat and sentinel-1 images," *Water*, vol. 14, no. 9, 1454, 2022.

[14] Y. Hou *et al.*, "Improving satellite retrieval of coastal aquaculture pond by adding water quality parameters," *Remote Sensing*, vol. 14, no. 14, 3306, 2022.

[15] Y. Li and Z. Niu, "Systematic method for mapping fine-resolution water cover types in China based on time series Sentinel-1 and 2 images," *International Journal of Applied Earth Observation and Geoinformation*, vol. 106, 102656, 2022.

[16] A. Hardy *et al.*, "Automatic detection of open and vegetated water bodies using Sentinel 1 to map African malaria vector mosquito breeding habitats," *Remote Sensing*, vol. 11, no. 5, 593, 2019.

[17] F. Mohammadimanesh *et al.*, "An efficient feature optimization for wetland mapping by synergistic use of SAR intensity, interferometry, and polarimetry data," *International Journal of Applied Earth Observation and Geoinformation*, vol. 73, pp. 450–462, 2018.

[18] X. Yang *et al.*, "Mapping of urban surface water bodies from Sentinel-2 MSI imagery at 10 m resolution via NDWI-based image sharpening," *Remote Sensing*, vol. 9, no. 6, 596, 2017.

[19] X. Huang *et al.*, "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*, vol. 8, no. 5, pp. 2097–2110, 2015.

[20] J. Patterson and A. Gibson, "Deep learning: A practitioner's approach," *OReilly*, 2017.
[21] A. Boyat and B. K. Kumar, "A review paper: Noise models in digital image processing," *Signal & Image Processing: An International Journal (SIPIJ)*, vol. 6, no. 2, 2015.
[22] M. Mafi *et al.*, "A comprehensive survey on impulse and Gaussian denoising filters for digital images," *Signal Processing*, vol. 157, pp. 236–260, 2019.
[23] S. Cao *et al.*, "Multi-level monitoring of three-dimensional building changes for megacities: Trajectory, morphology, and landscape," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 167, pp. 54–70, 2020.
[24] Z. Xie *et al.*, "Application of deep learning techniques in water level measurement: Combining improved SegFormer-UNet model with virtual water gauge," *Applied Sciences*, vol. 13, no. 9, 5614, 2023.
[25] K. N. Chaudhury and K. Rithwik, "Image denoising using optimally weighted bilateral filters: A sure and fast approach," in *Proc. 2015 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2015.
[26] N. S. Reddy *et al.*, "Water bodies detection and classification from VHRS images," in *Proc. 2022 International Conference on Futuristic Technologies (INCOFT)*, IEEE, 2022.