





Article

ECDSA-Based Water Bodies Prediction from Satellite Images with UNet

Anusha Ch ¹, Rupa Ch ^{1,*} , Samhitha Gadamsetty ¹, Celestine Iwendi ² , Thippa Reddy Gadekallu ^{3,*}  and Imed Ben Dhaou ^{4,5,6} 

¹ Department of Computer Science and Engineering, VR Siddhartha Engineering College, Vijayawada 520007, India; anusha3619@gmail.com (A.C.); samhitha126@gmail.com (S.G.)

² School of Creative Technologies, University of Bolton, Bolton BL3 5AB, UK; celestine.iwendi@ieee.org

³ School of Information Technology and Engineering, Vellore Institute of Technology, Vellore 632014, India

⁴ Department of Computer Science, Hekma School of Engineering, Computing, and Informatics, Dar Al-Hekma University, Jeddah 22246-4872, Saudi Arabia; imed.bendhaou@utu.fi

⁵ Department of Computing, University of Turku, 20500 Turku, Finland

⁶ Higher Institute of Computer Sciences and Mathematics, Department of Technology, University of Monastir, Monastir 5000, Tunisia

* Correspondence: rupamtech@gmail.com (R.C.); thippareddy.g@vit.ac.in (T.R.G.)

Abstract: The detection of water bodies from satellite images plays a vital role in research development. It has a wide range of applications such as the prediction of natural disasters and detecting drought and flood conditions. There were few existing applications that focused on detecting water bodies that are becoming extinct in nature. The dataset to train this deep learning model is taken from Kaggle. It has two classes, namely water bodies and masks. There is a total of 2841 sentinel-2 satellite images with corresponding 2841 masks. Additionally, the present work focuses on using UNet, Tensorflow to detect the water bodies. It uses a Nadam optimizer to reduce the losses. It also finds best-optimized parameters for the activation function, a number of nodes in each layer. This proposed model achieves integrity by embedding a security feature Elliptic Curve Digital Signature Algorithm (ECDSA). It generates a digital signature for the predicted area of water bodies which helps to secure the key and the detected water bodies while transmitting in a channel. Thus, the proposed model ensures the performance accuracy of 94% which can also work the same for edge detection, detection in blurred and low-resolution images. The model is highly robust.

Keywords: water bodies; satellite imagery; UNet; tensorflow; ECDSA; digital signature



Citation: Ch, A.; Ch, R.; Gadamsetty, S.; Iwendi, C.; Gadekallu, T.R.; Dhaou, I.B. ECDSA-Based Water Bodies Prediction from Satellite Images with UNet. *Water* **2022**, *14*, 2234. <https://doi.org/10.3390/w14142234>

Academic Editor: Thomas Meixner

Received: 13 June 2022

Accepted: 8 July 2022

Published: 15 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Extracting and detecting natural resources such as water bodies and forests from satellite imagery had a huge significance in the recent past. Exploitation of these water bodies is a huge complication. Water bodies need to be monitored periodically for their sustainable development.

Monitoring these water bodies has a large range of applications such as detecting drought and flood conditions and disaster management.

Satellite images have become a major source for detecting traffic, vegetation, buildings, water bodies and ships. This study mainly concentrates on detecting the water bodies from huge dataset of satellite images which were sourced from Kaggle. A deep learning approach of fully convolution network of UNet for semantic segmentation is used. There were many existing neural networks which tried to act as an image segmentation tool, but UNet does its best among all other existing approaches with lowest information loss [1].

This work proposes an implementation of U-Net with the help of tensor flow from scratch. Pooling is specifically embedded to minimize the number of arguments. The use of Nadam optimizer made the model run fast and increase the performance significantly in Training Phase. Nadam is an improved version of this Adam Optimization Technique [2].

The proposed model also ensures integrity by applying ECDSA (Elliptic Curve Digital Signature Algorithm) to the prediction mask of Water Bodies. ECDSA is used in many secure applications and is treated as the crucial factor for maintaining Bitcoin Security [3]. The feature that makes ECDSA more popular among its peers is that it uses short key lengths, which were derived from ECC and provides high security. This model can also be applied to detect the AquaTrash from WaterBodies by slightly changing the parameters for object detection. Thus, the model has huge range of applications in maintaining sustainable environment.

The rest of the paper is coordinated as follows: Section 2 illustrates the literature survey, Section 3 outlines the proposed methodology continued by result and analysis, and the concludes with the conclusion in Section 5. Figure 1 represents the architecture diagram of U-Net where in grey boxes represents the multi channel feature map and total number of channels are shown on the upper top of the rectangular box and unfilled boxes show the copied phenomena of feature maps.

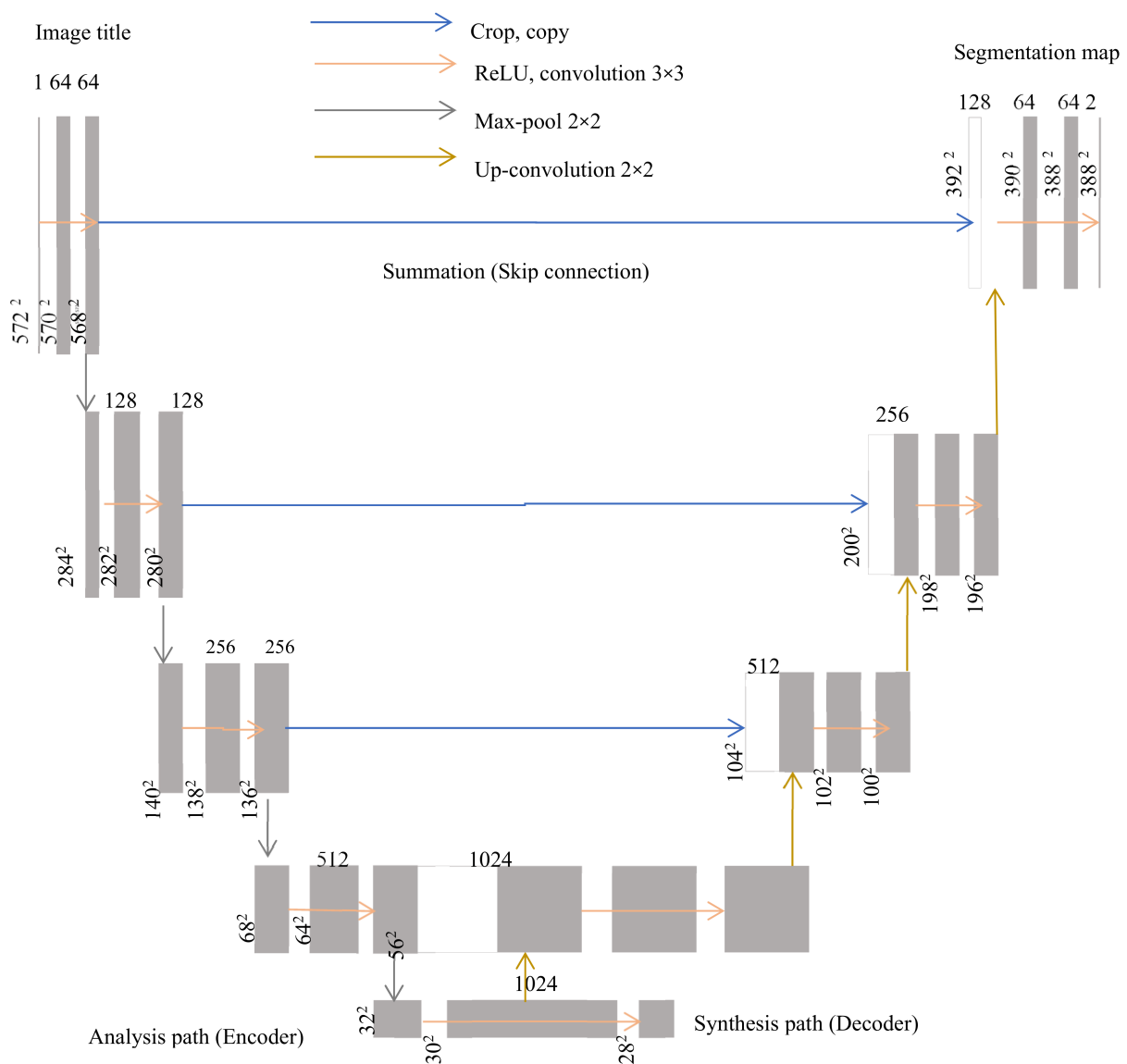


Figure 1. Architecture of U-Net.

2. Literature Survey

Teng Fei et al. [4] build up a model to detect small water bodies and quality analysis in Wuhan. The model is based on image classification and band analysis. A semi-empirical

deep learning method is used for quality analysis. Accuracy assessment is done at regular stages which are then concluded by the inversion of water quality as it is completely based on inversion model. The pitfalls in this analysis are that detection of water bodies is difficult in the case of mixed images.

Jorge Lira et al. [5] propose a mechanism for a large number of issues in the field of remote sensing. This model uses mechanisms such as segmentation and optical reflectance's. A special model named PCA is used to segment the water bodies from satellite images. Clustering process is used to divide the remote sensing satellite image into two based on pixels. The first segment is the water body and the next segment is rest of the satellite image. The flaws in the above mechanism is that False Overlay is observed at some parts of detection.

Fengyu Yang et al. [6] recommends a mechanism for segmentation of water bodies from remote sensing images using mask R-CNN. The proposed system overcomes the difficulties of having low resolution and pixel clarity in remote sensing images. The model was developed based on the imagery obtained from Google Earth. Fine-tuning of the dataset is obtained by Data Augmentation. The model was trained by using both ResNet-50 and Resnet-101 mechanisms. The supremacy of the work is (i) The accuracy of prediction is virtuous and (ii) the model is highly feasible and flexible.

Debashish Chakraborty et al. [7] put forward an approach that uses Holder exponents and variance for segmentation of remote sensing imagery. The main debate is that neither pixel-based nor texture-based segmentation techniques can perform well in yielding satisfactory result. Segmented image mask is used to derive textured and non-textured regions of an image. Later, ISODATA classification algorithm is applied to sourced data. The outputs of both classified and non-classified pixels are merged in the end to obtain a final output image. The method has its advantages by (i) classifying images with both textured and non-textured background and (ii) effectively classifying panchromatic spatial resolution images. The pitfall is that mutli-scale classification of textured background is not performed.

Gordana Kaplan et al. [8] presents a mechanism that combines the use of pixel-based index and object-based method on a multispectral image containing 13 bands. The model uses Sentinel-2 images, each having 10m to 60m resolution. The model considers an urban and mountainous range for better understanding the performance. The indexes are obtained using NWDI. Additional methods are added using a simple pixel-based method, which results in having low albedo. The added advantages of the model are (i) water monitoring in case of adverse climatic conditions and varied regions is accurate and (ii) water extraction was improved by the combined usage of pixel- and object-based classification. The model has its disadvantage is that it uses only two bands out of the available 13 bands.

R. Lalchhanhima et al. [9] proposed a method that uses SAR images which are not available in abundance and consists of speckle noise image segmentation is done using hybrid CNN considering U-Net and Inception. The result of the model in different variations is mainly dependent upon training time, size of dataset and accuracy, among others. Higher kernels increase training time. The output image is given to Otsu's threshold method to obtain 128X128 size binary image. The model shows dominance by (i) providing superior performance to other CNN methods and (ii) requires less training data and less training time to provide high performance.

Zhili Zhang et al. [10] put forward the challenges faced in accurately identifying the boundaries of water bodies due ariety of criteria. To overcome these challenges, a novel combination and multi-feature extraction are presented. Three feature extraction sub modules are used for varying spaces and relationships to gain rich feature representation. Concurrently, multi scale prediction fusion module is used for detecting contours. Additionally, encoding–decoding semantic feature fusion module is also used. The advantages are it (i) gains cutting-edge segmentation performance and (ii) the model is robust.

Xiao Yang et al. [11] introduced a mechanism, namely verifiable ECDSA. The main motto is that it can encrypt the produced digital signature using a public key which is provided by the trusted service. This work adds more features to the existing VES scheme. It can cover a huge number of signatures with almost same structures or similar ones under different variants. It was later applied to many crypto currencies. The only limitation of this proposed work is that it needs to store a 270 MB size look-up table for evaluation.

Yuanbo Shnag et al. [12] introduced an improvement for the existing ECDSA and their applications. This work improves the effectiveness and the security of the ECDSA. The work selects the ECDSA algorithm as is has shorter key length and more pros when compared to the RSA. It analyzes different attacks such as a side-channel attack and thereby improves the efficiency. This work has its own advantages by improving the efficiency and also protecting it from various attacks.

Lin et al. [13] proposed a protocol for IOV with a fault-tolerant ECDSA signature. The internet of vehicles is a growing domain these days. At the same time, it is also open to many attacks. Vehicles can communicate with each other, pedestrians. During this transmission there is a possibility where an unauthorized person can look into this personal information. To protect from the following attacks a fast fault tolerant ECDSA protocol is proposed.

Fang Chen et al. [14] proposed a method to detect the buildings using a U-Net. This model helps to detect the buildings from remote sensing images. This is formed by replacing n channel feature maps with a series of smaller networks. Furthermore, to improve the detection performance a boundary loss function is introduced. The main advantages of the model are (i) it can detect the buildings from a granular level, (ii) it reaches state-level accuracies and (iii) it is highly robust.

Serban et al. [15] implemented a model to detect the change in surface water. This case study is taken from a lake named Nuntasi-Tuzla from Romania. The main cause to propose this model is, in the year 2000, the surface water of the lake decreased gradually. The approach is based on water indices, Normalized Difference Vegetation Index (NDVI). At the end of the research, the detected causes for the problem were found. Drought and human intervention were the major causes for the drastic decrease in surface water.

Gadekallu et al. [16] proposed an approach to detect the ships from remote sensing satellite imagery. This is based on a deep learning approach, namely You only look once version 3 (YOLOV3). Several layers were trained to identify the region of detection. Later, SHA-256 is embedded to the region of detection as well as count of ships. This improves the integrity of the developed model. The main perks of the model are (i) it can effectively detect the ships in land-sea interface region and (ii) it works better for all sort of images such as blurred, reversed and small satellite images.

Jatin Sharma et al. [17] identified a case study for photovoltaic power forecasting using LSTM with Nadam Optimizer. The LSTM provides better results for time series data. The work compares the LSTM with two time series and eight different neural network models and optimizers. The accuracy is calculated for each optimizer and compared accordingly. In the end, Nadam with LSTM shows the better accuracy.

Kousalya et al. [18] predicted the best suitable optimizer for CNN models. The work is based on the deep learning models with a huge input data in a nonlinear fashion. The study is based on CNN models, which involves complex tasks such as recognition of speech, classification of pictures and many more. For training, the research uses the input as human emotions. The comparison study is performed for different optimizers, namely Adam, gradient descent and Nadam, of which Nadam provides the best results.

Rupa et al. [19] developed a model to identify the bounding box region from satellite images. It further produces the count of number of ships present in the given satellite image. The detection is made using 106 (53 + 53) layers of YOLOV3 stack. The images are detected from three predictions. The layers are partitioned for detecting smaller, middle shaped and large ships. Utmost care is given to the data preprocessing section to remove duplicate and false images. The limitations of this work is that it is not suitable for video input.

Erfani et al. [20] introduced a study of semantic segmentation for water bodies. This work is based on the ATLANTIS dataset. This can prevent emergencies during floods. The dataset contains several classes on several groups. For detecting under aquatic and non-aquatic regions, a model said to be AQUANET is developed. The best feature of the work its dataset which can cover wide range of classes with different classes of dataset.

Rajyalakshmi et al. [21] presented an approach to detect water bodies from high resolution satellite images. This work uses thresholding techniques. The dataset consists of images with wide range of spectral and temporal characteristics. The unique approach said to be single-band threshold using bilateral filtering is used in this work. The limitation of the work is that the work is limited to only high-resolution satellite images.

Carles et al. [22] introduced a prototype to secure the COVID-19 patient information using ECDSA in the blockchain. Patient data is very crucial for the treatment and diagnosis. There were cases where unauthorized person can intervene to the network and change this crucial data, which may result in the death of the person. Hence, the work proposed an approach to carefully transmit the data through block chain using ECDSA. The advantages observed is that the patient data becomes immutable once the model is applied.

Wang et al. [23] developed a framework for scientific computing of fluid flows using tensorflow. The work uses a graph based tensorflow module. The performance and accuracies are calculated further which exhibited strong scaling upto TPU v3 pod. The model is developed with the TPU platform.

Ikram et al. [24] presented a framework on the top of UNet. Semantic segmentation is handled using the VGG16 encoder. Transfer learning is used to make an effective UNet architecture. The tuned network architecture is used to predict the breast cancer in earlier stages. The developed Unet architecture produces an accuracy of over 80%. The limitation of the model is that the architecture can be further optimized to improve the accuracy.

Jing et al. [25] developed a model based on Mobile-UNet. It is a CNN approach for fabric detection. It can effectively detect end to end fabric defect segmentation. Simple imbalance is reduced via loss function of median frequency. The encoder part is built under MobileNetv2 and five layers of decoders are introduced. To detect segmentation mask softmax layer is used. The main advantages of the model is that it attains state of the art accuracy.

Table 1 specifies the analysis and summary of the literature survey. Several approaches are compared to the proposed method and analysis on different parameters is done.

Table 1. Analysis of Literature Survey.

Source	Methodology	Implemented/ Designed	Security Feature	Attains Integrity	Optimizer Used
Teng Fei [1]	Image Classification, Band Analysis	Designed	No	No	No
Jorge Lira [2]	PCA	Implemented	No	No	No
R.Lalchhanhima [6]	CNN	Implemented	No	No	No
ZhiliZhang [7]	Segmentation	Designed	No	No	No
Fang Chen [11]	U-Net	Implemented	No	No	No
Serban [12]	NDVI	Designed	No	No	No
Kousalya [15]	CNN	Designed	No	No	Gradient Descent
Jing [22]	Mobile-UNet	Implemented	No	No	No
Proposed Method	UNet	Implemented	Yes (ECDSA)	Yes	Nadam

3. Proposed Methodology

The proposed model derives data from the Water Bodies dataset sourced from kaggle. The dataset consists of original images added up with masks for the corresponding original images. The dataset constitutes of only the images and no other parameters. The model is developed in a step-wise fashion following agile software methodology to facilitate

the inclusion of any future advancement along with convenience for maintenance. Seven crucial steps on a high level are deployed for constructing the model.

Initially, the dataset is fed to the model. This dataset acts as fuel for the exploratory data analytics step. Exploratory data analytics means sourcing statistics of the data and is used for censuring any irregularities or commotion, which can result in dropping the performance of the model. This is a precautionary step, done at the beginning to ensure smooth performance of the model. As a part of exploratory data analytics, first, the original images and masks are comparatively visualized. A significant difference is observed in some images. Ground truths seem to be contradicted. Therefore, the images are verified for any rotation or padding. The images are transformed in the data pre-processing step. Several transformation functions are applied on the images. The prior transformation to be done is resizing. Some images having larger dimensions are complex to fit in memory and have enormous difference to other images. Therefore, a threshold of 1500 is applied to both height and width and all those images having dimensions greater than the threshold are rescaled.

The next transformation function applied to the data is mask conversion. When a histogram is plotted for the original images, most of the values are concentrated towards zero. This may be a result of rotation. For those values at the center, the contrast has to be improved. To do this, histogram equalization is used. On the other hand, masks are stored in jpeg format and they have to be separated using a threshold into two discrete values. For this purpose, mask conversion is used. The third transformation function required is scale values. The primary intention is to make the model converge faster. To do this, the original image is divided into the range [0, 1]. The masks are not considered for the range division.

The final transformation function to be applied on the data is padding images. The model accepts arbitrary-sized images, therefore predetermined padding values must not be considered. Depending on the strides, downsamplings must be able to divide the image size. Additional connections between non-consecutive layers and the same size of input and output images are present due to the specificity of the neural net used. Therefore, padding must be applied.

The fourth significant step is training the model using UNet. Compared to other approaches, UNet has higher performance in the segmentation task [26]. UNet is initialized using `he_normal` in addition to `relu` hidden activation and `sigmoid` output activation. Creating model using UNet involves building convolution, encoder and decoder blocks. For the construction of the convolution block, two important factors are considered. First, the input for the convolution layer and the other one is the filter count, which is by default 64. In the proposed model, partial convolution is used. Construction of encoder and decoder is simple. Encoder consumes consecutive inputs from top to bottom. It consists of two convolution layers along with ReLU and their corresponding batch normalization layers. Decoder requires inputs, skip connection and filter count in each block. The given input is up-sampled and the previous input is also concatenated to obtain final skip connection value.

Figure 2 depicts the methodology structure for the model detection of water bodies through UNet and tensor flow using ECDSA. It collects the dataset and follows several steps in the mid way and finally outputs the original image with the predicted mask and with ECDSA Digital Signature.

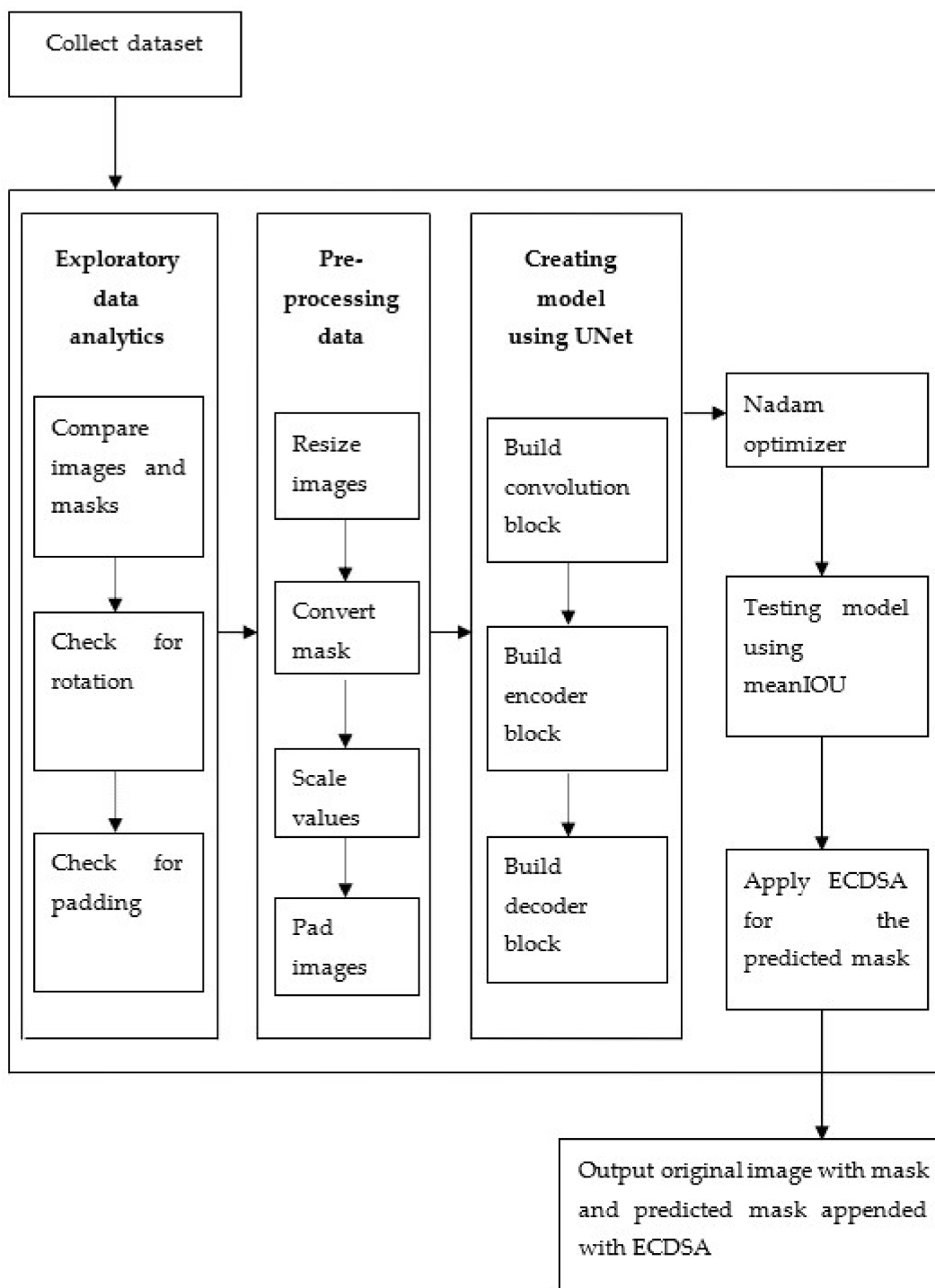


Figure 2. Proposed Architecture.

3.1. Module Wise Approach

3.1.1. Input: Dataset Collection

The input given to the model is the water bodies dataset. The dataset is sourced from the Kaggle with the name Satellite Images of WaterBodies. These images are obtained and captured by sentinel-2 Satellite. The dataset contains two directories. The first one is Images and the other one is Masks. Masks are in black–white where white shows the

presence of water and black shows something other than water. These masks are obtained by Normalized Water Difference Index (NWDI).

$$NWDI = (B3 - B8) / (B3 + B8) \tag{1}$$

where $B3$ = Band3 (Sentinental-2 Green Channel)

$B8$ = Band8 (Sentinental-2 Infrared Channel)

Figure 3 indicates the regions on the sentinel-2 images corresponding to their derived NWDI values.

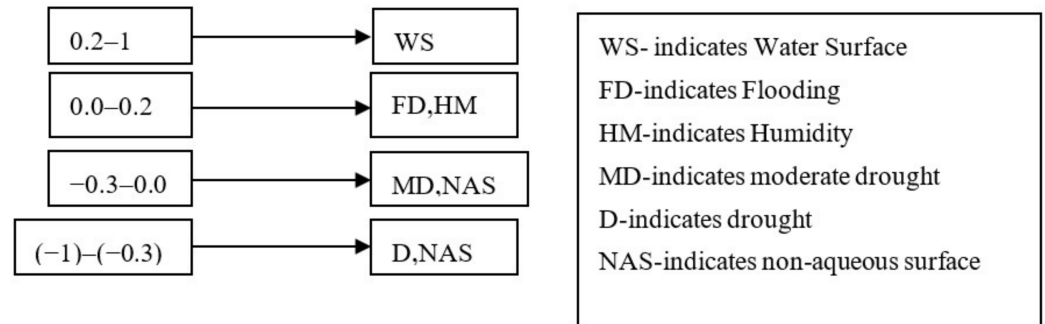


Figure 3. Detections corresponding to NWDI Values.

3.1.2. Approach

Module 1: Exploratory Data Analytics

Data preprocessing is the next step after feeding the input to the model. There are numerous ways to preprocess the image data.

Step 1: Comparing images and masks

Each image is compared with subsequent mask and if there are any duplicates they are removed. They are identified using difference in thresholds.

Step 2: Check for Rotation

Some of the input images are rotated these becomes hectic during training and testing phases. These can be modified using Histogram Equalization. Mathematically, to rotate with an angle θ the coordinates (x, y) are given by,

$$X = \cos \theta \times (x2 - x1) + \sin \theta \times (y2 - y1) \tag{2}$$

$$Y = -\sin \theta \times (x2 - x1) + \cos \theta \times (y2 - y1) \tag{3}$$

The co-ordinates $(x2, y2)$ are to be rotated to a specified angle θ from $(x1, y1)$ becomes (X, Y) .

Step 3: Check for padding

All the images should be of same size while passing to the network and these sizes and the count of downsamplings should be divisible. Hence, the images are divided by their size and masks are separated based on the threshold. Furthermore, target width and height are achieved after processing. For generating an image size of $(m \times m)$,

$$(m + 2pl) \times (m + 2pl) \text{image} \times (fl \times fl) = (m \times m) \tag{4}$$

where, pl = Number of padding layers added or subtracted to make everything into same Size and fl = Image filter.

Module 2: Data Preprocessing

Data preprocessing is the first step before training any model. Only once the data is preprocessed can it pass as an input through the model. This step removes the noise, unwanted and unused data. It helps in maintaining consistency with the dataset.

Step 1: Resizing Images

All images of dataset are resized to same value. Small-size images take less time compared to the larger ones. Hence, the images are resized using `resize` function from `cv2` module.

Step 2: Scale Values

Every image is passed to `mask_split_threshold` function and scales the value between [0, 1].

Step 3: Pad Images

Unnecessary padding bits are removed. During training and testing phases every image must be identical in size to produce better results. Hence, removal of these bits is necessary.

Module 3: Creating model using UNet

UNet model performs well for the image segmentation task. It also works best for binary classification. Multi-channel feature maps are used in the implementation. A series of operations are further applied.

Step 1: Building Convolution Block

Firstly, the kernel gets initialized, later on activation function for output layer is applied. Sigmoid is considered as the output activation. All the parameters such as hidden activation, initialization, output activation need to be defined with appropriate values. Convolution block performs basic operations. There were 64 filters in the convolution layer and the padding remains same.

Step 2: Build Encoder Block

The architecture of encoder takes sequence of input starting from topmost layer to the bottom. It utilizes convolution blocks which is then added to the batch normalization layers. Once input is passed to these layers downsampling and maxpooling are done. The subsequent outputs are passed after these functions gets applied.

$$MPO = \left\lceil \frac{I - W}{ST} \right\rceil + 1 \quad (5)$$

where *MPO* represents max-pooling spatial output, *I* represents the input shape, *W* represents pooling window length and *ST* represents the stride value.

Step 3: Build Decoder Block

Construction of decoder block is simple it takes the input from upper layers and the filters count in a particular cycle. Upsampling is done on the input and filters count. The output is used to execute the convolution block function and the obtained output value is returned.

Module 4: Nadam Optimizer

Once after building the model, Optimizer is invoked. Nadam is sourced from the keras module. Nadam is the best optimization technique, which is an improved version of Adam. Gradient descent is the basic optimization technique further versions like Adam, Nadam were introduced to remove flaws in the Gradient Descent [15]. NAG momentum is used in the Nadam optimizer.

$$X(s) = X(s - 1) - \alpha / \left(\sqrt{nh} + \epsilon \right) \times mh \quad (6)$$

where α represents learning rate, nh represents second moment decay factor, mh represents first moment decay factor.

Module 5: Testing using MeanIOU

The model is now tested with mean intersection over union metric. This is sourced from keras module. It contains three parameters number of classes, name and type. The model shows a value of 0.89 when tested with this metric.

$$MeanIOU = \frac{tp}{tp + fp + fn} \tag{7}$$

tp represents true positives, *fp* represents false positives and *fn* represents false negatives.

The cumulative pixel accuracy for the model is 94.1%. The precision obtained for the proposed model is 0.94. The recall is derived to be 0.939. The detailed derivation of these performance metric values is described under Section 5 Performance analysis.

Module 6: Applying ECDSA

To add integrity to the designed deep learning model Elliptic curve digital signature (ECDSA) is added to the predicted mask. ECDSA shows better performance and highly scalable than other algorithms such as Rivest, Shamir, Adleman (RSA) [27]. It uses keys obtained from elliptic curve cryptography. It does the same work as other digital signature algorithms with shorter key lengths. The output digital signature is obtained by converting signing key to bytes. The signing key is obtained by NIST192p, which is of the length of 24 bytes. Similarly, the verifying key can be obtained from this signing key to verify the digital signature at other end.

Figure 4 shows ECDSA verification steps. The obtained digital signature together with a public key is passed to the verification function and checks for the corresponding results.

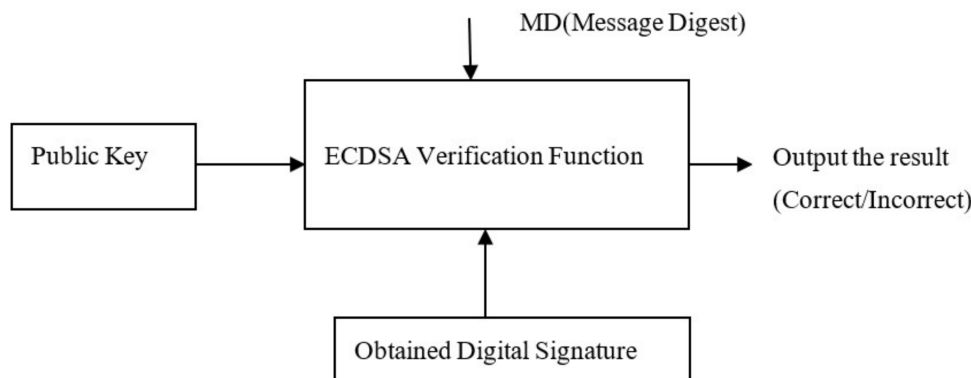


Figure 4. ECDSA verification.

3.1.3. Output

The original image with the mask and predicted mask appended with ECDSA digital signature is observed as the output. Algorithm 1 explains the total process of detection of water bodies. The algorithm takes the input dataset and produces the predicted mask for the given input. It follows several steps such as data preprocessing,

building model, apply optimizer, compiling the model and finally, displaying the output.

Algorithm 1: Water Bodies Detection

Input: Dataset with Water body images and their respective masks

Output: Detects the predicted mask of water body from sentinel-2 satellite image

Process

Step 1: Collects the DataSet

ClassA = WaterBodies images

ClassB = WaterBodies Masks

Step 2: Apply data preprocessing

Step 2a: Resize images

Resizes all the images to same width and using cv2.resize() method

Step 2b: convert the mask

Step 2c: scale the values

Scales the masks using split_threshold from [0, 1]

Step 2d: check for padding

Predetermined padded values are removed

$$p = size_{inp} + 2 \times size_{padding} - (size_{filter} - 1) \quad (8)$$

Step 3: Building model with UNet and Tensorflow

Step 3a: Build Convolution Block

Step 3b: Build Encoder Block

Step 3c: Build Decoder Block

Step 4: Apply Nadam Optimizer

$$Op = tf.keras.optimizers.Nadam() \quad (9)$$

Step 5: Compile the model

Loss = binary cross_entropy

optimizer = Op

Step 6: Check for summary of model

Step 7: Outputs the original image, mask and predicted mask

Algorithm 2 shows the procedure of appending an ECDSA module to the deep learning model. It involves the generation of the signing key, verifying the key and producing a signature using this signing key. The signing key uses default elliptic curve (ECC) known to be NIST192p [28].

Algorithm 2: Appending ECDSA to the predicted Mask

Input: Original image, mask and predicted mask

Output: Detects the predicted mask with ECDSA digital signature

Process

Step 1: Generation of Signing Key from NIST192p default curve (S_{key})

Step 2: Obtain verifying key from the signing key

$$V_{key} = S_{key}.verifying_key() \quad (10)$$

Step 3: Obtain the Signature by signing the co-ordinates of masked region with (S_{key})

$$ECDSA_{signature} = S_{key}.sign(Region\ of\ predicted\ waterbodies\ mask) \quad (11)$$

Step 4: Verify the signature using S_{key} upon $ECDSA_{signature}$

Step 5: Append the signature to predicted mask

Step 6: Outputs the original image, mask and predicted mask with ECDSA digital signature

4. Results

The proposed model is used for the segmentation of water bodies from satellite images and then the detected region is secured using elliptic curve digital signature. Figure 5 represents the sample image displayed along with its corresponding original mask.

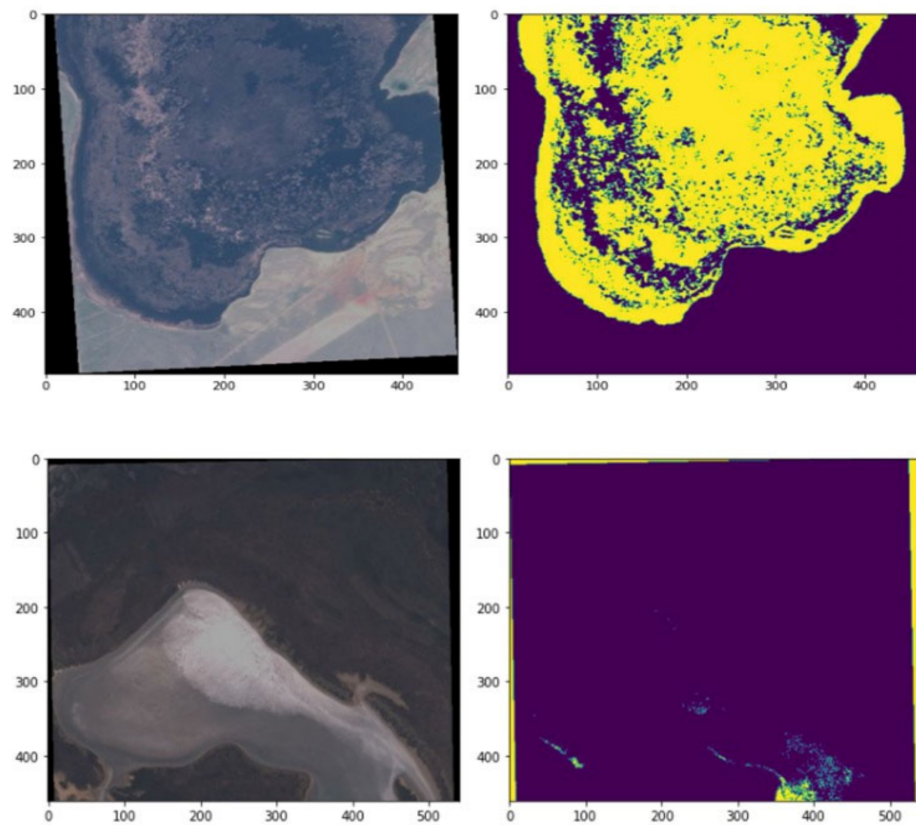


Figure 5. Sample image and corresponding mask.

Figure 6 represents the statistics drawn out of the exploratory data analysis. For the original image, ravel function is applied to transform the data in jpg format to flat list and then histogram is drawn to observe the behavior of the data. It is observed that most of the data is concentrated around zero which can be a result of rotation. For the values concentrated around the center, histogram equalization can be applied to improve the contrast.

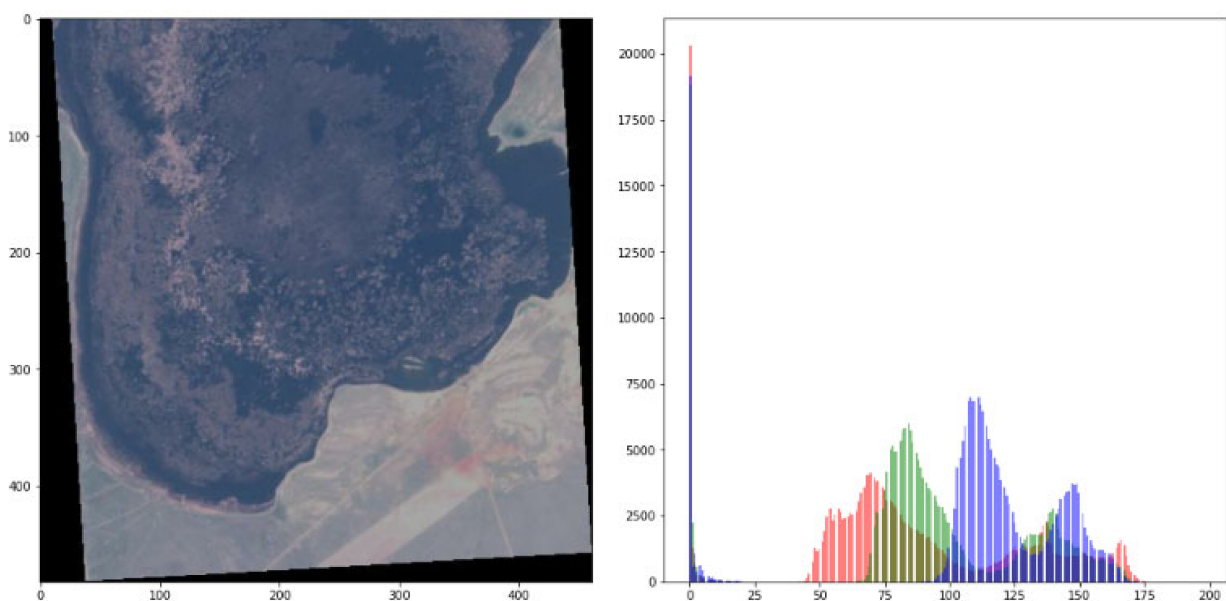


Figure 6. Exploratory data analysis.

Figure 7 represents the summary of the model after training with UNet. The total parameters are 1,925,001, of which all parameters are used for training. The figure clearly depicts the working of the UNet model for current data. The input to each layer, operation and the output are shown.

```

Model: "model"
-----
Layer (type)                Output Shape                Param #    Connected to
-----
input_1 (InputLayer)        [(None, None, None, 8
conv2d (Conv2D)              (None, None, None, 3 896   input_1[0][0]
conv2d_1 (Conv2D)            (None, None, None, 3 9248  conv2d[0][0]
max_pooling2d (MaxPooling2D) (None, None, None, 3 8   conv2d_1[0][0]
conv2d_2 (Conv2D)            (None, None, None, 6 18496  max_pooling2d[0][0]
conv2d_3 (Conv2D)            (None, None, None, 6 36928  conv2d_2[0][0]
max_pooling2d_1 (MaxPooling2D) (None, None, None, 6 8   conv2d_3[0][0]
conv2d_4 (Conv2D)            (None, None, None, 1 73856  max_pooling2d_1[0][0]
conv2d_5 (Conv2D)            (None, None, None, 1 147584  conv2d_4[0][0]
max_pooling2d_2 (MaxPooling2D) (None, None, None, 1 8   conv2d_5[0][0]
conv2d_6 (Conv2D)            (None, None, None, 2 295168  max_pooling2d_2[0][0]
conv2d_7 (Conv2D)            (None, None, None, 2 590336  conv2d_6[0][0]
up_sampling2d (UpSampling2D) (None, None, None, 2 8   conv2d_7[0][0]
conv2d_8 (Conv2D)            (None, None, None, 1 131200  up_sampling2d[0][0]
concatenate (Concatenate)    (None, None, None, 2 8   conv2d_5[0][0]
                                     conv2d_8[0][0]
conv2d_9 (Conv2D)            (None, None, None, 1 295040  concatenate[0][0]
conv2d_10 (Conv2D)           (None, None, None, 1 147584  conv2d_9[0][0]
up_sampling2d_1 (UpSampling2D) (None, None, None, 1 8   conv2d_10[0][0]
conv2d_11 (Conv2D)           (None, None, None, 6 32832  up_sampling2d_1[0][0]
concatenate_1 (Concatenate)  (None, None, None, 1 8   conv2d_3[0][0]
                                     conv2d_11[0][0]
conv2d_12 (Conv2D)           (None, None, None, 6 73792  concatenate_1[0][0]
conv2d_13 (Conv2D)           (None, None, None, 6 36928  conv2d_12[0][0]
up_sampling2d_2 (UpSampling2D) (None, None, None, 6 8   conv2d_13[0][0]
conv2d_14 (Conv2D)           (None, None, None, 3 8224  up_sampling2d_2[0][0]
concatenate_2 (Concatenate)  (None, None, None, 6 8   conv2d_1[0][0]
                                     conv2d_14[0][0]
conv2d_15 (Conv2D)           (None, None, None, 3 18464  concatenate_2[0][0]
conv2d_16 (Conv2D)           (None, None, None, 3 9248  conv2d_15[0][0]
conv2d_17 (Conv2D)           (None, None, None, 1 33   conv2d_16[0][0]
-----
Total params: 1,925,601
Trainable params: 1,925,601
Non-trainable params: 0
    
```

Figure 7. Summary of the model.

Figure 8 shows the original image along with its mask and corresponding mask detected by the proposed model. To get most out of visualization, the images are displayed along with scale for precise understanding.

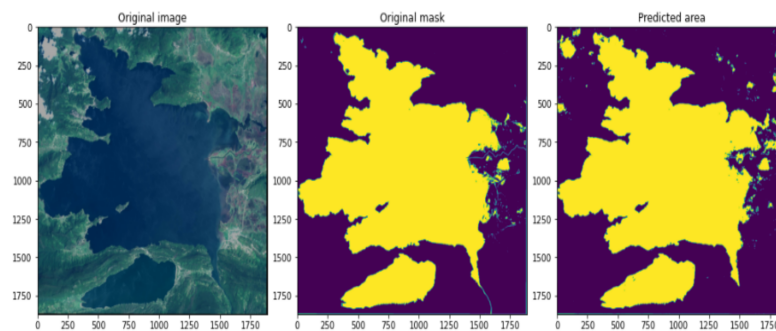


Figure 8. Predicted mask area for the water body.

Figure 9 shows the added advantage of the proposed model achieved by the inclusion of Elliptic curve digital signature. Detection of any tampering along the way of transmission can be done with ease.

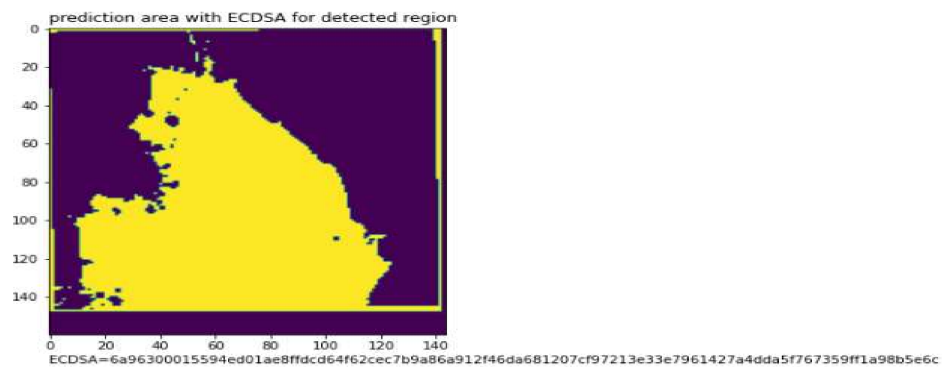


Figure 9. Predicted mask with ECDSA value—Case I.

Figure 10 represents another case of predicted mask for a water body image. The ECDSA function works by constructing a signing key from NIST192p. This signing key is used to obtain a verifying key. By signing the coordinates of the masked region with the signing key, the ECDSA signature is obtained. Then, the ECDSA signature is verified using verifying key. Finally, the ECDSA signature is applied to the predicted mask.

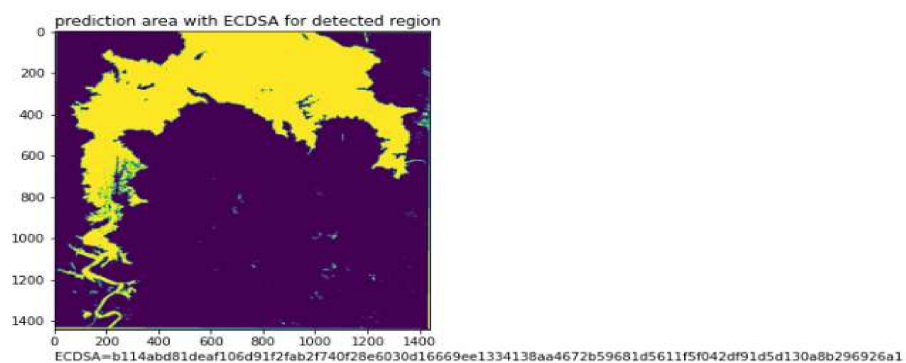


Figure 10. Predicted mask with ECDSA value—Case II.

Table 2 represents Elliptic curve digital signature values for corresponding images. The ECDSA function is applied to the detected mask to achieve integrity. The main advantage of such a feature enables detecting any change in the transmitted data. This is because the hash function, which is also termed as message digest, is one way. While transmitting the mask, the message digest is also concatenated and sent to the destination. In case, there is any modification to the data, it can be verified by checking the hash value at the destination.

Table 2. Elliptic Curve digital signature values for corresponding images.

Image	ECDSA Value
PredictionMask_1.jpg	3yu86fg1vu787gdw3bv34gt564bjw436686473434b3571616c6c70494d2f4c542b456652524d3475v6r49o2s5g7t8u88
PredictionMask_2.jpg	7y8u0p34943494c4c374c476852754f795a58575970475655393850564d73394439394d775330776f5069395477507
PredictionMask_3.jpg	9i5r3t349514441354a643558752f52674663376a4f5375356b6547424a587351664d5174584f4f7355786c753365647
PredictionMask_4.jpg	1y7r4e0o8u6yf5444766b4d2f6141663068686465432b3277574f346f44704b4466554756346f4e77507a62b7iqw7c1
PredictionMask_5.jpg	3e5r9k00b2564b774f48645a42336b6b49506d4b6a6a4e2b6d472b5557734f6531394a4f7a665955536236323954724
PredictionMask_6.jpg	4d455543494356344e4f6430564f7437674439694341676d344f7376454b5a414a2b71303442636138426d6a4c466c
PredictionMask_7.jpg	6y7t9b5v7b0155306f69677851474a555870474e6c65726b4f636d5548475a474a64554b76506c444263554937441
PredictionMask_8.jpg	4r7z8y4944776155306f69677851474a555870474e6c65726b4f636d5548475a474a64554b76506c4442635549374

Figure 11 represents Predicted Masks for water bodies sentinel-2 satellite Images. The ECDSA values for these predicted Masks are present in the Table 2.

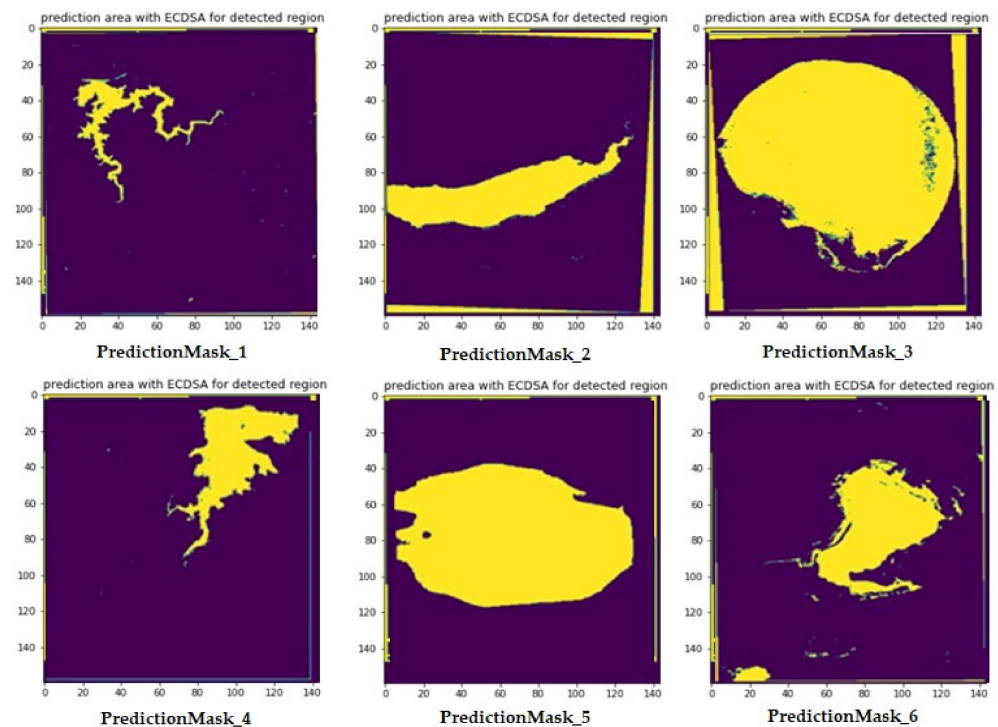


Figure 11. Predicted Masks for different Water bodies satellite Images.

5. Performance Analysis

Performance Analysis is the key factor for any model. It helps to check how good the model shows results. It is also used to compare the model with other existing ones. This section shows the analysis of the water bodies detection model with different metrics.

5.1. MeanIOU

It is a popular evaluation metric used to evaluate segmentation models. It is mainly used to identify the overlap percentage between the original output and predicted mask.

$$MeanIOU = \frac{Overlap_{area}}{Union_{area}} \tag{12}$$

$$MeanIOU = \frac{A \cap B}{A \cup B} \tag{13}$$

The model shows the MeanIOU score of 0.89. Table 3 shows the different metric values for the water bodies prediction model. The score > 0.5 is considered to be a good model.

$$MeanIOU = 93.4 / (93.4 + 5.8 + 6.05) = 0.89$$

Table 3. Internal Metrics in MeanIOU.

Metrics to Calculate MeanIOU	Respective Values
True _{positive}	93.4
True _{negative}	93.6
False _{positive}	5.8
False _{negative}	6.05

5.2. Pixel Accuracy

The pixel accuracy is another evaluation metric where it checks the percentage of the pixel that is correctly predicted. It compares Ground Truth and Prediction. *True_{positive}* represents the prediction is in that certain class and similarly *True_{negative}* indicates the pixel doesn't belong to the wrong class. The cumulative Pixel Accuracy is given by,

$$P_{Acc} = \frac{True_{Positive} + True_{Negative}}{True_{Positive} + True_{Negative} + False_{Positive} + False_{Negative}} \tag{14}$$

The model shows the Cumulative Accuracy of 94.1%.

$$P_{Acc} = (93.4 + 93.6) / (93.4 + 93.6 + 5.8 + 6.05) = 94.1\%$$

5.3. Instance Segmentation

Instance segmentation contains a group of local segmentation masks. IOU for each mask is calculated instead of calculating on a single bounding box in Object detection models. To calculate this metric precision values are obtained for each mask.

5.3.1. Precision Calculation

True positive is seen when truth and predicted pair IOU value exceeds certain threshold. Figure 12 depicts the prediction of TP and FN based on IOU values. Let the IOU threshold be 0.8. TP and FN are observed based on the threshold value.

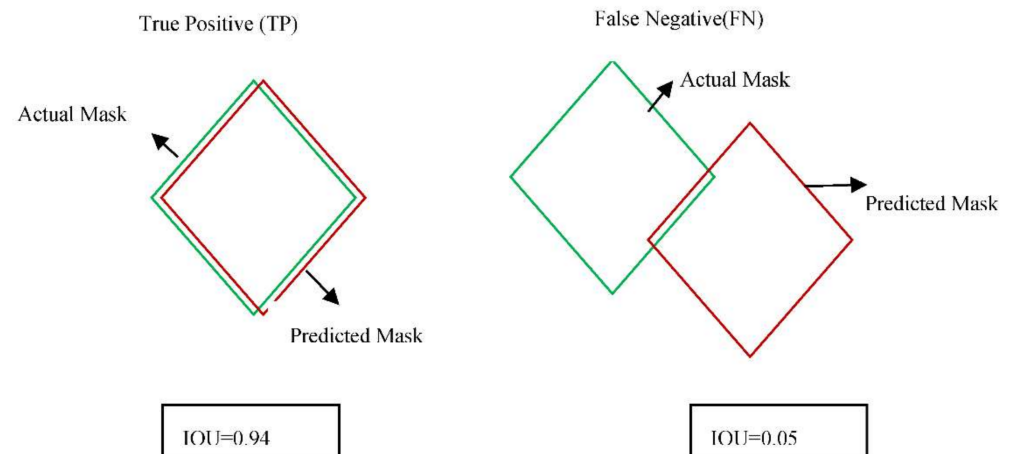


Figure 12. TP and FN prediction based on IOU.

The precision value for the model is 0.94.

$$Pr = \frac{TP}{TP + FP} \tag{15}$$

5.3.2. Recall

It determines total positive predictions in compared with ground truth. To determine precision and recall true positive, they need to be done before hand. To calculate T_{po} , IOU is processed between target and prediction. The recall value for the model is 0.939.

$$R_C = \frac{T_{po}}{T_{po} + F_{Neg}} \tag{16}$$

Figure 13 shows the difference between predicted image threshold to the default threshold.

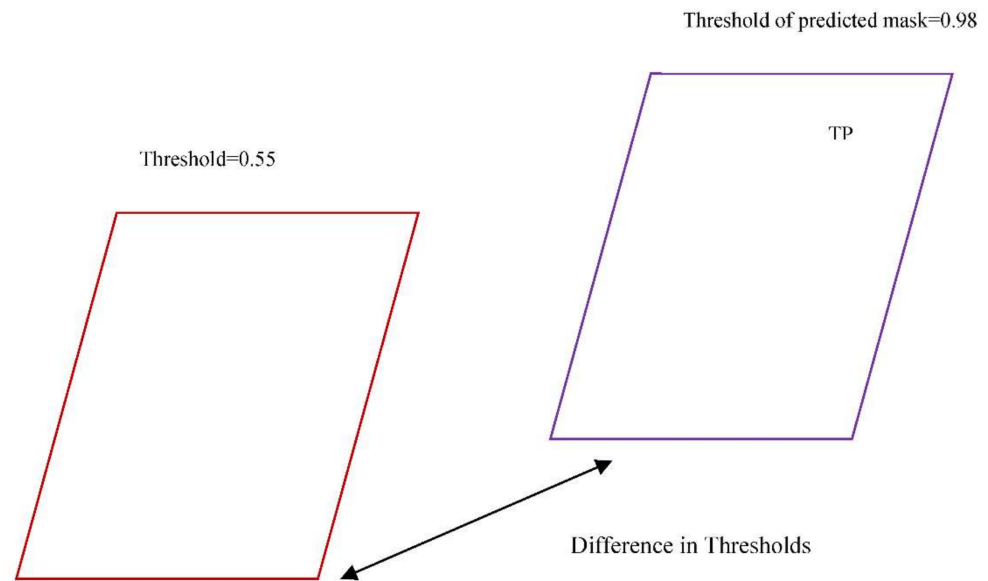


Figure 13. Difference in Thresholds.

5.4. Comparison of U-Net with Other Methods

Table 4 shows the comparison of metrics between the existing method and other methods. The proposed UNet model with tensorflow is compared to Region-Based Convolution Network (R-CNN), Conditional Random Field CNN (CRF-CNN), Efficient Neural Network (ENet) and Semantic Segmentation Model (SegNet).

Table 4. Comparison of UNet with Other Methods.

Method	Total Pixel Accuracy	Recall	Precision	MeanIOU
UNet	94.1	0.939	0.94	0.89
R-CNN	91.4	0.90	0.918	0.84
CRF-CNN	90.3	0.908	0.8989	0.82
ENet	88.3	0.888	0.879	0.792
SegNet	87.5	0.875	0.875	0.7777

Based on the above represented statistics, it is clear that the proposed model shows a higher degree of total pixel accuracy, recall, precession and also MeanIOU, which is the metric that is used for evaluating segmentation rate of a semantic image. This indicates the better performance of the model on a whole.

Figure 14 shows the comparison of the present method with different existing methods. The metrics used for comparison are Recall, Accuracy, Precision and MeanIOU. The X-Axis represents different methods. Y-Axis represents metric values in percentages.

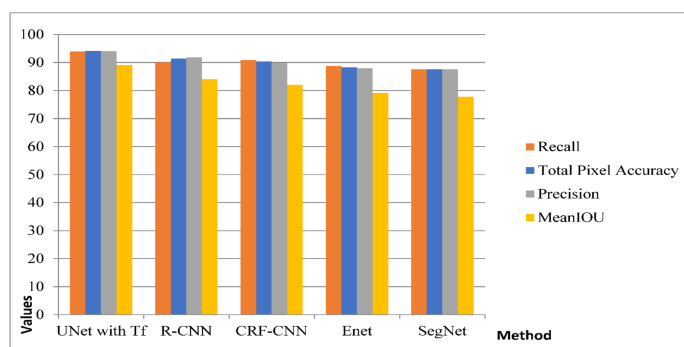


Figure 14. Comparison of Methods with different metrics.

5.5. Comparison between Sentinel-2 and Landsat 8 OLI Datasets

Table 5 shows the deviation in values obtained for the same metrics MeanIOU, Precision, Recall, Pixel Accuracy for the Sentinel-2 and Landsat 8 OLI Datasets.

Table 5. Comparison between sentinel-2 and Landsat 8 OLI.

Dataset	MeanIOU	Precision	Recall	Pixel Accuracy
Sentinel-2	0.89	0.94	0.939	94.1%
Landsat 8	0.88	0.93	0.936	93.3%

5.6. Comparison between Optimizers

Table 6 shows the comparison of different optimizers under different parameters Learning Rate, Test Loss, Test Accuracy and Momentum. The test loss and test accuracy are for the UNet model.

Table 6. Comparison of Different Optimizers.

Optimizer	Learning Rate	Test Loss	Test Accuracy	Momentum
Adam	0.001	0.0467	95.5	GD with Momentum
Nadam	0.001	0.0321	96.2	Nesterov Momentum
Gradient Descent	0.01	0.0433	93.33	No Momentum

From the above drawn table, it can be observed that Gradient descent optimizer has a decline in the learning rate capability compared to others. While Adam and Nadam optimizers have same same learning rate, there is a less test loss and higher test accuracy in the case of the Nadam optimizer. Therefore, the Nadam optimizer was considered the best to be implemented with the model.

Figure 15 shows the Comparison between Adam and Nadam optimizers. The graph shows the Training Loss of two optimizers when executed under different epochs. The x-axis indicates the Iterations and Y-axis indicates the training loss.

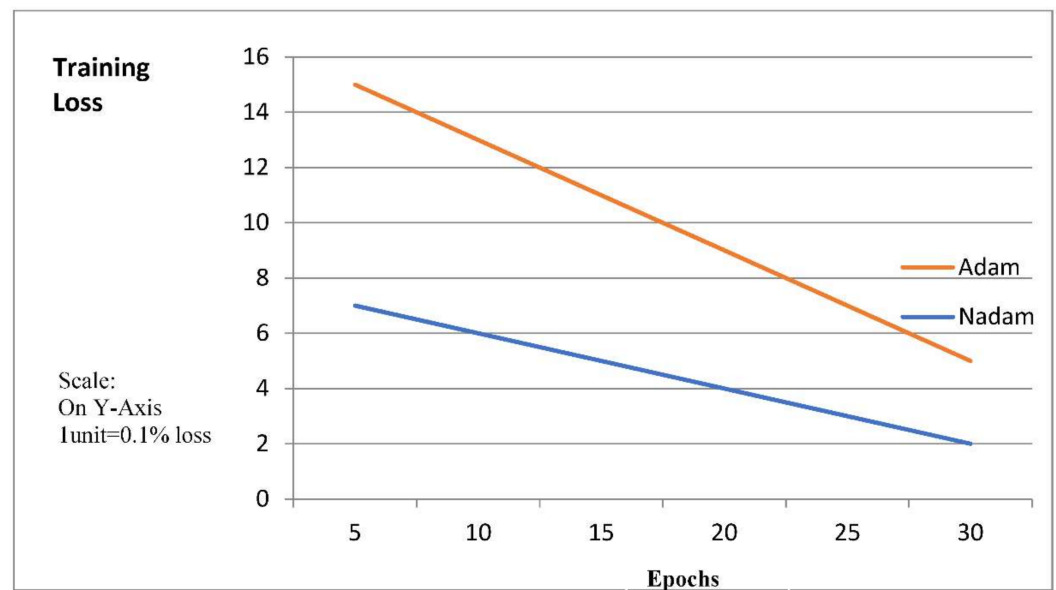


Figure 15. Comparison between different optimizers.

This comparison clearly states that as number of iterations increases the loss is reduced gradually in Nadam as compared to Adam.

5.7. Analysis of Security Algorithms

5.7.1. RSA vs. ECDSA

A major comparison is done between ECDSA and Rivest, Shamir, Adleman (RSA). Although both algorithms use public key cryptography, they are differed by many aspects. RSA has longer keys when compared to ECDSA [29–38]. Signing the message undergoes numerous multiplication and division operation in which ECDSA is faster compared to RSA. Shor’s Algorithm is a popular approach to break keys, but is not practically present. RSA is vulnerable to Shor’s algorithm of breaking keys but it is not possible to break the keys using shor’s in ECDSA.

Table 7 shows the comparison of key sizes between RSA algorithm and ECDSA algorithm with different security levels. It is observed that ECDSA has shorter keys compared to RSA.

Table 7. Comparison of key sizes between RSA and ECDSA.

Level of Security	Key Size of RSA	Key Size of ECDSA	Ratio
80	1024 bits	165–220 bits	6:1
128	3072 bits	256–383 bits	12:1
256	15360 bits	512 bits	30:1

Figure 16 shows the comparison between RSA and ECDSA in terms of CPU Utilization.

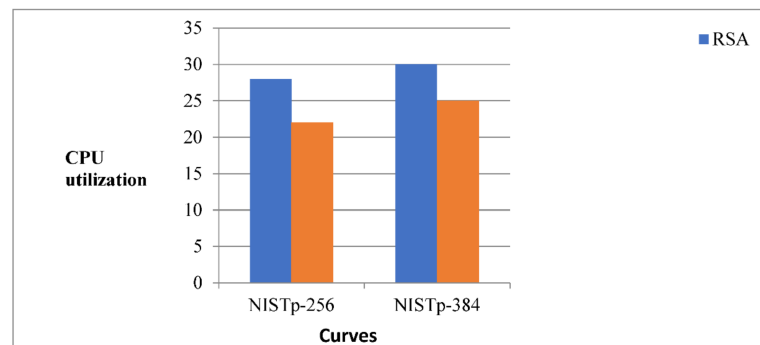


Figure 16. CPU Utilization of RSA and ECDSA.

5.7.2. RSA vs. DSA vs. ECDSA

Figure 17 shows the round Trip Time of RSA, Digital Signature Algorithm (DSA), ECDSA. Round Trip Time (RTT) indicates the total time in which the signature is sent through the channel and time taken to receive the acknowledgement. ECDSA takes less RTT compared to other algorithms.

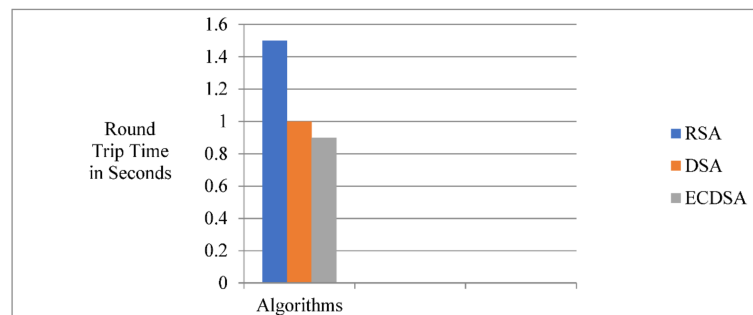


Figure 17. Round Trip Time of RSA, DSA and ECDSA.

6. Conclusions

In this work, a deep learning model is proposed to detect water bodies from satellite Images using UNet and Tensorflow. This model works best even in the case of small and blurred satellite images. Furthermore, it also gives best results in the case of water bodies present on the land interface because of the architecture of UNet. Additionally, the model also uses a best suitable optimization technique (Nadam Optimizer). The computation time is relatively faster using this Nadam Optimizer. Moreover, data analysis is done with keen interest by following the stages like resizing, converting to masks, padding as it defines the perfection and performance of the model. The proposed UNet architecture uses multiple channels and feature maps. It also initiates hidden activation function, output activation function, kernel Initializers. Max pooling is done in the model to reduce the dimensionality of images which increases the performance. The following model detects the prediction area from the satellite images with a higher accuracy compared to other existing ones. It also works best under images with adverse climatic conditions. The superiority of the model is that it uses ECDSA to secure the detected water region and produces the secured signature, such that this water bodies region can be protected from unauthorized person. In future, this model can be extended to video inputs. The detected regions of water bodies can be stored to the cloud.

Author Contributions: Conceptualization, S.G. and R.C.; data curation, S.G.; formal analysis, I.B.D., R.C. and A.C.; funding acquisition, C.I.; investigation, C.I.; resources, R.C. and T.R.G.; software, S.G.; supervision, T.R.G. and I.B.D.; validation, T.R.G.; visualization, A.C.; writing—original draft, S.G., R.C. and A.C.; writing—review and editing, I.B.D., C.I. and T.R.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not Applicable.

Informed Consent Statement: Not Applicable.

Data Availability Statement: The dataset used in this work is available at <https://www.kaggle.com/datasets/franciscoescobar/satellite-images-of-water-bodies> (accessed on 20 January 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ying, L.; Qian Nan, Z.; Fu Ping, W.; Tuan Kiang, C.; Keng Pang, L.; Heng Chang, Z.; Lu, C.; Jun, L.G.; Nam, L. Adaptive weights learning in CNN feature fusion for crime scene investigation image classification. *Connect. Sci.* **2021**, *33*, 719–734. [[CrossRef](#)]
2. Ghaderizadeh, S.; Abbasi-Moghadam, D.; Sharifi, A.; Zhao, N.; Tariq, A. Hyperspectral image classification using a hybrid 3D-2D convolutional neural networks. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 7570–7588. [[CrossRef](#)]
3. Ye, A.; Zhou, X.; Miao, F. Innovative Hyperspectral Image Classification Approach Using Optimized CNN and ELM. *Electronics* **2022**, *11*, 775. [[CrossRef](#)]
4. Wang, L.; Bie, W.; Li, H.; Liao, T.; Ding, X.; Wu, G.; Fei, T. Small Water Body Detection and Water Quality Variations with Changing Human Activity Intensity in Wuhan. *Remote Sens.* **2022**, *14*, 200. [[CrossRef](#)]
5. Lira, J. Segmentation and morphology of open water bodies from multispectral images. *Int. J. Remote Sens.* **2006**, *27*, 4015–4038. [[CrossRef](#)]
6. Yang, F.; Feng, T.; Xu, G.; Chen, Y. Applied method for water-body segmentation based on mask R-CNN. *J. Appl. Remote Sens.* **2020**, *14*, 014502. [[CrossRef](#)]
7. Chakraborty, D.; Singh, S.; Dutta, D. Segmentation and classification of high spatial resolution images based on Hölder exponents and variance. *Geo-Spat. Inf. Sci.* **2017**, *20*, 39–45. [[CrossRef](#)]
8. Kaplan, G.; Avdan, U. Object-based water body extraction model using Sentinel-2 satellite imagery. *Eur. J. Remote Sens.* **2017**, *50*, 137–143. [[CrossRef](#)]
9. Lalchhanhima, R.; Saha, G.; Sur, S.N.; Kandar, D. Water body segmentation of Synthetic Aperture Radar image using Deep Convolutional Neural Networks. *Microprocess. Microsyst.* **2021**, *87*, 104360. [[CrossRef](#)]
10. Hang, Z.; Lu, M.; Ji, S.; Yu, H.; Nie, C. Rich CNN Features for water-body segmentation from very high resolution aerial and satellite imagery. *Remote Sens.* **2021**, *13*, 1912.
11. Yang, X.; Liu, M.; Au, M.H.; Luo, X.; Ye, Q. Efficient Verifiably Encrypted ECDSA-Like Signatures and Their Applications. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 1573–1582. [[CrossRef](#)]
12. Shang, Y. Efficient and Secure Algorithm: The Application and Improvement of ECDSA. In Proceedings of the 2022 International Conference on Big Data, Information and Computer Network (BDICN), Sanya, China, 20–22 January 2022; pp. 182–188. [[CrossRef](#)]
13. Lin, H.Y.; Hsieh, M.Y.; Li, K.C. A Fast Fault-Tolerant Routing with ECDSA Signature Protocol for Internet of Vehicles. In Proceedings of the Sixth International Congress on Information and Communication Technology; Lecture Notes in Networks and Systems. Yang, X.S., Sherratt, S., Dey, N., Joshi, A., Eds.; Springer: Singapore, 2022; Volume 217. [[CrossRef](#)]
14. Chen, F.; Wang, N.; Yu, B.; Wang, L. Res2-Unet, a New Deep Architecture for Building Detection From High Spatial Resolution Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 1494–1501. [[CrossRef](#)]
15. Șerban, C.; Maftai, C.; Dobrică, G. Surface Water Change Detection via Water Indices and Predictive Modeling Using Remote Sensing Imagery: A Case Study of Nuntasi-Tuzla Lake, Romania. *Water* **2022**, *14*, 556. [[CrossRef](#)]
16. Gadamssetty, S.; Ch, R.; Ch, A.; Iwendu, C.; Gadekallu, T.R. Hash-Based Deep Learning Approach for Remote Sensing Satellite Imagery Detection. *Water* **2022**, *14*, 707. [[CrossRef](#)]
17. Sharma, J.; Soni, S.; Paliwal, P.; Saboor, S.; Chaurasiya, P.K.; Sharifpur, M.; Khalilpoor, N.; Afzal, A. A novel long term solar photovoltaic power forecasting approach using LSTM with Nadam optimizer: A case study of India. *Energy Sci. Eng.* **2022**. [[CrossRef](#)]
18. Kousalya, K.; Mohana, R.S.; Jithendiran, E.K.; Kanishk, R.C.; Logesh, T. Prediction of Best Optimizer for Facial Expression Detection using Convolutional Neural Network. In Proceedings of the 2022 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 25–27 January 2022; pp. 1–7. [[CrossRef](#)]
19. Anusha, C.; Rupa, C.; Samhitha, G. Region based Detection of Ships from Remote Sensing Satellite Imagery using Deep Learning. In Proceedings of the 2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM), Pradesh, India, 23–25 February 2022; pp. 118–122. [[CrossRef](#)]
20. Erfani, S.M.H.; Wu, Z.; Wu, X.; Wang, S.; Goharian, E. ATLANTIS: A benchmark for semantic segmentation of waterbody images. *Environ. Model. Softw.* **2022**, *149*, 105333. [[CrossRef](#)]
21. Rajyalakshmi, C.; Rao, K.R.M.; Rao, R.R. Compressed High Resolution Satellite Image Processing to Detect Water Bodies with Combined Bilateral Filtering and Threshold Techniques. *Trait. Signal* **2022**, *39*, 669–675. [[CrossRef](#)]
22. Juliandy, C.; Pribadi, O. Securing Medical Records of COVID-19 Patients Using Elliptic Curve Digital Signature Algorithm (ECDSA) in Blockchain. *CommIT (Commun. Inf. Technol.) J.* **2022**, *16*, 87–96.
23. Wang, Q.; Ihme, M.; Chen, Y.-F.; Anderson, J. A tensorflow simulation framework for scientific computing of fluid flows on tensor processing units. *Comput. Phys. Commun.* **2022**, *274*, 108292. [[CrossRef](#)]

24. Ben Ahmed, I. Hybrid UNET Model Segmentation for an Early Breast Cancer Detection using Ultrasound Images. *TechRxiv* **2022**. [[CrossRef](#)]
25. Jing, J.; Wang, Z.; Rättsch, M.; Zhang, H. Mobile-Unet: An efficient convolutional neural network for fabric defect detection. *Text. Res. J.* **2022**, *92*, 30–42. [[CrossRef](#)]
26. Cai, Y.; Wang, Y. Ma-unet: An improved version of unet based on multi-scale and attention mechanism for medical image segmentation. In Proceedings of the Third International Conference on Electronics and Communication; Network and Computer Technology (ECNCT 2021), Harbin, China, 3–5 December 2021; SPIE: Bellingham, DC, USA, 2022; Volume 12167.
27. Oh, H.; Nam, K.; Jeon, S.; Cho, Y.; Paek, Y. MeetGo: A trusted execution environment for remote applications on FPGA. *IEEE Access* **2021**, *9*, 51313–51324. [[CrossRef](#)]
28. Sharma, P.; Jain, S.; Gupta, S.; Chamola, V. Role of machine learning and deep learning in securing 5G-driven industrial IoT applications. *Ad Hoc Netw.* **2021**, *123*, 102685. [[CrossRef](#)]
29. Xiong, H.; Jin, C.; Alazab, M.; Yeh, K.H.; Wang, H.; Gadekallu, T.R.; Wang, W.; Su, C. On the design of blockchain-based ECDSA with fault-tolerant batch verification protocol for blockchain-enabled IoMT. *IEEE J. Biomed. Health Inform.* **2021**, *26*, 1977–1986. [[CrossRef](#)]
30. Yadav, A.; Joshi, D.; Kumar, V.; Mohapatra, H.; Iwendi, C.; Gadekallu, T.R. Capability and Robustness of Novel Hybridized Artificial Intelligence Technique for Sediment Yield Modeling in Godavari River, India. *Water* **2022**, *14*, 1917. [[CrossRef](#)]
31. Rupa, C.; Harshita, M.; Srivastava, G.; Gadekallu, T.R.; Maddikunta, P.K.R. Securing Multimedia using a Deep Learning based Chaotic Logistic Map. *IEEE J. Biomed. Health Inform.* **2022**. [[CrossRef](#)]
32. Srivastava, G.; Jhaveri, R.H.; Bhattacharya, S.; Pandya, S.; Maddikunta, P.K.R.; Yenduri, G.; Gadekallu, T.R. XAI for Cybersecurity: State of the Art, Challenges, Open Issues and Future Directions. *arXiv* **2022**, arXiv:2206.03585.
33. Alazab, M.; Gadekallu, T.R.; Su, C. Guest Editorial: Security and Privacy Issues in Industry 4.0 Applications. *IEEE Trans. Ind. Inform.* **2022**, *18*, 6326–6329. [[CrossRef](#)]
34. Yenduri, G.; Gadekallu, T.R. Recent Advancements in Network and Cyber Security Using RNN. In *Trust, Security and Privacy for Big Data*; CRC Press: Boca Raton, FL, USA, 2022; pp. 129–143.
35. Anajemba, J.H.; Tang, Y.; Iwendi, C.; Ohwoekevw, A.; Srivastava, G.; Jo, O. Realizing efficient security and privacy in IoT networks. *Sensors* **2020**, *20*, 2609. [[CrossRef](#)]
36. Iwendi, C.; Jalil, Z.; Javed, A.R.; Reddy, T.; Kaluri, R.; Srivastava, G.; Jo, O. Keysplitwatermark: Zero watermarking algorithm for software protection against cyber-attacks. *IEEE Access* **2020**, *8*, 72650–72660. [[CrossRef](#)]
37. Alazab, M.; Tang, M. (Eds.) *Deep Learning Applications for Cyber Security*; Springer: Berlin/Heidelberg, Germany, 2019.
38. Etaher, N.; Weir, G.R.; Alazab, M. From zeus to zitmo: Trends in banking malware. In Proceedings of the 2015 IEEE Trust-com/BigDataSE/ISPA, Helsinki, Finland, 20–22 August 2015; IEEE: Washington, DC, USA, 2015; Volume 1, pp. 1386–1391.