Development of Continuous Spatiotemporal Flood Masks using Deep Learning and Remote Sensing

A Thesis

submitted to

Indian Institute of Science Education and Research Pune

in partial fulfillment of the requirements for the

BS-MS Dual Degree Programme

by

Vinaykumar Daivajna

Indian Institute of Science Education and Research Pune

Dr. Homi Bhabha Road,

Pashan, Pune 411008, INDIA.

May 2024

Supervisor: Dr. Manmeet Singh (IITM Pune)

Certificate

This is to certify that this dissertation entitled **"Development of Continuous Spatiotemporal Flood Masks using Deep Learning and Remote Sensing."** towards the partial fulfillment of the BS-MS dual degree program at the Indian Institute of Science Education and Research, Pune represents the work carried out by Vinaykumar Daivajna at the Indian Institute of Tropical Meteorology, Pune, under the supervision of Dr.Manmeet Singh, during the academic year 2023-2024.

Manmeet Singh

Dr. Manmeet Singh

Committee:

Supervisor: Dr. Manmeet Singh

TAC Expert: Dr. Joy Merwin Monteiro

I dedicate this thesis to my family and friends.

Declaration

I hereby declare that the matter embodied in the report entitled **"Development of Continuous Spatiotemporal Flood Masks using Deep Learning and Remote Sensing"** is the result of the work carried out by me at the Indian Institute of Tropical Meteorology, Pune, under the supervision of Dr. Manmeet Singh, and the same has not been submitted elsewhere for any other degree.

Vinaykumar Daivajna

Acknowledgments

I am grateful to my supervisor, Dr. Manmeet Singh, for allowing me to work under his supervision and always being supportive. I thank my TAC expert, Dr. Joy Merwin Monteiro, for his valuable suggestions during this work and for giving me space to work in his lab. I thank my Father, Mother, and Brother for believing in me and always being supportive no matter what happened. I thank my friends Rahul, Piyush, Rohit, and Mohit for their constant support during my difficult times and for giving valuable suggestions.
I thank my IISER Pune community for all the love and support they gave me during my time with them.

Abstract

Sentinel-1 is a satellite with a synthetic aperture radar(SAR) instrument. It is an active microwave satellite that provides data up to 10m resolution in all weather conditions and day/night, making it suitable for detecting floods. Water bodies appear dark in sentinel-1 due to microwaves' high absorbance, making detecting the water from the background possible. Satellites operating in the visible range of EM spectra are well-suited for detecting water bodies. However, clouds and shadows block visible light from reaching the satellite, making it challenging to get satellite images during flood events. In this work addressing this problem, we chose 20 different places on different kinds of terrains worldwide, like large complex rivers, high urban areas, small rivers, large lake areas, etc., to detect the water in all types of terrains. We have evaluated the performance of Logistic Regression, XGBoost, and U-Net that use sentinel-1 image as input and dynamic worl data(first 10m resolution near-real time land use land cover data developed by Google) as the target to detect the water bodies. After testing different models, U-Net outperformed all other models in various terrains and has an average F1 Score greater than 0.9.

Keywords: Sentinel-1, Dynamic World, Water Mask, U-Net, Logistic Regression, XGBoost, Otsu, F1Score.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Floods are one of the most devastating disasters (natural and human-induced); they cause significant loss of human lives, damage properties, and disrupt communities worldwide. Due to climate change, deforestation, and heavy urbanization, the frequency of floods has increased worldwide in recent years. A World Health Organization(WHO) report has estimated that more than 2 billion people were affected by floods between 1998 and 2017, and floods accounted for 80-90% of the disasters around the world in the past ten years (WHO,2018). According to the United Nations Office for Disaster Risk Reduction (UNDRR), 2017 floods accounted for 8.6 million displacements in the USA. (Yer-raniga et al., 2020) Over 11,000 houses have been damaged, and more than 400 people lost their lives during recent floods in Himachal Pradesh (India) in Sept 2023 (The Hindu, 2023). Heavy rain is one of the leading causes of flooding. Urbanization and deforestation alter the natural rainwater movements and do not allow the water to infiltrate, which increases the possibility of flooding and soil erosion. (Feng et al. 2021). Dams play a vital role in water management. If dams are not appropriately managed during heavy rainfall, dams can overflow and cause heavy flooding. One such event was in Bihar in August 2016, where the water was released only when it reached almost complete. (Times of India,2016). The availability of high-resolution satellite images and the recent advancements in machine learning have helped to improve flood prediction and flood inundation. Hydrological models such as HEC-RAS and Mike Flood have been used for flood prediction and inundation mapping. These physics-based models iteratively solve the simplified version of Navier-Stokes equations to get to the results.
There are some studies in which people have used combined machine learning with hydrological models for flood prediction(HEC-RAS)(Tamiru et al. 2021). Google has

developed an operational flood warning system using deep learning using the river gauge data that alerts the user (Nevo et al. 2022). Remote sensing data provides vital information about the Earth's structure. The periodic and continuous data collected by satellite helps to observe spatial and temporal patterns on the earth's surface. Satellites operate in visible light, and some operate in the microwave region of the EM spectrum. Sentinel-1 is an active microwave satellite launched by the European Space Agency. Sentinel-2 is a satellite that operates in the visible region. The large amount of data the satellite collects is suitable for training deep learning models and extracting valuable information. Satellite data have been used for various tasks like flood mapping (Mateo-Garcia et al. 2021), fire detection (Kang et al. 2022), land cover classification (Brown et al. 2022), road detection, etc.

Different approaches have been used to map the water from the remote sensing data. Spectral indices such as the Normalized Difference Water Index (NDWI) use near-infrared (NIR) and short-infrared to detect the presence of water using optical images. (Moharrami et al 2021). used the thresholding approach to map water bodies from sentinel1 images. The thresholding approach works well when the image has a bimodal distribution and fails to map water accurately.

Deep learning has emerged as one of the go-to methods for segmentation tasks because of its high accuracy and reliability. Different deep learning models, such as ResNet-34 (Yi et al. 2022), SegNet (Badrinarayanan et al. 2021), and U-net(Ronneberger et al.,2015) (which is a famous architecture that's used for biomedical image segmentation)., are the popular ones that have been used for the segmentation tasks. The remote sensing community has used a similar architecture for water detection from sentinel-1 images. (Zhang et al.,2021) have used the sentinel1 images to extract roads using U-net-based architecture. (Jiang et.al. 2022) have used the attention-based Unet3+ architecture to map surface water in China. (Pech-May et al. 2023) have used U-net-based architecture for water body mapping in Mexico. Much of this research work done in this field has focused on a single region.

Training Deep Learning algorithms requires a large amount of data. Considering this necessity, some datasets have been developed. (Rambour et al. 2020) I have created a sen12Flood dataset, which contains weakly labeled sentinel-1 and sentinel-2 images

from different regions collected between December 2019 and May 2020, and many people have used this dataset to train deep-learning models to analyze flooding events. (Drakonakis at al 2022, Mateo-Garcia et al. 2021)..Another dataset developed by (Bonafilia et al. 2020) has also created a labeled sentinel-1 and sentinel-2, which includes both permanent water bodies and 11 flood events from different places for labeling sentinel-1 images; sentinel-2 images are used as the base.

In this work, we have trained and evaluated the performance threesholding method called OTSU's method and different machine learning and deep learning models, such as Logistic Regression, XGBoost, and CNN-based U-Net model, to generate water mask from sentinel-1 images.

Chapter 2

Theoretical Background

2.1 Artificial Intelligence

Intelligence is the ability to understand, learn, and think. Humans are one of the most intelligent species on this planet. Even though humans are not the strongest, tallest, and biggest species on the earth, humans have been ruling this planet, and the reason for this is human intelligence. Humanity has always been curious to understand how the human brain works. Much research has been done to understand how this most complex, fantastic machine works, but we still need to fully know how the human brain works.

The human brain is one of the most complex machines on earth. This complex machine has helped humans become the most intelligent species on earth. How the brain works and responds to different conditions has always been a matter of curiosity to humanity. Much research has been done to understand how the brain works, but we still need to understand it fully.

Artificial intelligence refers to the simulation of human-level intelligence in machines. Even though the term artificial intelligence has been prevalent in the 21st century, the idea of artificial intelligence is quite old. Artificial intelligence (AI) was first coined by Howard McCarthy in 1956, a computer and cognitive scientist.

Figure 2.1 below shows how the terms artificial intelligence, machine learning, and deep learning are related.

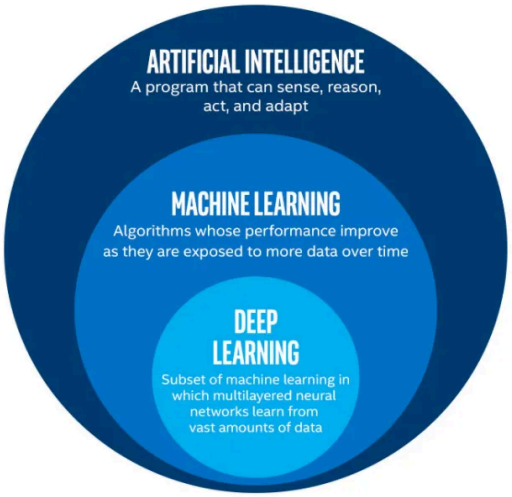


Figure 2.1 AI, Machine Learning, and Deep Learning

2.2 Machine learning

Machine learning is a subfield of artificial intelligence in computer science that is concerned with studying and developing algorithms that can learn from data, generalize to new data, and carry out tasks without explicit instructions. There are mainly three kinds of machine-learning approaches they are:

1. Supervised learning

Training the algorithm on a labeled dataset in which, for each training input, there is a corresponding output. The algorithm learns to map the relationship between the input and the output.

Examples of supervised learning algorithms are linear Regression, Logistic Regression, Support Vector machines (SVM), and Neural Networks (image recognition tasks).

2. Unsupervised Learning

Unlike supervised learning, this approach uses unlabelled data, in which the algorithm tries to find the patterns and relationships within the data without any explicit guidance from the output—for example, Clustering algorithms dimensionality reduction.

3. Reinforcement Learning

In this kind of learning, an agent picks up decision-making skills via interacting with its environment. This is achieved by rewarding the desired behavior and penalizing the undesired behavior. This is very similar to how humans learn. Unlike supervised learning, where the labeled data is provided, and the model understands the relationship between the input and output, the agent can explore the environment and learn from the result of its action. This self-teaching approach is where the model gets trained from the trial and error method, and the actions are performed with the aim of maximizing the reward. Examples: Self-Driving Cars, Algorithmic Trading, Content Recommendations.

2.2.1 Linear Regression

Linear Regression is one of the most basic machine learning models. The model learns the linear relationship between the input and output. Let X_1, X_2, X_3…., X_n be the independent variable, and Y be the dependent variable in the dataset we want to find the linear relationship. So, our goal is to find the best fit that maps the linear relationship between the dependent and independent features.

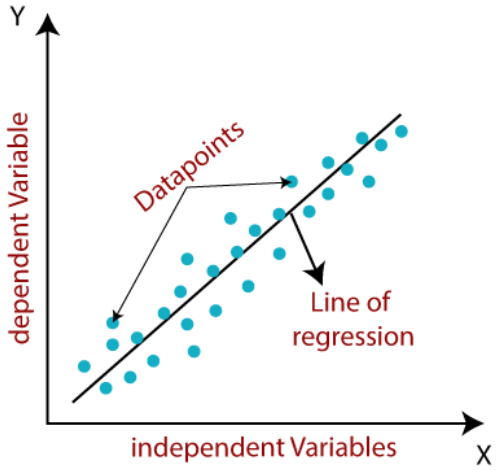


Figure 2.2 Linear Regression

Any linear relationship between an independent variable and a dependent variable can be expressed as

$$Y = w_1X_1 + w_2X_2 + w_3X_3 + \ldots\ldots\ldots w_nX_n + b \quad (2.1)$$

where $w_1w_2w_3\ldots\ldots w_n$ are called weights and b is called bias.

We define a cost function to find the best fit for the data and minimize the cost by adjusting the weights and bias. Mean Squared Error is used as a cost function in the case of linear regression.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)2 \quad (2.2)$$

where

n represents the number of data points, y_i represents the ith dependent data point, $\hat{y}_i$ and represents the ith prediction value. Equation 2.2 is a convex function with a single minimum, so we can iteratively update the weights and get the minimum. This method is called gradient descent.

2.2.2 Gradient Descent

The French mathematician Augustin-Louis Cauchy first described the gradient descent algorithm $in\ 1897$. This is a first-order iterative algorithm for finding the local minimum of a differentiable multivariable function. From the fundamentals of calculus, we know that any multivariable function defined and differentiable in a neighborhood of points increases fastest in the gradient direction $F(x)$ at a. Using this idea, the gradient descent algorithm tries to minimize the cost. It updates the parameter values by moving opposite the gradient $F(x)$ at a, i.e. -$\nabla F(a)$. ($F(x)$ decreases the fastest in the direction of -$\nabla F(a)$ at a).

$$W_{n+1} = W_n - \lambda \nabla F(a_n) \tag{2.3}$$

For any small enough learning rate $\lambda \in R^+$ $F(a_{n+1}) \leq F(a_n)$. As we want to move towards the local minimum, against the gradient, the term $\lambda \nabla F(a_n)$ is subtracted from the W(W represents the parameters). Generally, one starts with a guess W_0 for a local minimum and then iteratively updates the parameters until convergence.

The function that is minimized in machine learning is called the cost function.

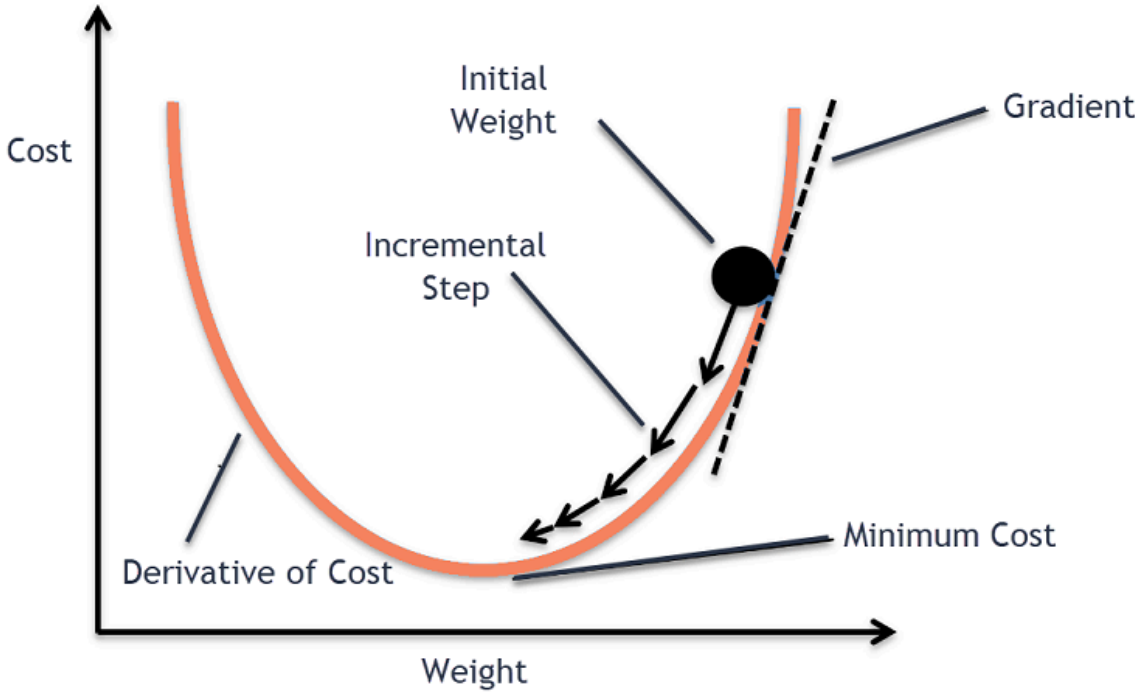


Figure 2.3 Gradient Descent (Haji et al.2021)

2.2.3 XGBoost

XGBoost(extreme gradient boosting) is a highly efficient tree-based algorithm that can be used for both classification and regression tasks. The XGBoost is a family of gradient-boosting algorithms that iteratively trains decision trees on weak learners and

combines them to create strong learners. The final prediction is the weighted sum of all the weak learners.

2.3 Deep Learning

Deep learning is a type of machine learning that uses multi-layered neural networks to learn and make predictions from the data—the architecture of neural networks tries to mimic the structure and function of the human brain; the Deep learning models can be trained on an extensive amount of data. Currently, deep learning is used in self-driving cars, speech recognition, language translation, and generative artificial intelligence.
Large Language Models (LLM) like ChatGpt and Gemini that use transformer architecture have become famous and are trained on extensive amounts of data available on the internet; one can interact with natural language with these models.

2.3.1 Artificial Neural Network

Artificial neurons are the basic building block of all deep learning models. It takes one or more inputs to sum up the input values and apply an activation function like sigmoid,relu, leaky relu, etc, to produce the final output. The defined cost function is minimized using the Gradient Descent and backpropagation algorithm.

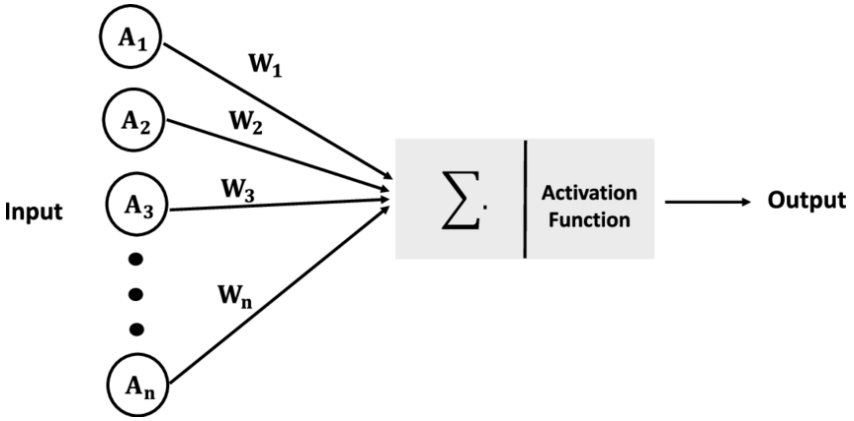


Figure 2.4 Artificial Neuron

For a neural network to learn complex tasks like image recognition, speech recognition, language translation, etc, the artificial neural network needs to be complicated enough to catch all the patterns in the data. An artificial neural network with multiple hidden layers can achieve these complex tasks. An artificial neural network with more than three layers, including input and output, is called a deep learning algorithm. For a deep learning algorithm, there are multiple hidden layers(see Fig below), and various parameters need to be updated in each iteration; for this, the error measured through a cost function needs to propagate from the final output layers back to the first layer. This can be achieved by an algorithm called backpropagation.

2.3.2 Backpropagation

Backpropagation is a gradient estimation method needed to train an artificial neural network. An optimization algorithm (such as gradient descent) uses backpropagation to compute the updates required for the parameters at each iteration while training an artificial neural network. Backpropagation is an application of the Leibniz chain rule to artificial neural networks.

Backpropagation computes the gradients of the loss function with respect to the weights of the artificial neural network. When ANN has multiple layers, this algorithm calculates the gradients for all the layers using dynamic programming, which helps avoid the iterative calculation of the gradients. For an ANN having multiple layers,

$$g(x) = f^{l}(W^{l} f^{l-1}(W^{l-1} f^{l-1} \ldots\ldots f^{1}(W^{1}(x))) \qquad (2.4)$$

x: input features

l:layer number

f^{l} : activation functions of l^{th} layer

W^{l} : weight matrix of l^{th} layer

$g(x)$: output

C : Cost function

a^{l} : activations of l^{th} layer

If we want to calculate the gradient of the loss function of the second layer, we need all the gradients from the later layers using the chain rule. The formula to calculate the loss function would look like the one below.

$$\frac{dC}{da^{l}}\frac{da^{l}}{dz^{l}}\frac{dz^{l-1}}{da^{l}}\frac{da^{l-1}}{dz^{l-1}}\ldots\ldots\ldots\ldots\ldots\frac{da^{2}}{dz^{2}} \quad (2.5)$$

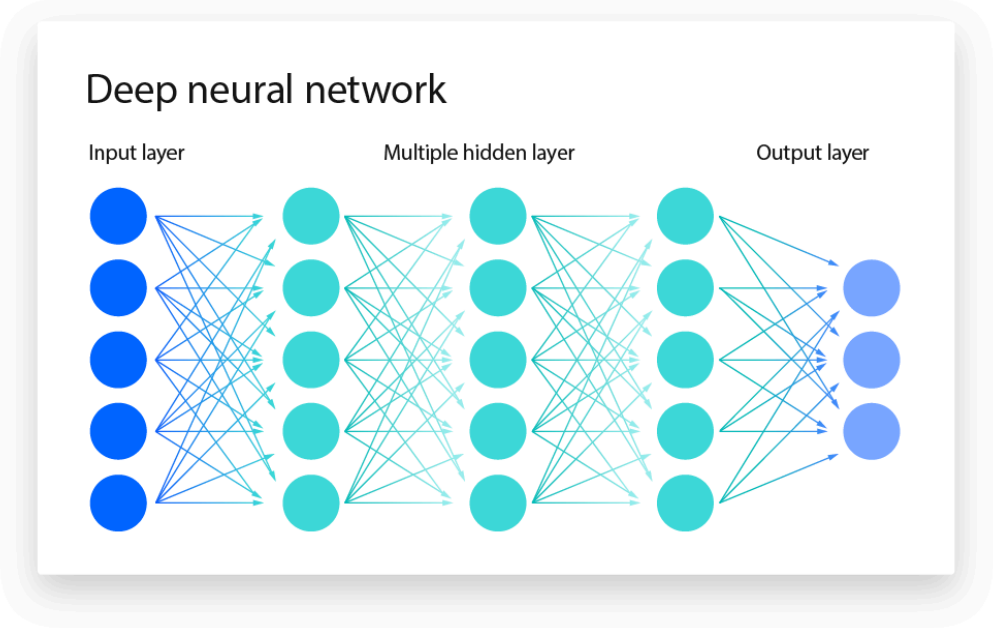


Figure 2.5 Deep Neural Network

2.3.3 Convolutional Neural Network

In artificial neural networks, as the size of the input increases, the number of parameters also increases. When it comes to images, let's say we have an image of size (100*100); if we try to use ANN, then we will have $\sim 10^{4}$ weights in one single neuron, and if we try to train with 7-8 layers, then the number of parameters quickly can reach a near million. For images, we know that the information in one pixel in the image is related to the nearby pixels. Also, the pixels far away from each other are not entirely different features. They represent similar kinds of information; keeping this in mind,

instead of treating each pixel in the image as a distinct feature, we can use some fixed number of weights that consider the nearby pixel values to give output irrespective of the size of the image. This dramatically reduces the number of parameters needed. This is what a convolution operation does, i.e., parameter sharing. In any image, we gather information by collecting information from some collection of pixels rather than individual pixels. The convolution operation, which uses the collection of pixels, better catches the pattern in the image. Hence, convolutional neural networks have become very popular in image-related tasks.

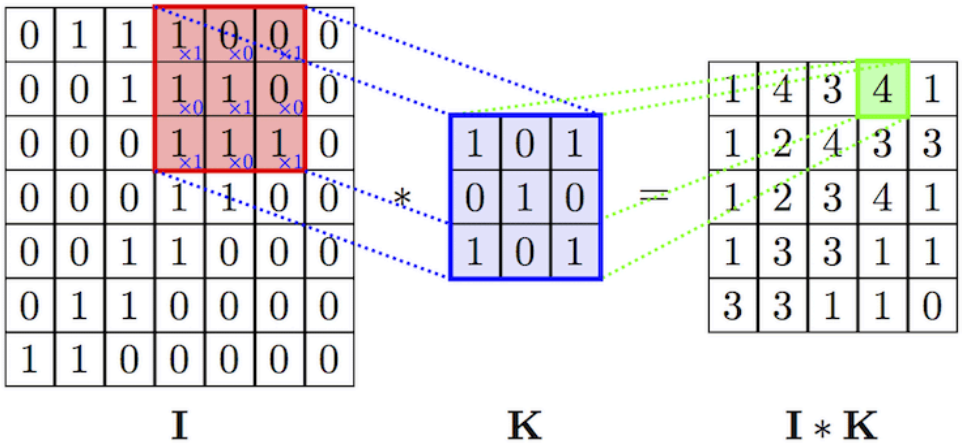


Figure 2.6 Convolution Operation (Mohamed et al.2017)

Figure (2.6) shows the pictorial representation of convolution operation. 3*3 and 5*5 are the most commonly used kernel sizes. After a convolutional operation, the size of the image decreases. One can adjust the padding and strides to get the desired output size. Various convolutional operations are used in deep learning models, such as regular convolutional operation, max pooling, averaging, upsampling, transposed convolutional, etc.

Common Layers Used In Convolutional Neural Networks

1. Convolutional Layer

This layer does a standard convolutional operation, as represented in Figure 2.6. The kernel has learnable parameters that move over the image and produce the output. The image size generally decreases after the convolution operation; one can adjust the padding and stride to get the desired size output. The convolutional operation extracts the various features. Typically, multiple filters (kernels) are applied, and each filter tries to extract different features from the image.

2. Max-Pool Layer

This layer is commonly used in CNN; as the name suggests, it outputs the maximum values inside the defined kernel size.

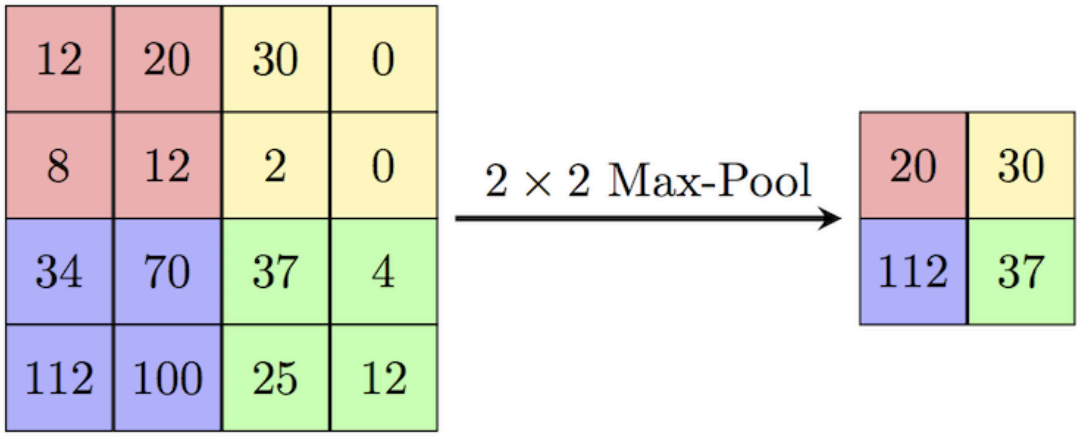


Figure 2.7 Max-Pool Layer

When the max pooling is applied, the image size decreases. For example, if we use (2*2) kernel size with the stride of 2 on an image of size (n*n), the output image will be of size (n/2*n/2). Max pooling operation reduces the size of the image by retaining the most crucial information. There are no learnable parameters in the max-pooling operation.

3. Transposed Convolution Layer

This is an upsampling layer that outputs an image size greater than the input image size. In transposed convolution operation, the input image is moved over the kernel to produce the output; one can adjust the stride and padding to get the desired output size.

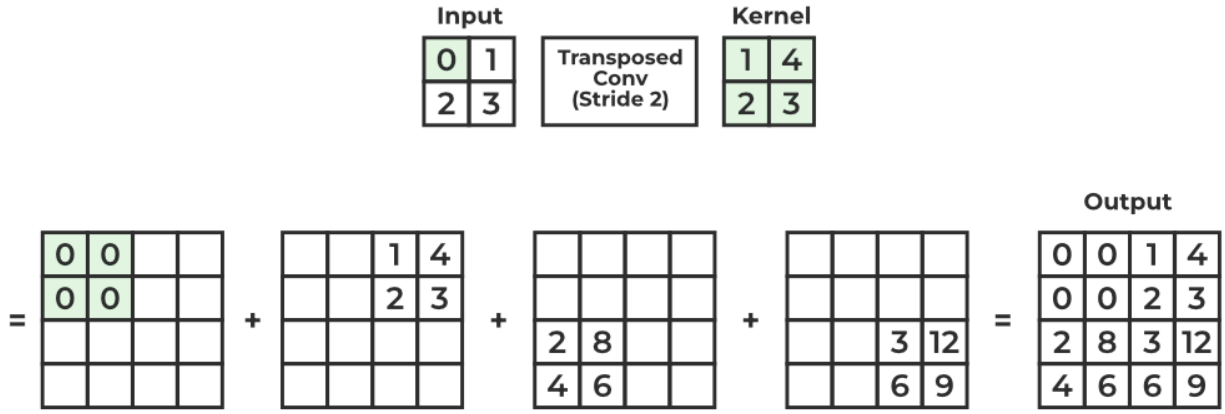


Figure 2.8 Transposed Convolution

Like regular convolution operation, the transposed convolution also has learnable parameters. Transposed convolutions are used in autoencoders, generative adversarial networks(GANs), semantic segmentation, etc.

Chapter 3

Data and Methodology

3.1 Sentinel-1

Sentinel1 is an active microwave satellite. It was developed and launched by the European Space Agency under the Copernicus Program satellite constellation. This mission was initially composed of Sentinel-1A and Sentinel-1B, which share the same orbital plane. Sentinel-1 satellites operate in the C-band (frequency range:4-8 GHz). The satellites carry synthetic aperture radar (SAR) (a form of the radar in which the radar antenna moves over a target region to provide finer spatial resolution). Sentinel1 data is available in all weather conditions, day and night, even in the presence of clouds, as microwaves can easily pass through clouds. The SAR instrument has a spatial up to 5m and a swath(area of the earth's surface imaged by the satellite during a single pass) of 410 km. The satellite orbits in a Sun-synchronous orbit near a polar orbit. This orbit has 12 days repeat cycle.

```
ee.ImageCollection("COPERNICUS/S1_GRD")
```

The sentinel1 data for this project is accessed through Google Earth's engine using the above code snippet. This collection contains the Sentinel-1 Ground Range Detected(GRD) scenes, calibrated and ortho-corrected using the sentinel-1 Toolbox. The collection is updated daily in Google Earth's engine. New data is injected within two days after the data becomes available. Each image contains 1 or 2 out of 4 possible polarization bands depending on the instrument polarization settings. Single-band VV or HH and dual-band VV+VH and HH+HV are the possible combinations.

- VV stands for vertical transmit/vertical receive, single co-polarization.
- HH stands for horizontal transmit/horizontal receive single co-polarization.
- VH+VV stands for vertical transmit/horizontal receive dual-band cross-polarization

- HV+HH stands for horizontal transmit/vertical receive dual-band cross-polarization

All the values represent the backscattered values for different polarizations. All the images also include another band called ‘angle,’ representing the approximate incidence angle from the ellipsoid in degrees at each location. Each image is preprocessed using the Sentinel-1 toolbox. The steps followed are as follows:

1. **Thermal Noise Removal**

   Thermal noise is caused by the random motion of charge carriers(usually electrons). Thermal noise is directly proportional to the temperature in conductors. Removal of thermal noise is essential as it can degrade the quality of the image. There are many methods used for thermal noise reduction, such as Temporal filtering(which involves averaging the values over time for the same area), spatial filtering(Gusssain Filter, Median Filter), and some statistical methods, such as Markov Random Fields (MRFs)

2. **Radiometric Calibration**

   The spectral data from the satellite sensors vary over time due to many variables, including atmospheric absorption, scattering, sensor calibration, and others. We must eliminate the impact of these fluctuating elements to identify any changes (surface reflectance or absorbance) in the landscape over time. This procedure is known as radiometric calibration. There are two methods for handling this: relative and absolute. We require ground measurements for the absolute method, which entails significant expense. An alternative strategy is a relative technique, which entails normalizing the various date pictures to an analyst-selected reference image.

3. **Terrain Correction**

   Terrain correction refers to correcting the distortion caused by the earth's topography (mostly mountains). It helps move the image pixels into a proper spatial relationship with each other.

3.3 Sentinel-2

Sentinel2 is an optical imaging satellite launched by the European Space Agency; this is an Earth Observation mission from the Copernicus Programme that acquires high-resolution (10m to 60m) optical images over land and coastal water.
Sentinel-2A was launched on 23 June 2015, and Sentinel-2B was found on 7 March 2017. Sentinel-2C is scheduled to launch this year. Sentinel-2 is multispectral with 13 bands that cover a wide range of wavelengths; this allows the satellite to capture images in visible, near-infrared, and shortwave infrared. The sentinel-2 data can be accessed through the Google Earth engine using the below snippet:

```
ee.ImageCollection("COPERNICUS/S2")
```

3.2 Dynamic World

Dynamic World is the first and only near real-time 10m resolution land use land cover dataset that has class probabilities of each class and labels of nine different courses.
The dynamic world uses sentinel-2 L1C for prediction.
Nine different classes in the dynamic world

1. Water
2. Trees
3. Grass
4. Flooded Vegetation
5. Crops
6. Shurb and Scrub
7. Built Area
8. Bare Ground
9. Snow and Ice

The dynamic world land use land cover data can be accessed through the Google Earth engine using the snippet below:

```
ee.ImageCollection("GOOGLE/DYNAMICWORLD/V1")
```

3.4 Data Information

3.4.1 Sentinel1 VV band

This is one of the two bands used as input features to train the model to detect water. The maximum value of this band observed in all the training data chosen is 35, and the minimum value observed is -55.

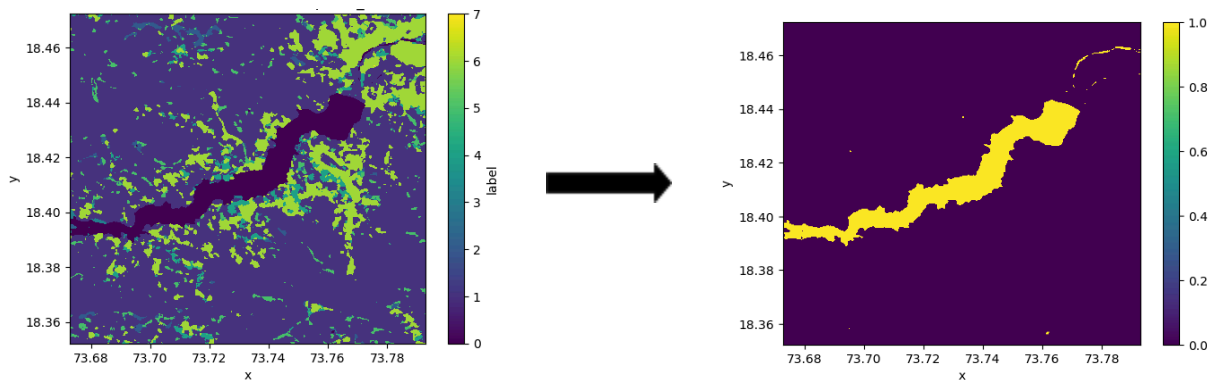
3.4.2 Dynamic World label

A 10 m resolution near real-time land use land cover data, representing the land cover type at each location, has nine different classes, as explained in the earlier section. We have downloaded data from 20 different places to cover all kinds of terrains such as highly urban areas, small rivers, large complex rivers, and low urban areas, with each scene having (1280*1280) pixels with 10m resolution, ensuring a good balance between water and nonwater pixels, with almost 20% pixels as water, which will help train the model better. We have used three different models to compare the binary classification results.

3.5 Data Processing

3.5.1 Water Mask Generation

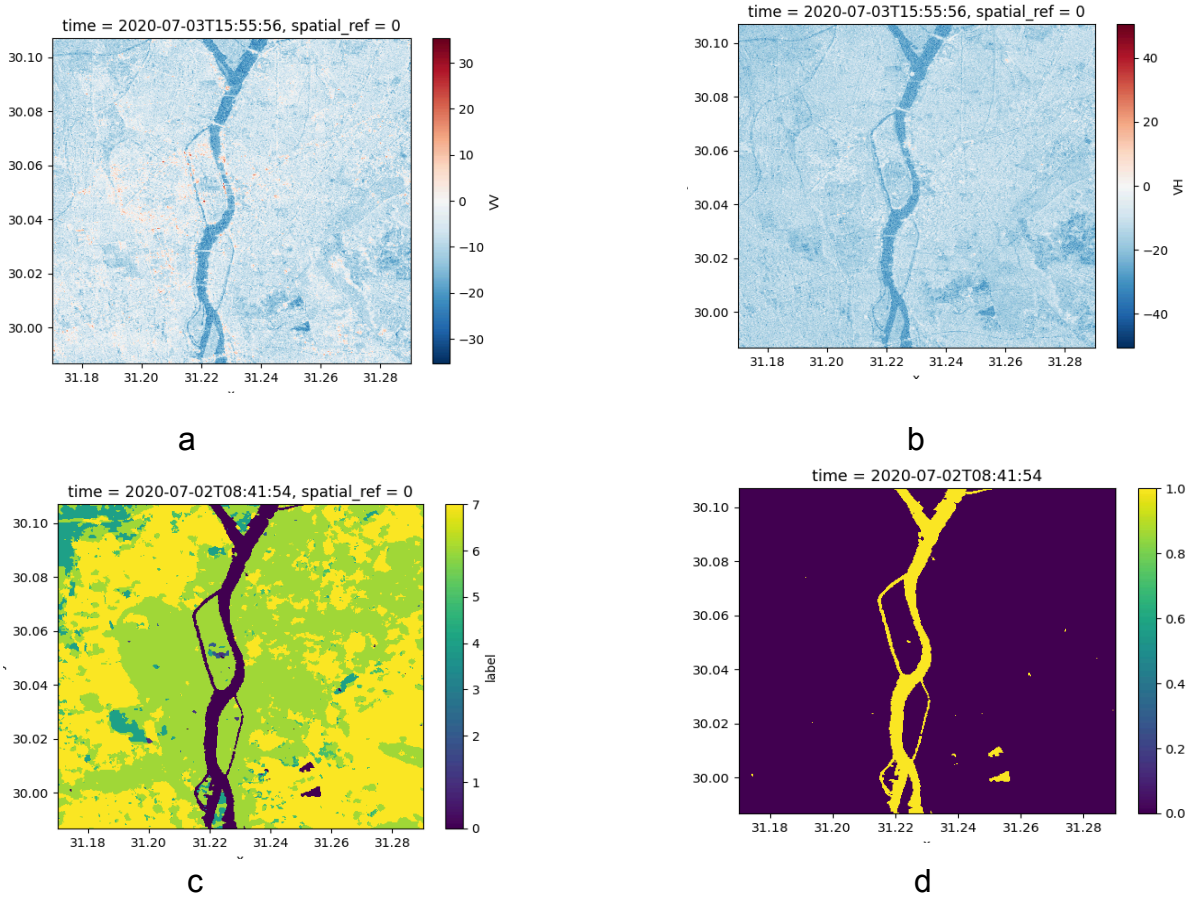
The dynamic world has a softmax value for each of the nine different classes and a label that uses all nine class values to create a label for each pixel, which tells the land cover type. The target (water mask) is generated using this label. This label has values from 0 to 8. To get the water mask, all the water pixels and the flooded vegetation as are marked as one, and all the other pixel values are marked as zero. (0 represents the water, and 3 represents flooded vegetation in the dynamic world label.)



a b

Figure 3.1 Water Mask Generation (a) Dynamic world (b) Water Mask

3.5.2 Training Data



a b c d

Figure 3.2 Training Dataset (a) Sentinel1 VV band, (b) Sentinel1 VH band, (c) Dynamic World Land Use Land Cover (d) Dynamic World Water Mask for Cairo (Egypt)

Similar to the above satellite images, the data from 20 different places such as (Wuhan(china), Cairo(Egypt), Houston (USA), Austin(USA), Zambia (Africa), Mumbai (India), Brahmaputra River(Bangladesh), etc.. is used for training, covering the various terrain types. Sentinel-1 and Sentinel-2 are two different satellites. To avoid the mismatch between the input and the labels, we have chosen the maximum 24-hour time difference between the sentinel-1 and sentinel-2 data(dynamic world data), ensuring no rainfall during that period.

3.5.3 Normalization

In machine learning, data normalization refers to scaling all the feature values to a similar range. The normalization is beneficial because:

1. It gives fair importance to all the features
2. Normalization helps achieve faster convergence during model training.
3. Reduces the influence of the outliers that can skew the training process.

Depending on the requirement, different normalization techniques, such as Z-Score Normalization, Min-Max scaling, log scaling, and clipping, are used in machine learning. We have two different bands in sentinel1 data that we have used as the input features. These bands have different ranges of values; hence, we have used Min-max scaling.
It rescales all the values between 0 and 1. As the target variables(water mask) values are between 1 and hence, we did not apply any normalization step.

$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

X_{min} : Minimum values of the feature X, X_{max} : Maximum value of the feature X

3.6 Models

3.6.1 OTSU's Method

Otsu's method, named after Nobuyuki Otsu, is used for automatic image threesholding in computer vision and image processing. This algorithm gives a single intensity threshold that divides the pixels into foreground and background classes. This is achieved by maximizing the inter-class variance. Inter-class variance is defined as

$$\sigma_b^2(t) = w_0(t)w_1(t)\left[\mu_0(t) - \mu_1(t)\right]^2 \quad (3.2)$$

w_0 and w_1 represents the class probabilities separated by a threshold t,

$w_0 + w_1 = 1$ as they represent the probabilities.

$\mu_0(t)$ and $\mu_1(t)$ are,

$$\mu_0(t) = \frac{\sum_{i=0}^{t-1} ip(i)}{w_0(t)} \quad (3.3)$$

$$\mu_1(t) = \frac{\sum_{i=t}^{L-1} ip(i)}{w_1(t)} \quad (3.4)$$

L represents the number of bins in the histogram.

Algorithm

1. Calculate the probability and histogram for each intensity level.
2. Start with some initial $w_i(0)$ and $\mu_i(0)$
3. Iterate through all the thresholds from the minimum to maximum intensity.
    1. Update w_i and μ_i
    2. Compute $\sigma_b^2(t)$
4. The threshold which gives the maximum $\sigma_b^2(t)$ will be the desired threshold.

Pixel values are divided into bins, all the threshold values are used, and the intra-class variance is calculated.

3.6.2 Logistic Regression

We want to classify each pixel from the sentinel1 image as water or no water. The one fundamental idea that comes to mind is starting with the logistic regression model, a linear binary classification algorithm. The sentinel1 images have built-in speckle and thermal noise, limiting our ability to use one-to-one mapping of sentinel1 and dynamic world water masks. The nearby pixel values influence the value of a pixel; hence, to classify any pixel as water or nonwater, we used the values of all the (5*5) pixels surrounding that pixel as input features to the logistic regression model.

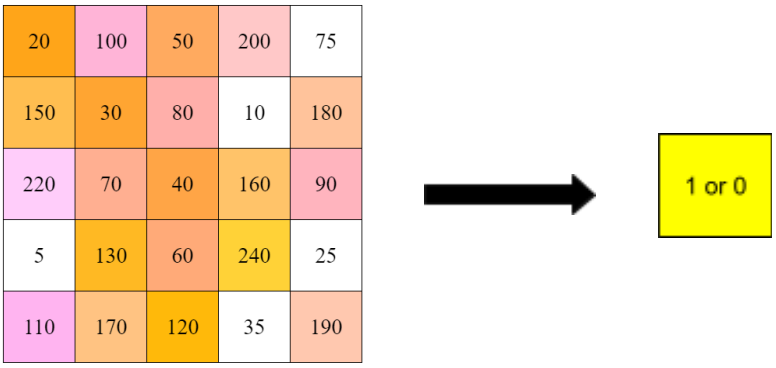


Figure 3.3 Features used for Logistic Regression and XGBoost model

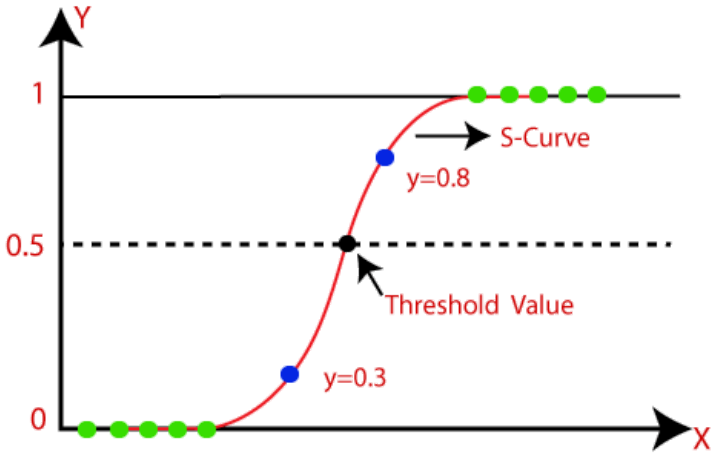


Figure 3.4 Logistic Regression

For training the logistic regression model, 90% of the data is used for training the model, and 10% of the data is used for validation. The loss function used is Binary logistics. The optimizer used is Adam, which uses a learning rate of 0.001. The model is trained for 100 epochs using Pytorch.

3.6.3 XGBoost Model Training

To train the xgboost classification model, we used a 5*5 vectorized input feature, as explained in the logistic regression model section. We used 25 estimators because using more estimators did not give any significantly better results.

3.6.4 U-Net Model

U-Net is a famous convolutional neural network architecture used in image segmentation. The U-Net architecture was first proposed by (Ronneberger et al. 2015), who used this architecture for biomedical image segmentation. The U-Net has an encoder-decoder architecture. The architecture is named after its U-shaped structure.

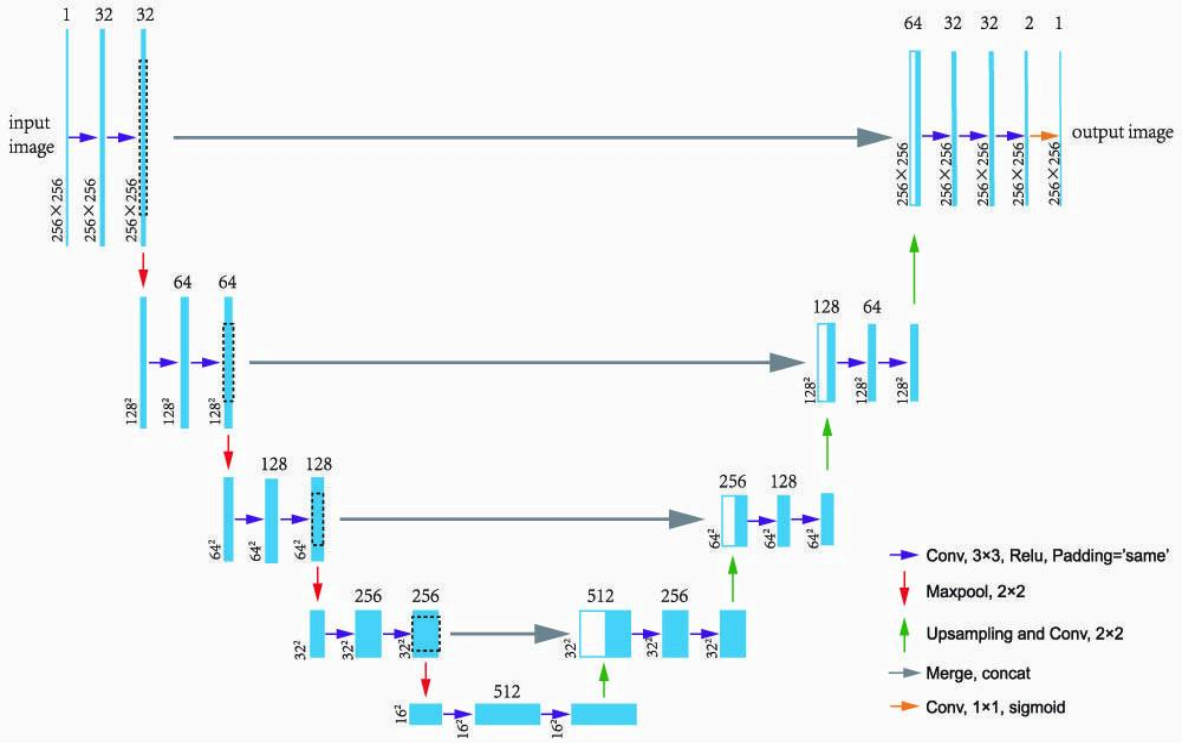


Figure 3.5 U-Net Architecture (Ou et al. 2023)

Components of the U-Net Architecture

Contracting Path

The contracting path consists of regular convolutional and max pool layers. This part of U-net architecture acts as the encoder. In this part of the U-Net, the spatial dimension of the image is reduced, and the number of channels increases. Different features and the context of the input image are captured in this part.

Expanding Path

This part of the U-Net serves as a decoder path. Transposed convolution or upsampling layers are used to increase the size of the image. The segmentation mask is created using the features learned from the encoder part.

Skip Connections

Skip connections are an essential feature in the U-Net architecture, where the layers at the same level in the contracting and expanding path are connected. Skip connections allow the decoder to access the features at different scales. Combining the information from different scales, the U-Net architecture better localizes the objects and produces more accurate segmentation masks.

U-net model Training

The architecture in Figure (3.5) is used to train the model. Batch Normalization is used after each convolution operation while training the model. Each sentinel-1 and dynamic world image was downloaded from the Google Earth engine is of dimensions (1280*1280) at 10m resolution. We know that deep learning models need a large amount of data for training, and as we have downloaded data from 20 different places, we have applied data augmentation steps using vertical flips and horizontal flips. After this, we have created (256*256) pixel non-overlapping image chunks. This helped us to generate a total of 1500 images. We randomly shuffled the image chunks to avoid bias

in learning during batch training. 1350 images are used for training the model and 150 for validation. Adam is used as an optimizer and binary cross-entropy is used as a loss function. A learning rate of 0.0005 is used as this learning rate gives better convergence. All other parameters are kept as default. A 2D kernel size of 5*5 because it gave the best results. Each convolution operation is used in all the convolutional operations. A padding of 2 is used in all convolution operations to maintain size. ReLU is used as an activation function except for the last layer, where the sigmoid activation function is used. The model is trained for 100 epochs. Figure 3.6 shows the training and validation loss. We can see from the figure that the training loss decreases continuously. Still, when we look at the validation loss, it increases after around 60 epochs, which indicates the model overfits the training data. To avoid this affecting our model, we have used the parameters that gave the least validation loss.

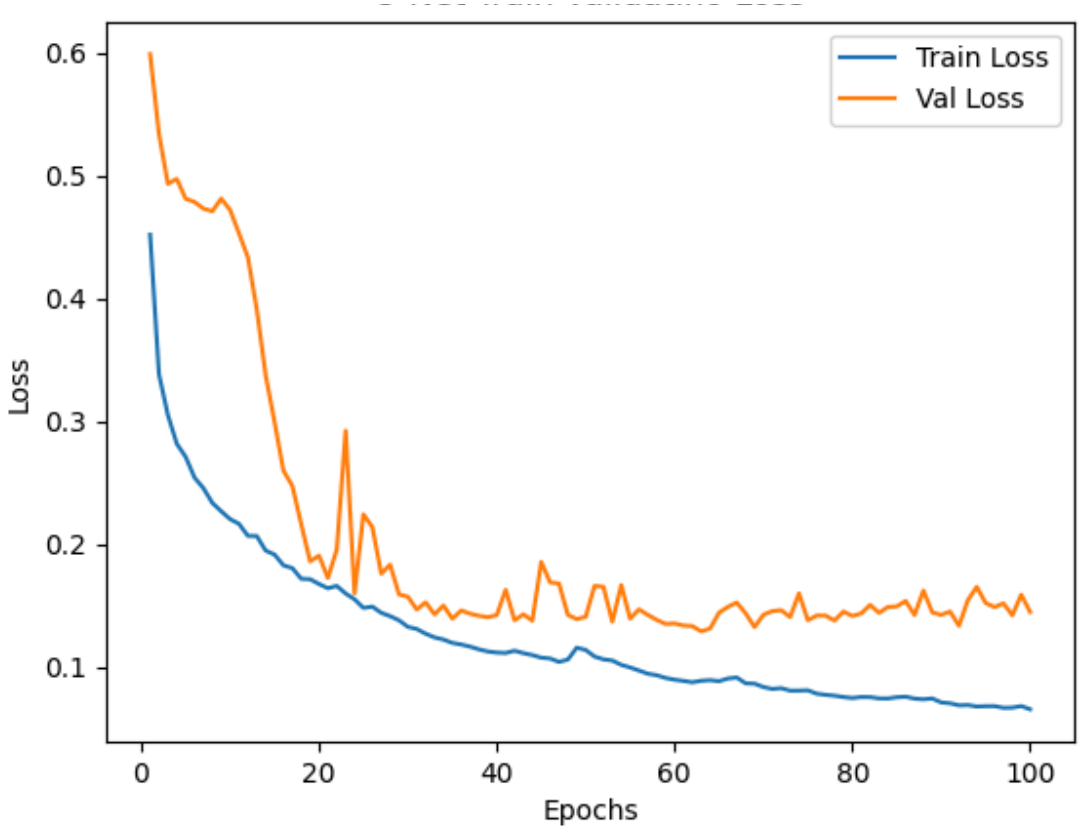


Figure 3.6 Training and Validation Loss

Chapter 4

Results and Discussion

4.1 Evaluation Matric:

The following matrices evaluate the performance of all the models used to classify the water and non-water pixels.

Accuracy

This metric calculates the proportion of values that are correctly classified. This matrix indicates how well the model performs but cannot capture the errors distributed across different classes. Accuracy is defined as below.

$$Accuracy = \left(\frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative}\right) \quad (4.1)$$

Precision

This metric allows us to calculate the quality of the positive prediction made by the model. High precision indicates the high quality in positive classification.

$$Precision = \left(\frac{True\ Positive}{True\ Positive + False\ Positive}\right) \quad (4.2)$$

Recall:

This metric calculates how well the model can recall the positive samples. High recall indicates that the model can collect all the positive samples.

$$Recall = \left(\frac{True\ Positive}{True\ Positive + False\ Negative}\right) \quad (4.3)$$

F1Score

This matric is a harmonic mean of precision and recall. There is always a tradeoff between precision and recall. The F1 score is a good metric for measuring the model's overall performance, as it considers both precision and recall.

$$F1Score = 2 * \left(\frac{Precision*Recall}{Precision+Recall}\right) \quad (4.4)$$

4.2 Methods

As discussed in earlier chapters, we have used four different models: OTSU Thresholding Method, Logistic Regression, Linear Regression, XGBoost, and U-Net to compare the performance of the water direction from the sentinel-1 image; we have tested each model's performance in different kinds of terrains, such as Large Lake areas, high urban areas, large complex rivers, small rivers, etc.

4.3 Results

After processing the data and training the different machine-learning models, we obtained the following results:

4.3.1 Large Lake Area (Kerala)

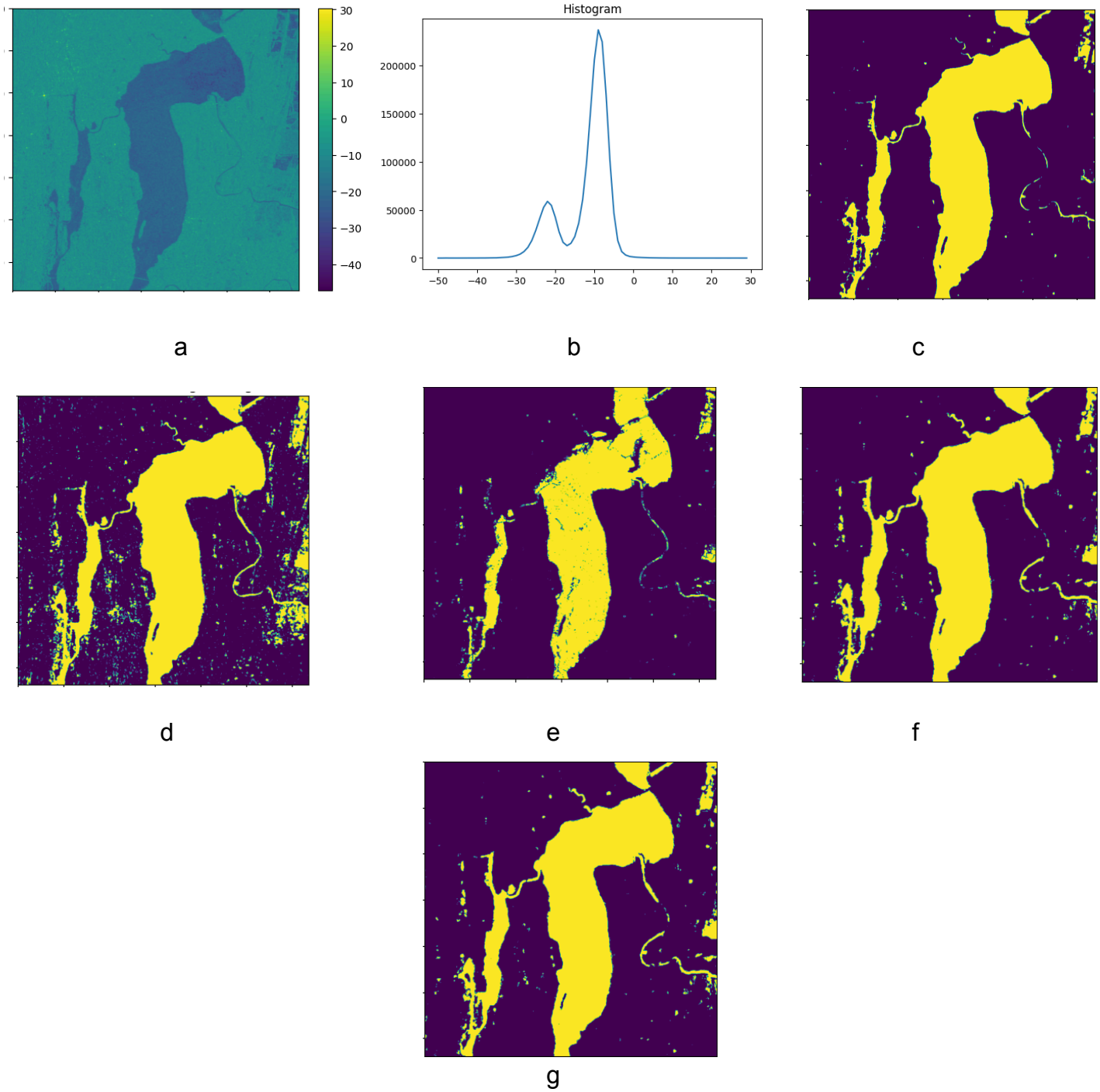


Figure 4.1: Large Lake Result(Kerala) (a) Sentinel-1 VV, (b)Sentinel-1 VV Histogram, (c)True Water Mask, (d) Otsu Water Mask, (e) Logistic Regression Water Mask (f) XGBoost Water Mask (g) U-Net Water Mask

The yellow color in the above images represents the water.

Model	Accuracy	Precision	Recall	F1 Score
Otsu	0.9677	0.9357	0.9250	0.9303
Logistic Regression	0.9409	0.8075	0.9813	0.8860
XGBoost	0.9546	0.9463	0.8545	0.8980
U-Net	0.9792	0.9481	0.9637	0.9558

Table 4.1 Model Comparision For Large Lake (Kerala)

Figure 4.1 shows the output of the different models, and Table 4.1 compares the performance of various models used to create water masks from sentinel-1 images. We can see from Table 4.1 that the most basic OTSU(threesholding method) is able to create a water mask with a very good F1 score of 0.93. From the histogram of the sentinel-1 image (Figure 4.1 b), we can see a precise bimodal distribution in the sentinel-1 values. Because of this, the OTSU thresholding model is able to discriminate between water and land more accurately. If we look at recall, we can see that the logistic regression model has the highest value but an average precision value. Overall, the U-Net model outperforms all other models. The difference in performance of the best and worst performing models is minimal in this case, and that could be because of the type of terrain as we are testing the models. The large lake is a simple terrain; almost all models can differentiate between water and non-water.

4.3.2 High Urban Area (New York)

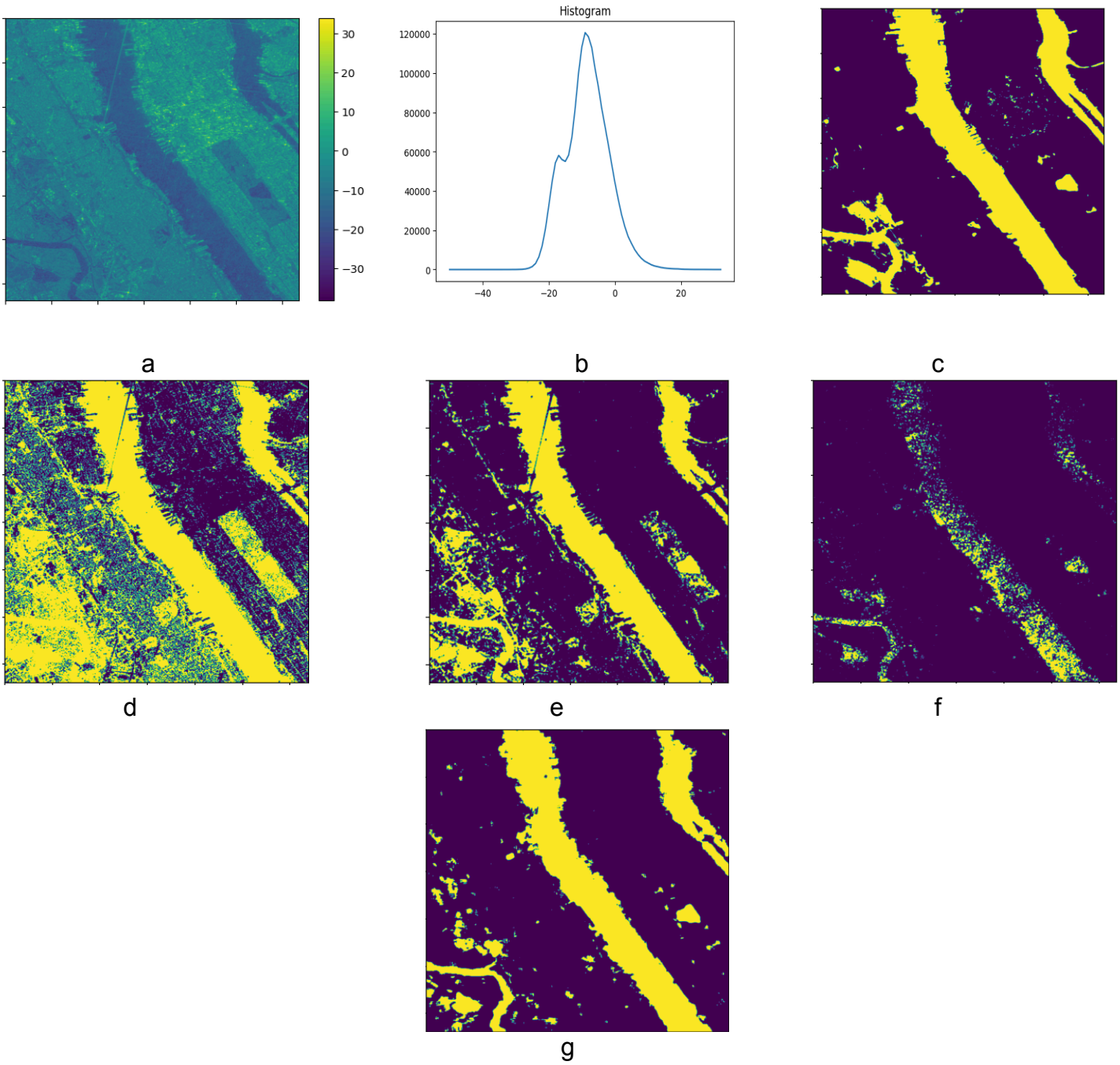


Figure 4.2 High Urban Area Results(New York) (a) Sentinel-1 VV, (b) Sentinel-1 VV Histogram (c)True Water Mask (d) Otsu Water Mask (e) Logistic Regression Water Mask (f) XGBoost Water Mask (g) U-Net Water Mask

The yellow color in the above images represents the water.

Model	Accuracy	Precision	Recall	F1 Score
Otsu	0.6649	0.3960	0.9362	0.5566
Logistic Regression	0.9291	0.9885	0.6931	0.8149
XGBoost	0.8274	0.9728	0.2401	0.3852
U-Net	0.9585	0.9585	0.8575	0.9052

Table 4.2 Model Comparision For High Urban Area (New York)

Figure 4.2 shows the output of the different models. Table 4.2 compares the performance of various models used to create water masks from sentinel-1 images. Looking at Figure 4.1(b), a sentinel-1VV histogram, we can see that the histogram doesn't have a bimodal distribution. Hence, as expected, the OTSU method performs poorly in segmenting water from the background. Table 4.2 shows that the Logistic Regression and XGBoost models have very high precision values; however, their Recall is poor. In urban areas, the complex terrain, wide roads, tall buildings, etc, make it difficult to accurately separate the water and land. Hence, the simple models are not able to perform well in this case. Still, the CNN-based U-Net model is complex enough to overcome this problem and is able to segment water from the background in a complex, high urban terrain with a very good F1 score, which is greater than 0.9.

4.3.3 Large Complex River (Bangladesh)

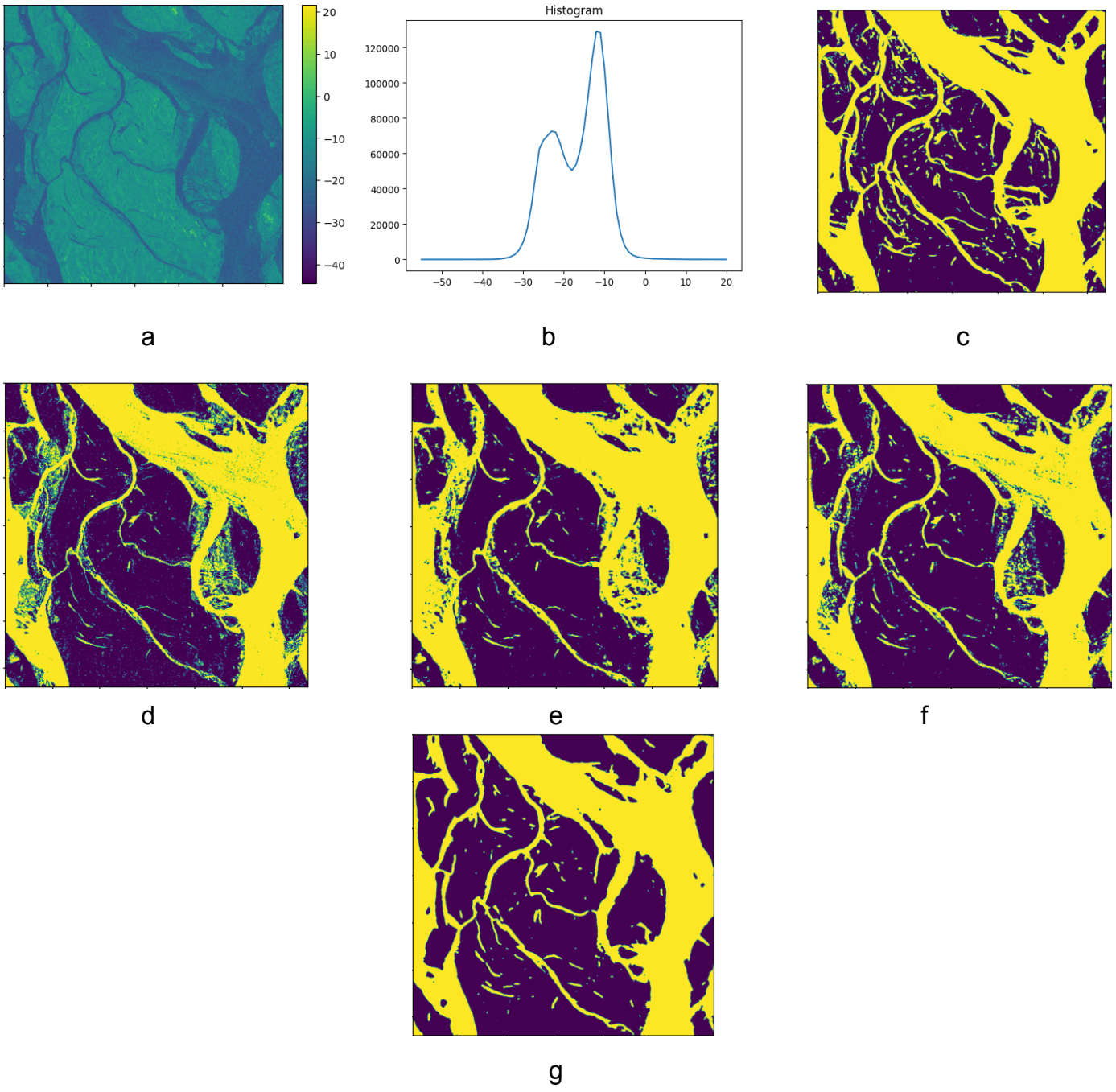


Figure 4.3 Complex River (Bangladesh) (a) Sentinel-1 VV, (b)sentinel-1 VV Histogram, (c)True Water Mask, (d) Otsu Water Mask, (e) Logistic Regression Water Mask (f) XGBoost Water Mask (g) U-Net Water Mask

The yellow color in the above images represents the water.

Model	Accuracy	Precision	Recall	F1 Score
Otsu	0.8462	0.8636	0.7996	0.8304
Logistic Regression	0.8471	0.8337	0.8433	0.8385
XGBoost	0.9367	0.9367	0.8218	0.8755
U-Net	0.9194	0.9802	0.8459	0.9081

Table 4.3 Model Comparision Large Complex River (Bangladesh)

Figure 4.3 shows the output of the different models. Table 4.3 compares the performance of various models used to create water masks from sentinel-1 images.

Figure 4.3 (b) shows that the sentinel-1 VV values have a bimodal distribution; as expected, the OTSU model performs well in detecting the water. Near Large river areas, the terrains are mostly simple; hence, simple models such as logistic regression and OTSU perform well. There isn't much difference between the XGBoost and U-Net models, which capture low-level and high features and better segment the water from the background. Like other terrains, the CNN-based U-Net model is outperforming all other models.

4.3.4 Small River (Karnataka)

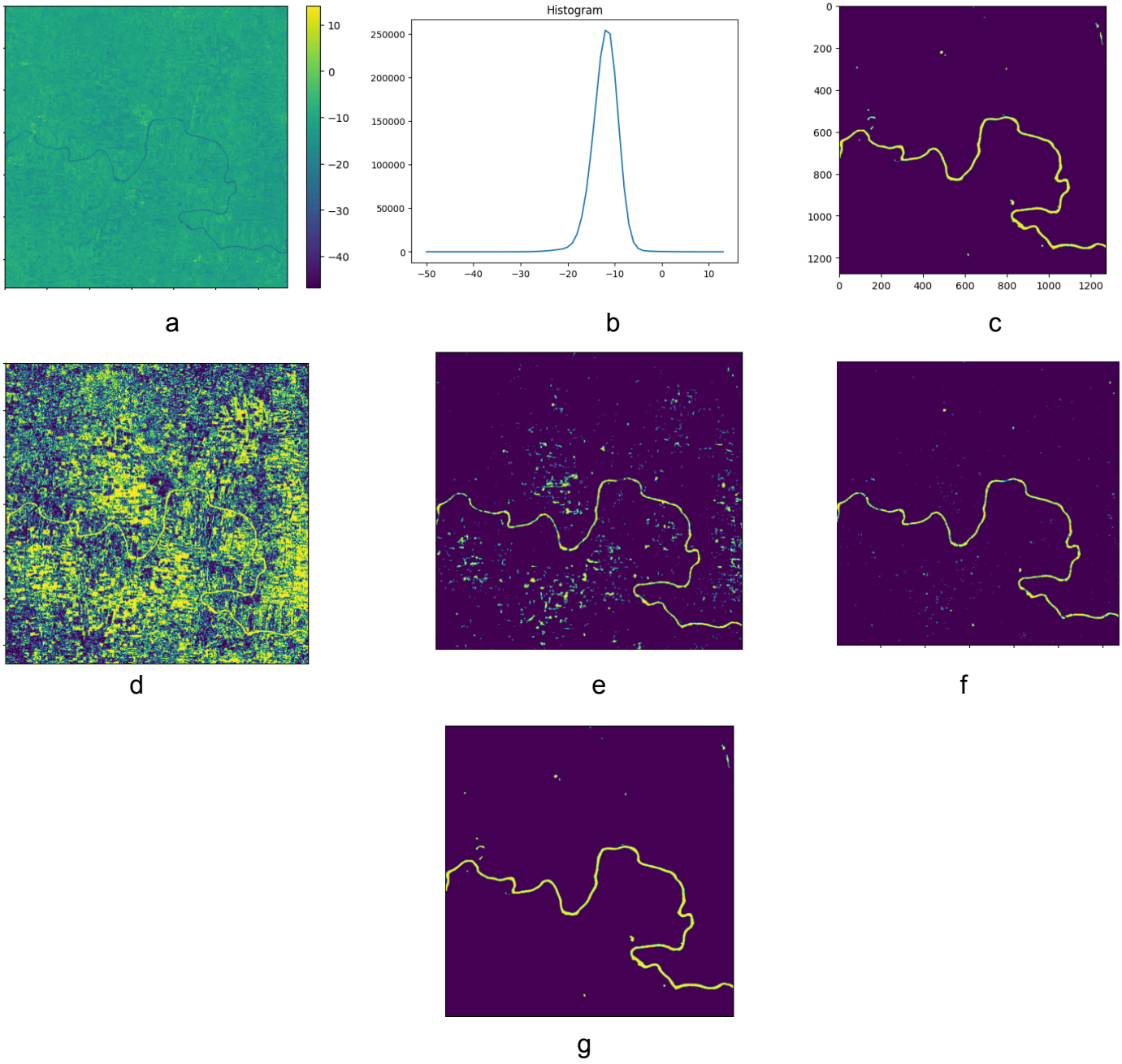


Figure 4.4 Small River Results (Karnataka) (a) Sentinel1 VV, (b)sentinel-1VV Histogram, (c)True Water Mask, (d) Otsu Water Mask, (e) Logistic Regression Water Mask (f) XGBoost Water Mask (g) U-Net Water Mask

The yellow color in the above images represents the water.

Model	Accuracy	Precision	Recall	F1 Score
Otsu	0.5268	0.0320	0.7784	0.06144
Logistic Regression	0.9708	0.3607	0.5968	0.4496
XGBoost	0.9924	0.8833	0.7112	0.7884
U-Net	0.9956	0.8853	0.8966	0.8909

Table 4.4 Model Comparision Small River Results (Karnataka)

Figure 4.4 shows the output of the different models. Table 4.4 compares the performance of various models used to create water masks from sentinel-1 images.
Looking at Figure 4.3 (b), we can see a single mode in the histogram of sentinel-1 VV values. As expected, the OTSU method performs very poorly when looking at the histogram plot. The Logistic Regression method also does not correctly segment water from the background. The XGBoost model does a good job and has a precision that is comparable to the U-Net, but the U-Net model beats it in the recall by a good margin; hence, once again, the U-Net is the best-performing model in this terrain.

4.3.5 Overall Performance

Model	Accuracy	Precision	Recall	F1 Score
Otsu	0.7514	0.5568	0.8598	0.5949
Logistic Regression	0.9220	0.7476	0.7786	0.7473
XGBoost	0.9278	0.9348	0.6569	0.7368
U-Net	0.9545	0.9430	0.8988	0.9150

Table 4.5 Overall Performance of All Models. (Averaged across various terrains)

Chapter 5

Conclusion

In this work, we aimed to create a generalized model capable of generating water masks from sentinel-1 images in various terrains; in doing so, we evaluated the performance of different methods, such as Otsu, Logistic Regression, XGBoost, and CNN-based U-Net model. Upon assessing the performance of each technique, we found that in simple terrains such as large lake areas and plain river areas, even simple methods such as OTSU and Logistic Regression can do a good job of segmenting water from the background. However, when it comes to complex terrain (high urban areas) and small rivers, the U-Net model does a very good job segmenting water from the background and outperforms all other methods by a very high margin. The U-Net has an overall F1 score(average across different terrains) greater than 0.9. So, we propose a CNN-based U-Net model for generating water masks from sentinel-1 images for all kinds of terrains. Our trained model can be used to detect water in crucial conditions, such as flood mapping, water assessments, lake water availability, dam water management, etc.

5.1 Future Work

Using the U-Net model, which outperformed all other models, we have generated a time series of water masks from sentinel-1 images(which is added in the appendix). This time series of images can train a deep-learning model using numerical weather prediction, land cover type, digital elevation model, and solid type that predicts the extent of flooding. This predictive model can safeguard a lot of human lives. By providing advanced warnings and precise flood extents, communities can enact and minimize the damage the flood can cause.

Appendix

Some examples of Generated Water Masks for a Region of Bhagalpur(Bihar)

(Yellow Color represents the water)

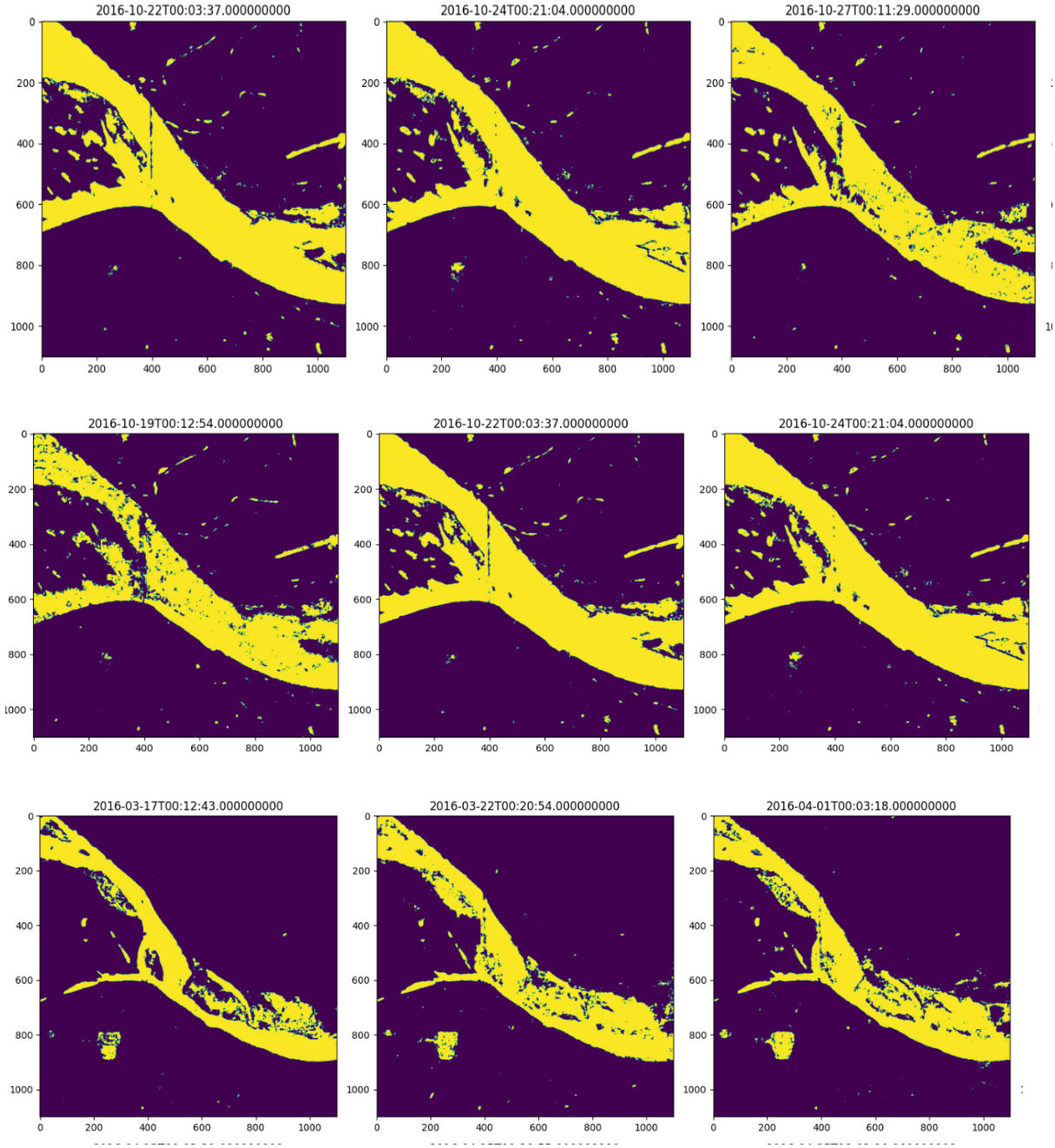


Figure 6.1 Generated Water Mask (Bhagalpur(Bihar))

Water masks are generated for all the available sentinel-1 images from 2016 to 2023, which we plan to use for flood water extent mapping.

Bibliography

1. WHO 2018 Report on Floods
2. Iyer-Raniga, Usha, and Mittul Vahanvati. "Resilience of the poor and vulnerable against disasters and associated economic shock." *No Poverty: Encyclopedia of the UN Sustainable Development Goals* 1 (2020).
3. The Hindu (Sept 02, 2023) Monsoon fury: “ 400 people died during a disaster in Himachal Pradesh, says Govt ”.
4. Feng, Boyu, Ying Zhang, and Robin Bourke. "Urbanisation impacts on flood risks based on urban growth data and coupled flood models." *Natural Hazards* 106.1 (2021): 613-627.
5. Times of India (Aug 24, 2016) “Poor dam management responsible for Bihar flood”
6. Tamiru, Habtamu, and Megersa O. Dinka. "Application of ANN and HEC-RAS model for flood inundation mapping in lower Baro Akobo River Basin, Ethiopia." *Journal of Hydrology: Regional Studies* 36 (2021): 100855.
7. Nevo, Sella, et al. "Flood forecasting with machine learning models in an operational framework." *Hydrology and Earth System Sciences* 26.15 (2022): 4013-4032.
8. Mateo-Garcia, Gonzalo, et al. "Towards global flood mapping onboard low-cost satellites with machine learning." *Scientific Reports* 11.1 (2021): 7249.
9. Kang, Yoojin, et al. "A deep learning model using geostationary satellite data for forest fire detection with reduced detection latency." *GIScience & Remote Sensing* 59.1 (2022): 2019-2035.
10. Brown, Christopher F., et al. "Dynamic World, Near real-time global 10 m land use land cover mapping." *Scientific Data* 9.1 (2022): 251.

11. Moharrami, Meysam, Mohammad Javanbakht, and Sara Attarchi. "Automatic flood detection using sentinel-1 images on the Google Earth engine." *Environmental monitoring and assessment* 193.5 (2021): 248.
12. Yi, [a.] Lin, Guangxiang Yang, and Yicheng Wan. "Research on Garbage Image Classification and Recognition Method Based on Improved ResNet Network Model." *Proceedings of the 2022 5th International Conference on Big Data and Internet of Things*. 2022.
13. Badrinarayanan, Vijay, Ankur Handa, and Roberto Cipolla. "Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labeling." *arXiv preprint arXiv:1505.07293* (2015).
14. Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer International Publishing, 2015.
15. Zhang, Qianqian, et al. "A new road extraction method using Sentinel-1 SAR images based on the deep fully convolutional neural network." *European Journal of Remote Sensing* 52.1 (2019): 572-582.
16. Jiang, Chaowei, et al. "A fundamental mechanism of solar eruption initiation." *Nature Astronomy* 5.11 (2021): 1126-1138.
17. Pech-May, Fernando, Raúl Aquino-Santos, and Jorge Delgadillo-Partida. "Sentinel-1 SAR Images and Deep Learning for Water Body Mapping." *Remote Sensing* 15.12 (2023): 3009.
18. Rambour, C., et al. "Sen12-flood: a sar and multispectral dataset for flood detection." *IEEE: Piscataway, NJ, USA* (2020).
19. Drakonakis, Georgios I., et al. "Ombrianet—supervised flood mapping via convolutional neural networks using multitemporal sentinel-1 and sentinel-2 data fusion." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 15 (2022): 2341-2356.

20. Mateo-Garcia, Gonzalo, et al. "Towards global flood mapping onboard low-cost satellites with machine learning." *Scientific Reports* 11.1 (2021): 7249.

21. Bonafilia, Derrick, et al. "Sen1Floods11: A georeferenced dataset to train and test deep learning flood algorithms for sentinel-1." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2020.

22. Haji, Fardin, et al. "A deep learning approach for inverse design of the metasurface for dual-polarized waves." *Applied Physics A* 127 (2021): 1-7.

23. Mohamed, Ihab S. *Detection and tracking of pallets using a laser rangefinder and machine learning techniques*. Diss. European Master on Advanced Robotics+(EMARO+), University of Genova, Italy, 2017.

24. Ou, Haiyan, et al. "Suppressing defocus noise with U-net in optical scanning holography." *Chinese Optics Letters* 21.8 (2023): 080501.