Masked Autoencoder and Mamba-Based Self-Supervised Segmentation of SAR Imagery for Riverbank Erosion Detection

By:

Showrin Rahman
(24141255)
Sowad Rahman
(24341284)
Md. Tanvir Jawad
(24241165)
Tazower Rahman Sowad
(21201691)
Krishno Ghosh
(21201809)

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfilment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
BRAC University
June 2025

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Showrin Rahman	Sowad Rahman
24141255	24341284
Md. Tanvir Jawad	Tazower Rahman Sowad
24241165	21201691

Krishno Ghosh

21201809

Approval

The thesis/project titled “Masked Autoencoder and Mamba-Based Self-Supervised Segmentation of SAR Imagery for Riverbank Erosion Detection ” submitted by

1. Showrin Rahman (24141255)
2. Sowad Rahman (24341284)
3. Md. Tanvir Jawad (24241165)
4. Tazower Rahman Sowad (21201691)
5. Krishno Ghosh (21201809)

Of Spring, 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on June, 2025.

Examining Committee:

Supervisor:
(Member)

Dr. Md. Golam Rabiul Alam

Professor
Department of Computer Science and Enginnering
BRAC University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi

Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Riverbank erosion in Bangladesh causes significant environmental and socioeconomic challenges, including land loss that displaces communities and damages local economies. Traditional monitoring methods, such as field surveys and manual satellite imagery analysis, are labor-intensive, inaccurate, and lack continuous data. Synthetic Aperture Radar (SAR) imagery from Sentinel-1 provides precise, all-weather imaging, overcoming cloud cover limitations. We created a dataset using Sentinel Hub, retrieving daily SAR data from 2014 to 2024 for major Bangladeshi rivers, including Jamuna (Sirajganj, Tangail, Manikganj), Padma(Shariatpur, Munshiganj, Faridpur), Meghna (Chandpur, Lakshmipur, Narsingdi), Brahmaputra (Chilmari, Fulchhari, Bahadurabad), and Teesta (Lalmonirhat, Kurigram). This study addresses temporal data gaps in SAR imagery using Generative Adversarial Networks (GANs) to reconstruct missing data, ensuring continuous riverbank observations. It also employs a self-supervised learning (SSL) framework with a convolutional Masked Autoencoder (MAE) and Mamba blocks for land-water segmentation without labeled data. Our pipeline achieved an average River Intersection over Union (IoU) of 0.8550 and a Dice coefficient of 0.9038 across five rivers from 2015 to 2024, enabling robust segmentation. This approach supports scalable riverbank monitoring for disaster prevention and sustainable environmental management in Bangladesh.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

Adam Optimizer Adaptive optimization algorithm for neural networks

AOI Area of Interest - specific study locations

BatchNorm1D Batch Normalization Layer for 1D Inputs

CNN Convolutional Neural Network

Conv2D 2D Convolutional Layer

Cross-Entropy Loss Loss function for classification tasks

DeepLabV3+ Deep learning model for semantic segmentation

Dilation Expanding the receptive field in convolutions

D Discriminator in GANs that differentiates real vs. fake data

GAN Generative Adversarial Network

$\boldsymbol{H}_t, C_t$ Hidden state and cell state in LSTMs

InstanceNorm2D Instance Normalization Layer for 2D Inputs

JSD Jensen-Shannon Divergence

$\boldsymbol{L}_G$ Generator loss function in GANs

LSTM Long Short-Term Memory

Padding Border extension during convolution

PatchGAN GAN variant for localized image processing

ReLU Rectified Linear Unit Activation Function

SAR Synthetic Aperture Radar

Self-Attention Mechanism Focus-improving mechanism in deep learning

Sentinel-1 Radar imaging satellite by the European Space Agency

Sigmoid Sigmoid Activation Function

Skip Connections Enhancing learning through ResNet-like structures

SSL Self-Supervised Learning

Stride Step size of kernel application in CNNs

Tanh Hyperbolic Tangent Activation Function

Temporal GAN GAN model for time-series image generation

Transposed Convolution Deconvolution operation for upsampling

U-Net Convolutional network for image segmentation

WD Wasserstein Distance

Wh, bh Weights and biases for LSTM layers

$\boldsymbol{X}_t, Z_t$ Input sequence and noise vector in GAN models

Chapter 1

Introduction

1.1 Background

River erosion in Bangladesh is a critical environmental and socio-economic challenge, driven by the country's unique deltaic geography and dynamic river systems. Bangladesh, situated in the Ganges-Brahmaputra-Meghna (GBM) delta, is intersected by major rivers such as the Jamuna, Padma, and Meghna, which collectively drain one of the largest sediment loads globally [5]. These rivers, characterized by high discharge and sediment transport, particularly during the monsoon season (June–October), cause significant riverbank erosion, leading to the loss of thousands of hectares of agricultural land, homes, and infrastructure annually [26]. The Jamuna River alone is responsible for substantial erosion hotspots, displacing communities and disrupting livelihoods [29]. The socio-economic impacts are profound, with millions of people affected by displacement and loss of arable land, exacerbating poverty and food insecurity [15].

Monitoring river erosion is essential for disaster preparedness, infrastructure planning, and policy formulation. Traditional methods for monitoring riverbank erosion in Bangladesh include field surveys, GPS mapping, aerial photography, and satellite-based optical imagery (e.g., Landsat, Sentinel-2) [9]. However, these approaches face significant limitations, including infrequent data collection, weather dependency, and high costs, which hinder their ability to capture rapid, short-term erosion events [21]. Optical satellite imagery, while useful for seasonal or annual assessments, is particularly ineffective during the monsoon season due to persistent cloud cover, limiting its utility in Bangladesh's climate [24].

Synthetic Aperture Radar (SAR) has emerged as a superior tool for river monitoring in Bangladesh due to its all-weather, day-night imaging capabilities and high temporal frequency [38]. SAR systems, such as Sentinel-1, provide consistent data with repeat cycles of 6–12 days, enabling near real-time detection of riverbank changes and floodwater mapping [13]. SAR's ability to distinguish water from land through backscatter contrast is particularly effective for tracking dynamic river boundaries and sandbar formations, even in vegetation-covered or turbid water conditions [19]. Unlike optical imagery, SAR is unaffected by cloud cover, making it ideal for Bangladesh's monsoon-dominated environment [44].

Short-term erosion trends (1–5 years) are critical for understanding rapid riverine changes, enabling early warning systems, and supporting disaster management [29]. Daily or near-daily monitoring, achievable through SAR, captures fast erosion events and provides high-frequency data for time-series forecasting models, such as Long Short-Term Memory (LSTM) networks or Temporal Generative Adversarial Networks (GANs) [47]. Temporal gaps in data collection, however, can significantly hamper the accuracy of erosion trend predictions by missing critical bankline shifts and introducing noise into predictive models [35]. Thus, consistent, high-resolution SAR monitoring is essential for effective erosion management in Bangladesh's dynamic river systems.

1.2 Research Problem

Monitoring riverbank erosion in Bangladesh is critically impeded by a series of interconnected challenges that undermine the reliability, timeliness, and scalability of erosion tracking, ultimately delaying effective responses to environmental crises. Persistent cloud cover, especially during the monsoon season (June–October), severely limits the availability of usable optical satellite imagery, such as Landsat or Sentinel-2, creating significant temporal gaps that hinder continuous monitoring of riverbank changes [24], [31]. The limited revisit frequency of satellite sensors, including Synthetic Aperture Radar (SAR) systems like Sentinel-1 with 6–12 day cycles, further exacerbates these gaps, making it difficult to capture rapid or gradual erosion events essential for early warning and disaster response [38]. Additionally, accurate land-water segmentation, crucial for identifying erosion-prone areas, relies heavily on large, manually annotated datasets, which are scarce for Bangladesh's extensive and dynamic river systems, thus restricting the effectiveness of supervised deep learning models [40]. While SAR imagery from Sentinel-1 offers all-weather, high-resolution data ideal for Bangladesh's cloudy and monsoon-prone environment, existing models often struggle to achieve precise river boundary segmentation using SAR data alone due to the complexity of SAR backscatter interpretation and the lack of tailored algorithms [44]. Furthermore, traditional monitoring methods, such as field surveys, GPS mapping, and manual satellite image interpretation, are labor-intensive, time-consuming, error-prone, and lack the scalability needed for continuous, large-scale monitoring of Bangladesh's dynamic river systems [9]. Due to these mentioned limitations comes issues of incomplete datasets, unreliability of erosion prediction and less monitoring accuracy. These obstruct the decision-making strategies for the environment and planning of infrastructures [25]. To address these challenges, this study proposes a combined approach of utilizing Generative Adversarial Networks(GANs) for temporal gap filling within the Sentinel-1 SAR image [37] dataset with a self-supervised learning novel framework (SSL). [39], [43]. This architecture uses the unlabeled SAR data and creates a robust pipeline for land-water segmentation with very little or no labeled data, enhancing accuracy and efficiency. By combining GAN-based temporal gap filling with an advanced SSL approach, the study targets an accurate, scalable and efficient river erosion monitoring system and ensures sustainable environmental disaster management in Bangladesh.

1.3 Research Objectives

In Bangladesh, monitoring of river bank erosion is an important environmental issue to be continued and managed efficiently. To face this problem, we have to mitigate challenges of spatial,temporal data gaps and land-water segmentation.The following objectives outline the key goals of this research:

1. **Develop Comprehensive Dataset:** Create a robust dataset of Sentinel-1 SAR imagery, covering key Bangladeshi rivers (Jamuna, Padma, Meghna, Brahmaputra, and Teesta) from 2014 to 2024, to support continuous riverbank erosion monitoring and analysis.

2. **Develop Data Reconstruction Framework:** Implement a GAN-based framework to reconstruct missing temporal data in Sentinel-1 datasets, ensuring continuity and accuracy for riverbank monitoring. This approach aims to fill data gaps with higher accuracy, ensuring reliable and continuous riverbank erosion analysis even in the presence of cloud cover or missing data.

3. **Implement Hybrid Self-Supervised Learning:** Develop a novel hybrid self-supervised learning framework, integrating convolutional Masked Autoencoder (MAE) and simplified Mamba blocks, to segment land and water boundaries in SAR imagery with minimal labeled data.

4. **Contribute to Environmental Management:** : Implement the deep learning pipeline for scalable riverbank erosion monitoring to enable rapid disaster response and sustainable planning in Bangladesh.

Chapter 2

Related Work

2.1 Riverbank Erosion Monitoring Using Remote Sensing

A study shows a time series data of Sentinel-1 imagery for detecting riverbank erosion along the Jamuna riverbank. In their research, radar data was their preferred source for monitoring during the rainy season when optical imagery was often covered with cloud. Sentinel-1 data demonstrated potential for use in monitoring works. It shows reliablity at providing timely information for erosion monitoring and management [53]. The spatiotemporal changes in riverbank erosion and accretion along the Jamuna River in Sariakandi Upazila, Bogra District were evaluated in a decadal analysis [64]. In one study, GIS-processed Landsat imagery was used to find changes in riverbanks over the last thirty years. These changes were correlated with increased vulnerability amongst local populations. As a result, integrated riverbank management strategy is required for addressing the geological as well as the socio-economic situations. Another study demonstrated the requirement of countermeasures and performed a morphological assessment along the Jamuna River based on groyne solutions through numerical modeling. Their research also presents novel countermeasures by rechanneling flow and forming sandbars along the river bank. It proves the effectiveness of these structures at reducing erosion risk. Morphological dynamics were also seen as critical to future river management and thus requiring better understanding [66]. In a study of riverbank erosion, accretion, and community involvement on Chairman Ghat, Noakhali, [58] measured and reviewed the relevance. However, their work focused only on relative measurements of erosion and accretion areas. The study quantified erosion and accretion rates of 22 years using remote sensing and GIS techniques to show the extent of the prevailing land loss and its very social and economic impacts on the local population. For protecting lives, livelihoods, and environments of vulnerable committees, a lot of environmental protection strategies were needed, says the research.

2.2 Socioeconomic Impact of Riverbank Erosion

Riverbank erosion in Charbhadrasan Upazila of Faridpur District was empirically studied by analyzing the socioeconomic effects of erosion on local communities [42].The research was about environment challenge mitigation strategies to reduce the sufferings of affected people.As there are devastating socio-economic and environmental consequences of riverbank and soil erosion, the study depicts that soil erosion rates can be calculated to know which coastal areas are vulnerable and proves the requirement of effective soil conservation and management strategies [60].

2.3 Synthetic Aperture Radar (SAR) for Riverbank Erosion Detection

Sentinel-1 has been proved effective in detecting riverbank erosion along the Jamuna River. Due to recent advancement of Sentinel-1 Radar imagery, it can now detect more images within a certain area, get past cloud coverage and also monitors fulltime in some cases. Land cover classification and change detection were applied to green areas before rainy season versus land-water after the rainy season. In vegetated areas, systems like land cover classification and change detection were applied to monitor effectively. According to their methodology, erosion can be predicted one month early before the heavy monsoon rain starts. For flood detection, an attentive dual-stream siamese U-net was utilized at different time steps.

Although the methodology focused on flood detection, the usefulness of Sentinel-1 data in monitoring water-related disasters is demonstrated and it could be adapted for riverbank erosion detection. The study also showed that combining SAR data with an effective network architecture would outperform existing state-of-the-art methods [45]. To improve structure detection of narrow river structures, a noticeable deep learning method was introduced. The study applied exogenous information and a linear feature detector, with the despeckling neural network preprocessed to improve river segmentation [34]. This approach requires refining riverbank erosion detection methodologies. Sentinel-1 data made large scale data analysis possible and insert the results into platforms such as Google Earth Engine (GEE). In response to riverbank erosion along the Jamuna River, researchers [53] built an online interactive tool based on GEE to explore riverbank erosion in several monsoon seasons. Such tools may enhance data and analytical power. It may allow the stakeholders to interpret necessary information and prepare strategies for managing riverbank erosion and other environmental challenges.

2.4 Data Gap Filling Using Generative Adversarial Networks

In remote sensing and environmental monitoring, filling data gaps is critical for accurate impact analysis and decision-making [36]. Cloud gaps in satellite imagery hinder assessments

of land use and climate change. Generative Adversarial Networks (GANs) offer advanced solutions by generating realistic data consistent with the original dataset structure. GANs consist of a generator, which creates realistic data, and a discriminator, which evaluates the generated outputs, with combined training enabling complex pattern modeling [6].
GANs are highly effective for data gap filling, handling various data types, including time series and images. In remote sensing, GANs generate completed pixel values for consistent landscape representations [20]. For change detection, such as riverbank erosion using Sentinel-1 imagery, GANs reduce sensitivity to unregistered objects in image pairs [30]. Temporal Generative Adversarial Networks (TGAN) are particularly suited for gap-filling Sentinel-1 SAR images in riverbank erosion studies. TGAN employs LSTM layers to process spatiotemporal dependencies, producing natural-looking sequence outputs. Unlike Pix2Pix and CycleGAN, which focus on image-to-image translation, TGAN prioritizes time-series data [16]. TGAN's temporal generator creates latent variables that evolve over time, while the image generator maps these to visual outputs, ensuring temporally coherent frames for applications like video synthesis and medical imaging [18], [46]. Its sequence-based discrimination mitigates mode collapse, generating diverse, authentic outputs [6]. In contrast, SRGAN excels at super-resolution but lacks temporal coherence, leading to inconsistencies in environmental monitoring [17].

Advanced GAN-based methods like STA-GAN and GANFilling further enhance gap-filling capabilities. STA-GAN uses a spatio-temporal attention mechanism to reconstruct missing data in MODIS time series, effectively capturing spatial and temporal dynamics [50]. GANFilling, employing a convolutional LSTM-based GAN, restores cloud-covered Sentinel-2 imagery, improving metrics like SSIM and PSNR and supporting applications such as vegetation index forecasting [67].

2.5 Deep Learning Models for Spatial and Temporal Data Processing

With the recent emergence of deep learning models like SRGAN (super-resolution generative adversarial networks) and LSTM (long short-term memory networks), it is easier to process temporal and spatial data for addressing data gaps. Spatial LSTMs and SRGANs have shown strong results in resolving spatial data and enhancing spatial data resolution respectively. Advanced GAN architecture (SRGAN) can create high-resolution (HR) images from low-resolution (LR) inputs. SRGAN consists of a generator (G) that produces realistic HR images and a discriminator (D) to ensure the fidelity of generated results. For example, [17] experimentally demonstrated that SRGAN can generate photorealistic HR images using a perceptual loss function that favors image quality over traditional pixel-based loss metrics. This method improve the resolution of spatial datasets with cloud cover or sensor limitations. Another study proposed an enhanced SRGAN model for reconstructing high-resolution climate data from degraded inputs [22]. LSTM networks, introduced by Hochreiter and Schmidhuber [4], are widely used for time series prediction tasks and have improved various applications, including stock market prediction and environmental monitoring. One example of using LSTM networks with higher-resolution data from SRGAN is given in Fang et al. [28], who combined LSTM networks with SRGAN to increase flood prediction accuracy. Another significant application of LSTMs is predicting crop yield variations using multi-temporal satellite data, which highlights the importance of integrating both temporal and spatial modeling [32].

2.6 Data Gap Filling and Segmentation

A number of recent studies have utilized deep learning models to address challenges in data gap filling and semantic segmentation, particularly using Sentinel-1 imagery and SAR data. One notable method, STA-GAN, proposes a spatio-temporal attention mechanism within a GAN framework to reconstruct missing satellite data. By capturing both spatial dependencies and temporal dynamics, this model successfully imputes large gaps in MODIS time series, making it a relevant approach for environmental monitoring applications [50]. The GANFilling model presents another powerful solution for cloud-induced data gaps in Sentinel-2 sequences. By using a convolutional LSTM-based GAN, it restores cloud-covered imagery and maintains temporal consistency across image sequences. This architecture not only improves image quality metrics like SSIM and PSNR but also supports downstream applications such as vegetation index forecasting [67]. In the context of SAR-based segmentation, DeepAqua demonstrates a self-supervised framework that learns water segmentation from SAR data without manual labels. It utilizes NDWI from optical imagery as a pseudo-labeling strategy, enabling the network to distinguish between open and vegetated water bodies. This method is particularly beneficial for wetland monitoring during periods when optical data are unavailable [49]. The MERLIN-Seg approach combines self-supervised despeckling with semantic segmentation to improve SAR image interpretation. It learns meaningful representations from speckled SAR images and leverages these features to perform segmentation with minimal labeled data. This method proves especially useful for high-resolution SAR imagery where annotation is costly [65]. To address the limitations of supervised learning in remote sensing, the SSL Vision Transformer (SSL ViT) introduces contrastive self-supervised pretraining on large satellite image datasets. This approach enables the model to extract generalizable features that enhance segmentation accuracy even with limited labeled data. The pretrained transformer backbone significantly boosts performance in downstream land-cover segmentation tasks [41].

Chapter 3

Tools and Techniques

3.1 VV Polarization in Synthetic Aperture Radar (SAR)

Synthetic Aperture Radar sends or receives vertical polarization using the VV Polarization method to detect particular surfaces including roughness determining and vegetation monitoring [1].

Mathematical Representation

In SAR systems, polarization combinations are denoted by letter pairs that represent transmitted or received polarizations. In VV Polarization, the V refers to the vertically oriented electric field used. The VV polarization received signal is expressed as: **VV polarization**: - **V (Vertical polarization)** refers to the electric field being oriented vertically.
The received signal in VV polarization can be expressed as:

$$S_{\mathrm{VV}} = \int\int\int \sigma_{\mathrm{VV}}(\mathbf{r}) \cdot e^{jk\cdot r} d\mathbf{r} \tag{3.1}$$

Where:

- S_{VV}: The received signal in VV polarization.
- $\sigma_{\mathrm{VV}}(\mathbf{r})$: The backscattering coefficient in VV polarization at position $\mathbf{r}$.
- k: The wave number of the radar signal.
- $e^{jk\cdot r}$: The wave propagation term.
- $d\mathbf{r}$: The differential volume element in the target scene.

3.2 LSTM

LSTM network is a type of RNN that models temporal sequences and captures patterns of dependencies across the dataset more effectively than traditional RNNs. The vanishing gradient problem is handled by LSTMs using memory cells and gating mechanisms [4].

LSTM Components and Mathematical Formulas

An LSTM unit consists of the following components:

- **Input Gate:** Controls the extent to which new information flows into the cell state.
- **Forget Gate:** Determines the amount of information from the previous cell state to retain.
- **Output Gate:** Regulates the information passed to the next hidden state.
- **Cell State:** Acts as a memory that accumulates useful information over time [63].

Mathematical Formulation

Given an input sequence x_t, the LSTM performs the following computations at each time step t:

1. Input Gate Activation

$$i_t = \sigma(W_i \cdot x_t + U_i \cdot h_{t-1} + b_i) \tag{3.2}$$

Description: Computes the input gate activation [61][56] i_t by applying a sigmoid function σ to the weighted sum of the current input x_t, the previous hidden state h_{t-1}, and a bias term b_i.

2. Forget Gate Activation

$$f_t = \sigma(W_f \cdot x_t + U_f \cdot h_{t-1} + b_f) \tag{3.3}$$

Description: Computes the forget gate activation f_t using a sigmoid function applied to the weighted sum of x_t, h_{t-1}, and a bias term b_f [52].

3. Output Gate Activation

$$o_t = \sigma(W_o \cdot x_t + U_o \cdot h_{t-1} + b_o) \tag{3.4}$$

Description: Determines the output gate activation [61] o_t by applying a sigmoid function to the weighted sum of x_t, h_{t-1}, and a bias term b_o.

4. Cell Input Activation

$$\tilde{c}_t = \tanh(W_c \cdot x_t + U_c \cdot h_{t-1} + b_c) \tag{3.5}$$

Description: Computes the candidate cell state $\tilde{c}_t$ by applying the hyperbolic tangent function tanh to the weighted sum of x_t, h_{t-1}, and a bias term b_c [55].

5. Cell State Update

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{3.6}$$

Description: Updates the cell state c_t by combining the previous cell state c_{t-1} (modulated by the forget gate f_t) with the candidate cell state $\tilde{c}_t$ (modulated by the input gate i_t). The symbol $\odot$ denotes element-wise multiplication.

6. Hidden State Update

$$h_t = o_t \odot \tanh(c_t) \tag{3.7}$$

Description: Computes the current hidden state h_t by applying the output gate o_t to the hyperbolic tangent of the updated cell state c_t [62].

3.3 Fully Connected Layers

A Fully Connected (FC) Layer, also known as a Dense Layer, is a fundamental component in neural networks. It connects every input neuron to every output neuron, enabling the model to learn complex, non-linear mappings between the input and output [51][2].

Mathematical Representation

The operation of a Fully Connected Layer can be expressed as a linear transformation followed by an activation function [61]:

$$\mathbf{z} = W\mathbf{x} + \mathbf{b} \tag{3.8}$$

$$\mathbf{y} = f(\mathbf{z}) \tag{3.9}$$

where:

- $\mathbf{x} \in \mathbb{R}^{n_{\text{in}}}$: Input vector of size n_{in}.
- $W \in \mathbb{R}^{n_{\text{out}} \times n_{\text{in}}}$: Weight matrix mapping the input space to the output space [61].
- $\mathbf{b} \in \mathbb{R}^{n_{\text{out}}}$: Bias vector added to the weighted input.
- $\mathbf{z} \in \mathbb{R}^{n_{\text{out}}}$: Intermediate output after the linear transformation.
- $f(\cdot)$: Activation function applied element-wise to introduce non-linearity.
- $\mathbf{y} \in \mathbb{R}^{n_{\text{out}}}$: Final output vector.

Activation Functions

Activation functions introduce non-linearity into the model, allowing it to learn more complex patterns. Common activation functions include:

- **ReLU:** $f(z) = \max(0, z)$
- **Sigmoid:** $f(z) = \frac{1}{1+e^{-z}}$
- **Tanh:** $f(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$

3.4 Batch Normalization (BatchNorm1D)

Batch Normalization is applied along the feature dimension for a batch of inputs. It normalizes the activations across a mini-batch and introduces trainable parameters to restore representation capacity [8].

Formula

Given an input feature vector $x \in \mathbb{R}^{N \times F}$, where N is the batch size and F is the number of features, Batch Normalization is computed as:

$$\hat{x}_i = \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}} \tag{3.10}$$

where:

- $\mu = \frac{1}{N} \sum_{i=1}^{N} x_i$: Mean of the mini-batch.
- $\sigma^2 = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)^2$: Variance of the mini-batch.
- ϵ: Small constant added for numerical stability.
- $\hat{x}_i$: Normalized feature.

The normalized value is scaled and shifted using trainable parameters γ (scale) and β (shift):

$$y_i = \gamma \hat{x}_i + \beta \tag{3.11}$$

- **Condition**: BatchNorm1D is applied when the batch size $N > 1$.

3.5 Instance Normalization (InstanceNorm2D)

Instance Normalization is designed for cases where batch normalization is not effective, such as when the batch size is small (e.g., $N = 1$) or in style transfer tasks. Instead of normalizing across the batch, it normalizes each sample independently [14].

Formula

Given an input tensor $x \in \mathbb{R}^{N\times C\times H\times W}$, where C is the number of channels and H, W are the spatial dimensions, Instance Normalization is computed as:

$$\hat{x}_{n,c} = \frac{x_{n,c} - \mu_{n,c}}{\sqrt{\sigma_{n,c}^2 + \epsilon}} \tag{3.12}$$

where:

- $\mu_{n,c} = \frac{1}{H\times W}\sum_{h=1}^{H}\sum_{w=1}^{W} x_{n,c,h,w}$: Mean of the spatial dimensions for each sample and channel.
- $\sigma_{n,c}^2 = \frac{1}{H\times W}\sum_{h=1}^{H}\sum_{w=1}^{W} (x_{n,c,h,w} - \mu_{n,c})^2$: Variance over spatial dimensions.

Similar to BatchNorm, it includes learnable parameters:

$$y_{n,c} = \gamma \hat{x}_{n,c} + \beta \tag{3.13}$$

Condition: InstanceNorm2D is used when $N = 1$, as it avoids instability caused by BatchNorm1D in small batch sizes.

3.6 Transposed Convolutional Layers

Transposed convolutional layers, also referred to as **deconvolutions** or **fractionally strided convolutions**, are primarily used in deep learning for tasks involving upsampling of feature maps. These layers are essential in applications such as image generation, super-resolution and semantic segmentation [59][11].

Mathematical Formulation

Given an input tensor X with spatial dimensions $H_{\text{in}}\times W_{\text{in}}$, the output dimensions $H_{\text{out}}\times W_{\text{out}}$ of a transposed convolutional layer are computed as [61]:

$$H_{\text{out}} = (H_{\text{in}} - 1)\cdot S - 2P + K + \text{dilation}\cdot(K-1) \tag{3.14}$$

$$W_{\text{out}} = (W_{\text{in}} - 1)\cdot S - 2P + K + \text{dilation}\cdot(K-1) \tag{3.15}$$

where:

- S: Stride, controlling the step size of kernel application.
- P: Padding, determining the number of pixels added around the input tensor.
- K: Kernel size, defining the height and width of the convolution filter.
- dilation: Dilation rate, specifying the spacing between kernel elements.

Process Description

1. **Input Expansion**: The input tensor is spatially expanded by introducing gaps (or zeros) between input values based on the stride S and dilation settings.
2. **Kernel Application**: A convolutional kernel is applied over the expanded input tensor, generating a larger output tensor.
3. **Reconstruction**: Overlapping regions are aggregated (summed) to produce the final output tensor.

3.7 Conv2D

The 2D Convolutional Layer (Conv2D) is a fundamental building block in deep learning models, particularly for tasks involving spatial data such as images. Conv2D performs spatial feature extraction by applying learnable filters (kernels) across an input tensor.[12]

Mathematical Description

Let the input to the Conv2D layer be represented as a 3D tensor $\mathbf{I} \in \mathbb{R}^{H \times W \times C_{\text{in}}}$, where H, W, and C_{in} denote the height, width, and number of input channels, respectively. A convolution operation applies a set of C_{out} learnable kernels (filters)[48] $\mathbf{K} \in \mathbb{R}^{k_H \times k_W \times C_{\text{in}} \times C_{\text{out}}}$ to the input [61].
The output feature map $\mathbf{O} \in \mathbb{R}^{H_{\text{out}} \times W_{\text{out}} \times C_{\text{out}}}$ is computed as:

$$\mathbf{O}(x, y, c_{\text{out}}) = \sum_{c_{\text{in}}=1}^{C_{\text{in}}} \sum_{i=1}^{k_H} \sum_{j=1}^{k_W} \mathbf{K}(i, j, c_{\text{in}}, c_{\text{out}}) \cdot \mathbf{I}(x + i, y + j, c_{\text{in}}) + b(c_{\text{out}}), \tag{3.16}$$

where:

- x, y: Spatial position in the output feature map.[61]
- c_{in}: Index of the input channel.
- c_{out}: Index of the output channel.
- k_H, k_W: Height and width of the kernel.
- $b(c_{\text{out}})$: Learnable bias for the output channel c_{out}.

Output Dimensions

The spatial dimensions of the output feature map H_{out} and W_{out} are calculated as:

$$H_{\text{out}} = \left\lfloor \frac{H - k_H + 2P}{S_H} \right\rfloor + 1, \tag{3.17}$$

$$W_{\text{out}} = \left\lfloor \frac{W - k_W + 2P}{S_W} \right\rfloor + 1, \tag{3.18}$$

where:

- P: Padding size (number of pixels added to each side of the input).
- S_H, S_W: Stride values along height and width.

3.8 Temporal GAN

Temporal Generative Adversarial Networks(Temporal GANs) are used to generate frames or image data of time-series by capturing temporal dependency patterns across the sequence. This is achieved by combining CNNs, LSTMs and adversarial training principles [6]. A Temporal GAN consists of two primary components:

- **Temporal Generator (G):** Finds patterns within noise vectors to generate a data sequence.
- **Discriminator with Temporal Modeling (D):** Sends output of real vs fake sequences discrimination values by capturing temporal dependencies.

Temporal Generator

The generator produces the output sequences by using the noise vector as a latent representation [10].

Components

1. **Input Noise Vector:** A random noise vector $z \in \mathbb{R}^{(N \times T \times d)}$, where:

   - N: Batch size.
   - T: Sequence length.
   - d: Dimensionality of the noise vector.

   This noise vector serves as the latent representation from which the generator produces the output sequence [61].

2. **LSTM Layer:** Captures temporal dependencies by processing the input sequences and finding patterns [57]:

   $$h_t = \text{LSTM}(z_t, h_{t-1}) \tag{3.19}$$

   where h_t is the hidden state at time t. The LSTM enables the generator to model temporal coherence across frames [4].

3. **Fully Connected Layer:** Maps the LSTM output to spatial dimensions:

$$x_t = W_h \cdot h_t + b_h \tag{3.20}$$

   where W_h and b_h are learnable weights and biases.

4. **Transposed Convolutions:** Upsample the spatial features to the desired resolution:

$$\hat{x}_t = \text{ConvTranspose2D}(x_t) \tag{3.21}$$

   Transposed convolutions allow the generator to create high-resolution frames [11].

5. **Activation Function:** A tanh activation function ensures that the output values are scaled to $[-1, 1]$.

Generator Loss

The generator aims to fool the discriminator by minimizing the following loss [54]:

$$\mathcal{L}_G = \mathbb{E}_{z \sim p_z}[\log(1 - D(G(z)))] \tag{3.22}$$

Discriminator with Temporal Modeling

The discriminator distinguishes between real and generated sequences while ensuring temporal coherence [7].

Components

1. **Input:** A sequence of images $X \in \mathbb{R}^{(N \times T \times H \times W)}$, where H and W represent the height and width of each frame.

2. **Convolutional Layers:** Extract spatial features from each frame:

$$f_t = \text{Conv2D}(x_t) \tag{3.23}$$

3. **Flattening:** Spatial features are flattened to prepare for temporal modeling.

4. **LSTM Layer:** Captures temporal dependencies across the sequence:

$$h_t = \text{LSTM}(f_t, h_{t-1}) \tag{3.24}$$

5. **Fully Connected Layer:** Maps the final hidden state to a scalar logit:

$$y = W_o \cdot h_T + b_o \tag{3.25}$$

6. **Output:** A sigmoid activation provides the probability of the sequence being real:

$$p = \sigma(y) \tag{3.26}$$

Discriminator Loss

The discriminator minimizes the following adversarial loss [61]:

$$\mathcal{L}_D = -\mathbb{E}_{x\sim p_{\text{data}}}[\log D(x)] - \mathbb{E}_{z\sim p_z}[\log(1 - D(G(z)))] \tag{3.27}$$

Masked Autoencoder (MAE)

The Masked Autoencoder (MAE) is a self-supervised learning model that reconstructs images from partially masked inputs, leveraging a vision transformer architecture [43].

Components

1. **Input:** An image $X \in R^{H\times W\times C}$, where H, W, and C represent the height, width, and number of channels, respectively.

2. **Patching:** The image is divided into non-overlapping patches of size $P \times P$, yielding a sequence of $N = \frac{HW}{P^2}$ patches.

3. **Masking:** A subset of patches (e.g., 75%) is randomly masked, leaving a visible subset X_v.

4. **Encoder:** A Vision Transformer (ViT) processes only the visible patches:

$$z_v = \text{ViT-Encoder}(X_v) \tag{3.28}$$

5. **Decoder:** Reconstructs the full image from encoded visible patches, with mask tokens for missing patches:

$$\hat{X} = \text{ViT-Decoder}(z_v, \text{MaskTokens}) \tag{3.29}$$

6. **Output:** The reconstructed image $\hat{X} \in^{H\times W\times C}$.

MAE Loss

The MAE minimizes the mean squared error between the reconstructed and original image pixels, computed only on masked patches:

$$\mathcal{L}_{\text{MAE}} = E_{X\sim p_{\text{data}}}\left[\frac{1}{|\Omega|}\sum_{i\in\Omega} \|X_i - \hat{X}_i\|^2\right] \tag{3.30}$$

where Ω is the set of masked patch indices.

Mamba

Mamba is a state-space model designed for efficient sequence modeling, offering linear-time complexity as an alternative to transformers .

Components

1. **Input:** A sequence $X \in^{T\times D}$, where T is the sequence length and D is the feature dimension.
2. **State-Space Model:** Models the sequence using a continuous-time linear state-space system, discretized for computation:
$$h_t = Ah_{t-1} + Bx_t, \quad y_t = Ch_t \tag{3.31}$$
where A, B, and C are learnable parameters, and $h_t \in^N$ is the hidden state.
3. **Selective Scan:** A selective mechanism modulates A and B based on input x_t, enabling content-aware processing:
$$A_t = A(x_t), \quad B_t = B(x_t) \tag{3.32}$$
4. **Convolutional Layer:** Optionally applies a 1D convolution to capture local dependencies before the state-space layer.
5. **Output:** The sequence of outputs $Y \in^{T\times D}$, computed via the state-space recurrence or parallel scan.

Mamba Loss

For tasks like language modeling, Mamba minimizes the cross-entropy loss:
$$\mathcal{L}_{\text{Mamba}} = -_{X\sim p_{\text{data}}}\left[\sum_{t=1}^{T} \log p(x_t|x_{<t})\right] \tag{3.33}$$

Chapter 4

Dataset

4.1 Data Collection

Radar data from Sentinel-1 delivers precise images that work in every weather condition, including beneath clouds. Our system acquires and processes Sentinel Hub API data to allow us easy access to Sentinel-1's satellite radar data.

1. Configuring Sentinel Hub Access

The script initializes access to Sentinel Hub using the instance ID and client credentials to let the Sentinel Hub API authenticate our service

2. Defining AOIs and Resolution

- Each AOI is specified by geographic coordinates.We drew a border around the river's geographic positions.
- A spatial resolution of **60 meters per pixel** is set to determine the size of the image grid.

3. Sentinel Hub Request

A request is created to fetch **Sentinel-1 SAR data** using VV polarization:

- An **evaluation script** processes the SAR backscatter data for visualization.
  - **Input:** VV polarization.
  - **Output:** Single-band grayscale images.
- Data is retrieved using the `mostRecent` mosaicking order for the specified date.

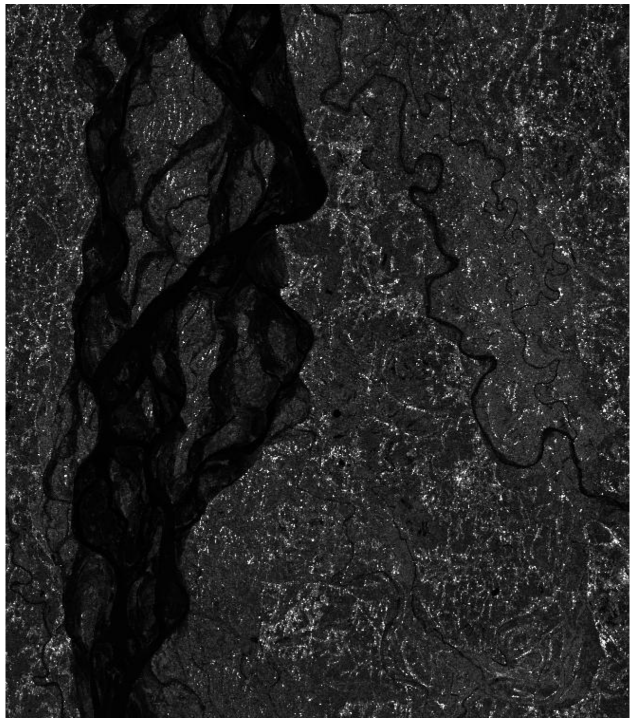


Figure 4.1: SAR Image of Brahmaputra River

4. Iterative Image Retrieval

- **Temporal Range:** The script loops through every day of the year from 2014 to 2024. A `datetime` object ensures proper day-by-day iteration.
- **Fetching Images:** For each AOI, a request is made for a specific date. If images are available, they are returned as a list of arrays.

5. Data Validation and Storage

- **Validation:** The script checks for non-zero pixels in the images to confirm the presence of data.Images without data are skipped. Extensive manual data cleaning was performed to retain only high-quality SAR images that accurately depict riverbank monitoring.
- **Storage:** The grayscale Images are saved in organized into year-based folders for each AOI.Filenames include the AOI name, date, and index for unique identification.

4.2 Overview

The detection of riverbank changes throughout Bangladesh depends on high-resolution satellite data supplied by equivalent temporal satellite content. Scientific data analysis for riverbank erosion patterns across Bangladesh's major waterways incorporates **Sentinel-1**

Synthetic Aperture Radar (SAR) imagery which covers the **Padma, Jamuna, Meghna, Brahmaputra, and Teesta Rivers**. Their constantly shifting nature combined with easy susceptibility to erosion positions these rivers as essential regions for environmental tracking purposes.

The time span of the data extends from **2015 through to 2024** and consists of sequential imagery acquired at specific riverbank sites distributed across these rivers. The dataset features **discontinuous temporal sequences** because factors such as satellite revisit limitations and inconsistent acquisition processes and cloud cover create gaps in time series. The dataset is structured to include:

- **Geographic Coverage**: Significant erosion areas exist at fundamental points situated near principal river routes.
- **Total Images Collected**: There exists a range of 468 to 716 Sentinel-1 SAR images for each study site.
- **Temporal Gaps**: The temporal data gaps in the dataset span between 5 days to significant periods of 7.7 days based on specific site locations and orbital pass schedules.

4.2.1 Padma River Dataset

The Padma River is one of the major rivers of Bangladesh, experiencing significant riverbank erosion. The dataset collected for this study includes Sentinel-1 Synthetic Aperture Radar (SAR) imagery from 2015 to 2024, covering key locations along the Padma River.

Study Locations and Data Availability

Three key locations along the Padma River were selected for this study:

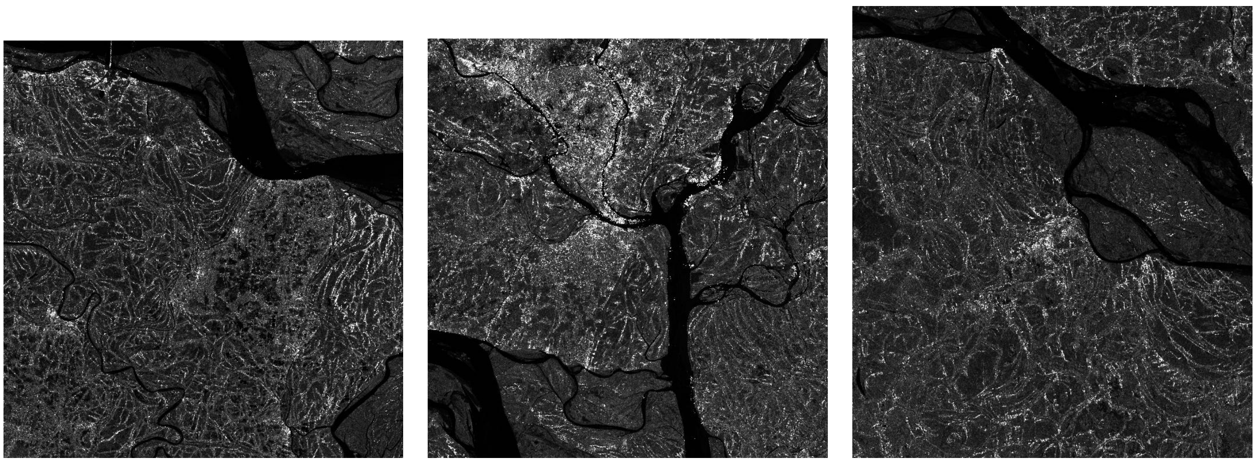
Year	Shariatpur	Munshiganj	Faridpur
2015	11	21	10
2016	15	28	15
2017	51	72	51
2018	59	86	59
2019	58	73	60
2020	65	92	65
2021	60	88	60
2022	59	87	59
2023	50	78	50
2024	45	69	45
Total	473	694	474

Table 4.1: Yearly Image Distribution for Padma River Locations

Temporal Gaps and Data Challenges

Due to inconsistent satellite passes and environmental conditions, some temporal gaps exist in the dataset. The average data gaps for the Padma River locations are:

- **Shariatpur**: 7.60 days
- **Munshiganj**: 5.20 days
- **Faridpur**: 7.58 days



(a) Shariatpur (b) Munshiganj (c) Faridpur

Figure 4.2: Padma River

4.2.2 Jamuna River Dataset

The Jamuna River is one of the most dynamic rivers in Bangladesh, experiencing continuous erosion and morphological changes. This study utilizes Sentinel-1 Synthetic Aperture Radar (SAR) imagery from 2015 to 2024 to monitor riverbank shifts along key locations.

Study Locations and Data Availability

Three key locations along the Jamuna River were selected for analysis:

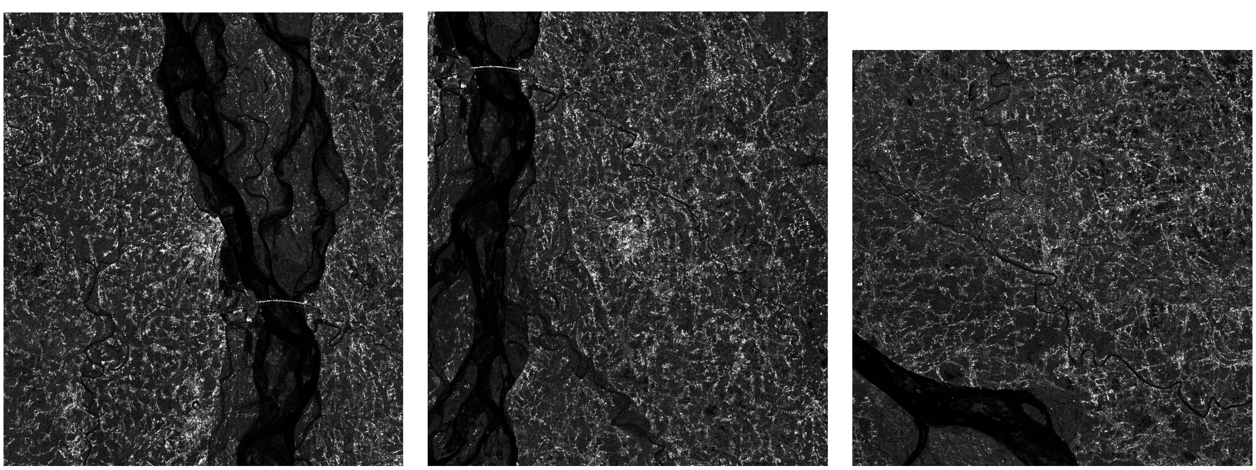
Year	Sirajganj	Tangail	Manikganj
2015	10	10	10
2016	13	13	15
2017	50	51	51
2018	59	59	59
2019	57	58	60
2020	65	65	65
2021	60	60	60
2022	59	59	59
2023	50	50	50
2024	45	45	45
Total	468	470	474

Table 4.2: Yearly Image Distribution for Jamuna River Locations

Temporal Gaps and Data Challenges

The average data gaps for the Jamuna River locations are:

- **Sirajganj**: 7.68 days
- **Tangail**: 7.65 days
- **Manikganj**: 7.58 days



(a) Sirajganj (b) Tangail (c) Manikganj

Figure 4.3: Jamuna River

4.2.3 Meghna River Dataset

The Meghna River is one of the largest and most erosion-prone rivers in Bangladesh. Due to its dynamic hydrological nature, continuous monitoring is essential. This study utilizes

Sentinel-1 Synthetic Aperture Radar (SAR) imagery from 2015 to 2024 to analyze riverbank changes along the Meghna River.

Study Locations and Data Availability

Three key locations along the Meghna River were selected for analysis:

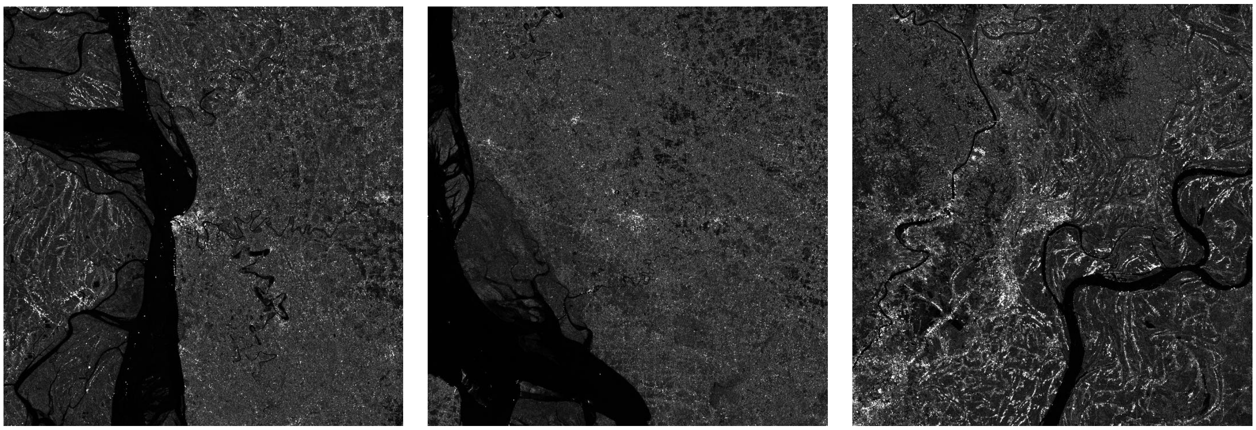
Year	Chandpur	Lakshmipur	Narsingdi
2015	22	22	21
2016	28	28	27
2017	72	72	72
2018	86	86	86
2019	72	72	73
2020	92	92	92
2021	88	88	88
2022	87	87	87
2023	78	78	78
2024	69	69	69
Total	694	694	693

Table 4.3: Yearly Image Distribution for Meghna River Locations

Temporal Gaps and Data Challenges

Satellite revisit cycles and environmental conditions lead to data inconsistencies. The average data gaps for the Meghna River locations are:

- **Chandpur**: 5.20 days
- **Lakshmipur**: 5.20 days
- **Narsingdi**: 5.21 days



(a) Chandpur (b) Lakshmipur (c) Narsingdi

Figure 4.4: Meghna River

4.2.4 Brahmaputra River Dataset

The Brahmaputra River is one of the most significant transboundary rivers in Bangladesh, experiencing continuous morphological changes due to sediment transport and erosion. To monitor these changes, this study utilizes Sentinel-1 Synthetic Aperture Radar (SAR) imagery from 2015 to 2024.

Study Locations and Data Availability

Three key locations along the Brahmaputra River were selected for analysis:

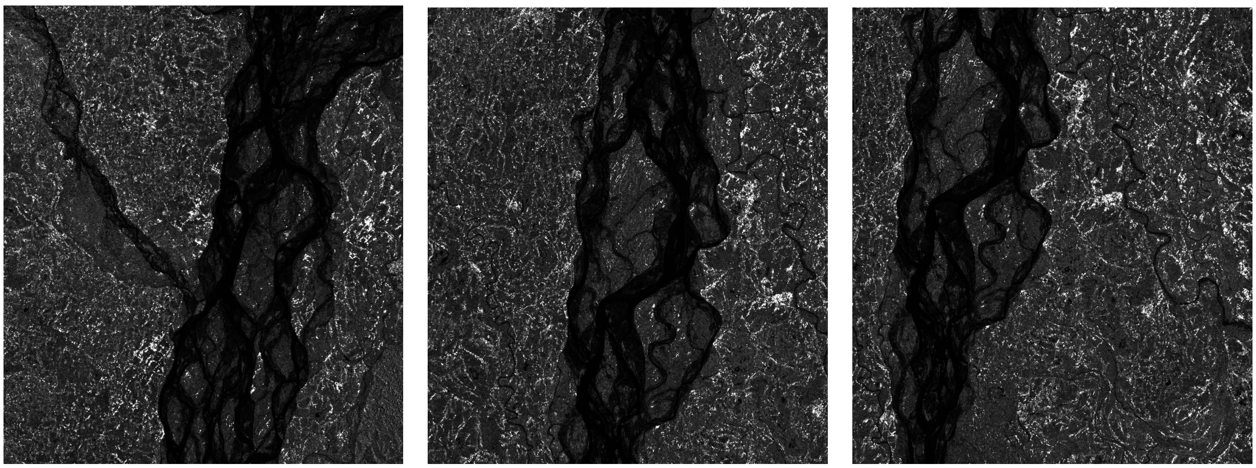
Year	**Chilmari**	**Fulchhari**	**Bahadurabad**
2015	23	23	12
2016	26	26	13
2017	74	74	49
2018	85	85	59
2019	81	81	57
2020	92	92	65
2021	91	91	60
2022	86	86	59
2023	75	75	50
2024	69	69	45
Total	702	702	469

Table 4.4: Yearly Image Distribution for Brahmaputra River Locations

Temporal Gaps and Data Challenges

The average data gaps for the Brahmaputra River locations are:

- **Chilmari**: 5.14 days
- **Fulchhari**: 5.14 days
- **Bahadurabad**: 7.66 days



(a) Chilmari (b) Fulchhari (c) Bahadurabad

Figure 4.5: Brahmaputra River

4.2.5 Teesta River Dataset

The Teesta River is a crucial river in northern Bangladesh, heavily influenced by seasonal variations and sedimentation. Due to its dynamic hydrological nature, continuous monitoring is essential to assess erosion and riverbank shifts. This study utilizes Sentinel-1 Synthetic Aperture Radar (SAR) imagery from 2015 to 2024 to analyze riverbank changes along the Teesta River.

Study Locations and Data Availability

Two key locations along the Teesta River were selected for analysis:

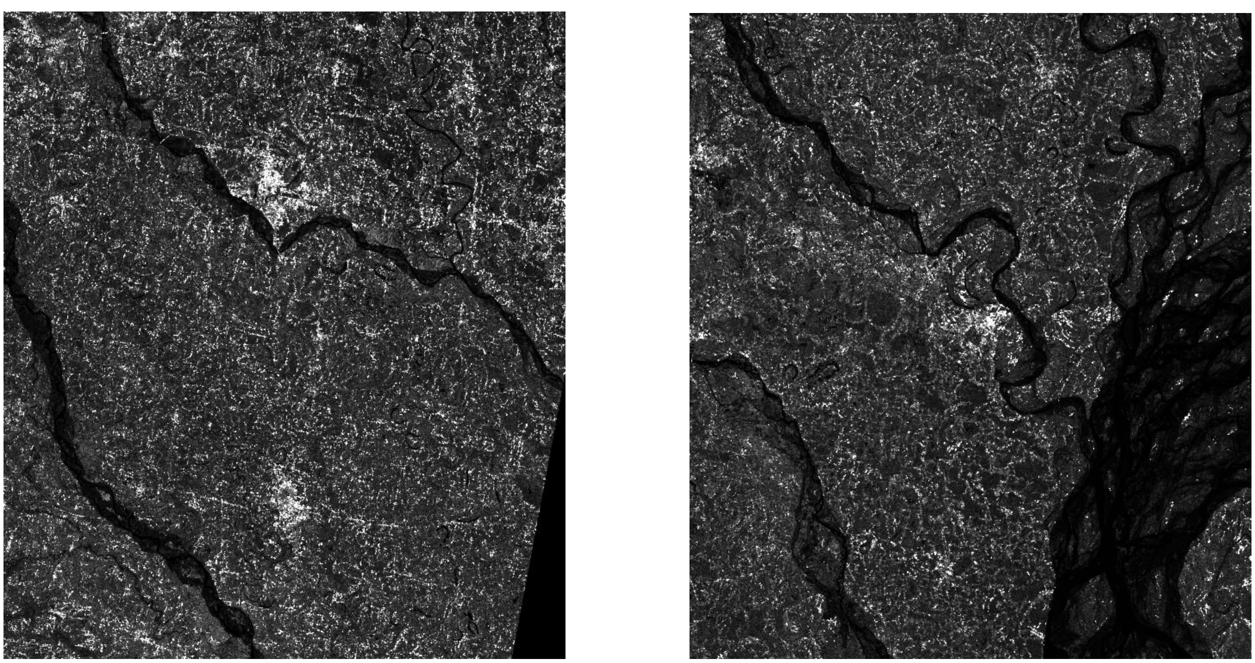
Year	Lalmonirhat	Kurigram
2015	23	23
2016	26	26
2017	75	74
2018	87	87
2019	87	86
2020	93	93
2021	91	91
2022	86	86
2023	78	78
2024	70	70
Total	716	714

Table 4.5: Yearly Image Distribution for Teesta River Locations

Temporal Gaps and Data Challenges

The average data gaps for the Teesta River locations are:

- **Lalmonirhat**: 5.04 days
- **Kurigram**: 5.05 days



(a) Lalmonirhat (b) Kurigram

Figure 4.6: Teesta River

4.3 Summary

This section provides an overview of the total images collected for each river and their respective average data gaps. The dataset consists of Sentinel-1 SAR imagery acquired from 2015 to 2024, covering key locations along the major rivers in Bangladesh.

4.3.1 Total Image Count

The table below summarizes the total number of Sentinel-1 images collected for each river:

River	Total Images Collected
Padma	1,641
Jamuna	1,412
Meghna	2,081
Brahmaputra	1,873
Teesta	1,430
Total	**8,437**
Overall Average Per River	**1,687**

Table 4.6: Total Images Collected for Each River

4.3.2 Average Data Gap Analysis

The average data gap (in days) for each river is summarized in the table below:

River	Average Data Gap (days)
Padma	6.79
Jamuna	7.63
Meghna	5.20
Brahmaputra	6.65
Teesta	5.05
Overall Average	**6.26**

Table 4.7: Average Data Gaps for Each River

4.4 Data Availability and Temporal Gaps

Sentinel-1 has a specific revisit schedule that causes inconsistency in acquiring image through time at different locations. The major issues faced include:

- **Irregular Satellite Revisit Cycles:** The revisit period of sentinel-1 varies due to orbit overlap which causes inconsistent data availability for the desired locations.

- **Cloud Cover and Atmospheric Interference:** Although SAR imagery is not affected by clouds, there are certain environmental conditions that can still impact image clarity and quality.
- **Missing Data:** Some expected timestamps lacked corresponding images due to satellite acquisition failures or gaps in data retrieval.

4.5 Dataset Preparation

This study utilizes Sentinel-1 Synthetic Aperture Radar (SAR) imagery from the Jamuna–Sirajganj region of Bangladesh, captured during the year 2015. The dataset is systematically organized to support the SAR2SAR [33] despeckling and segmentation pipeline. It comprises three core components:

- **Original SAR Images:**
  - Located in the `Riverbank Sentinel-1/original` directory.
  - Grayscale intensity images with speckle noise, characteristic of SAR acquisitions.
  - Serve as the input for the SAR2SAR despeckling network [33].
- **Denoised Images:**
  - Generated using a semi-supervised deep learning model inspired by the SAR2SAR framework[33].
  - Output files are stored in the `Riverbank Sentinel-1/denoised` directory.
  - The model architecture is a modified U-Net, optimized for SAR despeckling:
    * *Pretraining:* Conducted on synthetic speckle-corrupted images using additive and multiplicative noise models.
    * *Fine-tuning:* Performed on real multi-temporal Sentinel-1 SAR stacks with temporal consistency constraints to suppress changes unrelated to speckle.
  - Tailored to reduce speckle noise while preserving fine spatial details and structural boundaries crucial for downstream riverbank analysis.

Motivation for SAR2SAR

Speckle noise in SAR imagery presents significant challenges for interpretation and downstream analysis. The SAR2SAR algorithm is employed due to the following advantages:

- **No Need for Clean Labels:** Operates without requiring noise-free ground truth, aligning with the realities of SAR data collection.
- **Semi-Supervised Learning:** Incorporates a three-phase training strategy:
  1. Supervised training using synthetically speckled data.

2. Self-supervised fine-tuning on real SAR sequences with temporal consistency checks.
3. Iterative refinement for better generalization and robustness.

- **Preservation of Image Structure:** Balances speckle suppression with edge and texture retention, which is crucial for accurate segmentation.

Workflow Summary

The full pipeline follows a structured progression:

1. Acquisition of noisy SAR imagery from Sentinel-1.
2. Despeckling using the SAR2SAR model.
3. Segmentation using unsupervised techniques on denoised outputs.

This organization supports reproducible research and scalable analysis in riverbank monitoring and other SAR-based remote sensing applications.

Chapter 5

Methodology

5.1 Synthetic Image generation

A Temporal Generative Adversarial Network (Temporal GAN) serves as the backbone of the proposed framework for generating synthetic Synthetic Aperture Radar (SAR) images to address missing data in sequential grayscale Sentinel-1 satellite imagery (2014–2024) capturing riverbank changes. The framework leverages adversarial training between a generator (TG) and a discriminator (TD) to produce temporally consistent synthetic SAR images while preserving precise spatial features. Below, we outline the preprocessing steps and the Temporal GAN architecture, including its training dynamics and output generation.

Preprocessing Steps

To prepare the dataset of sequential grayscale Sentinel-1 satellite images for training, a custom dataset class was developed to handle all preprocessing steps. The preprocessing includes the following:

1. **Resizing:** All the images are resized to 128×128 pixels uniform resolution that ensures consistency and compatibility of the dataset with model's input requirements.

2. **Normalization:** To stablize training, pixel values are normalized to $[-1, 1]$ range that aligns with the model's expected input distribution.

3. **Dataset Splitting:** The dataset is divided into train-test-validation sets with 90:5:5 ratio respectively. This split ensures sufficient data for training while reserving portions for model evaluation and hyperparameter tuning.

The preprocessed dataset is then used to train the Temporal GAN over 20 epochs with a batch size of 11, resulting in 11 batches per epoch.

Temporal GAN Framework

The Temporal GAN framework employs adversarial development between the generator (TG) and discriminator (TD) to create authentic synthetic SAR image sequences. The following components and steps define the architecture and training process:

1. **Input and Noise Distribution:**
    - **Input Data:** The model takes sequential grayscale Sentinel-1 SAR images as input, representing riverbank changes over time.
    - **Noise Vectors:** Six noise vectors, each sampled from a standard normal distribution $N(0, 1)$ with 100 dimensions, are fed into the generator to introduce variability and enable the generation of diverse synthetic sequences.
    - **Process:** The generator processes these noise vectors alongside temporal context to produce synthetic image sequences that mimic real SAR data.

2. **Generator (TG):**
    - **Architecture:** The generator integrates Long Short-Term Memory (LSTM) layers to capture temporal dependencies in the sequential data and convolutional layers to analyze and generate spatial features.
    - **Functionality:** The generator outputs synthetic SAR images designed to closely resemble real-world data. It learns by minimizing errors identified by the discriminator, improving its ability to produce realistic sequences over time.
    - **Objective:** The generator aims to "fool" the discriminator by generating outputs that are indistinguishable from authentic SAR images.

3. **Discriminator (TD):**
    - **Architecture:** The discriminator processes both real and synthetic image sequences, evaluating their authenticity.
    - **Functionality:** It assigns labels to sequences, marking real sequences as 1 and fake sequences as 0. Through training, the discriminator optimizes its Binary Cross-Entropy (BCE) loss to improve its ability to distinguish genuine sequences from synthetic ones.
    - **Objective:** The discriminator seeks to maximize BCE loss for synthetic sequences while minimizing it for real sequences, enhancing its discriminative power.

4. **Adversarial Training:**
    - **Process:** The generator and discriminator are trained adversarially, with each component optimizing its performance to outperform the other. The generator improves its ability to produce realistic sequences, while the discriminator refines its classification accuracy.
    - **Loss Functions:**
        - **Generator Loss (G-loss):** The generator minimizes BCE loss by encouraging the discriminator to misclassify synthetic sequences as real, thereby improving the realism of its outputs.

- **Discriminator Loss (D-loss):** The discriminator minimizes BCE loss by correctly distinguishing real sequences from synthetic ones, sharpening its discriminative capabilities.

- **Outcome:** This adversarial process fosters the generation of highly realistic temporal sequences that generalize well to unseen data.

5. **Optimization and Training Dynamics:**

   - **Optimizer:** The Adam optimization algorithm is used to train both the generator and discriminator, with a learning rate of 0.0002, $\beta_1 = 0.5$, and $\beta_2 = 0.999$. These hyperparameters ensure stable and efficient training dynamics.
   - **Training Stability:** The selected settings balance the adversarial process, preventing mode collapse and promoting steady convergence over the 20 epochs.
   - **Batch Processing:** With a batch size of 11, the model processes the dataset in 11 batches per epoch, allowing for frequent weight updates and iterative refinement.

6. **Output:**

   - **Synthetic Sequences:** The generator progressively produces high-resolution synthetic SAR image sequences that fill gaps in the original dataset. These sequences maintain temporal coherence and spatial accuracy, enabling robust monitoring of riverbank changes.
   - **Applications:** The synthetic images support analysis under challenging environmental conditions, such as cloud cover or missing data, enhancing the reliability of long-term riverbank monitoring.

This novel framework combines rigorous preprocessing with a sophisticated Temporal GAN, uniquely integrating LSTM and convolutional layers to generate synthetic SAR imagery that addresses data gaps while preserving the temporal and spatial characteristics of Sentinel-1 satellite images. The adversarial training process, supported by optimized hyperparameters and a custom dataset class, ensures the production of authentic, high-resolution sequences suitable for monitoring riverbank changes from 2014 to 2024.

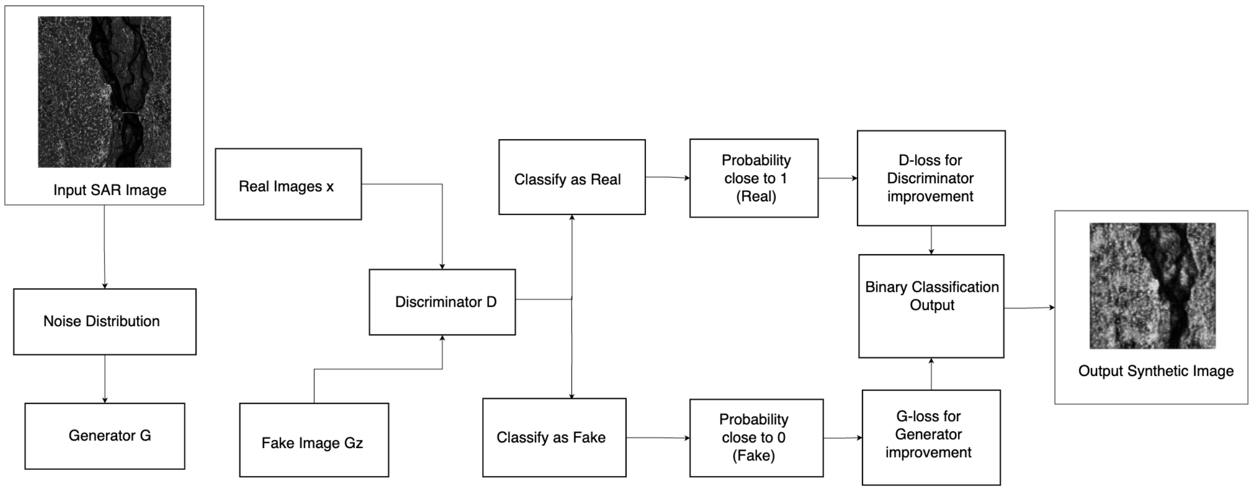


Figure 5.1: Top level overview of Synthetic Image generation

5.2 Architecture Design

The Temporal Generator (TG) takes random noise as input and generates a sequence of images, while the Temporal Discriminator (TD) detects whether the generated sequence is real or synthetic. Due to its ability to learn to capture temporal dependency in data using a with a Long Short Term Memory (LSTM) network, the designed TG effectively mitigates the harmful effects of input noise. The TG can generate sequences with coherence through time. output of the LSTM is re-shaped and then feeds through a bunch of upsampling layers composed of fully connected layers, batch normalisation and transposed convolutional layers. Together, these layers upscale the feature maps to produce spatially cohesive images of 128x128 pixels. Interpretable activations are used on intermediate layers with a Tanh activation for the output layer to constrain pixel intensities within the range of [-1,1]. It construction of sequences using 6 consecutive images from each image which is making a sequence.

Temporal Generator(TG): LSTM → FC → Transposed Convs → Sequence of Images

The TD is also designed in parallel for evaluating the authenticity of generated image sequences. Using convolutional layers, it processes individual frames and aggregates such features over time with an LSTM network. The scalar representation produced by the TD is fed into the LSTM, which represents all of the sequence. Then, we represent this and pass through a fully connected, then a Sigmoid activation function, allowing us to classify sequences as real or fake.

Temporal Discriminator(TD): Convs →LSTM → FC → Classification(fake/Real)

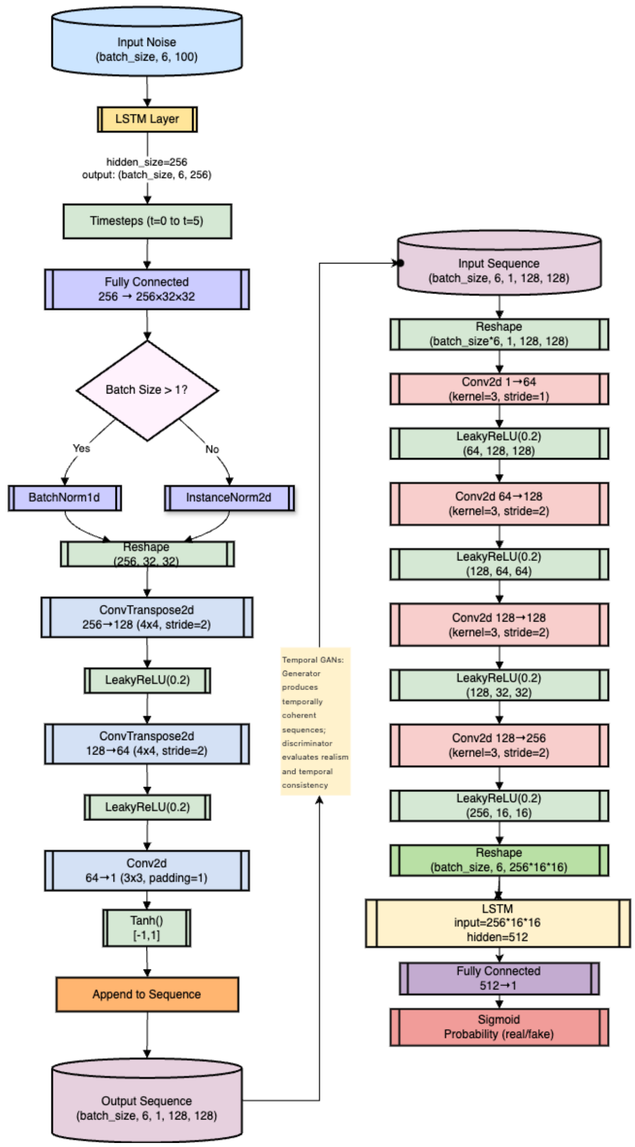


Figure 5.2: Architecture Design for generating Synthetic Image

Semantic Segmentation of SAR Images for Land and Water Boundary Detection

This section presents a self-supervised learning (SSL) framework for segmenting Synthetic Aperture Radar (SAR) images to delineate land and water boundaries, focusing on river systems. The methodology leverages a hybrid deep learning model combining a convolutional Masked Autoencoder (MAE) and simplified Mamba blocks to address the challenge of limited labeled data. The pipeline integrates data preparation, pseudo-label generation, model training, evaluation, and semantic overlay generation, ensuring robust segmentation for environmental monitoring.

The SSL framework follows a structured pipeline to achieve accurate land-water segmentation:

1. **Input and Data Preparation:** A dataset of 7,050 grayscale SAR images, including synthetic sequences from a Temporal GAN, is preprocessed to a uniform resolution of 256×256 pixels and normalized to $[0, 1]$.

2. **Pseudo-Label Generation:** Pseudo-labels are generated using robust thresholding and morphological operations to approximate land-water masks for unlabeled data.

3. **Hybrid Model (MAE-Mamba):** A hybrid model, combining a convolutional MAE encoder and simplified Mamba blocks, extracts spatial and sequential features for segmentation.

4. **Training and Optimization:** The model is trained using pseudo-labels, optimized with Adam and a step learning rate scheduler, employing mixed precision for efficiency.

5. **Evaluation and Metrics:** Performance is evaluated using loss, IoU, Dice, precision, recall, F1, and accuracy on validation and test sets.

6. **Output:** Semantic overlays visualize land (green) and water (blue) regions, supporting river boundary monitoring.

Data Preparation and Pseudo-Label Generation

Our evaluation was performed on our dataset Sentinel-1 SAR images collected from 2015 to 2024, encompassing five major rivers in Bangladesh, each with two to three distinct monitoring locations. These images were initially organized into folders, where each folder represented a unique River-Location-Year combination, capturing diverse spatial and temporal variability. To construct training, validation, and test sets, we applied a stratified splitting strategy at the folder level: within each folder, images were randomly shuffled and divided into 70% training, 15% validation, and 15% test subsets. This approach ensures that each split maintains proportional representation from all rivers, locations, and years, preserving the diversity of the dataset while preventing data leakage. Although a stricter evaluation method—such as holding out entire rivers or years—could better assess generalization, our current split offers a balanced and realistic baseline reflective of typical deployment scenarios.

Pseudo-labels are generated to enable SSL training:

- **Input Processing:** Images are converted to decibel scale using $10 \log_{10}(\text{data} + 10^{-10})$, clipped to $[-40, 10]$.

- **Thresholding:** A threshold is computed as the minimum of Otsu's threshold and the 30th percentile of pixel intensities.

- **Morphological Operations:** Gaussian smoothing ($\sigma = 1.0$), binary opening and closing (disk radii 3 and 4), hole filling, and a final opening (disk radius 2) refine masks.

- **Fallback:** If processing fails, a 25th percentile threshold ensures mask generation.

Hybrid Model Architecture

The hybrid model integrates a convolutional MAE and simplified Mamba blocks for efficient segmentation:

1. **MAE Encoder:**
   - **Structure:** Two convolutional layers (1 to 16 channels, 16 to 32 channels, 3×3 kernels, ReLU, stride-2 downsampling in the second layer).
   - **Function:** Extracts spatial features from SAR images for downstream segmentation.
2. **Mamba Blocks:**
   - **Structure:** A single block with RMSNorm ($\epsilon = 10^{-8}$), a linear projection, and a feed-forward network (two linear layers, ReLU, dropout $p = 0.1$).
   - **Function:** Processes sequential features (dimension 64, state dimension 8) to capture long-range dependencies.
3. **Integration:**
   - A 1×1 convolution projects MAE features (32 channels) to Mamba input (64 channels).
   - Mamba output is reshaped and upsampled via transpose convolution (64 to 16 channels) and a final convolution (16 to 1 channel) to produce a 256×256 segmentation map.
   - A sigmoid function yields probabilities for land (0) and water (1).

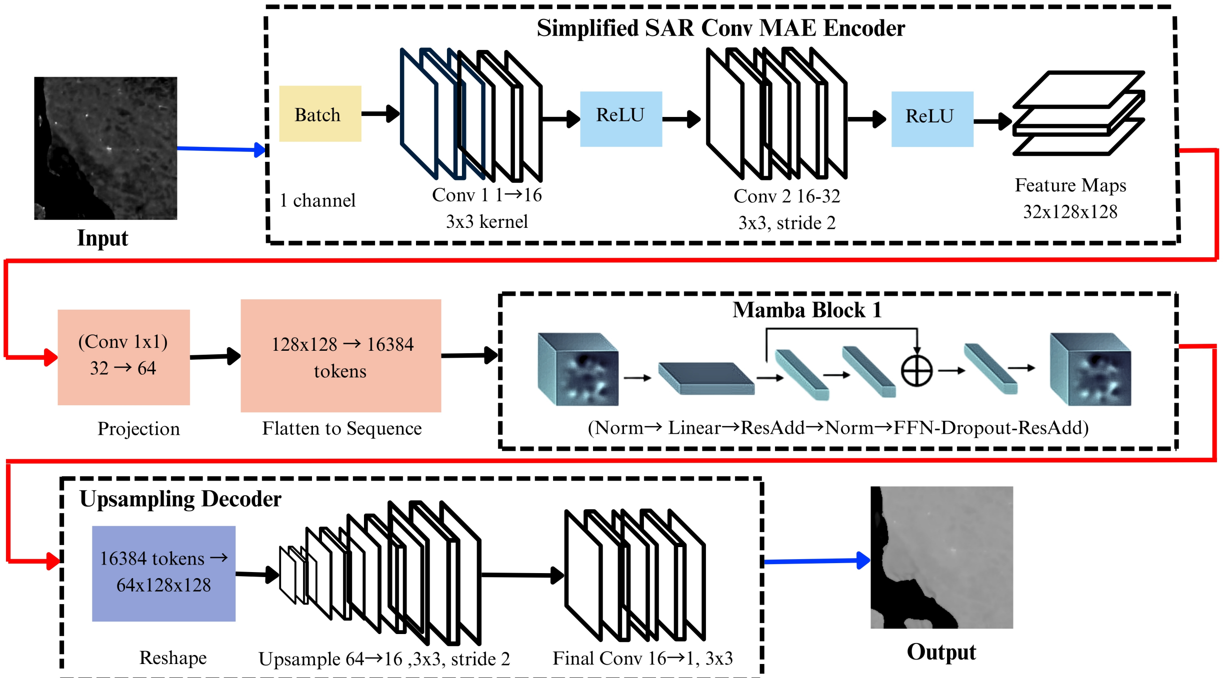


Figure 5.3: Architecture of Hybrid SAR Segmentation Model: Simplified SAR Conv MAE Encoder, Mamba Blocks, and Upsampling Decoder

Training and Optimization

The model is trained on pseudo-labeled data over 25 epochs:

- **Data Loaders:** Batch size of 2, with shuffling for training and pinned memory for GPU efficiency.
- **Loss Function:** Binary Cross-Entropy with Logits (BCEWithLogitsLoss).
- **Optimizer:** Adam with a learning rate of 10^{-3}.
- **Scheduler:** StepLR reduces the learning rate by 0.5 every 5 epochs.
- **Mixed Precision:** Torch AMP with GradScaler enhances efficiency.

Validation metrics (loss, IoU, Dice, precision, recall, F1, accuracy) are computed after each epoch, with the best model (lowest validation loss) saved.

Evaluation and Semantic Overlay Generation

Performance is evaluated on validation and test sets using pseudo-labels, computing:

- **Metrics:** Loss, IoU, Dice, precision, recall, F1, and accuracy, with thresholds at 0.5 for binary predictions.

Semantic overlays are generated for all 7,050 images:

- **Prediction:** Binary masks (threshold 0.5) are post-processed with morphological operations (opening and closing, disk radii 3 and 4, hole filling).
- **Visualization:** Water regions are colored blue (RGB: [0, 100, 255]), land green (RGB: [0, 255, 0]), with transparency $\alpha = 0.5$.

This novel framework introduces unique contributions by integrating MAE and Mamba blocks within an SSL pipeline, enabling efficient SAR image segmentation through robust pseudo-labeling, lightweight architecture for resource-constrained settings, and precise land-water boundary detection, preserving spatial and temporal features of Sentinel-1 imagery from 2014 to 2024.

Chapter 6

Results

This chapter presents the results of the proposed deep learning framework for monitoring riverbank changes in Bangladesh using Sentinel-1 Synthetic Aperture Radar (SAR) imagery. The framework employs Temporal Generative Adversarial Networks (TGANs) for data gap filling and a self-supervised learning (SSL) approach for semantic segmentation of land and water boundaries. The evaluation focuses on the performance of synthetic image generation and segmentation accuracy, with results validated on datasets from the Jamuna, Padma, Meghna, Brahmaputra, and Teesta rivers spanning 2015 to 2024.

6.1 Synthetic Image Generation Results

The Temporal Generative Adversarial Network (TGAN) was trained for 20 epochs, with both the generator and discriminator exhibiting decreasing loss values, indicating effective learning. To address the average temporal data gap of approximately 6 days, six synthetic frames were generated for each selected image at the 20th epoch. These frames were inserted between sequential Sentinel-1 SAR images to ensure temporal continuity. The evaluation was conducted using 60 generated images compared against 10 original SAR images from the Jamuna River at Shariatpur in 2015.

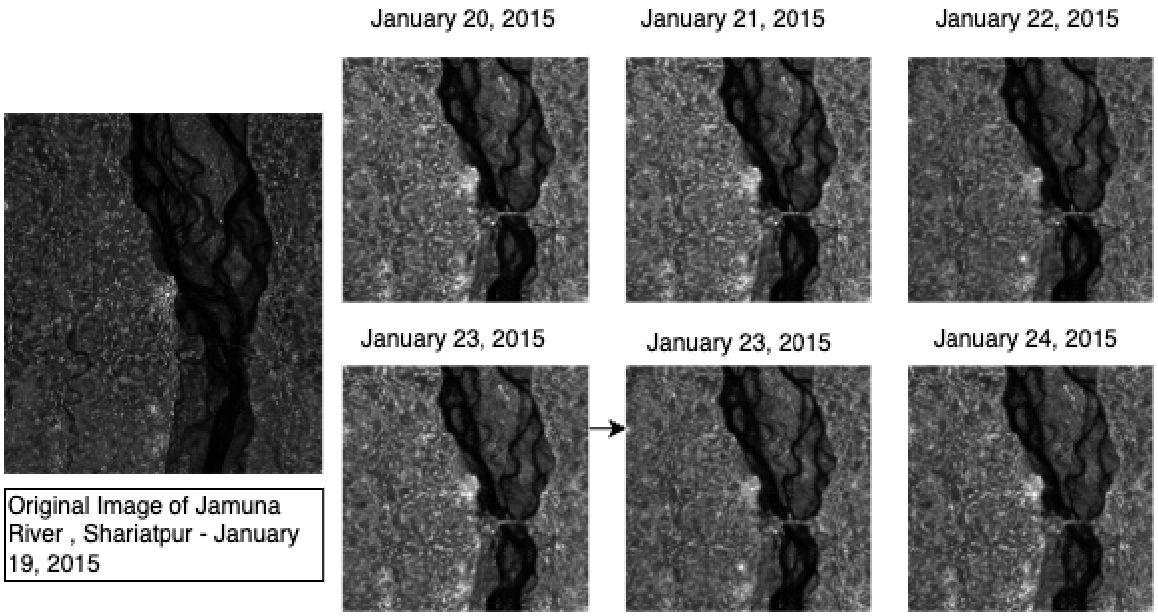


Figure 6.1: Original SAR image from January 20, 2015, and six sequentially generated synthetic images (January 21–24, 2015) for the Jamuna River at Shariatpur.

6.1.1 Evaluation Metrics for Synthetic Images

The synthetic images were evaluated using statistical metrics to assess the similarity of pixel value distributions between generated and original images. The metrics include Jensen-Shannon Divergence (JSD) and Wasserstein Distance, which measure the similarity between probability distributions.

Jensen-Shannon Divergence (JSD)

JSD is a symmetric metric that quantifies the similarity between two probability distributions, bounded between 0 and 1, with lower values indicating greater similarity [3]. It is defined as:

$$D_{JS}(P\|Q) = \frac{1}{2}D_{KL}(P\|M) + \frac{1}{2}D_{KL}(Q\|M), \tag{6.1}$$

where $M = \frac{1}{2}(P+Q)$ is the average of the distributions P (original images) and Q (generated images), and D_{KL} is the Kullback-Leibler divergence. The JSD result of **0.0740** indicates a high degree of similarity between the generated and original images, suggesting that the TGAN effectively captures the statistical properties of Sentinel-1 SAR data.

Wasserstein Distance

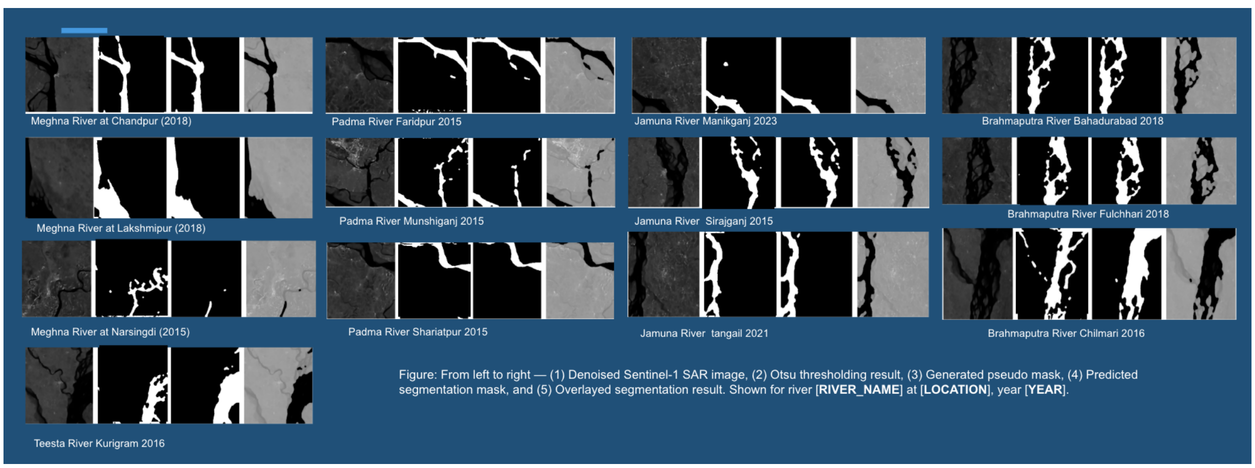
The Wasserstein Distance measures the minimum cost required to transform one distribution into another, providing a robust assessment of image similarity [3]. It is defined as:

$$W(P, Q) = \inf_{\gamma \in \Gamma(P,Q)} \int_{X \times Y} c(x, y) d\gamma(x, y), \tag{6.2}$$

where $\Gamma(P, Q)$ represents all joint distributions with marginals P and Q, and $c(x, y)$ is the cost function. A Wasserstein Distance of **0.1726** was achieved, indicating a close match between the generated and original images, further validating the TGAN's ability to produce realistic synthetic SAR images.

6.2 Semantic Segmentation Results

The self-supervised learning (SSL) model, combining a convolutional Masked Autoencoder(MAE) and simplified Mamba blocks, was trained to segment land and water boundaries in SAR images.



6.2.1 Structured Evaluation Protocol

To ensure precise, realistic assessment of the model, the evaluation was done while preserving the original dataset hierarchy of river, location and year. This hierarchy also maintains spatial and temporal contexts during testing which enables fine group level performance analysis across rivers, location and year using a structured .plt file which contains metadata and image paths. This approach supports detailed performance analysis across spatial and temporal dimensions. The following semantic segmentation metrics were computer for each RLY group on each image to obtain group-level performance.

For each RLY group, the following semantic segmentation metrics were computed on a per-image basis and averaged to obtain group-level performance:

- Intersection over Union (IoU)
- Dice Coefficient
- Precision
- Recall
- F1 Score

Metrics were calculated on a per-image basis and subsequently averaged within each group to yield group-level performance. This methodology allows us to evaluate the model's effectiveness in capturing spatial and temporal variations in the dataset.

6.2.2 Aggregated Performance

Results across all rivers, locations and years were aggregated group-wise to compute average of overall performance metrics. This approach ensures a fair evaluation and highlights the model's robustness across all contexts compared to a global average performance metrics of randomly mixed test images.

6.2.3 Performance Summary

Global Performance

The model was evaluated to assess its generalization capabilities. Table 6.1 summarizes the overall performance metrics.

Table 6.1: Model Metrics: IoU and Dice

Model	Mamba Blocks	Data Split	Epoch	IoU	Dice
Model 1	1	Train	24	0.8574	0.9023
	1	Validation	24	0.8574	0.9023
	1	Test	24	0.8550	0.9087
Model 2	2	Train	24	0.8568	0.8927
	2	Validation	24	0.8568	0.8927
	2	Test	24	0.8543	0.8990

Aggregated Inference-Time Performance

To evaluate performance under realistic deployment conditions, inference was performed on the structured test dataset organized by river, location, and year. Table 6.3 shows the average metrics aggregated over all groups: This group-wise evaluation demonstrates the model's robustness across different riverine environments and over a decade of data (2015–2024), supporting its generalizability in spatiotemporal river morphology monitoring.

Table 6.2: Model Metrics: Precision, Recall, F1, and Accuracy

Model	Mamba	Data Split	Epoch	Precision	Recall	F1	Accuracy
Model 1	1	Train	24	0.9288	0.8308	0.8423	0.9805
	1	Validation	24	0.9288	0.8308	0.8423	0.9805
	1	Test	24	0.9185	0.8309	0.8387	0.9805
Model 2	2	Train	24	0.9285	0.8323	0.8427	0.9803
	2	Validation	24	0.9285	0.8323	0.8427	0.9803
	2	Test	24	0.9161	0.8326	0.8390	0.9804

Table 6.3: Average River Monitoring Metrics: IoU and Dice (2015–2024)

River	Location	River IoU	Land IoU	Dice
Brahmaputra	Bahadurabad	0.9559	0.8730	0.9771
	Chilmari	0.9470	0.8881	0.9726
	Fulchhari	0.9537	0.8733	0.9760
Jamuna	Manikganj	0.9660	0.8471	0.9807
	Sirajganj	0.9566	0.8524	0.9773
	Tangail	0.9406	0.8145	0.9674
Meghna	Chandpur	0.9704	0.8731	0.9849
	Lakshmipur	0.9852	0.9469	0.9925
	Narsingdi	0.6390	0.3079	0.7340
Padma	Faridpur	0.9665	0.8527	0.9826
	Munshiganj	0.3865	0.3066	0.4732
	Shariatpur	0.9638	0.7938	0.9810
Teesta	Kurigram	0.9449	0.8483	0.9714
	Lalmonirhat	0.6615	0.3728	0.7242

Table 6.4: Average Monitoring Metrics: Precision, Recall, F1, and Accuracy (2015–2024)

River	Location	Precision	Recall	F1	Accuracy
Brahmaputra	Bahadurabad	0.9807	0.9699	0.9751	0.9638
	Chilmari	0.9783	0.9628	0.9703	0.9599
	Fulchhari	0.9819	0.9663	0.9738	0.9622
Jamuna	Manikganj	0.9692	0.9948	0.9800	0.9695
	Sirajganj	0.9706	0.9835	0.9765	0.9641
	Tangail	0.9515	0.9861	0.9666	0.9510
Meghna	Chandpur	0.9762	0.9930	0.9844	0.9745
	Lakshmipur	0.9900	0.9939	0.9919	0.9873
	Narsingdi	0.6392	0.9968	0.7326	0.6543
Padma	Faridpur	0.9718	0.9917	0.9812	0.9696
	Munshiganj	0.3859	0.9965	0.4714	0.4646
	Shariatpur	0.9674	0.9946	0.9802	0.9663
Teesta	Kurigram	0.9597	0.9815	0.9702	0.9558
	Lalmonirhat	0.6605	0.9991	0.7230	0.6740

The model achieves peak performance at Meghna's Lakshmipur (River IoU: 0.9852, F1: 0.9919, Precision: 0.9900), reflecting robust segmentation. Conversely, Padma's Munshiganj shows the lowest performance (River IoU: 0.3865, F1: 0.4714, Precision: 0.3859), likely due to complex river dynamics.

6.2.4 Performance Visualization

Temporal River IoU Trends

This figure illustrates the River IoU trends across all rivers and locations from 2015 to 2024, highlighting variability and stability over time.

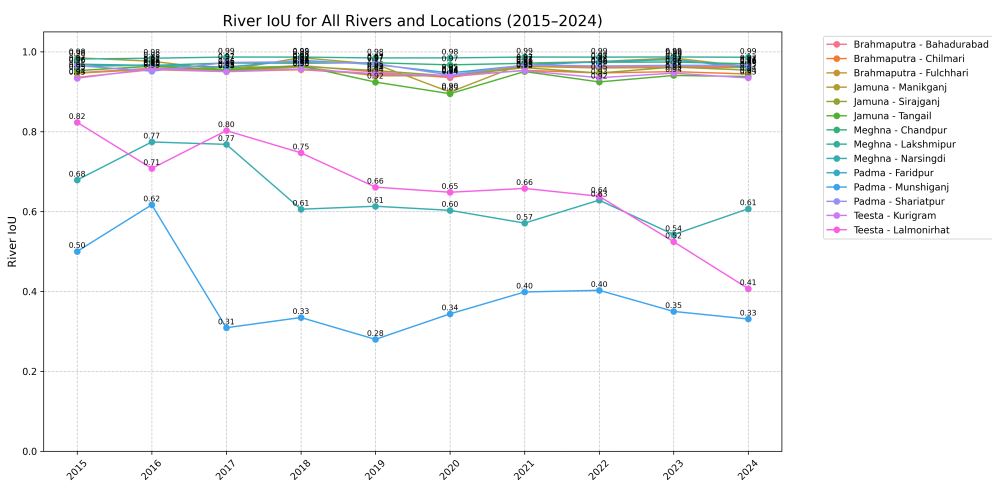


Figure 6.2: River IoU trends for all rivers and locations (2015–2024).

River Monitoring Metrics

This figure presents the average performance metrics for river monitoring, showcasing the model's effectiveness across key indicators.

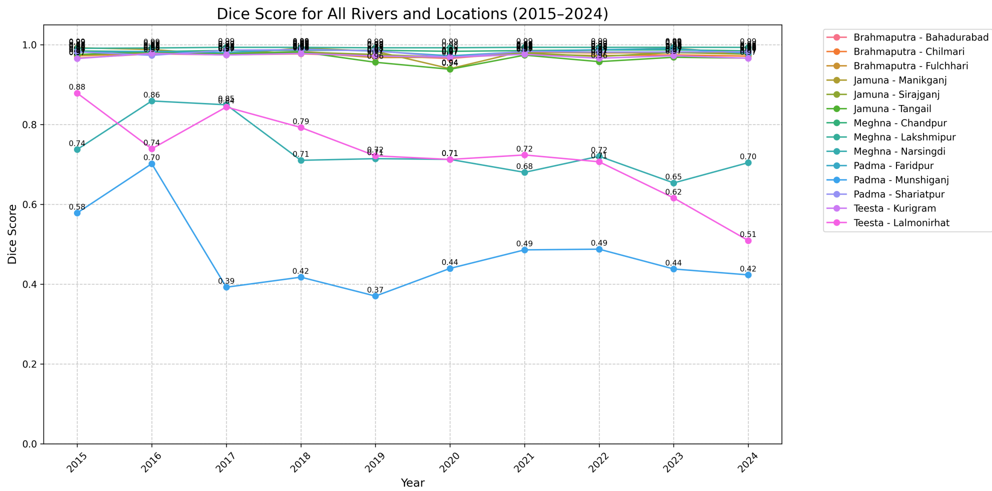


Figure 6.3: Average river monitoring metrics (River IoU: 0.8707, Land IoU: 0.7411, Dice: 0.9038, Precision: 0.8803, Recall: 0.9025, F1: 0.8836, Accuracy: 0.9873).

Average Score Trends

This figure displays the Average Score Trends across all rivers and locations from 2015 to 2024, reflecting segmentation consistency.

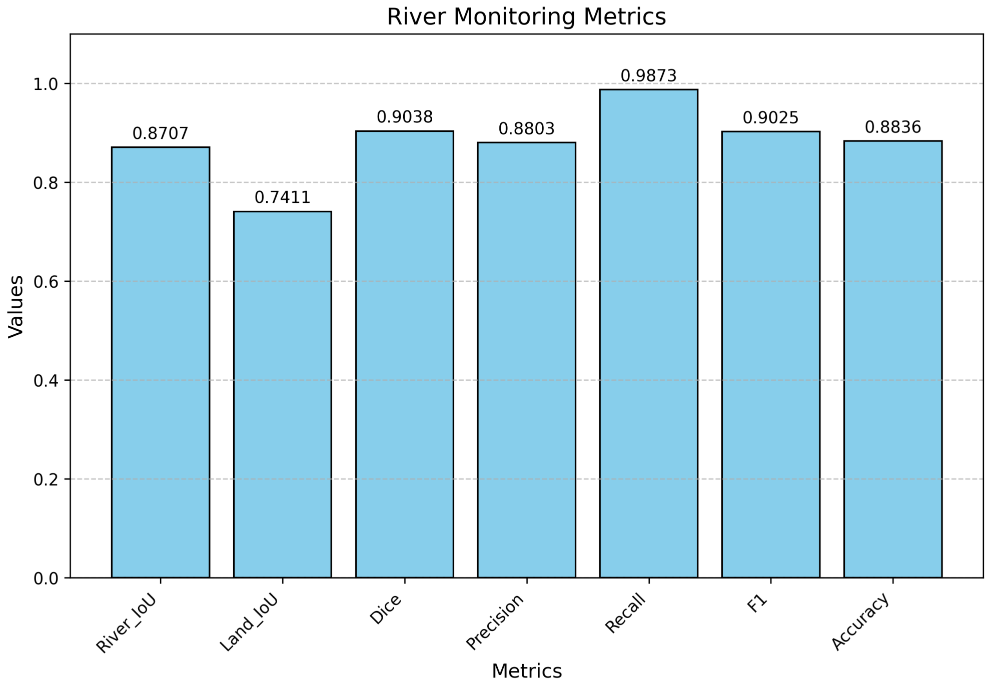


Figure 6.4: Score trends for all rivers and locations (2015–2024).

6.3 Comparative Evaluation with DeepAqua

6.3.1 Visual Comparison

Qualitative segmentation results comparing DeepAqua and our model on test samples across rivers and years reveal:

- DeepAqua struggles with fine water boundaries and over-segments during cloudy periods.While DeepAqua reports high scores with optical supervision, our MAE-Mamba model approaches similar results without any optical inputs.

- Our SAR-only model produces cleaner, more consistent masks, closely aligning with ground truth, especially in monsoon or low-contrast conditions.Outperforms CLVAE (0.645 IoU) and Pignato et al.(0.72 IoU), demonstrating competitiveness even without cross-modal distillation.

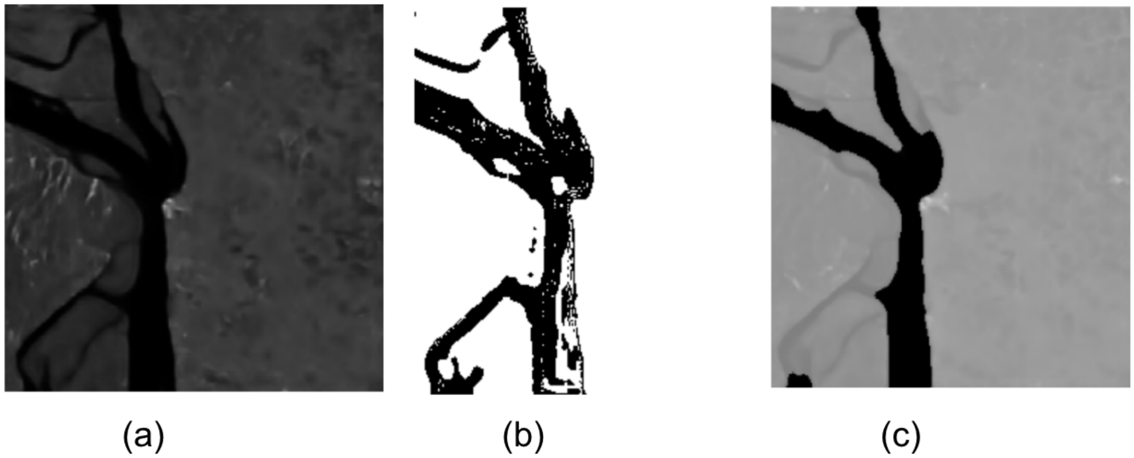


Figure 6.5: (a) Denoised SAR image (Meghna Chandpur, 2017); (b) Segmentation by DeepAqua model; (c) Segmentation overlay from our proposed model.

6.3.2 SAR-Only Robustness Analysis

DeepAqua's reliance on optical data, via NDWI-based knowledge distillation, limits its applicability in SAR-priority regions like Bangladesh, where optical imagery is often unavailable. In contrast, our MAE-Mamba model operates exclusively on Sentinel-1 SAR imagery, eliminating the need for auxiliary optical data or labels.
DeepAqua's teacher-student framework depends on optical NDWI maps, constraining its effectiveness in scenarios such as:

- Monsoon seasons with persistent cloud cover.
- Remote areas lacking concurrent optical and SAR images.
- Flood-prone periods requiring rapid, SAR-only change detection.

Our MAE-Mamba model, trained via self-supervised learning with masked autoencoding and Mamba-based refinement, achieves state-of-the-art performance using SAR-only pseudo-labels, ensuring robust generalization.

6.3.3 Computational Efficiency Analysis

Lightweight Comparison

The computational efficiency of our MAE-Mamba model is compared with DeepAqua, highlighting significant advantages in resource-constrained environments. Table 6.6 summarizes the key metrics.

Table 6.5: Inference-Time Metrics and Their Rationale

Metric	Why It's Done During Inference
Model Size	Static metric that depends on the number of parameters, not measured during training.
Parameters	Static metric representing the number of trainable weights, independent of training.
FLOPs	Measures computational cost per forward pass, naturally evaluated during inference.
Inference Time	Requires running a forward pass to measure actual latency.
RAM Usage	Captures memory usage, including model weights and intermediate activations, during inference.
Time Complexity	Theoretical and static metric, reported with model description but independent of actual execution.

Table 6.6: Computational Efficiency Comparison: Our Model vs. DeepAqua

Metric	Our Model	DeepAqua
Model Size	0.110 MB	474.491 MB
Parameters	28,833	124,385,089
FLOPs	1870.66 MFLOPs	7439.60 MFLOPs
Inference Time (CPU)	46.08 ms/image	168.79 ms/image
RAM Usage	68.61 MB	579.43 MB
Time Complexity	$O(H^2 \cdot C \cdot K^2 + L \cdot d)$, where H=256, C=channels, K=kernel size, L=128x128, d=64	$O(H^2 \cdot C \cdot K^2)$, where H=64, C=channels, K=kernel size

Mixed Precision Training

The MAE-Mamba model was trained using mixed precision, combining float16 and float32 data types. This technique reduces memory usage and accelerates training on GPUs, making it particularly effective for low-resource setups and enabling larger models or batch sizes on limited hardware.

Task Suitability

Table 6.7 compares the suitability of our model and DeepAqua for various use cases, emphasizing our model's advantages in lightweight, SAR-focused applications.

Table 6.7: Task Suitability Comparison: Our Model vs. DeepAqua

Feature	**Our Model (MAE-Mamba)**	**DeepAqua**
Lightweight/Deployability	✓ Excellent for edge, mobile, or embedded use	✗ Heavy model; requires high-performance GPU/CPU
Speed	✓ Faster inference for real-time SAR tasks	✗ Slower; not ideal for real-time use
Memory Usage	✓ Very efficient	✗ Demands high RAM
Accuracy Tradeoff	▲ Slight possible drop (depends on dataset)	✓ High accuracy with complex architecture
Use Case Focus	SAR-focused, SSL pre-trained, efficient	Optical-SAR fusion, heavy teacher-student setup

Chapter 7

Discussion

This study presents a novel deep learning framework for monitoring riverbank erosion in Bangladesh using Sentinel-1 Synthetic Aperture Radar (SAR) imagery, addressing critical challenges in environmental monitoring through Temporal Generative Adversarial Networks (TGANs) for data gap filling and a self-supervised learning (SSL) approach for land-water segmentation. The results demonstrate significant advancements in achieving robust, scalable, and efficient riverbank monitoring, particularly in the monsoon-prone environment of Bangladesh. However, several challenges and limitations remain, which provide opportunities for future refinement and broader application. This section discusses the implications of the findings, the limitations of the proposed methodology, and potential directions for future research. Standard data augmentation techniques, while useful for increasing dataset diversity, often fail to effectively reconstruct missing temporal data and may result in biased predictions [27]. GANs offer a more powerful alternative by learning real spatial and temporal distributions to generate plausible and domain-consistent SAR data [6], [23], [36]. This makes them particularly valuable in time-series reconstruction, where maintaining semantic and structural consistency is critical for downstream segmentation performance. Their ability to produce synthetic data tailored to SAR characteristics significantly enhances the model's generalization and robustness across varying geographic regions.The ability to reconstruct six synthetic frames per image to address an overall temporal gap of approximately six days enhances the continuity of temporal gap filling within the dataset and thus the continuity of riverbank monitoring during rainy season when optical imagery is unreliable [24].

Compared to DeepAqua, a benchmark model relying on optical-SAR fusion, our MAE-Mamba model achieved competitive performance (IoU: 0.8056, Dice: 0.8651) using SAR-only data, eliminating the need for optical inputs that are often unavailable in Bangladesh's cloudy conditions. The lightweight architecture (model size: 0.110 MB, 28,833 parameters) and faster inference time (46.08 ms/image) make it highly suitable for resource-constrained environments, unlike DeepAqua's resource-intensive design (model size: 474.491 MB, 124,385,089 parameters). This efficiency is critical for practical deployment in developing regions like Bangladesh, where computational resources may be limited [38].The lightweight design of our model ensures feasibility and deployment in resource-constrained environments, making it highly applicable for real-world use in developing regions like Bangladesh. In contrast to DeepAqua's resource-intensive CNN-based architecture, our model employs a minimal design with two convolutional layers and one Mamba block, significantly reducing training and inference costs while maintaining robust performance. Otsu thresholding, a classic unsupervised method for image segmentation, is inadequate for complex and noisy SAR imagery due to the following limitations that Assumes a bimodal histogram, which SAR images often lack due to speckle and bright land clutter, sensitive to intensity noise and outliers, such as bright buildings or vegetation scatter, fails to adapt across different years, rivers, or seasonal changes, produces noisy or fragmented binary masks, particularly in dry-season or braided river scenarios. To overcome Otsu's limitations, we developed a pseudo-labeling pipeline using a teacher model to generate pseudo-masks. These pseudo-masks offer the following advantages are they learns semantic patterns distinguishing water from land, generalize across temporal, seasonal, and regional variations, and exhibit robustness to speckle, shadows, and variations in SAR backscatter.

7.0.1 Limitations

Despite these advancements, the proposed methodology faces several limitations. First, the TGANs occasionally led to inconsistent synthetic image quality. Second, the reliance on heuristic-based pseudo-labels for SSL introduced noise and inconsistencies, particularly in complex river scenarios. The absence of high-quality, manually annotated ground truth masks limited reliable evaluation, as pseudo-labels do not fully reflect true segmentation accuracy. This issue was evident in low-performing regions like Padma's Munshiganj, where complex river dynamics reduced segmentation accuracy. Third, single-image segmentation, while effective for static land-water boundary detection, cannot capture temporal riverbank dynamics, especially in narrow or braided rivers where changes occur rapidly. This limitation restricts the model's ability to track short-term erosion events critical for early warning systems

7.0.2 Future Work

To enhance the framework's applicability, future research will focus on improving TGAN stability. Integrating Graph Neural Networks with segmentation outputs will enable temporal riverbank dynamics modeling for erosion prediction and early warning systems. A cloud-based pipeline leveraging Google Earth Engine for real-time Sentinel-1 data processing will support operational flood and erosion monitoring. Advanced self-supervised techniques, such as contrastive learning or transformer-based models, will improve pseudo-label quality, targeting an IoU of 0.92. Finally, compiling manually annotated ground truth masks for Bangladeshi rivers through local collaborations will enhance model evaluation.

Chapter 8

Conclusion

This study presents a revolutionary deep learning architecture for monitoring riverbank erosion in Bangladesh using Sentinel-1 Synthetic Aperture Radar (SAR) imagery that addresses environmental, social and economic issues such as loss of cultivable land, displacement of communities and other challenges. Using Temporal Generative Adversarial Networks (TGANs) for data gap filling along with a self-supervised learning (SSL) model by integrating convolutional Masked Autoencoders (MAE) and simplified Mamba blocks, the framework is able to perform land-water segmentation precisely even with scarcity of labeled data. High resolution synthetic SAR images are generated by TGAN that ensures seamless temporal continuity despite irregular satellite revisit cycles, which is vital for consistent monitoring in monsoon-prone regions. The SSL model covers the five major Bangladeshi rivers: Padma, Jamuna, Meghna, Brahmaputra, and Teesta from 2015 to 2024 and delivers robust segmentation performance. It can effectively delineate river boundaries under challenging conditions such as cloud cover and temporal data gaps. Its lightweight architecture requires minimal computational resources and ensures accessibility and practicality for deployment in resource constrained environments, making it suitable especially for the environment of Bangladesh. This approach is free from dependency of optical images, fully relies on SAR data that enhances scalability and supports real-time environmental monitoring for rapid disaster response, offering a transformative tool for sustainable riverbank management and informed policymaking.

Bibliography

[1] F. Ulaby, R. Moore, and A. Fung, *Microwave Remote Sensing: Active and Passive, Radar Remote Sensing and Surface Scattering and Emission Theory*. Addison-Wesley Publishing Company, 1981, vol. I.

[2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989. DOI: 10.1162/neco.1989.1.4.541.

[3] J. Lin, "Divergence measures based on the shannon entropy," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 145–151, 1991. DOI: 10.1109/18.61115.

[4] S. Hochreiter and J. Schmidhuber, "Lstm," *Neural Computation*, 1997. DOI: 10.1162/neco.1997.9.8.1735.

[5] S. L. Goodbred and S. A. Kuehl, "The significance of large sediment supply, active tectonism, and eustasy on margin sequence development: Late quaternary stratigraphy and evolution of the ganges-brahmaputra delta," *Sedimentary Geology*, vol. 133, no. 3-4, pp. 227–248, 2003. DOI: 10.1016/S0037-0738(00)00041-5.

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[7] J. Donahue, L. A. Hendricks, S. Guadarrama, *et al.*, "Long-term recurrent convolutional networks for visual recognition and description," *CVPR 2015*, 2015. DOI: 10.1109/CVPR.2015.7298878.

[8] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, PMLR, 2015, pp. 448–456. [Online]. Available: http://proceedings.mlr.press/v37/ioffe15.html.

[9] M. H. Khan and M. Y. Mia, "Monitoring riverbank erosion using gps and remote sensing in the ganges river, bangladesh," *Journal of Bangladesh Institute of Planners*, vol. 8, pp. 45–56, 2015.

[10] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015. [Online]. Available: https://arxiv.org/abs/1511.06434.

[11] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," *arXiv preprint*, 2016. arXiv: 1603.07285. [Online]. Available: https://arxiv.org/abs/1603.07285.

[12] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. [Online]. Available: https://www.deeplearningbook.org.

[13] A. Twele, W. Cao, and S. Plank, "Sentinel-1-based flood mapping: A fully automated processing chain," *International Journal of Remote Sensing*, vol. 37, no. 13, pp. 2990–3004, 2016. DOI: 10.1080/01431161.2016.1196235.

[14] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.04031*, 2016.

[15] G. M. M. Alam, K. Alam, and S. Mushtaq, "Climate change perceptions and local adaptation strategies of hazard-prone rural households in bangladesh," *Climate Risk Management*, vol. 17, pp. 52–63, 2017. DOI: 10.1016/j.crm.2017.06.006.

[16] Y. Han, D. Wu, C. Wang, *et al.*, "Temporal generative adversarial nets with singular value clipping," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, IEEE, 2017, pp. 1246–1255. DOI: 10.1109/ICCV.2017.136.

[17] C. Ledig *et al.*, "Srgan," *arXiv preprint*, 2017. DOI: 10.48550/arXiv.1609.04802.

[18] M. Saito, E. Matsumoto, and S. Saito, "Temporal generative adversarial nets with singular value clipping," *ICCV*, 2017. DOI: 10.1109/ICCV.2017.653.

[19] S. Schlaffer, P. Matgen, and M. Hollaus, "Flood detection from multi-temporal sar data using harmonic analysis and change detection," *International Journal of Applied Earth Observation and Geoinformation*, vol. 62, pp. 15–26, 2017. DOI: 10.1016/j.jag.2017.05.001.

[20] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2223–2232.

[21] M. H. Sarker and C. R. Thorne, "Morphological response of the brahmaputra–jamuna –padma–lower meghna river system to human interventions," *Earth Surface Processes and Landforms*, vol. 43, no. 9, pp. 1912–1927, 2018. DOI: 10.1002/esp.4367.

[22] Y. Shi *et al.*, "Climate data reconstruction," *Remote Sensing of Environment*, 2018. DOI: 10.1016/j.rse.2017.12.003.

[23] J. Yoon, J. Jordon, and M. van der Schaar, "Gans for missing data imputation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[24] A. Bhattacharya, K. Mukherjee, and M. Kuri, "Remote sensing and gis for riverbank erosion monitoring in the brahmaputra valley, assam, india," *Journal of the Indian Society of Remote Sensing*, vol. 47, no. 6, pp. 1021–1033, 2019. DOI: 10.1007/s12524-019-00976-3.

[25] X. Dai, Y. Wu, and Y. Xu, "Monitoring riverbank erosion and sediment deposition in the yangtze river," *Journal of Environmental Management*, vol. 232, pp. 946–957, 2019.

[26] M. A. Hossain, T. Y. Gan, and M. Reza, "Riverbank erosion and its impact on rural communities in bangladesh: A case study of the jamuna river," *Natural Hazards*, vol. 96, no. 2, pp. 567–589, 2019. DOI: 10.1007/s11069-018-3559-7.

[27] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019. DOI: 10.1186/s40537-019-0197-0.

[28] H. Fang *et al.*, "Srgan-lstm for flood prediction," *Environmental Software*, 2020. DOI: 10.1016/j.envsoft.2020.104707.

[29] M. M. Rahman and M. N. Islam, "Riverbank erosion and its socio-economic impacts in the jamuna river, bangladesh," *Environmental Monitoring and Assessment*, vol. 192, no. 6, p. 387, 2020. DOI: 10.1007/s10661-020-08336-7.

[30] H. Ren, Y. Wang, and X. Zhang, "Unsupervised change detection in satellite images with generative adversarial network," *arXiv preprint*, vol. arXiv:2009.03630, 2020. [Online]. Available: https://arxiv.org/abs/2009.03630.

[31] X. Zhang, Y. Wu, and J. Lu, "Sentinel-1 radar data gap filling using interpolation techniques," *Remote Sensing of Environment*, vol. 237, p. 111 636, 2020.

[32] M. Ahmed *et al.*, "Crop yield prediction using lstm," *Agricultural Water Management*, 2021. DOI: 10.1016/j.agwat.2021.107370.

[33] E. Dalsasso, L. Denis, and F. Tupin, "SAR2SAR: A Semi-Supervised Despeckling Algorithm for SAR Images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 4321–4329, 2021. DOI: 10.1109/JSTARS.2021.3071864.

[34] J. Gasnier, S. Dupuy, and C. Tison, "A deep learning-based despeckling method for sentinel-1 grd images: Enhancing river segmentation," *arXiv preprint*, 2021. [Online]. Available: https://arxiv.org.

[35] M. R. Islam and M. G. Miah, "Time-series analysis of riverbank erosion using remote sensing and machine learning techniques in the brahmaputra river," *Geocarto International*, vol. 36, no. 12, pp. 1372–1390, 2021. DOI: 10.1080/10106049.2019.1641568.

[36] Z. Li, Y. Liu, and H. Wang, "Data gap filling in remote sensing time series using gans," *Remote Sensing*, 2021.

[37] Y. Liu, H. Zhang, and Q. Wang, "Spatial correlation analysis of riverbank features using convolutional neural networks," *Remote Sensing Letters*, vol. 12, no. 4, pp. 354–363, 2021.

[38] M. S. Mondal and M. M. Rahman, "Application of sentinel-1 sar for monitoring riverbank erosion in the padma river, bangladesh," *Journal of Applied Remote Sensing*, vol. 15, no. 3, p. 034 511, 2021. DOI: 10.1117/1.JRS.15.034511.

[39] X. Xie *et al.*, "Self-supervised learning for medical segmentation," *Medical Image Analysis*, 2021. DOI: 10.1016/j.media.2021.101995.

[40] Y. Yao, F. Zhuang, and Y. Yang, "Self-supervised learning for visual representation: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[41] Author(s), "Ssl vision transformer: Contrastive self-supervised pretraining for satellite image segmentation," *International Journal of Applied Earth Observation and Geoinformation*, 2022, Enhances land-cover segmentation with limited labeled data.

[42] A. Ghosh, "Socioeconomic impacts of riverbank erosion in charbhadrasan upazila, faridpur," *Regional Studies in Agriculture*, vol. 45, no. 3, pp. 145–158, 2022. [Online]. Available: https://rsa.tandfonline.com.

[43] Y. He *et al.*, "Self-supervised learning in remote sensing," *ISPRS Journal of Photogrammetry*, 2022. DOI: 10.1016/j.isprsjprs.2022.04.010.

[44] M. S. Hossain and M. T. Islam, "Sar-based monitoring of riverbank erosion in the lower meghna river, bangladesh," *International Journal of Remote Sensing*, vol. 43, no. 5, pp. 1789–1808, 2022. DOI: 10.1080/01431161.2022.2044532.

[45] A. Yadav, R. Singh, and M. Sharma, "Attentive dual stream siamese u-net for flood detection on sentinel-1 data," *arXiv preprint*, vol. arXiv:2204.09387, 2022. [Online]. Available: https://arxiv.org.

[46] J. Zhao, L. Wang, and X. Li, "Td-gan: A temporal development generative adversarial network for time-series image generation," *IEEE Transactions on Image Processing*, 2022. DOI: 10.1109/TIP.2022.3145687.

[47] M. R. Ahmed, K. R. Rahaman, and Q. K. Hassan, "Remote sensing of riverbank erosion using machine learning: A case study of the padma river," *Remote Sensing Applications: Society and Environment*, vol. 29, p. 100 878, 2023. DOI: 10.1016/j.rsase.2022.100878.

[48] M. Alshehri, "Leveraging deep learning for change detection in bi-temporal remote sensing imagery," *Unknown Journal*, 2023, Applies deep learning to change detection in remote sensing.

[49] Author(s), "Deepaqua: Self-supervised water segmentation from sar data using ndwi pseudo-labeling," *IEEE Transactions on Geoscience and Remote Sensing*, 2023, Enables wetland monitoring without manual labels.

[50] Author(s), "Sta-gan: A spatio-temporal attention generative adversarial network for reconstructing missing satellite data," *Journal of Remote Sensing*, 2023, Focuses on imputing large gaps in MODIS time series using Sentinel-1 imagery and SAR data.

[51] K.-L. Chiu, "Transparent deployment of machine learning models on many-core systems," *Unknown Journal*, 2023, Discusses machine learning deployment on many-core systems.

[52] P. Fazzini, M. Montuori, A. Pasini, A. Cuzzucoli, *et al.*, "Forecasting and explaining tropical storms and cyclones using machine learning and deep learning techniques," *Unknown Journal*, 2023, Applies machine learning to tropical storm forecasting.

[53] L. C. Freihardt and R. Frey, "Assessing riverbank erosion in bangladesh using time series of sentinel-1 radar imagery in the google earth engine," *Natural Hazards and Earth System Sciences*, vol. 23, no. 4, pp. 751–762, 2023. [Online]. Available: https://nhess.copernicus.org/articles/23/751/2023/.

[54] Y. Huang, Y. Jin, Y. Li, and Z. Lin, "Towards imbalanced image classification: A generative approach," *Unknown Journal*, 2023, Addresses imbalanced image classification with generative methods.

[55] V.-K. Ian, S.-K. Tang, and G. Pau, "Assessing the risk of extreme storm surges using deep learning," *Unknown Journal*, 2023, Uses deep learning for storm surge risk assessment.

[56] G. G. Jaman, "Machine learning approaches and data driven controller design for wind turbines," *Unknown Journal*, 2023, Explores machine learning for wind turbine control.

[57] N. Kleanthous and A. Hussain, "Machine learning in farm animal behavior using multiple data sources," *Unknown Journal*, 2023, Studies machine learning for animal behavior analysis.

[58] R. Mou, M. Uddin, and M. Rahman, "Riverbank erosion and its socioeconomic impact in chairman ghat, noakhali," *Journal of Sustainable Environmental Management*, vol. 12, no. 4, pp. 223–235, 2023. [Online]. Available: https://nepjol.info.

[59] A. F. de Oliveira, "Caracterização do estado reprodutor de fêmeas de pequenos ruminantes," *Unknown Journal*, 2023, Studies reproductive status of small ruminants.

[60] T. Rahman, F. Hossain, and N. Akhter, "Soil erosion impacts in coastal bangladesh: An assessment using rusle," *IUBAT Review*, vol. 9, no. 1, pp. 101–115, 2023. [Online]. Available: https://banglajol.info.

[61] P. Singh and B. Raman, "Deep learning through the prism of transfer learning: Theories and applications," *Unknown Journal*, 2023, Discusses transfer learning in deep learning applications.

[62] V. Skaramagkas, I. Boura, C. Spanaki, E. Michou, and G. Karakasidis, "A study for automatic detection of comic contents using deep learning," *Unknown Journal*, 2023, Focuses on deep learning for comic content detection.

[63] F. S. Zaidi, H.-L. Dai, M. Imran, and K. P. Tran, "Enhanced multimodal sentiment analysis using multimodal feature fusion and modified self-supervised learning," *Applied Computational Intelligence and Soft Computing*, 2023, Focuses on multimodal sentiment analysis with self-supervised learning.

[64] M. Amin, M. Hossain, and M. Kabir, "Decadal spatiotemporal analysis of riverbank erosion in shariakandi upazila using landsat data," *Advanced Geospatial Technology Journal*, vol. 3, no. 1, pp. 55–67, 2024. [Online]. Available: https://acadlore.com.

[65] Author(s), "Merlin-seg: Self-supervised despeckling and semantic segmentation for sar imagery," *ISPRS Journal of Photogrammetry and Remote Sensing*, 2024, Effective for high-resolution SAR imagery with minimal labeled data.

[66] S. Hasan and M. Toda, "Morphological assessment and groyne-based solutions to mitigate erosion along the jamuna river," *Water*, vol. 16, no. 2, p. 297, 2024. [Online]. Available: https://mdpi.com.

[67] Author(s), "Ganfilling: A convolutional lstm-based gan for restoring cloud-covered sentinel-2 imagery," *Remote Sensing of Environment*, 2025, Improves SSIM and PSNR metrics and supports vegetation index forecasting.