



A Novel Deep Learning Model for Flood Detection from Synthetic Aperture Radar Images

Thanh-Nghi Doan ^{1,2,*} and Duc-Ngoc Le-Thi ^{1,2}

¹ Faculty of Information Technology, An Giang University, An Giang, Vietnam

² Vietnam National University, Ho Chi Minh City, Vietnam

Email: dtngchi@agu.edu.vn (T.-N.D.); ltdngoc97@gmail.com (D.-N.L.-T.)

*Corresponding author

Abstract—Flooding, a common natural disaster, causes widespread damage globally. Detecting flood extents rapidly and accurately using Synthetic Aperture Radar (SAR) images is crucial for effective disaster response and mitigation. This paper proposes a novel machine learning model specifically designed for SAR image analysis to detect floodwaters. The model leverages change detection techniques and operates on pairs of satellite images captured at different time points. The feature extraction module employs a parallel Siamese architecture with a Swin-Transformer backbone to extract features at various levels. Prior to entering the decoding module, the features undergo enhancement by computing the difference between feature maps at the same level. The decoding process predicts changing regions at each level and integrates them into the final result. Experimental results demonstrate that our proposed model outperforms other methods, achieving a recall of 94.6%, a precision of 96.9%, and an F1-score of 95.7%, with a computational cost of 32.3 G FLOPs.

Keywords—flood detection, deep learning model, Synthetic Aperture Radar (SAR) image, Swin-Transformer, vision transformer

I. INTRODUCTION

Flood is a natural disaster wherein a land area is temporarily inundated with a substantial amount of water. According to Ref. [1], solutions for mitigating the impact of floods can be broadly classified into two types: prevention solutions and complementary solutions. While prevention solutions focus on infrastructure and physical interventions to control water flow and prevent water accumulation [2], complementary solutions work in conjunction with prevention measures to enhance flood management [3]. They include flood monitoring, early warning systems, flood emergency response, communication, and awareness. Among these mentioned complementary solutions, early warning is considered promising, as it provides accurate and timely information about the size of the affected territory beforehand. This information allows for the calculation and mapping of

damages, aiding in crisis management and recovery processes.

This study aims to develop a novel deep learning model for accurate flood detection using Synthetic Aperture Radar (SAR) images. The proposed method leverages a Siamese network architecture combined with Swin-Transformer to enhance feature extraction from bi-temporal SAR images, incorporating residual connections and multi-level feature fusion to improve detection accuracy. Comprehensive experiments were conducted to compare the model's performance against existing state-of-the-art models, evaluating it based on precision, recall, and F1-Score. The study also analyzed the computational efficiency of the model, focusing on convergence and learning rate optimization. The practical applicability of the model for flood mapping and disaster mitigation was demonstrated using the SIGfloods dataset, highlighting its potential benefits for natural disaster response and remote sensing applications.

II. LITERATURE REVIEW

Various methods have been explored to map flooding and assess flood risks. Traditional approaches, like those in Ref. [4], have been employed to address these issues. While high resolution hydrologic models are effective on a small scale, applying them to community-level urban flooding is hindered by computational and input requirements [5]. This highlights the need for innovative techniques that can predict flooding accurately without extensive computational demands.

Remote sensing, particularly SAR, offers a cost-effective solution for large-scale flood mapping without stringent accuracy requirements or resource-intensive processes. This approach stands in contrast to conventional methods, which often struggle with the complexities of urban flooding. A comprehensive exploration of these remote sensing methods, including SAR and multi-frequency Polarimetric SAR for terrain classification, was documented in [6]. SAR, as an active sensor, excels in depicting the Earth's surface regardless of temporal constraints or cloud cover, and it has significantly advanced flood detection.

For instance, in Ref. [7], innovative multi-temporal COSMOSkyMed data from Northern Italy were harnessed

to create a classification algorithm for mapping flood evolution. Tanguy *et al.* [8] demonstrated the effective use of RADARSAT-2 SAR images and flood stage data for the 2011 Richelieu River flood.

Furthermore, Giustarini *et al.* [9] employed TerraSARX in conjunction with high-resolution aerial imagery to depict intricate flooding dynamics along England's River Severn. Furthermore, integrating SAR imagery with machine learning models opens up significant potentials. One of the most common solutions for flood detection tasks relies on supervised learning processes on bi-temporal SAR image datasets. This method requires a pair of images: one before the event and one after the event [10]. The model then learns to classify pixels based on the changes between the two images. In this field, Convolutional Neural Networks (CNNs) are regarded as a promising method for flood detection from satellite images. There are two main types of CNN-based methods: two-stage and one-stage methods.

In the two-stage method, as described in Wang *et al.* [11], CNNs are initially trained to segment the components that need to be detected in the image. Then, these regions in images from different time points are compared to identify changes. However, this method leads to the problem that accumulated errors in the segmentation process may affect the final performance.

On the contrary, the one-stage solution often employs Siamese architectures [12] to extract features from images at different time points in a shared feature space. Then, comparing these features yields change detection results. Several effective models based on this approach have been developed. These include FC-Siam-Diff [13], which combines Siamese architecture with a Fully Convolutional Network (FCN) and UNet architecture to extract features from pairs of satellite images and connect feature maps regarding differences from encoder layers to corresponding decoder layers; DTCDSN [14] develops self-attention mechanisms to capture more distinct features and expands CNN's perceptual area to gather more information in a broader context; and Siam-NestedUNet [15] develops a model based on the encoding and decoding architecture of NestedUnet with a Siamese encoder backbone.

Despite the potential for change detection, these methods often face limitations in flood detection. The main reason is that flood-affected areas often require global contextual information, while CNNs often cannot model global dependencies due to the local nature of convolution operations. Therefore, to address the requirement for learning broad correlations, recent studies have employed Vision Transformer (ViT) models to extract global relationships in change detection tasks. Bandara and Patel [16] proposed a method using multiple Transformer blocks combined with MSA modules to extract features at multiple levels and improve feature representation of optical images from two different time points. However, this method does not perform well on global satellite images. Due to the process of collecting images from satellites often introducing noise, this leads to a significant

degradation in the method's performance in change detection tasks.

In a recent study [17], a machine learning model was developed with three main components: first, a CNN was utilized to extract features from global satellite images; next, a ViT module was incorporated to enhance the learning capacity of the input data and create feature representations within a broader context; and finally, a decoder was employed to produce the final output. This approach effectively reduced the impact of noise in global satellite images, improving the accuracy of the results. Similarly, the BIT model developed by Chen *et al.* [18] used a CNN as a backbone to extract features from multitemporal images, then used a Transformer model to find global correlations between features. The absolute difference results between two feature maps were fed into an FCN to generate a change map. However, experimental results showed that these methods work well for static objects such as buildings or agricultural areas. For dynamic and widespread objects like floods, relying on the feature representation of these models' CNNs cannot capture enough global contextual information.

Cheng *et al.* [19] have thoroughly reviewed the advancements in change detection for remote sensing over the past decade, covering problem definitions, datasets, evaluation metrics, and transformer basics. It categorizes existing algorithms by granularity, supervision modes, and learning frameworks, and summarizes the performance of state-of-the-art algorithms, highlighting their strengths and limitations. Future research directions are also identified to guide and inspire further work in this field. Recently, Yan *et al.* [20] have introduced the Fully Transformer Network (FTN) for change detection in remote sensing images. FTN enhances global feature extraction and integrates multi-level visual features using a pyramid structure with a progressive attention module. Leveraging transformers for long-range dependency modeling, FTN learns discriminative features and generates complete change detection regions. Deeply-supervised learning with boundary-aware loss functions further optimizes the framework. Our method achieves state-of-the-art performance on four public change detection benchmarks.

Hyperspectral images are valuable for remote sensing but have redundant information due to many spectral bands. Band Selection (BS) helps reduce data volume, speed up processing, and improve accuracy. Traditional BS methods often fail to fully capture band interactions and redundancy. Esmaeili *et al.* [21] have introduced a BS method using a deep network with 3D-convolutional layers in a genetic algorithm, enhanced by a parent check box for effective genetic operations. Adding an attention layer and converting the model to spike neural networks were also explored. The method achieved accuracy improvements of 6% to 21%, reaching 90% to 99% in various evaluations.

To address the aforementioned shortcomings, this study proposes a novel deep learning model for flood detection from SAR images. This model leverages the Siamese architecture with the Swin-Transformer (Hierarchical ViT using shifted windows) as the backbone of the encoder,

coupled with a pyramid algorithm as the decoder. This innovative approach aims to enhance the accuracy and efficiency of flood detection in satellite imagery, offering a promising solution for effectively managing flood-related disasters. The primary contribution of this paper is:

- Overcomes the challenge of local feature representation inherent in CNNs.
- Mitigates the computational complexity associated with ViTs when processing large-scale images.
- Effectively and economically detects floods compared to traditional approaches using hydrologic systems.

The remainder of the paper is structured as follows: Section III presents the materials, methods and the overall framework of our proposed model. Section IV details the experimental procedures and results. Finally, Section V discusses conclusions, limitations, and potential directions for future research.

III. MATERIALS AND METHODS

As illustrated in Fig. 1, the workflow of the framework proposed in this study begins with two input images (T1 representing the pre-flood and T2 representing the post-flood). The model extracts feature from both images using a Siamese architecture with Swin-Transformer as the backbone, then compares the obtained feature maps to produce the prediction results.

During the prediction process, pixel values in the output image are assigned either 0 or 1. Pixels predicted to belong to a flood area receive a value of 1, while those predicted not to belong to a flood area receive a value of 0. Consequently, the output displays white areas representing flood regions and black areas representing non-flood regions. Based on this workflow, our proposed model, illustrated in Fig. 2, consists of two main components: the encoder (feature extractor) and the decoder (change predictor).

In this study, we utilize a hybrid method for change detection, combining transformer and pyramid modules. This hybrid approach integrates the strengths of both pixel-based and region-based methods to achieve higher accuracy in identifying flood extents from pre-flood and post-flood SAR images. The transformer method manages region-based analysis with attention modules in the encoder, while the pyramid module enables pixel-level comparison in the decoder, ensuring accurate predictions of flood extents. The Siamese network architecture, used for feature extraction, excels in detecting similarities or differences between inputs by comparing the feature vectors extracted from them. The Swin Transformer, chosen as the backbone for the Siamese network, leverages the shifted window technique, which confines the computation scope of attention points to non-overlapping local windows while maintaining connectivity between them. This technique reduces computational complexity and provides flexibility for different input sizes.

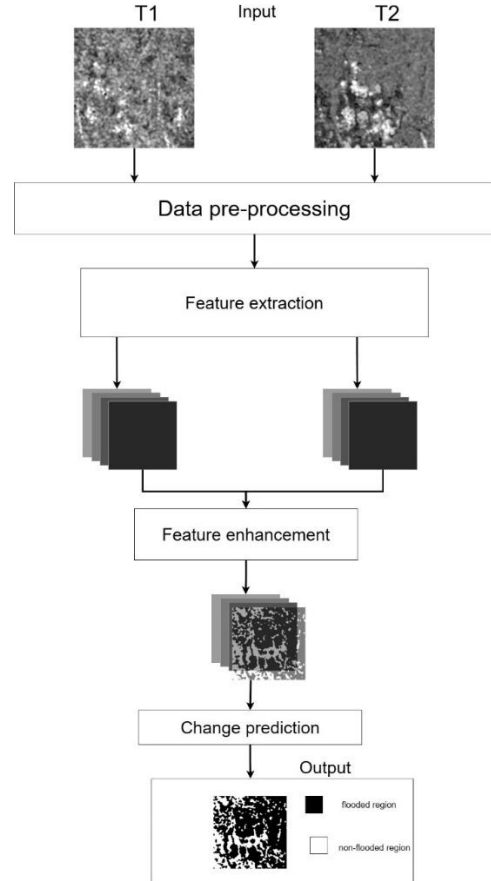


Fig. 1. The workflow of our proposed framework.

A. Synthetic Aperture Radar Imagery

Synthetic Aperture Radar images, generated by an active system, use microwave signals transmitted to the ground and received back by a sensor platform, either airborne via planes or satellites [22]. This process involves signal transmission, reception, and specific filtering, as shown in Fig. 3. The advantages of SAR imagery include:

- Operation in various weather conditions, ensuring uninterrupted monitoring, regardless of cloud cover or night-time conditions.
- Penetration of obstacles like clouds and vegetation, enabling the reflection of diverse surface features for more insightful analysis and detailed terrain mapping.
- Provision of high-resolution images, allowing for precise identification of flood-affected areas and differentiation of various land features.
- Capability to capture large areas in a single pass, making it efficient for monitoring extensive regions during natural disasters.
- Consistent data acquisition over time, facilitating change detection and trend analysis in flood-prone areas.
- Reduced dependency on daylight and favorable weather, making SAR an invaluable tool for continuous and reliable earth observation.

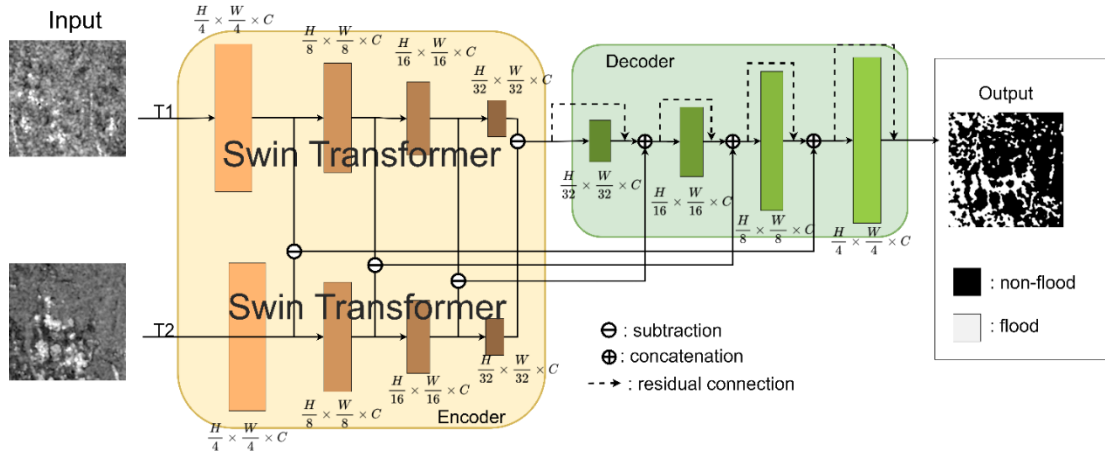


Fig. 2. The overall structure of our proposed framework.

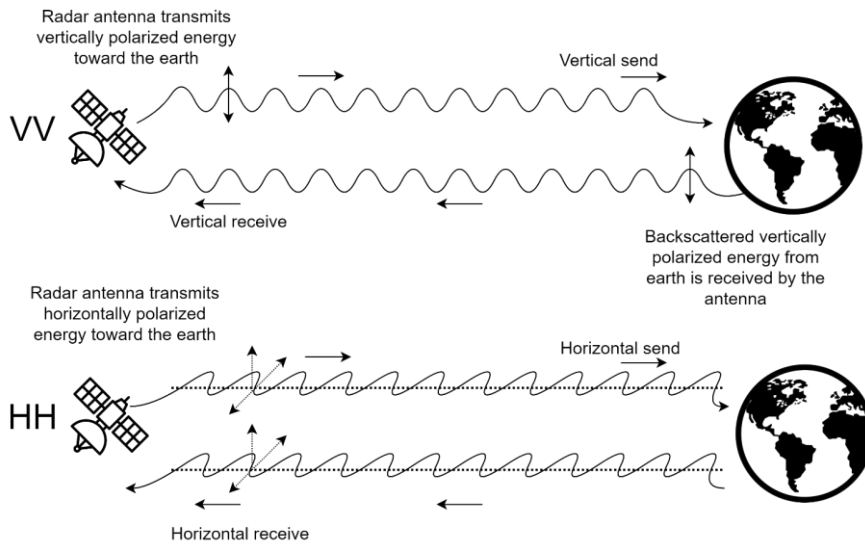


Fig. 3. Illustration of radar transmission and reception of polarized energy using specific filters. The first symbol indicates the direction of signal transmission, and the second indicates the direction of signal reception.

B. Change Detection Algorithm

Change detection in remote sensing involves identifying and quantifying changes in the Earth’s surface over time using data captured by remote sensing technologies [23]. In the context of flood detection, change detection plays a crucial role in identifying flood-affected areas based on bi-temporal SAR images, captured before and after a flood event [24]. Change detection methods can be classified based on algorithm granularity into three categories:

- **Pixel-based methods** assign labels to individual pixels in an image based on their spectral characteristics, aiming to partition the image into different classes. While traditional pixel-based methods may suffer from false positives and false negatives, advancements in deep learning techniques, like the pyramid pooling module [25] and attention module [26], have improved their accuracy and performance.
- **Region-based methods** use image segmentation techniques to group pixels into meaningful regions such as objects [27, 28], superpixels, or bounding boxes, based on their spatial, spectral, and

contextual characteristics. These regions are then analyzed for changes using traditional or deep learning methods.

- **Hybrid methods** combine the strengths of pixel-based and region-based methods to achieve higher accuracy in change detection [29–31]. By integrating different methodologies, hybrid approaches can overcome individual method limitations and provide a more robust solution for detecting changes in remote sensing imagery.

In this study, we utilize a hybrid method for change detection, combining transformer and pyramid modules. This approach is aimed at identifying flood extents from pre-flood and post-flood SAR images. The transformer method manages region-based analysis with attention modules in the encoder, while the pyramid module enables pixel-level comparison in the decoder, ensuring accurate predictions of flood extents.

C. Siamese Architecture

The Siamese network was first introduced in [32] to address the problem of signature verification using images. The Siamese architecture is a variant of a traditional CNNs,

which includes two or more branches for the input and concatenates the outputs of each branch into a new feature map or feature vector in the end [12]. Despite the different inputs, these sub-networks have the same configuration, parameters, and weights. This symmetry ensures that the parameter updating process is consistent across all sub-networks.

The Siamese network excels in detecting similarities or differences between inputs by comparing the feature vectors extracted from them. Leveraging this feature, the Siamese network is used to extract features from input pairs. These extracted features are then used to detect changes between input sets. With training and optimization, Siamese networks can efficiently detect changes.

D. Hierarchical ViT Using Shifted Windows

In this study, the hierarchical ViT using shifted windows (Swin-Transformer) introduced in [33] will serve as the backbone for the Siamese network for feature extraction. As depicted in the encoder part in Fig. 2, the Swin-Transformer comprises 4 stages, where the size of each patch doubles after each stage while the number of patches is halved. Specifically, the patch size increases from 4×4 pixels to 32×32 pixels, and the number of patches decreases from $\frac{H}{4} \times \frac{W}{4}$ to $\frac{H}{32} \times \frac{W}{32}$. To reduce computational complexity and ensure synchronization across feature maps, beneficial for subsequent operations, the number of channels in each layer of our model is mapped to a specific number of channels ($C = 96$) using a 1×1 Convolution layer.

The Swin-Transformer model was chosen over other ViT models because it leverages the shifted window technique. This technique confines the computation scope of attention points to non-overlapping local windows while maintaining connectivity between them. It is the hierarchical partitioning technique that implements this self-attention computation, providing flexibility for different input sizes and helping to reduce computational complexity. Given that the images are tokenized into $h \times w$ patches, the complexity can be described as Eqs. (1) and (2):

$$\omega(MSA) = 4hwC^2 + 2(hw)^2C \quad (1)$$

$$\omega(MSA) = 4hwC^2 + 2hwM^2C \quad (2)$$

where Eq. (1) is quadratic with respect to the number of patches hw , and Eq. (2) is linear when M is fixed (set to 7 by default). Global self-attention computation is generally impractical for a large hw , while window-based self-attention is scalable.

The shifted window partition is illustrated in Fig. 4, where the first module employs a regular window partitioning strategy starting from the top-left pixel. Subsequently, the following module adopts a windowing configuration shifted from that of the preceding layer, displacing the windows by $\left(\left\lfloor \frac{M}{2}, \left\lfloor \frac{M}{2} \right\rfloor \right\right)$ pixels from the regularly partitioned windows. This shifted window leads to a difference in the structure of the Transformer block in

Swin-Transformer compared to the original one (Fig. 5). There are two successive Transformer blocks in Swin Transformer, one for the original window partition and the next for the shifted one.

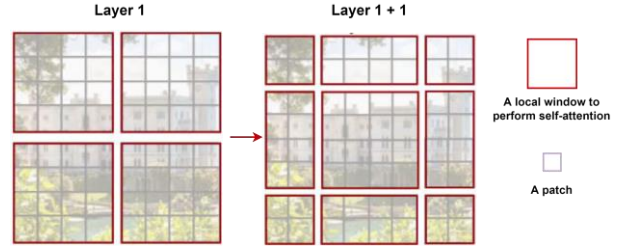


Fig. 4. An illustration of the shifted window approach for computing self attention in the proposed Swin-Transformer architecture [33].

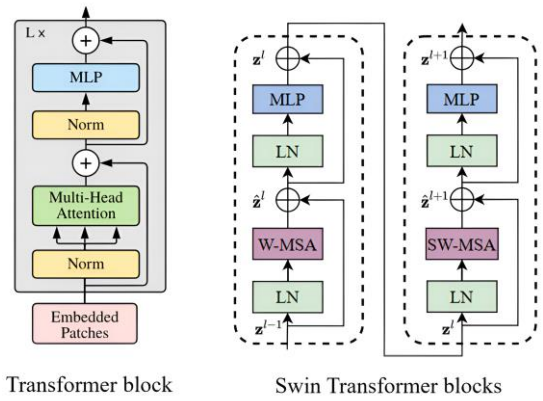


Fig. 5. Structure of Transformer block in ViT [32, 34] and structure of two successive Swin-Transformer blocks [22, 33].

However, the traditional shifted window approach mentioned previously leads to the issue of resulting in more windows in the shifted configuration, from $\left\lfloor \frac{h}{M} \right\rfloor \times \left\lfloor \frac{w}{M} \right\rfloor$ to $\left(\left\lfloor \frac{h}{M} \right\rfloor + 1\right) \times \left(\left\lfloor \frac{w}{M} \right\rfloor + 1\right)$, and some of the windows will be smaller than $M \times M$. A naive solution is to pad the smaller windows to a size of $M \times M$ and mask out the padded values during attention computation, leading to increased computation. Hence, the authors of Swin-Transformer propose a more efficient batch computation approach by cyclically shifting toward the top-left direction, as illustrated in Fig. 6.

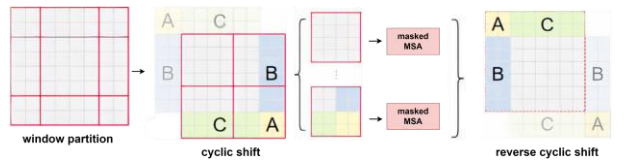


Fig. 6. Illustration of an efficient batch computation approach for self attention in shifted window partitioning [33].

After the shift, a batched window may consist of several non-adjacent sub-windows in the feature map. To address this, a masking mechanism is used to restrict self-attention computation within each sub-window. Despite the cyclic shift, the number of batched windows remains unchanged from regular window partitioning, thereby maintaining efficiency. Using the Swin-Transformer as the backbone

in our encoder helps solve the self-attention computation issues with largescale images and addresses the limitations of the original shifted windows.

E. Feature Enhancement

Following the extraction phase, feature maps extracted from the pre-flood and post-flood inputs of each stage of the two branches in the Siamese network are: $[E_{T1}^1, E_{T1}^2, E_{T1}^3, E_{T1}^4]$ and $[E_{T2}^1, E_{T2}^2, E_{T2}^3, E_{T2}^4]$. Features at higher stages (with larger patch sizes) tend to capture more global information, while those at lower stages focus more on local details.

Before entering the decoding stage, the features undergo an enhancement process. During this stage, the difference between pairs of feature maps from the same stage is computed to highlight the extracted features. Specifically, the feature map at time $T1$ (pre-flood) is subtracted from the feature map at time $T2$ (post-flood), as described by Eq. (3).

$$E_D^k = E_{T1}^k - E_{T2}^k \quad (3)$$

where E_D^k with $k = 1, 2, 3, 4$ represents the enhanced feature map obtained through this differencing operation at each pixel.

This step effectively highlights the discrepancies, particularly the alterations in floodwater, as water regions typically exhibit lower pixel values compared to other areas in the satellite image, allowing us to detect different regions without the need for an absolute operation during subtraction.

F. Flood Prediction

Inspired by the feature pyramid [35], we propose a progressive change prediction approach, as illustrated in the decoder part of Fig. 2. The decoding process for predicting flood regions occurs at each level corresponding to each stage of the Swin Transformer. Subsequently, the predictions from each level are integrated through concatenation operations to generate the final flood region prediction. This methodology enables the capture of flood regions of various scales that undergo significant changes.

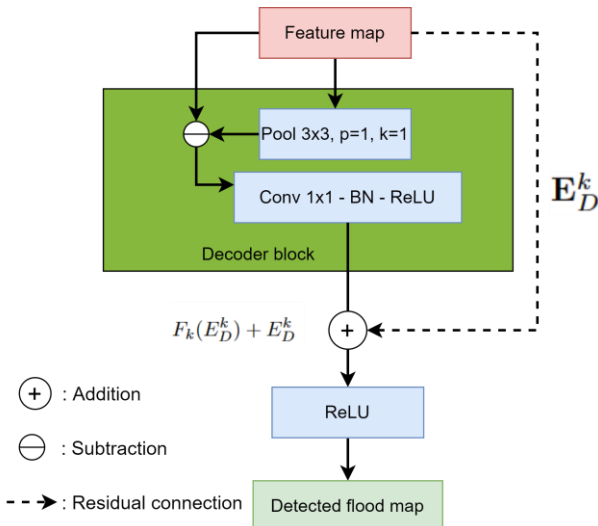


Fig. 7. Illustration of the process within each decoder block.

Furthermore, we incorporate the contrast feature associated with each local feature, as introduced by Luo *et al.* [36], to emphasize changes within the local context. Within each decoder block (Fig. 7), the differentiated feature map is subtracted from the value obtained by average pooling of 3×3 pixels. This step filters out redundant contextual information, focusing only on detected changes that exceed the average, thereby identifying flood regions. Formally, this contrast feature approach is represented by Eq. (4).

$$H_k = \text{ReLU} \left(\text{BN} \left(\text{Conv} \left(E_D^k - \text{Pool}(E_D^k) \right) \right) \right) \quad (4)$$

where:

- Conv 1×1 : Convolution operation ensures the number of channels in the feature maps remains consistent, in this case, it is denoted as *Conv*.
- BN: Batch Normalization operation normalizes the feature values within each batch to a standard distribution, enhancing the stability of the learning process.
- ReLU: Rectified Linear Unit activation function ensures that the values in the feature maps are non-negative.
- H_k : Represents the transformed feature map obtained at layer k .

To obtain the flood prediction map, the predictions from each level are integrated through concatenation operations, as shown in Eq. (5).

$$H_p = H_1 + H_2 + H_3 + H_4 \quad (5)$$

G. Residual Connection

Residual connections, introduced by He *et al.* [37], play a crucial role in mitigating the vanishing gradient problem encountered during the training of deep neural networks. This issue arises due to the attenuation of gradients as they propagate through multiple layers, hindering effective learning.

Residual connections work by introducing skip connections that offer an alternative route for gradient propagation, allowing gradients to circumvent certain layers and directly access earlier layers. Mathematically, a residual connection adds the input of a layer to its output, effectively creating a shortcut connection. In our study, we integrate residual connections bypassing each decoder block, as illustrated by the dashed arrow in Figs. 2 and 7. This integration can be expressed by Eq. (6).

$$H_k = F_k(E_D^k) + E_D^k \quad (6)$$

where, H_k represents the output of the layers with the residual connection, E_D^k denotes the input to the layer, and $F_k(E_D^k)$ represents the transformation applied by the layer.

The inclusion of residual connections enables the model to learn residual mappings, which are easier to optimize compared to directly learning the desired mappings. This is because the network can adjust its predictions by considering both the original input and the transformation applied by the layer. Overall, the integration of residual

connections enhances the learning process by promoting smoother gradient flow and optimizing model training.

H. Loss Function

The loss function serves to measure the disparity between predicted values and ground truth labels during the model training phase, essential for evaluating model performance and guiding model optimization strategies. To enhance our model’s learning process, we adopt the logarithmic loss function discussed in [38]. This function relies on predicted probability values for each sample, derived from the conditional probability distribution. Higher probabilities for correct predictions yield smaller loss values, while lower probabilities result in larger loss values.

In this study, the overall loss for supervised training is defined as the sum of the loss from the final fusion prediction (L^f) and the losses from each side-prediction (L^p), weighted by coefficients α_p , as shown in Eq. (7).

$$L = L^f + \sum_{p=1}^A \alpha_p L^p \quad (7)$$

where, L^f represents the loss of the final fusion prediction, and L^p represents the loss of the p -th side-prediction.

Both L^f and L^p are computed using the Cross-Entropy (CE) loss function, a popular logarithmic loss function in machine learning for assessing classification model performance. The standard binary cross-entropy loss function is defined in [39] as shown in Eq. (8).

$$L_{CE} = -\frac{1}{N} \sum_{x=1}^N [g(x) \log(p(x)) + (1 - g(x)) \cdot \log(1 - p(x))] \quad (8)$$

In our context, CE evaluates prediction loss for each pixel between two classes: “flood” (label 1) and “non-flood” (label 0). Here, N is the number of pixels in a sample, $g(x)$ represents the ground truth label of pixel x ($g(x) \in \{0,1\}$), and $p(x)$ is the probability of pixel x being classified as a flood pixel after passing through the ReLU activation function.

For pixels with ground truth “flood” ($g(x) = 1$), the loss value at pixel x is $l = -\log(p(x))$. Smaller losses occur when the predicted probability $p(x)$ is high (correct prediction), while larger losses result from lower probabilities. Conversely, for pixels labeled “non-flood” ($g(x) = 0$), the loss value at pixel x is $l = -\log(1 - p(x))$. Here, smaller losses occur when $p(x)$ is low (correct prediction), while larger losses result from higher probabilities.

IV. RESULT AND DISCUSSION

A. Dataset

The dataset utilized to train our model is provided by Saleh *et al.* [40]. This dataset comprises a series of satellite images collected from the Sentinel-1 SAR, a satellite system designed by the European Space Agency with the capability to observe various weather conditions, including both day and night. The selected events encompass the most prevalent and representative causes of floods, such as heavy rainfall, river overflow, dam

breaches, tropical storms, and hurricanes. This geographic diversity enables the flood detection method to be sensitive to different geographic contexts and flood scenarios, including rural areas, mountainous regions, urban areas, vegetated land, rivers, ponds, lakes, and reservoirs.

The dataset consists of a total of 4830 image sets, including pre-flood images, post-flood images, and change-labeled images, each with a resolution of 256×256 pixels. All images have 3 channels corresponding to the R, G, B values of the RGB images, while the labels have a single channel with pixel values of 0 (flood) and 1 (non-flood). Fig. 8 provides insight into some samples of the dataset.

To ensure the effectiveness of the flood detection model training process, the dataset is divided into two subsets: a training set comprising 4300 image sets, which accounts for 90% of the entire dataset, and a test set consisting of 530 image sets, representing 10% of the total dataset. To optimize the learning process, enhance generalization, and prevent overfitting, the subsets within the training set are further randomly divided into two smaller subsets: a training subset and a validation subset, with an 80:20 ratio during model training in this study.

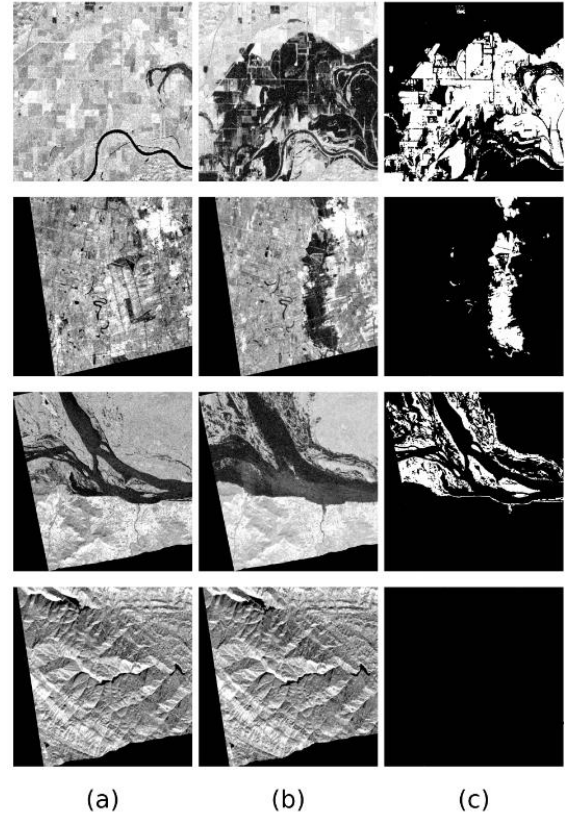


Fig. 8. Samples from the S1GFloods dataset. The images displayed in (a) and (b) are Sentinel-1 SAR imagery of pre- and post-flood events, respectively. Image (c) shows the corresponding annotation maps indicating the ground truth flooded areas with pixel precision.

B. Comparative Methods

We compare our proposed model with state-of-the-art methods for change detection in remote sensing images, applied to global flood detection. Specifically, we present

results from three CNN-based methods and one ViT-based method, summarized in Table I and detailed below:

- **FC-Siam-Diff:** This method integrates the Siamese architecture with a Fully Convolutional Network (FCN) and UNet, extracting features from pairs of satellite images and linking feature maps that capture differences between encoder layers and their corresponding decoder layers.
- **DTCDSN:** This method develops self-attention mechanisms to capture more distinct features and expands CNN's perceptual area to gather more information in a broader context.
- **Siam-NestedUNet:** This method develops a model based on the encoding and decoding architecture of NestedUNet with a Siamese encoder backbone.
- **BIT:** This method combines a CNN for feature extraction, a Transformer for global correlations, and an FCN to create a change map from the feature map differences.

Table I provides a comparative analysis of various models used for flood detection from SAR images. The table includes the models, their corresponding references, architecture types, and sizes in kilobytes. Specifically, it lists the following models: FC-Siam-Diff, DTCDSN, Siam-NestedUNet, BIT, and the proposed model. Each model's architecture (CNN or ViT) and size are detailed, highlighting the computational footprint and complexity of the models. The numerical results demonstrate the relative efficiency and compactness of the proposed model, with our ViT-based method having a size of only 103.952 KB compared to existing models, thereby showcasing its effectiveness and lower computational requirements.

TABLE I. COMPARATIVE MODELS

Model	Study	Architecture	Size (KB)
FC-Siam-Diff	[7, 13]	CNN	258.586
DTCDSN	[21, 14]	CNN	126.521
Siam-NestedUNet	[19, 15]	CNN	202.049
BIT	[3, 18]	ViT	193.529
Ours		ViT	103.952

C. Model Training

1) *Data Pre-processing:* The data preprocessing steps before feeding into the model training include the following:

Step 1: Image loading: Loading pairs of pre-flood and postflood images along with their corresponding labels using the Python Imaging Library.

Step 2: Transformation: The loaded images are converted into Pytorch tensors to ensure compatibility with the model. Paralelly, inverting label values by taking the difference operation from 1 for each pixel value, as initially the labels had values of 0 for 'flood' pixels and 1 for 'non-flood' pixels, while the model assigns values in the opposite way.

Step 3: Resizing: The images are resized to 348x348 to match the input requirements of the Swin-Transformer model.

Step 4: Data augmentation: The image sets, including pre-flood images, post-flood images, and labels, are

rotated, flipped, and adjusted for brightness according to the values listed in Table II. These data augmentation techniques, widely recognized for improving model performance, have been extensively utilized in previous studies [41, 42].

TABLE II. AUGMENTATION CONFIGURATION SETTINGS

Method	Value
Horizontal Flip	Probability: 50%
Random Rotation	Angle: 15°
Brightness	Range: [0.5, 1.5]

2) *Experimental setup:* The model was developed using the Python programming language. All models were trained using the PyTorch framework on an NVIDIA GRID RTX8000 GPU (8GB). To ensure unbiased comparison of results, the compared models were all trained from publicly available source code provided by the authors.

To ensure consistency and fairness in the comparison, the hyperparameters were set according to the values provided by Chen *et al.* [18]. In their study, the authors developed the BIT model and compared it to other models, including those used for comparison in this study (FC-Siam-Diff, DTCDSN, SiamNestedUNet). BIT was also selected for comparison with the model developed in this study. Specifically, the hyperparameters for training the models are set as shown in Table III.

TABLE III. TRAINING CONFIGURATION PARAMETERS FOR THE MODELS

Parameter	Value
Optimizer	Stochastic gradient descent
Weight Decay	5×10^{-4}
Momentum	0.99
Learning Rate	0.01
LR Scheduler	Cosine
Batch Size	8
Loss Function	Cross-Entropy

3) *Early stopping:* was employed during model training to ensure the model achieved the best possible performance [43]. This method is typically implemented using a callback function called EarlyStopping provided by the Keras library.

In this study, the Cross-Entropy loss function was used to monitor the model's improvement. A minimum change threshold for the loss value was set to 0.001. If the loss value dropped below this threshold, the training process was considered to have no further improvement. If the model showed no improvement after 3 consecutive epochs, the training process was stopped. The model parameters from the last epoch were saved and used for testing, considered the resulting model of the training process.

The convergence of each model required a different number of epochs: FC-Siam-Diff (123 epochs), DTCDSN (159 epochs), Siam-NestedUNet (137 epochs), BIT (116 epochs), and Siam-Swin (125 epochs). Figs. 9–13 show the variation of the loss value on the validation set over the epochs during the training of the models.

The number of epochs required for model optimization reflects the relationship between the complexity of the model's architecture and the optimization performance of

the models. Models with a larger number of parameters or a complex architecture typically require a larger number of epochs to converge. Therefore, the DTCDCSCN model with a complex architecture achieved optimization with the highest number of epochs, 159. Conversely, BIT, with a simpler architecture, achieved convergence in the fewest epochs, only 116.

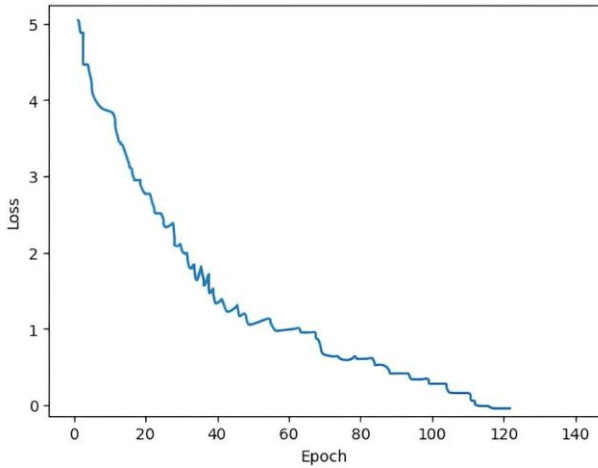


Fig. 9. Validation loss of FC-Siam-Diff over 123 epochs.

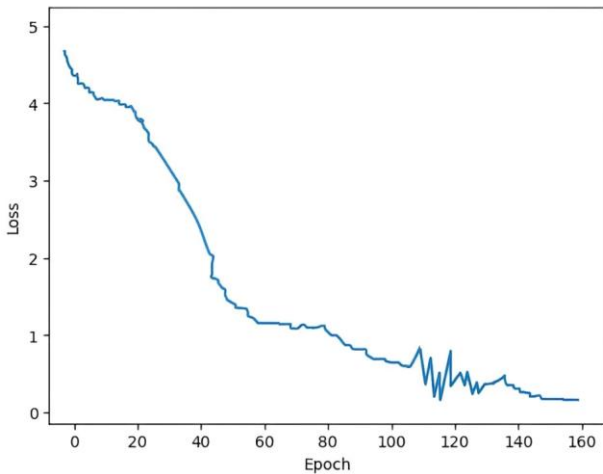


Fig. 10. Validation loss of DTCDCSCN over 159 epochs.

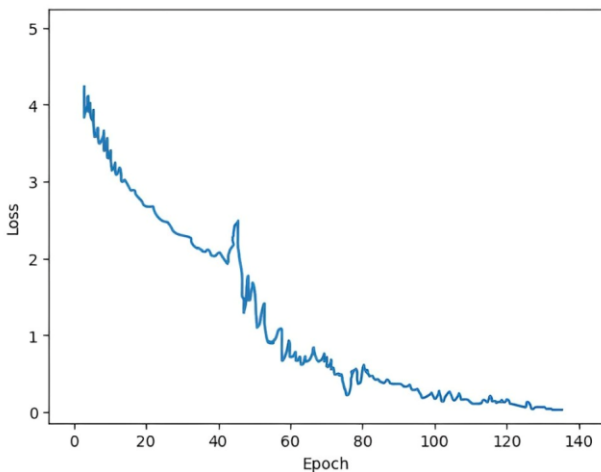


Fig. 11. Validation loss of Siam-NestedUNet over 137 epochs.

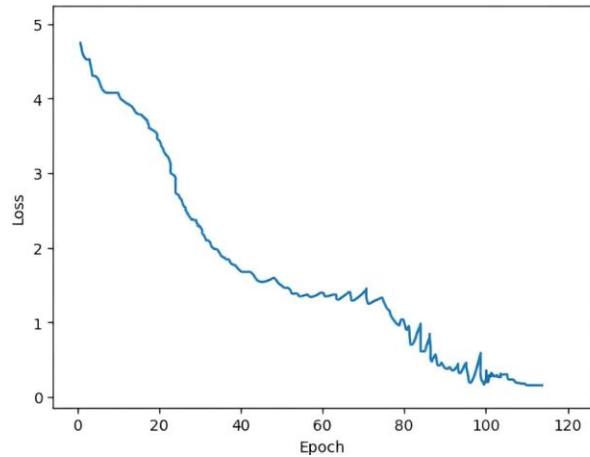


Fig. 12. Validation loss of BIT over 116 epochs

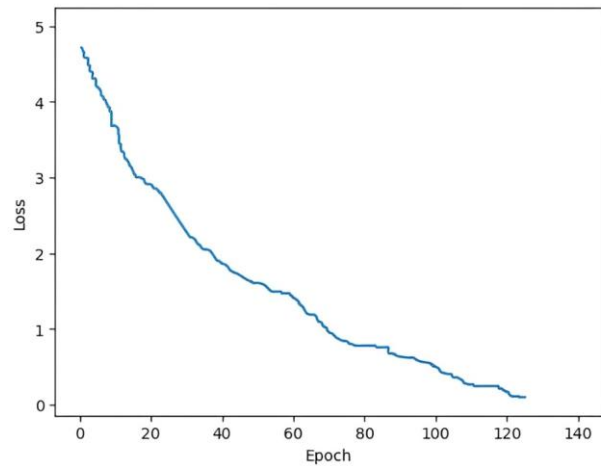


Fig. 13. Validation loss of our proposed model over 125 epochs.

BIT's efficient optimization in 116 epochs demonstrates a balance between speed and optimization efficiency compared to other models. While not achieving the lowest number of epochs, Siam-Swin's ability to optimize after 125 epochs also shows superior performance compared to the CNN models compared in this study.

4) *Learning rate adjustment*: Learning Rate (LR) is a tuning parameter in an optimization algorithm that determines the step size at each iteration while moving toward a minimum of a loss function [44]. During the training process, the LR hyperparameter is dynamically adjusted to optimize the training process and model performance. LR adjustment is a crucial technique in machine learning model training, as it significantly impacts the model's improvement rate. Initially, the LR hyperparameter is set to $LR = 0.01$ for the optimizer using the Stochastic Gradient Descent technique.

Based on the method proposed by Loshchilov and Hutter [45], in this study, LR is adjusted after each epoch using a Cosine function. This method ensures the stability of the convergence process for the model being trained. Instead of abruptly reducing LR at each epoch, the Cosine function allows the model to make smaller and more refined adjustments. This enables a gradual approach to the minimum of the loss function. Consequently, the LR value

is carefully examined, and the optimal value is determined more precisely. The LR update formula used is as shown in Eq. (9).

$$lr = \frac{LR}{2} \left(1 + \cos \left(\frac{T}{T_{max}} \pi \right) \right) \quad (9)$$

where:

- T : The value of the epoch at the time of LR adjustment.
- T_{max} : The maximum number of epochs.

D. Performance Metrics

To quantitatively evaluate the performance of our model, we utilize precision, recall, and F1-Score. Precision measures the accuracy of positive predictions relative to the total number of positive predictions made by the model. Recall, on the other hand, quantifies how many relevant positive instances the model captures. F1-Score provides a single metric that balances both precision and recall, particularly valuable in scenarios with imbalanced class distributions. These metrics are computed as follows:

$$Precision = \frac{TP}{TP+FP} \quad (10)$$

$$Recall = \frac{TP}{TP+FN} \quad (11)$$

$$F1 - Score = 2 \times \frac{P \times R}{P+R} \quad (12)$$

where:

- TP (True Positives): pixels correctly predicted as “flood”.
- TN (True Negatives): pixels correctly predicted as “non-flood”.
- FP (False Positives): pixels predicted as “flood” but actually belong to the “non-flood” class.
- FN (False Negatives): pixels predicted as “non-flood” but actually belong to the “flood” class.

E. Comparison Results

After being trained, the models will be compared based on evaluation results when executing the models on the test set of the S1GFlood dataset. Table IV shows the results of the models on the S1GFloods dataset. FC-Siam-Diff appears to be the least efficient among all CNN-based methods. On the other hand, the BIT model, using ViT, achieves approximately 95% for all metrics without needing a complex structure like SiamNestedUNet, showing that considering wide dependencies is crucial in flood detection.

However, it’s important to note that while the BIT method doesn’t outperform CNN-based methods in terms of recall, DTCDSN achieves the highest recall (96.7%). This is because CNN networks excel at modeling local

features, aiding in better identification of boundaries and small details of changing areas, alongside the integrated multi-layer feature mechanism that helps reduce information loss.

Hence, our proposed model focuses on both modeling wide dependencies and integrating multi-layer features, achieving a recall of 94.6%, along with the highest precision and F1-score among all compared models, at 96.9% and 95.7% respectively, while using only 32.3G FLOPs. This shows that our model not only performs well in flood detection tasks but also has low computational complexity. These improvements can be attributed to the effective integration of the Siamese architecture with the Swin-Transformer backbone, which enhances feature extraction and change detection accuracy.

Additionally, we analyzed the convergence and learning rate optimization of our model. The learning rate was adjusted dynamically using a cosine annealing schedule, which further improved the model’s performance and training efficiency. The practical applicability of our model for flood mapping and disaster mitigation was demonstrated using the S1GFloods dataset, highlighting its potential benefits for natural disaster response and remote sensing applications.

To facilitate intuitive comparisons, we present representative detection results (Fig. 14), where different colors denote true positives (white), true negatives (black), false positives (red), and missed detections (blue). The first two SAR image pairs showcase diverse geographical features (top to bottom): agriculture, urban area, and flood-affected river. These scenarios pose challenges for flood detection due to factors like agricultural activities, complex urban planning, and seasonal river flow variations. The final image pair depicts a mountainous area unaffected by floods. However, changing shadows on hillsides, caused by variations in viewing angle or satellite orbit, can lead to misclassification as flooded areas. This confusion arises from the spectral similarity (similar reflectance patterns) between dark areas and floodwater.

We can see that FC-Siam-Diff and DTCDSN models exhibit high false positive rates for inherently dark areas due to their limited receptive fields (inability to capture long-range dependencies) and lack of global context in feature maps. Siam-NestedUnet struggles with undersegmentation (incomplete flood area detection). Both BIT and our proposed model achieve comparable accuracy in flood map generation. While BIT experiences misclassification in urban areas, our model is susceptible to confusion with mountain shadows. However, both outperform CNNs due to the ViT architecture’s ability to capture global dependencies and reduce class confusion through enhanced feature and context representation.

TABLE IV. COMPARISON OF EFFECTIVENESS AND COMPLEXITY OF MODELS. THE HIGHEST VALUE OF EACH PERFORMANCE INDEX IS IN **BOLD**. PARAMS AND FLOPS ARE THE NUMBER OF MODEL PARAMETERS AND THE NUMBER OF FLOATING-POINT OPERATIONS INDICATING THE COMPLEXITY OF THE MODEL

Methods	Params (M)	FLOPs (G)	Precision (%)	Recall (%)	F1-score (%)
FC-Siam-Diff	1.2	10.4	87.5	95.6	91.3
DTCDSN	31.6	25.4	90.2	96.7	93.3
Siam-NestedUNet	11.7	109.2	94.5	96.2	95.3
BIT	26.4	23.4	94.8	95.4	95.1
Ours	19.3	32.3	96.9	94.6	95.7

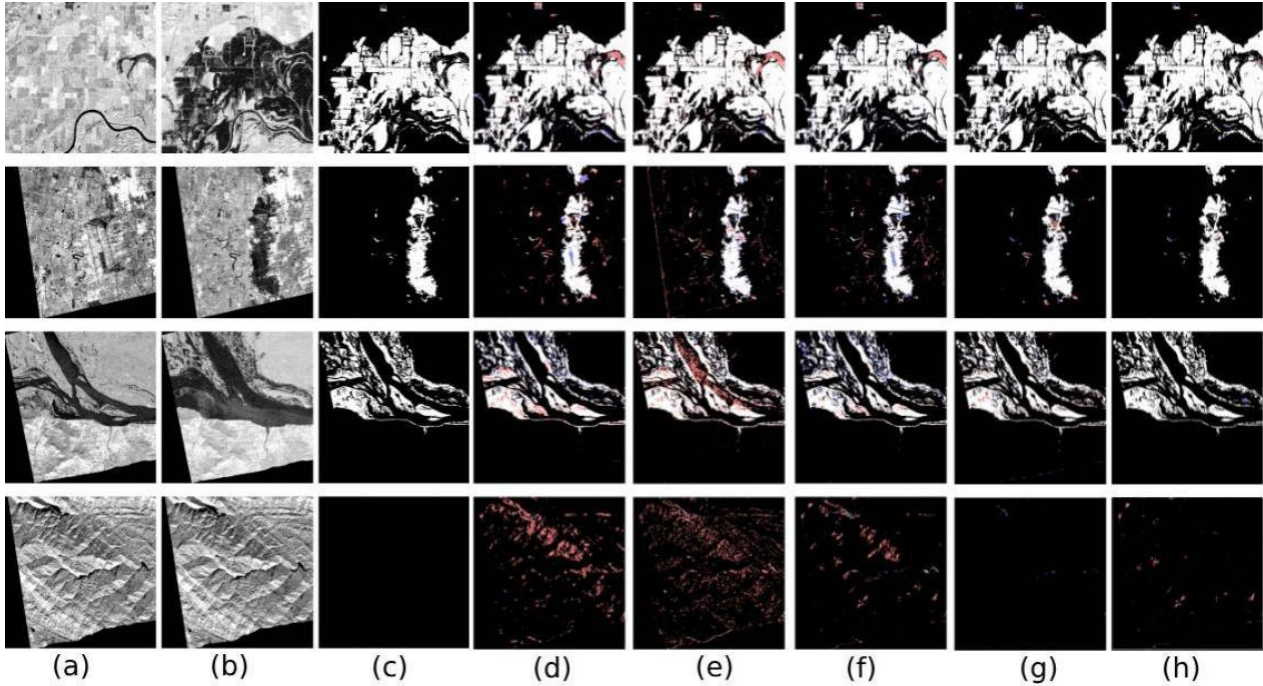


Fig. 14. Visual comparison of model performance. (a), (b), and (c) represent the input images from the dataset: pre-flood image, post-flood image, and ground truth label. (d)–(h) show the corresponding flood detection results from the FC-Siam-Diff, DTCDCSCN, Siam-NestedUnet, BIT, and our proposed models, respectively.

Despite limitations, our model generates detailed flood maps with high fidelity to ground truth labels. This highlights the effectiveness of ViT-based architectures, emphasizing the need for further research to improve generalizability.

F. Large-Scale Flood Mapping

Following the evaluation of our proposed model's performance, we applied it to large-scale flood inundation mapping. To ensure compatibility with the model, the remote sensing data for this experiment originated from the same source as the training dataset. As described in [40] for building the SIGFlood dataset, the data used here was collected from the Sentinel-1 SAR dataset using the Google Earth Engine Editor tool. Sentinel-1, a satellite system designed by the European Space Agency, provides high-resolution data for monitoring natural disasters.

The chosen area encompasses 150,781 km² within Quang Nam province, Vietnam. The imagery was captured on October 17, 2020, coinciding with a period of severe flooding in Central Vietnam from October 6 to 22. Quang Nam province was among the most heavily impacted regions. The terrain in Quang Nam is relatively complex, featuring a gradual west-to-east slope with hilly areas and a dense network of rivers. These rivers and streams often intersect major transportation routes, contributing to a complex geographical system. This complexity is a key factor in the frequent occurrence of severe floods in the low-lying eastern areas of Quang Nam province.

The selected area for mapping is a low-lying coastal region significantly affected by the 2020 flood. The presence of both rivers and urban areas within this region presents a significant challenge for testing the model's performance, particularly in differentiating between floodwater, infrastructure, and permanent water bodies. Fig. 15 illustrates the flood map generated by our model for this large area. Red areas represent flood zones identified by the model, while blue areas depict permanent water bodies (rivers, lakes, etc.). A specific region (highlighted in yellow) is magnified, showcasing both an optical pre-flood image and a satellite image captured during the flood for a more detailed visual comparison.

The model's flood detection results on a large scale demonstrate good accuracy. This is evident when comparing the pre-flood image (Fig. 15b) with the satellite image acquired during the flood (Fig. 15c) for the magnified area. While most of the area comprises residential areas, excluding the visible river branches, the model successfully detects flood inundation within these residential zones. It's important to acknowledge some misidentifications, such as certain river branch locations being classified as flooded areas. Nevertheless, the model displays a strong overall capability for flood identification using remote sensing data, particularly for vast areas like the one tested. This success highlights the model's potential for future applications in real-time flood detection and warning systems.

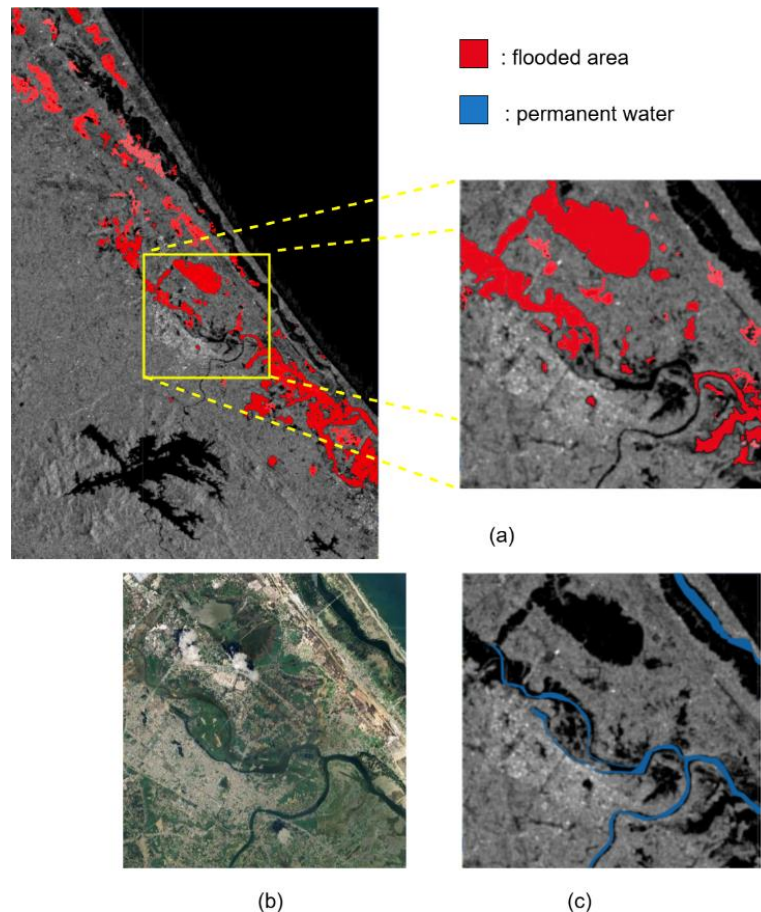


Fig. 15. Flood Detection Map in Quang Nam Province (Vietnam). Red areas represent flood regions identified by the model, while blue areas represent permanent water bodies. (a) The entire selected area for testing, image taken on October 17, 2020. (b) and (c) Enlarged optical image before the flood and satellite image during the flood from the yellow box in (a).

However, this study has some limitations. High computational requirements may limit accessibility for those with fewer resources. The dataset's coverage might not encompass all flood scenarios, affecting the model's generalizability. The need for bi-temporal images limits realtime applicability, essential for timely disaster response. The model's predictions lack interpretability, which is crucial for practical deployment in disaster management. Differentiating floodwaters from permanent water bodies and infrastructure, especially in urban areas, poses challenges. Additionally, complex terrains with varying topography can cause misclassifications, necessitating further refinement for diverse environmental conditions.

V. CONCLUSION

This novel machine learning model represents a notable advancement in applying SAR imagery for flood detection, offering a valuable tool for disaster management agencies with its ability to deliver quick and dependable results even in challenging conditions. However, the study encountered several challenges, including the high computational demands of the deep learning architecture, which could limit accessibility for users with fewer resources. Although the dataset used is comprehensive, it may not encompass all potential flood scenarios, which could impact the model's generalizability. Moreover, the

reliance on bi-temporal images constrains realtime applicability, and the study does not fully explore the interpretability of the model's predictions, which is critical for its practical deployment in disaster management. Future research should aim to enhance the model's accuracy further, broaden its applicability to other disaster types, and investigate integration with additional satellite data sources.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Thanh-Nghi Doan conceptualized the study, designed the deep learning model, and developed the methodology for flood detection using Synthetic Aperture Radar images. Thanh-Nghi also led the software development, conducted experiments, performed data analysis, and wrote the initial manuscript draft, contributing to its revision and editing. Duc-Ngoc Le-Thi significantly contributed to the methodology, assisted in software development, and supervised the project, offering critical insights during the investigation, reviewing the manuscript for intellectual content, and ensuring the study's accuracy and integrity. All authors had approved the final version.

ACKNOWLEDGMENT

This study was supported by the National Geographic Society Exploration, Microsoft AI for Earth, and the technical support from An Giang University, and Vietnam National University in Ho Chi Minh City, Vietnam.

REFERENCES

- [1] L. P. Hoang *et al.*, “Managing flood risks in the Mekong Delta: How to address emerging challenges under climate change and socioeconomic developments,” *Ambio*, vol. 47, pp. 635–649, 2018.
- [2] M. Marchand, D. Pham, and T. Le, “Mekong Delta: Living with Water, But for How Long?” *Built. Environ.*, vol. 40, 2014. doi: 10.2148/benv.40.2.230
- [3] V. Tri, N. Trung, and T. Vo, “Vulnerability to flood in the Vietnamese Mekong Delta: Mapping and uncertainty assessment,” *Journal of Environmental Science and Engineering*, vol. 2, pp. 229–237, 2013.
- [4] M. J. Hammond, A. S. Chen, S. Djordjević, D. Butler, and O. Mark, “Urban flood impact assessment: A state-of-the-art review,” *Urban Water J.*, vol. 12, no. 1, pp. 14–29, 2015. doi: 10.1080/1573062X.2013.857421
- [5] F. Dottori, G. di Baldassarre, and E. Todini, “Detailed data is welcome, but with a pinch of salt: Accuracy, precision, and uncertainty in flood inundation modeling,” *Water Resour. Res.*, vol. 49, no. 9, pp. 6079–6085, 2013. doi: 10.1002/wrcr.20406
- [6] L. C. Smith, “Satellite remote sensing of river inundation area, stage, and discharge: A review,” *Hydrol Process*, vol. 11, pp. 1427–1439, 1997.
- [7] L. Pulvirenti, M. Chini, N. Pierdicca, L. Guerriero, and P. Ferrazzoli, “Flood monitoring using multi-temporal COSMO-SkyMed data: Image segmentation and signature interpretation,” *Remote Sens. Environ.*, vol. 115, pp. 990–1002, 2011. doi: 10.1016/j.rse.2010.12.002
- [8] M. Tanguy, K. Chokmani, M. Bernier, J. Poulin, and S. Raymond, “River flood mapping in urban areas combining Radarsat-2 data and flood return period data,” *Remote Sens. Environ.*, vol. 198, pp. 442–459, 2017. doi: 10.1016/j.rse.2017.06.042
- [9] L. Giustarini, R. Hostache, P. Matgen, G. J.-P. Schumann, P. D. Bates, and D. C. Mason, “A change detection approach to flood mapping in urban areas using TerraSAR-X,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 4, pp. 2417–2430, 2013. doi: 10.1109/TGRS.2012.2210901
- [10] M. Huang and S. Jin, “Rapid flood mapping and evaluation with a supervised classifier and change detection in Shouguang using Sentinel-1 SAR and Sentinel-2 optical data,” *Remote Sens. (Basel)*, vol. 12, no. 13, 2020. doi: 10.3390/rs12132073
- [11] J. Wang *et al.*, “FWENet: A deep convolutional neural network for flood water body extraction based on SAR images,” *Int. J. Digit. Earth.*, vol. 15, no. 1, pp. 345–361, 2022.
- [12] S. Zagoruyko and N. Komodakis, “Learning to compare image patches via convolutional neural networks,” in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4353–4361.
- [13] R. C. Daudt, B. Le Saux, and A. Boulch, “Fully convolutional siamese networks for change detection,” in *Proc. the 2018 25th IEEE International Conference on Image Processing (ICIP)*, 2018, pp. 4063–4067.
- [14] Y. Liu, C. Pang, Z. Zhan, X. Zhang, and X. Yang, “Building change detection for remote sensing images using a dual-task constrained deep Siamese convolutional network model,” *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 5, pp. 811–815, 2020.
- [15] K. Li, Z. Li, and S. Fang, “Siamese NestedUNet networks for change detection of high resolution satellite image,” in *Proc. the 2020 1st International Conference on Control, Robotics and Intelligent System*, 2020, pp. 42–48.
- [16] W. G. C. Bandara and V. M. Patel, “A transformer-based siamese network for change detection,” in *Proc. the IGARSS 2022-2022 IEEE International Geoscience and Remote Sensing Symposium*, 2022, pp. 207–210.
- [17] Y. Du, R. Zhong, Q. Li, and F. Zhang, “TransUNet++ SAR: Change detection with deep learning about architectural ensemble in SAR images,” *Remote Sens. (Basel)*, vol. 15, no. 1, 2022.
- [18] H. Chen, Z. Qi, and Z. Shi, “Remote sensing image change detection with transformers,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2021.
- [19] G. Cheng *et al.*, “Change detection methods for remote sensing in the last decade: A comprehensive review,” arXiv preprint, arXiv:2305.05813, 2023.
- [20] T. Yan, Z. Wan, and P. Zhang, “Fully transformer network for change detection of remote sensing images,” in *Proc. ACCV 2022, the 16th Asian Conference on Computer Vision*, Macao, China, 2023, pp. 75–92. doi: 10.1007/978-3-031-26284-5_5
- [21] M. Esmaeili, D. Abbasi-Moghadam, A. Sharifi, A. Tariq, and Q. Li, “Hyperspectral image band selection based on CNN embedded GA (CNNeGA),” *IEEE J. Sel. Top Appl. Earth Obs. Remote. Sens.*, vol. 16, pp. 1927–1950, 2023. doi: 10.1109/JSTARS.2023.3242310
- [22] R. K. Vincent, “RADAR | Synthetic Aperture Radar (Land Surface Applications),” in *Encyclopedia of Atmospheric Sciences*, 2015. doi: 10.1016/b978-0-12-382225-3.00331-5
- [23] A. Moreira, P. Prats-Iraola, M. Younis, G. Krieger, I. Hajnsek, and K. P. Papathanassiou, “A tutorial on synthetic aperture radar,” *IEEE Geosci. Remote Sens. Mag.*, vol. 1, pp. 6–43, 2013.
- [24] G. Cheng *et al.*, “Change detection methods for remote sensing in the last decade: A comprehensive review,” arXiv preprint, arXiv:2305.05813, 2023.
- [25] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2881–2890.
- [26] A. Vaswani *et al.*, “Attention is all you need,” *Adv. Neural. Inf. Process. Syst.*, vol. 30, 2017.
- [27] W. Yang, X. Yang, T. Yan, H. Song, and G.-S. Xia, “Region-based change detection for polarimetric SAR images using Wishart mixture models,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 11, pp. 6746–6756, 2016.
- [28] P. Chen *et al.*, “A region-based feature fusion network for VHR image change detection,” *Remote Sens. (Basel)*, vol. 14, no. 21, 5577, 2022.
- [29] H. Li, M. Li, P. Zhang, W. Song, L. An, and Y. Wu, “SAR image change detection based on hybrid conditional random field,” *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 4, pp. 910–914, 2014.
- [30] L. Wang and H. Li, “HMCNET: Hybrid efficient remote sensing images change detection network based on cross-axis attention MLP and CNN,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022.
- [31] O. L. F. De Carvalho, O. A. de C. Júnior, A. O. de Albuquerque, N. C. Santana, and D. L. Borges, “Rethinking panoptic segmentation in remote sensing: A hybrid approach using semantic segmentation and non-learning methods,” *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.
- [32] J. Bromley *et al.*, “Signature verification using a ‘Siamese’ time delay neural network,” *Int. J. Pattern Recognit. Artif. Intell.*, vol. 7, pp. 669–688, 1993.
- [33] Z. Liu *et al.*, “Swin transformer: Hierarchical vision transformer using shifted windows,” in *Proc. the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10012–10022.
- [34] G. Sharir, A. Noy, and L. Zelnik-Manor, “An image is worth 16x16 words, what is a video worth?” arXiv preprint, arXiv:2103.13915, 2021.
- [35] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2117–2125.
- [36] Z. Luo, A. Mishra, A. Achkar, J. Eichel, S. Li, and P.-M. Jodoin, “Non-local deep features for salient object detection,” in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6609–6617.
- [37] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [38] Q. Wang, Y. Ma, K. Zhao, and Y. Tian, “A comprehensive survey of loss functions in machine learning,” *Annals of Data Science*, pp. 1–26, 2020.
- [39] Y. Ho and S. Wookey, “The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling,” *IEEE Access*, vol. 8, pp. 4806–4813, 2019.
- [40] T. Saleh, X. Weng, S. Holail, C. Hao, and G.-S. Xia, “DAM-Net: Global Flood detection from SAR imagery using differential

- attention metric-based vision transformers,” arXiv preprint, arXiv:2306.00704, 2023.
- [41] A. Mikołajczyk and M. Grochowski, “Data augmentation for improving deep learning in image classification problem,” in *Proc. 2018 International Interdisciplinary PhD Workshop (IIPhDW)*, 2018, pp. 117–122.
- [42] S. Yang, W.-T. Xiao, M. Zhang, S. Guo, J. Zhao, and S. Furao, “Image data augmentation for deep learning: A survey,” arXiv preprint, arXiv:248240105, 2022.
- [43] L. Prechelt, “Early stopping—But when?” in *Neural Networks*, 1996.
- [44] K. P. Murphy, “Machine learning—A probabilistic perspective,” in *Adaptive Computation and Machine Learning Series*, 2012.
- [45] I. Loshchilov and F. Hutter, “SGDR: Stochastic gradient descent with warm restarts,” arXiv preprint, arXiv: 14337532, 2016.

Copyright © 2025 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).