Environmental Monitoring and Change Detection Using Dual-Stream Convolutional Networks with Fusion Techniques on Satellite Imagery

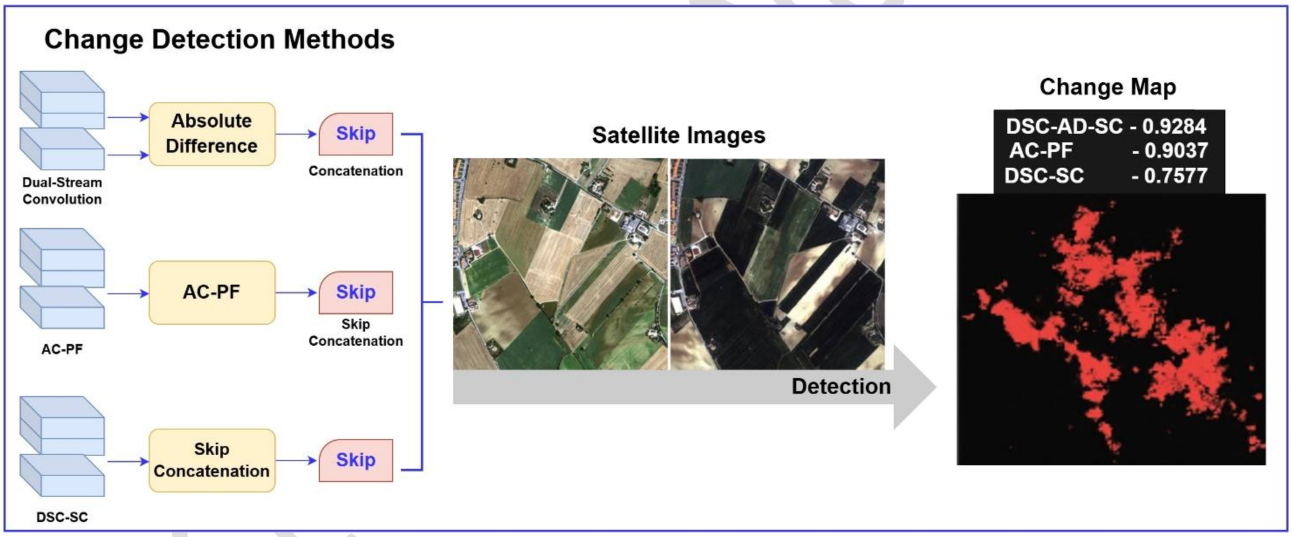
Aiswarya Jeevan[1, *] **and Amala Shanthi S**[2]

[1]Research Scholar, Department of Electronics and Communication Engineering, Noorul Islam Centre for Higher Education, Thuckalay, Kumaracoil, Tamil Nadu, India.

[2]Associate Professor, Department of Electronics and Communication Engineering, Noorul Islam Centre for Higher Education, Thuckalay, Kumaracoil, Tamil Nadu, India.

*Corresponding Author: Aiswarya Jeevan; E-mail: aiswaryajeevan572@gmail.com

Graphical Abstract:



ABSTRACT

Change monitoring on the Earth's surface has become vital for understanding environmental processes and developing sustainable resource management decisions. For numerous applications, ranging from mapping urban sprawl to assessing the impacts of natural hazards, approximating environmental damage, and analyzing forest loss, precision in change detection (CD) methods is paramount. This work entails a thorough examination of enhanced CD techniques that are particularly designed to improve accuracy when detecting changes in satellite data. The three methods that are examined include Dual-stream convolution with absolute difference of skip concatenation (DSC-AD-SC), absolute convolutional prior fusion (AC-PF), and dual-stream convolution with skip concatenation (DSC-SC). To optimize CD accuracy and effectiveness in monitoring changes in geographical features over time, each method employs specialized convolutional operations. The research analyzes environmental changes over time using Onera satellite change detection dataset. A data augmentation step is introduced to the pipeline to enhance dataset diversity and model robustness. Performance is compared using multiple parameters, and the results indicate the highest Dice Similarity Score of 0.9284 for the DSC-AD-SC model, followed by AC-PF at 0.9037 and DSC-SC at 0.7577. The proposed methods are found to be effective in enhancing change detection performance and yield informative measures for use in environmental monitoring and disaster response applications.

Keywords: Change detection, Skip concatenation, Environmental dynamics, geographical features, Environmental monitoring.

1. Introduction

Change detection is an important method for detecting surface change and finds numerous applications in catastrophe evaluation, agricultural monitoring, city planning, ecological monitoring, and map revision. Innovative change detection methods have recently centered on integrated artificial intelligence (AI) technology (Shi et al., 2020). Urbanization, agriculture, and forest removal are some human operations that can significantly disturb water supplies. Between 1972 and 2017, the proportion of the watershed that consisted of cultivated land and agroforestry increased from 24.2% to 62%. Natural vegetation has lost a large amount of land to these two land use types. For the entire watershed, 74.34% of its land cover transformed in 45 years (Degife et al., 2019). Policy initiatives targeted at raising agricultural output remain one of the most effective methods to relieve strain on Nigeria's increasingly restricted land resources while also conserving natural ecosystems (Arowolo and Deng, 2018). Change detection plays an important role, environmental monitoring, and detecting land use and land cover change. Remote sensing satellites gather satellite images with varied resolutions and utilize them to identify changes (Asokan and Anitha 2019). Figure 1 compares the satellite and aerial images of the same location.

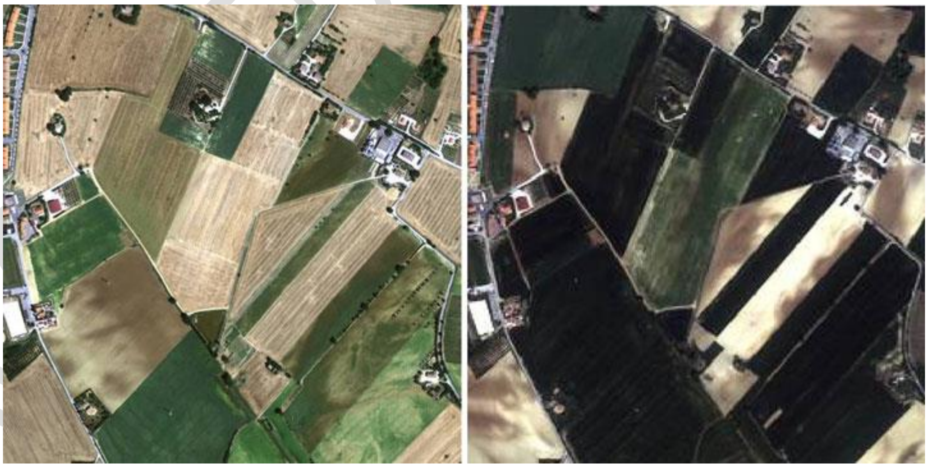


Figure 1. Satellite image and aerial image of the same area

Remote sensing provides extensive geographical and spectral resolution for monitoring desertification, agroforestry, rangelands, soil, water, watersheds, crop use patterns, urban expansion of agricultural land, and monitoring of climate variations and their effects on agriculture. Several programs/projects in India, including FASAL, CAPE, NNRMS, NADAMS, IMSD, CHAMAN, and others, are successfully supporting agricultural management (Kumar et al., 2022). Principal component analysis (PCA), iteratively reweighted-multivariate alteration detection (IRMAD), and change vector analysis (CVA) were used to create three change intensity images. These three intensity images were utilized to construct various binary PBCD maps, which were then merged with the segmented image using wDST to produce the OBCD map (Han et al., 2020). The proposed change detection method was also tested utilizing very high resolution (VHR) satellite images for binary class change detection to map a natural disaster-impacted area, and the results were evaluated against reference data from the Federal Emergency Management Agency (FEMA) (Liu et al., 2021). The training regions are automatically generated using the geographical information system (GIS) database. Even if the problems mentioned above are addressed, there remain a lot of inherent issues in RS datasets that still have not been tackled due to evolving requirements and diverse data, such as heterogeneous data, multiresolution images, and global knowledge of large-scale and high-resolution images (Walter, 2004; Shafique et al., 2022).

Kalinicheva et al., (2020) introduced an unsupervised method of satellite image (SITS) change detection and clustering. Bitemporal change masks were designed for every next image pair in their system based on neural network autoencoders. The system that incorporated graph-based method with unsupervised feature learning via a neural network was independent of the duration and temporal resolution of the SITS. They also highlighted the issue of the algorithm having difficulty in interpreting changes that occurred over different time periods, particularly the requirement that a graph must be constructed to screen for objects.

Song et al., (2020b) proposed an object-based change detection for high-resolution satellite images. The process consists of two main steps. Initially, Objects were constructed based on unsupervised CD methods to train the network, which comprised LSTM layers. To identify changes in satellite images, Sefrin et al., (2020c) merged a fully convolutional neural network (FCN) with a LSTM network. Whereas the FCN was limited to processing input data with one temporal dimension, the hybrid FCN-LSTM method employed sequential information for multitemporal data. Both variable and fixed sequences were supplied and fixed sequences as training inputs for the combination method. Two change detection techniques were tested by Goswami et al., (2022) with a decision tree algorithm. To compare post-classification, employing the separability matrix, the first technique utilized the decision tree algorithm in order to detect individual images. The second method, image differencing, separated the pixels that were changed from those that remained unchanged through the use of the corner method. The limitations of the study were that algorithm they employed was relatively outdated and the technique was evaluated on a small dataset. Khankeshizadeh et al., (2022) proposed a U-Net deep neural network for forest change detection.

A novel difference image was developed to allow for more accurate discrimination between changed and unchanged regions. R2U-Net was employed in order to discriminate the EFFDI among areas that had changed and those that were unchanged since it preserved geometric shapes more than previous versions of U-Net. Kalinaki et al., (2023) improved the architecture of U-Net using attention methods to emphasize important features and residual links to achieve smooth information passing and gradient propagation. The capability of extracting features accurately enhanced the computer performance compared to Standard U-Net and Attention U-Net. The design choices of the method lead to issues like inadequate feature extraction, lost spatial context, and limited adaptability, all of which would ultimately compromise the accuracy and reliability of the system. Difference-aware attention network was proposed by Mei et al., (2023) for dual-temporal satellite data-based simultaneous building localization and multi-level change detection. The authors developed paired features to activate change-sensitive channels and learn the global change pattern. In summary, while these U-Net–based extensions focus on improving feature extraction or spatial attention, they do not explicitly exploit feature-level absolute differences between temporal image pairs, which is a key aspect addressed in the proposed approach.

For urban flood detection, Tanim et al., (2022) presented an unsupervised machine learning framework that integrates the fuzzy rules, Otsu algorithm, and iso-clustering techniques. According to the results of the performance evaluation of the support vector machine, random forest, accuracy measures of 0.87and 0.69. A study comparing the effectiveness of current deep learning methods in satellite imagery was carried out by Tahir et al., (2022). They used convolutional neural network-based frameworks, such as satellite imagery multiscale rapid detection, and the You Only Look Once (YOLO) method, to perform object detection on a dataset of satellite imagery.

While many change detection methods based on deep learning have been proposed, these approaches have limitations. Existing approaches are mostly based on a single stream or loosely coupled dual-stream architectures, which do not explicitly model the feature-level differences between bi-temporal images. While U-Net–based variants performed better in spatial localization, they focus more on feature fusion and less on temporal differences. While attention-based models are good at selecting influential features, they come with added computational complexity, and subtle changes in the environment. The findings suggest that developing architectures that directly emphasize temporal differences while ensuring spatial consistency is warranted, leading to the proposal of dual-stream models in this study.

Change detection datasets lack diversity due to frequent limitations on geographic location, seasonal variations, or specific imaging conditions. This affects the model's generalizability across several environments and time periods. Therefore, there should be larger and more comprehensive datasets that reflect a wider variety of geographical and temporal changes. Handling VHR data is another issue. The pixel-based change detection methods are unsuitable due to the computational requirements and radiometric discrepancy of VHR data. Although object-based approaches yield improved accuracy, research must still establish techniques capable of appropriately handling high-resolution data without degrading output. Most change detection algorithms are not compatible with other sensors or resolutions, as they are generally designed for a specific sensor or resolution. Another domain that requires enhancement is the use of multimodal data. In most of the existing approaches, while only spectral information from optical/SAR images is employed, there is an opportunity to study multimodal approaches employing extra data sources like LiDAR, multispectral/hyperspectral images. Eventually, there is little inquiry into sophisticated structures. Many existing methods utilize conventional DL models, which miss the opportunity for more complex designs, such as Self-Supervised Learning, Dual-Stream Networks, as well as attention processes. Better satellite analysis architectures can detect more complex temporal and spatial correlations in satellite images, resulting in better interpretation and detection of changes. This means that improving satellite image change detection in these aspects can enhance its scalability and reliability. This will allow its use for more applications such as disaster response and management, environmental monitoring, urban planning.

The major objectives of the study are as follows:

- To conduct a comprehensive evaluation of various change detection techniques, including dual-stream convolution with absolute difference of skip concatenation, absolute convolutional prior fusion, and DSC-SC, to understand their effectiveness and efficiency in detecting changes in satellite imagery.
- To assess and compare the performance of each change detection method using standardized evaluation metrics, namely Dice Similarity Score.

2. Materials and Methods

As shown in Figure 2, this study examines several change detection methods, with a specific focus on the DSC-SC method, absolute difference-based convolutional fusion, and a dual-stream convolutional network without skip concatenation. The strategies are utilized on satellite images acquired at different time intervals to observe the changes in features. All the architectures aim to improve the accuracy of change detection through different feature fusion methods. Their advantages and disadvantages are subsequently examined. Data augmentation is used to increase dataset diversity to increase robustness. Finally, standard performance metrics are used to assess the effectiveness of the methods proposed.

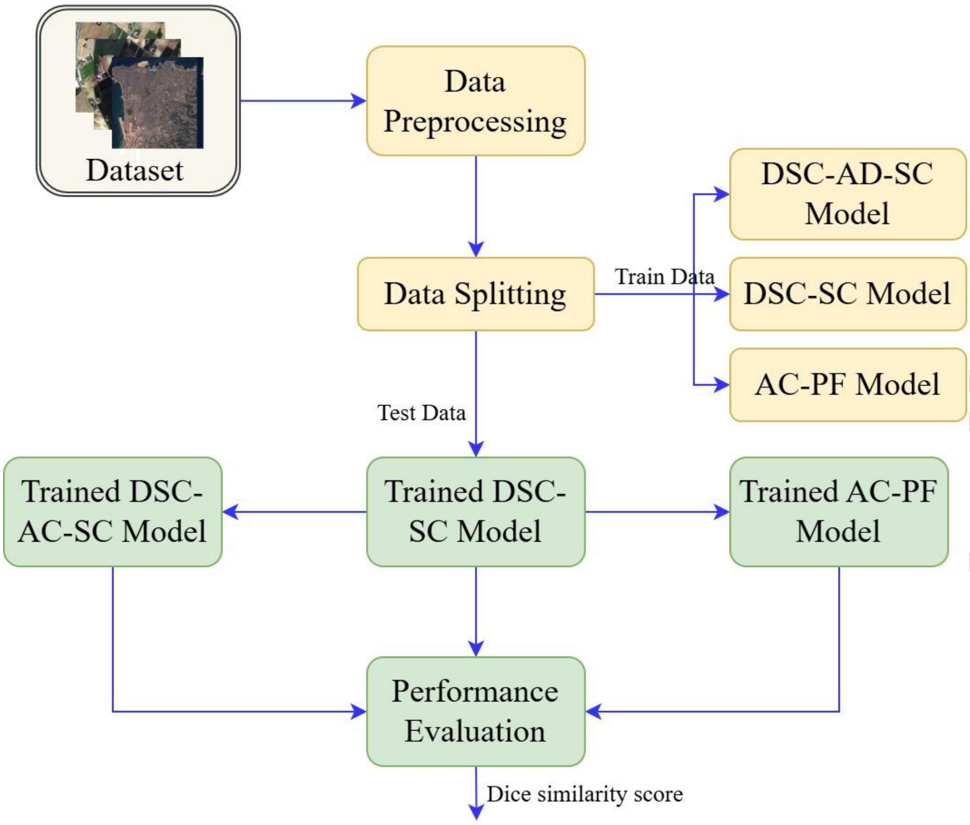


Figure 2. Block diagram of the Proposed approach

2.1 Dataset

The proposed study uses the Onera Satellite Change Detection Dataset which serves as a valuable resource for satellite image change detection research. The dataset includes geographical areas in Brazil, the USA, Europe, the Middle East, and Asia. The dataset includes 24 pairs of multispectral images captured by the Sentinel-2 satellites at during the period 2015–2018. A sample image from the dataset is shown in Figure 3. The dataset contains 13-band multispectral images in each pair.

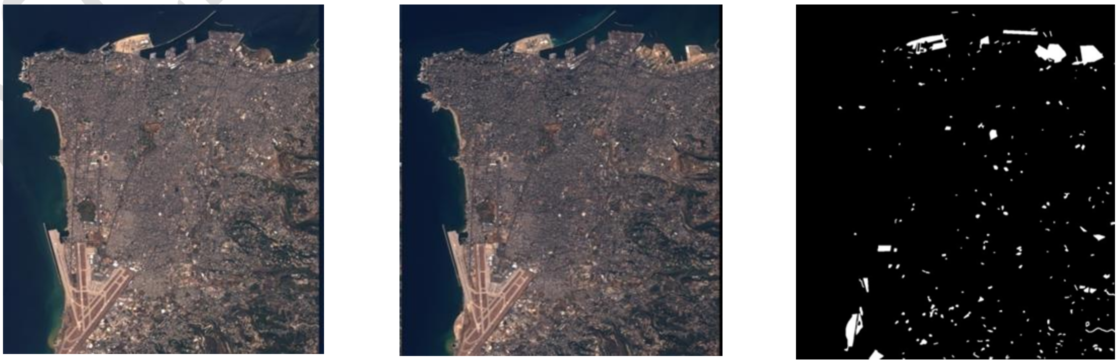


Figure 3. Sample image from the dataset

The dataset is accompanied by pixel-level ground truth masks for the changes, focusing on urbanization. Algorithms can therefore be developed and evaluated for detecting changes.

This study uses all 13 spectral bands of the Sentinel-2 imagery to fully exploit the multispectral information for change detection. These bands enhance accurate spectral and spatial measurement throughout the visible, near-infrared and shortwave infrared wavelengths. They can be used for assessing urban growth, vegetation cover and other land-use changes. Using all available bands allows for the use of a richer feature space and helps the model detect smaller changes in the environment.

2.2 Data Preprocessing and Augmentation

In satellite image analysis, data preprocessing has been widely applied to enhance data consistency and improve feature learning by removing noise and overcoming spatial variations. This study follows the following preprocessing steps namely normalizing spectral bands and bi-temporal images alignment in order to achieve pixel-wise. In order to increase the diversity of the training dataset and make the model more robust, augmentation techniques such as rotation, flipping, and minor spatial alterations are applied as the number of labeled satellite image pairs is limited.

2.3 Model Development

To enhance clarity and accessibility, the workflow of the methodology is clearly explained step-by-step, as well as the role of each architectural component in the change detection process. This dual-stream design allows simultaneous feature extraction from bi-temporal images using shared weights to learn high-quality feature representations. Approaches to feature fusion are implemented at varied stages to focus on temporal differences, thus enhancing sensitivity to minor environmental changes as well as changes. This clarification has been structured for a better comprehension of the proposed methods so that researchers in the fields of remote sensing and deep learning.

The proposed work provides a comprehensive analysis of various change detection techniques. This analysis is based on three key methods. They are: dual-stream convolution with absolute difference of skip concatenation, absolute convolutional prior fusion, and DSC-SC. These approaches are applied to satellite imagery to detect and evaluate changes in geographical features over time.

2.3.1 Change detection using Absolute Convolutional Prior Fusion (AC-PF) model

The AC-PF model is designed as an adaptation of the well-known U-Net architecture. This model operates on two input satellite images (“before” and “after”), which are chipped into smaller patches of size 96 x 96 to ensure that the input is effectively transformed into manageable patches. The choice of a 96 × 96 patch size represents a balance between preserving sufficient spatial context for meaningful change detection and maintaining computational efficiency during model training. This patch size is large enough to capture localized environmental changes while allowing efficient batch processing on limited GPU resources. The ultimate goal of the AC-PF model, as shown in Figure 4, is to produce one output image, highlighting regions of change by comparing the two input images through a semantic segmentation approach.

The AC-PF model retains the foundational encoder-decoder structure of U-Net but incorporates several adjustments to better suit the available training data and to improve computational efficiency. This results in a more comprehensive understanding of the changes between the two images. The encoder part of the network involves several layers of convolutional operations followed by max-pooling. The convolutional operations are mathematically represented in Equation (1).

$$Z^{(l)} = f\left(W^{(l)} * Z^{(l-1)} + b^{l}\right) \qquad (1)$$

Where the feature map at layer l is denoted by $Z^{(l)}$, $W^{(l)}$are the learned weights (filters) of the convolution, the output from the previous layer is denoted by $Z^{(l-1)}$, b^{l}represents the bias term, f is the activation function, and $*$ denotes the convolution operation. The max-pooling operation is performed after every convolution block to reduce the spatial dimensions and retain only the most prominent features. Mathematically, max pooling is described as per Equation (2).

$$P^{(l)}(i,j) = max\left\{Z^{(l)}(i+x, j+y) : 0 \leq x, y < n\right\} \qquad (2)$$

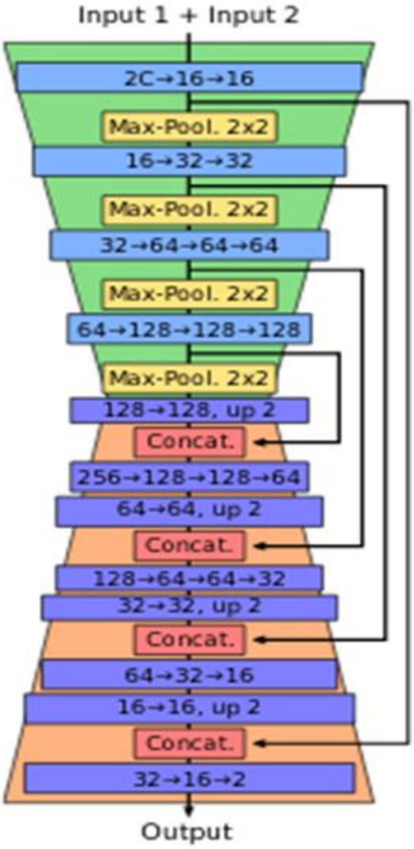


Figure 4. Architecture of the AC-PF model

Where $P^{(l)}(i, j)$ represents the pooled output, and n is the size of the pooling window. This process reduces the resolution of the feature maps while preserving the most important information. In the decoder portion, the model applies more sampling layers to increase the spatial dimensions of the feature maps. By concatenating the feature maps, upsampling is achieved from the encoder, which brings in more detailed information from the earlier layers. The up-sampling operation is mathematically represented as per Equation (3).

$$Z_{up}^{(l)} = Upsampling\left(Z^{(l)}\right) \qquad (3)$$

Where $Z_{up}^{(l)}$ represents the upsampled feature map at layer. The upsampling and concatenation steps ensure that information lost during the downsampling process is recovered, enabling the network to make more precise predictions. The concatenation is represented as per Equation (4).

$$Z_{concat} = Concatenate(Z_{encoder}, Z_{decoder}) \qquad (4)$$

Through the rest of the convolutional layers, this is passed to produce the final output image highlighting the regions of change between two images. The segmentation map is the outcome of the

AC-PF model that predicts changes accurately in the two satellite images. Tuning the hyperparameters properly improves model performance, allowing the model to generalize better with unseen data and achieve the best results. The values of the hyperparameters used in AC-PF model are given in Table 1.

Table 1. Hyperparameters in the AC-PF model

Parameters	**Values**
Loss	Binary cross entropy
Epoch	50
Batch size	2
Activation	ReLU, Sigmoid
Optimizer	Adam

2.3.2 Satellite change detection using Dual-Stream Convolution with Skip Concatenation

The DSC-SC model is a two-stream encoder-decoder architecture that takes two images as input for a change detection task and produces an output of one segmented image. The model illustrated in Figure 5 is derived from the U-Net architecture (Ibtehaz and Rahman 2020), although with a new arrangement of skip connections which combines features from both streams at the decoding stage for accurate class predictions with sharp boundaries.

The encoder part of the DSC-SC model is divided into two parallel streams with the same structure and shared weights. The two streams process one input each independently (“before” and “after”). By sharing weights, both streams extract similar types of features from the two input images, making the architecture computationally efficient while allowing it to capture shared features across the two images. The convolutional operation in the shared encoder is defined as per Equation (5).

$$F = \emptyset(I) = W * I + b \qquad (5)$$

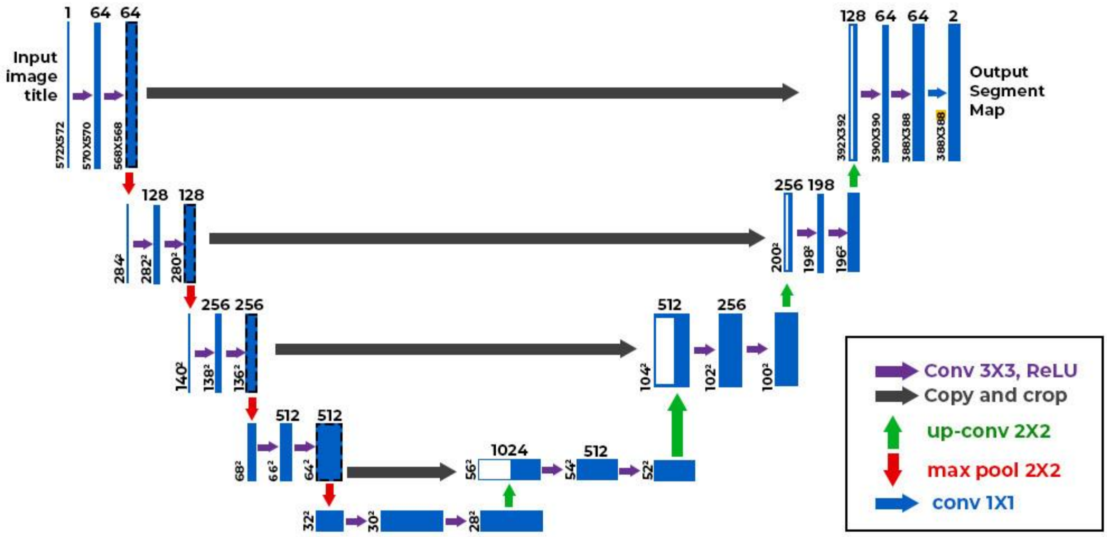


Figure 5. Basic U-Net architecture

Where the feature map generated from the encoder (the same for both streams) is represented by F is, W is the shared convolutional filter weights, I is the input image, and b is the shared bias term. Each branch includes additional branch-specific convolutional layers, as represented in Equation (6).

$$F_{branch} = \Psi(F) \tag{6}$$

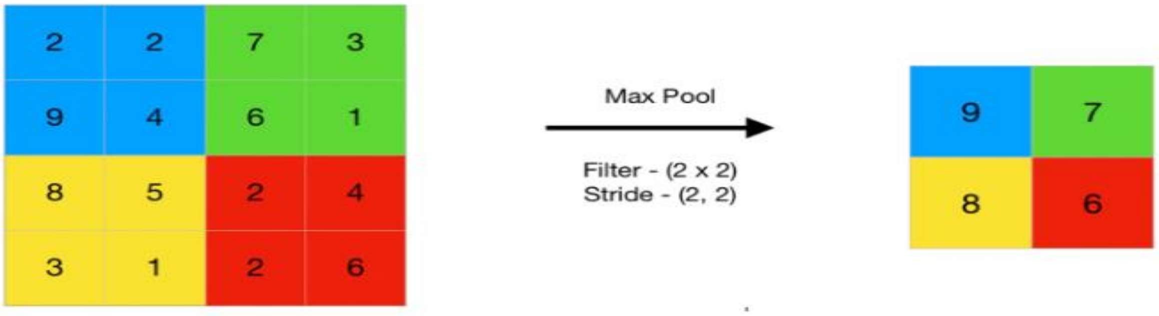
Where F_{branch} is the feature map after branch-specific processing (which differs for each stream), Ψ denotes the branch-specific convolutional layers. The convolutional layers in each stream progressively extract higher-level features, while max pooling layers reduce the spatial resolution of the feature maps, retaining the most prominent features in each stream. The max pooling operation is defined as per Equation (7).

$$P(i,j) = max\{F(i+x, j+y): 0 \leq x, y < n\} \tag{7}$$

Where n is the size of the pooling window and the concatenation is denoted in Equation (8). Figure 6 illustrates the max pooling operation, by selecting the maximum value within a specified window, it effectively downs sampling the input while retaining the most significant features for subsequent processing.

$$X_{concat} = [S_{branch\ 1}, S_{branch\ 2}] \tag{8}$$

Where X_{concat} is the concatenated feature map, $S_{branch\ 1}$ and $S_{branch\ 2}$ are the skip connections from branches 1 and 2, respectively.



2	2	7	3
9	4	6	1
8	5	2	4
3	1	2	6

9	7
8	6

Figure 6. Max pooling operation illustration

This concatenation preserves both low-level features and high-level information, resulting in a richer feature representation. The decoder incorporates the concatenated skip connections from both streams along with up-sampled feature maps from the previous decoding layer. The up-sampling operation enlarges the feature maps to restore the spatial resolution, making it easier for the network to predict fine details in the output segmentation mask. The up-sampling operation is denoted in Equation (9).

$$U = Upsample(F_{decoder}) \tag{9}$$

Where U is the up-sampled feature map from the decoder. The upsampled feature map is then concatenated with X_{concat} and passed through additional convolutional layers in the decoder, as represented in Equation (10).

$$D = \Omega(\mathrm{U} \oplus X_{concat}) \tag{10}$$

Where D is the feature map after processing in the decoder, $\oplus$ denotes the element-wise addition, and Ω are the decoder layer parameters. The final stage of the decoder processes the combined feature maps through additional convolutional layers $'\Omega'$ to refine the output and predict the segmentation mask. The segmentation mask identifies changes between the two input images and assigns class probabilities for each pixel. The segmentation mask is generated using a sigmoid function to convert the final layer's output into probabilities, defined as per Equation (11).

$$M_i = sigmoid(\Omega'(D))_i = \frac{\exp\left((\Omega'(D))_i\right)}{\sum_j \exp\left((\Omega'(D))_j\right)} \tag{11}$$

To a specific class, M_i is the predicted probability for pixel i, $\Omega'(D)_i$ is the output of the final convolutional layer for pixel i, the sigmoid function ensures the output is bounded between zero and one, allowing the model to predict whether each pixel belongs to the change or not. Table 2 outlines the hyperparameters utilized in the DSC-SC model, providing detailed values and configurations employed during the training.

Table 2. Hyperparameters in the DSC-SC model

Parameters	**Values**
Loss	Binary cross entropy
Epoch	50
Learning rate	0.0001
Batch size	2
Activation	ReLU
Optimizer	Adam

2.3.3 Proposed Dual-Stream Convolution with Absolute Difference of Skip Concatenation

The DSC-AD-SC model as shown in Figure 7 is an extension of the previous DSC-SC architecture, specifically designed for tasks requiring the comparison of two input images, such as change detection. The DSC-AD-SC model inputs two images ("before" and "after") and outputs a single image, which is a segmentation mask representing changes between the two images. Similar to the previous method, the input images are divided into 96 x 96 chips for processing.

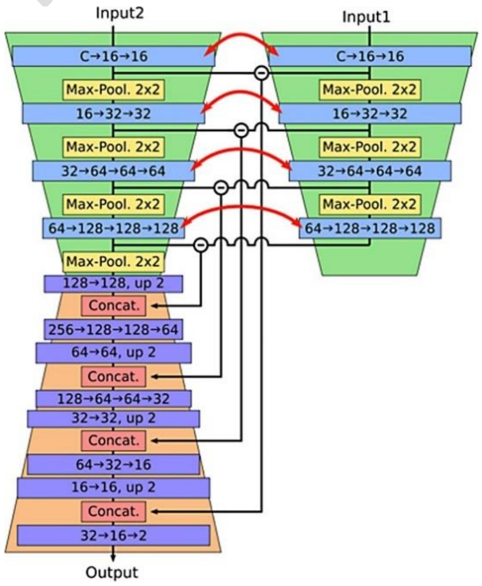


Figure 7. Block schematic of the Proposed architecture

The novel aspect of the proposed model is in the treatment of skip connections. In particular, at the decoding stage, the absolute difference in the skip connections from both streams is computed and concatenated. An identical weight encoder is used in the dual stream architecture where two encoder streams encode features jointly. This is the encoding stage of the DSC-AD-SC model. The stream of processing one of the two input images ("before" and "after") ensures that the features from the images are extracted consistently. The shared convolutional encoder function is defined as per Equation (12).

$$F = \Phi(I) = W * I + b \tag{12}$$

Where the feature map generated by the shared encoder for both branches is represented by F, W represents the shared convolutional filter weights, I is the input image (before or after), and b is the shared bias term. Each stream includes additional branch-specific convolutional layers for further processing of the shared feature map. This branch-specific processing is defined as per Equation (13).

$$F_{branch} = \Psi(f) \tag{13}$$

Where F_{branch} is the feature map after branch-specific convolution operations, Ψ represents the parameters of the branch-specific convolution layers. This step ensures that while the two streams share initial feature extraction operations, they can adapt their feature representations to capture specific characteristics of the "before" and "after" images.

Rather than just concatenating skip connections of both streams (as in the case of the DSC-SC model), the skip connections of the two branches are computed as their absolute difference by the DSC-AD-SC model. This method highlights the differences in the features learned from the "before" and "after" images, thus making the model more sensitive to change. The skip connection operation is represented as per Equation (14).

$$S_{diff} = |S_{branch1} - S_{branch\,2}| \tag{14}$$

Where S_{diff} represents the absolute skip connection difference between branch 1 and branch 2, $S_{branch1} and\, S_{branch\,2}$ represent skip connections from the two respective encoder streams. This operation makes the model concentrate on areas where the two images are not the same, so it is highly appropriate for change detection applications. The absolute skip connections are concatenated with upsampled decoder features. The decoder of the DSC-AD-SC model is based on the general encoder-decoder architecture. The decoder's upsampled feature maps are merged with the absolute difference of skip connections at each step of decoding. The up-sampling operation is denoted as given by Equation (15).

$$U = Upsample(F_{decoder}) \tag{15}$$

Where U is the upsampled feature map from the decoder. The up-sampled feature map is concatenated with the absolute difference of skip connections to form a more enriched feature representation for the decoding process as indicated per Equation (16).

$$X_{concat} = \left[U, S_{diff}\right] \tag{16}$$

Where X_{concat} is the concatenated feature map consisting of the upsampled decoder features and the absolute difference of the skip connections. This combined feature map is further processed through additional convolutional layers in the decoder. In the final stage of the decoder, the combined feature maps are passed through additional convolutional layers to refine the output and generate the segmentation mask. This mask assigns class probabilities (foreground or background) to each pixel, representing areas of change between the input images. The final prediction is generated using a sigmoid function as per Equation (17).

$$M_i = sigmoid(\Omega'(X_{concat}))_i = \frac{\exp\left((\Omega'(X_{concat}))_i\right)}{\sum_j \exp\left((\Omega'(X_{concat}))_j\right)} \tag{17}$$

Where M_i is the predicted probability (indicating change), Ω'represents the final convolutional layers in the decoder, the sigmoid function ensures that the output is a probability distribution, with values between 0 and 1. Figure 8 depicts the model architecture of the DSC-AD-SC model, showcasing the layers and structural components that contribute to its performance in addressing the specified task.

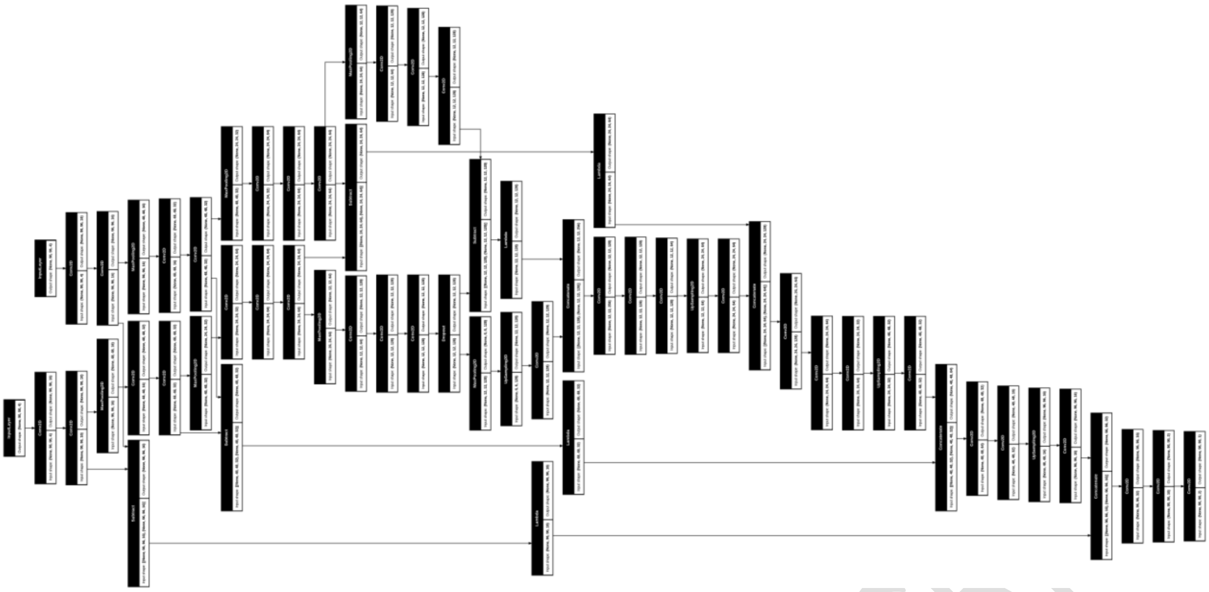


Figure 8. Model architecture of the DSC-AD-SC model

The proposed model, DSC-AD-SC, efficiently extracts and combines spatial features from both inputs for improved binary classification performance. The design is initiated with two input layers that each accept sequences with dimension (96, 96, 4). In addition to this, there are several convolutions and poolings performed per input branch, which ensures that the model operates efficiently, while capturing essential features. Initially, the model consists of two repeated Conv2D layers. The number of filters is 16, with a size of 3×3. Moreover, ReLU is the activation function used. After that, a MaxPooling layer is applied to downsample the feature maps.

Starting from the fusion layer, the model executes up-sampling using Conv2D layers that feature 128, 64, and 32 filters, which help restore image details. The output or final layer is a Conv2d layer with a single filter and the sigmoid activation, making a binary prediction. Essentially, this allows the model a probability score to indicate the classification for each input sequence. The hyperparameters of DSC-AD-SC model mentioned in the Table 3 show the different values used in the training and optimization.

Algorithm 1: *DSC_AD_SC Model for satellite Image change detection*

Input: Dual input image data $X_1, X_2 \in R^{H \times W \times C}$ where H is the weight, W is the width, C is the number of channels.

Output: Predicted segmentation mask with $Y \in R^{H \times W \times 1}$ with values between 0 and 1.

```
Step 1: Data preparation
Input initialization: Load the two inputs $X_1, X_2$ into the model with shape (96,96,4)
Step 2: Define the model
        ➢ CNN Branch 1
    • Apply two 2D convolutional layers with 16 filters, each with a $3 \times 3$ kernel and ReLU activation.
    • Apply max pooling with a $2 \times 2$ pool size.
    • Repeat the process with convolutional layers with 32 and 64 filters, each followed by max pooling.
    • Further apply three 2D convolutional layers with 128 filters each to capture higher-level features.
        ➢ CNN Branch 2
    • Repeat the steps of CNN Branch 1, creating a similar structure with 16, 32, 64, and 128 filters.
        ➢ Merge and Refinement
    • After processing both branches separately, concatenate the feature maps from each branch using `Lambda` layers to compute the absolute difference at each level.
    • Perform up-sampling and additional convolutions to progressively reconstruct the segmentation mask.
        ➢ Output Layer
    • Apply a final 2D convolutional layer with a sigmoid activation to produce the output mask.
Step 3: Compile the Model
    • Compile the model using Adam optimizer, learning rate of $1 \times 10^{-4}$and as the loss function as binary cross-entropy. Set accuracy as the evaluation metric.
Step 4: Training and Evaluation
Step 5: Save the Model
```

Table 3. Hyperparameters in the DSC-AD-SC model

Parameters	**Values**
Learning rate	0.0001
Loss	Binary cross entropy
Epoch	50
Batch size	2
Activation	ReLU
Optimizer	Adam

2.4 Hardware and Software Setup

The models were run on Google Collaboratory as the platform for the workstation. Google Colab enables one to write and run Python scripts, gratis usage of resources like GPUs and TPUs. Google Colab was chosen because it is flexible, user-friendly, and capable of parallel processing all of which are requirements for running deep learning models on large sets of data. Further, Colab's integration with Google Drive facilitates seamless data storage and recovery during model training and testing. The libraries, in addition to Python's extensive community support, provided the instruments necessary for quick model implementation, training, and testing. The Keras library was employed to develop the pieces, with TensorFlow being the backend. TensorFlow's flexibility and support for GPU enabled it to train the model efficiently, and Keras provided an easy way to describe the suggested architecture. Python, Google Colab, Keras, and TensorFlow offered a scalable and efficient framework for building and deploying the model.

3. Results and Discussion

The accuracy of the method for satellite image change detection is assessed by the Dice Similarity Score. It provides a quantitative measure of how accurately the model detects and marked areas of change between the two input images. The Dice score is computed as per Equation (18).

$$Dice\ score = \frac{2\times|X\cap Y|}{|X|+|Y|} \tag{18}$$

where X represents the predicted regions of change, and Y are the ground truth regions. The value ranges from 0 to 1, with a higher score indicating better overlap. A Dice score of 1 denotes that the model's prediction has perfectly overlapped with the ground truth, thereby indicating perfect accuracy in the detection of changes. Taking areas of overlap into consideration allows for the counting of both false positive (areas model wrongly identifies as change) and false negative (real change not identified). Therefore, a high Dice score means that this model is efficiently capturing the true changes and missing the wrong ones rather than the other two.

The improvements observed in the Dice similarity scores particularly for the DSC-AD-SC model matches a recent study, which encourages the modeling of feature differences in such bi-temporal change detection tasks. The same performance improvement has been achieved when leveraging absolute difference-based fusion strategies to enhance sensitivity toward temporal variations in satellite images (Mei et al., 2023; Kalinaki et al., 2023). Also, Shafique et al. (2022) proved that dual-stream methods with shared encoder can achieve better performance with less complicated models. According to the findings, the proposed approach is effective and capable of capturing the environmental changes.

The AC-PF model's training accuracy as shown in Figure 9 reveals an initial increase in accuracy, with validation accuracy consistently hovering around 96.80% over multiple epochs. By epoch 20, the model achieved a stable accuracy of approximately 96.80%. As training progressed, accuracy gradually improved, surpassing 97.50% by epoch 30, although validation accuracy remained stable around 96.5%.

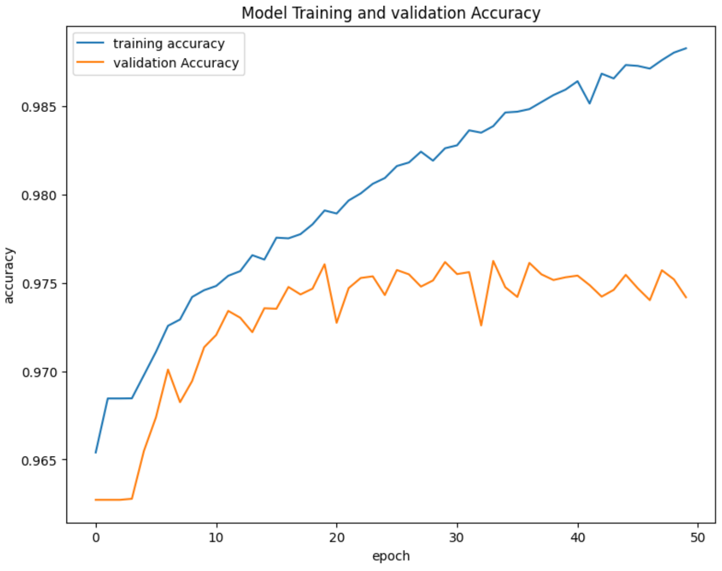


Figure 9. Accuracy plot of the AC-PF model

According to the training of the DSC-SC model shown in Figure 10, there is a sharp increase in accuracy followed by a stabilization phase. The validation accuracy during various epochs remained consistent at 95.20%. By epoch 15, the model achieved 96.10% accuracy. As training continued, the accuracy gradually improved, surpassing 96% by epoch 25. It can be observed that training loss reduced but validation loss fluctuated. This fluctuation indicates the model is suffering from overfitting.

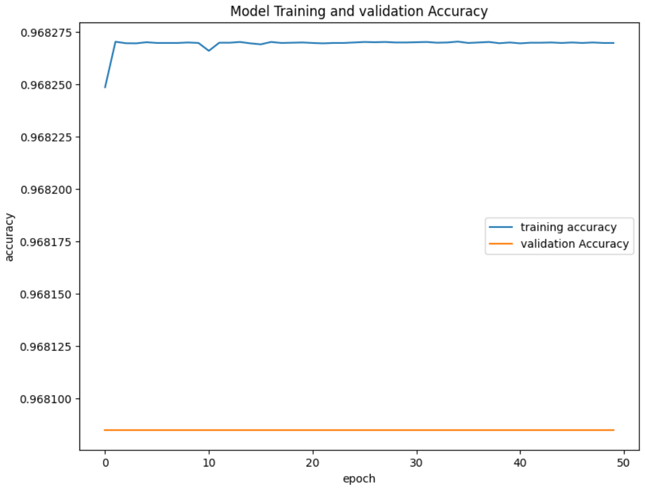


Figure 10. Accuracy plot of the DSC-SC model

The DSC-AD-SC model’s training results, initially improved, as indicated in Figure 11, with validation accuracy remaining around 96.75% throughout many epochs. By epoch 10, the model achieved a consistent accuracy of about 96.75%. As training progressed, the model continued to improve incrementally in accuracy, reaching values above 97% by epoch 30, while validation accuracy remained close to 96.5%.

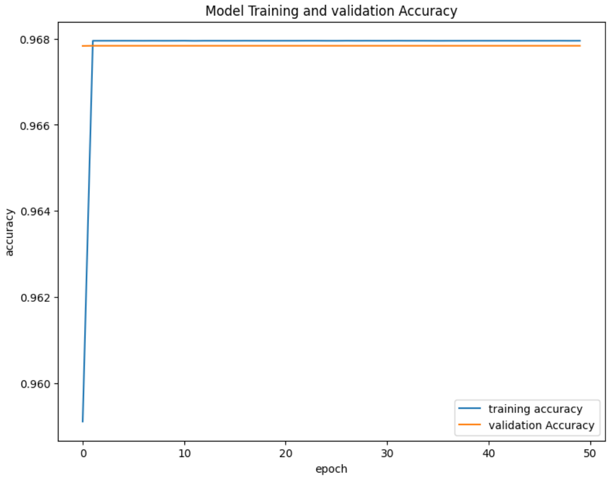


Figure 11. Accuracy plot of the model DSC-AD-SC model

The AC-PF model's loss, as exhibited in Figure 12, exhibited a steady decline in the initial training epochs, indicating effective learning; however, it began to show signs of instability as training advanced, particularly after it reached a lower threshold around 0.10. This instability became more pronounced in the later epochs, where validation loss increased to approximately 0.15-0.17.

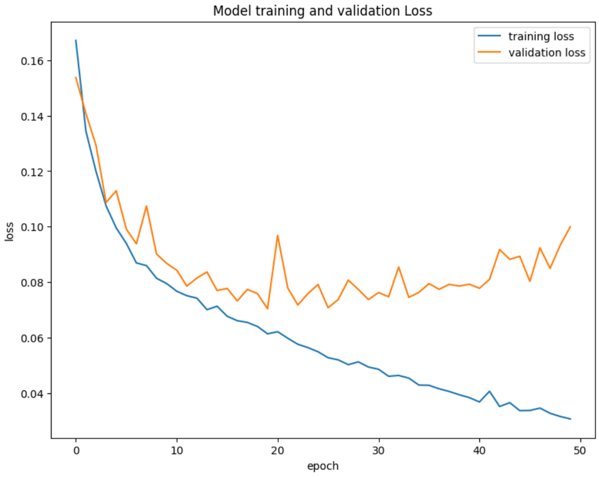


Figure 12. Loss plot of the AC-PF model

The DSC-SC system's loss exhibited a notable decline during the initial phases of training, as shown in Figure 13, signifying effective learning and adaptation to the dataset. However, as training progressed, the loss began to show irregular fluctuations, particularly after reaching a stabilization point around 0.12. This variability became more pronounced in the later epochs, where validation loss increased to values ranging from 0.16 to 0.19, indicating potential overfitting.

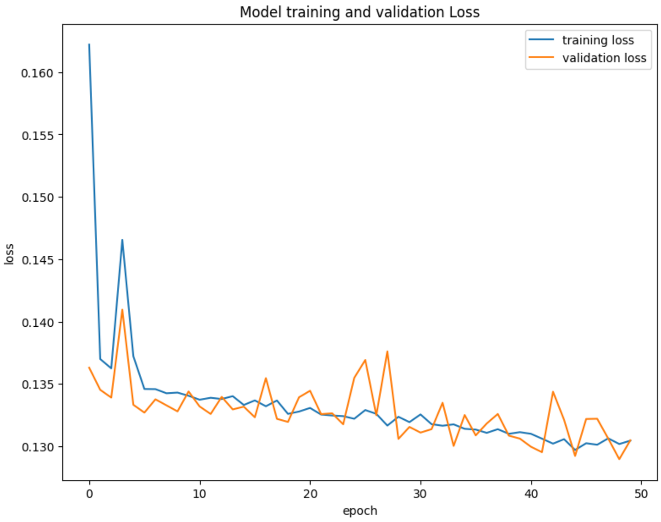


Figure 13. Loss plot of the model DSC-SC model

The DSC-AD-SC model's loss initially decreased, reflecting effective learning in the early training stages, as shown in Figure 14, but began to fluctuate as training progressed, particularly after reaching a lower threshold around 0.13 to 0.14. This fluctuation in validation loss, especially towards the later epochs where it increased to 0.18–0.20, indicates that the model is experiencing some degree of overfitting. As illustrated in Figures 9-11, all three models converge stably during training. In addition, the DSC-AD-SC model achieves higher validation accuracy. The loss curves in Figures 12-14 indicate effective learning behavior, although small fluctuations do suggest overfitting in the later epochs. While some minor signs of overfitting were recognized during training, various measures were taken to prevent it. The sample varieties were increased using the data augmentation so that the models become less sensitive to spatial patterns. Furthermore, shared-weight encoders were used in the dual-stream architectures to limit model complexity and encourage similar feature learning from bi-temporal inputs. Validation performance was also monitored because training beyond convergence was avoided to prevent over-optimization. Despite the small datasets, these strategies collectively help limit overfitting.

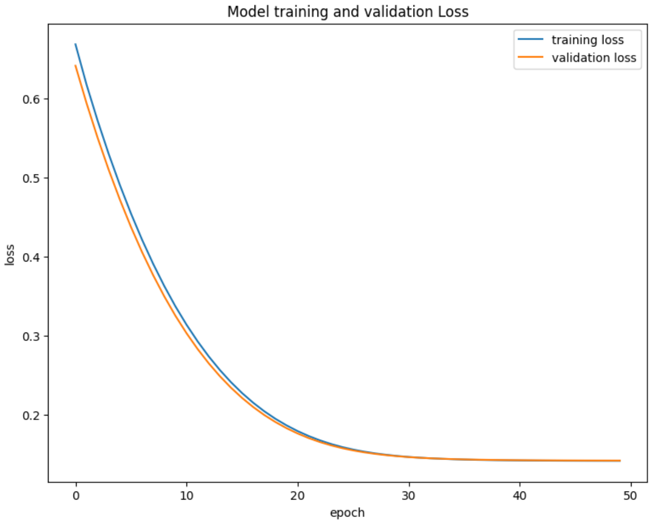


Figure 14. Loss plot of the DSC-AD-SC model

Table 4 presents the Dice similarity scores for various images in the AC-PF model. Figure 15 shows the visualization of these images.

Table 4. Dice similarity score calculation in AC-PF model

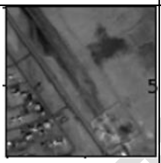
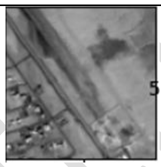
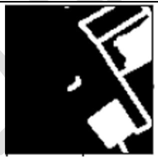
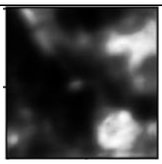
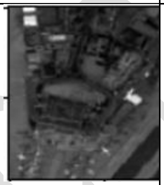
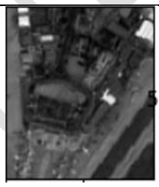
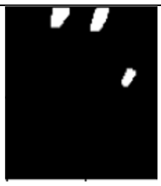
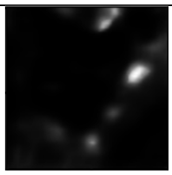
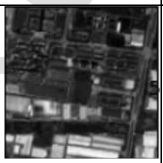
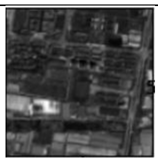
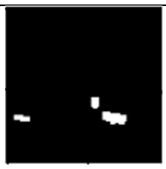
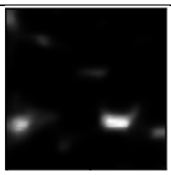
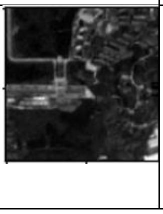
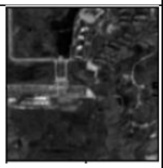
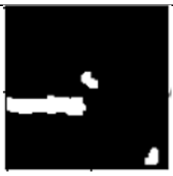
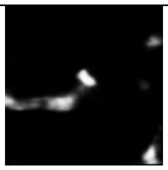
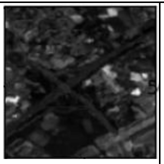
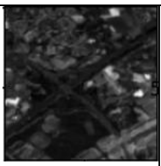
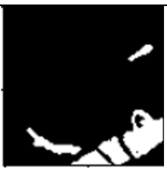
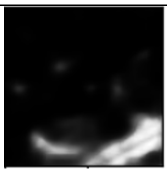
Image	**Initial Image**	**Changed Image**	**Ground Truth**	**Prediction**	**Dice Similarity Score**
Image 1					0.6978368327274562
Image 2					0.59949129493561
Image 3					0.4754219845073826
Image 4					0.6678255080203945

Image 5					0.9037366465587999

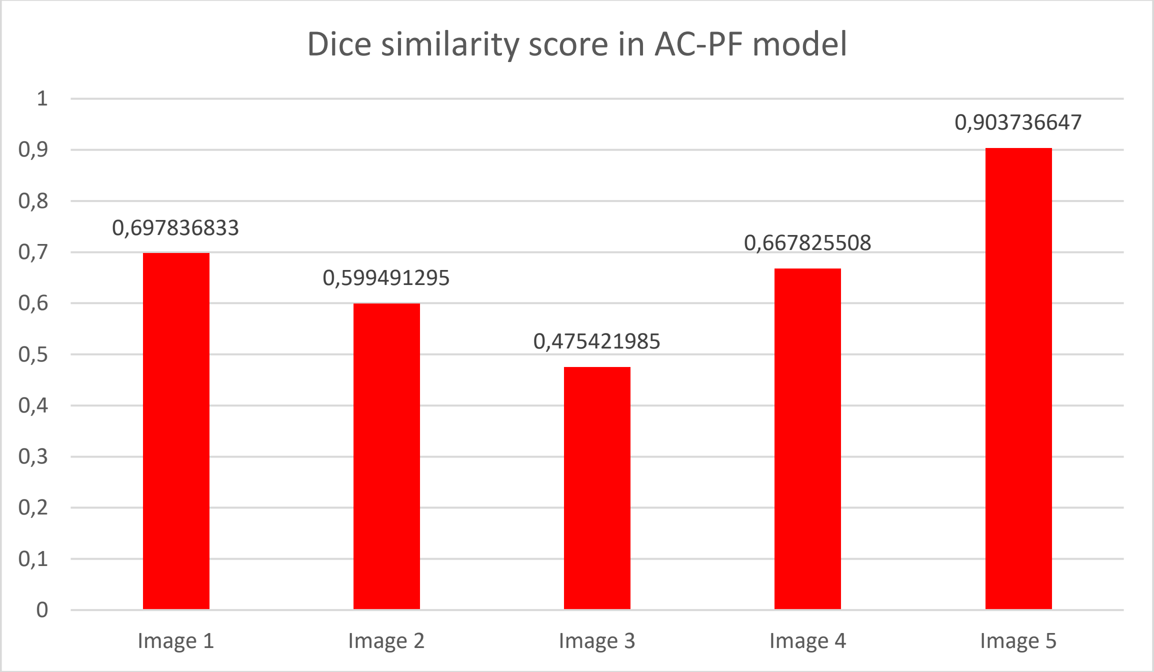


Figure 15. Dice similarity score of different images in AC-PF model

Table 5 presents the Dice similarity scores for various images in the DSC-SC model. Figure 16 shows the visualization of these images.

Table 5. Dice similarity score calculation in DSC-SC model

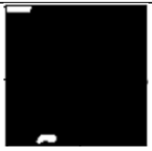
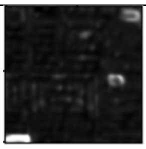
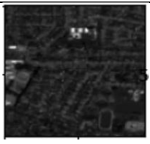
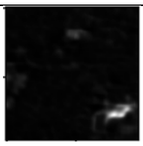
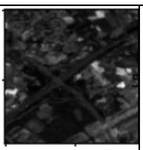
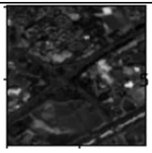
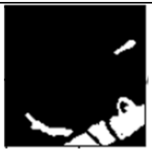
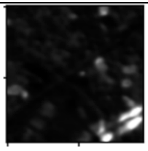
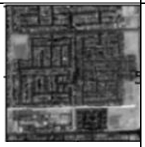
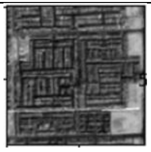
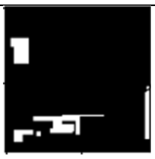
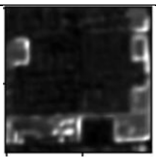
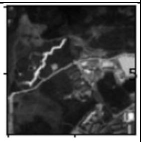
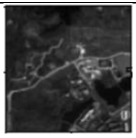
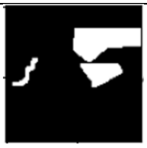
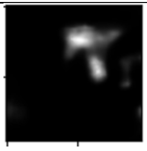
Image	**Initial Image**	**Changed Image**	**Ground Truth**	**Prediction**	**Dice Similarity Score**
Image 1					0.5147571393047798
Image 2					0.6789535143347377

Image 3					0.7359147499259855
Image 4					0.7577480297543387
Image 5					0.6989535143347377

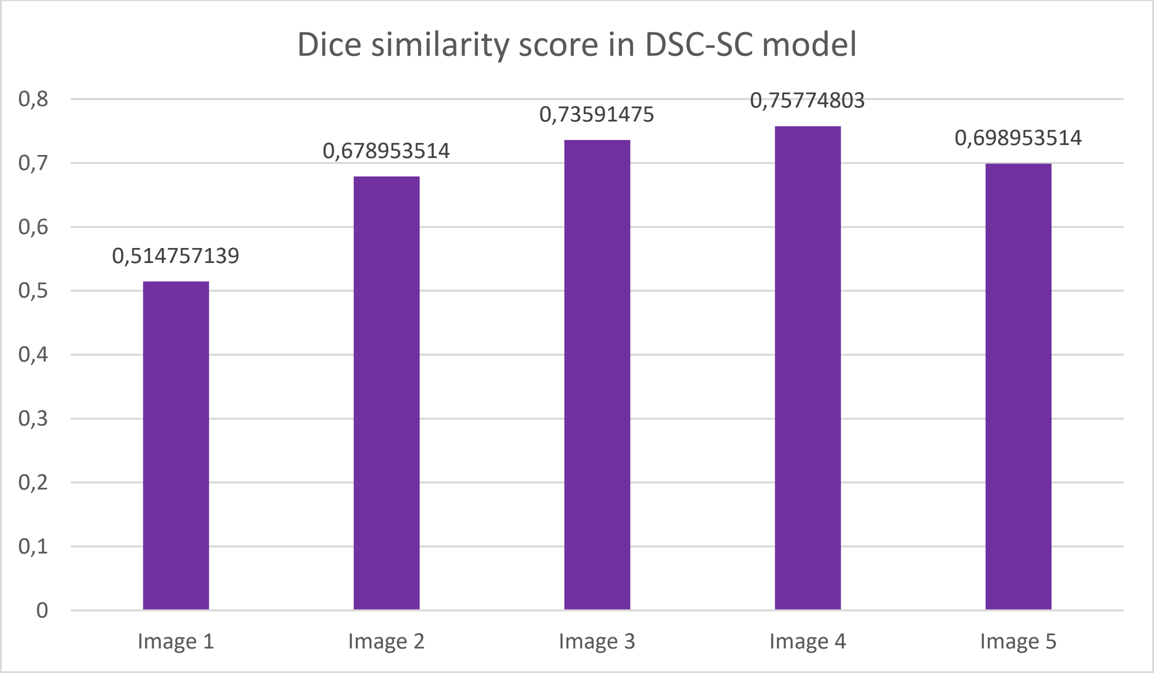
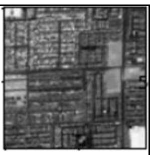
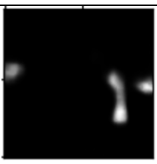
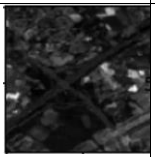
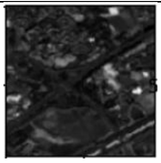
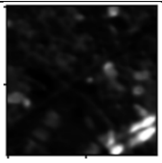
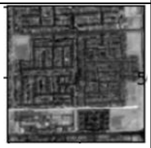
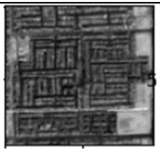
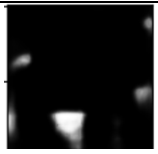
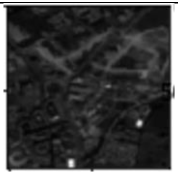
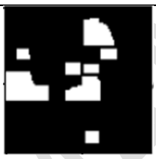
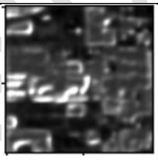


Figure 16. Dice similarity score of different images in DSC-SC model

Table 6 presents the Dice similarity scores for various images in the DSC-AD-SC model. Figure 17 shows the visualization of these images. A comparative analysis of Tables 4-6 highlights that the DSC-AD-SC model consistently outperforms the other architectures across most test images, confirming the benefit of incorporating absolute difference–based feature fusion for change detection.

Table 6. Dice similarity score calculation in DSC-AD-SC model

Image	Initial Image	Changed Image	Ground Truth	Prediction	Dice Similarity Score

Image 1					0.7147571393047798
Image 2					0.7359147499259855
Image 3					0.7577480297543387
Image 4					0.7284033284077162
Image 5					0.9284033284077162

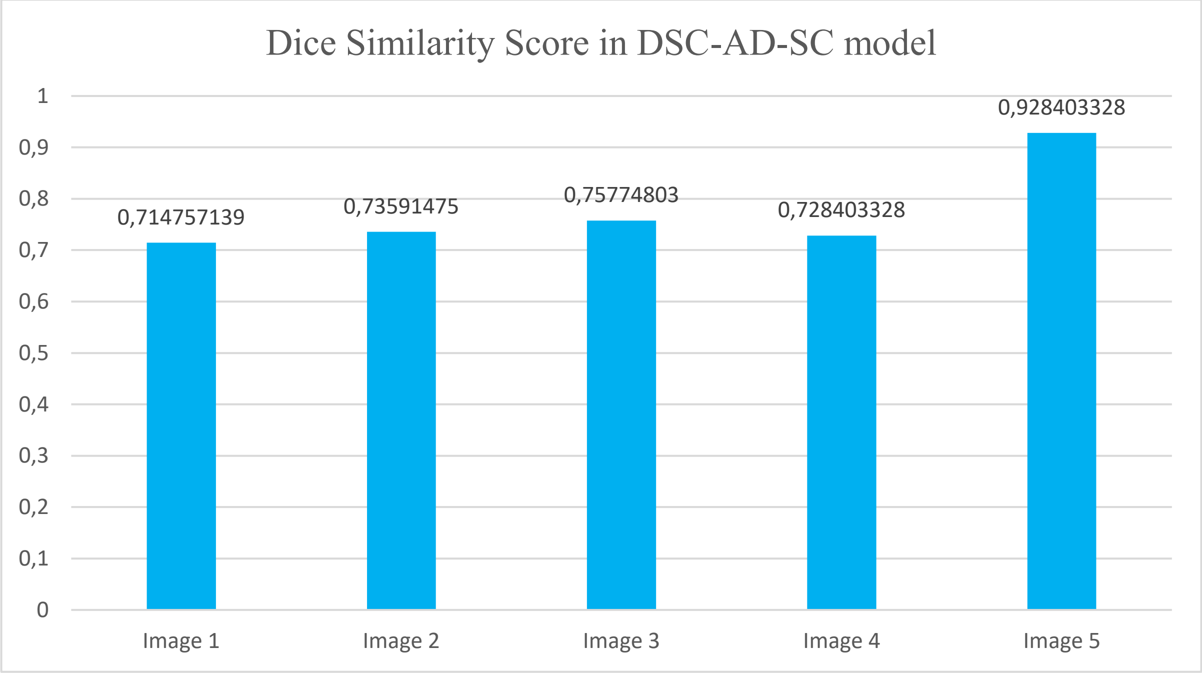


Figure 17. Dice similarity score of different images in DSC-AD-SC model

Although the Dice Similarity Score is a useful quantitative measure for comparing segmentation performance, the current study does not perform formal significance testing or confidence interval estimation. Incorporating repetitions of such analyses across a few runs and datasets would enhance the robustness of the claims. This is an important direction for future work, particularly for the large-scale deployment and benchmarking of change detection models.

In real-time environmental monitoring applications, false positives and false negatives have different effects. When change is not detected, false positives cause unnecessary allocation (e.g., inspecting or intervening). On the other hand, false negatives are much more damaging, as they could mean missing significant environmental changes like deforestation, urban encroachment, and damage from disasters. The DSC-AD-SC model proposed here is designed to additionally raise the sensitivity to actual temporal variations while maintaining the spatial coherence in a model that is less likely to suffer from any of the two failures.

4. Conclusion

Accurate and reliable change detection using satellite imagery plays a crucial role in understanding environmental dynamics related to urbanization, deforestation, and natural disasters. This study investigated three deep learning–based change detection architectures: DSC-SC, AC-PF, and the proposed DSC-AD-SC using the Onera Satellite Change Detection Dataset. Experimental results demonstrated that incorporating absolute difference–based skip connections within a dual-stream framework significantly improves change localization performance. Among the evaluated methods, the DSC-AD-SC model achieved the highest Dice Similarity Score, indicating its effectiveness in capturing meaningful temporal changes while preserving spatial detail.

Besides quantitative performance, the study shows that architectural design choices can remedy the limitations observed in conventional methods for change detection. The method proposed effectively increases the sensitivity to actual changes in the environment such as land-cover change and disaster assessment by modeling the differences that exist at the feature level between two temporal images.

The deployment of change detection systems that use deep learning will face challenges in the real world. When processing a large number of high-resolution satellite images, scalability may be hampered due to high computational requirements. In addition, access to properly labeled and temporally aligned datasets is often limited, which can affect practical deployment. Changes in sensor settings, seasonal conditions, and acquisition settings may influence model generalization further, necessitating caution before operational use.

In the future, these challenges will be addressed through computationally efficient model architectures and improved training strategies. Integrating multispectral and SAR data and other data types is a promising way forward to achieving robustness in various conditions. In addition, expand the evaluation to include a larger set of data, and performing statistical significance testing based on multiple runs can increase confidence in the application of the proposed framework for longer environmental monitoring and decision support.

During the preparation of this work, the author(s) used Microsoft Word, Grammarly and https://www.drawio.com/ in order to improve grammar, enhance language clarity and design technical diagrams. After using these tools/services, the author(s) reviewed and edited the content as needed and take full responsibility for the content of the publication.

References

Arowolo A.O. and Deng X. (2018), Land use/land cover change and statistical modelling of cultivated land change drivers in Nigeria, *Regional Environmental Change*, **18**, 247-259.

Asokan A. and Anitha J.J. (2019), Change detection techniques for remote sensing applications: A survey, *Earth Science Informatics*, **12**, 143-160.

Degife A., Worku H., Gizaw S. and Legesse A. (2019), Land use land cover dynamics, its drivers and environmental implications in Lake Hawassa Watershed of Ethiopia, *Remote Sensing Applications: Society and Environment*, **14**, 178-190.

Goswami A., Sharma D., Mathuku H., Gangadharan S.M.P., Yadav C.S., Sahu S.K. et al. (2022), Change detection in remote sensing image data comparing algebraic and machine learning methods, *Electronics*, **11**, 431.

Han Y., Javed A., Jung S. and Liu S. (2020), Object-based change detection of very high resolution images by fusing pixel-based change detection results using weighted Dempster–Shafer theory, *Remote Sensing*, **12**, 983.

Ibtehaz N. and Rahman M.S. (2020), MultiResUNet: Rethinking the U-Net architecture for multimodal biomedical image segmentation, *Neural Networks*, **121**, 74-87.

Kalinaki K., Malik O.A. and Lai D.T.C. (2023), FCD-AttResU-Net: An improved forest change detection in Sentinel-2 satellite images using attention residual U-Net, *International Journal of Applied Earth Observation and Geoinformation*, **122**, 103453.

Kalinicheva E., Ienco D., Sublime J. and Trocan M. (2020), Unsupervised change detection analysis in satellite image time series using deep learning combined with graph-based approaches, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, **13**, 1450-1466.

Khankeshizadeh E., Mohammadzadeh A., Moghimi A. and Mohsenifar A. (2022), FCD-R2U-net: Forest change detection in bi-temporal satellite images using the recurrent residual-based U-net, *Earth Science Informatics*, **15**, 2335-2347.

Kumar S., Meena R.S., Sheoran S., Jangir C.K., Jhariya M.K., Banerjee A. and Raj A. (2022), Remote sensing for agriculture and resource management, In: *Natural Resources Conservation and Advances for Sustainability*, Elsevier, 91-135.

Liu T., Yang L. and Lunga D. (2021), Change detection using deep learning approach with object-based image analysis, *Remote Sensing of Environment*, **256**, 112308.

Mei J., Zheng Y.B. and Cheng M.M. (2023), D2ANet: Difference-aware attention network for multi-level change detection from satellite imagery, *Computational Visual Media*, **9**, 563-579.

Sefrin O., Riese F.M. and Keller S. (2020), Deep learning for land cover change detection, *Remote Sensing*, **13**, 78.

Shafique A., Cao G., Khan Z., Asad M. and Aslam M. (2022), Deep learning-based change detection in remote sensing images: A review, *Remote Sensing*, **14**, 871.

Shi W., Zhang M., Zhang R., Chen S. and Zhan Z. (2020), Change detection based on artificial intelligence: State-of-the-art and challenges, *Remote Sensing*, **12**, 1688.

Song A., Kim Y. and Han Y. (2020), Uncertainty analysis for object-based change detection in very high-resolution satellite images using deep learning network, *Remote Sensing*, **12**, 2345.

Tahir A., Munawar H.S., Akram J., Adil M., Ali S., Kouzani A.Z. and Mahmud M.P. (2022), Automatic target detection from satellite imagery using machine learning, *Sensors*, **22**, 1147.

Tanim A.H., McRae C.B., Tavakol-Davani H. and Goharian E. (2022), Flood detection in urban areas using satellite imagery and machine learning, *Water*, **14**, 1140.

Walter V. (2004), Object-based classification of remote sensing data for change detection, *ISPRS Journal of Photogrammetry and Remote Sensing*, **58**, 225-238.