European Journal of Remote Sensing

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/tejr20

Change detection using multi-scale convolutional feature maps of bi-temporal satellite high-resolution images

Rasha Alshehhi & Prashanth R. Marpu

To cite this article: Rasha Alshehhi & Prashanth R. Marpu (2023) Change detection using multi-scale convolutional feature maps of bi-temporal satellite high-resolution images, European Journal of Remote Sensing, 56:1, 2161419, DOI: 10.1080/22797254.2022.2161419

To link to this article: https://doi.org/10.1080/22797254.2022.2161419

Published online: 27 Jan 2023.

EUROPEAN JOURNAL OF REMOTE SENSING
2023, VOL. 56, NO. 1, 2161419
https://doi.org/10.1080/22797254.2022.2161419

Change detection using multi-scale convolutional feature maps of bi-temporal satellite high-resolution images

Rasha Alshehhi[a] and Prashanth R. Marpu[b]

[a]Center for Space Science, New York University, Abu Dhabi, United Arab Emirates; [b]G42 Company, Abu Dhabi, United Arab Emirates

ABSTRACT

Change detection in high-resolution satellite images is essential to understanding the land surface (e.g. agriculture and urban change) or maritime surface (e.g. oil spilling). Many deep-learning-based change detection methods have been proposed to enhance the performance of the classical techniques. However, the massive amount of satellite images and missing ground-truth images are still challenging concerns. In this paper, we propose a supervised deep network for change detection in bi-temporal remote sensing images. We feed multi-level features from convolutional networks of two images (feature-extraction) into one architecture (feature-difference) to have better shape and texture properties using a dual attention module We also utilize a multi-scale dice coefficient error function to decrease overlapping between changed and background pixel. The network is applied to public datasets (ACD, SYSU-CD and OSCD). We compare the proposed architecture with various attention modules and loss functions to verfiy the performance of the proposed method. We also compare the proposed method with the stateof-the-art methods in terms of three metrics: precision, recall and F1-score. The experimental outcomes confirm that the proposed method has good performance compared to benchmark methods.

ARTICLE HISTORY
Received 25 April 2022
Revised 11 October 2022
Accepted 18 December 2022

KEYWORDS
Change detection; supervised deep network; dual attention module

Introduction

Change detection is a process to identify disparities in the state of an object from different images of the same area at different times. Monitoring differences has been widely applied for various applications such as urban expansion, vegetation mapping, sea ice, surface water, disaster assessment, planetary surface, etc. (Chen et al. 2019; Parente et al. 2019; Kaiyu et al., 2020; Chen et al. 2020; Mohsenifar et al. 2021; Zhao et al. 2022). Satellite images have been widely used to observe variations in shape and texture properties. However, change detection is still a challenging problem because of the massive amount of digital Earth observations that vary in spatial resolutions from kilometers to centimeters from all kinds of satellite sensors such as Landsat, Worldview and DeepGlobal. Also, many remote sensing studies suffer from the unavailability of labeled observations to train efficient machine learning models.

Change detection methods are categorized into two approaches. The first approach is pixel-based, which is based on the comparison of corresponding pixels from multi-temporal images to produce change maps based on arithmetic operations such as image difference, image ratio, etc., or transformation operations such as principal component analysis, canonical correlation analysis and change vector analysis, etc. (Hussain et al. 2013; Liu et al. 2019). However, pixel-based methods neglect spatial contextual information and unsupervised separate changed pixels from unchanged pixels. The second approach is patch-based, which is based on deriving features from patches or segmentation maps. However, patch-based methods applied in low- or middle-resolution images fail to work in high-resolution images because of the variability of image objects. The classical patch-based methods are mainly based on applying traditional machine-learning-based techniques (e.g. support vector machine, clustering, kernel regression, etc.) after extraction of hand-crafted features (Dengkui et al., 2008; Celik 2009; Luppino et al. 2018). The recent patch-based methods are based on deep learning techniques such as deep belief networks, autoencoder, etc. Zhang et al. (2016); Lei et al. (2019); Rostami et al. (2019).

The previous studies used unsupervised methods by transform or arithmetic operation or unsupervised machine learning. Recently many works use supervised change detection methods (Peng, Zhang, and Guan 2019; Sherrie et al., 2020; Zhang et al. 2020; Chen et al. 2022). The supervised methods have many advantages. First, using a training process based on massive labeled data helps to create a robust model. In particular in the case of an imbalance problem, which is the case in detecting changes in satellite images; the number of changed pixels is very small compared to unchanged pixels. Also, in case of detecting changes in fine image details and complex texture features in high-resolution images Zhang et al. (2020). In convolution neural networks, supervision

CONTACT Rasha Alshehhi ra130@nyu.edu Center for Space Science, New York University, Abu Dhabi, United Arab Emirates

improves the learning ability to extract multi-scale features from input raw images based on labeled image samples (Peng, Zhang, and Guan 2019; Zhang et al. 2020; Kaiyu et al., 2020). In addition, it introduces change-detection loss in intermediate layers (Zhang et al. 2020). Second, supervised learning produces good model performance with higher evaluation scores (specificity, sensitivity, precision, etc.) (Goswami et al. 2022).

Nowadays, supervised deep networks are applied using two approaches. First, single network architecture is used to extract multi-scale features based on arithmetic operations between two bi-temporal images (Daudt, Bertr, et al., 2018). Second, two parallel network architectures are used to extract multi-scale features from each image (Daudt, Bertr, et al., 2018; Chen et al. 2020; Zhang et al. 2020). In this paper, we use a supervised patch-based deep method. The method has three parts; two parts to extract features of two sequence image patches and one part to differentiate between change and unchanged image patches in high-resolution images. This paper is summarized as follows:

- It uses end-to-end architecture: encoder to extract multi-scale features from two sequence images and decoder to differentiate between learned features.
- It integrates feature maps from the same convolutional layer into dual attention maps (DAM) that concentrate on the spatial and channel difference of combined feature maps.
- It uses the Dice Coefficient as an error function between multi-scale predicted probability change maps and multi-scale reference change maps.

The rest of this paper is organized as follows. Section 2 presents some of the related works. Section 3 describes the proposed method. Section 4 shows the experimental results and important findings. Section 5 summarizes this paper.

Related work and problem definition

There are three deep approaches for detecting the changes from satellite images: early fusion, late fusion and the combination of early and late fusion. Fully convolutional-early fusion (FCEF) is one of the benchmark deep change-detection methods that fuses early the difference between bi-temporal images. It shares low-level features using skip-connection but fails to provide details of individual raw images. Mainly, the output change-detection maps have irregular object boundaries and lower object compactness (Daudt, Bertr, et al., 2018).

Caye Daudt et al. (2018) proposed fully convolutional Siamese-concatenation (FCSC) and fully convolutional Siamese-difference (FCSD) that solve the weakness of the previous method. Firstly, both methods apply a Siamese encoding stream to extract deep features from bi-temporal images and then combine the extracted deep features on the decoding stream to produce a change detection map. The difference is that the FCSD is based on the difference between in-depth features from the encoding stream. In contrast, the FCSC depends on the concatenation of in-depth features from the encoding stream. The back-propagation is performed from feature-discrimination/difference layers (decoder) to feature-extraction layers (encoder).

Chen et al. (2020) proposed a deep Siamese multi-scale convolutional network (DSMSCN) architecture using multi-scale feature convolution units (MFCU) layers to extract multi-scale spatial and spectral features from raw images before the feature-difference stage. These methods may produce uninformative features and poor image qualities. To fix the problem, many studies concatenate on raw image features and image difference features; however, the main concern is how to effectively combine features.

Zhang et al. (2020), one of the latest studies, used the image difference feature (IDF), dual attention module (DAM), spatial attention module (SAM) and channel attention module (CAM) to integrate the raw-image feature (encoder stream) into the feature-difference (decoder stream).

In this work, we extract multi-scale features from convolutional layers (encoder) into feature-difference layer (decoder) to acquire better change maps with accurate structures capturing variations in pixel-level (e.g. intensity) to region-level (e.g. shape and texture) to object-level. We also use a dual attention module based on spatial and channel modules. We add difference-feature and difference-image to improve the output from the feature-difference stage.

Method

The network architecture and loss functions are presented in Section 3.1 and Section 3.2.

Network architecture

The network architecture has three branches I, II and III, as shown in Figure 1. The branch I presents network architecture (encoder I) of the first input image X_{T_0} (dimensions = $W \times H \times C$) at time T_0; where W, H and C are width, height and number of channels. The branch II presents network architecture (encoder II) of the second image X_{T_1} ($W \times H \times C$) at time T_1. Both branches present a feature-extraction stage (Figure 1-a) from X_{T_0} and X_{T_1}. The convolutional feature maps of both branches I and II are fed to branch III. Branch III presents network architecture (decoder). The first input of branch III is feature maps of the latest convolutional layers of branches I and II. The

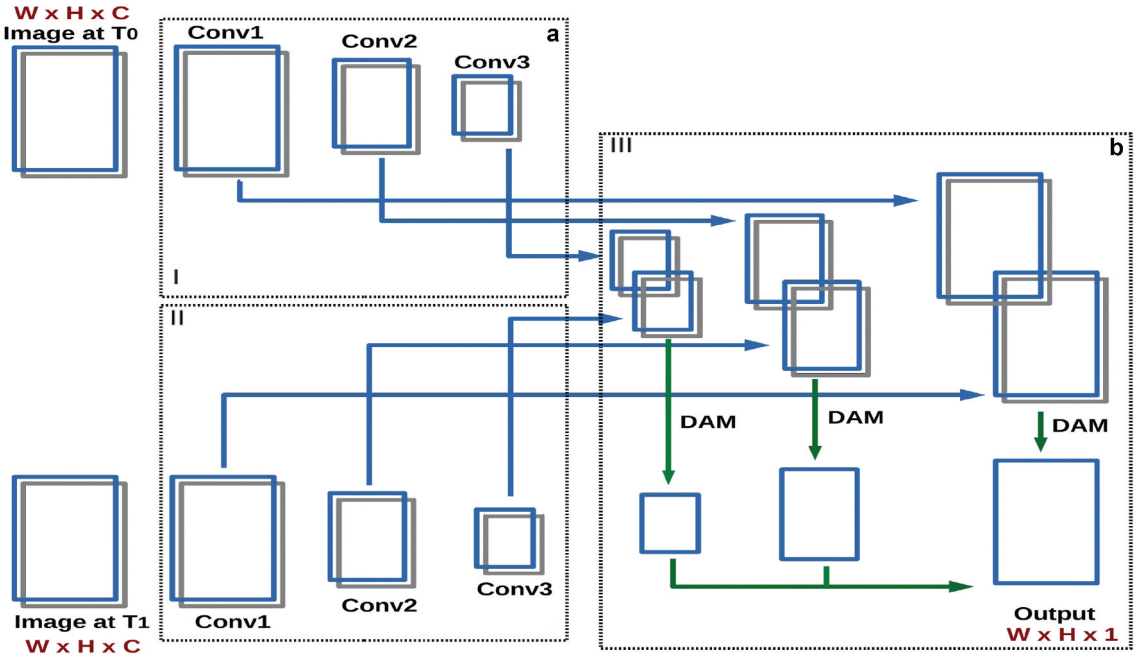

Figure 1. An overview of network architecture: (a) feature-extraction and (b) difference-extraction. *W*, *H* and *C* are width, height and number of channels, respectively.

branch I and II present the feature-extraction stage (Figure 1-a) and branch III presents the feature-difference stage (Figure 1-b). The feature maps from the same convolutional layers of branches I and II are combined into one dual attention module (DAM) to produce multi-outputs in branch III. Each DAM consists of a spatial attention module (SAM) and a channel attention module (CAM) (Jun et al., 2019). The final binary change-map is a result of aggregating all DAM outputs after up-sampling to the scale of input images X_{T_0} and X_{T_1}. In Section 3.1.1, 3.1.2 and 3.1.3, we will illustrate all processes to produce SAM and CAM and then DAM outputs.

Spatial Attention Module (SAM)

We use the SAM to increase the distance between changed and unchanged pixels in difference-maps of feature convolution maps of the branch I and II (Figure 2). The input map into the SAM is M_{conv} ($W \times H \times C$); which is a combination of three random difference-maps between feature-maps of same convolutional layers (M_{conv,T_0} and M_{conv,T_1}). To produce SAM map M_{sam}, the M_{conv} is fed in pooling, summation and multiplication operations. First, it is fed into maximum-pooling and average-pooling operations to produce M_{max} ($W \times H \times 1$) and M_{avg} ($W \times H \times 1$). Second, both avg-matrix and max-matrix are wise-element summed into M_i ($W \times H \times 2$). Third, the sum-matrix is fed into

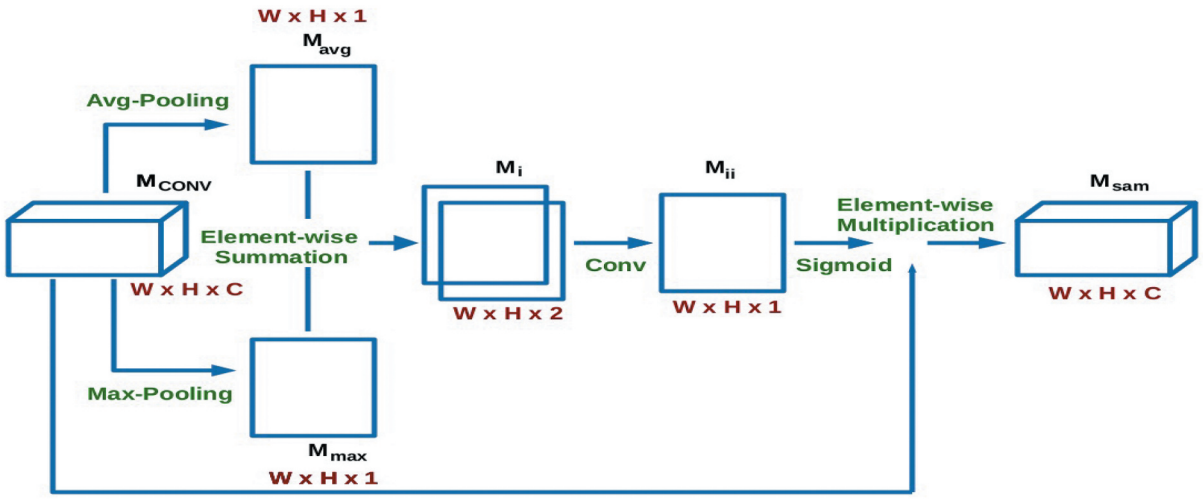

Figure 2. An overview of spatial attention modules (SAM). *W*, *H* and *C* are the width, height and number of maps from the branch I and II, respectively.

a convolution operation along the channel axis to produce M_{ii} $(W \times H \times 1)$ (Eq. 1); which is activated using the sigmoid function to generate M_{sam}. Finally, the SAM output is M_{sam} (Eq. 2) is element-wise multiplied with M_{conv} to refine feature maps, where changed pixels are emphasized by multiplying with higher weights while unchanged pixels are suppressed by multiplying with lower weights.

$$M_{ii} = \mathbb{C}(M_{avg} \oplus M_{max}), \tag{1}$$

$$M_{sam} = \sigma(M_{ii}) \otimes M_{conv}, \tag{2}$$

where $\mathbb{C}$ is a convolution operation with filter size 5×5, $\oplus$ is an element-wise summation process, σ is a sigmoid function and $\otimes$ is a element-wise multiplication process.

Channel Attention Module (CAM)

We use the CAM to emphasize target-relevant channels while suppressing target-irrelevant channels (Figure 3). The input into the CAM is the same input into SAM M_{conv}. To produce CAM output M_{cam}, the M_{conv} is fed in reshape,transpose, summation and multiplication operations as follows. First, M_{conv} is reshaped and transposed to produce M_i $(N \times C)$ and M_{ii} $(C \times N)$; where $N = W \times H$. The reshaped and transposed matrices are multiplied to generate M_c $(C \times C)$ after applying softmax activation function (Eq. 3). The M_c measures the impact of each channel of M_i on each channel of M_{ii}. The weaker the connection between two channels, the smaller values of matrix M_c. Second, the M_c is multiplied with the M_i to produce M_{cc} $(W \times H \times C)$ (Eq.4). Finally, the CAM output M_{cam} $(W \times H \times C)$. The M_{cam} is produced by element-wise summed the CAM input M_{conv} with M_{cc} (Eq. 5).

$$M_c = \frac{exp(M_i \otimes M_{ii})}{\sum exp(M_i \otimes M_{ii})}, \tag{3}$$

$$M_{cc} = M_c \otimes M_i, \tag{4}$$

$$M_{cam} = M_{cc} \oplus M_{conv}, \tag{5}$$

where $\otimes$ is a matrix multiplication operation and $\oplus$ is an element-wise sum operation.

Dual Attention Module (DAM)

The DAM output is a result of element-wise sum operation of M_{sam} and M_{cam} after multiplication with the last difference-maps of feature convolution layers $(M_{conv,F-1})$ of branches I and II, where F is number of feature maps in one convolutional layer.

$$M_{dam} = (M_{sam} \oplus M_{cam}) \otimes M_{conv,M-1}, \tag{6}$$

The final binary change-map $\hat{Y}$ (Eq. 7) is a result of maximum operation of all M_{dam} element-wise multiplied by difference-image between input images X_{T_0} and X_{T_1} of branch I and II.

$$\begin{aligned} \hat{Y} &= \sigma(\mathbb{C}(Max(M_{dam}) \otimes M_{D(T_0,T_1)})) \\ &= \sigma(\mathbb{C}(\mathbb{M}(M_{dam}) \otimes M_{D(T_0,T_1)})), \end{aligned} \tag{7}$$

where $\mathbb{C}$ and σ are 1×1 convolution and sigmoid functions.

Loss function

We use a multi-scale error function between the predicted probability change-map from each layer I of the branch III $\hat{Y}_l$ and its corresponding reference change-map Y_l of the same dimension. We use

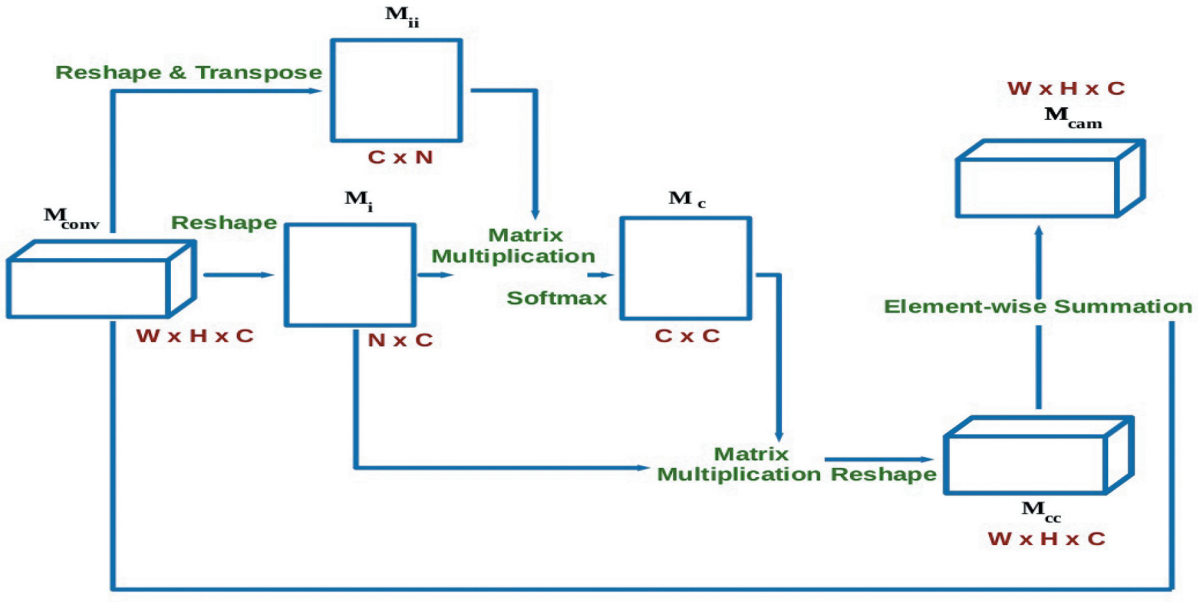

Figure 3. An overview of the channel attention module (CAM). *W*, *H* and *C* are the width, height and number of channels from both branches, respectively.

binary cross-entropy function (Eq. 8) and weighted dice-coefficient function (Eq. 9).

We use a binary cross-entropy function to measure error of each pixel in ground-truth change-map Y and its corresponding pixel in predicted probability change-map $\hat{Y}$ (Eq. 8).

$$E_C(Y,\hat{Y}) = \frac{1}{N}\sum_{n}^{N-1} y_n * log(\hat{y}_n) + (1-y_n) * log(1-\hat{y}_n), \tag{8}$$

where y represents the ground truth value of the pixel; $y = 1$ if the ground-truth pixel belongs to the changed class. Otherwise $y = 0$. $\hat{y}$ represents the predicted probability of the pixel belonging to the change class. $1-\hat{y}$ presents the probability of a pixel belonging to the unchanged class. N is the number of image samples.

We use the weighted dice-coefficient function because it is effective for the class-imbalance scenario, which is the case in the change-detection problem; the number of changed pixels is very small compared to the unchanged pixels:

$$E_D(Y,\hat{Y}) = \frac{1}{N*L}\sum_{n}^{N-1}\sum_{l=0}^{L-1}\left(1 - \frac{y_{n,l} \cap \hat{y}_{n,l}}{|y_{n,l}| + |\hat{y}_{n,l}|}\right), \tag{9}$$

where L is the number of layers in branch III. y and $\hat{y}$ present each pixel in ground-truth change-map and probability change-map.

The total loss of the network is a combination of two functions, as follows:

$$E(Y,\hat{Y}) = E_C(Y,\hat{Y}) + E_D(Y,\hat{Y}), \tag{10}$$

Performance

In this section, we describe the used satellite datasets in Section 4.1, evaluation metrics and training parameter settings in Section 4.2. We evaluate the network design and experimental results in Section 4.4 and Section 4.5.

Data

Air Change Detection-ACD

cThe ACD dataset consists of Szada and Tisza subsets (Benedek and Sziranyi 2009, 2008). In this paper, we use the Szada dataset to train the convolutional model. It comprises 42 pairs of optical aerial images acquired from different years of different seasonal conditions. Image pairs consist of red, green and blue bands with dimensions 952 × 640 and a spatial resolution of 1.5 meters per pixel. Histogram matching is applied to the two co-registered images for achieving color consistency. The annotated changes focus on changes in agriculture areas (new built-up regions, fresh plough-land and groundwork before building). The number of image pairs is relatively small and we divide images into 20,000 patches of size 256 × 256. We use 12,000, 3000 and 5000 pairs as training, validation and testing images, respectively. We use the proposed model and retrain it in the Tisza dataset (24 pairs). This dataset can be downloaded from http://web.eee.sztaki.hu/remotesensing/airchange_benchmark.html

SYSU-CD

The SYSU-CD dataset consists of 20,000 pairs of aerial images of dimensions 256 × 256 with a spatial resolution of 5 meters per pixel between the year 2007 to the year 2014 in Hong Kong, (Shi et al. 2021). We use 12,000, 3000 and 5000 pairs in training, validation and testing datasets, respectively. The major changes include newly built urban buildings, suburban dilation, groundwork before construction, change of vegetation, road expansion and sea construction. This dataset can be downloaded from https://github.com/liumency/SYSU-CD.

Onera Satellite Change Detection-OSCD

The OSCD dataset comprises 24 pairs of multi-spectral images from the Sentinel-2 satellites between 2015 and 2018 (Daudt, Bertr, et al., 2018). The pairs of multi-spectral images are picked worldwide, in Brazil, USA, Europe, Middle-East and Asia. Each image consists of 13 bands. Images vary in spatial resolution between 10 meters, 20 meters to 60 meters per pixel. We use 20,000 patches of size 256 × 256 (12000, 3000 and 5000 pairs as training, validation and testing images, respectively). The annotated changes focus on urban changes, such as new buildings or new roads. This dataset can be downloaded from https://rcdaudt.github.io/oscd/.

Evaluation metrics

To evaluate the performance, we measure precision (positive predictive value (PPV)) (Eq.11), recall or sensitivity (true positive rate (TPR)) (Eq. 12), specificity (true negative rate (TNR)) (Eq. 13) and F1-score (Eq. 14) after testing five times in all test sets, where TP, FN, FP and TN are the numbers of changed pixels correctly classified as changed pixels, the number of changed pixels classified as unchanged pixels, the number of unchanged pixels classified as changed pixels, and the number of unchanged classified correctly respectively Maxwell et al., (2021). We also evaluate the network design (attention modules and loss functions) based on the average of intersection-over-union IoU. The IoU is defined as an area of intersection of the predicted change map $\hat{Y}$ with the ground-truth map Y divided by the area of the union between $\hat{Y}$ and Y (Eq. 15).

$$PPV = \frac{TP}{TP + FP}, \tag{11}$$

$$TPR = \frac{TP}{TP + FN}, \tag{12}$$

$$TNR = \frac{TN}{TN + FP}, \tag{13}$$

$$F1 - score = \frac{2 \times TP}{2 \times TP + FN + FP}, \tag{14}$$

$$IoU == \frac{TP}{TP + FP + FN}, \tag{15}$$

We compare the proposed method with the state-of-the-art methods:

- FCEF Daudt, Le Saux, et al., 2018: the first step of this network is image fusion; two image pairs at T_0 and T_1 are concatenated as an input image into the Siamese network.
- FCSD Daudt, Bertr, et al., 2018: two parallel Siamese network streams are used to extract features from the input image at T_0 and input image at T_1 (encoder). The output maps of the second stream are subtracted from the output maps of the first stream to produce inputs to the third network stream (decoder). The output map of the third stream is the probability change map.
- FC-Siam-Con Daudt, Bertr, et al., 2018: similar to the FCSD, it uses two parallel Siamese network streams. However, the output maps of the second stream are summed into the output maps of the first stream to produce input maps for the third stream (decoder).
- DSMSCN Chen et al. (2020): the encoder is divided into two Siamese networks with multi-scale feature convolution units (MFCU). The decoder network uses the difference between convolutional layers in two encoder networks.
- NestNet2 Li, Li, and Fang (2020): It uses UNet++ and fully convolutional Siamese networks as encoder networks. The decoder network uses the channel attention module (CAM) to concentrate the multi-scale convolutional layers of two encoder networks.
- DSIFN Zhang et al. (2020): it uses two VGG16 networks as encoders. The decoder integrates the difference-maps of convolutional feature maps into multi-scale dual attention modules (DAM).

Training and parameter setting

The first step of detecting changes in satellite images is pre-processing stage including radiometric normalization (e.g. IRMAD (Canty and Nielsen 2008) and key point-based RRN (Moghimi et al. 2021, 2022)) and co-registration. The ACD and SYSU-CD image-pairs are already radiometric normalized with zero mean and unit variance (Chen et al. 2020). The OSCD image-pairs are radiometric normalized and co-registered using GEFolki toolbox Brigot et al. (2016); Daudt, Bertr, et al., 2018.

We use ResNet architecture for parallel branches I and II. We train the model with 5000 epochs and a batch size of 16. We use Adam optimizer (Kingma and Jimmy, 2015). The learning rate is initiated at 0.0001. It is multiplied by the learning rate decay, which is empirically set to 0.2, if loss stops decreasing after 10 epochs. We use Keras 2.2 with Tensorflow 1.9 with a high-performance computing (HPC) server with Nvidia Tesla V100 GPUs to run all experiments. For FCEF, FC-Siam-Diff, FC-Siam-Con, DSMSCN, NestNet2 and DSIFN architectures, we use same parameter settings used in Daudt, Bertr, et al., 2018; Daudt, Bertr, et al., 2018; Chen et al. (2020); Li, Li, and Fang (2020); Zhang et al. (2020).

Ablation study for attention modules and loss function

To verify the performance of the attention modules and loss functions, we conduct experiments with different settings in the SYSU-CD dataset, as shown in Tables 1, 2 and Table 3. We build various attention architectures with and without difference-map, and with and without multi-scale dice-coefficient error functions. We use TPR and TNR metrics because IoU may be biased.

Dual attention module maps with difference-images vs. loss functions

The network architecture based on the mean of DAM maps-wise produced with the difference-image ($\mathbb{M}(M_{dam}) \otimes M_{D(T_0,T_1)}$) improves the performance remarkably with approximately 3% IoU, 3% TPR and 6% TNR compared to the architecture with only last DAM map wise-produced with the difference-image ($M_{dam} \otimes M_{D(T_0,T_1)}$), as shown in Table 1.

Employing the multi-scale dice-coefficient function in addition to binary cross-entropy function ($E_C(Y, \hat{Y}) + E_{D,L}(Y, \hat{Y})$) (1st row) enhances the performance because it reduces the overlapping rate between hierarchical structures in change binary maps with around 3-8% IoU, 2-11% TPR and 3-12% TNR improvements compared to other error functions; binary cross-entropy error and dice-coefficient error of the last layer ($E_C(Y, \hat{Y}) + E_D(Y, \hat{Y})$) (2nd row), sum of the multi-scale dice-coefficient error ($E_{D,L}(Y, \hat{Y})$) (3rd row), dice-coefficient error of the last layer ($E_D(Y, \hat{Y})$) (4th row) and binary cross-entropy of the last layer ($E_C(Y, \hat{Y})$) (5th row).

Table 1. Dual attention module map M_{dam} and the mean of DAM maps $\mathbb{M}(M_{dam})$ wise-produced with difference-image $M_{D(T_0,T_1)}$ vs. loss functions; cross-entropy binary error $E_C(Y,\hat{Y})$, single-scale dice error $E_D(Y,\hat{Y})$, and multi-scale dice error $E_{D,L}(Y,\hat{Y})$. The highest score is shown in blue.

Attention module	Loss function	IoU (%)	TPR (%)	TNR (%)
$\mathbb{M}(M_{dam}) \otimes M_{D(T_0,T_1)}$	$E_C(Y,\hat{Y}) + E_{D,L}(Y,\hat{Y})$	74.87 ± 2.2	80.92 ± 2.3	82.81 ± 2.1
$\mathbb{M}(M_{dam}) \otimes M_{D(T_0,T_1)}$	$E_C(Y,\hat{Y}) + E_D(Y,\hat{Y})$	71.97 ± 2.7	78.21 ± 2.7	78.12 ± 2.5
$\mathbb{M}(M_{dam}) \otimes M_{D(T_0,T_1)}$	$E_{D,L}(Y,\hat{Y})$	70.21 ± 2.7	75.21 ± 2.7	75.21 ± 2.7
$\mathbb{M}(M_{dam}) \otimes M_{D(T_0,T_1)}$	$E_D(Y,\hat{Y})$	69.17 ± 2.7	72.21 ± 2.7	72.11 ± 2.5
$\mathbb{M}(M_{dam}) \otimes M_{D(T_0,T_1)}$	$E_C(Y,\hat{Y})$	67.87 ± 2.7	69.21 ± 2.7	70.31 ± 1.9
$M_{dam} \otimes M_{D(T_0,T_1)}$	$E_C(Y,\hat{Y}) + E_{D,L}(Y,\hat{Y})$	71.23 ± 2.5	77.11 ± 2.3	76.98 ± 1.9
$M_{dam} \otimes M_{D(T_0,T_1)}$	$E_C(Y,\hat{Y}) + E_D(Y,\hat{Y})$	68.21 ± 2.7	74.23 ± 2.5	73.21 ± 2.1
$M_{dam} \otimes M_{D(T_0,T_1)}$	$E_{D,L}(Y,\hat{Y})$	67.91 ± 2.7	72.31 ± 2.1	72.31 ± 3.3
$M_{dam} \otimes M_{D(T_0,T_1)}$	$E_D(Y,\hat{Y})$	63.25 ± 3.2	$68.2.6 \pm 1.9$	68.33 ± 2.0
$M_{dam} \otimes M_{D(T_0,T_1)}$	$E_C(Y,\hat{Y})$	64.61 ± 3.1	67.31 ± 2.8	69.12 ± 2.7

Table 2. Dual attention module map M_{dam} and the mean of DAM maps $\mathbb{M}(M_{dam})$ vs. loss functions: cross-entropy binary error $E_C(Y,\hat{Y})$, single-scale dice error $E_D(Y,\hat{Y})$ and multi-scale dice error $E_{D,L}(Y,\hat{Y})$. The highest score is shown in blue.

Attention module	Loss function	IoU (%)	TPR (%)	TNR (%)
$\mathbb{M}(M_{dam})$	$E_C(Y,\hat{Y}) + E_{D,L}(Y,\hat{Y})$	71.43 ± 2.2	72.11 ± 1.8	73.12 ± 2.7
$\mathbb{M}(M_{dam})$	$E_C(Y,\hat{Y}) + E_D(Y,\hat{Y})$	68.21 ± 2.8	69.33 ± 2.4	68.16 ± 2.8
$\mathbb{M}(M_{dam})$	$E_{D,L}(Y,\hat{Y})$	68.57 ± 3.1	67.34 ± 2.6	69.11 ± 1.7
$\mathbb{M}(M_{dam})$	$E_D(Y,\hat{Y})$	66.12 ± 2.9	66.11 ± 2.0	66.12 ± 3.1
$\mathbb{M}(M_{dam})$	$E_C(Y,\hat{Y})$	64.38 ± 3.0	66.37 ± 2.3	65.32 ± 3.1
M_{dam}	$E_C(Y,\hat{Y}) + E_{D,L}(Y,\hat{Y})$	68.23 ± 2.6	70.21 ± 3.0	72.21 ± 2.4
M_{dam}	$E_C(Y,\hat{Y}) + E_D(Y,\hat{Y})$	66.12 ± 3.2	69.17 ± 2.5	71.23 ± 3.3
M_{dam}	$E_{D,L}(Y,\hat{Y})$	65.23 ± 3.4	67.27 ± 2.9	69.42 ± 3.2
M_{dam}	$E_D(Y,\hat{Y})$	66.22 ± 3.3	67.32 ± 2.8	66.12 ± 1.9
M_{dam}	$E_C(Y,\hat{Y})$	64.92 ± 3.5	65.11 ± 3.4	67.33 ± 3.3

Table 3. Channel attention module map M_{cam} (Zhang et al. 2020) and (M_{idf} (Zhang et al. 2020) $\otimes$ M_{cam}) $\otimes$ M_{sam} (Zhang et al. 2020) vs. cross-entropy $E_C(Y,\hat{Y})$ and dice single-scale $E_D(Y,\hat{Y})$ and dice multi-scale $E_{D,L}(Y,\hat{Y})$ functions. The highest score is shown in blue.

Attention module	Loss function	IoU (%)	TPR (%)	TNR (%)
M_{CAM}	$E_C(Y,\hat{Y}) + E_{D,L}(Y,\hat{Y})$	70.56 ± 2.2	73.62 ± 3.2	71.83 ± 2.6
M_{CAM}	$E_C(Y,\hat{Y}) + E_D(Y,\hat{Y})$	68.43 ± 2.7	70.86 ± 2.4	70.33 ± 3.0
$(M_{idf} \otimes M_{cam}) \otimes M_{sam}$	$E_C(Y,\hat{Y}) + E_{D,L}(Y,\hat{Y})$	72.64 ± 2.5	75.21 ± 2.7	73.33 ± 2.8
$(M_{idf} \otimes M_{cam}) \otimes M_{sam}$	$E_C(Y,\hat{Y}) + E_D(Y,\hat{Y})$	69.57 ± 2.8	71.31 ± 2.2	72.73 ± 2.6

Dual attention module maps vs. loss functions

Using the network architecture based on the mean of all DAM maps ($\mathbb{M}(M_{dam})$, as shown in Table 2), has higher accuracy scores compared to only using DAM map of the last layer and difference-image ($M_{dam} \otimes M_{D(T_0,T_1)}$, as shown Table 1); maximum-difference scores 3% IOU, 6% TPR, 5% TNR. In addition, the mean of multi-scale DAM maps ($\mathbb{M}(M_{dam})$) enhances accuracy scores compared to the final DAM map (M_{dam}) at most by 3% IoU, 2% TPR and 3% TNR (Table 2).

It is also worth mentioning using the sum of binary cross-entropy and the multi-scale dice-coefficient functions ($E_C(Y,\hat{Y}) + E_{D,L}(Y,\hat{Y})$) brings great benefits to detect changes between two images when using M_{dam} or $\mathbb{M}(M_{dam})$ in Tables 1 and 2.

DSIFN attention maps vs. dice loss functions

We also verify the performance of the network by adding M_{cam}, M_{sam} and M_{idf}, cited in Zhang et al. (2020), instead of the used M_{cam}, with and without multi-scale dice-coefficient function. In Zhang et al. (2020), the raw image features M_{conv,T_0} and M_{conv_T1} with image difference feature M_{idf} of the previous layer (in decoder part) is an input to the CAM. To produce M_{cam}, an input is fed into multi-layer perception (MLP) operation after average-pooling and maximum-pooling of the input map. It is expected that the multi-scale dice-coefficient function (1st row and 3rd row) improves the IoU, TPR and TNR scores with 1-4% compared to single-scale function (2nd row and 4th row). Moreover, deriving M_{idf}; difference-map of two convolution maps from same layers at time T_0 and time T_1, wise-produced with used M_{cam} and then wise-produced with M_{sam} (3rd row and 4th row) yields higher scores than using the M_{cam} (Zhang et al. 2020) (1st row and 2nd row) with around 1-2% improvements.

Comparison between the proposed method and benchmark methods

We compare the results of the proposed method with the previous techniques reported in literature (Daudt, Bertr, et al., 2018; Daudt, Bertr, et al., 2018; Chen et al. 2020; Li, Li, and Fang 2020; Zhang et al. 2020) based on visual interpretation and quantitative assessment in ACD, SYSU-CD and Onera datasets. For quantitative assessment, we used precision, recall and F1-score.

ACD-Szada

The ACD-Szada dataset mainly consists of open-area images, which usually are easier to identify the differences between them. Figure 4 shows RGB images at T_0 and T_1 and binary change maps after applying benchmark methods; where the *TP* pixels (changed pixels are classified correctly), *FN* pixels (changed pixels are classified as background pixels) and *FP* pixels (unchanged pixels classified incorrectly) are shown in yellow, red and green, respectively. All benchmark

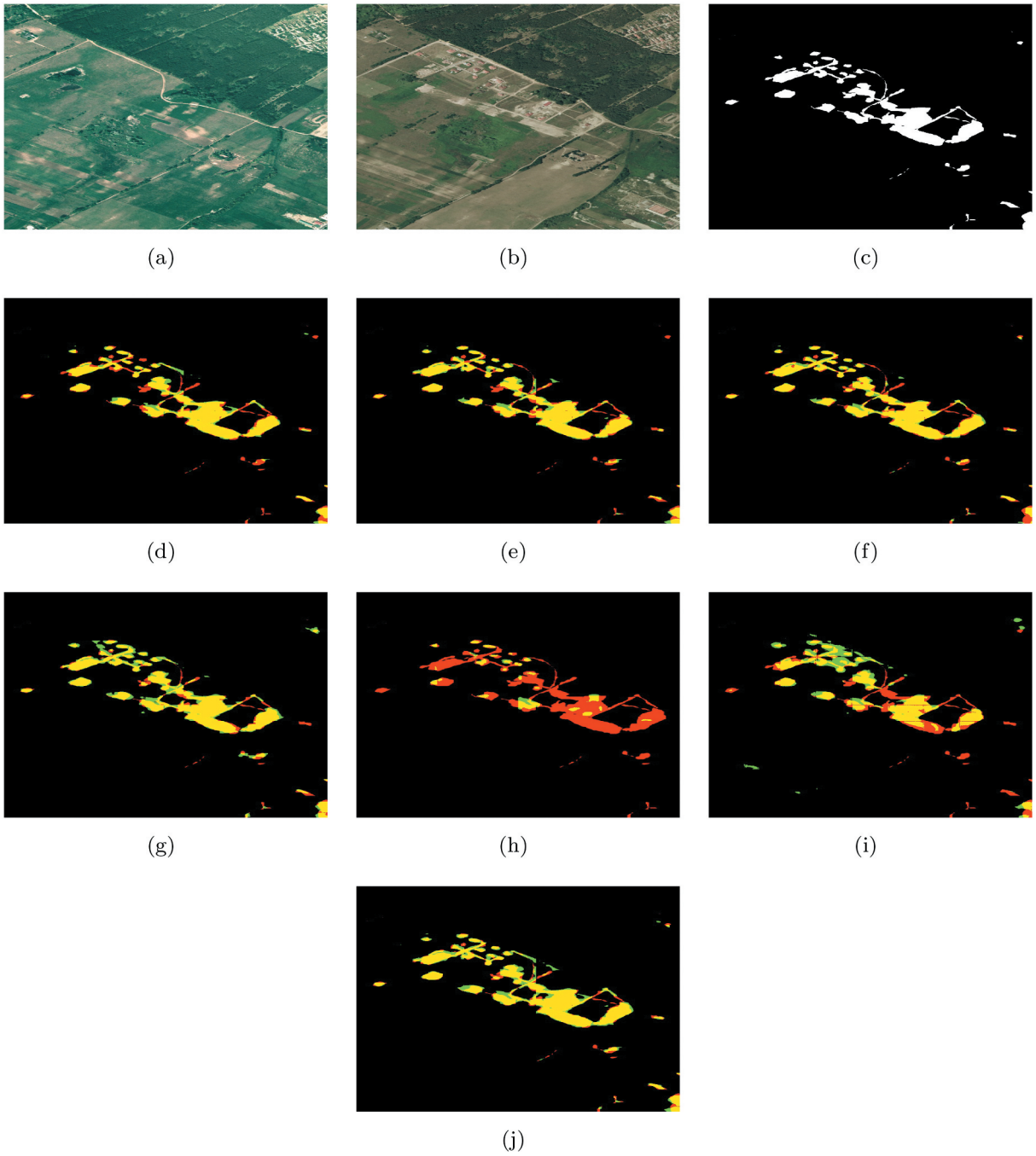

Figure 4. Comparison between the proposed method and benchmark methods in the Szada dataset: (a) image at T_0, (b) image at T_1, (c) ground-truth image (changed and unchanged pixels are depicted in white and black, respectively), (d) FCEF, (e) FCSD, (f) FCSC, (g) DSMSCN, (h) NestNet2, (i) DSIFN and (j) the proposed method (true positives are depicted in yellow, missed changes in red and false positives in green).

methods succeed in identifying general changed areas in the Szada images (Figure 4). However, the NestNet2 (Figure 4-g) mainly misses all multi-scale structures. The DSIFN (Figure 4-h) classifies changed pixels incorrectly, producing inaccurate properties of many changed structures, although it employs multi-scale attention modules. In Figure 4-i, the proposed method almost distinguishes changed small structures, including entire boundaries and continuous lines. This is because the change detection architecture depends on the hierarchical features from different convolutional layers that maintain low-level and high-level details. Moreover, error functions depend on the error function from various multi-scale convolutional layers. On the other hand, it does not succeed to retrieve completely changed structures because of the higher noise level.

Table 4 shows the quantitative assessment of the proposed method compared to the benchmark methods in the Szada dataset. The proposed method achieves the highest scores with a precision $64.57 \pm 2.2\%$, recall $74.88 \pm 1.8\%$ and F1-score $70.12 \pm 2.0\%$. The precision is relatively small because of the noise embedded between changed and unchanged pixels that would produce a high-value in difference-image and consequently the higher *FP* cases (green regions in Figure 4-i). Although the NestNet2 (2020 concentrates on the variations between the convolutional feature maps, it has low true-positive cases and consequently the lowest precision, recall and F1 scores. Compared to CD methods which are dependent on the early/late fusion of convolutional feature maps (e.g. FCEF, FCSD and FCSC), the proposed method improves precision, recall and F1-score by around 11-14% PPV, 10-12% TPR and 18-20% because it depends on both difference-image ($M_{D_{T_0,T_1}}$) with attention module on the pixel-to-pixel level (M_{sam}) and on the channel-to-channel level (M_{cam}) from multi-scale convolution layers. Also, it uses a multi-scale dice coefficient loss function to better capture overlapping between changed and unchanged areas starting from smaller-scale to larger-scale structures. The DSIFN also uses the difference-image and attention modules from the multiple convolutional layers; however, it uses a combination of the binary cross-entropy error and dice coefficient error at the final layer in the feature-difference stage, unlike the proposed method which uses it based on different layers. Therefore, it has higher *FP* cases and consequently lower precision ($47.13 \pm 2.4\%$). Also, it is worth mentioning that adding a difference-image to change probability maps improves the recall scores (e.g. DSMSCN, DSIFN and the proposed method have $67.53 \pm 2.8\%$, $65.01 \pm 2.6\%$ and $74.88 \pm 1.8\%$, respectively). We also compare the average inference time for all testing images. Mainly, all methods have identical inference time; required to produce binary change maps, but the proposed method spends the second shortest time to predict the binary change map, with a lower standard deviation.

ACD-Tisza

We use the previous model and retrain in the Tisza dataset (fine-tuning), which consists of open-area images similar to the previous dataset. Figure 5 shows binary change maps after applying all benchmark methods. All methods mainly restore large-size structures (change in vegetation, new buildings). However, many methods suffer from incomplete detection of curved boundaries (Figure 5-c-f). The DSIFN (Figure 5-h) and the FCSC (Figure 5-e) restore additional structures which are not parts of changed regions because it is based on concentrated features yielding a high value in the binary change map. Although the DSIFN (Figure 4-h) has ideal performance in identifying small-scale structures, it does not pinpoint the main structures. On the other hand, the NestNet2 (Figure 5-g) misses all changes, similar to the previous results. A subjective visual comparison with other CD methods shows that the proposed method works the best in terms of boundary accuracy and the internal structure of the new buildings. It is essentially consistent with reference ground-truth images. It is also remarkable to notice that all methods classify some regions as changed pixels; however, they are unchanged regions in ground-truth images and changed regions in bi-temporal images (Figure 5-a-b). This could be interpreted as missing changes in manual reference images.

Table 5 shows the average precision, recall and F1-score in the Tisza images. Compared to the Szada dataset, all methods have higher scores because the Tisza dataset consists of large continuous structures that are changed from T_0 to T_1. It is worth noting that

Table 4. Comparison between the previous methods and the proposed method based on quantitative metrics in the Szada dataset. The best score and the worst score are presented in blue and red colors, respectively.

Network	PPV (%)	TPR (%)	F1-score (%)	Time (ms)
FCEF (Daudt, Bertr, et al., 2018)	43.57 ± 2.7	62.65 ± 2.3	51.40 ± 2.5	1.8 ± 1.0
FCSD (Daudt, Bertr, et al., 2018)	41.38 ± 2.8	63.38 ± 2.1	52.66 ± 2.5	2.1 ± 0.8
FCSC (Daudt, Bertr, et al., 2018)	40.93 ± 2.6	64.61 ± 2.7	50.41 ± 2.1	2.4 ± 0.9
DSMSCN (Chen et al. 2020)	48.35 ± 2.1	67.53 ± 2.8	56.35 ± 2.5	2.7 ± 0.7
NestNet2 (Li, Li, and Fang 2020)	31.12 ± 3.2	48.0 ± 2.8	41.12 ± 2.7	2.9 ± 1.7
DSIFN (Zhang et al. 2020)	47.13 ± 2.4	65.01 ± 2.6	57.21 ± 2.1	2.7 ± 1.1
Proposed	64.57 ± 2.2	74.88 ± 1.8	70.12 ± 2.0	1.9 ± 0.6

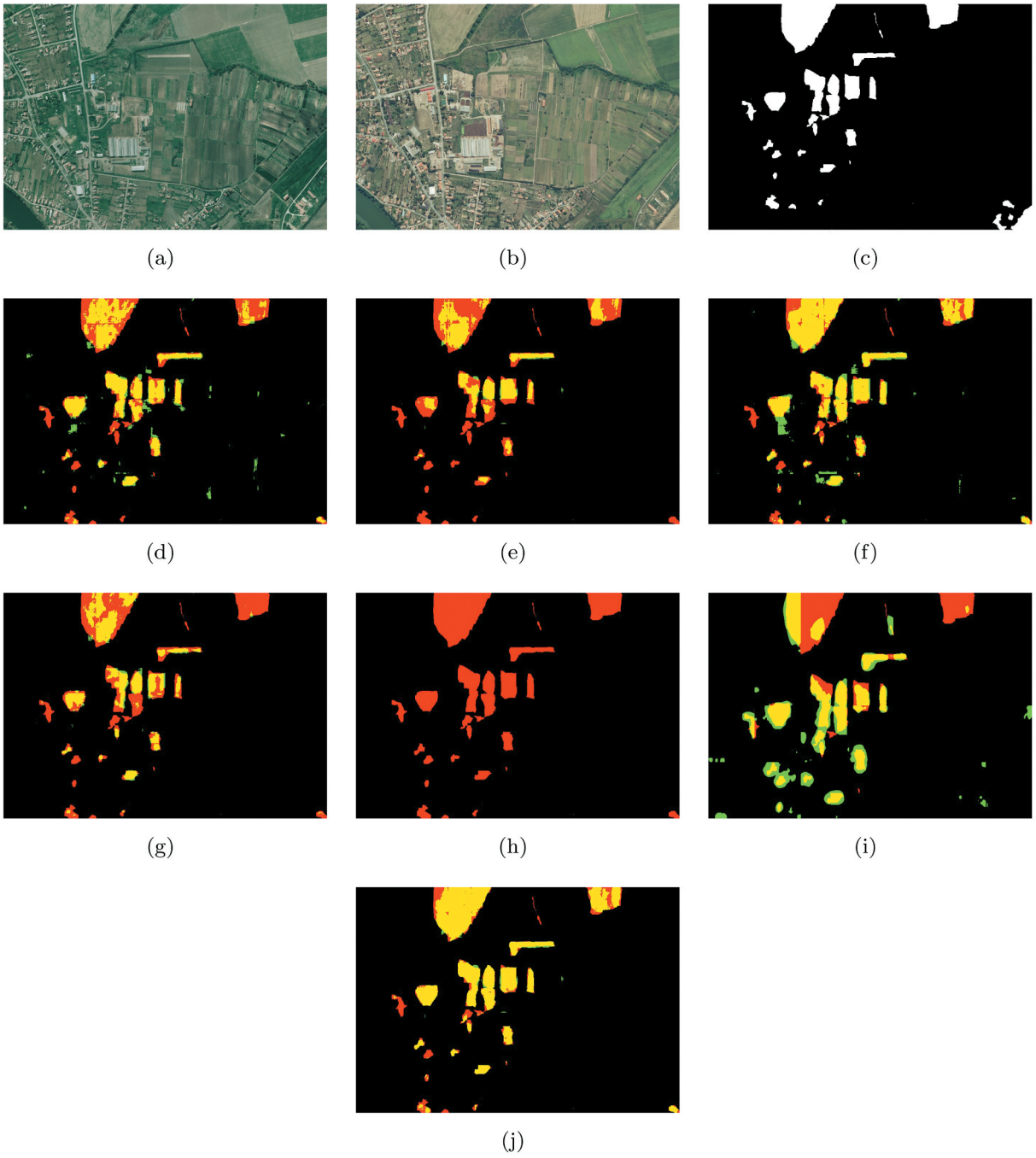

Figure 5. Comparison between the proposed method and benchmark methods in the Tisza dataset: (a) image at T_0, (b) image at T_1, (c) ground-truth image (changed and unchanged pixels are depicted in white and black, respectively), (d) FCEF, (e) FCSD, (f) FCSC, (g) DSMSCN, (h) NestNet2, (i) DSIFN and (j) the proposed method (true positives are depicted in yellow, missed changes in red and false positives in green).

Table 5. Comparison between the previous methods and the proposed method based on quantitative metrics in the tisza dataset. The best score and the worst score are presented in blue and red colors, respectively.

Network	PPV (%)	TPR (%)	F1-score (%)	Time(ms)
FCEF (Daudt, Bertr, et al., 2018)	86.28 ± 2.1	92.74 ± 1.5	85.40 ± 1.4	1.7 ± 1.0
FCSD (Daudt, Bertr, et al., 2018)	61.61 ± 2.1	82.29 ± 2.9	73.78 ± 2.8	2.1 ± 1.0
FCSC (Daudt, Bertr, et al., 2018)	68.07 ± 2.8	89.87 ± 2.6	79.65 ± 2.9	2.3 ± 1.2
DSMSCN (Chen et al. 2020)	82.91 ± 2.1	72.90 ± 2.8	82.90 ± 2.5	2.9 ± 0.9
NestNet2 (Li, Li, and Fang 2020)	49.21 ± 3.1	48.12 ± 2.9	47.31 ± 3.0	3.1 ± 1.2
DSIFN (Zhang et al. 2020)	71.71 ± 2.6	78.91 ± 2.4	68.32 ± 2.1	2.9 ± 1.0
Proposed	89.88 ± 1.5	88.21 ± 1.7	87.81 ± 1.9	1.6 ± 0.8

the FCEF has a higher score compared to other methods with precision, recall and F1-score equal to $86.28 \pm 2.1\%$, $92.74 \pm 1.5\%$ and $89.40 \pm 1.4\%$, respectively. This could be interpreted as the majority of changes in spectral features because the FCEF mainly extracts the differences on the first convolutional layers (low-level features). The proposed method has the second-highest recall score (around $88.21 \pm 1.7\%$) because it misses some changed small structures and consequently more *FN* cases (as shown in Figure 5-i)). Similar to the previous dataset, the inference time in the testing dataset shows that NestNet2 spends a long time to produce the binary change-map compared to other architectures and the proposed spends less than 2 ms.

SYSU-CD

We train the proposed method in the SYSU-CD dataset to produce binary change maps. As shown in Figure 6, all methods succeed in detecting many changed regions. However, all miss internal structures of the changed regions; because images at T_0 and T_1 have high noise levels (e.g. shadow, overlapping between trees and roads, or buildings and roads or buildings

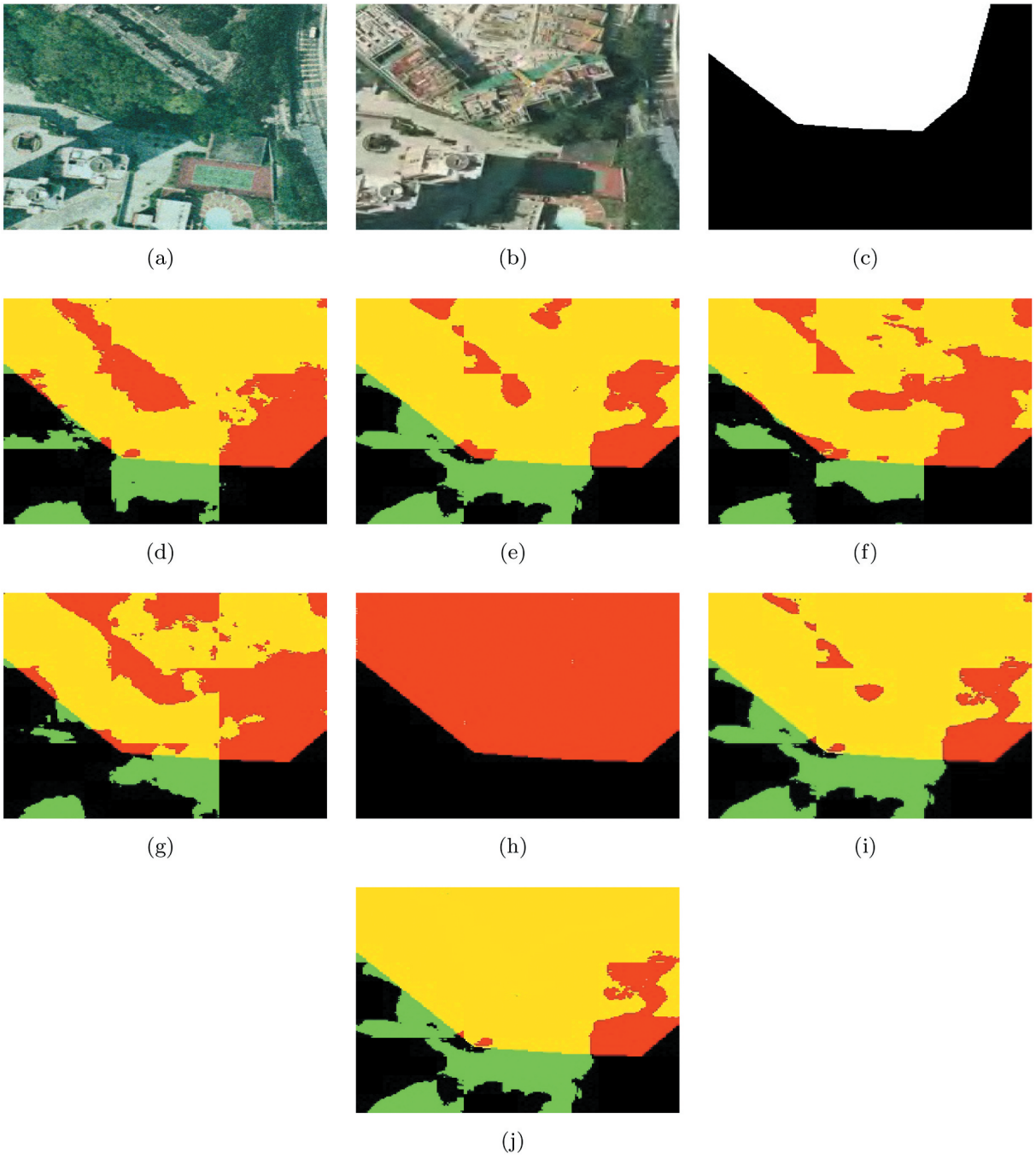

Figure 6. Comparison between the proposed method and benchmark methods in the SYSU-CD dataset: (a) image at T_0, (b) image at T_1, (c) ground-truth image (changed and unchanged pixels are depicted in white and black, respectively), (d) FCEF, (e) FCSD, (f) FCSC, (g) DSMSCN, (h) NestNet2, (i) DSIFN and (j) the proposed method (true positives are depicted in yellow, missed changes in red and false positives in green).

Table 6. Comparison between the previous methods and the proposed method based on quantitative metrics in the SYSU-CD dataset. The best score and the worst score are presented in blue and red colors, respectively.!

Network	PPV (%)	TPR (%)	F1-score (%)	Time (ms)
FCEF (Daudt, Bertr, et al., 2018)	70.32 ± 2.2	71.84 ± 1.8	71.07 ± 2.1	2.0 ± 0.2
FCSD (Daudt, Bertr, et al., 2018)	85.13 ± 1.7	57.21 ± 2.1	68.57 ± 2.3	2.1 ± 0.1
FCSC (Daudt, Bertr, et al., 2018)	78.54 ± 2.3	67.03 ± 2.7	72.35 ± 1.5	2.2 ± 0.3
DSMSCN (Chen et al. 2020)	70.81 ± 2.1	77.86 ± 2.3	74.18 ± 2.3	2.7 ± 0.3
NestNet2 (Li, Li, and Fang 2020)	61.16 ± 3.1	58.12 ± 2.9	61.11 ± 3.8	2.9 ± 0.7
DSIFN (Zhang et al. 2020)	71.11 ± 2.1	67.21 ± 1.9	69.31 ± 1.9	2.4 ± 0.2
Proposed	82.12 ± 1.8	83.90 ± 1.6	80.92 ± 2.3	1.7 ± 0.4

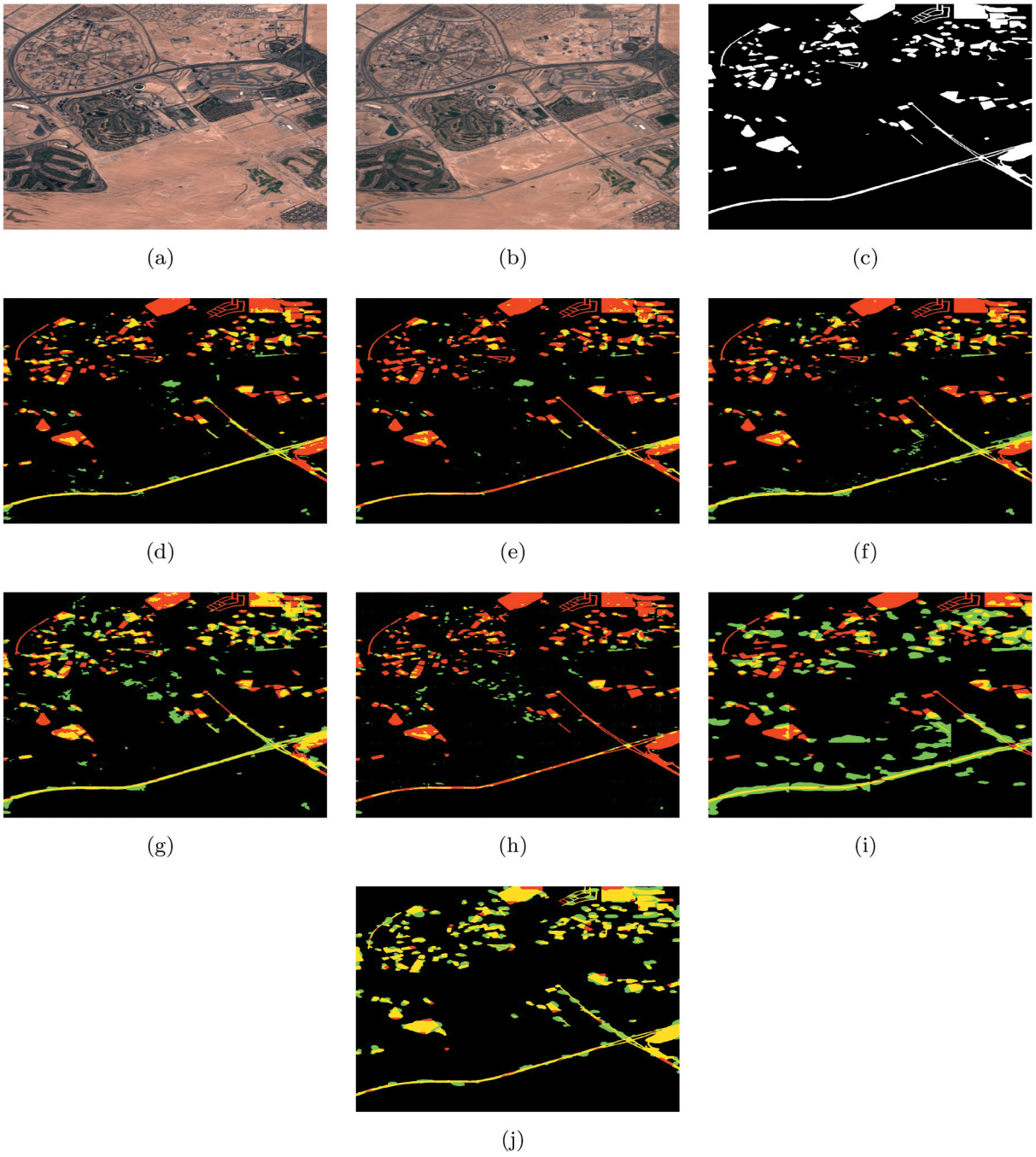

Figure 7. Comparison between the proposed method and benchmark methods in the onera dataset: (a) image at T_0, (b) image at T_1, (c) ground-truth image (changed and unchanged pixels are depicted in white and black, respectively), (d) FCEF, (e) FCSD, (f) FCSC, (g) DSMSCN, (h) NestNet2, (i) DSIFN and (j) the proposed method (true positives are depicted in yellow, missed changes in red and false positives in green).

and roads, etc.). In addition, reference ground-truth images do not identify many changed regions. The DSIFN (Figure 6-h) and the proposed method (Figure 6-i) are consistent with reference ground-truth images in retrieving overall structures. The NetNet2 (Figure 6-g) does not restore all structures. The remaining methods (Figure 6-c-f) identify some changed outlines of roads and buildings which are changed in Figure 6-b.

In Table 6, we compare all CD methods in the SYSU-CD dataset. All methods, excluding NestNet2, which employ difference-images or attention modules have high scores (above 70%) because reference ground-truth images consist of irregularly changed broad regions, which are easier to distinguish. The proposed method has the highest recall score because it uses the average of binary change maps from multiple layers and multi-scale dice-coefficient error in the feature-difference stage to better capture detailed information. It is worth noticing that the proposed method spends the shortest time predicting the binary change map.

Onera

We train the proposed method in the Onera dataset. Figure 7 presents changed areas in Dubai city from the Onera dataset. We expect that all methods, which are dependent on the late-feature fusion such as FCSD and FSDC, focus on learning contextual object-level features such as compacted buildings, continuous roads and complete boundaries. However, the FSCD (Figure 7-d) fails to retrieve some contextual shape features (e.g, outlines of roads); however, the FCEF (Figure 7-c) identifies these features. The FCEF change map (Figure 7-c) shows broken object boundaries and poor object internal compactness because low-level features of raw images can hardly be provided to help image reconstruction through skip-connections. Surprisingly, the FCSD (Figure 7-d) and the FSDC (Figure 7-e) also do not succeed in reconstructing large-scale to small-scale structures. The DSMSCN (Figure 7-f) and the DSIFN (Figure 7-h) retrieve uninterrupted lines and many compact buildings but miss many outlines. On the other hand, the NestNet2 (Figure 7-g) fails to recover many urban structures. The proposed method succeeds in distinguishing continuously changed lines from the entire region (e.g. roads). It shows complete composite structures (e.g. buildings). It also presents some small structures which are not shown in all previous maps.

In Figure 8, we show predicted change maps from small areas of Chongqing and Las Vegas cities from the Onera dataset. These examples demonstrate that the proposed method has good performance in distinguishing changes in multi-level structures starting from small, middle to large details (pixel, region to object-level) such as illumination variations to new whole building structures.

Table 7 shows precision, recall and F1-scores of binary change maps from the Onera dataset. When we use three bands (true color images), all methods do not succeed to retrieve changed areas and the proposed method achieves the best performance with precision, recall and F1-score equal to $50.21 \pm 2.0\%$ and $55.81 \pm 1.9\%$ and $52.12 \pm 2.3\%$, respectively. One of the future directions is to optimize the proposed method to adapt it with multi-spectral images.

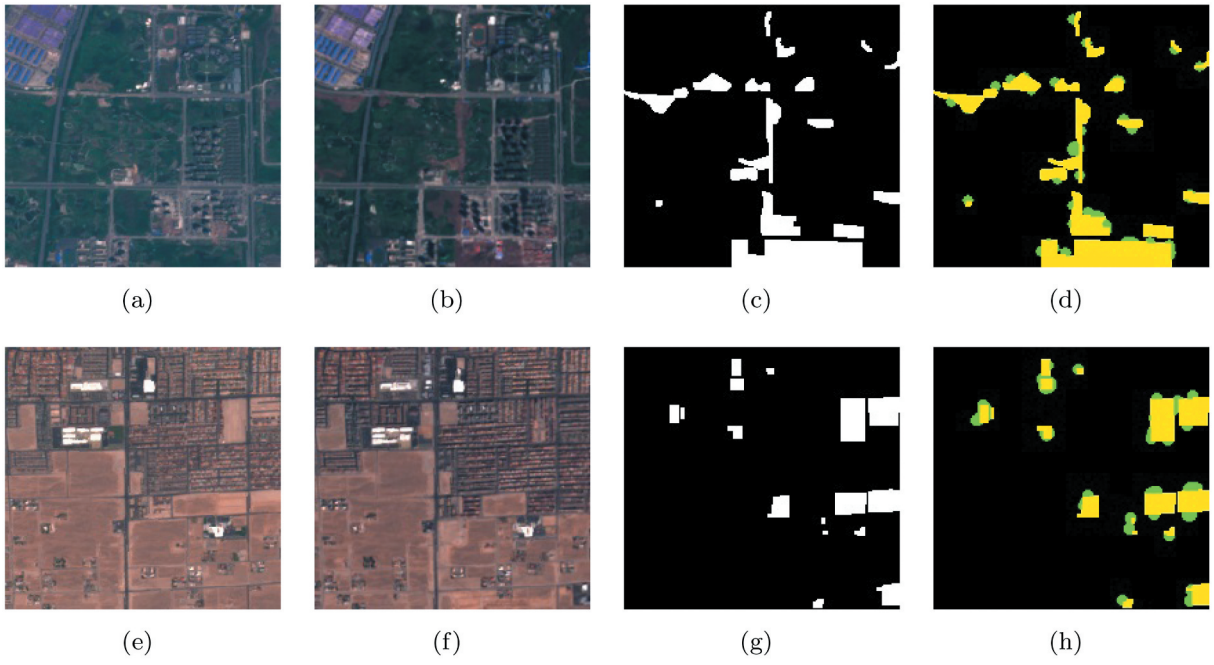

Figure 8. Building detection on from chongqing and Las Vegas cities: (a) and (e) input image at T0, (b) and (f) input image at T1, (c) and (g) ground-truth images (changed and unchanged pixels are depicted in white and black, respectively), and (d) and (h) change maps (true positives are depicted in yellow, missed changes in red and false positives in green).

Table 7. Comparison between the proposed method and previous methods based on quantitative metrics in the Onera dataset. The highest score and the lowest score are presented in blue and red colors, respectively.

Networks	PPV(%)	TPR (%)	F1-score (%)
FCEF (Daudt, Bertr, et al., 2018)	44.72 ± 3.1	53.92 ± 3.2	48.89 ± 3.3
FCSD (Daudt, Bertr, et al., 2018)	49.81 ± 3.3	47.94 ± 2.8	48.86 ± 3.2
FCSC (Daudt, Bertr, et al., 2018)	42.89 ± 2.8	47.77 ± 2.6	45.20 ± 2.8
DSMSCN (Chen et al. 2020)	48.91 ± 3.2	49.12 ± 3.0	49.32 ± 3.4
NestNet2 (Li, Li, and Fang 2020)	40.12 ± 5.2	48.17 ± 5.2	42.11 ± 4.3
DSIFN (Zhang et al. 2020)	42.21 ± 3.2	43.21 ± 3.3	44.12 ± 3.1
Proposed	52.21 ± 2.0	55.81 ± 1.9	53.12 ± 2.3

Conclusion and future directions

Change detection is one of the main problems in remote sensing applications. In this paper, we propose a multi-scale change-detection method. The network architecture consists of three branches: I, II (feature-extraction) and II (feature-difference) to distinguish changed multi-scale structures from unchanged ones in bi-temporal images at time T_0 and time T_1. We extract the bi-temporal image features separately (encoder) and then feed them together (decoder) with the different images at T_0 and T_1. We also utilize a multi-scale dice coefficient error function to decrease overlapping between changed and background pixels. We train the model in the ACD, SYSU-CD and Onera datasets. Based on the experiential analysis, we prove that the proposed attention method has a good accuracy score compared to benchmark methods, it successfully identifies changed multi-scale structures with the highest precision, recall and F1 scores in various datasets.

We use datasets that consist of open-area images. One of the future directions is to build an attention module to concentrate on dense areas. We also use RGB images to evaluate the proposed method. We aim to adapt the proposed method to more complicated environments (e.g. multi-spectral or hyperspectral images, images from various sensors). Compared to the state-of-the-art techniques, the proposed method achieves good performance in various datasets. In the future, we intend to use transfer-learning methods to transfer the change-detection method from one dataset to another dataset. We will use domain adaptation with minimum loss function between multi-scale features from two convolutional networks of the first image at T_0 and the second image at T_1 based on similarity metrics such as maximum mean discrepancy (MMD), central moment discrepancy (CMD) and correlation alignment.

Acknowledgment

This material is based upon work supported by Tamkeen under the New York University, Abu Dhabi Research Institute grant G1502.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported by the New York University Abu Dhabi Institute Grant (ADH01-73-71210-G1502-ADHPG) and Emirati Research Grant (ADH01-76-71202-EMIRP-ADHPG)

Data availability statement

The dataset supporting this article is available in the http://web.eee.sztaki.hu/remotesensing/airchange_benchmark.html, https://drive.google.com/drive/folders/1ALb8rzw9zEMSxwNTvIrIaA83zjjs04CE and https://rcdaudt.github.io/oscd/.

References

Benedek, C., & Sziranyi, T. 2008. "A mixed markov model for change detection in aerial photos with large time differences." In *International Conference on Pattern Recognition* (pp. 1-4), Florida, USA.

Benedek, C., & Sziranyi, T. (2009). Change detection in optical aerial images by a multilayer conditional mixed markov model. *IEEE Transactions on Geoscience and Remote Sensing*, *47*(10), 3416–3430. https://doi.org/10.1109/TGRS.2009.2022633

Brigot, G., Colin-Koeniguer, E., Plyer, A., & Janez, F. (2016). Adaptation and evaluation of an optical flow method applied to coregistration of forest remote sensing images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *9*(7), 2923–2939. https://doi.org/10.1109/JSTARS.2016.2578362

Canty, M. J., & Nielsen, A. A. (2008). Automatic radiometric normalization of multitemporal satellite imagery with the iteratively re-weighted MAD transformation. *Remote Sensing of Environment*, *112*(3), 1025–1036. https://doi.org/10.1016/j.rse.2007.07.013

Celik, T. (2009). Unsupervised change detection in satellite images using principal component analysis and k-means clustering. *IEEE Geoscience and Remote Sensing Letters*, *6*(4), 772–776. https://doi.org/10.1109/LGRS.2009.2025059

Chen, H., Chen, W., Du, B., & Zhang, L. 2019. "Deep siamese multi-scale convolutional network for change detection in multi-temporal VHR Images." In *International Workshop on the Analysis of Multitemporal Remote Sensing Images* (pp.1-4), Shanghai, China.

Chen, H., Chen, W., Du, B., & Zhang, L. (2020). Change detection in multi-temporal vhr images based on deep siamese multi-scale convolutional networks. *arxiv.org/abs/1906.11479*.

Chen, P., Zhang, B., Hong, D., Chen, Z., Yang, X., & Baipeng, L. (2022). FCCDN: Feature constraint network for VHR image change detection. *Isprs Journal of Photogrammetry and Remote Sensing*, *187*, 101–119. https://doi.org/10.1016/j.isprsjprs.2022.02.021

Daudt, R., Bertr, L., & Boulch, A. 2018. "Fully convolutional siamese networks for change detection." In *IEEE International Conference on Image Processing*, 4063–4067.

Daudt, R., Le Saux, B., Boulch, A., & Gousseau, Y. 2018. "Urban Change Detection for Multispectral Earth Observation Using Convolutional Neural Networks." In *IEEE International Geoscience and Remote Sensing Symposium* (pp. 2115-2118), Valencia, Spain.

Dengkui, M., Lin, H., Jiping, L., Sun, H., Zhang, Z., & Xiong, Y. 2008. "A SVM-Based change detection method from bi-temporal remote sensing images in forest area." In *International Workshop on Knowledge Discovery and Data Mining* (pp. 209-212), Adelaide, Australia.

Goswami, A., Sharma, D., Mathuku, H., Machinathu Parambil Gangadharan, S., Shekhar Yadav, C., Kumar Sahu, S., Kumar Pradhan, M., Singh, J., & Imran, H. (2022). Change detection in remote sensing image data comparing algebraic and machine learning methods. *Electronics*, *11*(3), 431. https://doi.org/10.3390/electronics11030431

Hussain, M., Chen, D., Cheng, A., Wei, H., & Stanley, D. (2013). Change detection from remotely sensed images: from pixel-based to object-based approaches. *Isprs Journal of Photogrammetry and Remote Sensing*, *80*, 91–106. https://doi.org/10.1016/j.isprsjprs.2013.03.006

Jun, F., Liu, J., Tian, H., Yong, L., Bao, Y., Fang, Z., & Hanqing, L. 2019. "Dual attention network for scene segmentation." In *Conference on Computer Vision and Pattern Recognition* (pp. 3146-3154), California, USA.

Kaiyu, L., Zhe, L., & Fang, S. 2020. "Siamese nestedunet networks for change detection of high resolution satellite image." In *International Conference on Control, Robotics and Intelligent System* (pp. 42-48), Xiamen, China.

Kingma, D. P., & Jimmy, B. 2015. "Adam:A method for stochastic optimization." In *International Conference on Learning Representations*.

Lei, Y., Liu, X., Shi, J., Lei, C., & Wang, J. (2019). Multiscale superpixel segmentation with deep features for change detection. *IEEE Access*, *7*, 36600–36616. https://doi.org/10.1109/ACCESS.2019.2902613

Liu, S., Marinelli, D., Bruzzone, L., & Bovolo, F. (2019). A review of change detection in multitemporal hyperspectral images: current techniques, applications, and challenges. *IEEE Geoscience and Remote Sensing Magazine*, *7*(2), 140–158. https://doi.org/10.1109/MGRS.2019.2898520

Luppino, L. T., Bianchi, F. M., Moser, G., & Anfinsen, S. N. 2018. "Remote sensing image regression for heterogeneous change detection." In *International Workshop on Machine Learning for Signal Processing* (pp 1-6), AALBORG, DENMARK.

Maxwell, A. E., Warner, T. A., & Andrés Guillén, L. (2021). Accuracy assessment in convolutional neural network-based deep learning remote sensing studies—Part 1: Literature review. *Remote Sensing*, *13*(13), 2450. https://doi.org/10.3390/rs13132450

Moghimi, A., Celik, T., Mohammadzadeh, A., & Kusetogullari, H. (2021). Comparison of keypoint detectors and descriptors for relative radiometric normalization of bitemporal remote sensing images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *14*, 4063–4073. https://doi.org/10.1109/JSTARS.2021.3069919

Moghimi, A., Sarmadian, A., Mohammadzadeh, A., Celik, T., Amani, M., & Kusetogullari, H. (2022). Distortion robust relative radiometric normalization of multitemporal and multisensor remote sensing images using image features. *IEEE Transactions on Geoscience and Remote Sensing*, *60*, 1–20. https://doi.org/10.1109/TGRS.2021.3063151

Mohsenifar, A., Mohammadzadeh, A., Moghimi, A., & Salehi, B. (2021). A novel unsupervised forest change detection method based on the integration of a multiresolution singular value decomposition fusion and an edge-aware Markov Random Field algorithm. *International journal of remote sensing*, *42*(24), 9376–9404. https://doi.org/10.1080/01431161.2021.1995075

Parente, L., Taquary, E., Paula Silva, A., Souza, C., & Ferreira, L. (2019). Next generation mapping: Combining deep learning, cloud computing, and big remote sensing data. *Remote Sensing*, *11*(23), 2881. https://doi.org/10.3390/rs11232881

Peng, D., Zhang, Y., & Guan, H. (2019). End-to-End change detection for high resolution satellite images using improved UNet++. *Remote Sensing*, *11*(11), 1382. https://doi.org/10.3390/rs11111382

Rostami, M., Kolouri, S., Eaton, E., & Kim, K. (2019). Deep transfer learning for few-shot SAR image classification. *Remote Sensing*, *11*(11), 1374. https://doi.org/10.3390/rs11111374

Sherrie, W., Chen, W., Michael Xie, S., Azzari, G., & Lobell, D. B. (2020). Weakly supervised deep learning for segmentation of remote sensing imagery. *Remote Sensing*, *12*(2), 207. https://doi.org/10.3390/rs12020207

Shi, Q., Liu, M., Shengchen, L., Liu, X., Wang, F., & Zhang, L. 2021. "A deeply supervised attention metric-based network and an open aerial image dataset for remote sensing change detection." *IEEE Transactions on Geoscience and Remote Sensing*.

Zhang, H., Gong, M., Zhang, P., Linzhi, S., & Shi, J. (2016). Feature-level change detection using deep representation and feature change analysis for multispectral imagery. *IEEE Geoscience and Remote Sensing Letters*, *13*(11), 1666–1670. https://doi.org/10.1109/LGRS.2016.2601930

Zhang, C., Yue, P., Tapete, D., Jiang, L., Shangguan, B., Huang, L., & Liu, G. (2020). A deeply supervised image fusion network for change detection in high resolution bi-temporal remote sensing images. *Isprs Journal of Photogrammetry and Remote Sensing*, *166*, 183–200. https://doi.org/10.1016/j.isprsjprs.2020.06.003

Zhao, X., Yang, Y., Duan, F., Zhang, M., Jiang, G., Yan, X., Cao, S., & Zhao, W. (2022). Identification of construction and demolition waste based on change detection and deep learning. *International journal of remote sensing*, *43*(6), 2012–2028. https://doi.org/10.1080/01431161.2022.2054296