Article

Fully Convolutional Networks with Multiscale 3D Filters and Transfer Learning for Change Detection in High Spatial Resolution Satellite Images

Ahram Song [1] and **Jaewan Choi [2],***

[1]   Department of Civil and Environmental Engineering, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 08826, Korea; aram200@snu.ac.kr
[2]   School of Civil Engineering, Chungbuk National University, 1 Chungdae-ro, Seowon-gu, Cheongju, Chungbuk 28644, Korea
*   Correspondence: jaewanchoi@chungbuk.ac.kr

check for updates

Abstract: Remote sensing images having high spatial resolution are acquired, and large amounts of data are extracted from their region of interest. For processing these images, objects of various sizes, from very small neighborhoods to large regions composed of thousands of pixels, should be considered. To this end, this study proposes change detection method using transfer learning and recurrent fully convolutional networks with multiscale three-dimensional (3D) filters. The initial convolutional layer of the change detection network with multiscale 3D filters was designed to extract spatial and spectral features of materials having different sizes; the layer exploits pre-trained weights and biases of semantic segmentation network trained on an open benchmark dataset. The 3D filter sizes were defined in a specialized way to extract spatial and spectral information, and the optimal size of the filter was determined using highly accurate semantic segmentation results. To demonstrate the effectiveness of the proposed method, binary change detection was performed on images obtained from multi-temporal Korea multipurpose satellite-3A. Results revealed that the proposed method outperformed the traditional deep learning-based change detection methods and the change detection accuracy improved using multiscale 3D filters and transfer learning.

Keywords: multiscale three-dimensional filters; transfer learning; change detection; high spatial resolution satellite image; fully convolutional network; convolutional long short-term memory

1. Introduction

Change detection is a major research field in remote sensing; change detection methods are used for detecting the areas damaged by natural disasters [1–3]; monitoring vegetation [4–6]; as well as urban expansion [7–9] by analyzing spatial, spectral, and temporal changes in an area [10]. The wide availability of satellites and unmanned aerial vehicles worldwide and the improvements in sensor manufacturing technology have enabled acquiring images with a spatial resolution within 1 m and detecting regions of interest from high spatial resolution images. To use high spatial resolution satellite images for change detection, problems associated with spatial complexity, geometric inconsistency between images, and reflectance variability in each class must be considered [11–13].

Pixel- and object-based change detection methods are used for analyzing high spatial resolution satellite imagery [14]. Pixel-based change detection methods, such as image differencing [15], change vector analysis [16], and principal component analysis [17], detect changes based on the pixel, which is the basic unit of image analysis. Although these methods detect differences in detailed spectral characteristics at the pixel-level, the spatial context cannot be considered and the detection is easily

affected by noise [14,18]. Object-based change detection methods were developed to minimize the effects of georeferencing and high spectral variability [14,19,20], wherein the texture, shape, and spatial relationship of the image object are considered [21]. These methods involve the segmentation and extraction of features from high spatial resolution images, followed by the integration of each object. However, validating the results of object-based change detection methods remains a challenge and image segmentation suffers from under- or oversegmentation errors, often generating objects that are non-representative of actual features [22].

Data-driven approaches, such as deep learning, are recently being used for effective change detection in high spatial resolution images because they use computing equipment with sophisticated algorithms [14]. High level features are automatically extracted from the input data through multiple layers via deep learning. Deep learning-based change detection methods have the characteristics of pixel- and object-based methods because pixel-wise classification maps are predicted through semantic abstraction of the spatial context from the original data [23]. Convolutional neural network (CNN) is a popular network used for image classification and pattern recognition. Change detection methods based on CNN belong to two groups, with the extracting meaningful features from multi-temporal images using CNN, followed by comparing the feature maps or classification results to detect changes [23–25]. The second group transforms multi-temporal images to reflect changes, and the transformed data are used as input for CNNs [26–28]. Although CNN-based change detection methods effectively extract changes in multi-temporal images, such extraction is difficult without employing pre- or postprocessing techniques such as data transformation and post-classification because CNN cannot handle multi-temporal data within its structure.

Recurrent neural network (RNN) is an important part of deep learning and can handle temporal data through a recurrent hidden state with activation at each time step dependent on the past computation. RNN considers a current input and the output learned from the previous input. Change detection methods that use the long short-term memory (LSTM)-based RNN to deal with multi-temporal images can learn temporal features from sequential data [29–31]. Hybrid change detection networks were proposed to combine the advantages of the CNN and LSTM. Deep Siamese convolutional multiple layers with RNN were proposed for detecting changes in high spatial resolution images; this method could be used for detecting changes in homogeneous and heterogeneous images such as LiDAR intensity and optical satellite image [32]. Two-dimensional (2D) CNN for spatial–spectral feature extraction from input data and LSTMs for supporting sequence prediction have been utilized for change detection [33]. CNN extracts spatial–spectral–temporal features from multispectral and hyperspectral images. However, maintaining the spatial structure of the input image is difficult using 2D CNN-LSTM hybrid network, because all pixels are converted to one-dimensional (1D) vectors during the process. Three-dimensional (3D) fully convolutional network (FCN) and convolutional LSTM are combined to detect changes in images [34]. The FCN maintains the 2D structure of images and handles semantic segmentation [35]. Convolutional LSTM networks have replaced fully connected operators by convolutional operators for learning the spatiotemporal features. However, training data is insufficient for verifying the use of these networks for low-resolution hyperspectral images with a spatial resolution of 30 m.

Limited training samples and computing resources degrade the performance of deep learning networks. Labeled samples are usually limited in remote sensing images; therefore, building an efficient network and training with a small number of samples are challenging [36]. Transfer learning is a technique that involves the pre-training a deep learning model using a large, but different, dataset and adapting the trained model to specific problems with smaller image datasets [37]. In other words, given a source domain $\mathcal{D}_S$ and learning task $\mathcal{T}_S$ as well as a target domain $\mathcal{D}_T$ and learning task $\mathcal{T}_T$, transfer learning improves learning of the target predictive function $f_T(\cdot)$ in $\mathcal{D}_T$ using the knowledge of $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_S$ or $\mathcal{T}_S \neq \mathcal{T}_S$ [38]. A previous study attempted using pre-trained weights and biases from a semantic segmentation network for the initial convolutional layer of a hybrid change detection network [39]. The study confirmed that transfer learning can improve the performance

of the change detection on average and transfer learning using multispectral dataset and several benchmark hyperspectral datasets could solve the limitation of small sample problems in hyperspectral images processing.

However, this was limited by differences in the spatial and spectral resolution of the source and target domain datasets. For example, multispectral images with 0.5 m spatial resolution and 4 bands were used as a source domain dataset in the semantic segmentation network, whereas hyperspectral images with 30 m spatial resolution and 150 bands were employed as a target dataset in the change detection network. To improve the previous study, transfer learning was performed on a large dataset of aerial images to detect changes in multi-temporal high spatial resolution images using the recurrent FCN. The network is composed of 3D convolutional layers and convolutional LSTM. The 3D convolutional layers extract the spatial–spectral features of the input images, and the convolutional LSTM analyzes the temporal relationship between feature maps obtained from temporal images. Therefore, extracting meaningful feature maps by considering spatial–spectral features of input images can improve the results of change detection network. Herein, multiscale 3D filters were used in the initial convolutional layer of the change detection network; the layer exploits pre-trained weights and biases of semantic segmentation network trained on an open benchmark dataset. The main contributions of this study as follows.

1. Specialized 3D filters for spatial and spectral information were utilized to combine optimal multiscale filters considering the complexity of the calculation process and to prevent the redundancy of extracted information. Different surface materials can be detected using high spatial resolution satellite images; therefore, spatial and spectral filters of different sizes can be used to extract meaningful features, with the corresponding features maps improving the accuracy of the change detection.
2. We attempted to address the training data limitation using the proposed change detection method and the pre-trained information trained on high spatial resolution aerial images. The spatial and spectral resolutions of these images are similar to those of the satellite images used herein. Trained weights and biases can provide reasonable initial points of initial layer in the change detection network and prevent overfitting problems.
3. To confirm the effectiveness of the multiscale 3D filter and transfer learning for change detection in high spatial resolution satellite images, accuracies of other change detection methods based on deep learning and the proposed method with and without transfer learning were compared; then, the conditions for change detection were analyzed.

The remainder of this paper is organized as follows. Section 2 presents the architecture of the proposed method, and the datasets and environmental conditions for the experiments are described in Section 3. Sections 4 and 5 present the results and discussion, respectively, and Section 6 presents the conclusions.

2. Methods

The proposed change detection method primarily (i) trains the FCN for semantic segmentation using a large remote sensing dataset as the source domain, and (ii) performs transfer learning from the pre-trained FCN to the recurrent FCN for change detection. The FCNs for semantic segmentation and change detection includes the three multiscale 3D filters in the initial convolutional layer to extract various spatial and spectral features from high spatial resolution images. After the layer with 3D filters is trained on the source dataset, the pre-trained filters are transferred and fine-tuned on the target dataset.

2.1. Fully Convolutional Network (FCN) for Semantic Segmentation

Figure 1 shows the FCN architecture. First, the 3D FCN performing the semantic segmentation was trained on an aerial image dataset containing images obtained from the International Society for

Photogrammetry and Remote Sensing (ISPRS). The network was adapted for 3D convolutions with downsampled images followed by upsampled images with transpose convolutions for recovering the image dimensions. The 3D convolution is calculated as follows,

$$v_{l,j}^{x,y,z} = \phi\left(\sum_n \sum_{h=0}^{H-1} \sum_{w=0}^{W-1} \sum_{r=0}^{R-1} w_{ljn}^{hwr} o_{(l-1)n}^{(x+h)(y+w)(z+r)} + b\right), \tag{1}$$

where $v_{l,j}^{x,y,z}$ is the pixel value of position (x, y, z) on the jth feature map in layer l (the layer of the current operation), and H and W are the width and height of the kernel, respectively. The parameter R is the spectral dimension of the 3D kernel, w_{ljn}^{hwr} is the weight value at the position (h, w, r) connected to the nth feature in the $(l-1)$th layer, $o_{(l-1)n}^{(x+h)(y+w)(z+r)}$ represents the input at the position $(x + h)(y + w)(z + r)$ in (h, w, r) denoting its offset to (x, y, z), ϕ is the activation function, and the b is a bias parameter. The *ReLU*, which rectifies negative values to zero, is used as the activation function.
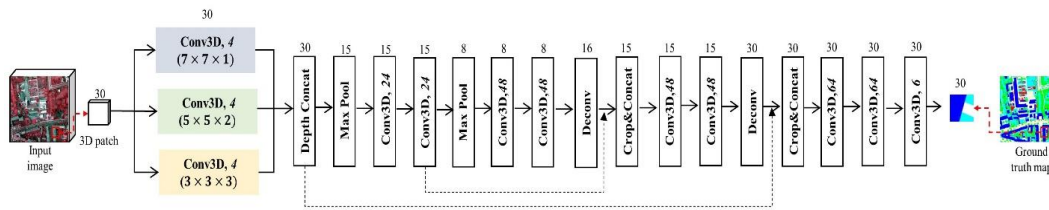

Figure 1. Illustration of a fully convolutional network containing multiple three-dimensional (3D) filters for semantic segmentation. "Conv3D", "Max pool", and "Deconv" represent the 3D convolutional layer, max pooling layer, and deconvolutional layer, respectively. The numbers on the boxes represent the pixel size of the input images and italic fonts mean the output number of layers. For example, "30" means the output size of the layers is 30 × 30 pixels and "6" represents the number of final feature maps is six.

Metric to sub-metric-level spatial resolution images reflect ground objects of different sizes varying from very small neighborhoods to large regions; therefore, multiscale filters were used to extract different features. These features were successively applied in the classification task [40–43]. Multiscale filters also allow observations from broad and micro perspectives [43]. Unlike previous study, herein, the spatial dimension and aspect of the spectral band of the filter are considered. Generally, features with smaller spatial scales such as edges of buildings and roads, respond to small convolutional filters, whereas coarse structures are extracted by large filters [41]. Furthermore, different spectral features are extracted depending on the number of adjacent bands included. As the input image contains red, green, blue (RGB), and near-infrared bands, the spectral information can be sued for identifying different materials. For example, materials with similar colors in the RGB bands are discriminated by considering the near-infrared band characteristics.

Therefore, multiscale 3D spatial and spectral filters with different sizes were used to extract meaningful features and improve the feature extraction robustness from the high spatial resolution satellite images. Initially, 3D convolutional layers were applied for parallel input of the image. The network uses multiple 3D filters in the first convolutional layer with different sizes. To confirm the effectiveness of the multiscale 3D filters in the semantic segmentation and determine appropriate shapes for the filters $(x \times y \times z)$, the classification accuracies at pixel level for different cases were compared using 3D filters, namely, $(1 \times 1 \times 4)$, $(3 \times 3 \times 3)$, $(5 \times 5 \times 2)$, and $(7 \times 7 \times 1)$. The size of the filter can be determined according to the size of the input image. The width and height of the 3D filter (x, y) were selected as 1, 3, 5, and 7 based on previous studies on the classification of satellite images using multiscale filters [40,42,43]. The size of spectral band z ranged from 1 to 4, covering the ISPRS dataset containing four bands (three visible bands and a near-infrared band). In particular, to combine optimal multiscale filters, considering the complexity of the calculation process and prevent

the redundancy of the extracted information, specialized 3D filters for spatial or spectral information were utilized. Therefore, the sizes of 3D filters were controlled, e.g., if the spatial dimension of the filters were large, the spectral dimension of the filters was set to be small.

The three feature maps obtained from three convolutional layers in the first layer were then combined to create a joint feature map. Feature maps from each convolutional filter have different sizes; therefore, they must be readjusted to match before creating the composite feature map. The features share all dimensions using padding except for the channels used, which may differ, and all feature maps are collected in a tensor.

The composite feature map was then used as the input for the sequential convolutional layers. The network mainly comprises nine convolutional layers with 3D filters, two pooling layers, and deconvolutional layers. For the downsampling, a filter size of 3 was used ($3 \times 3 \times 3$), with pooling followed by two sets of convolutions with size and stride of 2. This step generated the spatial size of the output map, with two convolution operations of identical output dimension followed by a pooling layer from a block of operations. Successive blocks reduced the spatial size, and many upsampling blocks were followed by downsampling blocks to recover the spatial size of the original image. Upsampling was achieved via transpose convolutions; after each transpose convolution, slicing of the output map occurred to match the size of the corresponding output map in the downsampling block, followed by the concatenation and convolution operations. This process was repeated until the original spatial size was recovered. The experiment was performed using Keras with TensorFlow as the backend, and the network was trained using the NVIDIA GeForce RTX 2070 GPU memory of 8 GB; the ISPRS multispectral datasets was used as the source data input. The size of the ISPRS image was too large; slices of shapes with labels were extracted, separated into batches and stored with the 3D FCN trained using 960,000 sub-images. In the experiment, the structure of the FCNs was identical to Figure 1 but the initial convolutional layer was different. When one filter was used, the initial layer comprised sequential two 3D convolutional layers. All patches obtained from the ISPRS dataset were used as training and test samples. The networks were trained with the Adam optimizer, which had a learning rate of 10^{-3}, batch size of 256, and 300 Epochs.

2.2. Recurrent FCN for Change Detection

The proposed change detection network combined 3D FCN and convolutional LSTM, wherein a fully connected layer at the end of the network was replaced with a convolutional layer. The 3D convolutional layers with multiscale 3D filters extracted spatial–spectral features from the input images, whereas the convolutional LSTM recorded and analyzed the change information of the multi-temporal image. The network was developed as a recurrent 3D FCN (Re3FCN) [34] by adding the transfer learning and multiscale 3D filters to apply for high spatial resolution satellite images. Figure 2 shows the architecture of the proposed change detection network.

Creating meaningful feature maps from multi-temporal images improves change detection accuracy because the change detection network detects changes based on the temporal information from the feature maps generated from temporal images. Therefore, the transfer learning resolves the problem of insufficient training samples using many remote sensing images as the source data. As the ultimate goal of transfer learning is to improve the change detection performance, the low-level features learned by deep networks from the source domain are transferred to the target domain. This provides excellent initial configurations in the transfer learning method to quickly initiate meaningful feature extraction from the multi-temporal high spatial resolution satellite images; proper initialization is crucial for network training [44]. The hypothesis is that the lowest layers of the FCN extract general features from the images; therefore, the learned weights are extended to other recognition tasks, as these mostly detect generic features. Concurrently, the topmost layers detect higher level features from the images, and therefore these are specific for the trained network's classification task. Thus, it is hypothesized that initializing a convolutional network with weights from a network pre-trained

on another dataset accelerates the training process, and improves performance because the low-level features are generic across different tasks.
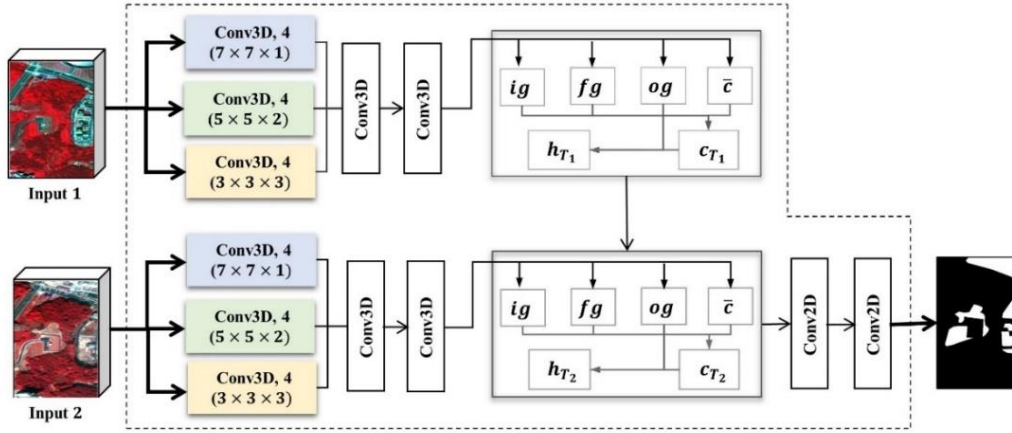

Figure 2. Architecture of the proposed change detection network, with "Conv2D" representing the 2D convolutional layer. "*ig*", "*fg*", "*og*", "$\bar{c}$","*c*", and "*h*" represent input gate, forget gate, output gate, candidate memory cell, memory cell, and hidden state, respectively. The colored layers exploit the pre-trained weights and biases from semantic segmentation network.

Considering I^{T_1} and I^{T_2} as the images obtained at times T_1 and T_2, respectively, the 3D patches obtained from each image are passed through the 3D convolutional layers with different 3D filters in parallel. Multiscale 3D filters were used to create different feature maps with identical 3D filters employed in the classification network. The weights and biases of initial layer with multiscale filters were later fine-tuned, with two more 3D convolutional layers included after the multiscale 3D convolutional layers. This is because small patches were used as input, which naturally reduced the network depth, and the predicted classes are relatively simple (change and no change) compared to other classification tasks. For example, the ImageNet classification involves 1000 categories, whereas the PASCAL VOC classification shows 20 classes [33]. A simple network is suitable for detecting changes in high spatial resolution images.

The spectral–spatial feature maps obtained from 3D convolutional layers were fed into the convolutional LSTM layer. In this phase, the temporal information between two images was reflected. Let f^{T_1} and f^{T_2} be the spectral–spatial feature maps obtained from I^{T_1} and I^{T_2}, respectively. The RNN architecture recollects values over arbitrary intervals using a memory cell c_t at a time step t. The convolutional LSTM involves three gates, namely, the input gate ig, output gate og, and forget gate fg, each of which has a learnable weight. fg_t is the gate for forgetting the previous information, and the output range of the sigmoid function is 0–1. If $\sigma = 0$, the previous state information is forgotten; if $\sigma = 1$, the previous state information is memorized. ig_t is the gate for remembering the current information, and the cell states are regulated by deleting or adding information through the gates. These gates are expressed as follows.

$$fg_t = \sigma\left(W_{hfg} * h_{t-1} + W_{ffg} * f^{T_t} + b_{fg}\right) \tag{2}$$

$$ig_t = \sigma\left(W_{hig} * h_{t-1} + W_{fig} * f^{T_t} + b_{ig}\right) \tag{3}$$

$$og_t = \sigma\left(W_{hog} * h_{t-1} + W_{fog} * f^{T_t} + b_{og}\right) \tag{4}$$

$$\overline{c_t} = \tanh\left(W_{h\bar{c}} * h_{t-1} + W_{f\bar{c}} * f^{T_t} + b_{\bar{c}}\right) \tag{5}$$

$$c_t = fg_t \odot c_{t-1} + ig_t \odot \overline{c_t} \tag{6}$$

$$h_t = og_t \odot tanh(c_t) \tag{7}$$

The subscripts associated with the weight matrix W have specific meaning. For example, W_{hfg} and b_{fg} are the weight matrices between h_{t-1} and fg and the bias of fg, respectively. $\overline{c_t}$ is the candidate cell value for constructing a new candidate value with ig_t, which is then added to the memory cell c_t to influence the next state. Finally, the output h_t is determined by multiplying $\tanh(c_t)$ and og_t; "$*$" is the convolutional operator and "$\odot$" is the element-wise multiplication. The three gates of the convolutional LSTM are represented by 3D tensors, and the convolutional LSTM determines the future state of a cell in the pixel-based on the input and past state of its adjacent region using a convolutional operator [45]. The outputs of the convolutional LSTM layer were then fed into the prediction layers to generate a score map, and the number of final feature maps and classes is equal. Finally, the pixels were classified into final classes according to the score map.

2.3. Quality Evaluation

To evaluate the utility of the proposed change detection method, the overall accuracy, Kappa coefficients, and F1 scores by class were calculated. The overall accuracy is the number of correctly classified pixels divided by the total number of sampled pixels. It is described as true positive (TP), true negative (TN), false negative (FN), or false positive (FP), with its calculation expressed in Equation (8). The F1 score considers the precision (Equation (9)) and recall (Equation (10)) in computing scores. Precision is the number of correct positive results divided by the number of all positive results returned by the classifier, and recall is the number of correct positive results divided by the number of all samples that should have been identified as positive. F1 score is expressed in Equation (11) as the harmonic mean of the precision and recall.

$$\text{overall accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{8}$$

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})} \tag{9}$$

$$\text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})} \tag{10}$$

$$\text{F1 score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precison} + \text{Recall}} \tag{11}$$

The Kappa coefficient measures the closeness of the classified images using the specific classifier with the ground truth map. For the Kappa coefficient, values greater than 0.8 imply a strong agreement between the classification result and ground truth, 0.6–0.8 indicates good accuracy, 0.4–0.6 indicates moderate accuracy, and <0.4 indicates poor accuracy [46]. The Kappa coefficient is defined as Equation (12), and it uses the overall accuracy and random accuracy (Equation (13)). Random accuracy is the sum of the products of reference likelihood and results likelihood for each class.

$$\text{Kappa coefficient} = \frac{\text{OA} - \text{RA}}{1 - \text{RA}} \tag{12}$$

$$\text{RA} = \frac{(\text{TN} + \text{FP}) \times (\text{TN} + \text{FN}) + (\text{FN} + \text{TP}) \times (\text{FP} + \text{TP})}{n^2} \tag{13}$$

where OA and RA represent overall accuracy and random accuracy, respectively. n is total number of samples.

Herein, the proposed network was compared with other methods, such as fully connected LSTM, 2D CNN-fully connected LSTM (2D CNN-LSTM), and Re3FCN combination composed of 3D convolutional layers and convolutional LSTM [34]. The LSTM deals with temporal information and is used to detect changes between two images [29]. The 2D CNN-LSTM involves the same structure of the paper of Mou et al. [33], and it comprises 2D convolutional layers and fully connected LSTM layers. The Re3FCN from a previous study was designed for extracting spatial and spectral features

from hyperspectral images. The difference between the Re3FCN and the proposed network is that the Re3FCN used a sequence of three convolutional layers, whereas the proposed network uses multiple convolutional layers and pre-trained values in the initial phase. To assess the effectiveness of the transfer learning, the change detection network with and without pre-trained weights and biases were compared. All three cases involved identical training and test samples and experimental parameters such as learning rate, Epochs, and optimizer type.

3. Datasets

3.1. The International Society for Photogrammetry and Remote Sensing Dataset

ISPRS 2D semantic labeling challenge Potsdam is an online open benchmark dataset [47] that provides high spatial resolution airborne images with a spatial resolution of 5 cm. The data contain near-infrared, red, blue, and green orthorectified imagery and corresponding digital surface models. Furthermore, the data include ground truth images that show the impervious surface, buildings, trees, low vegetation, cars, and unidentified features (Figure 3).
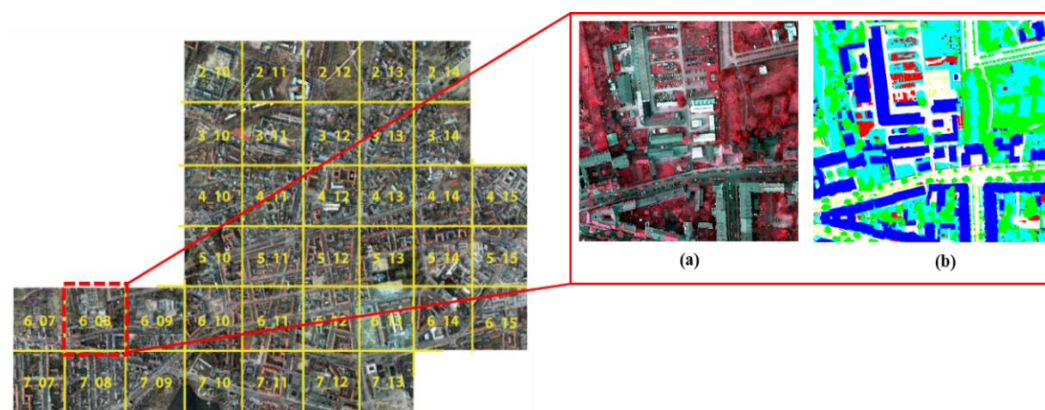

Figure 3. Example of the International Society for Photogrammetry and Remote Sensing (ISPRS) dataset with patch number of 6–8: (**a**) color-infrared image and (**b**) ground truth map. The classes involve impervious surface (white), buildings (blue), low vegetation (cyan), trees (green), and cars (yellow).

Although the dataset contains 38 patches, only 24 images with ground truth images were used for training and validation, and the patch numbers of the labeled data are presented in Table 1. Twenty-four large multispectral images containing $6000 \times 6000 \times 4$ pixels in tiff format were the initial data input sources. Because the size of the ISPRS images was too large, slices of $30 \times 30 \times 4$ pixels with labels (a total of 960,000 images) were extracted, separated into batches, and stored. The classification network was then trained using the sub-images.

Table 1. The patch numbers of labeled data utilized for the ISPRS dataset.

	Patch Numbers
Potsdam dataset	2_10, 2_11, 2_12, 3_10, 3_11, 3_12, 4_10, 4_11, 4_12, 5_10, 5_11, 5_12, 6_7, 6_8, 6_9 6_10, 6_11, 6_12, 7_7, 7_8, 7_9, 7_10, 7_11, 7_12

3.2. KOMPSAT 3A

This dataset of the Korea multipurpose satellite (KOMPSAT)-3A involves multi-temporal images obtained from Daejeon in South Korea (Figure 4). KOMPSAT-3A is Korea's first earth observation/infrared satellite with two imaging systems on board and was developed by the Korea Aerospace Research Institute (KARI) [48]. It provides high spatial resolution panchromatic and

multispectral imagery in the near-infrared, red, blue, and green bands. The spatial resolutions of the KOMPSAT-3A are 0.55 m (panchromatic image) and 2.20 m (multispectral images with five bands). The multi-temporal images were acquired in October 2015(T_1) and July 2018 (T_2), with vegetation distribution changes due to seasonal change and changes in urban areas attributed to new construction. To improve the spatial resolution of the KOMPSAT-3A images, a hybrid pan-sharpening method based on the normalized difference vegetation index (NDVI) [49] was applied during preprocessing. Locations of the images with improved spatial resolution are shown in Figure 4. Before the change detection, geometric correction was applied to the multi-temporal images using ground control points.
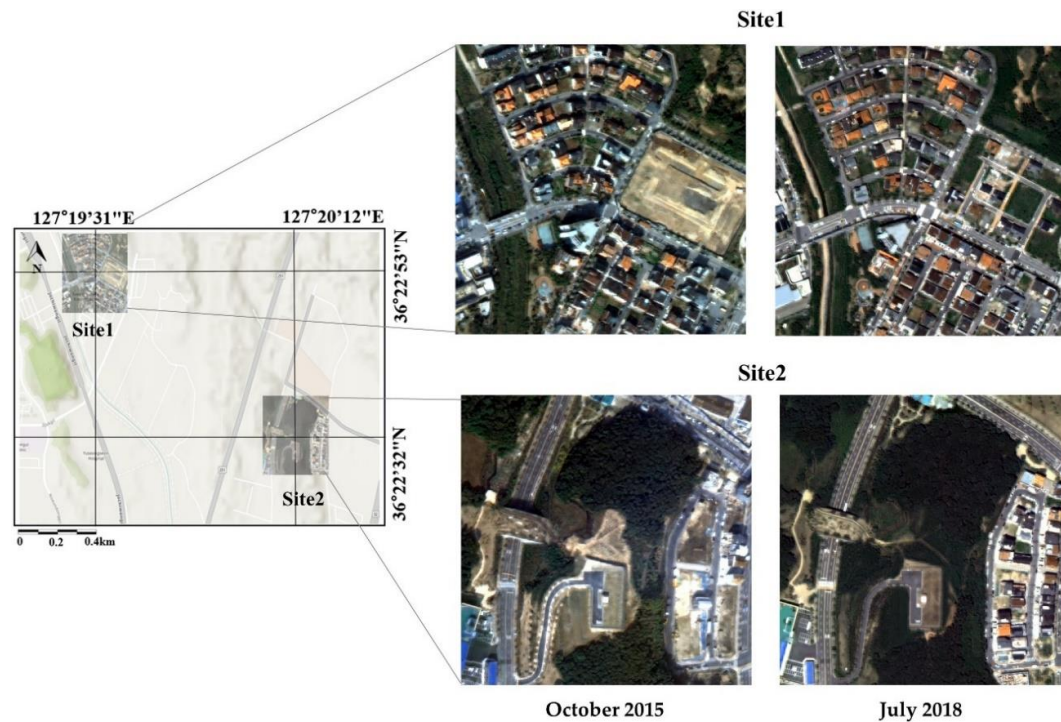

Figure 4. Locations of the two study sites; the background map was obtained from the ArcGIS world map [50]. The upper images are of site1 and the lower images are of site 2. The first images were obtained in October 2015, and the second images were obtained in July 2018. Both the images highlight the differences due to seasonal changes and newly constructed buildings.

Binary change detection distinguishes the pixels of the sites into changed (Ω_c) and unchanged (Ω_u) classes. Ground truth data were generated using web maps and KOMPSAT images with high spatial resolution. We defined changes involving land cover classes, from waterbodies to building areas. The land cover classes include vegetation, bare soil, buildings, waterbodies, and roads. Colored roofs, such as blue, brown, and white are classified as "buildings." The "bare soil" represents ground without buildings and vegetation, whereas "road" encompasses asphalt roadways. Changes due to relief displacement and shadows are not considered as changes in the ground truth data.

4. Results

4.1. Semantic Segmentation Results

The semantic segmentation results of the FCN for differently sized filters are presented in Table 2. The FCNs with ($1 \times 1 \times 4$) and ($7 \times 7 \times 1$) filters produced lower overall accuracy than ($3 \times 3 \times 3$) and ($5 \times 5 \times 2$) filters. In contrast, ($3 \times 3 \times 3$) and ($7 \times 7 \times 1$) filters have relatively higher F1 score than other 3D filters. In particular, the ($3 \times 3 \times 3$) filter shows the highest F1 scores for all classes and the ($7 \times 7 \times 1$) filter more effectively classifies the five classes—impervious surface, building, low vegetation, tree, and

car—than $(5 \times 5 \times 2)$ and $(1 \times 1 \times 4)$ filters. The $(1 \times 1 \times 4)$ filter addresses spectral correlation rather than spatial information, whereas the $(7 \times 7 \times 1)$ filter addresses local spatial correlation rather than spectral information. The filter that considers only the spectral information could not classify materials of five classes. Sematic segmentation results demonstrate that 3D filters, which consider spatial and spectral features, improve the classification results; further, spatial information significantly influences the classification of five classes than spectral information. Therefore, the $(3 \times 3 \times 3)$, $(5 \times 5 \times 2)$, and $(7 \times 7 \times 1)$ filters were selected to create multiscale 3D filters. When the multiscale 3D filters were used in the initial convolutional layer of the semantic segmentation network, the F1 scores and overall accuracy display remarkable improvements. For example, the F1 scores and OA displayed remarkable improvements. For example, the F1 score of the impervious surface, buildings, low vegetation, trees, and car increased by 0.0303, 0.0143, 0.0656, 0.049, and 0.104, respectively, compared with the highest existing values. Furthermore, the FCN with multiscale 3D filters delivered an improved OA value of 87.17%, which was 2.9% higher compared with when the $(3 \times 3 \times 3)$ filter was used.

Table 2. Classification results for validating the ISPRS Potsdam dataset using the fully convolutional network (FCN) with differently-sized 3D filters. "OA" represents overall accuracy.

Filter Shape	F1 Score					OA
	Impervious Surface	Building	Low Vegetation	Tree	Car	
$(1 \times 1 \times 4)$	0.7770	0.8306	0.5817	0.4703	0.4589	0.7532
$(3 \times 3 \times 3)$	0.8745	0.9088	0.6775	0.7370	0.6733	0.8427
$(5 \times 5 \times 2)$	0.8365	0.8696	0.6057	0.6040	0.6419	0.8134
$(7 \times 7 \times 1)$	0.8386	0.8611	0.6121	0.6263	0.6855	0.7842
$\begin{pmatrix} 3 \times 3 \times 3 \\ 5 \times 5 \times 2 \\ 7 \times 7 \times 1 \end{pmatrix}$	0.9048	0.9231	0.7431	0.7819	0.7895	0.8717

4.2. Change Detection Results

During the process, the Adam optimizer with a learning rate of 10^{-3} was used and the Epoch was set to 500. Training data were randomly generated from the ground truth data, and the number of training, validation, and test samples was 40,000, 20,000, and 30,000 pixels, respectively. *ReLU* served as the activation function of the convolutional layers, whereas *softmax* served as the activation function of the last convolutional layer. The final output of the change detection network could be classified into changed and unchanged classes.

Figures 5 and 6 display change detection maps generated using the proposed and other change detection methods for sites 1 and 2. The overall accuracy, Kappa coefficient, and F1 score for all classes from different methods are presented in Tables 3 and 4. The LSTM network shows the lowest overall accuracy, Kappa coefficient, and F1 score for sites 1 and 2 (for site 1, overall accuracy = 0.9136, Kappa coefficient = 0.6384, and F1 score = 0.6876, and for site 2, overall accuracy = 0.8826, Kappa coefficient = 0.5350, and F1 score = 0.6010). In site 1, LSTM classified the pixel changes from gray bare soil to green vegetation into unchanged classes (Figure 5b). In the same way, LSTM did not recognize the pixel changes from dark colored bare soil to building with brown roof. In contrast, the 2DCNN-LSTM and Re3FCN produced relatively higher accuracies than the LSTM and could classify the changed and unchanged pixels according to the training data. The accuracy of the 2D CNN-LSTM for site 1 is an overall accuracy of 0.9597, Kappa coefficient of 0.8443, and F1 score of 0.8680, and that for site 2 is overall accuracy of 0.9565, Kappa coefficient of 0.8518, and F1 score of 0.8783. In addition, the Re3FCN yielded an overall accuracy of 0.9674, Kappa coefficient of 0.8984, and F1 score of 0.8978 for site 1, and overall accuracy of 0.9633, Kappa coefficient of 0.8766, and F1 score of 0.8990 for site 2. However, many spot noises and errors are noted at the boundaries of buildings, road, and trees.

Because the proposed change detection method uses transfer learning, change detection was performed with and without transfer learning to assess the effectiveness of transfer learning. To briefly explain the method and avoid confusion, the proposed change detection method without transfer

learning is termed "multiscale Re3FCN without transfer learning" and the proposed change detection method with transfer learning is named "multiscale Re3FCN with transfer learning". The change detection method with the multiscale 3D filters outperformed other change detection methods for both study sites. The method without transfer learning produced an overall accuracy of 0.9717, Kappa coefficient of 0.8923, and F1 score of 0.9090 for site 1, and an overall accuracy of 0.9759, Kappa coefficient of 0.9158, and F1 score of 0.9304 for site 2. The multiscale Re3FCN with transfer learning showed the best results for all approaches. It produced an overall accuracy of 0.9790, Kappa coefficient of 0.9201, and F1 score of 0.9326 for site 1, and overall accuracy of 0.9795, Kappa coefficient of 0.9288, and F1 score of 0.9412 for site 2. The proposed change detection method could detect the pixels with the changes in class type although the colors appeared to be similar in RGB images. In addition, the spot noises were reduced and edges of changes were detected clearly.
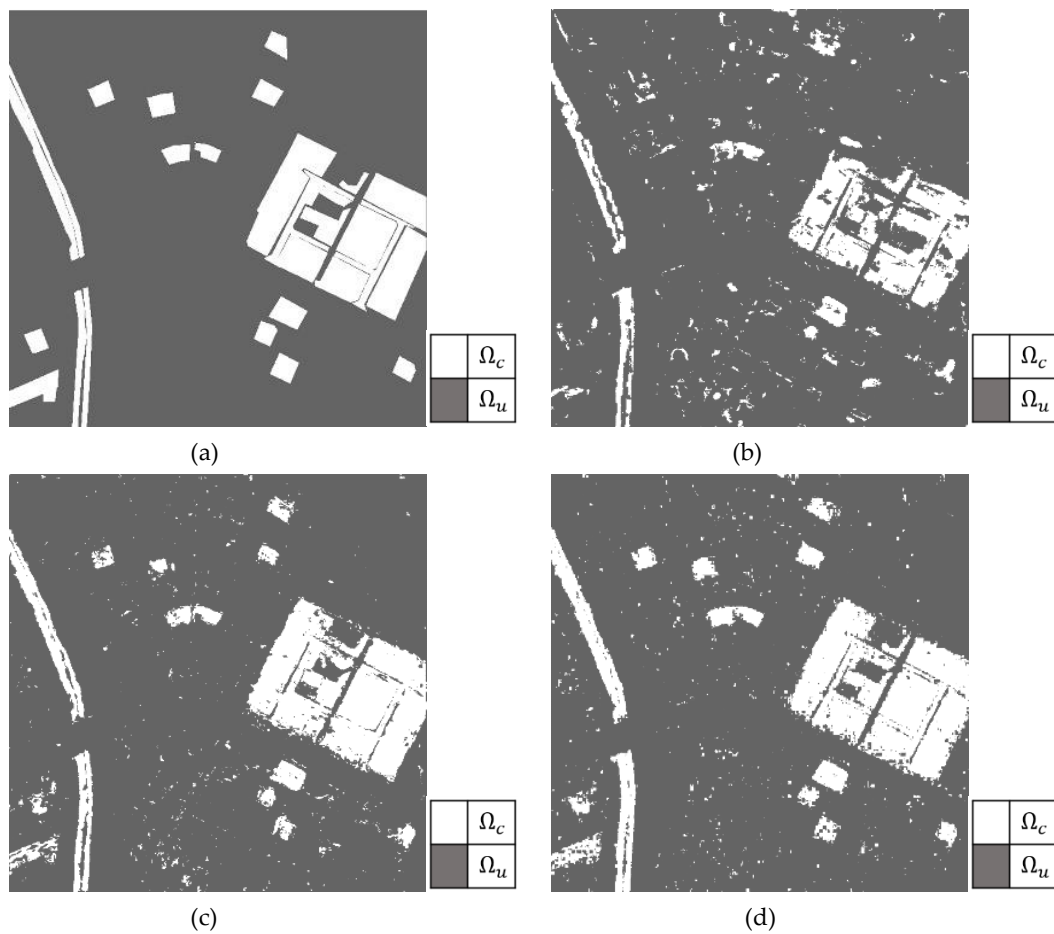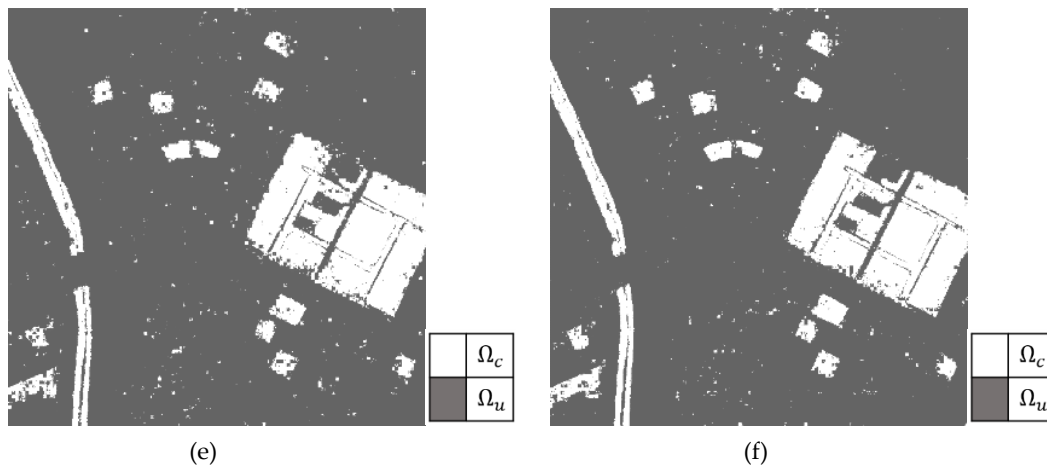

(a)

(b)

(c)

(d)

Figure 5. *Cont.*

(e)

(f)

Figure 5. Change detection maps obtained from the proposed and other methods for site 1 including (**a**) ground truth map (**b**) LSTM, (**c**) 2D CNN-LSTM, (**d**) Re3FCN, (**e**) the proposed method without transfer learning, and (**f**) the proposed method with transfer learning. Ω_c and Ω_u represent changed and unchanged classes, respectively.

Table 3. Comparison of results of assessment parameters for different change detection methods for site 1. "Kappa" and "TL" represent Kappa coefficient and transfer learning, respectively.

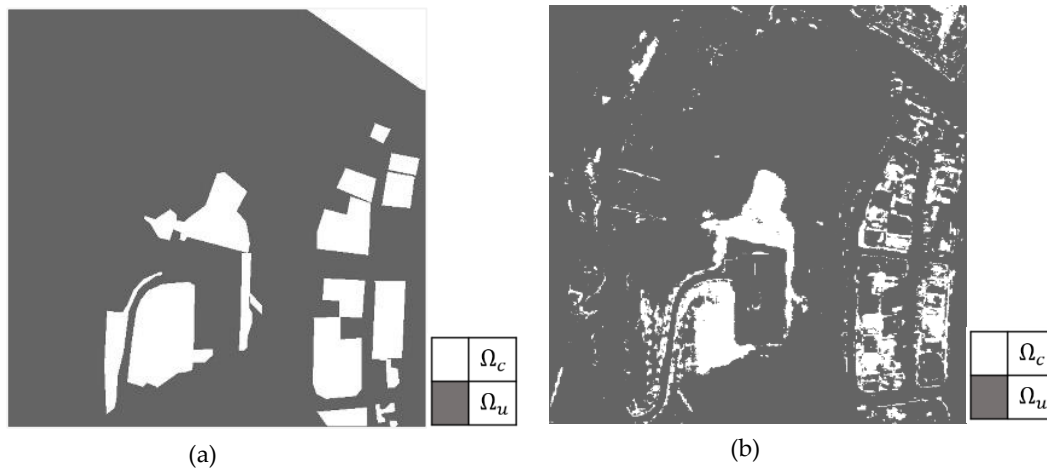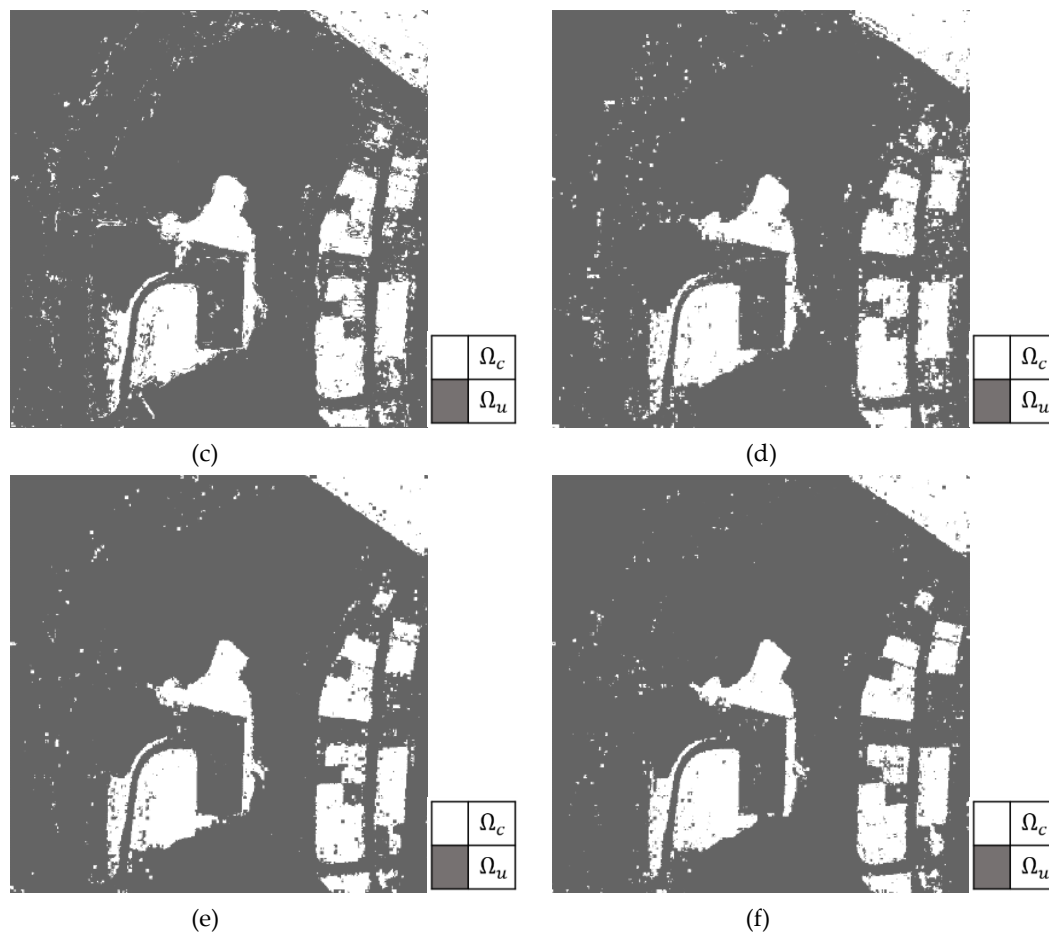| Change Detection Methods	OA	Kappa	F1 Score
LSTM	0.9136	0.6386	0.6876
2DCNN-LSTM	0.9597	0.8443	0.8680
Re3FCN	0.9674	0.8984	0.8978
Multiscale Re3FCN without TL	0.9717	0.8923	0.9090
Multiscale Re3FCN with TL	0.9790	0.9201	0.9326

(a)

(b)

Figure 6. *Cont.*

Figure 6. Change detection maps obtained from the proposed method and other methods for site 2 including (**a**) ground truth map (**b**) LSTM, (**c**) 2D CNN-LSTM, (**d**) Re3FCN, (**e**) the proposed method without transfer learning, and (**f**) the proposed method with transfer learning. Ω_c and Ω_u represent changed and unchanged classes, respectively.

Table 4. Comparison of results of evaluation parameters for different change detection methods for Site 2.

Change Detection Methods	OA	Kappa	F1 Score
LSTM	0.8826	0.5350	0.6010
2DCNN-LSTM	0.9565	0.8518	0.8783
Re3FCN	0.9633	0.8766	0.8990
Multiscale Re3FCN without TL	0.9759	0.9158	0.9304
Multiscale Re3FCN with TL	0.9795	0.9288	0.9412

5. Discussion

5.1. Comparison with Previous Studies

Although the LSTM learns the rules for change detection between temporal data, the images must be flattened for use with the fully connected LSTM network. Therefore, the LSTM is unsuitable for image analysis because it ignores spatial connectivity and the large weight matrix size increases the computational cost [44]. Therefore, change detection methods using LSTM, such as LSTM and 2D CNN-LSTM, relatively detect changes as unchanged areas than FCN-based change detection methods. However, using LSTM with 2D CNN, the change detection accuracies increase compared with when only LSTM is used. For example, the improvements in overall accuracy and Kappa coefficient are 4.6%

and 0.2057 for site 1 and 7.3% and 0.3168 for site 2. The results show that convolutional layers extract meaningful features from temporal images, with the features improving change detection accuracies.

When comparing 2D CNN-LSTM with Re3FCN, the results show superior performance for the Re3FCN for sites 1 and 2. The difference in the two change detection methods is that the fully connected LSTM is replaced by the convolutional LSTM and 3D filters are used instead of 2D filters in the convolutional layers. The convolutional LSTM models the temporal dependency of inputs, maintaining the spatial structure, whereas the 3D convolution effectively exploits spatial and spectral information simultaneously [34]. The reflectance pattern through spectral bands is crucial information for high spatial resolution spatial satellite images ranging from the visible to near-infrared. For example, when reflected radiation is in the near-infrared than visible bands, the vegetation in that pixel is likely dense vegetation. Therefore, the spectral information is crucial for change detection using satellite images.

The Re3FCN was developed using multiscale 3D filters and transfer learning. The improvements in overall accuracies are 1.16% and 1.62% for sites 1 and 2, respectively. Furthermore, the F1 score for sites 1 and 2 increased to 0.0348 and 0.0422, respectively. Results show that changed pixels are correctly classified as changed classes by the proposed change detection method. Objects with different shapes and characteristics are identified in high spatial resolution images; therefore, multiscale 3D filters assist in extracting meaningful features and improving the change detection results.

5.2. The Effect of Transfer Learning

The multiscale Re3FCN without transfer learning was randomly initialized at the start of the iteration. Conversely, the multiscale Re3FCN with transfer learning used pre-trained weights and biases, which are convolutional layers with multiscale filters, in the FCN for semantic segmentation. When the network involves pre-trained convolutional layer with multiscale filters, the change detection results slightly improved. For example, the overall accuracy increased from 0.9717 to 0.9790 and Kappa coefficient from 0.8923 to 0.9201 for site 1. Furthermore, the F1 scores increased to 0.0236 (site 1) and 0.0108 (site 2), respectively. Thus, transfer learning provides more rational initial values than the randomly selected values, thereby improving the change detection performance under the same experimental conditions.

6. Conclusions

In this study, change detection was conducted using an FCN with multiscale 3D filters and convolutional LSTM. As the proposed change detection network detects changes by analyzing the temporal information of feature maps obtained from temporal images, extracting meaningful features can improve the change detection results. Therefore, multiscale 3D filters were used in the initial phase of change detection network development to extract various spatial and spectral features from high spatial resolution images. Furthermore, the filters used pre-trained values from the ISPRS dataset to overcome the lack of training samples. The appropriate combination of 3D filters was determined by analyzing accuracy by class, and the classification and change detection performance were improved using multiscale 3D filters. The change detection results on the KOMPSAT-3A were compared with those of the LSTM, 2D CNN-fully connected LSTM, and Re3FCN; results revealed that the proposed change detection method outperformed others. Particularly, the change detection results were improved when using pre-trained values.

However, several problems are associated with the proposed change detection method. For example, since it uses multiple 3D filters for temporal images in parallel, the computing cost may increase depending on the learning environment. Furthermore, differences in spatial resolution and the class types between the source domain (ISPRS dataset) and target domain (KOMPSAT-3A) were not considered. To solve this problem, we developed the transfer learning technique applicable for broad applications. This is expected to improve usage of the large amounts of data extracted for detecting changes in different high spatial resolution satellite images.

References

1. Washaya, P.; Balz, T.; Mohamadi, B. Coherence change-detection with sentinel-1 for natural and anthropogenic disaster monitoring in urban areas. *Remote Sens.* **2018**, *10*, 1026. [CrossRef]

2. Giustarini, L.; Hostache, R.; Matgen, P.; Schumann, G.; Bates, P.D.; Mason, D.C. A change detection approach to flood mapping in urban areas using TerraSAR-X. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 2417–2430. [CrossRef]

3. Brisco, B.; Schmitt, A.; Murnaghan, K.; Kaya, S.; Roth, A. SAR polarimetric change detection for flooded vegetation. *Int. J. Digit. Earth* **2013**, *6*, 103–114. [CrossRef]

4. Schultz, M.; Shapiro, A.; Clevers, J.; Beech, C.; Herold, M.; Schultz, M.; Shapiro, A.; Clevers, J.G.P.W.; Beech, C.; Herold, M. Forest cover and vegetation degradation detection in the Kavango Zambezi transfrontier conservation area using BFAST monitor. *Remote Sens.* **2018**, *10*, 1850. [CrossRef]

5. Muro, J.; Canty, M.J.; Conradsen, K.; Hüttich, C.; Nielsen, A.A.; Skriver, H.; Remy, F.; Strauch, A.; Thonfeld, F.; Menz, G. Short-term change detection in wetlands using Sentinel-1 time series. *Remote Sens.* **2016**, *8*, 795. [CrossRef]

6. Manavalan, P.; Kesavasamy, K.; Adiga, S. Irrigated crops monitoring through seasons using digital change detection analysis of IRS-LISS 2 data. *Int. J. Remote Sens.* **1995**, *16*, 633–640. [CrossRef]

7. Deng, J.; Huang, Y.; Chen, B.; Tong, C.; Liu, P.; Wang, H.; Hong, Y. A methodology to monitor urban expansion and green space change using a time series of multi-sensor SPOT and sentinel-2A images. *Remote Sens.* **2019**, *11*, 1230. [CrossRef]

8. Khanal, N.; Uddin, K.; Matin, M.; Tenneson, K. automatic detection of spatiotemporal urban expansion patterns by fusing OSM and landsat data in Kathmandu. *Remote Sens.* **2019**, *11*, 2296. [CrossRef]

9. Ji, C.Y.; Qinhua, L.; Sun, D.; Wang, S.; Lin, P.; Li, X. Monitoring urban expansion with remote sensing in China. *Int. J. Remote Sens.* **2001**, *22*, 1441–1455. [CrossRef]

10. Singh, A. Review article digital change detection techniques using remotely-sensed data. *Int. J. Remote Sens.* **1989**, *10*, 989–1003. [CrossRef]

11. Jeong, J. Developments of urban change detection methods according to spatial resolution of satellite images application of KOMPSAT 1 images into urban area. *Geogr. J. Korea* **2005**, *39*, 161–170. (In Korean with English Abstract).

12. Dellinger, F.; Delon, J.; Gousseau, Y.; Michel, J.; Tupin, F. Change Detection for High Resolution Satellite Images, Based on SIFT Descriptors and an a Contrario Approach. In Proceedings of the IEEE Geoscience and Remote Sensing Symposium, Quebec City, QC, Canada, 13–18 July 2014; pp. 1281–1284.

13. Wulder, M.A.; Ortlepp, S.M.; White, J.C.; Coops, N.C.; Coggins, S.B. Monitoring tree-level insect population dynamics with multi-scale and multi-source remote sensing. *J. Spat. Sci.* **2008**, *53*, 49–61. [CrossRef]

14. Hussain, M.; Chen, D.; Cheng, A.; Wei, H.; Stanley, D. Change detection from remotely sensed images: From pixel-based to object-based approaches. *ISPRS J. Photogramm.* **2013**, *80*, 91–106. [CrossRef]

15. Bindschadler, R.A.; Scambos, T.A.; Choi, H.; Haran, T.M. Ice sheet change detection by satellite image defferencing. *Remote Sens. Environ.* **2010**, *114*, 1353–1362. [CrossRef]

16. Malila, W.A. Change vector analysis: An approach for detecting forest changes with Landsat. In Proceedings of the LARS Symposia, West Lafayette, IN, USA, 3–6 June 1980; p. 385.

17. Deng, J.S.; Wang, K.; Deng, Y.H.; Qi, G.J. PCA-based land-use change detection and analysis using multitemporal and multisensor satellite data. *Int. J. Remote Sens.* **2008**, *29*, 4823–4838. [CrossRef]

18. Niemeyer, I.; Marpu, P.; Nussbaum, S. Change detection using object features. In *Object-Based Image Analysis*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 185–201.

19. Papadomanolaki, M.; Vakalopoulou, M.; Karantzalos, K. A novel object-based deep learning framework for semantic segmentation of very high-tesolution temote sensing data: Comparison with convolutional and fully convolutional networks. *Remote Sens.* **2019**, *11*, 684. [CrossRef]

20. Guirado, E.; Tabik, S.; Alcaraz-Segura, D.; Cabello, J.; Herrera, F. Deep-learning versus OBIA for scattered shrub detection with Google earth imagery: Ziziphus Lotus as case study. *Remote Sens.* **2017**, *9*, 1220. [CrossRef]

21. Aguirre-Gutiérrez, J.; Seijmonsbergen, A.C.; Duivenvoorden, J.F. Optimizing land cover classification accuracy for change detection, a combined pixel-based and object-based approach in a mountainous area in Mexico. *Appl. Geogr.* **2012**, *34*, 29–37. [CrossRef]

22. Möller, M.; Lymburner, L.; Volk, M. The comparison index: A tool for assessing the accuracy of image segmentation. *Int. J. Appl. Earth Obs.* **2007**, *9*, 311–321. [CrossRef]

23. Zhang, C.; Wei, S.; Ji, S.; Lu, M. Detecting large-scale urban land cover changes from high spatial resolution remote sensing images using CNN-based classification. *ISPRS Int. Geo-Inf.* **2019**, *8*, 189. [CrossRef]

24. Zhan, Y.; Fu, K.; Yan, M.; Sun, X.; Wang, H.; Qiu, X. Change detection based on deep siamese convolutional network for optical aerial images. *IEEE Geo. Sci. Remote S.* **2017**, *14*, 1845–1849. [CrossRef]

25. Wiratama, W.; Lee, J.; Park, S.E.; Sim, D. Dual-dense convolution network for change detection of high-resolution panchromatic imagery. *Appl. Sci.* **2018**, *8*, 1785. [CrossRef]

26. Wang, Q.; Zhang, X.; Chen, G.; Dai, F.; Gong, Y.; Zhu, K. Change detection based on Faster R-CNN for high-resolution remote sensing images. *Remote Sens. Lett.* **2018**, *9*, 923–932. [CrossRef]

27. Wang, Q.; Yuan, Z.; Du, Q.; Li, X. GETNET: A general end-to-end 2-D CNN framework for hyperspectral image change detection. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 3–13. [CrossRef]

28. Gong, M.; Yang, H.; Zhang, P. Feature learning and change feature classification based on deep learning for ternary change detection in SAR images. *ISPRS J. Photogramm.* **2017**, *129*, 212–225. [CrossRef]

29. Lyu, H.; Lu, H.; Mou, L. Learning a transferable change rule from a recurrent neural network for land cover change detection. *Remote Sens.* **2016**, *8*, 506. [CrossRef]

30. Geng, J.; Fan, J.; Wang, H.; Ma, X. Change Detection of Marine Reclamation Using Multispectral Images via Patch-based Recurrent Neural Network. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Fort Worth, TX, USA, 23–28 July 2017; pp. 612–615.

31. Kong, Y.-L.; Huang, Q.; Wang, C.; Chen, J.; Chen, J.; He, D. Long Short-Term Memory Neural Networks for Online Disturbance Detection in Satellite Image Time Series. *Remote Sens.* **2018**, *10*, 452. [CrossRef]

32. Chen, H.; Wu, C.; Du, B.; Zhang, L.; Wang, L. Change detection in multisource VHR images via deep Siamese convolutional multiple-layers recurrent neural network. *IEEE Trans. Geosci. Remote Sens.* **2019**, 1–17. [CrossRef]

33. Mou, L.; Bruzzone, L.; Zhu, X.X. Learning spectral-spatial-temporal features via a recurrent convolutional neural network for change detection in multispectral imagery. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 924–935. [CrossRef]

34. Song, A.; Choi, J.; Han, Y.; Kim, Y. Change detection in hyperspectral images using recurrent 3d fully convolutional networks. *Remote Sens.* **2018**, *10*, 1827. [CrossRef]

35. Fu, G.; Liu, C.; Zhou, R.; Sun, T.; Zhang, Q. Classification for high resolution remote sensing imagery using a fully convolutional network. *Remote Sens.* **2017**, *9*, 498. [CrossRef]

36. Li, Y.; Zhang, H.; Xue, X.; Jiang, Y.; Shen, Q. Deep learning for remote sensing image classification: A survey. *Data Min. Knowl. Disc.* **2018**, *8*, e1264. [CrossRef]

37. Liang, Y.; Monteiro, S.T.; Saber, E.S. Transfer Learning for High Resolution Aerial Image Classification. In Proceedings of the IEEE Applied Imagery Pattern Recognition Workshop, Washington, DC, USA, 18–20 October 2016; pp. 1–8.

38. Sinno, J.P.; Qiang, Y. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **2009**, *22*, 1345–1359.

39. Song, A. A Novel Deep Learning Framework for Multi-Class Change Detection of Hyperspectral Images. Ph.D. Thesis, Seoul National University, Seoul, Korea, February 2019.

40. Lee, H.; Kwon, H. Going deeper with contextual CNN for hyperspectral image classification. *IEEE Trans. Image Process.* **2017**, *26*, 4843–4855. [CrossRef] [PubMed]

41. Yuan, Q.; Wei, Y.; Meng, X.; Shen, H.; Zhang, L. A multiscale and multidepth convolutional neural network for remote sensing imagery pan-sharpening. *IEEE J. Sel. Top. Appl.* **2018**, *11*, 978–989. [CrossRef]

42. Gong, Z.; Zhong, P.; Yu, Y.; Hu, W.; Li, S. A CNN with multiscale convolution and diversified metric for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 3599–3618. [CrossRef]

43. Liang, M.; Jiao, L.; Yang, S.; Liu, F.; Hou, B.; Chen, H. Deep multiscale spectral-spatial feature fusion for hyperspectral images classification. *IEEE J. Sel. Top. Appl.* **2018**, *11*, 2911–2924. [CrossRef]

44. Liao, W.; Wang, X.; An, D.; Wei, Y. Hyperspectral Imaging Technology and Transfer Learning Utilized in Haploid Maize Seeds Identification. In Proceedings of the International Conference on High Performance Big Data and Intelligent Systems, Shenzhen, China, 9–11 May 2019; pp. 157–162.

45. Xingjian, S.H.I.; Chen, Z.; Wang, H.; Yeung, D.Y.; Wong, W.K.; Woo, W.C. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Adv. Neur. Inf.* **2015**, 802–810.

46. Landis, J.R.; Koch, G.G. The measurement of observer agreement for categorical data. *Biometrics* **1977**, *33*, 159–174. [CrossRef]

47. ISPRS WG III/4. ISPRS 2D Semantic Labeling Contest. Available online: http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html (accessed on 16 January 2020).

48. Acharya, T.; Yang, I.; Lee, D. Land cover classification using a kompsat-3a multi-spectral satellite image. *Appl. Sci.* **2016**, *6*, 371. [CrossRef]

49. Choi, J.; Kim, G.; Park, N.; Park, H.; Choi, S. A hybrid pansharpening algorithm of high spatial resolution satellite images that employs injection gains based on NDVI to reduce computational costs. *Remote Sens.* **2017**, *9*, 976. [CrossRef]

50. ArcGIS Webmap. Available online: https://www.arcgis.com/home/webmap/viewer.html (accessed on 16 January 2020).