Change Detection in High Resolution Satellite Images Using an Ensemble of Convolutional Neural Networks

Kyungsun Lim, Dongkwon Jin, and Chang-Su Kim
School of Electrical Engineering, Korea University, Seoul, Korea
E-mail: {kslim, dongkwonjin}@mcl.korea.ac.kr, changsukim@korea.ac.kr

***Abstract*—In this paper, we propose a novel change detection algorithm for high resolution satellite images using convolutional neural networks (CNNs), which does not require any pre-processing, such as ortho-rectification and classification. When analyzing multi-temporal satellite images, it is crucial to distinguish viewpoint or color variations of an identical object from actual changes. Especially in urban areas, the registration difficulty due to high-rise buildings makes conventional change detection techniques unreliable, if they are not combined with pre-processing schemes using digital surface models or multi-spectral information. We design three encoder-decoder-structured CNNs, which yield change maps from an input pair of RGB satellite images. For the supervised learning of these CNNs, we construct a large fully-labeled dataset using Google Earth images taken in different years and seasons. Experimental results demonstrate that the trained CNNs detect actual changes successfully, even though image pairs are neither perfectly registered nor color-corrected. Furthermore, an ensemble of the three CNNs provides excellent performance, outperforming each individual CNN.**

I. Introduction

Change detection for satellite imagery is used in global remote sensing [1]; identifying land cover changes over wide areas is important in various applications, including environmental monitoring, disaster evaluation, and urban expansion study [2]–[4]. Since recent satellite cameras support high resolutions and can even capture people on the ground, it is possible to monitor small objects such as buildings in a city. However, high resolution temporal satellite images pose new challenges since the complete matching is infeasible. The images of objects vary since the locations and/or the photographic angles of cameras are not consistent. Also, the colors of objects are distorted depending on the camera sensors and the environments. Moreover, the shadows of high-rise buildings and the color variations of plants due to season change should be distinguished from actual changes. These low correlations of position and color between corresponding pixels make change detection very challenging. Thus, various techniques have been proposed to eliminate the variations between a pair of images or to boost the distinguishing capability of actual changes from the variations.

Several pre-processing methods have been proposed to minimize the effects of undesirable variations in temporal satellite images. Matching pixels for the same geographical coordinates is essential, and radiometric correction between an image pair is also helpful. However, according to the studies in [6], [7], the accuracy of image registration tends to deteriorate as the spatial resolution of satellite images increases. Different photographic angles change the appearance of objects, and these variations are magnified especially in urban areas due to the complicated structures. The side effects of these variations can be alleviated by ortho-rectification. Alternatively, object classification can be performed to determine land cover and land use classes [6], [8]. After the classification, the registration becomes less important, and changed regions can be found by verifying whether two images contain the same object or not. However, this pre-processing is a hard task in itself and may be unreliable in practice.

Meanwhile, change detection methods have been devised to separate distortions from actual changes. An approach is to use the wide spectrum analysis over the optical range. Principal component analysis (PCA) is often used to reduce the dimensionality of a multi-band data. Nielsen [9] proposed the iteratively re-weighted multivariate alteration detection to generate transformed images using multi-spectral data. Another approach is based on machine learning. Celik [10] proposed an unsupervised algorithm based on k-means, which clusters feature vectors derived by PCA. Pacifici *et al.* [11] trained a neural network to classify pixels into land cover classes, as a pre-processing step for change detection. More recently, Gong *et al.* [12] used a multi-layer perceptron to generate feature maps from two satellite images and then classified pixels using a convolutional neural network (CNN).

Among the machine learning techniques, CNNs have become popular recently in many vision tasks [13]–[20], as well as change detection. For change detection, Braham and Droogenbroeck [21] used a CNN to compare patches in a current frame with the background image. Sakurada and Okatani [22] compared CNN features of a pair of street images. Alcantarilla *et al.* [23] developed a street-view change detection technique by training a CNN to separate color changes from seasonal variations. On the other hand, CNNs are also used in matching problems that compare two images with significant appearance variations. For example, Nam and Han [24] exploited a CNN to track a moving object in an image sequence, which may experience severe appearance changes, such as partial occlusion. Parkhi *et al.* [25] utilized a CNN for face recognition. Zbontar and LeCun [26] compared

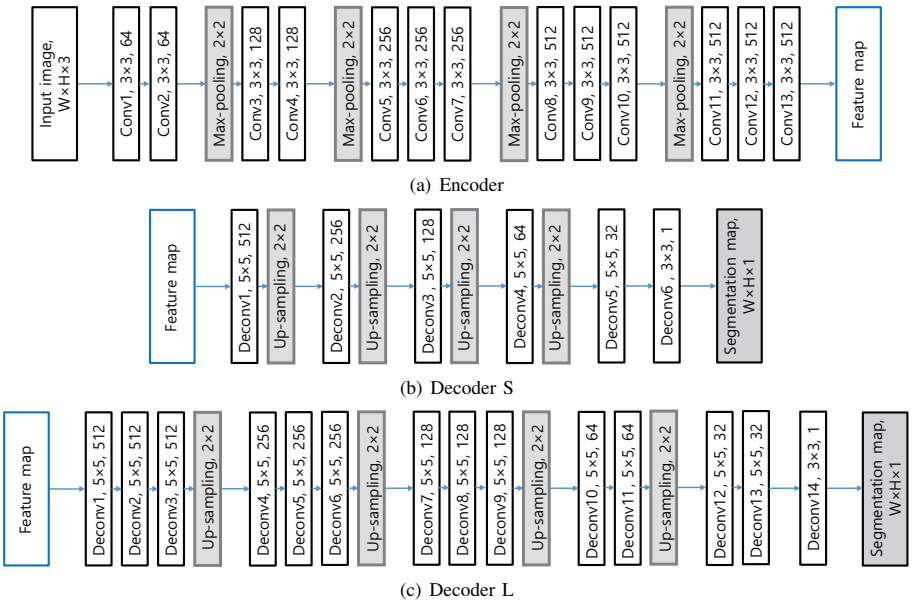


Fig. 1. The structure of the encoder and the decoders. The encoder is modified from the VGG16 network [5]. The decoders extract segmentation maps using the feature vector from the encoder. The numbers following the name of a convolutional layer indicate the kernel size and the number of filters (or equivalently output channels).

CNN features of a stereo image pair to achieve pixel-wise matching. The successful employment of CNNs in the aforementioned researches implies that CNN features are effective for identifying objects in spite of their appearance variations.

In this paper, we develop a CNN-based change detection algorithm for temporal satellite images. We design three CNNs with the encoder-decoder structure [15], [17], [27]–[29] each of which yields a 1-channel segmentation map representing changed regions. This work is motivated by our previous work [30], which proposed a background subtraction algorithm to extract change areas using an encoder-decoder structured CNN. The algorithm [30] exhibits robustness even in videos with jitters and noises, such as blizzard. In this encoder-decoder architecture, the encoder extracts high-level features from an input image, and then the decoder converts it into a prediction result suitable for a specific task.

We fine-tune the image classification network in [5] and employ it as the encoder. On the other hand, we design the decoders for the purpose of change detection. By combining the encoder and the decoders, we construct the single short network (SSN), the single long network (SLN), and the double long network (DLN). SSN is a modified, improved version of the network in our previous work [30]. SLN is a combination of a deeper decoder with the encoder in SSN. Also, DLN is a Siamese network [25], [31] that contains two identical encoders. We obtain segmentation maps from the three CNNs, respectively, and get a final change mask using the average of the three maps. Experimental results demonstrate that the proposed algorithm provides promising results.

For the supervised learning of these CNNs, we construct a large fully-labeled dataset, by capturing time series of Landsat images over 13 urban areas in Seoul, South Korea. We divide them into 600×600 images and get 1,000 pairs of temporal satellite images. We also manually extract the binary ground truth maps, whose pixel values are 1 if the corresponding pixel experiences a change and 0 otherwise. We will make this dataset publicly available.

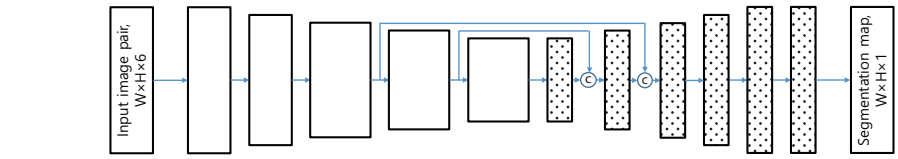
The rest of this paper is organized as follows: Section II presents the CNN structures and learning details. Sections III and IV describe experimental setting and results, respectively. Finally, Section V concludes this work.

II. Proposed Algorithm

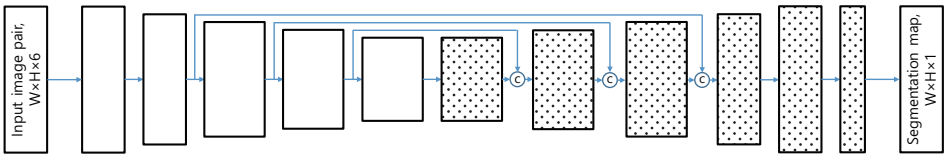
We propose three encoder-decoder-structured convolutional neural networks (encoder-decoder CNNs) for the purpose of change detection. The proposed algorithm detects changes in bi-temporal input images and yields a segmentation map. To this end, the three networks are trained in an end-to-end manner using temporal images and the corresponding ground-truth binary map, which represents change regions. By combining an encoder with two decoders, we design three encoder-decoder CNNs and train them separately. Finally, we get the final binary change map, by thresholding the average output of the three CNNs.

A. Encoder and Decoders

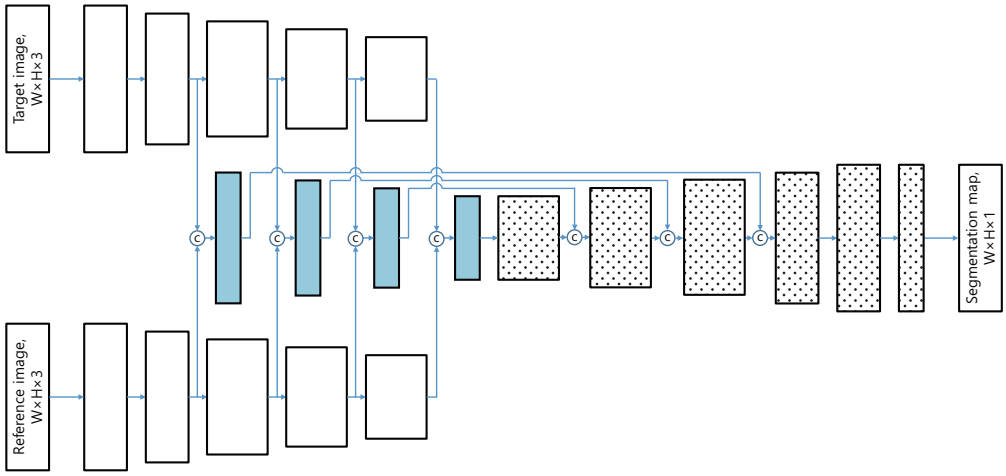
An encoder-decoder CNN can be configured by connecting an encoder network and a decoder network. In general, an encoder network comprises convolutional layers and max-pooling layers, which transform an input image into a feature



(a) Single short network (SSN)



(b) Single long network (SLN)



(c) Double long network (DLN)

Fig. 2. The architecture of the three networks, which are combinations of the encoder and the decoders. White blocks depict the encoder parts, while patterned blocks the decoder parts. Blue blocks are added convolutional layers for merging features. The 'c' operation concatenates two features along the channel dimension.

map with a lower spatial resolution and more channels. We adopt the VGG16 network [5] without the fully-connected layers as the encoder, as shown in Fig. 1(a). Through 13 convolutional layers and 4 max-pooling layers, the encoder yields a feature map, which has 512 channels and one-sixteenth spatial resolution of an input image. Note that a network including convolutional layers only without fully-connected layers can accept input images of any spatial sizes.

We design two decoders, which are compatible with the encoder and suitable for the purpose of change detection. In the decoders, we use deconvolutional layers, which reduce the numbers of channels, and up-sampling layers, which increase spatial resolutions based on the bilinear interpolation. We make the two decoders with different lengths, as shown in Fig. 1(b) and (c). The shorter one, referred to as 'Decoder S,' consists of 6 deconvolutional layers and 4 up-sampling layers, and the longer one, 'Decoder L,' consists of 14 deconvolutional layers and 4 up-sampling layers. To all deconvolution layers in the decoders except for the last layers, the batch normalization (BN) [32] is applied and then the parametric rectified linear unit (PReLU) activation function is employed. The last layers of the decoders are followed by sigmoid layers, which normalize output values into $[0, 1]$. Consequently, the decoders convert the feature map from the encoder to a 1-channel segmentation map. Each pixel in the segmentation map represents the likelihood that the corresponding pixel in the target image experiences a change.

Inspired by the CNN applications in [28], [33], we exploit intermediate features from the encoder as well as its last output. We reuse these features, each of which is the input to a certain pooling layer in the encoder, by concatenating them with the corresponding features in the decoder along the channel dimension. The concatenated vector is input to the next layer. We experimentally determine which vectors to reuse.

B. Networks

1) Single Short Network: SSN in Fig. 2(a) is a combination of the encoder and the decoder S. It is designed to take two images as the input and produce a segmentation map, which identifies changed regions. Since the VGG16 network

takes a 3-channel image, we modify its front convolutional layer to take 6-channel input. We combine a target image and a reference image to make the 6-channel input. In SSN, concatenating intermediate features from the encoder occurs before the first and the second up-sampling layers in the decoder.

2) Single Long Network: Similarly, SLN is formed by combining the encoder and the decoder L, as shown in Fig. 2(b). As in SSN, SLN also takes the combined target and reference images as the input and the front convolutional layer is modified. The concatenation of intermediate features occurs before the first, the second, and the third up-sampling layers in the decoder.

3) Double Long Network: DLN is a Siamese network to use twin encoders. It consists of two identical copies of the encoder and the decoder L. The two encoders of DLN take a target image and a reference image, respectively. Therefore, the front convolutional layers are not modified in contrast to SSN and SLN. As illustrated in Fig. 2(c), we add convolutional layers to merge the features from the two encoders. Each of these additional layers combines two features into a single feature with the same dimension. These combined features are input to the decoder or concatenated before the first, the second, and the third up-sampling layers in the decoder.

C. Training and Testing

To train and test the proposed networks, we use the Caffe library [34]. To train SSN, we set the input size to 448×448. For the deeper networks, *i.e.* SLN and DLN, we reduce the input size to 224×224 due to the memory limitation of a GPU and instead retain the details of an input satellite image by cropping. To initialize the encoder parameters, we adopt the VGG16 parameters pre-trained on the ImageNet dataset [35] for the image classification task. The pre-trained parameters are applied to the encoders, except for the front layers of SSN and SLN. We randomly initialize the other parameters. To optimize the parameters, we calculate the cross-entropy loss between a ground-truth binary image and a predicted segmentation map and then update the parameters using the Adam technique [36]. We set the initial learning rate to 0.001 for the randomly initialized layers and to 0.0001 for the pre-trained layers. After every 10,000 iterations, the learning rates are reduced by a factor of $\frac{3}{4}$. We fix the weight decay to 0.02. We train SSN through 26,000 iterations, SLN through 36,000 iterations, and DLN through 40,000 iterations. We set the batch size to 12 for all three networks.

To test the networks, we resize test images to 672×672. The trained networks yield a 1-channel image, called the segmentation map, which is the result of the sigmoid activation function and has values between 0 and 1.

D. Ensemble

We make a final result by fusing the output maps of the three networks. More specifically, the change regions are determined by thresholding the average of the three segmentation maps. We select the threshold value to achieve the best performance.

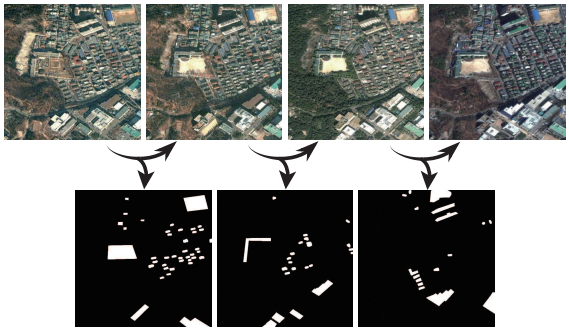


Fig. 3. Illustration of the dataset construction for a subarea within Area 4.

III. Experimental Setting

A. Dataset

We create a multi-temporal dataset for change detection using Google Earth images, which were captured from the Landsat 7 and 8 satellites. We acquire three-band (RGB) images including urban areas around Seoul in South Korea. At 10 areas of Seoul and 3 areas of cities around Seoul, we gather satellite images that were taken at different time instances. Table I lists the detailed information of those images. The Seoul areas are densely populated urban ones, while the areas around Seoul are agricultural areas that vary in color depending on seasonal conditions. The temporal images are geometrically registered, but the registration is partly inaccurate by a few pixels. Moreover, the images are not orthophotos, so tall buildings look different at different time instances. There are also color variations, but we do not perform radiometric correction.

We divide all satellite images into 600×600 images. We consider a pair of images for the same area but at different time instances. We can make 1242 such image pairs in 328 subareas. Among them, we make pixel-wise binary change maps for 1,000 image pairs, after excluding inappropriate pairs due to ambiguity (*e.g.* mostly composed of river or cloud). Fig. 3 illustrates how we construct the dataset.

We utilize those 1,000 image pairs and corresponding ground-truth maps to train or test the proposed CNNs. As test data, we consider 14 subareas and use the 50 image pairs taken over the subareas. We use the others as train data.

B. Evaluation Metrics

For the performance assessment, we classify pixels in a change detection map using the corresponding ground-truth. True positive (TP) and true negative (TN) denote the numbers of pixels correctly predicted as changed and unchanged pixels, respectively. False alarm (FA) is the number of pixels predicted as changed but unchanged in the ground-truth, and miss alarm (MA) is the number of inverse cases. Then, the precision and the recall are defined as

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FA}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{MA}}. \qquad (1)$$

TABLE I
DETAILED DESCRIPTION OF THE IMAGES IN THE PROPOSED DATASET FOR CHANGE DETECTION.

	Location	Resolution (m)	Size (pixels)	Acquisition Date					
				Feb. 15, 2002	Jan. 5, 2006	Oct. 2, 2008	Mar. 20, 2012	Oct. 16, 2013	Jun. 26, 2016
Area 1	Gangnam-gu	0.74	1200×2400	✓	✓	✓	✓	✓	✓
Area 2	Gwangjin-gu	0.37	2715×4780		✓		✓		✓
Area 3	Jongno-gu	0.51	2830×4755	✓	✓		✓	✓	✓
Area 4	Seongbuk-gu	0.51	2949×4780	✓	✓	✓	✓	✓	✓
Area 5	Seongdong-gu	0.38	2725×4740		✓	✓	✓	✓	✓
Area 6	Songpa-gu	0.74	1200×2385	✓	✓	✓	✓	✓	✓
Area 7	Songpa-gu	0.74	1200×2400	✓	✓	✓	✓	✓	✓
Area 8	Yeongdeungpo-gu	0.74	2830×4780	✓		✓		✓	✓
Area 9	Mapo-gu	0.50	2400×4800	✓		✓	✓	✓	✓
Area 10	Gwanak-gu	0.74	2400×4800				✓	✓	✓
Area 11	Bucheon-si	0.50	1200×2400	✓	✓	✓	✓	✓	✓
Area 12	Gimpo-si	0.74	2400×4800	✓	✓	✓		✓	✓
Area 13	Siheung-si	0.74	2400×4800	✓	✓	✓	✓	✓	

TABLE II
PERFORMANCE COMPARISON OF THE THREE NETWORKS AND THE CONVENTIONAL ALGORITHM [10].

Network	F1-score	F2-score
SSN	66.26	**72.19**
SLN	**66.74**	71.02
DLN	64.01	70.19
PCA&k-means	17.56	14.95

TABLE III
PERFORMANCE COMPARISON OF DIFFERENT ENSEMBLE METHODS.

	Networks	F1-score	F2-score
Average	SSN+SLN	69.04	73.76
	SSN+DLN	69.76	74.39
	SLN+DLN	69.74	73.82
	SSN+SLN+DLN	**71.16**	**75.21**

Also, we calculate two types of F-measure as evaluation metrics. The F-measure is given by

$$F_\beta = (1+\beta^2) \cdot \frac{\text{Precision} \cdot \text{Recall}}{(\beta^2 \cdot \text{Precision}) + \text{Recall}} \tag{2}$$

where β determines the ratio of the influence of precision and recall. We use F1-score, which is the traditional F-measure with $\beta = 1$. F1-score is the harmonic mean of precision and recall, so it is influenced by precision and recall with equal strength. We also use F2-score, which weighs recall more importantly than precision with $\beta = 2$. Therefore, F2-score gets a larger penalty by miss alarms than by false alarms. In a surveillance system, false alarms can be double checked by personnel while miss alarms do not have such opportunities. Accordingly, we decide that F2-score is more suitable as an assessment tool for change detection techniques.

IV. EXPERIMENTAL RESULTS

A. Network Structure

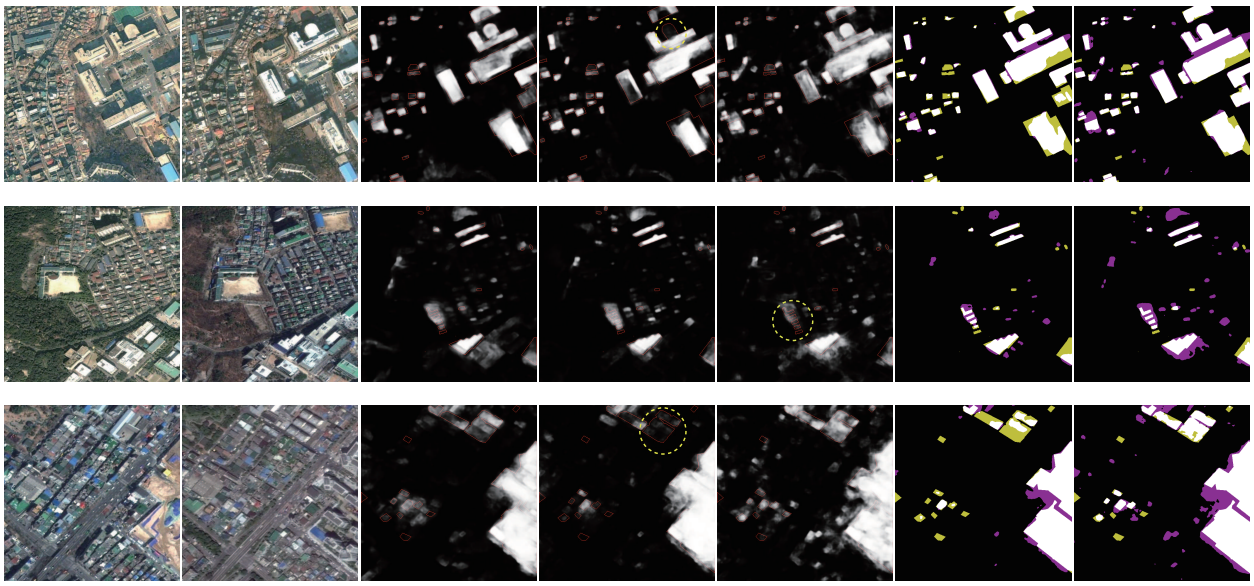
Table II compares the performances of the three networks. Whereas SLN yields a higher F1-score than SSN, SSN provides a higher F2-score than SLN. This means that SLN shows a better precision rate, but SSN yields a better recall rate. On the other hand, DLN yields the worst performances in terms of both F1-score and F2-score. In general, DLN causes more false alarms and miss alarms than the single networks. However, we observed from detection results that DLN succeeds to detect some change regions that both SSN and SLN fail to detect. Thus, it is complementary to SSN and SLN. Fig. 4 illustrates how the three networks yield different detection results.

B. Ensemble Performance

We test whether hybrids of the networks outperform the individual networks. Table III compares the results of various combinations. In case of SSN, the combination with DLN improves the performance by a bigger margin than the combination with SLN, even though DLN yields the worst individual performances. In case of SLN as well, the same tendency is observed. Moreover, the average of the three networks outperforms all the pairwise combinations. These synergistic effects of the ensemble scheme can be explained by the fact that errors from different networks occur at different areas. Fig. 4 shows the segmentation maps of the three networks and the ensemble scheme. Even when one of the networks causes false positives or false negatives, the right decisions of the other two networks can compensate for the errors. In Fig. 4, for example, yellow dashed circles indicate false positives and false negatives, which are corrected by the ensemble scheme.

C. Comparison Results

Table II compares the performances of the proposed algorithm with the conventional algorithm, PCA&k-means [10]. PCA&k-means is an unsupervised method using a difference image of the temporal image pair. Conventional algorithms, which simply use a difference image, cannot be expected to provide high performance without pre-processing, such as radiometric correction, ortho-rectification, and object classification. Fig. 5 compares change detection results of PCA&k-means with those of the proposed algorithm. In the first row,



(a) Target (b) Reference (c) SSN (d) SLN (e) DLN (f) Average (θ=0.38) (g) Average (θ=0.20)

Fig. 4. Qualitative comparison of detection results: (a) and (b) are temporal image pairs. (c)-(e) are the segmentation maps, detected by the three networks. (f) and (g) shows the ensemble results which are determined by binarizing the average of the three maps with a threshold 0.38 and 0.20 respectively. (g) True positives, false positives, false negatives are depicted in white, lime, and purple, respectively.

the input image pair has color distortions. Without radiometric correction, PCA&k-means yields lots of false negatives. In the second row, the same building exhibits different appearances due to different photographic angles. Especially, the rooftop of the building shifts and causes considerable differences. In the last row, plant colors in the agricultural area vary according to seasonal conditions. To overcome this difficulty, object classification may be necessary for the conventional algorithm to distinguish color changes of plants from actual changes. Compared with the conventional algorithm, the proposed algorithm provides more faithful change detection results without any pre-processing.

V. Conclusions

We proposed a change detection method for satellite images using CNNs, and constructed a large dataset to train the networks. The three networks have the encoder-decoder architectures and yield decent segmentation maps. Moreover, the average combination of the three networks yields F1-score of 71.16%, F2-score of 75.21%. These results are promising and indicate that the CNN-based technology facilitates superior change detection without requiring pre-processing, such as radiometric correction, ortho-rectification, and object classification.

VI. Acknowledgment

This work was supported in part by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2018-2016-0-00464) supervised by the IITP(Institute for Information & communications Technology Promotion) and in part by the Agency for Defense Development (ADD) and Defense Acquisition Program Administration (DAPA) of Korea (UC160016FD)

References

[1] A. Singh, "Review article digital change detection techniques using remotely-sensed data," *Int. J. Remote Sens.*, vol. 10, no. 6, pp. 989–1003, Jun. 1989.

[2] C. Mucher, K. Steinnocher, F. Kressler, and C. Heunks, "Land cover characterization and change detection for environmental monitoring of pan-europe," *Int. J. Remote Sens.*, vol. 21, no. 6-7, pp. 1159–1181, 2000.

[3] F. Bovolo and L. Bruzzone, "A split-based approach to unsupervised change detection in large-size multitemporal images: Application to tsunami-damage assessment," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 6, pp. 1658–1670, 2007.

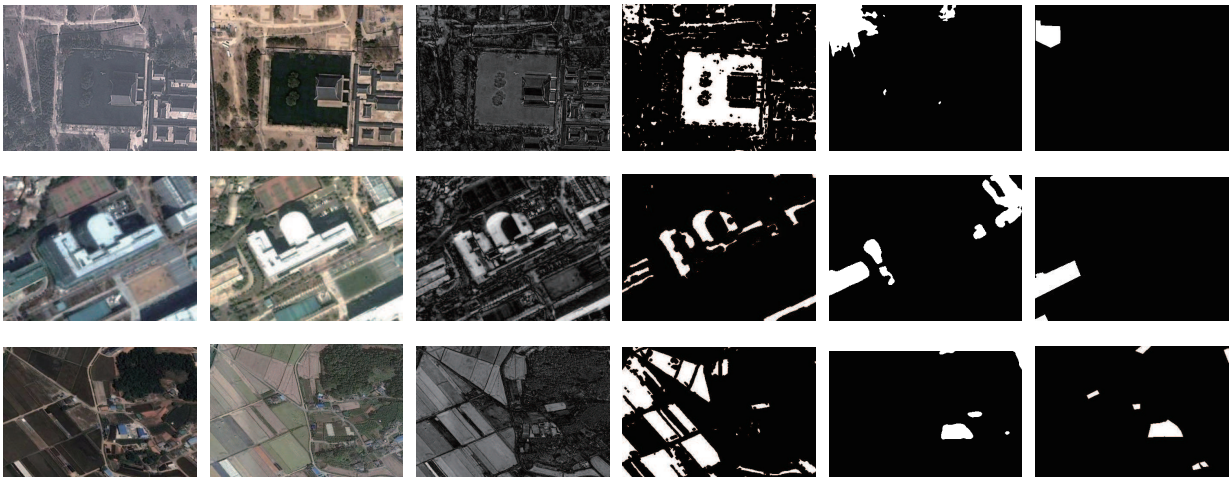
[4] L. Yang, G. Xian, J. M. Klaver, and B. Deal, "Urban land-cover change detection through sub-pixel imperviousness mapping using remotely sensed data," *Photogramm. Eng. Remote Sens.*, vol. 69, no. 9, pp. 1003–1010, 2003.

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

[6] C. Cleve, M. Kelly, F. R. Kearns, and M. Moritz, "Classification of the wildlandurban interface: A comparison of pixel-and object-based classifications using high-resolution aerial photography," *Comput. Environ. Urban Syst.*, vol. 32, no. 4, pp. 317–326, Jul. 2008.

[7] J. Im, J. Jensen, and J. Tullis, "Object-based change detection using correlation image analysis and image segmentation," *Int. J. Remote Sens.*, vol. 29, no. 2, pp. 399–423, 2008.

[8] V. Walter, "Object-based classification of remote sensing data for change detection," *ISPRS J. Photogram. Rem. Sens.*, vol. 58, no. 3, pp. 225–238, Jan. 2004.



(a) Target (b) Reference (c) Difference (d) PCA&k-means (e) Proposed (f) Ground-truth

Fig. 5. Change detection results by the conventional algorithm and the proposed algorithm: (a)-(b) temporal image pairs, (c) the difference image, (d) PCA&k-means [10] (e) the proposed algorithm, and (f) ground-truth.

[9] A. A. Nielsen, "The regularized iteratively reweighted MAD method for change detection in multi-and hyperspectral data," *IEEE Trans. Image Process.*, vol. 316, no. 2, pp. 463–478, Feb. 2007.

[10] T. Celik, "Unsupervised change detection in satellite images using principal component analysis and k-means clustering," *IEEE Trans. Geosci. Remote Sens.*, vol. 6, no. 4, pp. 772–776, 2009.

[11] F. Pacifici, F. D. Frate, C. Solimini, and W. J. Emery, "An innovative neural-net method to detect temporal changes in high-resolution optical satellite imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 9, pp. 2940–2952, Sep. 2007.

[12] M. Gong, H. Yang, and P. Zhang, "Feature learning and change feature classification based on deep learning for ternary change detection in SAR images," *ISPRS J. Photogram. Rem. Sens.*, vol. 129, pp. 212–225, Jul. 2017.

[13] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *CVPR*, 2016, pp. 3150–3158.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[15] J. Yang, B. Price, S. Cohen, H. Lee, and M.-H. Yang, "Object contour detection with a fully convolutional encoder-decoder network," in *CVPR*, 2016, pp. 193–202.

[16] Y. J. Koh and C.-S. Kim, "Primary object segmentation in videos based on region augmentation and reduction," in *CVPR*, 2017, pp. 7417–7425.

[17] W.-D. Jang and C.-S. Kim, "Online video object segmentation via convolutional trident network," in *CVPR*, 2017, pp. 5849–5858.

[18] H.-U. Kim and C.-S. Kim, "CDT: cooperative detection and tracking for tracing multiple objects in video sequences," in *ECCV*, 2016, pp. 851–867.

[19] Y. J. Koh and C.-S. Kim, "CDTS: collaborative detection, tracking, and segmentation for online multiple object segmentation in videos," in *ICCV*, 2017, pp. 3621–3629.

[20] J.-T. Lee, H.-U. Kim, and C.-S. Kim, "Semantic line detection and its applications," in *ICCV*, 2017, pp. 3229–3237.

[21] M. Braham and M. V. Droogenbroeck, "Deep background subtraction with scene-specific convolutional neural networks," in *ICSSIP*, 2016.

[22] K. Sakurada and T. Okatani, "Change detection from a street image pair using CNN features and superpixel segmentation," in *BMVC*, 2015.

[23] P. F. Alcantarilla, S. Stent, G. Ros, R. Arroyo, and R. Gherardi, "Street-view change detection with deconvolutional networks." in *Robotics: Science and Systems*, 2016.

[24] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *CVPR*, 2016.

[25] O. M. Parkhi, A. Vedaldi, A. Zisserman *et al.*, "Deep face recognition." in *BMVC*, 2015.

[26] J. Zbontar and Y. LeCun, "Computing the stereo matching cost with a convolutional neural network," in *CVPR*, 2015.

[27] F. J. Huang, Y.-L. Boureau, Y. LeCun *et al.*, "Unsupervised learning of invariant feature hierarchies with applications to object recognition," in *CVPR*, 2007.

[28] N. Liu and J. Han, "DHSNet: Deep hierarchical saliency network for salient object detection," in *CVPR*, 2016.

[29] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *ICCV*, 2015, pp. 1520–1528.

[30] K. Lim, W.-D. Jang, and C.-S. Kim, "Background subtraction using encoder-decoder structured convolutional neural network," *AVSS*, 2017.

[31] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *CVPR*, 2005.

[32] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015, pp. 448–456.

[33] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*. Springer, 2015, pp. 234–241.

[34] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACM Multimedia*, 2014, pp. 675–678.

[35] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "ImageNet: a large-scale hierarchical image database," in *CVPR*, 2009.

[36] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.