An Application of Deep Learning in Remote Sensing: Automatic Change Detection in Urban Area

Chun Yang*, Bowen Zhou
Hangzhou Gauss-Humboldt Technology Ltd., Hangzhou, China; chun.yang@gauss-h.com (C.Y.), bowen.zhou@gauss-h.com (B.Z.)

*Corresponding author's email: chun.yang@gauss-h.com

Abstract. Change detection of earth’s appearance is a core task for urban management, whose main goal is to identify the change of physical materials via comparing remote sensed images (satellite images, aerial images etc.) of many time epochs. For instance, due to urbanization bare soil can be changed to building; river or lake can be filled up by soil for greening purpose. Recently, deep learning based methods for change detection have been widely applied. The standard pipeline here is: given two images of two epochs and the corresponding changed binary labels, they are fed into a deep learning model to learn the change incidents. However, this kind of change detection focuses solely separating change from no-change, yet ignores the information of “from-to”. For example, the change from soil to building is different to the change from water to soil, but in the standard pipeline they are taken as the same changed incident. In this paper, we propose a deep learning method to tackle this problem, i.e. not only detecting the change incidents, but also predict the change types (the “from-to” information). Our methods are evaluated on the competition dataset released by the SenseTime, and achieve promising detection results i.e. >70% in terms of OA.

Keywords. deep learning, change detection, remote sensing, urban management

1. Introduction
Change detection is one of the most important remote sensing tasks, because it delivers useful information for applications such as urban management, ecosystem monitoring, continental and national land use map update etc. Remotely sensed data such as satellite images play important role on assessing the changes which can be quite specific regarding to their spectral, temporal and spatial characteristics. Thus, there are many various methods to detect changes [1]. Moreover, nowadays more and more new satellites are designed and launched into space to monitor the earth. As a consequence, the automation of processing the captured images is desirable, leading the development of new algorithms. In 2012, Krizhevsky et al. [2] introduced AlexNet, a large and deep convolutional neural network that won the 2012 ImageNet Large-Scale Visual Recognition Challenge (ILSVRC). Since that time, deep learning techniques have been widely applied for various tasks such as image classification [3,4], image retrieval [5-7] semantic segmentation [8-10], object detection [11,12] etc. In remote sensing field, deep learning based methods have been gained more and more attentions as well.
With respect to change detection specifically, many deep learning based methods have been aiming at the definition and the development of completely automatic processing chains, so that the applications become fully operational in both regional and global scale [1]. In this paper, change detection is similar to semantic segmentation where each image pixel is assigned with a meaningful label (e.g. building, tree etc.). Thus, the naïve approach is to semantically segment the bi-temporal image sequences separately, and then perform image differencing (e.g. subtraction of the label maps) techniques to gain the changes [13]. Here, the methods for semantic segmentation can be either patch-based approach [14] or fully convolutional approach [15]. However, the naïve approach ignores the intrinsic relationships between the two images, leading to the developments of complex models to learn joint representations of the input sequences. For instance, reference [16] presents a model based on deep belief networks to learn joint features of two images, and the changes are detected by image differencing based on these representations; reference [17] makes use of an end-to-end recurrent neural network (RNN) to solve the multi/hyperspectral change detection task, due to the fact that RNN is quite suitable of processing sequential data. Their framework employ a RNN based long short-term memory to lean the joint spectral feature representations from bi-temporal image sequences. Furthermore, the authors show that their methods can also tackle problems of multiclass changes; reference [18] also introduces a RNN based approach to detect annual urban dynamics of four cities (Beijing, New York, Melbourne and Munich), yet ranging from a large temporal scale (1984-2016) on the basis of Landsat data.

Although RNNs are good on processing sequential data, they may face the problem of vanishing gradients during training, thus, they are possible to be stuck to a sub-optimal point. In this paper, we propose a method solely relying on convolutional neural networks (CNN) to detect changes: the bi-temporal images are feed into the encoder-decoder based CNN, and there are two separate encoders to extract features; afterwards, the features are fused to send to the decoder. At the end of decoder, there are three convolution blocks to convert the learned representations for specific tasks, i.e. two blocks are for semantic segmentation of each input image, and one block for detecting changes. Here, other than separating change from no-change (corresponding to binary labels) we are learning the information of “from-to”. In summary, the scientific contributions are:

- We propose a CNN-based method for semantic segmentation, change detection simultaneously. In the detection of changes, we predict the “from-to” information as well.
- We conduct experiments in the competition dataset released by Sense Time, and highlight the benefits of our method, as well as investigate its limits.

The remainder of this paper is structured as follows. We present our change detection approach in section 2, and experimental results in section 3. In the end, we give conclusions and a short outlook for future work.

2. Methodology

In this section, we present our CNN for change detection. Our network is on the basis of UNet [19] and SegNet [20] and extended to incorporate two inputs with a new designed skip-connections.

2.1. Backbone network

We refer our network to as DualNet (see Figure 1), which needs two bi-temporal images located in the same place, each of size 256 x 256 pixels. Each image is separately encoded by an encoder, resulting in an encoded representation. Subsequently, the two encoded representations are concatenated together and sent to a common decoder.

In our method, a convolution block (CB) consists of a convolution layer, batch normalization [21] and rectified linear unit in a series. Each encoder branch has 3 such CBs to learn representations. Max-pooling is appended to the end of each CB for down-sampling purpose. The decoder is symmetrical to encoder by upsampling the encoded features layer by layer. In total, there are four convolution blocks, each starting with an upsampling layer via bilinear interpolation. Afterwards, three convolution layers are followed. In addition, there are skip-connections between symmetric convolution blocks from 2 to 4, which is explained in detail in section 2.2. To predict the semantic categories of each input and the
changed information (i.e. the “from-to”), after the last upsampling, three individual convolutional representations are created.

A convolution layer with 1 x 1 kernel is employed to convert each final representation to a class score vector of size $C$ for each image pixel (thus, $H \times W$ pixels totally). Here, $C$ corresponds the number of to be discerned classes, and obviously its value depends on the corresponded task. For instance, in this paper there are 7 semantic classes and 31 “from-to” classes to be differentiated. We now take $C$ as a general class number for further explanation. For each pixel in input image a class score vector $\mathbf{z}^i = (z^i_1, \ldots, z^i_C)^T$ is generated where $z^i_c$ is the class score of pixel $i$ while it is assigned class $c \in C$. Obtaining probabilistic scores $P_I(c|x)$ is done by normalizing them with a softmax function:

$$P_I(c|x) = \frac{\exp(z^i_c)}{\sum_{l=1}^C \exp(z^i_l)}, \quad (1)$$

Where $x$ is the given image data. To train this network, we adapt the focal loss [22] and redesign it so as to be suitable for multi-class classification:

$$L = - \frac{1}{W \cdot H \cdot N} \sum_{i,c} (\gamma^i_c (1 - P_I(c|X))^\varepsilon \cdot \log(P_I(c|X))), \quad (2)$$

Where $X$ is image the mini-batch which has $N$ images in total. $\gamma^i_c$ takes value of 1 if the groundtruth label of pixel $i$ is class $c$ and 0 otherwise. $\varepsilon$ is hyper-parameter to control the penalty term and is set to 1 in our experiments.

2.2. Skip-connections

Normally, the skip-connections is to add feature maps of encoder and decoder directly (such as in FCN [15]) or concatenated and then followed by convolutions (such as in U-Net [19]). For a position in a feature map, the features of its neighbors (i.e. context feature values) should have a great impact on its feature value. However, either addition or concatenation of feature maps may ignore this fact. Thus, we propose an alternative way to skip-connect feature maps, termed as context-skip.
Figure 2. Skip-connections applied in this paper: context-skip. The colours of each feature is corresponded to Figure 1.

Figure 2 shows the mechanism of the context-skip. We first apply depth-wise convolution to aggregate context information inside a small neighborhood for the features from the encoder branches. In this paper, we take a window of 5 x 5 as neighborhood. Afterwards, the convolved features are concatenated together with decoder features and convolved by a 1 x 1 kernel for dimension reduction.

2.3. Implementation detail
Our CNN is implemented based on tensorflow framework. It is trained with stochastic gradient descent (SGD) with base learning rate of 0.01 for total 10 epochs. The learning rate is decreased to 0.001 after 5 epochs. We also use GPU TitanX (12GB Memory) for accelerating training and inference. The mini batch size is set as 4.

3. Test data and test setup

Test Data: the data used for evaluation is from the change detection competition released by SenseTime (https://rs.sensetime.com/competition/index.html#/info). Totally, there are 4662 RGB image pairs, each pair is from two time epochs. The image resolution is 512 x 512 pixels. The competition committee has already separated 2968 image pairs for training (with groundtruth), 847 image pairs for public evaluation and 847 image pairs for private evaluation. There is no groundtruth for evaluation pairs. The goal of this competition is to detect changes among six semantic classes (i.e. the “from-to” information): water, ground, low vegetation, tree, building and sport square. Except the detection of changes, it is also required to predict the semantic classes of each image separately. Figure 3 shows two examples of image pairs.

Test Setup: Because there is no groundtruth released currently, we randomly select 85% image pairs for training and the rest ones for validation and test, whereas 100 image pairs are randomly picked for testing. In addition, we also perform data augmentation strategy so as to avoid overfitting. For each image pair, we perform rotations of 90°, 180° and 270°, resulting in 4 times more data. Subsequently, each image pair are randomly rotated of a degree in the interval of 3° to 20°; random vertical and horizontal flipping; scaling with a random factor from [0.75, 1.25]; random brightness, contrast, conversion into HSV space etc. Because our network requires input size of 256 x 256 pixels, random cropping of such size is performed before feed into CNN.
To obtain the “from-to” classes, first we give an integer number for each semantic class, based on which we create a conversion matrix. For each conversion we assign a unique class number, stated in Table 1. In the end, there are 31 “from-to” classes to be differentiated.

Table 1. Conversion matrix from one semantic class to another class. non: no changed area; -: non-existed conversion.

| class   | non | water | ground | low-veg. | tree | build. | sport |
|---------|-----|-------|--------|----------|------|--------|-------|
| non     | 0   | 0     | -      | -        | -    | -      | -     |
| water   | 1   | -     | 0      | 1        | 2    | 3      | 4     |
| ground  | 2   | -     | 6      | 0        | 7    | 8      | 9     |
| low-veg.| 3   | -     | 11     | 12       | 0    | 13     | 14    |
| tree    | 4   | -     | 16     | 17       | 18   | 0      | 19    |
| building| 5   | -     | 21     | 22       | 23   | 24     | 0     |
| sport   | 6   | -     | 26     | 27       | 28   | 29     | 30    |

The evaluation quality metrics we used are the overall accuracy (OA) and the pixel-wise F1 score, both in pixel level.

4. Experiments
We evaluate the results with respect to two aspects. First, we show the semantic segmentation of each image along with the evaluation results; second, we show the results of change detection by giving the “from-to” changed information.

4.1. Semantic segmentation
Table 2 presents the evaluation of semantic segmentation results of the 100 test image pairs.

Table 2. Evaluation of semantic segmentation.

| Image            | OA [%] | mF1 [%] |
|------------------|--------|---------|
| image of time epoch 1 | 72.7   | 38.3    |
| image of time epoch 2  | 72.2   | 34.6    |

The low evaluated values in terms of OA and mF1 score are caused by the released dataset itself: in the dataset, all unchanged areas are labeled as 0, yet in such areas they contain different physical materials which confuse the labeled ones. Figure 4 shows an example and our prediction.
Figure 4. Our prediction of one image. White represents unchanged area. In the predicted image, we have already rendered the non-changed areas as white. However, there is still large area detected as changed. Please refer to Fig. 3 for colour bars.

Therefore, it is hard for the network to obtain a clean semantic segmentation while inter-classes are confused.

4.2. Change detection

Now, turning our focus on the change detection. As Table 1 states, there are totally 31 changing types (“from-to” information), where class 0 represents unchanged. The evaluation of change detection shows that we got 71% OA whereas most contribution is from unchanged areas. For other “from-to” classes, most of them are pretty falsely identified or totally not detected. The reason is that most of “from-to” classes have very small amount of samples (even zero) for training, resulting in an extreme problem of imbalanced classes. Figure 5 two examples of change detection. Although there are mis-detected changes, the real changed areas are quite well detected.

Figure 5. Two examples of change detection. White: unchanged; other colours are randomly generated to represent one class.

5. Conclusion

In this paper, we have shown an application of deep learning in remote sensing to detect changes in urban area. Here, we are not only focusing on detecting changes, but also interest on the semantic meaning in the original input images. The proposed CNN-based method can tackle this problem. However, in the evaluation the results are not quite satisfied. One big reason is that the released data contain much confused information. In next step, we are going to develop methods or strategies to fix this issue, and also find ways to propagate information among semantic scores and change detection scores in the network.

6. References

[1] Thonfeld F, Hecheltjen A, Braun M, Menz G 2010 From algorithms to processing chains: a review of land cover and land use change detection methodologies https://www.zfl.uni-bonn.de/thonfeld_2010_esa_bergen
[2] Krizhevsky A, Sutskever I, Hinton G 2012 Imagenet classification with deep convolutional neural networks NIPS 2012 pp 1097-1115

[3] He K, Zhang X, Ren S, Sun J 2016 Deep residual learning for image recognition CVPR 770-778

[4] Szegedy C, Liu W, Jia Y.Q, et al. 2014 Going deeper with convolutions arXiv 1409.4842

[5] Alzubi A, Amira A, Ramzan N 2015 Semantic content-based image retrieval: A comprehensive study J Vis Commun Image Represent 32 pp 20-54

[6] Lin Z, Ding G, Hu M, Wang J 2015 Semantics-preserving hashing for cross-view retrieval CVPR pp 3864-3872

[7] Kim G, Moon S, Sigal L 2015 Ranking and retrieval of image sequences from multiple paragraph queries CVPR pp 1993-2001

[8] Audebert N, Saux B. L., Lefevre S, 2018 Beyond RGB: Very high resolution urban remote sensing with multimodal deep networks ISPRS J. Photogramm Remote Sens. 140 pp 20-32

[9] Maggiori E, Tarabalka Y, Charpiat G, Alliez P 2017 High-resolution semantic labelling with convolutional neural networks IEEE Trans Geosci Remote Sens 55(12) pp 7092-7103

[10] Zhao H.S, Shi J.P, Qi X.J, Wang X.G, Jia J.Y 2017 Pyramid scene parsing network CVPR pp 2881-2890

[11] Ren S, He K, Girshick R, Sun J 2015 Faster R-CNN: towards real-time object detection with region proposal networks NIPS 28 pp 91-99

[12] Redmon J. Divvala S, Girshick R, Farhadi A 2015 You only look once: unified, real-time object detection arXiv 1506.02640

[13] Längkvist M, Kiselev A, Alirezaie M, Loutfi A 2016 Classification and segmentation of satellite ortho-imagery using convolutional neural networks Remote Sensing 8(4) pp 329-350

[14] Paisitkriangkrai S, Sherrah J, Janney P, Hengel A 2016 Semantic labeling of aerial and satellite imagery IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 7(9) pp 1-14

[15] Long J, Shelhamer E, Darrell T 2014 Fully convolutional networks for semantic segmentation CVPR pp 3431-3440

[16] Gong M, Zhan T, Zhang P, Miao Q 2017 Superpixel-based difference representation learning for change detection in multispectral remote sensing images IEEE Trans. Geosci. Remote Sens. 55(5) pp 2658-2673

[17] Lyu H, Lu H, Mou L 2016 Learning a transferable change rule from a recurrent neural network for land cover change detection Remote Sens. 8(6) p 506

[18] Lyu H, Lu H, Mou L, Li W, Li X, Wang J, Zhu X. X., Yu L, Gong P 2016 A deep information based transfer learning method to detect annual urban dynamics of four developed cities from 1984–2016 by Landsat data Remote Sens. Environment

[19] Ronneberger O, Fischer P, Brox T 2015 U-Net: Convolutional networks for biomedical image segmentation MICCAI 2015 pp 234-241

[20] Badrinarayanan V, Kendall A, Cipolla R 2017 SegNet: A deep convolutional encoder-decoder architecture for image segmentation PAMI 39(12) pp 2481-2495

[21] Ioffe S, Szegedy C 2015 Batch Normalization: accelerating deep network training by reducing internal covariate shift ICML pp 448-456
[22] Lin T.Y., Goyal P, Girshick R, He K, Dollar P 2017 Focal loss for dense object detection ICCV pp 2999-3007