Deep learning based change detection using UNet++ architecture on VHR images

Mohammad Salmani, Reza Shah-Hosseini *
School of Surveying and Geospatial Engineering, College of Engineering, University of Tehran, Tehran, Iran

Abstract. Change detection is now one of the most important duties in remote sensing science. Over time it has been proven that deep-learning-based methods are one of the most effective ways to identify meaningful changes in a pair of co-registered images. In this project, we are using an end-to-end CD method based on an effective encoder-decoder architecture for binary change detection named UNet++, where change maps could be learned from scratch using a VHR dataset called LEVIR-CD+. UNet++ has been an improved version of UNet that uses a series of nested and dense skip connections, rather than only connections between encoder and decoder networks. In this research, it has been proven that the more we use nested outputs in the result, the better the results can get.

Keywords: remote sensing, change detection, UNet++, deep learning, nest network

E-mail: salmani.mohammad@ut.ac.ir

1 Introduction

With the rapid development in the technology of Remote Sensing (RS), many sensors can capture high quallity images. These photographs include aerial images, satellite images, etc. Addintionally by using new technologies, it is now possible to prepare datasets of bi-temporal images taken from a particular spot. The enhancements has enabled us to perform change detection (CD) process using different datasets and algorithms. CD is the process of identifying meaningful changes that appear in a pair of images taken from a location. In this paper, we will design a deep learning method to detect changes.

One of the main issues in change detection is the accuracy of attaining the changes. Because of that, in the past decades, a large number of CD strategies have been developed. These methods are usually based on deep convolutional networks. In the past few years, deep learning (DL) methods have achieved dominant advantages over traditional methods in the fields of image analysis, especially in change detection problems. The stated fact is the reason why a lot of RS

problems have been shifted towards DL methods. However using DL based methods can lead to some issues that can decrease our model's accuracy.

This is why we are going to use a novel change detection method. To address above-mentioned DL method's issues, we proposed a novel end-to-end method based on improved UNet++, which is a robust encoder-decoder architecture for semantic segmentation.

2 Dataset

As we know, There has been a great development in the quantity and quality of the datasets that can be used in DL based change detection methods. However, these existing datasets have many drawbacks. Firstly, all these datasets do not have enough data for supporting most deep-learning-based CD algorithms, which are tending to suffer overfitting problems when the data quantity is much to scare for the number of the model parameters. Secondly, these CD datasets have low image resolution, which blurs the contour of the change targets and brings ambiguity to annotated images.

We will use a large-scale remote sensing binary change detection dataset called LEVIR-CD+. The dataset consists of 637 very high-resolution (VHR, 0.5m/pixel) Google Earth (GE) image patch pairs with the size of 1024 × 1024 pixels. These bitemporal images with time span of 5 to 14 years have significant land-use changes, especially the construction growth. LEVIR-CD covers various types of buildings, such as villa residences, tall apartments, small garages and large warehouses. Here, we focus on building-related changes, including the building growth (the change from soil/grass/hardened ground or building under construction to new build-up regions) and the building decline. A few samples of LEVIR-CD dataset is illustrated in Figure 1.

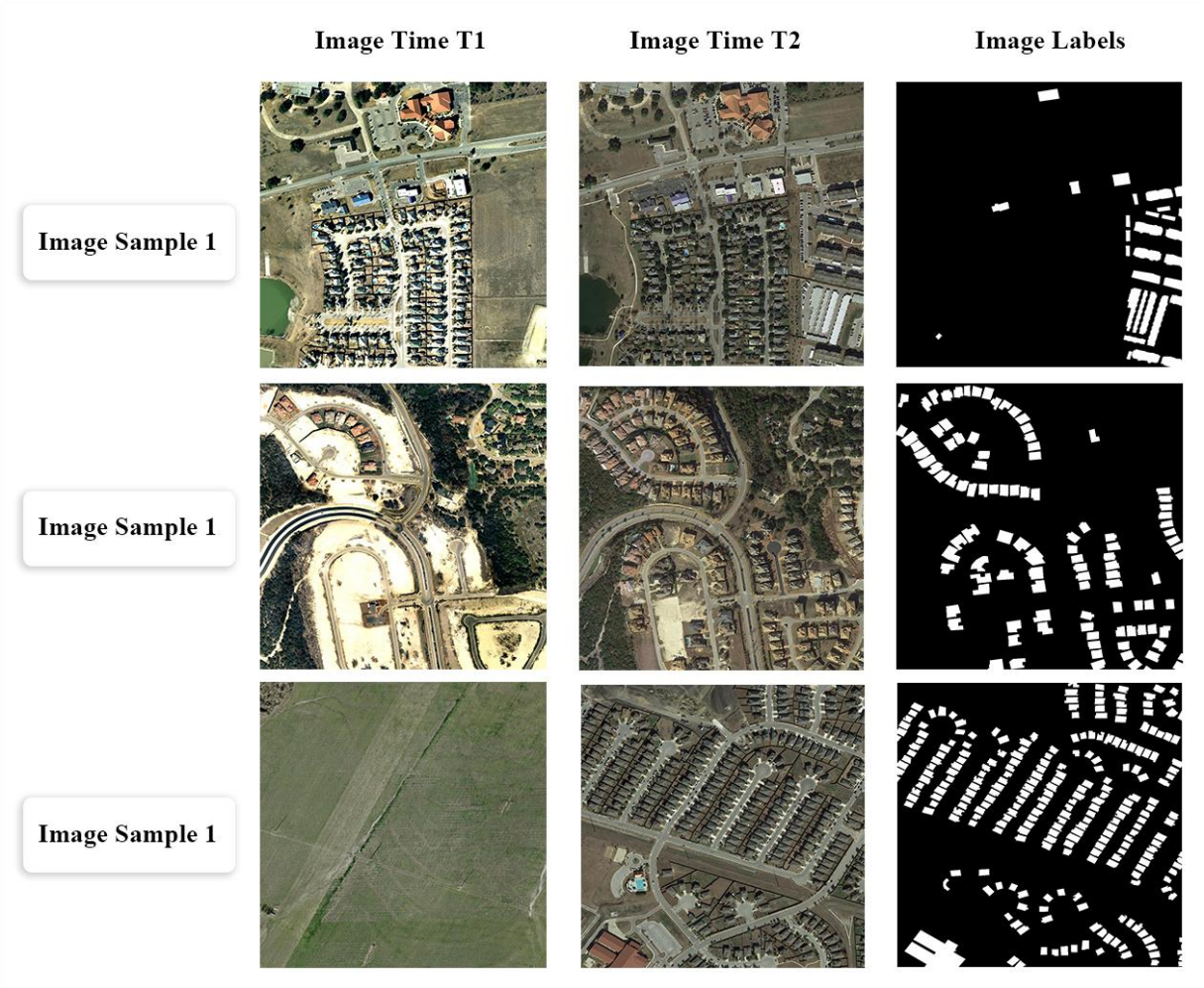

Figure 1. Samples of the dataset

3 Methodology

In this section, we are going to describe the data-pre-processing stage that we performed to prepare our dataset for training the model first. Then we will introduce the network architecture and the methods we used to achieve a change map of the CD process.

3.1 Data pre-processing

As we mentioned above, the dataset we are using is called Levir-CD that includes 637 pairs of images taken from many different locations. The images in the dataset has the resolution of 1024

× 1024 pixels. However using the original resolution needs a robust hardware and can take a lot of time. So we decided to decrease the resolution of images to 256 × 256 pixels.

The next step is to concatenate images. As we know bitemporal images are taken from a specified area in two different times. We need to concatenate the images taken in time T1 and time T2. Considering the images are in RGB format, the input image of our U-net++ network will be in the shape of 256 × 256 × 6.

3.2 Network architecture

Having the pre-processed images and ground truth images in hand, now we can train our network. As we mentioned above, we are using an improved version of U-net++ architecture. This architecture is based on convolutional neural networks (CNN) and is a superb method for CD tasks of VHR images. The flowchart of the proposed method is illustrated in Figure 2.

In the last section, two periods of images are concatenated along the 3rd axis as the input of the network. Concatenation of these images has been proven to be an adequate method for binary change detection images in training the network. In the architecture of our UNet++ network, we have used 2D convolutional layers. Additionally, downsampling and upsampling layers are used. For the Downsampling part of the method, we have used MaxPooling2D, which is one of the most common functions in deep learning projects.

Similarly, for the Up-sampling part of the architecture, we used the Conv2DTranspose function. The convolution unit illustrated in Figure 3 consists of a convolutional layer followed by a Batch Normalization unit to avoid overfitting. Then another convolutional layer is applied to the unit.

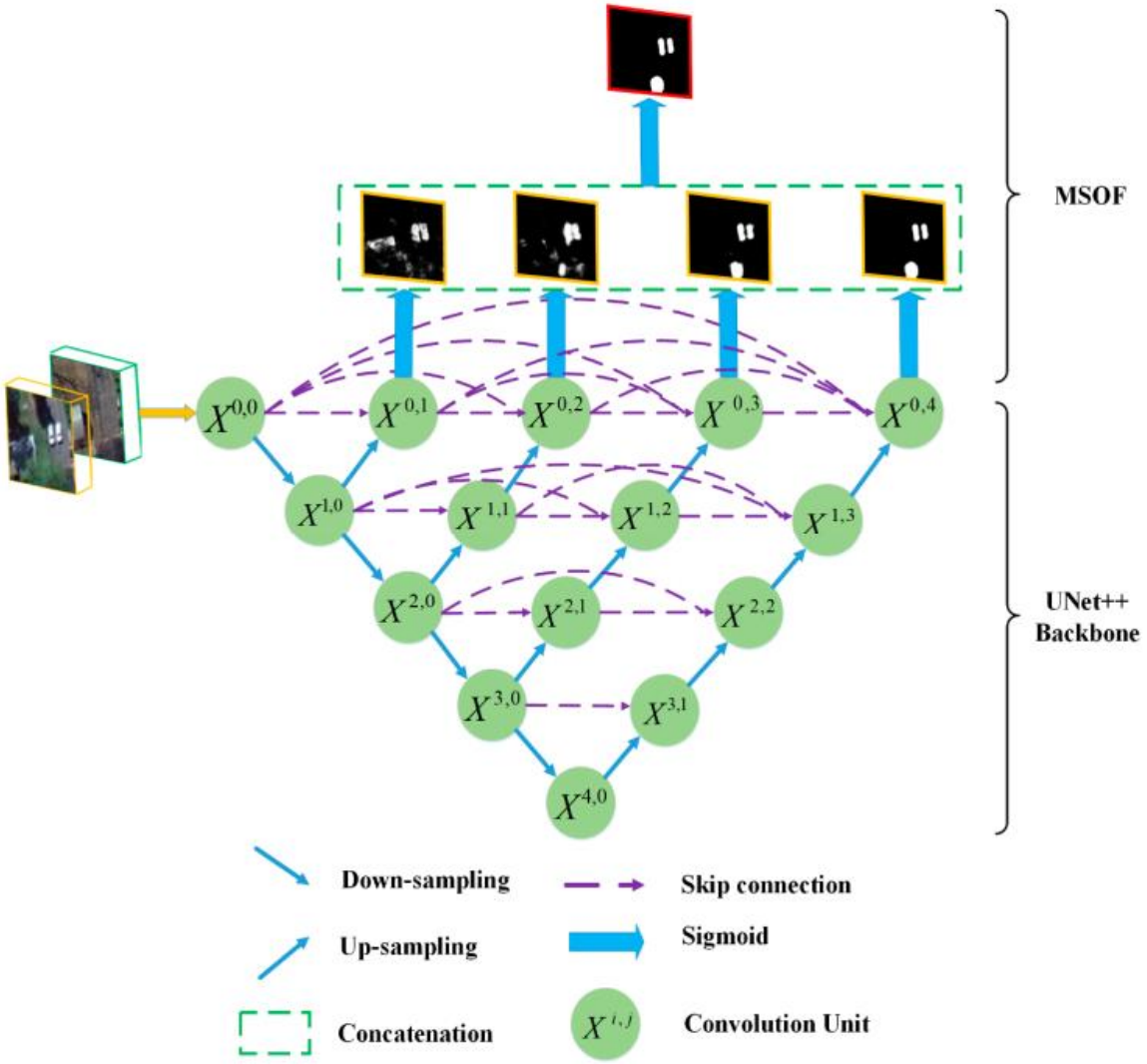

Figure 2. illustraion of UNet++ architecture

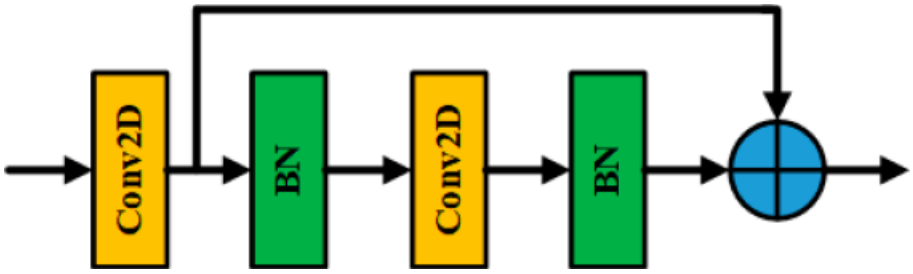

Figure 3. Convolution unit

With the pre-processed and label images in hand, we can now train our network. As mentioned above, we are using an improved version of U-net++ architecture. This architecture is based on

convolutional neural networks (CNN) and is a superb method for CD tasks of VHR images. The flowchart of the proposed method is illustrated in Figure 2.

In the last section, two periods of images are concatenated along the 3rd axis as the input of the network. Concatenation of these images has been proven to be an adequate method for binary change detection images in training the network. In the architecture of our UNet++ network, we have used 2D convolutional layers. Additionally, downsampling and upsampling layers are used. For the downsampling part of the method, we have used MaxPooling2D, which is one of the most common functions in deep learning projects.

Similarly, for the Up-sampling part of the architecture, we used the Conv2DTranspose function. The convolution unit illustrated in Figure 1 consists of a convolutional layer followed by a Batch Normalization unit to avoid overfitting. Then another convolutional layer is applied to the unit. The diagram of the convolution unit is illustrated in Figure 3.

4 Results and discussion

As you see in Figure 2, There are four different outputs in the network. The network is trained in two epochs. Each epoch has a batch size of 50. The result of the training of the network is illustrated in Figure 4.

```
Epoch 1/2
128/128 [==============================] - 2639s 21s/step - loss: 0.8689 - output_1_loss: 0.2602
- output_2_loss: 0.1007 - output_3_loss: 0.0475 - output_4_loss: 0.0308 - output_1_accuracy:
0.8930 - output_2_accuracy: 0.9592 - output_3_accuracy: 0.9813 - output_4_accuracy: 0.9877 -
output_5_accuracy: 0.1420
Epoch 2/2
128/128 [==============================] - 2571s 20s/step - loss: 0.4817 - output_1_loss: 0.0058
- output_2_loss: 0.0017 - output_3_loss: 0.0018 - output_4_loss: 0.0023 - output_1_accuracy:
0.9998 - output_2_accuracy: 0.9998 - output_3_accuracy: 0.9998 - output_4_accuracy: 0.9998 -
output_5_accuracy: 0.1739
```

Figure 4. Result of training the network

The first output denoting $X^{0,1}$, has been computed by the use of $X^{0,0}$ and $X^{1,0}$. $X^{0,0}$ is skip-connected to the output and an upsampling function has been applied to $X^{1,0}$ to produce the first output. In the end, a sigmoid function is applied to result in a binary change map. This part of the output attained an accuracy of 89.3% in the first epoch and 99.9% accuracy in the second epoch of the training.

The second output is denoted as $X^{0,2}$, and consists of three input nodes. The first node is the skip-connected node of $X^{0,0}$ and similarly the second one is the skip-connected node of $X^{0,1}$. The last input node of the output is the up sampled output of the node named $X^{1,1}$. The accuracy metric of the second output has resulted in 95.9% in the first epoch.

The third output of the UNet++ network is the convolution applied node of $X^{0,3}$. Similar to the other outputs, this output is the combination of the last outputs and the up-sampled result of the node $X^{1,2}$. This output has resulted in better accuracy compared to the last outputs. The accuracy metric for this method is 98.13%.

Finally, the last output is $X^{0,3}$. This node has been achieved through skip-connection of the last three outputs and then concatenating with the $X^{1,3}$ node. The $X^{1,3}$ itself is resulted by applying an up sampling function such as transposed convolution. This output resulted in the best accuracy among other methods. The accuracy, related to this output is 98.77%. The quantitative results for the above-mentioned model are shown in Table 1.

Figure 5. Quantitative results of four different outputs.

Output	**Accuracy (%)**
Output #1 $X^{0,1}$	89.3
Output #2 $X^{0,2}$	95.9
Output #3 $X^{0,3}$	98.13

Output #4 $X^{0,4}$ 98.77

Unlike UNet architecture, UNet++ uses a series of nested and dense skip pathways, rather than only connections between encoder and decoder networks. The UNet++ architecture possesses the advantages of capturing fine-grained details, thereby generating better segmentation results than UNet. Therefore, it is promising to exploit the potential of UNet++ for semantic segmentation on RS images. As we can see in the results, The more nested and dense skip pathways in the output, the better the results can get.

5 Conclusions

Deep learning methods have been widely used in remote sensing tasks especially change detection. In this project we used a deep learning method for a change detection task. UNet++ is an improved architecture of UNet method that uses nested convolution units and dense skip-connections. The UNet++ architecture has been proven to be an effective method in analyzing VHR images.

References

1. Daifeng Peng; Yongjun Zhang; Haiyan Guan. End-to-End Change Detection for High Resolution Satellite Images Using Improved UNet++. Nanjing, China, 2019.
2. Zhou, Z.; Siddiquee, M.M.R.; Tajbakhsh, N.; Liang, J. Unet++: A nested u-net architecture for medical image segmentation. In Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support; Springer: Cham, Switzerland, 2018.

3. Volpi, M.; Tuia, D.; Bovolo, F.; Kanevski, M.; Bruzzone, L. Supervised change detection in VHR images using contextual information and support vector machines. Int. J. Appl. Earth Obs. Geoinform, 2013.
4. Hao Chen, Zhenwei Shi. A Spatial-Temporal Attention-Based Method and a New Dataset for Remote Sensing Image Change Detection. Beihang University, Beijing, China, 2020.
5. E. Bousias Alexakis, C. Armenakis. Evaluation of UNet and UNet++ architectures in high resolution image change detection applications. York University, Toronto, Canada, 2020.