

# Self-Pair: Synthesizing Changes from Single Source for Object Change Detection in Remote Sensing Imagery

Minseok Seo, Hakjin Lee, Yongjin Jeon, Junghoon Seo

SI Analytics

70, Yuseong-daero 1689beon-gil, Yuseong-gu, Daejeon, Republic of Korea

{minseok.seo, hakjinlee, yongjin117, jhseo}@si-analytics.ai

## 1. Experimental Settings

We conduct experiments on a single NVIDIA RTX 8000 for all models and training settings. Note that Self-Pair’s photometric augmentation was used to randomly apply only one of RandomBrightnessContrast, RandomToneCurve, ColorJitter, and FDA (Fourier Domain Adaptation) of albumentations<sup>1</sup>.

### 1.1. SNUNet-CD

We experimented using the official SNUNet-CD code<sup>2</sup> provided by authors. Like the authors of the SNUNet-CD paper, we set the batch size to 16 per gpu and used the Adam optimizer. The learning rate was set to  $1e-3$  and decreased by 0.5 every 8 epochs. The model was trained up to 100 epochs.

### 1.2. BIT-CD

We experimented using the official BIT-CD code<sup>3</sup>. To train a ChangeStar model we have 40k iterations with a poly learning rate policy, where the initial learning rate was set to 0.03 and multiplied by  $(1 - \frac{step}{max\_step})^\gamma$  with  $\gamma = 0.9$ . We used SGD as the optimizer on a single RTX 8000 GPU with a minibatch of 16 images, weight decay of 0.0001 and momentum of 0.9. During training, data augmentation such as horizontal and vertical flip, rotation of 90-k ( $k = 1, 2, 3$ ) degree, and scale jitter was applied.

### 1.3. ChangeStar

We experimented using the official ChangeStar code<sup>4</sup>. Unless otherwise specified, ChangeStar were trained for 40k iterations with a poly learning rate policy, where the initial learning rate was set to 0.03 and multiplied by  $(1 - \frac{step}{max\_step})^\gamma$  with  $\gamma = 0.9$ . We used SGD as the optimizer on single RTX 8000 GPU with a minibatch of 16

images, weight decay of 0.0001 and momentum of 0.9. For training data augmentation, after horizontal and vertical flip, rotation of 90-k ( $k = 1, 2, 3$ ) degree, and scale jitter, the images are then randomly cropped into 512×512 pixels for xView2 pre-disaster dataset and 256×256 pixels for SpaceNet2 dataset.

## 2. Comparison within ChangeStar Benchmarks

The main table of this paper shows performance on mean IoU and mean F1, which are the average metric of each class for the change category and the un-change category. For example, mean IoU can be expressed as:

$$mIoU = \frac{1}{K} \sum_{k=1}^K \frac{\text{area of overlap on class } k}{\text{area of union on class } k}, \quad (1)$$

where  $K$  is 2, which represents the change category and the un-change category for our problem. Including performance over the un-change category is important in change detection so as to decrease false positives on the change category.

However, ChangeStar shows the performance for the change category only, which can be expressed as:

$$IoU = \frac{\text{area of overlap on the change}}{\text{area of union on the change}} \quad (2)$$

Here, we provide the performance with regard to only changed areas in order to compare with ChangeStar’s method fairly. In this experiment, we followed most of the hyperparameters provided in the official ChangeStar implementation. We used same settings in the main table for *Self-Pair* as shown in the Tab 1, our proposed method outperforms PCC baselines as ChangeStar method does. Moreover, *Self-Pair* achieves superior performance to the ChangeStar method across all datasets and models. It can be seen that the better the baseline model achieves, the bigger the performance gap between ChangeStar and *Self-Pair*.

<sup>1</sup><https://albumentations.ai/>

<sup>2</sup><https://github.com/licyoo/Siam-NestedUNet>

<sup>3</sup>[https://github.com/justchenhao/BIT\\_CD](https://github.com/justchenhao/BIT_CD)

<sup>4</sup><https://github.com/Z-Zheng/ChangeStar>

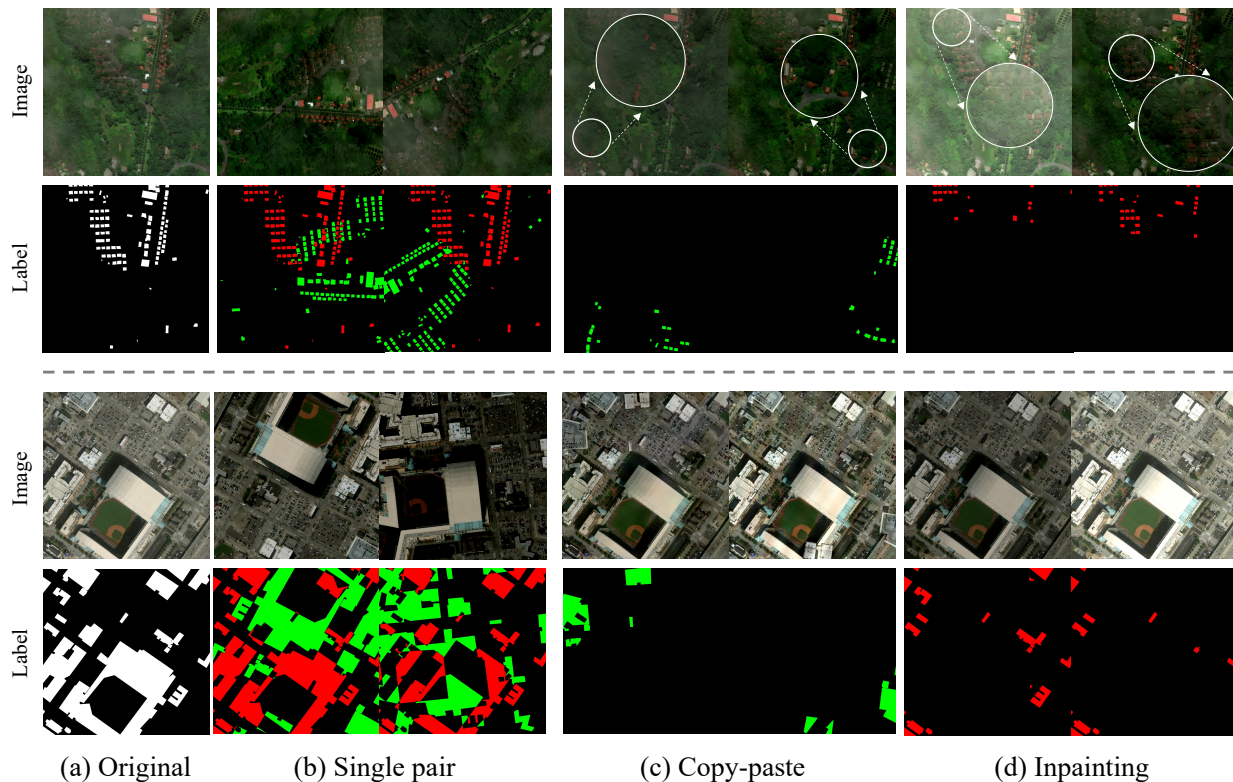


Figure 1. An example of *Self-Pair* that is generated in various ways by randomness even with the same strategy. The original image is  $t_0$ , and the image generated with each strategy is  $t_1$ , and a changed label is generated. Red is an object that existed and disappeared, and green is an object that did not exist and was newly generated.

Table 1. Experimental results that can be analyzed one-to-one with table1 of the ChangeStar paper.

Method	Segmentation Model	Train on xView2 pre-disaster				Train on SpaceNet 2			
		WHU		LEVIR-CD		WHU		LEVIR-CD	
		IoU	F1	IoU	F1	IoU	F1	IoU	F1
PCC	PSPNet	37.46	54.51	55.87	71.69	21.39	35.25	10.19	18.50
ChangeStar	+ ChangeMixin	56.44	72.15	61.63	76.26	25.56	40.72	15.25	26.47
Self-Pair	+ ChangeMixin	<b>57.41</b>	<b>73.35</b>	<b>63.09</b>	<b>77.97</b>	<b>32.34</b>	<b>49.19</b>	<b>24.85</b>	<b>34.63</b>
PCC	DeepLab v3	32.46	49.01	54.77	70.78	33.08	49.72	13.78	24.23
ChangeStar	+ ChangeMixin	56.86	72.49	60.94	75.73	35.57	52.48	15.92	27.46
Self-Pair	+ ChangeMixin	<b>58.10</b>	<b>75.18</b>	<b>63.37</b>	<b>78.29</b>	<b>39.22</b>	<b>56.62</b>	<b>21.39</b>	<b>35.08</b>
PCC	Semantic FPN	38.66	52.68	55.51	71.38	23.90	38.58	9.80	17.28
ChangeStar	+ ChangeMixin	55.37	71.27	65.21	78.94	37.63	54.68	25.86	41.10
Self-Pair	+ ChangeMixin	<b>57.43</b>	<b>74.01</b>	<b>68.40</b>	<b>82.08</b>	<b>40.10</b>	<b>68.26</b>	<b>39.56</b>	<b>56.38</b>
PCC	FarSeg	31.66	48.09	55.09	71.04	27.69	43.37	7.97	14.77
ChangeStar	+ ChangeMixin	58.22	73.59	65.71	79.31	39.02	56.14	30.42	46.65
Self-Pair	+ ChangeMixin	<b>66.44</b>	<b>82.91</b>	<b>68.01</b>	<b>85.94</b>	<b>54.87</b>	<b>71.20</b>	<b>50.39</b>	<b>67.69</b>

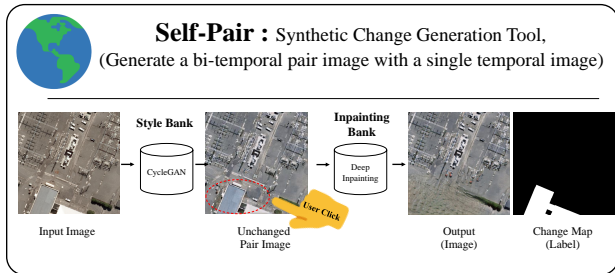


Figure 2. Synthetic change generation tool example.

Compared to the mean IoU and mean F1 with regard to both changes and unchangeds, the performance gap for the change category is smaller. This indicates that proposed method has an advantage in maintaining performance, especially in unchanged areas.

### 3. Example of *Self-Pair*

Figure 1 is an example of an image generated by each strategy of *Self-Pair*. Unlike bi-temporal pair images, *Self-Pair* can control the location and number of changes, and can generate various pair images by randomness. Therefore, *Self-Pair* generates a different pair for each iteration even if the same single image is input as shown in the figure. *Self-Pair* can train more various pair images than a bi-temporal pair in which the changed area is fixed, and thus the generation performance is higher than that of a model trained with a bi-temporal pair.

### 4. Details on the large or small objects

As shown in Fig. 1 of the supplementary, we exclude from copy-and-paste and inpainting candidates if the object size is too large (we find the upper bound of the object size empirically-objects larger than 30% in image size are excluded from inpainting or copy-paste. ).

We wanted to show that each component of *self-pair* is effective even when implemented in the simplest way. For this reason, if the object size is too large, the inpainting method (we used) will have a blurry effect. However, as shown in Fig. 2, the photometric augmentation and inpainting components in *self-pair* can be easily changed in a deep learning-based method Fig. 2, and *self-pair* can be applied even if the size is large.