CHANGE DETECTION BETWEEN DIGITAL SURFACE MODELS FROM AIRBORNE LASER SCANNING AND DENSE IMAGE MATCHING USING CONVOLUTIONAL NEURAL NETWORKS

Z. Zhang [1,*], G. Vosselman [1], M. Gerke [2], C. Persello [1], D. Tuia [3], M. Y. Yang [1]

[1] Dept. of Earth Observation Science, Faculty ITC, University of Twente, The Netherlands - (z.zhang-1, george.vosselman, c.persello, michael.yang)@utwente.nl
[2] Institute of Geodesy and Photogrammetry, Technical University of Brunswick, Germany - m.gerke@tu-bs.de
[3] Wageningen University and Research, The Netherlands - devis.tuia@wur.nl

Commission II, WG II/3

KEY WORDS: Change Detection, Digital Surface Model (DSM), Airborne Laser Scanning, Dense Image Matching, Convolutional Neural Network (CNN)

ABSTRACT:

Airborne photogrammetry and airborne laser scanning are two commonly used technologies used for topographical data acquisition at the city level. Change detection between airborne laser scanning data and photogrammetric data is challenging since the two point clouds show different characteristics. After comparing the two types of point clouds, this paper proposes a feed-forward Convolutional Neural Network (CNN) to detect building changes between them. The motivation from an application point of view is that the multimodal point clouds might be available for different epochs. Our method contains three steps: First, the point clouds and orthoimages are converted to raster images. Second, square patches are cropped from raster images and then fed into CNN for change detection. Finally, the original change map is post-processed with a simple connected component analysis. Experimental results show that the patch-based recall rate reaches 0.8146 and the precision rate reaches 0.7632. Object-based evaluation shows that 74 out of 86 building changes are correctly detected.

1. INTRODUCTION

To make the urban topographical database up-to-date is of vital importance for urban planning and management (Tran et al., 2018). A common data updating process is as follows: new remote sensing data are obtained at the new epoch and then changes are detected between the two epochs. This allows performing updates only where changes have happened. In practice, the two main remote sensing data used for this type of analysis are those issued from airborne laser scanning (ALS) and airborne photogrammetry. It is common that laser scanning data and photogrammetry data are available in different epochs. For example, in some mapping agencies the laser scanning point clouds are available as existing database, while aerial images are acquired as a new data set frequently. This paper aims to detect changes between laser scanning data and photogrammetry data.

Both airborne laser scanning and photogrammetry can generate point clouds, but their principles for point cloud generation are quite different. In airborne laser scanning, the time of flight of laser beam is recorded from emission to reception. The distance from the laser sensor and the object is calculated based on the travel time (Vosselman and Maas, 2010). The objects' location is calculated based on this distance, the instant sensor location and the attitude (i.e. the aircraft position). Finally, 3D coordinates of unordered points are obtained as a main product from laser scanning. In contrast, airborne photogrammetry starts from aerial image acquisition under strict flight control and image quality control. 3D point clouds are obtained from dense image matching (DIM) and forward intersection of corresponding rays. Additionally, 2.5D Digital Surface Models (DSMs) and 2D orthoimages can also be obtained after interpolation and ortho-rectification, respectively (McGlone et al., 2013).

The point clouds from airborne laser scanning and dense matching show different characteristics (Remondino et al. 2014; Nex et al., 2015; Ressl et al 2016; Mandlburger et al., 2017). Fig. 1 illustrates the data differences between laser scanning points and dense matching points. There are no object changes between the two epochs but the two point clouds still differ. The last column shows that a simple DSM differencing between the two data sets leads to many falsely detected changes due to data inaccuracy, noise and data gaps.

- **Accuracy:** The vertical accuracy of laser scanning can reach ±5 cm, while the vertical accuracy of dense matching points from strict quality control and state-of-the-art dense matching algorithms can be better than 1 Ground Sampling Distance (GSD), which is usually 10-20 cm from airborne platforms.
- **Precision:** Generally the point clouds from laser scanning contain less noise than point clouds from dense matching. Dense matching brings mis-matchings when the image contrast is poor, as, for example, in shadowed areas.
- **Data gap level:** In point clouds issued from dense matching, data gaps occur not only due to poor image contrast, but also due to limited visible rays. Therefore, data gaps in dense matching point clouds may occur on narrow streets, water surfaces, tree canopy or under shadow (Zhang et al., 2018). In laser scanning, data gaps occur due to occlusion or pulse absorption by the surface material.

Considering the differences between the two point clouds, it is difficult to detect changes based only on prior knowledge and ad hoc rules. Meanwhile, Convolutional Neural Network (CNN) have shown excellent performance in extracting semantic features and change detection tasks. This paper proposes a CNN-based framework to detect changes between point clouds. First, the point clouds are converted to 2.5D DSMs and then to 2D raster images. Square patches are selected from the raster images. Second, the ALS-DSM patch, the DIM-DSM patch, and the corresponding orthoimage patch (R, G, B) are stacked and fed into a feed-forward CNN for change detection. Finally, the change map is refined with connected component analysis.

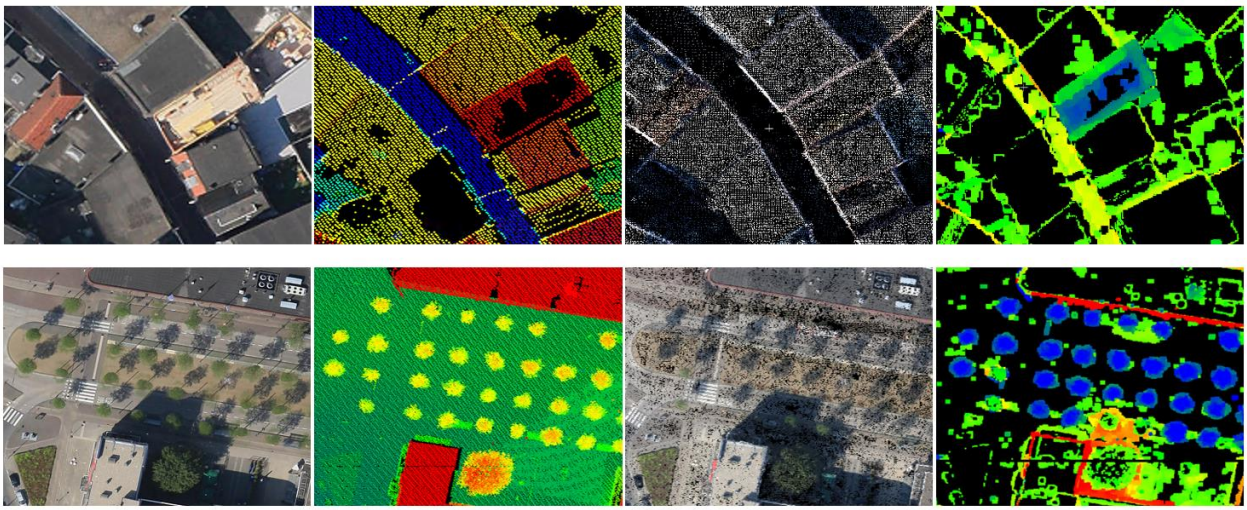

Figure 1. A comparison between laser scanning data and dense matching data. From left to right: orthoimages for reference, laser scanning points colored by height (red colors mean higher points), dense matching points with true colors, DSM differencing colored by height (red colors mean large differences). The top row shows a dense residential area; the bottom row a scene with one tall tree at the middle bottom and thirty short trees.

We apply the proposed method to detect changes between multimodal point clouds obtained over a densely-built urban area. Results show that 74 out of 86 building changes are correctly detected.

The paper is structured as follows. Section 2 reviews related work on point cloud change detection. Section 3 presents our proposed change detection method. Section 4 presents the study area and the experimental settings. Section 5 presents the results and discussions. Section 6 concludes the paper.

2. RELATED WORK

Change detection is the process of defining differences in an object by analyzing it at different epochs (Singh, 1989). The input data of two epochs can be either raw remote sensing data or from an existing database (Qin et al., 2016). Concerning 3D change detection, change detection can be performed either between 3D data or by comparing 3D data of a single epoch to a bi-dimensional map (Vosselman, 2004).

When 3D data are available at both epochs, a point-to-point comparison (also called surface differencing) is widely applied. Surface differencing is used to define the potential change locations, followed by a more accurate post-classification to recognize the specific types of changes (Lu et al., 2004). Basgall et al. (2014) compared laser points and dense matching points with the CloudCompare software. Single building changes were detected by visual inspection. Xu et al. (2015) detected changes on the DSM differencing map using knowledge-based rules. This method involved handcrafted rules which required heavy prior knowledge about the scene. Du et al. (2016) detected building changes in outdated dense matching point clouds using new laser points, which is the reverse setup with the one considered in this paper. Iterative Closest Point (ICP) algorithm was used to register the two point clouds. Height difference and grey-scale similarity were used with contextual information to detect changes in the point cloud space. Finally, the detected changes were refined based on handcrafted features. This framework required to set some thresholds based on prior-knowledge towards the scene.

When raw data are only available for the past epoch, while a map or 3D models are available for the new epoch, a direct point to point comparison or surface differencing is not feasible. Vosselman et al. (2004) detected and updated building changes in a 2D map using laser scanning data. After segmentation and filtering bare earth points, the object points were classified as buildings or vegetation based on surface roughness, segment size, height, color and first-last pulse difference. The building segments were compared with the building objects on 2D maps for change detection.

Olsen (2004) proposed a method to detect building changes in the 3D topographic database TOP10DK using imagery. The data preparation steps included the registration between the map database and the images, the generation of normalized DSM, followed by a training data evaluation. The change map was computed with a pixel-by-pixel comparison between the map database and the classified images. The overall accuracy of change detection was 50%; 45 false alarms were detected (which corresponds to three times the quantity of real changes). Chen and Lin (2010) proposed a method to update 3D building models using new LiDAR points and aerial images. In the change detection process, the height differences between LiDAR points and old polyhedral building models based on facet orientation analysis indicated the major information about changes. The line features on images were used to verify the change detection results. However, such framework cannot detect newly-built buildings. Stal et al. (2013) detected changes between DSMs derived from laser scanning and dense image matching. Their method was based on surface subtraction followed by a series of refinements to remove false detections.

Recently, CNNs show excellent performance in various computer vision tasks, e.g. image classification (Krizhevsky et al., 2012), semantic segmentation (Long et al., 2015; Volpi and Tuia, 2018; Audebert et al., 2018) and object detection (Ren et al., 2015). Compared to traditional classifiers with handcrafted features as inputs, convolutional neural networks learn the features directly from data and have inner hierarchical structures that allow learning features going from low level geometrical characteristics to more semantic features at the bottleneck of the network. Concerning image-based change detection, Mou et al. (2017) identified corresponding patches between SAR images

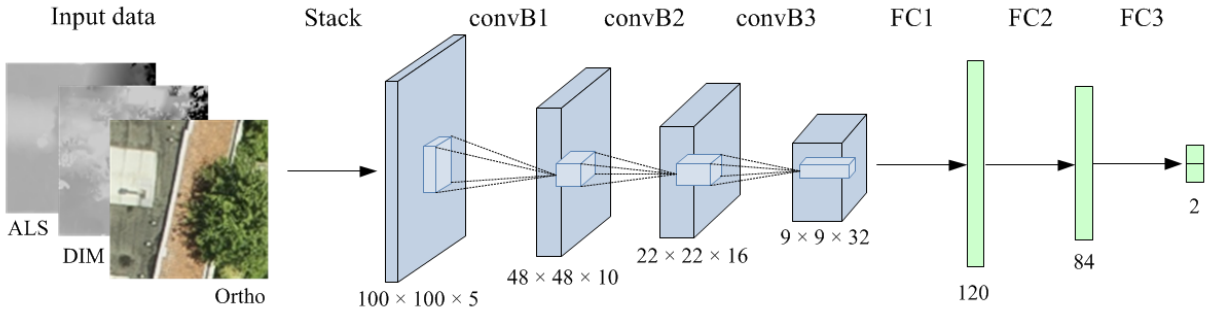

Figure 2. The CNN architecture proposed for multimodal change detection. The top row shows the operations, the parameters below feature maps show the feature map size and number of channels (e.g. 100 × 100 × 5).

and optical images using a pseudo SI-CNN. The feature maps from the two Siamese branches were concatenated, which worked as a patch comparison unit. Zhan et al. (2017) maintained the original input size in each convolutional layer in the two branches followed by a weighted contrastive loss function. The acquired change maps were acquired from threshold segmentation and post-processed by a K-nearest neighbor approach.

In addition, CNN also shows superior performance in extracting distinctive features from point clouds (Hu and Yuan, 2016; Rizaldy et al., 2018). Our change detection framework was inspired by these two papers. We aim at developing an automatic multimodal change detection method for a large urban data set. The proposed method should involve very scarce thresholds to tune.

3. METHOD

Our method includes three steps: First, the multimodal point clouds and orthoimages are converted to raster images. Square patches are cropped from the images as the minimum unit for change detection. Second, the patches are fed into CNN for binary classification with an output of changed or not. Third, the original change map is post-processed with connected component analysis.

3.1 Pre-processing

The DIM data are registered to the ALS data because we did bundle adjustment and quality control in the photogrammetric workflow using Ground Control Points (GCPs). In the pre-processing step, the ALS point cloud, DIM point cloud, and the orthoimage are converted to images. The ALS point cloud is converted following this sequence: *point cloud -> DSM -> raster image -> patches*. The ALS point cloud and DIM point cloud are converted to DSMs using Inverse Distance Weighting (IDW). The two DSMs are normalized to a raster image:

$$H = (H_0 - H_{min})/(H_{max} - H_{min}) \qquad (1)$$

where H_0 is the height of a node in the DSM grid. H_{min} and H_{max} are the minimum and maximum DSM height in the whole study area. After normalization, the values in the raster images range in [0,1]. This representation approach is able to maintain all the height details in the DSMs. In addition, the three channels R, G and B of the orthoimages from dense matching are also normalized to [0, 1] by simply dividing each pixel value by 255. At the end of this stage, all five channels are normalized into [0,1]: ALS-DSM, DIM-DSM, R, G and B.

The building changes are manually labeled on the orthoimage with guidance of ALS points, DIM points and DSM differencing map. When a building is new or heightened, the boundary is delineated from the DIM point clouds; When a building is demolished or lowered, its boundary is delineated from the ALS point cloud. Data gaps and water appear in both laser points and DIM points. When data gaps appear in either epoch, we simply cannot make any prediction about the change. In addition, we are not interested in the water height changes caused by tides. Data gaps and water are marked on the ground truth map and are not considered during change detection. Four types of changes are manually delineated on the ground truth map: *changed building*, *data gap*, *water*, and *other*. Specifically, the *changed building* class includes new, demolished, heightened and lowered buildings. *Other* includes all the irrelevant changes and unchanged areas.

After ground truth delineation, small square patches are cropped from the raster images based on the ground truth. When cropping the patches from images, the ALS-DSM, DIM-DSM and orthoimage patches are strictly registered with each other. A critical question is how to define a changed patch and an unchanged patch. Some previous patch-based classification work assigned the label of the central pixel of a patch to the whole patch (Hu and Yuan 2016; Daudt et al., 2018). However, this definition method is sensitive to slight displacement of the patch. In this paper, we label the patch as changed if the ratio of changed pixels in this patch is larger than a threshold. The rules used for patch labeling are as follows:

- 1. If the ratio of pixels for water and data gaps is larger than T_1, eliminate the patch.
- 2. If the ratio of changed pixels is larger than T_2, the patch is labeled as changed ($T_1 < T_2$); otherwise it is unchanged.

3.2 CNN architecture

The registered three patches with five channels (ALS-DSM, DIM-DSM and orthoimage) are stacked and fed into the CNN for change detection. The proposed CNN architecture is a typical feed-forward architecture as shown in Fig. 2. It contains three convolution blocks, three fully connected layers, and one classifier layer. The network is conceptually similar to AlexNet (Krizhevsky et al., 2012) and the change detection network proposed by (Mou et al., 2017), which has more convolution layers with respect to AlexNet. Our task is easier than theirs. In (Mou et al., 2017), the two patches to be compared are not only from different sensors (SAR and optical), but also involve translation, rotation and scale changes. In our case, the compared patches are strictly registered and normalized to the same scale. Therefore, we use only three convolution blocks for feature extraction.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-IV-2-W5-453-2019 | © Authors 2019. CC BY 4.0 License.

(1) Convolution blocks and fully connected layers
In Fig. 2, the input to the CNN contains 5 channels. The inputs are processed by three convolution blocks consecutively. Each convolution block contains a convolution operation followed by Rectified Linear Unit (ReLU) and max-pooling layers (Goodfellow et al., 2016). The size of convolution kernels is 5 × 5. The padding size is 0 and the sliding is 1. The feature map size decreases by 4 pixels in height and width, respectively after one convolution operation.

Fully connected layers are used in the final stages of the network for high-level reasoning. We use 3 fully connected layers. The last fully connected layer outputs a 2×1 vector, which is corresponding to the non-negative class scores.

(2) Loss function
Suppose that (x_1, x_2) is the 1D vector predicted from the last fully connected layer, the loss is computed between (x_1, x_2) and the ground truth (1 for changed and 0 for unchanged). First, the vector is normalized to (0,1) by a Softmax function:

$$p_i = \frac{\exp(x_i)}{\exp(x_1) + \exp(x_2)}, \qquad i = 1,2 \tag{2}$$

where $p_1 + p_2 = 1$. Then, a weighted binary cross entropy loss is calculated:

$$Loss = -(w_1 y \log(p_1) + w_2(1-y)\log(p_2)) \tag{3}$$

where y is the ground truth. p_i is the predicted probability from the Softmax function. $w_1 : w_2$ is the negative training samples to positive samples ratio. In urban scenes, the negative samples (unchanged) are usually several times and even more than the positive samples (changed). By assigning imbalanced weights to the loss function, we make a larger penalization to a false positive than a false negative to guarantee less false positives.

3.3 Connected component analysis

Connected component analysis is adopted as post-processing to remove isolated patches. The principle is that small isolated patches are not likely to be a real building change but rather a false positive. It contains two steps. First, the changed patches are connected with their 8-neighborhood on the orthoimage, which brings many candidate changed components. Second, the minimum enclosing rectangle is calculated for each connected component. If the maximum side length of the rectangle is smaller than T_3, this component is regarded as false positive and removed. T_3 is set according to the minimum size of the changed buildings we propose to detect in the study area. After post-processing, many isolated changed patches are amended to be unchanged.

4. EXPERIMENTS

4.1 Study area

The study area is located in Rotterdam, a densely-built port city. The airborne laser points and aerial images were acquired in 2007 and 2016, respectively. The study area is 14.5 km^2 as shown in Fig. 3. 2160 aerial images were acquired by CycloMedia from five perspectives. The tilt angle of the oblique view was approximately 45°. The image size is 7360 × 4912 pixels. The Ground Sampling Distance (GSD) of the nadir images equals 0.1 m. The bundle adjustment and dense matching were run in Pix4Dmapper. The vertical RMSE of 48 GCPs is 0.021 m and

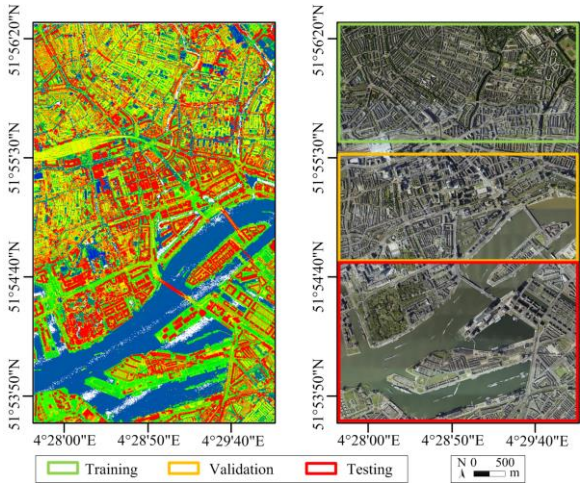

Figure 3. Study area. (a) ALS data colored according to point height. (b) Orthoimage colored based on true object color.

the vertical RMSE of 20 check points is 0.058 m. After dense matching, DSMs and orthoimages were generated at the same resolution of 0.1 m. Fig. 3(a) shows the laser scanning data and Fig. 3(b) shows the generated orthoimage. The training, validation and testing area make up 28%, 25% and 42% of the study area, respectively.

4.2 Experimental setup

After pre-processing, the grid cells of the two DSMs are strictly registered with the pixels on orthoimages. The unified interval is 0.1 m in X and Y directions. The patch size is 100 × 100 pixels, which corresponds to 10 m × 10 m in object space. During sample selection, T_1 and T_2 mentioned in Section 3.1 are both set to 0.1. Since only a few changed buildings exist in the training area, we use two strategies when preparing positive training samples: (1) Half-overlap sampling: Selecting positive patches with a stride of half-patch size allows us to make complete sampling of the changed areas. (2) Data augmentation: Each positive sample is rotated by 90°, 180°, and 270° and also horizontally and vertically flipped (Zhan et al., 2017). When selecting negative training samples, validation samples and testing samples, half-overlap sampling is adopted but data augmentation is not. The number of training, validation and testing samples are shown in Table 1. Table 1 shows that there are much more negative samples than positive samples in the validation and testing sets. And the ratio of positive to negative samples in three sets are different. The ratio $w_1 : w_2$ is set to $5.18 : 1$ based on the number of positive and negative samples in the training set.

	Positive	Negative	Total samples	Pos-to-neg ratio
Training	22,398	116,061	138,459	1 : 5.18
Validation	2,925	104,111	107,036	1 : 35.6
Testing	6,192	129,026	135,218	1 : 20.8

Table 1. Number of training, validation and testing samples.

Fig. 4 shows 5 positive and 5 negative training samples. Magenta indicates the building changes which are either new or heightened. Cyan indicates a demolished or lowered building. Yellow indicates data gaps in either laser points or dense matching points.

T_3, mentioned in Section 3.3, is set to 10 m and 20 m for a comparative study, which means that only changed buildings

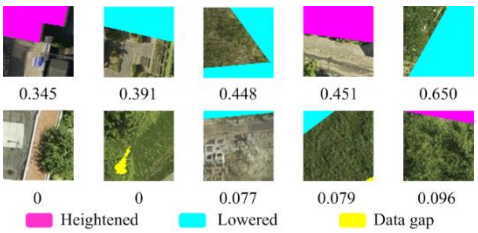

Figure 4. Training samples. (a) Top row: 5 Positive examples. (b) Bottom row: 5 negative examples. The digits below each sample are the ratio of changed pixels in the whole patch.

larger than 10 m or 20 m in length are kept after post-processing. The network is trained from scratch. The batch size is 128. The optimization algorithm is Stochastic Gradient Descent (SGD) with momentum (Goodfellow et al., 2016). The learning rate starts from 0.008 and decreases by 0.003 after every 30 epochs. We train the network for 80 epochs with a momentum of 0.90. The training process is run on a single NVIDIA GeForce GTX Titan GPU with 11G memory.

To evaluate the performance of our strongly-imbalanced classification, we consider widely-used evaluation metrics: recall, precision and F_1-score. Recall indicates the ability of a model to detect all the real changes. Precision indicates the ability of a model to detect real changes. F_1-score is a metric to combine recall and precision using the harmonic mean. $Recall = TP/(TP + FN)$, $Precision = TP/(TP + FP)$, $F_1 = 2 \cdot (Recall \cdot Precision)/(Recall + Precision)$. True Positive (TP) is the number of correctly detected changes. True Negative (TN) is the number of unchanged entities detected as unchanged. False Positive (FP) is the number of changes detected by the algorithm which are not changes in the real scene. False Negative (FN) is the number of undetected changes.

5. RESULTS AND ANALYSES

5.1 Validation and testing results

It should be noted that our classification problem is a strongly-imbalanced binary-classification problem. In real urban scenes, there are usually much more unchanged buildings than changed buildings. This brings two research problems: (1) The number of negative (unchanged) samples are several tens of times more numerous than the positive (changed) samples. The limited positive samples may not be enough to allow the model to learn change patterns. (2) Data distribution of positive and negative samples in the training set, validation set and testing set are different. In this case, the validation and testing performance of a CNN model will present a large difference.

During training, the model is evaluated on the validation set after every three epochs to check its performance and ensure that there is no overfitting. Towards the end of training, the model with the highest F_1-score is selected as the final trained model. The validation results are as follows: TP is 2,362; TN is 101,636; FP is 2,475; FN is 563. Recall equals 0.8075; Precision equals 0.4883; F_1-score equals 0.6086. That is, 80.75% positive samples are correctly inferred as positive; 97.62% negative samples are correctly inferred as negative.

The testing results are listed in the beginning rows of Table 2. The model *HHC-3convB* indicates that the CNN model contains 3 convolution blocks. And it takes ALS-DSM, DIM-DSM and orthoimage as input (*H* indicates height, while *C* indicates color).

Network	PP level	Recall	Precision	F_1-score
HHC-3convB	w/o PP	0.8217	0.6717	0.7392
	T_3 = 10 m	0.8212	0.7166	0.7654
	T_3 = 20 m	0.8146	**0.7632**	**0.7881**
HH-3convB	w/o PP	0.8143	0.6265	0.7081
	T_3 = 10 m	0.8135	0.6737	0.7370
	T_3 = 20 m	0.8112	0.7273	0.7670
HHC-4convB	w/o PP	0.7988	0.5866	0.6764
	T_3 = 10 m	0.7985	0.6402	0.7106
	T_3 = 20 m	0.7943	0.6995	0.7439
HH-4convB	w/o PP	**0.8240**	0.5789	0.6800
	T_3 = 10 m	0.8236	0.6258	0.7112
	T_3 = 20 m	0.8219	0.6798	0.7441

Table 2. Testing results of different CNN architectures (w/o PP: without post-processing).

As competing models, we also implement three other methods: *HH_3convB*, *HHC-4convB*, *HH-4convB*. HH indicates that only two DSMs are used as inputs and that the orthoimage is not used. *4convB* indicates that the CNN architecture contains 4 convolution blocks followed by 3 fully connected layers. The classified patches are then post-processed with two different thresholds in the connected component analysis: 10 m or 20 m. The maximum value in each column is highlighted in bold. The motivation of studying *4convB* is to check whether the model improves if deeper networks are adopted. The recall and precision in Table 2 are also visualized in Fig. 5.

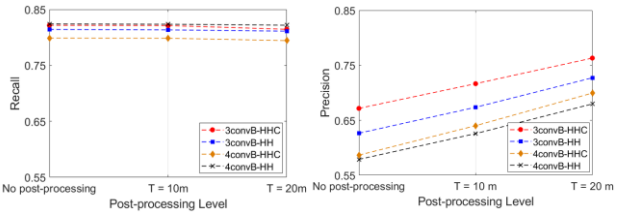

Figure 5. Recall and precision of different models with different post-processing levels. (a) Recall vs. post-processing level. (b) Precision vs. post-processing level. Four curves represent four different models.

To qualitatively evaluate our results, the change maps are visualized in Fig. 6. Fig. 6(a) is the ground truth. Fig. 6(b) is the original output from the model *HHC-3convB*. Figures 6(c) and 6(d) are the change maps after post-processing with 10 m and 20 m thresholds, respectively. Fig. 7 shows the change maps from four CNN architectures. All these change maps have been post-processed with T_3 = 20 m.

5.2 Discussions

(1) Comparison of the four models

In Table 2, the F_1-score is used to measure the overall model performance. Among the 12 configurations, HHC-3convB (T_3 = 20 m) achieves the highest F_1-score of 0.7881. Its recall rate and precision rate reach 0.8146 and 0.7632, respectively. Comparing the testing results from four CNN architectures with no post-processing, the F1-score of HHC-3convB is higher than the other three models. In the models with 3 convolution blocks, the F1-score of the model with HHC as input are higher than the model with HH as input by a margin of 3%. This can be explained by the fact that orthoimages provide additional information on making a correct prediction. Namely, the two DSM patches tell the height difference between the two point clouds, while color features tell whether the object is ground or vegetation.

Furthermore, if it is vegetation, the model might learn that a DSM height change is probably caused by heavy noise.

Table 2 also shows that the model with 3 convB performs better than the model with 4 convB concerning the F_1-score. It is easier for larger CNN models (with more layers or more nodes) to overfit the training data. Although both HHC-3convB and HHC-4convB fit well on the training data, the generalization capability of HHC-3convB is better than HHC-4convB. Based on Occam's razor principle (Novak et al., 2018), when two models have comparable performance, it is suggested to take the simpler one. In addition, Table 2 also shows that taking both DSMs and orthoimages as inputs (HHC-3convB, HHC-4convB) is better than taking merely DSMs as input (HH-3convB, HH-4convB). The orthoimage can provide color features, which contribute to making correct inference.

(2) Impact of the post-processing levels
Fig. 6(a) shows that post-processing only slightly impacts the recall rate but makes a remarkable improvement on the precision rate. As mentioned in Section 3.3, a lot of FPs are converted to TNs. When a certain patch is FN (namely omitted in the change map), it cannot be remedied by post-processing. According to Eq. (2), the recall rate is not affected while the precision rate is affected.
When selecting the post-processing threshold T_3, T_3 should be determined by the targeted size of changed buildings. Using T_3 = 20 m for post-processing will filter out all the patch components smaller than 20 m. Fig. 6(b) and (d) show that using post-processing with T_3 = 20 increases the precision rate by 9.15% at the expense of decreasing the recall rate by 0.69%. As mentioned earlier, post-processing converts many detected changed patches as non-changes. When the threshold is large, many small but real changed patches are regarded as non-changed and thus removed. This over-processing causes more FNs (omissions), which leads to a slightly decreasing recall rate.

(3) Analysis on the change maps
Fig. 6(a) shows the ground truth for the testing set. Fig. 6(b) shows the change map from HHC-3convB without post-processing. Generally, most changed patches and unchanged patches are correctly inferred. Increasing the level of post-processing (from Fig. 6(b) to (d)), the false positives (magenta) are gradually decreased, since post-processing with larger thresholds are employed. The original change map from the CNN model contains quite some isolated FPs. This can be explained by that these areas represented by the square patches are similar to the changed pattern learned from the training data. Therefore, they are misclassified into changed patches (FPs). In Fig. 6(b), several FPs can be viewed on the park covered by dense vegetation located in the middle-left of the testing area. As mentioned in Section 1, laser points and dense matching points show quite different properties on canopies. These patches on vegetated areas are wrongly inferred as changed due to large differences due to the acquisition types.

Some FPs appear on the terrain, especially on some construction sites. In these cases, the terrain is excavated or re-paved, the height is changed and the texture of terrain surface is close to that of roof surface: therefore the model misclassifies a changed terrain into a changed building, which leads to FPs. In addition, Fig. 6(b) also shows that FPs are more likely to appear along narrow alleys or in the shadow between tall buildings. In these areas, dense matching tends to perform poorly due to limited visible rays and poor image contrast. The point clouds are inclined to be less accurate and noisier in these areas.

Both FNs and FPs often appear along the edges of changed buildings. When a patch is exactly stretching over the edge of changed buildings, part of the patch is changed while part of it is unchanged. Patches along building edges are often difficult and ambiguous to infer for three reasons: First, dense matching performs poorly in narrow alleys so the DIM point cloud is noisy in those areas. Second, mis-registration between the two point clouds is more severe along building edges than on other smooth surfaces. Hereby, height differences between the two DSMs may appear due to mis-registration errors. Third, we define the changed patches based on the ratio of changed pixels within a patch. This leads to ambiguity if the changed ratio is close to the threshold (10% in our case).

In Fig. 6(b), some FNs appear due to small changed buildings. When building changes are too small, dense matching may not generate sufficient points to form an accurate DSM. The patches are thus misclassified into unchanged. In Fig. 7, (a) and (b) generally contain less FPs than (c) and (d). This is also reflected in Table 2(b) that the precision rate of CNN models with 3 convolution blocks is higher than those with 4 convolution blocks. A possible explanation is that the larger model has been overfit to the training data.

(4) Object-based evaluation
Until now, the evaluation has been made based on individual patches. We can also evaluate the performance on individual building level. Each connected component is counted as a detected building change. There are 86 buildings labeled as changed in the testing area. In Fig. 6(b), 6(c) and 6(d), 79, 76 and 74 building changes are detected, respectively. Using post-processing with T_3 = 20 removed many false detections as well as 5 true changes. These five changed buildings are all small changes and mis-classified into FPs in post-processing.

6. CONCLUSIONS

This paper proposes a framework to detect building changes between laser scanning points and dense matching points. The two types of point clouds present different characteristics and each of them contains noise and data gaps. A light-weighted feed-forward CNN with three convolution blocks and three fully connected layers is used for change detection. Square patches cropped from ALS-DSM, DIM-DSM and orthoimage are fed into the CNN architecture. The feature maps inferred by CNN are post-processed by connected component analysis. Patch-based evaluation shows that the recall rate after post-processing reaches 0.8146 while precision rate reaches 0.7632. Object-based evaluation shows that 74 out of 86 building changes are correctly detected although the change maps still contain many FPs.

The advantage of our method is that CNN allows to fastly localize the building changes without feature engineering or change vector analysis. The feature extraction and comparison steps are both implicitly included in the CNN network. Concerning the limitations of our method, there are still some FPs and FNs in the change map after post-processing. In the future work, the change detection framework can be improved from two aspects: First, we can add more contextual information between the patches. Specifically, Fully Convolutional Neural Network (FCN) might be a solution (Long et al., 2015). FCN model is more complicated than feed-forward CNN and requires much more samples to train. Second, the current feed-forward CNN architecture can be extended to a Siamese CNN (Mou et al., 2017), which extracts features separately in two branches and then concatenates them in a later stage.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-IV-2-W5-453-2019 | © Authors 2019. CC BY 4.0 License.

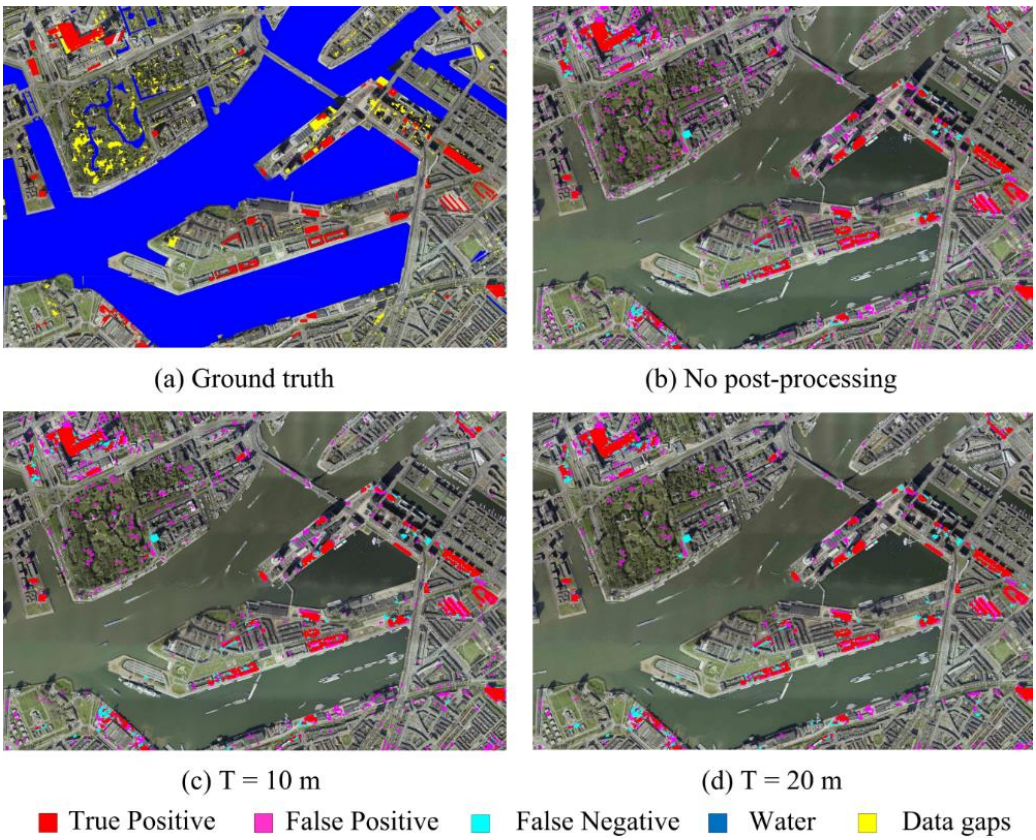

Figure 6. Change maps generated from model *HHC-3convB*. (a) Ground truth; (b) original change map without post-processing. (c) post-processed with T_3 = 10 m. (d) post-processed with T_3 = 20 m.

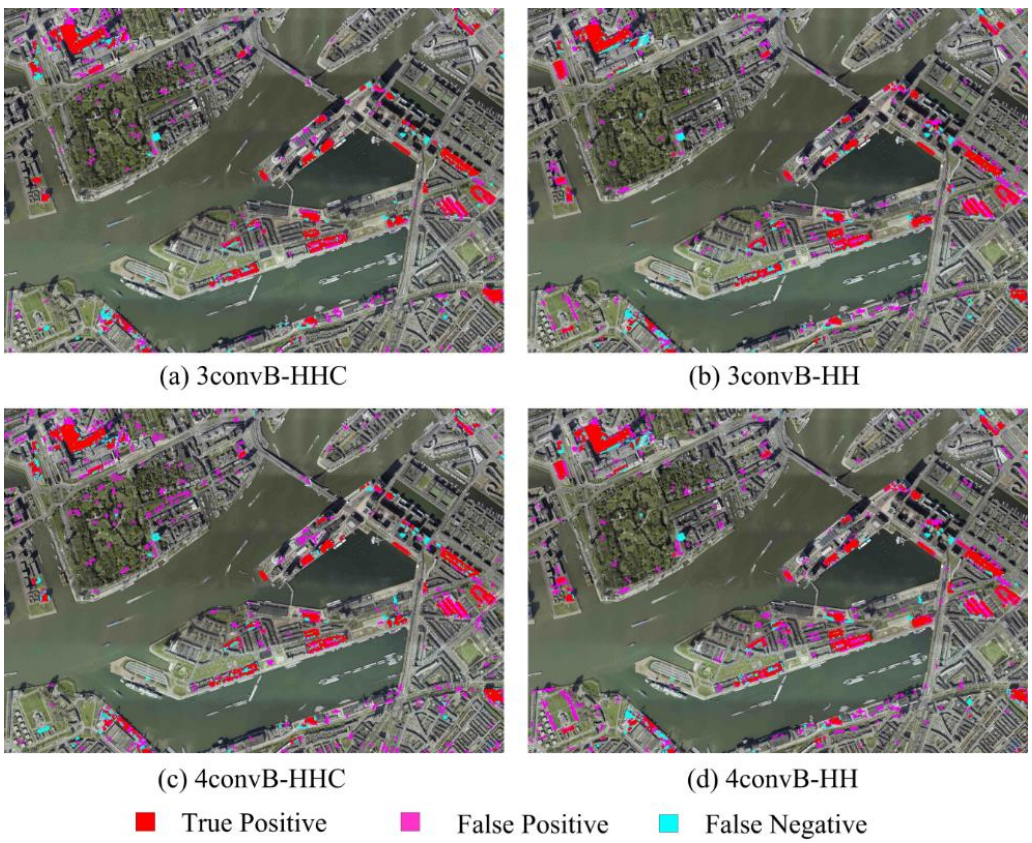

Figure 7. Change maps from four CNN architectures (T_3 = 20 m): 3convB-HHC, 3convB-HH, 4convB-HHC, 4convB-HH.

REFERENCES

Audebert, N., Le Saux, B. and Lefèvre, S., 2018. Beyond RGB: Very high resolution urban remote sensing with multimodal deep networks. ISPRS J. Photogram. Remote Sens., 140, pp.20-32.

Basgall, P.L., Kruse, F.A. and Olsen, R.C., 2014. Comparison of lidar and stereo photogrammetric point clouds for change detection. In Laser Radar Technology and Applications XIX; and Atmospheric Propagation XI. Vol. 9080, pp. 90800R.

Chen, L.C., and Lin, L. J., 2010. Detection of building changes from aerial images and light detection and ranging (LIDAR) data. J. of Appl. Remote Sens., 4(1), 041870.

Daudt, R.C., Le Saux, B., Boulch, A. and Gousseau, Y., 2018. Urban change detection for multispectral earth observation using convolutional neural networks. In Int. Geoscience and Remote Sens. Symp. (IGARSS).

Du, S., Zhang, Y., Qin, R., Yang, Z., Zou, Z., Tang, Y. and Fan, C., 2016. Building change detection using old aerial images and new LiDAR data. Remote Sens., 8(12), pp.1030.

Goodfellow, I., Bengio, Y., Courville, A. and Bengio, Y., 2016. Deep learning (Vol. 1). Cambridge: MIT press.

Hu, X. and Yuan, Y., 2016. Deep-learning-based classification for DTM extraction from ALS point cloud. Remote sens., 8(9), pp.730.

Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pp. 1097-1105.

Long, J., Shelhamer, E. and Darrell, T., 2015. Fully convolutional networks for semantic segmentation. CVPR, pp. 3431-3440.

Lu, D., Mausel, P., Brondízio, E., and Moran, E., 2004. Change detection techniques. Int. J. of Remote Sens., 25(12). pp. 2365–2401.

Mandlburger, G., Wenzel, K., Spitzer, A., Haala, N., Glira, P. and Pfeifer, N., 2017. Improved topographic models via concurrent airborne lidar and dense image matching. ISPRS Ann. Photogram. Remote Sens. Spatial Inf. Sci. IV-2/W4, 259-266.

McGlone, J.C., 2013. Manual of Photogrammetry (Sixth Edition). ASPRS.

Mou, L., Schmitt, M., Wang, Y. and Zhu, X.X., 2017, March. A CNN for the identification of corresponding patches in SAR and optical imagery of urban scenes. In Urban Remote Sensing Event (JURSE), 2017 Joint. IEEE. pp. 1-4.

Nex, F., Gerke, M., Remondino, F., Przybilla, H.J., Bäumker, M. and Zurhorst, A., 2015. ISPRS benchmark for multi-platform photogrammetry. ISPRS Ann. Photogram. Remote Sens. Spatial Inf. Sci. 2(3), pp. 135-142.

Novak, R., Bahri, Y., Abolafia, D.A., Pennington, J. and Sohl-Dickstein, J., 2018. Sensitivity and generalization in neural networks: an empirical study. arXiv preprint arXiv:1802.08760.

Olsen, B. P., 2004. Automatic change detection for validation of digital map databases. Int. Arch. of Photogram. and Remote Sens., 30, pp. 569–574.

Qin, R., Tian, J. and Reinartz, P., 2016. 3D change detection–approaches and applications. ISPRS J. Photogram. Remote Sens., 122, pp. 41-56.

Remondino, F., Spera, M.G., Nocerino, E., Menna, F. and Nex, F., 2014. State of the art in high density image matching. The Photogrammetric Record, 29(146), pp. 144-166.

Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pp. 91-99.

Ressl, C., Brockmann, H., Mandlburger, G. and Pfeifer, N., 2016. Dense image matching vs. airborne laser scanning–comparison of two methods for deriving terrain models. Photogrammetrie, Fernerkundung, Geoinformation (PFG). 2, pp. 57-73.

Rizaldy, A., Persello, C., Gevaert, C., Oude Elberink, S. and Vosselman, G., 2018. Ground and multi-class classification of Airborne Laser Scanner point clouds using Fully Convolutional Networks. Remote sens., 10(11), pp.1723.

Singh, A., 1989. Digital change detection techniques using remotely-sensed data. International journal of remote sensing, 10(6), pp. 989-1003.

Stal, C., Tack, F., De Maeyer, P., De Wulf, A., & Goossens, R., 2013. Airborne photogrammetry and lidar for DSM extraction and 3D change detection over an urban area – a comparative study. International Journal of Remote Sensing, 34(4), pp. 1087–1110.

Tran, T.H.G., Ressl, C. and Pfeifer, N., 2018. Integrated change detection and classification in urban areas based on airborne laser scanning point clouds. Sensors, 18(2), pp. 448.

Volpi, M. and Tuia, D., 2018. Deep multi-task learning for a geographically-regularized semantic segmentation of aerial images. ISPRS J. Photogram. Remote Sens., 144, pp. 48-60.

Vosselman, G. and Maas, H.G., 2010. Airborne and terrestrial laser scanning. CRC.

Vosselman, G., Gorte, B.G.H. and Sithole, G., 2004. Change detection for updating medium scale maps using laser altimetry. Int. Arch. of Photogramm., Remote Sens. and Spatial Inf. Sci., 34(B3), pp. 207-212.

Xu, S., Vosselman, G. and Oude Elberink, S., 2015. Detection and classification of changes in buildings from airborne laser scanning data. Remote sens., 7(12), pp. 17051-17076.

Zhan, Y., Fu, K., Yan, M., Sun, X., Wang, H. and Qiu, X., 2017. Change Detection Based on Deep Siamese Convolutional Network for Optical Aerial Images. IEEE Geos. and Remote Sens. Letters, 14(10), pp. 1845-1849.

Zhang, Z., Gerke, M., Vosselman, G. and Yang, M.Y., 2018. A patch-based method for the evaluation of dense image matching quality. Int. J. of Appl. Earth Observation and Geo-information, 70, pp. 25-34.