TECHNICAL REPORT: CHANGE DETECTION IN REMOTE SENSING USING DEEP SIAMESE CONVOLUTIONAL NEURAL NETWORKS

Vitek Ruzicka
ETH Zurich
previtus@gmail.com

Stefano d'Aronco
ETH Zurich
stefano.daronco@geod.baug.ethz.ch

Jan Dirk Wegner
ETH Zurich
jan.wegner@geod.baug.ethz.ch

ABSTRACT

We propose a method for change detection in aerial images acquired at two different points in time of the same region. For this task, we adopt a siamese deep convolutional neural network model with a U-Net [1] structure and pre-trained ResNet50 [2] encoder, which we call *Siamese U-Net ResNet50* in the following. We achieve a per pixel AUC score of 92.92% $\pm$ 0.84% and a per tile recall score of 92.45% $\pm$ 2.48%with this model. We show that using our automated method, it is possible to reduce 98.44% $\pm$ 0.16% of the necessary manual checking effort of tiles in the process of updating Swisstopo map products while achieving very high recall.

1 INTRODUCTION

In this work, we present a deep convolutional neural networks (CNN) approach for change detection in aerial images of a region in Switzerland. To this end, we introduce a Siamese variant of the widely used U-Net model [1], which is efficient to compute while delivering state-of-the-art performance for pixel-accurate semantic segmentation. All source code affiliated with this technical report is available on GitHub[1].

A large body of literature exists for change detection [3] and, similar to computer vision tasks, the comeback of deep learning has led to significant progress [4] and [5]. While their are different variants of change detection depending on the data source and specific task, we define change detection as labeling all pixels in an image that show a significant change of a building footprint compared to another, co-registered image of the same place acquire earlier.

Defining a comprehensive set of rules that would include any kind of possible change in build-up areas in Switzerland is unfeasible especially given that any other change caused by illumination differences, moving objects etc. should be ignored. We thus approach this problem in a completely data-driven way by training a classifier that learns to distinguish changed building footprints from unchanged areas. Although one could possibly train a model for any kind of object category that shows visible change in aerial images, we solely work with changes of building footprints in this study because it was the only object category available with labels in a format amenable to a classifier. After training our model on a dataset of aerial image pairs provided by Swisstopo, we were able to make pixel-accurate change predictions in the original resolution of the images. In addition, we propose a simple yet accurate method to be more robust against small outliers by summing up all changed pixels per image patch. We classify all image patches into patches without and with relevant changes, which is a first step towards a semi-automated method to help a human annotator check only patches classified as containing change.

[1]Code available at: https://github.com/previtus/ChangeDetectionBaseline/

To solve this task, we have developed and implemented a deep CNN model with a U-Net architecture [1] using ResNet modules [2] pre-trained on the ImageNet dataset [6] in a transfer learning setting [7]. In order to compare images acquired of two different points in time, we have designed a Siamese CNN variant of the original U-Net model, which will be explained in more detail in the following. Using these models we were able to detect changes of building instances at image patch level with high recall of 92.45% $\pm$ 2.48% and at pixel-level with an AUC of 92.92% $\pm$ 0.84%. With our method, it is possible to reduce the amount of tiles of the map which need to be checked by human annotators by 98.44% $\pm$ 0.16% (only 1.56% of the entire scene has be checked) while detecting 92.45% $\pm$ 2.48% of all existing changes.

2 METHOD

The task of change detection in our case consists of detecting change occurring between two aerial images of the same place between two different points in time. In our case, we have pixel precise annotations denoting change between building footprints as recorded in a vector map.

The dataset provided by Swisstopo consists of images recorded over a $303\ km^2$ region of Aarau (recorded in maps as area No. 1089 in 25 cm resolution) in 2012 and 2015. Each image contains infrared and RGB channels. However, here we only rely on RGB information to generalize better to possible further applications (e.g., street-level imagery from mobile mapping) that usually come without infrared information. Furthermore, we work with change annotations in an image of the same resolution where pixels with values "0" mark no change and pixels with values "1" mark change. We split the original, very large region into individual image patches of size $256 \times 256\ px$ with $32\ px$ overlap of adjacent patches in all four directions. Tiling the original images into small patches is done as a pre-processing step using ArcGIS and the Split Raster tool from the Raster Processing toolset. We empirically found an image size of $256 \times 256\ px$ a good compromise between containing entire buildings and their context while enabling computational efficiency.

We initially generated the change label reference map by automatic subtraction of two versions of vector maps made manually for both years. However, since both manually generated building footprint maps had been generated separately and with slightly different definitions of footprints, we had to manually clean change labels to reduce the amount of label noise. Our understanding is, that these inconsistencies of building footptint labels were partially caused by a change in the processing setup of Swisstopo for generating the vector maps. In addition, some building footprint labels were added to the maps from additional sources of information without the change being visually present in the corresponding pair of images. Moreover, we also noted some examples where a change present in the image pairs was not recorded in manually annotated building footprint maps. At a later poject stage we were supplied with an additional dataset that contained only the incremental building footprint updates. However, many updates seemed to originate from auxiliary sources and were not visible in the aerial images, while obvious, relevant building changes in the aerial images were still missing. We thus decided to proceed with our own, cleaned version as described above.

Tiling of the dataset into small image patches results in 83144 pairs of $256 \times 256 \times 3$ images, each pair accompanied with the same resolution image label of two classes indicating the presence of change. In addition to pixel-accurate change labels, we also marked each patch pair as containing a significant change or not. More precisely, we assigned each image patch with more than 3% of changed pixels to the "change" class and those with less than 1% of changed pixels to the "no change" class. We chose these thresholds empirically to account for small amounts of label noise near the boundaries of houses. An example for our patch classification strategy is shown in Fig. 1.

2.1 SIAMESE CNN APPROACH

Following the recent progress in deep learning [4], [5] we use a CNN model for change detection in aerial images. We adapt U-Net model [1] architecture, which consists of two segments - a down-scaling encoder section (left in Fig. 2) and an up-scaling decoder section (right in Fig. 2). These two parts are connected with so-called skip connections (blue and red arrows in Fig. 2), which are concatenated with features of the same resolution in the decoder stage. These skip connections help retain all detailed information of the original resolution (in our case $256 \times 256\ px$). In our implementation, we use the ResNet50 model [2] as the encoder with initial weights from pre-training on

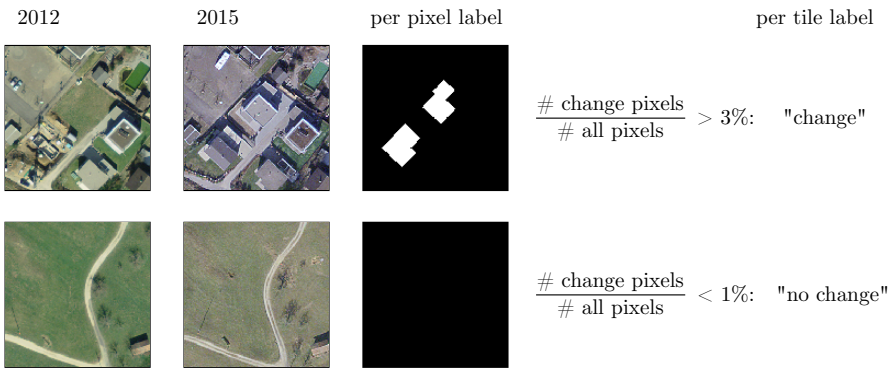


Figure 1: Example of pairs of aerial images in the dataset and their labels.

the ImageNet dataset [6]. Starting from a model with weights pre-trained on a very large, auxiliary dataset of the same image modality is good practice if labeled training data is scarce for a certain task (i.e., pixels with changes in our case). Moreoverm, we adopt the Siamese neural network paradigm, where for our pair of two input image patches of 2012 and 2015, an encoder with shared weights [8] is used. The high level features of these two inputs are concatenated whenever we are either using skip connections, or at the end of the shared encoder section.

See the full model structure on Figure 2. and note that the encoder is initiated from a pre-trained ResNet model.

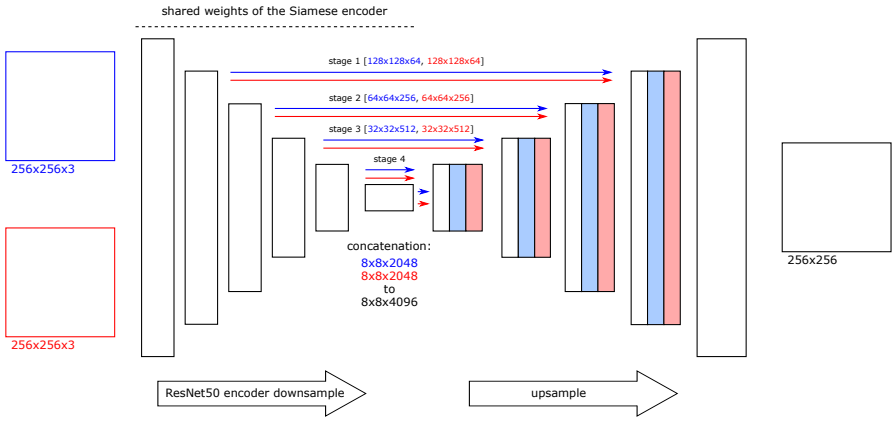


Figure 2: Siamese U-Net architecture with ResNet50 encoder

2.2 Experiment setup

We note that our dataset is severely unbalanced regarding labels changed and unchanged. Only 1072 image patches out of 83144 patches in total contain changes. Furthermore, for the vast majority of patches with changes, the proportion of changed pixels is very small (usually far below 10%). It should also be noted that there are many differences in the images of the two aerial campaigns due to different season, time during the day (different direction of shadows) and further changes in vegetation etc. that the classifier should ignore while correctly predicting changes of building footprints. To cope with this lack of balance in our dataset, we adopt two approaches. First, we give three times higher weight to all pixels containing change in the training loss function. Second, we curate a balanced subset of the whole dataset to train the model by sampling all 1072 pairs

Table 1: Results of the Siamese U-Net ResNet50 model

	AUC (per pixel)	**Recall (per tile)**
balanced set (class distribution 1:1)	92.02 ± 0.56	92.45 ± 2.48
unbalanced set (class distribution 1:80)	92.92 ± 0.84	92.45 ± 2.48

containing change and randomly sampling additional 1072 pairs without any change. During testing we evaluate on both, a dataset with the original dataset distribution (1:80) and a balanced set (1:1).

We also use data augmentation techniques on our full training set to generate one additional pair for each pair of images in the training set using one of the following transformations: flip horizontally, flip vertically, rotation of 90 or 270 degrees.

To achieve statistically meaningful results, we apply a k-fold cross validation scheme and we sample data from non-overlapping regions into the training, validation and testing sets for all experiments. The model predicts classes per pixels with output scores that range between 0 and 1. We therefore have to choose a threshold to assign predictions to the two classes. This threshold is selected by finding the one that maximizes the f1-score of per-tile evaluation on the validation set. Final score is reported with the AUC (area under curve, i.e., the integral under the recall-precision curve, higher is better) metric score on the non-thresholded per pixel predictions and on the thresholded per tile predictions with the recall metric. We report results for per-pixel and per-patch evaluation.

For our task, we mainly care about the recall metric as it reflects on how many "changed" tiles will be successfully detected from all changed tiles in the dataset. Finally we have also calculated the human annotators cost by the formula $cost = (TP + FP) / N$. This reflects the cost of how many tiles does the human annotator have to manually check after using our algorithm. All tiles labelled as change need to be manually checked, either to detect False Positive classifications (when there is in fact no change in the pair, but it has been marked as one containing change) or to add the manual annotation of the True Positives containing newly built houses to the maps.

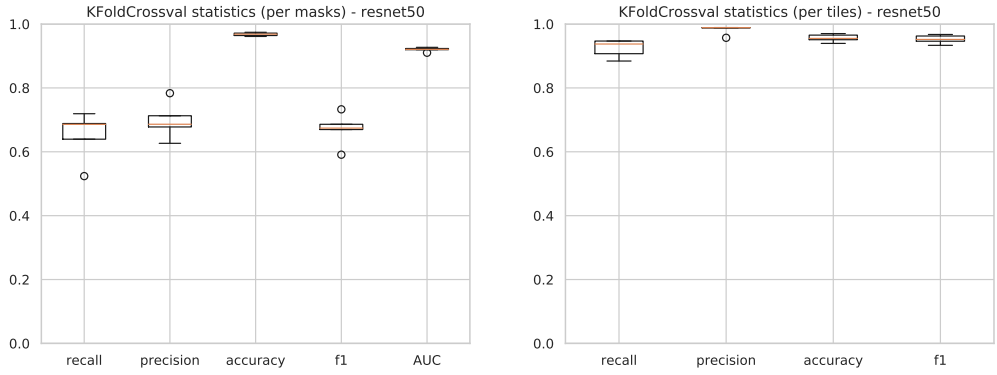
We train the proposed model for 100 epochs with batch size of 16 pairs, the Adam optimizer and learning rate of 0.00001 with weighted categorical cross entropy loss.

3 RESULTS

Table 1 shows the results of our trained model evaluated on a balanced test set (with class distribution 1:1) and on an unbalanced test set with class distribution corresponding to the one of the original dataset (1:80). Using a balanced test set, we report our results as 92.02 ± 0.56 AUC in the per-pixel evaluation and as 92.45 ± 2.48 recall in the per-patch evaluation. Using a test set with the original distribution of data we get results of 92.92 ± 0.84 AUC in the per-pixel evaluation and as 92.45 ± 2.48 recall in the per-patch evaluation. Fig. 3 and Fig. 4 show additional metrics measured on these test sets such as recall, precision, accuracy and the AUC score for the per-pixel prediction.

Qualitative results are shown in Fig. 5(a) for correctly predicted examples and in Fig. 5(b) for typical errors. The first row in Fig. 5(b) of the error cases shows a False Negative error, where the model has missed a change which was present in the image pair. This is the most severe type of error since it directly influences the update quality of the mapping procedure. Consequently, this error type is reflected by the recall metric (which is influenced by False Negatives). The second row shows a False Positive error, where the model is predicting a change in a case where there is none in the image pair. We note that this kind of error is less serious as we are not missing any changes from the dataset. The third row shows a case where our model correctly detected a change, while the correct annotation was missing in the labels[2]. The last row shows an example of a hard case, where the change is barely visible even for a human annotator.

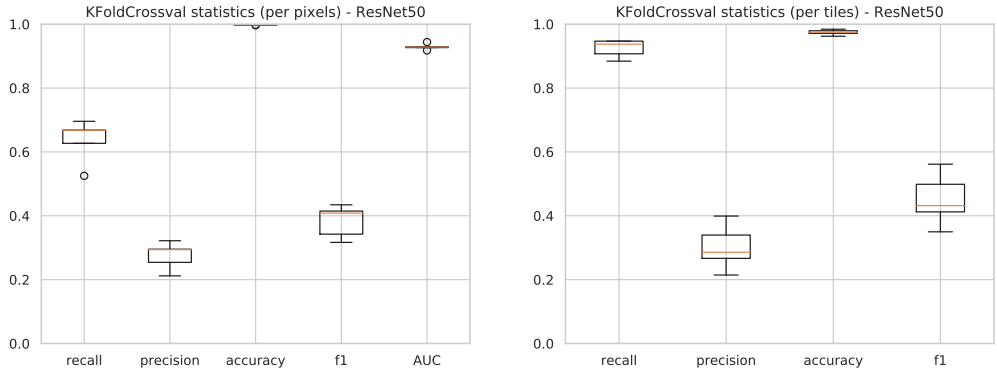
[2]We have excluded these cases from the evaluation statistics.



(a) Per pixel evaluation

(b) Per tile evaluation

Figure 3: Performance over balanced set (class distribution 1:1).



(a) Per pixel evaluation

(b) Per tile evaluation

Figure 4: Performance over unbalanced set (class distribution 1:80). Note that the lower precision (and consequentially f1 score) when compared with the balanced set is caused by the disproportionately large amount of additional "no change" points in the test set. This leads to more pixels/patches without any change being wrongly classified as changed (see Fig. 5(b) second row from top for an example). The performance on "change" remains the same as indicated by recall and AUC score.

4 Conclusion

In this work we propose a *Siamese U-Net ResNet50* architecture which is novel for the task of change detection and delivers promising results. We achieve an AUC of 92.92% $\pm$ 0.84% for per-pixel evaluation and a recall of 92.45% $\pm$ 2.48% for per-tile evaluation on a test set with the original unbalanced distribution. On the original dataset this results in 98.44% $\pm$ 0.16% reduction of manual change detection while maintaining the reported high recall. More precisely, instead of checking all image patches for changes, a human annotator would only have to check 1.56% of all patches (those labeled as changed) while detecting 92.45% $\pm$ 2.48% of all existing changes.

A promising direction to further improve performance while moving closer to an application scenario is active learning, which combines human experts with deep learning to sample the most meaningful examples for training the classifier. More precisely, a human annotator could retrain the classifier on hard cases that were initially missed to further fine-tune the model on-the-fly.

REFERENCES

[1] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015. [Online]. Available: http://arxiv.org/abs/1505.04597

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[3] D. L. C. author, P. Mausel, E. Brondzio, and E. Moran, "Change detection techniques," *International Journal of Remote Sensing*, vol. 25, no. 12, pp. 2365–2401, 2004. [Online]. Available: https://doi.org/10.1080/0143116031000139863

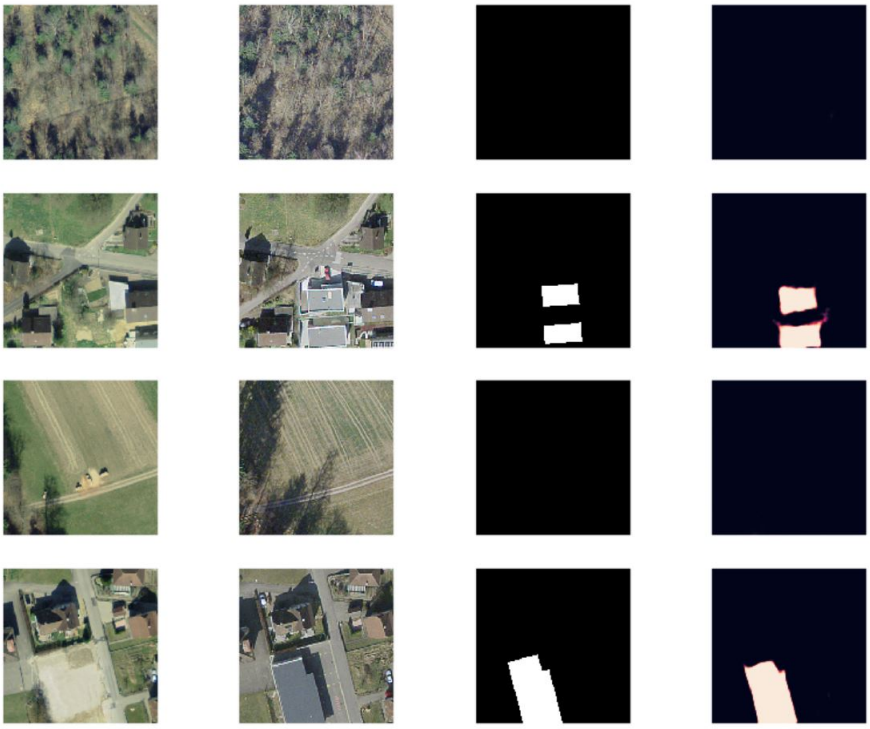
[4] R. C. Daudt, B. Le Saux, and A. Boulch, "Fully convolutional siamese networks for change detection," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 4063–4067.

[5] R. C. Daudt, B. L. Saux, A. Boulch, and Y. Gousseau, "High resolution semantic change detection," *arXiv preprint arXiv:1810.08452*, 2018.

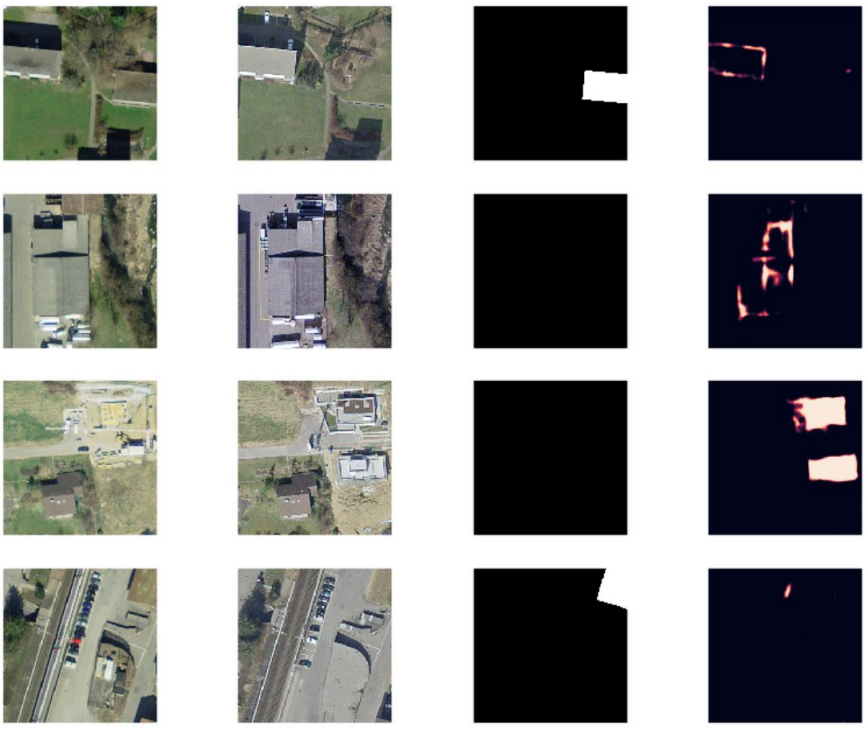
[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[7] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" *CoRR*, vol. abs/1411.1792, 2014. [Online]. Available: http://arxiv.org/abs/1411.1792

[8] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.



(a) Correct prediction examples (from left to right: 2012 aerial image, 2015 aerial image, ground truth, prediction)



(b) Incorrect prediction examples

Figure 5: Qualitative results