Urban Growth Task - Project Report

Tomáš Novák
tomas2211@post.cz
April 8, 2020

1 Introduction

The main goal of this project is to design an algorithm for the prediction of urban growth from satellite imagery, more specifically using Landsat-8 data.

We propose a simple algorithm that segments input images into three classes - *urban*, *non-urban*, and *cloud* using a convolutional neural network. Based on the segmentation, the ratio of the urban area with respect not-occluded area is computed. The ratios are then aggregated into time-series, outliers are removed, and Gaussian smoothing is applied.

The rest of this report is structured as follows. First, the provided dataset is analyzed and divided into training and testing parts. Second, the image segmentation approach is described and tested. Subsequently, the proposed urban growth index is presented and evaluated.

2 Data analysis

The data provided with the task come from two locations, which are referred to as Site 1 and Site 12. Figure 2.1 shows the time distribution of the data for each site and indicates the segmentation ground-truth availability. Class distribution on the labeled samples is depicted in Figure 4.1a for both sites separately.

The time distribution of the samples seems satisfactory to construct a time-series of the urbanization index. However, the class distribution among the labeled images is uneven, which may pose problems during training and validation of the segmentation model. Data from Site 1 contain more than half of *urban* area pixels. Furthermore, Site 12 completely misses labeled *cloud* occlusions.

Ideally, the segmentation model would be trained on data from one location and validated on data from the other and vice-versa, which would give a better overview of the model performance. However, as Site 12 lacks images containing the *cloud* label, this technique is done only in one direction.

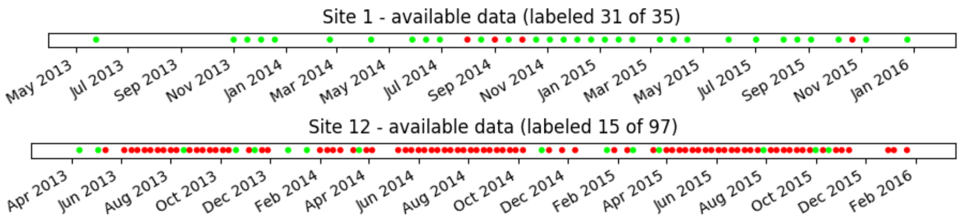
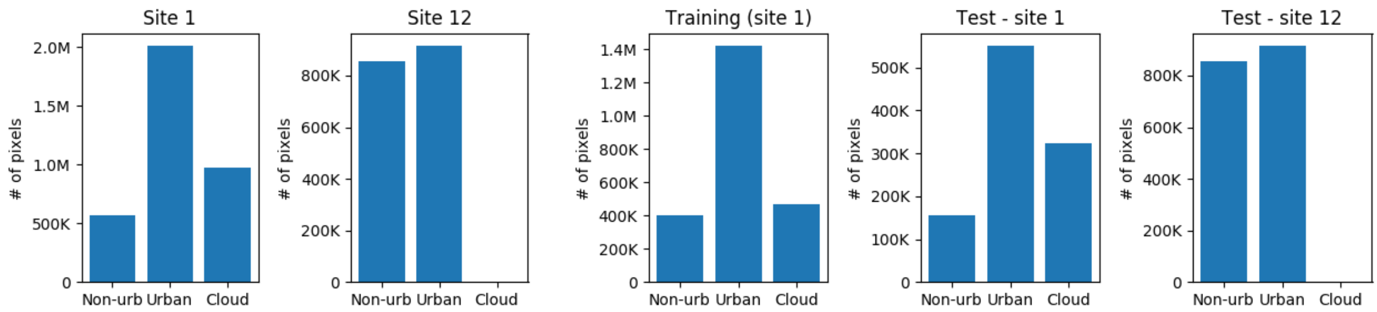


Figure 2.1: **Data time distribution.** Overview of the time distribution of the provided data on both give sites. Green/red point indicates a labeled/ unlabeled imagery respectively.

We hand-pick images from Site 1 for the training set so that it contains only well-labeled samples and a variety of *cloud* class examples. The rest of the images from Site 1 is used as a first test set, which serves mainly for the evaluation of cloud detection. All data from Site 12 are assigned to another test set that will prove the generalization ability of the trained model. The distribution of classes in the training and the test sets are shown in Figure 4.1b.



(a) Given dataset

(b) Training and test sets

Figure 2.2: **Label classes distribution.** Labels distribution in the given dataset and in the split to training and testing parts.

3 Image segmentation

A convolutional neural network was chosen as a segmentation model since it captures the spatial structure of individual labels as opposed to, e.g., per-pixel classification. As a small amount of training data is available, the main focus during the design was on regularization. Specifically, the following techniques were employed:

- Relatively shallow network architecture - as fewer parameters prevent overfitting, the proposed network has only six convolutional layers.
- Strong data augmentation - since even a small area within an image covers a large portion of land and thus contains enough information for classification of each pixel, random crops to size 50×50 px were used. Random affine transformations and data augmentations like random noise or brightness/contrast changes were employed alongside to improve generalization further.
- Common regularization techniques - Dropout (baseline 0.2), weight decay ($1e-4$) and batch normalization.
- As the classes are imbalanced in the training dataset, weighting the per-pixel loss by labels was tested. Two weight settings were investigated - in the order *non-urban*, *urban*, *cloud*: $(6.15, 1.59, 4.78)$ called "strong", which is the inverse of label distribution in the training set, and $(2, 1, 1)$ "weak". In the end, the weights did not have a significant impact on the generalization, possibly thanks to geometrical augmentations.
- AdamW optimizer was chosen as it generally improves the convergence speed. Cross-entropy was used as the loss function.

The network was trained for 50k iterations with batch size 10 (i.e., half of the training set) at a constant learning rate $1e-4$. Experiments with learning rate scheduling (multi-step) showed that a drop in LR only slows down the convergence and does not improve the test error.

Different settings for some parameters were also tested. Table 3.1 summarizes the results. The scores of individual experiments do not differ significantly. The accuracy on *non-urban* class is relatively low on data from Site 1, which is most likely caused by the combination of challenging samples at this location and the class representation disproportion.

The performance of cloud detection measured on the Site 1 dataset is high for all experiments. Also, all seem to generalize well to the unseen data from Site 12. The model with stronger dropout setting was chosen for the urban index evaluation as it seems to generalize the best. It is further evaluated in Table 3.2, which shows that the precision and recall on Site 12 for both non-urban and urban classes is reasonable. The segmentation examples are shown in Figure 3.1 also support this claim.

[F1-score]	Train set N-U	Urb	Cld	Test - Site 1 N-U	Urb	Cld	Test - Site 12 N-U	Urb
Baseline	0.54	0.85	0.98	0.48	0.85	0.94	0.85	0.86
Batch size 64	0.46	0.85	0.97	0.41	0.83	0.92	0.84	0.81
Dropout 0	0.58	0.86	0.98	0.52	0.85	0.95	0.84	0.84
Dropout 0.5	**0.51**	**0.87**	**0.98**	**0.47**	**0.82**	**0.94**	**0.87**	**0.87**
Label weights - strong	0.53	0.76	0.98	0.51	0.77	0.95	0.85	0.84
Label weights - weak	0.53	0.76	0.98	0.51	0.77	0.95	0.85	0.84

Table 3.1: **Parameter experiments.** F1-scores for for the given classes (N-U=non-urban, Urb=urban, Cld=cloud) under different settings on the given training and test data. Bold denotes the model chosen for final evaluation.

	Non-urban			Urban			Cloud		
	P	R	F1	P	R	F1	P	R	F1
Train set	0.57	0.47	0.51	0.85	0.89	0.87	0.98	0.97	0.98
Test - Site 1	0.46	0.47	0.47	0.80	0.83	0.82	0.98	0.90	0.94
Test - Site 12	0.84	0.90	0.87	0.90	0.84	0.87	-	-	-

Table 3.2: **Segmentation model evaluation.** Precision (P), recall (R) and F1-score (F1) of the best performing segmentation model. The metrics are provided for each class and dataset. Note that Test - Site 12 set misses labeled clouds.

4 Urban Growth Index

The urbanization index is calculated based on the segmentation probabilities from the segmentation model. The relative urban area cover in the given image is calculated as

$$U(I_t) = \frac{\sum_{\mathbf{x} \in I_t} p_U(\mathbf{x})}{\sum_{\mathbf{x} \in I_t} p_U(\mathbf{x}) + p_N(\mathbf{x})}, \tag{4.1}$$

where I_t is the input image from time t, $p_U(\mathbf{x})$ and $p_N(\mathbf{x})$ are the estimated probabilities of pixel $\mathbf{x}$ being in class *urban* or *non-urban* respectively, and $U(I_t)$ is the resulting urbanization index at time t.

To cope with the problem of cloud cover, which incurs a substantial amount of miss-labeled pixels, the images with more than 10% of pixels classified as *cloud* are discarded.

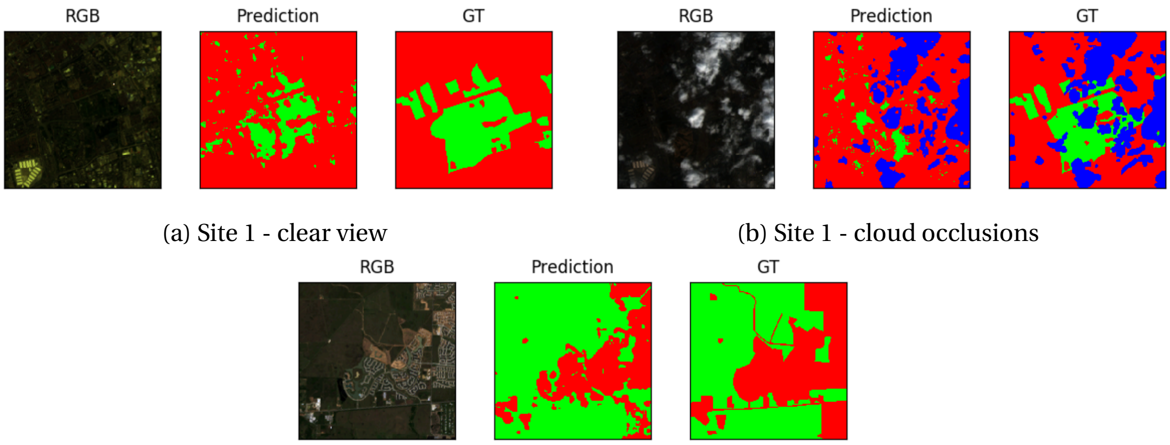
The index is then aggregated into a time-series and filtered. First, outliers are removed by comparing each sample with a median value in a time window around. The window size is set to 180 days, and the value is considered an outlier if it is outside ± 0.1 interval around the median. The time-series is then filtered by a Gaussian with $\sigma = 60$ days.

4.1 Evaluation

Figure 4.1 shows the plotted urban index estimates compared with ground-truth values, which were calculated based on the labels provided with the dataset.

The results from Site 1 are substantially inaccurate, which is caused by the poor performance of segmentation on this location (most likely caused by challenging image features). Filtering avoids coarse mistakes but is not able to mitigate all errors.

However, the model performs well on the test location, and after filtering, the estimated index copies the ground-truth reasonably well. The trained segmentation model (and thus the whole algorithm) seems to generalize to unseen locations. However, more extensive testing with more locations would be needed for conclusive proof.

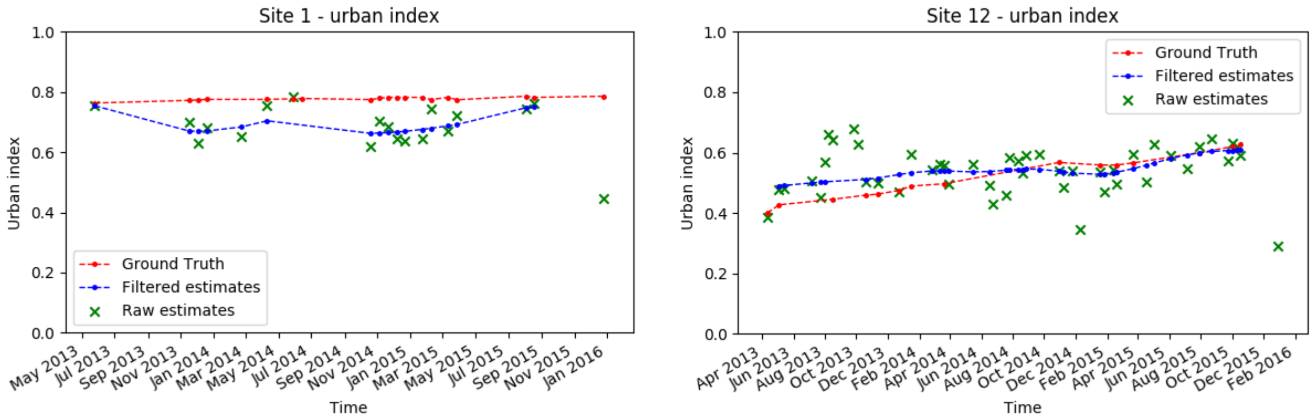


(a) Site 1 - clear view

(b) Site 1 - cloud occlusions

(c) Site 12

Figure 3.1: **Segmentation examples.** Qualitative evaluation of the segmentation. Colors on segmentation class are assigned: green=non-urban, red=urban, and blue=cloud label. For better visibility, the brightness of RGB images was improved.



(a) Site 1

(b) Site 12

Figure 4.1: **Resulting urban index timeseries.** Resulting urban index compared with ground-truth values based on provided labels.

4.2 Potential use cases

The proposed urban growth index has a variety of use cases, e.g., in business analysis or for intelligence services.

- Companies exploring different locations for expansion may use the urbanization information for comparison - higher urbanization growth suggests an emerging market location. Sometimes the government published data might not be accurate or unavailable in the required resolution. Satellite imagery copes with this problem.

- Intelligence services may use the metric to monitor foreign country activity, especially when figures based on the official sources may not be accurate. Measuring urbanization and other indexes from satellite imagery would provide more reliable information about the economic well-being of e.g., countries under sanctions.

5 Conclusion

An algorithm for urban growth analysis was proposed, implemented, and evaluated. First, available data were analyzed and split into training and testing parts. For a representative evaluation, the training set consists only of data from one location while the other available site is used for testing.

Second, the image segmentation model, which lies in the core of the whole algorithm, was designed, taking into account the limited amount of available data. The model was then tested with various hyper-parameters, and the best performing setting was selected.

Finally, the urban index calculation method based on the resulting image segmentation was proposed and evaluated on the given data.