Learning Aerial Image Segmentation From Online Maps

Pascal Kaiser, Jan Dirk Wegner, Aurélien Lucchi, Martin Jaggi, Thomas Hofmann, and Konrad Schindler, *Senior Member, IEEE*

***Abstract*— This paper deals with semantic segmentation of high-resolution (aerial) images where a semantic class label is assigned to each pixel via supervised classification as a basis for automatic map generation. Recently, deep convolutional neural networks (CNNs) have shown impressive performance and have quickly become the de-facto standard for semantic segmentation, with the added benefit that task-specific feature design is no longer necessary. However, a major downside of deep learning methods is that they are extremely data hungry, thus aggravating the perennial bottleneck of supervised classification, to obtain enough annotated training data. On the other hand, it has been observed that they are rather robust against noise in the training labels. This opens up the intriguing possibility to avoid annotating huge amounts of training data, and instead train the classifier from existing legacy data or crowd-sourced maps that can exhibit high levels of noise. The question addressed in this paper is: can training with large-scale publicly available labels replace a substantial part of the manual labeling effort and still achieve sufficient performance? Such data will inevitably contain a significant portion of errors, but in return virtually unlimited quantities of it are available in larger parts of the world. We adapt a state-of-the-art CNN architecture for semantic segmentation of buildings and roads in aerial images, and compare its performance when using different training data sets, ranging from manually labeled pixel-accurate ground truth of the same city to automatic training data derived from *OpenStreetMap* data from distant locations. We report our results that indicate that satisfying performance can be obtained with significantly less manual annotation effort, by exploiting noisy large-scale training data.**

***Index Terms*— Crowdsourcing, image classification, machine learning, neural networks, supervised learning, terrain mapping, urban areas.**

I. Introduction

HUGE volumes of optical overhead imagery are captured every day with airborne or spaceborne platforms, and that volume is still growing. This “data deluge” makes manual interpretation prohibitive, and hence machine vision must be employed if we want to make any use of the available data. Perhaps the fundamental step of automatic mapping is to assign a semantic class to each pixel, i.e., convert the raw data to a semantically meaningful raster map (which can then be further processed as appropriate with, e.g., vectorization or map generalization techniques). The most popular tool for that task is supervised machine learning. Supervision with human-annotated training data is necessary to inject the task-specific class definitions into the generic statistical analysis. In most cases, reference data for classifier training are generated manually for each new project, which is a time-consuming and costly process. Manual annotation must be repeated every time the task, the geographic location, the sensor characteristics, or the imaging conditions change, and hence the process scales poorly. In this paper, we explore the tradeoff between the following:

1) pixel-accurate but small-scale ground truth available;
2) less accurate reference data that are readily available in arbitrary quantities, at no cost.

For our study, we make use of online map data from *OpenStreetMap* [1]–[3] (OSM, http://www.openstreetmap.org) to automatically derive weakly labeled training data for three classes, *buildings*, *roads*, and *background* (i.e., all others). These data are typically collected using two main sources.

1) Volunteers collect OSM data either *in situ* with GPS trackers or by manually digitizing very high resolution (VHR) aerial or satellite images that have been donated.
2) National mapping agencies donate their data to OSM to make it available to a wider public.

Since OSM is generated by volunteers, our approach can be seen as a form of crowd-sourced data annotation; but other existing map databases, e.g., legacy data within a mapping agency, could also be used.

As image data for our study, we employ high-resolution RGB orthophotographs from *Google Maps*,[1] since we could not easily get access to comparable amounts of other high-resolution imagery [> 100 km^2 at $\approx$ 10-cm ground sampling distance (GSD)].

Clearly, these types of training data will be less accurate. Sources of errors include coregistration errors, e.g., in our case, OSM polygons and Google images were independently geo-referenced; limitations of the data format, e.g., OSM only has road centerlines and category, but no road boundaries; temporal changes not depicted in outdated map or image data; or simply sloppy annotations, not only because of a lack of training or motivation, but also because the use cases of most OSM users require not even meter-level accuracy.

Our study is driven by the following hypotheses.

1) The sheer volume of training data can possibly compensate for the lower accuracy (if used with an appropriate robust learning method).

Manuscript received January 30, 2017; revised April 11, 2017 and May 29, 2017; accepted June 19, 2017. *(Corresponding author: Jan Dirk Wegner.)*

The authors are with ETH Zürich, 8093 Zürich, Switzerland (e-mail: jan.wegner@geod.baug.ethz.ch).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2017.2719738

[1] specifications of Google Maps data can be found at https://support.google.com/mapcontentpartners/answer/144284?hl=en

2) The large variety present in very large training sets (e.g., spanning multiple different cities) could potentially improve the classifier's ability to generalize to new unseen locations.
3) Even if high-quality training data are available, the large volume of additional training data could potentially improve the classification.
4) If low-accuracy large-scale training data help, then it may also allow one to substitute a large portion of the manually annotated high-quality data.

We investigate these hypotheses when using deep convolutional neural networks (CNNs). Deep networks are at present the top-performing method for high-resolution semantic labeling and are therefore the most appropriate choice for our study.[2] At the same time, they also fulfill the other requirements for our study: they are data hungry and robust to label noise [4]. And they make manual feature design somewhat obsolete: once training data are available, retraining for different sensor types or imaging conditions is fully automatic, without scene-specific user interaction such as feature definition or preprocessing. We adopt a variant of the fully convolutional network (FCN) [5], and explore the potential of combining end-to-end trained deep networks with massive amounts of noisy OSM labels. We evaluate the extreme variant of our approach, without any manual labeling, on three major cities (Chicago, Paris, and Zurich) with different urban structures. Since quantitative evaluations on these large data sets are limited by the inaccuracy of the labels, which is also present in the test sets, we also perform experiments for a smaller data set from the city of Potsdam. There, high-precision manually annotated ground truth is available, which allows us to compare different levels of project-specific input, including the baseline where only manually labeled training data are used, the extreme case of only automatically generated training labels, and variants in between. We also assess the models' capabilities regarding generalization and transfer learning between unseen geographic locations.

We find in this paper that training on noisy labels does work well, but only with substantially larger training sets. Whereas with small training sets (≈ 2 km^2), it does not reach the performance of hand-labeled pixel-accurate training data. Moreover, even in the presence of high-quality training data, massive OSM labels further improve the classifier, and hence can be used to significantly reduce the manual labeling efforts. According to our experiments, the differences are really due to the training labels, since segmentation performance of OSM labels is stable across different image sets of the same scene.

For practical reasons, our study is limited to buildings and roads, which are available from OSM, and to RGB images from Google Maps, subject to unknown radiometric manipulations. We hope that similar studies will also be performed with the vast archives of proprietary image and map data held by state mapping authorities and commercial satellite providers. Finally, this is a step in a journey that will ultimately bring us closer to the utopian vision that a whole range of mapping tasks no longer need user input, but can be completely automated by the world wide Web.

II. Related Work

There is a huge literature about semantic segmentation in remote sensing. A large part deals with rather low-resolution satellite images, whereas our work in this paper deals with VHR aerial images (see [6] for an overview).

Aerial data with a ground sampling distance GSD ≤ 20 cm contains rich details about urban objects such as roads, buildings, trees, and cars, and is a standard source for urban mapping projects. Since urban environments are designed by humans according to relatively stable design constraints, early work attempted to construct object descriptors via sets of rules, most prominently for building detection in 2-D [7], [8] or in 3-D [9]–[11], and for road extraction [12]–[14]. A general limitation of hierarchical rule systems, be they top-down or bottom-up, is poor generalization across different city layouts. Hard thresholds at early stages tend to delete information that can hardly be recovered later, and hard-coded expert knowledge often misses important evidence that is less obvious to the human observer.

Machine learning thus aims to learn classification rules directly from the data. As local evidence, conventional classifiers are fed with raw pixel intensities, simple arithmetic combinations such as vegetation indices, and different statistics or filter responses that describe the local image texture [15]–[17]. An alternative is to precompute a large redundant set of local features for training and let a discriminative classifier (e.g., boosting and random forest) select the optimal subset [18]–[21] for the task.

More global object knowledge that cannot be learned from local pixel features can be introduced via probabilistic priors. Two related probabilistic frameworks have been successfully applied to this task, marked point processes (MPPs) and graphical models. For example, [22] and [23] formulate MPPs that explicitly model road network topologies, while [24] use a similar approach to extract building footprints. MPPs rely on object primitives like lines or rectangles that are matched to the image data by sampling. Even if data driven [25], such Monte Carlo sampling has high computational cost and does not always find good configurations. Graphical models provide similar modeling flexibility, but in general also lead to hard optimization problems. For restricted cases (e.g., submodular objective functions), efficient optimizers exist. Although there is a large body of literature that aims to tailor conditional random fields for object extraction in computer vision and remote sensing, relatively few authors tackle semantic segmentation in urban scenes (see [26]–[30]).

Given the difficulty of modeling high-level correlations, much effort has gone into improving the local evidence by finding more discriminative object features [21], [31], [32]. The resulting feature vectors are fed to a standard classifier (e.g., decision trees or support vector machines) to infer probabilities per object category. Some authors invest a lot of efforts to reduce the dimension of the feature space to

[2] All top-performing methods on big benchmarks are CNN variants, both in generic computer vision, e.g., the *Pascal VOC Challenge*, http://host.robots.ox.ac.uk/pascal/VOC/, and in remote sensing, e.g., the ISPRS semantic labeling challenge, http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html

a maximally discriminative subset (see [33]–[36]), although this seems to have only limited effect—at least with modern discriminative classifiers.

Deep neural networks do not require a separate feature definition step, but instead learn the most discriminative feature set for a given data set and task directly from raw images. They go back to [37] and [38], but at the time were limited by a lack of computing power and training data. After their comeback in the 2012 ImageNet challenge [39], [40], deep learning approaches, and in particular deep CNNs, have achieved impressive results for diverse image analysis tasks. State-of-the-art network architectures (see [41]) have many (often 10–20, but up to >100) layers of local filters and thus large receptive fields in the deep layers, which makes it possible to learn complex local-to-global (nonlinear) object representations and long-range contextual relations directly from raw image data. An important property of deep CNNs is that both training and inference are easily parallelizable, especially on GPUs, and thus scale to millions of training and testing images.

Quickly, CNNs were also applied to semantic segmentation of images [42]. Our approach in this paper is based on the FCN architecture of [5], which returns a structured spatially explicit label image (rather than a global image label). While spatial aggregation is nevertheless required to represent context, FCNs also include in-network upsampling back to the resolution of the original image. They have already been successfully applied to semantic segmentation of aerial images (see [43]–[45]). In fact, the top performers on the ISPRS semantic segmentation benchmark all use CNNs. We note that (nonconvolutional) deep networks in conjunction with OSM labels have also been applied for patch-based road extraction in overhead images of ≈ 1 m GSD at large scale [46], [47]. More recently, Máttyus *et al.* [48] combine OSM data with aerial images to augment maps with additional information from imagery like road widths. They design a sophisticated random field to probabilistically combine various sources of road evidence, for instance, cars, to estimate road widths at global scale using OSM and aerial images.

To the best of our knowledge, only two works have made attempts to investigate how results of CNNs trained on large-scale OSM labels can be fine-tuned to achieve more accurate results for labeling remote sensing images [49], [50]. However, we are not aware of any large-scale, systematic, comparative, and quantitative study that investigates using large-scale training labels from inaccurate map data for semantic segmentation of aerial images.

III. Methods

We first describe our straightforward approach to generate training data automatically from OSM, and then give technical details about the employed FCN architecture and the training procedure used to train our model.

A. Generation of Training Data

We use a simple automatic approach to generate data sets of VHR aerial images in RGB format and corresponding labels for classes *building*, *road*, and *background*. Aerial images are downloaded from Google Maps, and geographic coordinates of buildings and roads are downloaded from OSM. We prefer to use OSM maps instead of Google Maps, because the latter can only be downloaded as raster images.[3] OSM data can be accessed and manipulated in vector format, and each object type comes with meta data and identifiers that allow straightforward filtering. Regarding coregistration, we find that OSM and Google Maps align relatively well, even though they have been acquired and processed separately.[4] Most local misalignments are caused by facades of high buildings that overlap with roads or background due to perspective effects. It is apparent that in our test areas Google provides orthophotographs rectified with respect to a bare earth digital terrain model (DTM), not "true" orthophotographs rectified with a digital surface model (DSM). According to our own measurements on a subset of the data, this effect is relatively mild, generally < 10 pixels displacement. We found that this does not introduce major errors as long as there are no high-rise buildings. It may be more problematic for extreme scenes such as Singapore or Manhattan.

To generate pixel-wise label maps, the geographic coordinates of OSM building corners and road center lines are transformed to pixel coordinates. For each building, a polygon through the corner points is plotted at the corresponding image location. For roads, the situation is slightly more complex. OSM provides only coordinates of road center lines, but no precise road widths. There is, however, a road category label ("highway tag") for most roads. We determined an average road width for each category on a small subset of the data, and validated it on a larger subset (manually, one-off). This simple strategy works reasonably well, with a mean error of $\approx$11 pixels for the road boundary, compared with $\approx$100 pixels of road width.[5] In (very rare) cases where the *ad hoc* procedure produced label collisions, pixels claimed by both building and road were assigned to buildings. Pixels neither labeled building nor road form the background class. Examples of images overlaid with automatically generated OSM labels are shown in Fig. 1.

B. Neural Network Architecture

We use a variant of FCNs in this paper (see Fig. 2). Following the standard neural network concept, transformations are ordered in sequential layers that gradually transform the pixel values to label probabilities. Most layers implement learned convolution filters, where each neuron at level l takes its input values only from a fixed-size spatially localized window $\mathcal{W}$ in the previous layer $(l-1)$, and outputs a vector of differently weighted sums of those values, $c^l = \sum_{i \in \mathcal{W}} w_i c_i^{l-1}$. Weights w_i are shared across all neurons of a layer, which reflects the shift invariance of the image signal and drastically

[3] Note that some national mapping agencies also provide publicly available map and other geo-data, e.g., the USGS national map program: https://nationalmap.gov/

[4] Note that it is technically possible to obtain world coordinates of objects in Google Maps and enter those into OSM, and this might in practice also be done to some extent. However, OSM explicitly asks users not to do that.

[5] Average deviation based on ten random samples of Potsdam, Chicago, Paris, and Zurich.

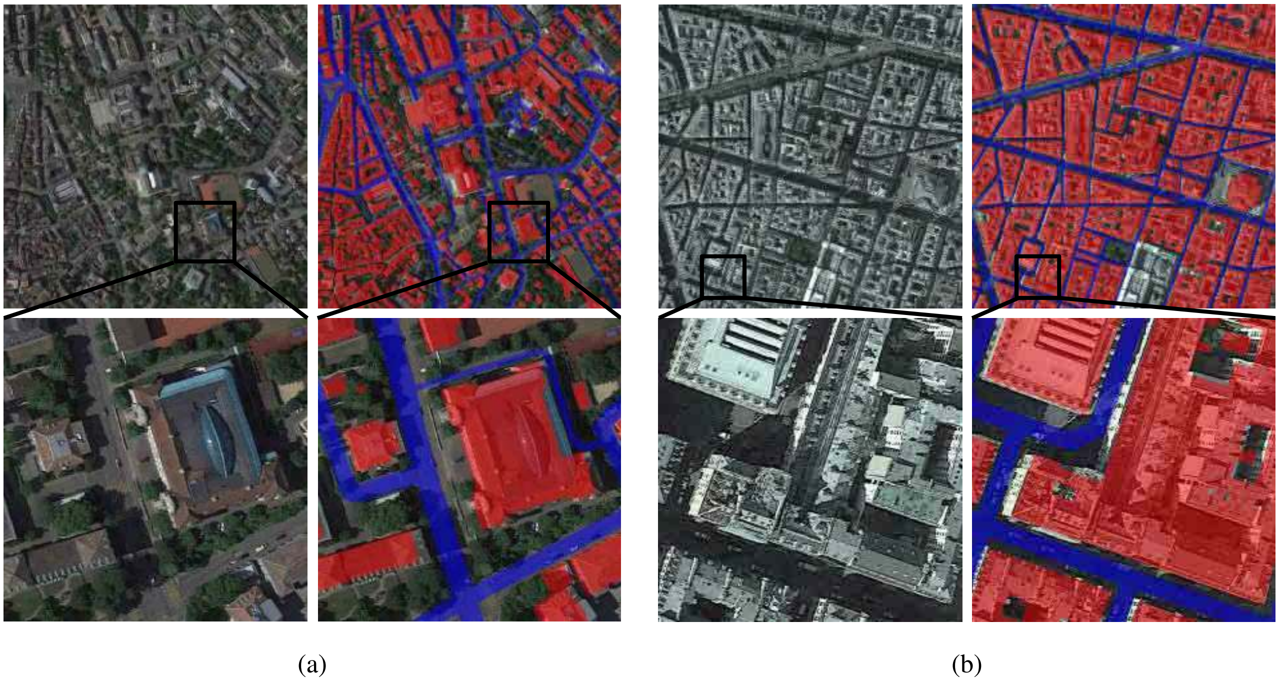

Fig. 1. Example of OSM labels overlaid with Google Maps images for (a) Zurich and (b) Paris. (Left) Aerial image and a magnified detail. (Right) Same images overlaid with building (red) and road (blue) labels. Background is transparent in the label map.

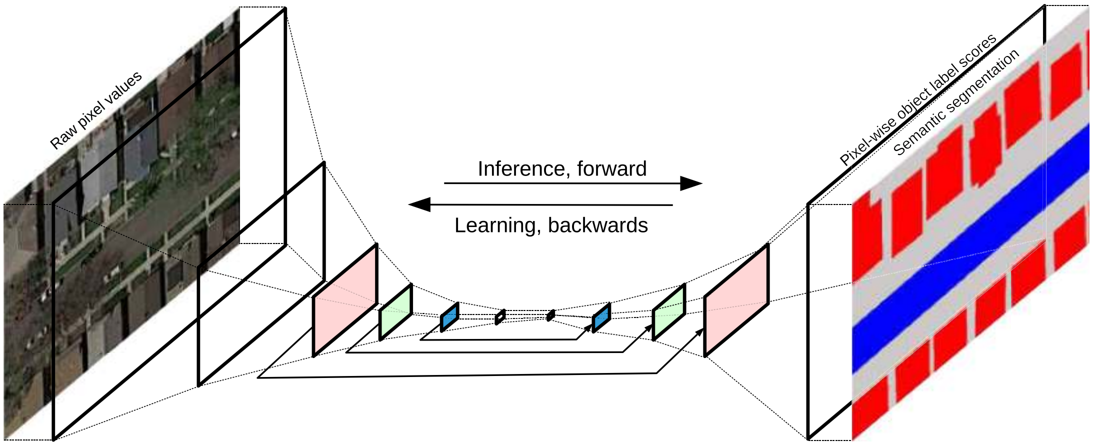

Fig. 2. Conceptual illustration of the data flow through our variant of an FCN, which is used for the semantic segmentation of aerial images. Three skip connections are highlighted by pale red, pale green, and pale blue, respectively. Note that we added a third (pale red) skip connection in addition to the original ones (pale green and pale blue) of [5].

reduces the number of parameters. Each convolutional layer is followed by a rectified linear unit ($ReLU$) $c^l_{\text{rec}} = \max(0, c^l)$, which simply truncates all negative values to 0 and leaves positive values unchanged [51].[6] Convolutional layers are interspersed with max-pooling layers that downsample the image and retain only the maximum value inside a (2×2) neighborhood. The downsampling increases the receptive field of subsequent convolutions, and lets the network learn correlations over a larger spatial context. Moreover, max-pooling achieves local translation invariance at object level. The outputs of the last convolutional layers (which are very big to capture global context, equivalent to a fully connected layer of standard CNNs) is converted to a vector of scores for the three target classes. These score maps are of low resolution, and hence they are gradually upsampled again with convolutional layers using a stride of only (12) pixel.[7] Repeated downsampling causes a loss of high-frequency content, which leads to blurry boundaries that are undesirable for pixel-wise semantic segmentation. To counter this effect, feature maps at intermediate layers are merged back in during upsampling (the so-called "skip connections," see Fig. 2). The final full-resolution score maps are then converted to label probabilities with the $softmax$ function.

[6] Other nonlinearities are sometimes used, but $ReLU$ has been shown to facilitate training (backpropagation) and has become the de-facto standard.

[7] This operation is done by layers that are usually called "deconvolution layers" in [5] (and also in Fig. 3) although the use of this terminology has been criticized since most implementations do not perform a real deconvolution but rather a transposed convolution.

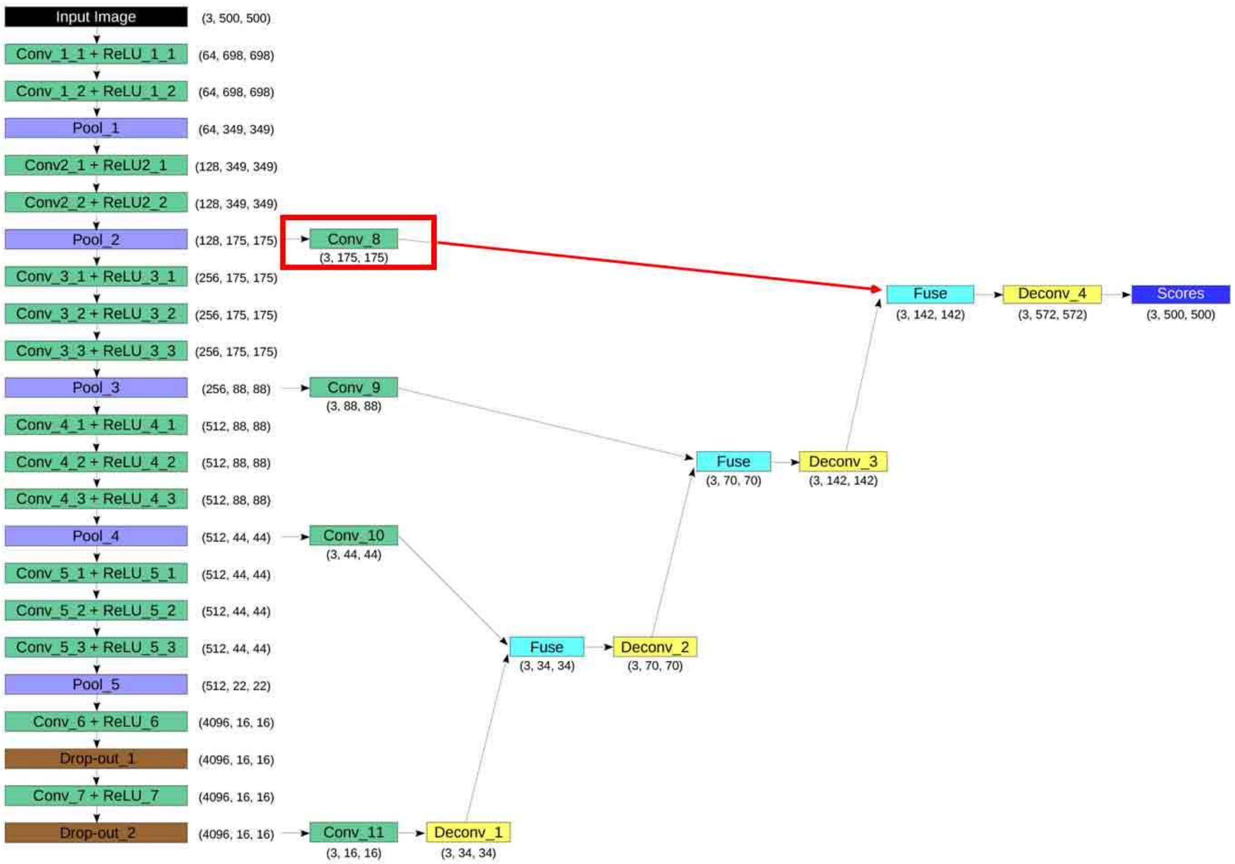
Input Image (3, 500, 500)
Conv_1_1 + ReLU_1_1 (64, 698, 698)
Conv_1_2 + ReLU_1_2 (64, 698, 698)
Pool_1 (64, 349, 349)
Conv2_1 + ReLU2_1 (128, 349, 349)
Conv2_2 + ReLU2_2 (128, 349, 349)
Pool_2 (128, 175, 175)
Conv_3_1 + ReLU_3_1 (256, 175, 175)
Conv_3_2 + ReLU_3_2 (256, 175, 175)
Conv_3_3 + ReLU_3_3 (256, 175, 175)
Pool_3 (256, 88, 88)
Conv_4_1 + ReLU_4_1 (512, 88, 88)
Conv_4_2 + ReLU_4_2 (512, 88, 88)
Conv_4_3 + ReLU_4_3 (512, 88, 88)
Pool_4 (512, 44, 44)
Conv_5_1 + ReLU_5_1 (512, 44, 44)
Conv_5_2 + ReLU_5_2 (512, 44, 44)
Conv_5_3 + ReLU_5_3 (512, 44, 44)
Pool_5 (512, 22, 22)
Conv_6 + ReLU_6 (4096, 16, 16)
Drop-out_1 (4096, 16, 16)
Conv_7 + ReLU_7 (4096, 16, 16)
Drop-out_2 (4096, 16, 16)
Conv_8 (3, 175, 175)
Conv_9 (3, 88, 88)
Conv_10 (3, 44, 44)
Conv_11 (3, 16, 16)
Deconv_1 (3, 34, 34)
Fuse (3, 34, 34)
Deconv_2 (3, 70, 70)
Fuse (3, 70, 70)
Deconv_3 (3, 142, 142)
Fuse (3, 142, 142)
Deconv_4 (3, 572, 572)
Scores (3, 500, 500)

Fig. 3. Our FCN architecture, which adds one more skip connection (after Pool_2, shown red) to the original model of [5]. Neurons form a 3-D structure per layer: dimensions are written in brackets, where the first number indicates the amount of feature channels, and second and third represent spatial dimensions.

C. Implementation Details

The FCN we use is an adaptation of the architecture proposed in [5], which itself is largely based on the VGG-16 network architecture [41]. In our implementation, we slightly modify the original FCN and introduce a third skip connection (marked red in Fig. 2), to preserve even finer image details. We found that the original architecture, which has two skip connections after Pool_3 and Pool_4 (see Fig. 3), was still not delivering sufficiently sharp edges. The additional higher resolution skip connection consistently improved the results for our data (see Section IV-B). Note that adding the third skip connection does not increase the total number of parameters but, on the contrary, slightly reduces it ([5]: $134'277'737$, ours: $134'276'540$; the small difference is due to the decomposition of the final upsampling kernel into two smaller ones).

D. Training

All model parameters are learned by minimizing a multinomial logistic loss, summed over the entire 500 × 500 pixel patch that serves as input to the FCN. Prior to training/inference, intensity distributions are centered independently per patch by subtracting the mean, separately for each channel (RGB).

All models are trained with stochastic gradient descent with a momentum of 0.9, and minibatch size of one image. Learning rates always start from 5×10^{-9} and are reduced by a factor of ten twice when the loss and average F_1 scores stopped improving. The learning rates for biases of convolutional layers were doubled with respect to learning rates of the filter weights. Weight decay was set to 5×10^{-4}, and dropout probability for neurons in layers ReLU_6 and ReLU_7 was always 0.5.

Training was run until the average F_1-score on the validation data set stopped improving, which took between 45 000 and 140 000 iterations (3.5–6.5 epochs). Weights were initialized as in [52], except for experiments with pretrained weights. It is a common practice in deep learning to publish pretrained models together with source code and paper, to ease repeatability of results and to help others avoid training from scratch. Starting from pretrained models, even if these have been trained on a completely different image data set, often improves performance, because low-level features like contrast edges and blobs learned in early network layers are very similar across different kinds of images.

We will use two different forms of pretraining. Either we rely on weights previously learned on the Pascal VOC benchmark [53] (made available by Long *et al.* [5]), or we pretrain ourselves with OSM data. In Section IV, it is always specified whether we use VOC, OSM, or no pretraining at all.

IV. Experiments

We present extensive experiments on four large data sets of different cities to explore the following scenarios.

1) *Complete Substitution:* Can semantic segmentation be learned without any manual labeling? What performance

can be achieved using only noisy labels gleaned from OSM?

2) *Augmentation:* Will pretraining with large-scale OSM data and publicly available images improve the segmentation of a project-specific data set of independently acquired images and labels?

3) *Partial Substitution:* Can pretraining with large-scale OSM labels replace a substantial part of the manual labeling effort? Phrased differently, can a generic model learned from OSM be adapted to a specific location and data source with only little dedicated training data?

We provide a summary of the results and explicit answers to these questions at the very end of this section. Note that all experiments are designed to investigate different aspects of the hypotheses made in the introduction. We briefly remind and thoroughly validate all hypotheses based on results of our experiments in the conclusion.

A. Data Sets

Four large data sets were downloaded from Google Maps and OSM, for the cities of Chicago, Paris, Zurich, and Berlin. In addition, we also downloaded a somewhat smaller data set for the city of Potsdam. For this location, a separate image set and high-accuracy ground truth are available from the ISPRS semantic labeling benchmark [54]. Table I specifies the coverage (surface area), number of pixels, and GSD of each data set. Example images and segmentation maps of Paris and Zurich are shown in Fig. 1. In Fig. 4, we show the full extent of the Potsdam scene, dictated by the available images and ground truth in the ISPRS benchmark. OSM maps and aerial images from Google Maps were downloaded and cut to cover exactly the same region to ensure a meaningful comparison—this meant, however, that the data set is an order of magnitude smaller than what we call "large scale" for the other cities. The ISPRS data set includes a portion [images x_13, x_14, x_15 in Fig. 4(right)], for which the ground truth is withheld to serve as test set for benchmark submissions. We thus use images 2_12, 6_10, and 7_11 as test set, and the remaining ones for training. The three test images were selected to cover different levels of urban density and architectural layout. This train-test split corresponds to 1.89 km^2 of training data, respectively, 0.27 km^2 of test data.

The ISPRS semantic labeling challenge aims at land-cover classification, whereas OSM represents land use. In particular, the benchmark ground truth does not have a label *street*, but instead uses a broader class *impervious surfaces*, also comprising sidewalks, tarmacked courtyards, and so on. Furthermore, it labels overhanging tree canopies that occlude parts of the impervious ground (including streets) as *tree*, whereas *streets* in the OSM labels include pixels under trees. Moreover, images in the ISPRS benchmark are "true" orthophotographs rectified with a DSM that includes buildings, whereas Google images are conventional orthophotographs, corrected only for terrain-induced distortions with a DTM. Building facades remain visible and roofs are shifted from the true footprint. To facilitate a meaningful comparison, we have manually relabeled the ISPRS ground truth to our target categories *street* and *background*, matching the land-use definitions extracted from OSM. The category *building* of the benchmark ground truth remains unchanged. To allow for a direct and fair comparison, we downsample the ISPRS Potsdam data, which comes at a GSD of 5 cm, to the same GSD as the Potsdam–Google data (9.1 cm).

TABLE I

STATISTICS OF THE DATA SETS USED IN THE EXPERIMENTS. NOTE THAT WE DOWNSAMPLED THE ORIGINAL POTSDAM–ISPRS (GSD=5 cm) TO THE RESOLUTION OF THE POTSDAM–GOOGLE DATA (GSD=9.1 cm) FOR ALL EXPERIMENTS

	Coverage	No. of pixels	GSD
Chicago	$50.6\ km^2$	4.1×10^9	$11.1\ cm$
Paris	$60.3\ km^2$	6.3×10^9	$9.8\ cm$
Zurich	$36.2\ km^2$	3.5×10^9	$10.1\ cm$
Berlin	$10.6\ km^2$	1.28×10^9	$9.1\ cm$
Potsdam-Google	$2.16\ km^2$	2.60×10^8	$9.1\ cm$
Potsdam-ISPRS	$2.16\ km^2$	2.60×10^8	$9.1\ cm$

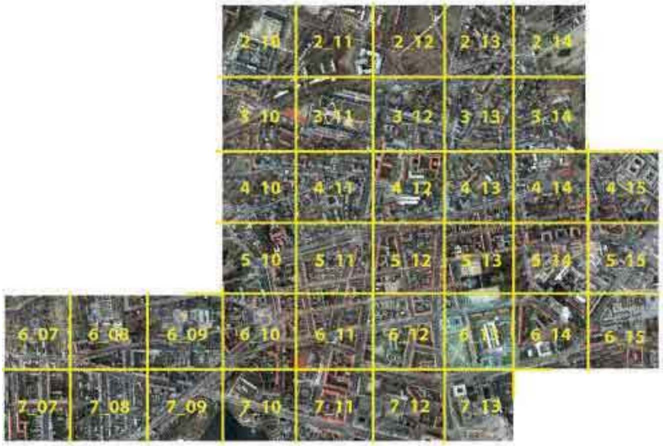

Fig. 4. Overview of the ISPRS Potsdam data set. The aerial images shown are those provided by the ISPRS benchmark [54].

For all data sets, we cut the aerial images as well as the corresponding label maps into nonoverlapping tiles of size 500 × 500 pixels. The size was determined in preliminary experiments, to include sufficient geographical context while keeping FCN training and prediction efficient on a normal single-GPU desktop machine. Each data set is split into mutually exclusive training, validation, and test regions. During training, we monitor the loss (objective function) not only on the training set, but also on the validation set to prevent overfitting.[8]

B. Results and Discussion

First, we validate our modifications of the FCN architecture, by comparing it with the original model of [5]. As error

[8] This is standard practice when training deep neural networks.

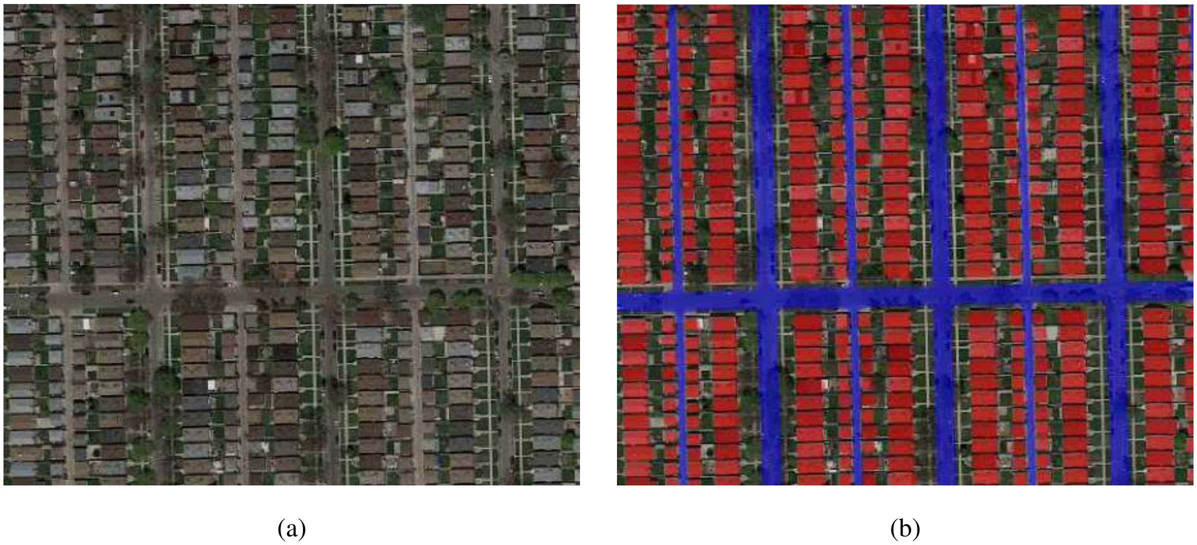

Fig. 5. FCN trained on Google Maps imagery and OSM labels of Chicago. (a) Original aerial image. (b) Overlaid with classification result.

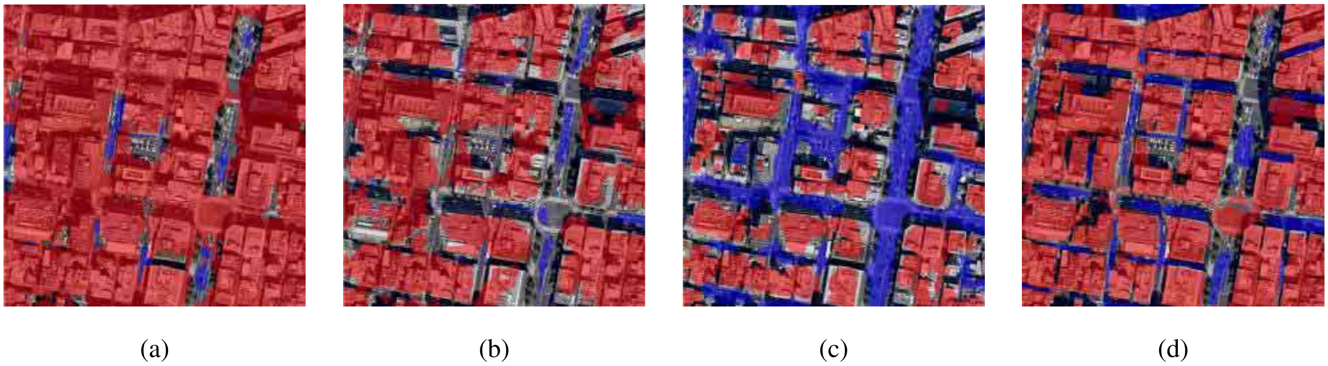

Fig. 6. Classification results and average F^1-scores of the Tokyo scene with a model trained on (a) Chicago (F^1: 0.485), (b) Paris (F^1: 0.521), (c) Zurich (F^1: 0.581), and (d) all three (F^1: 0.644).

metrics, we always compute precision, recall, and F_1-score, per class as well as averaged over all three classes. Precision is defined as the fraction of predicted labels that are correct with respect to ground truth, and recall is the fraction of true labels that are correctly predicted. The F_1-score is the harmonic mean between precision and recall. It combines the two competing goals into a scalar metric and is widely used to assess semantic segmentation. It also serves as our primary error measure. Quantitative results are shown in Table II, and an example result for Chicago is shown in Fig. 5. Our architecture with the additional early skip connection outperforms its counterpart slightly but consistently on average, albeit only by 1 percent point. Note that this performance improvement also comes with the benefit of lower runtimes. Our architecture consistently needs $\geq$ 30% less time for training compared with the original architecture of [5] (see Table II).

Another interesting finding is in terms of transfer learning, in the sense that training a model over multiple cities, with both different global scene structure and different object appearance, can help better predict a new previously unseen city. This again emphasizes the improved generalization ability that benefits from the increased amount of weak labels, in contrast to traditional supervised approaches with smaller label sets. We train the FCN on Zurich, Paris, and Chicago and predict Tokyo. We compare the results with those from training on only a single city (Fig. 6). It turns out that training over multiple different cities helps the model to find a more general "mean" representation of what a city looks like. Generalizing from a single city to Tokyo clearly performs worse [Fig. 6(a)–(c)] than generalizing from several different ones [Fig. 6(d)]. This indicates that FCNs are indeed able to learn location-specific urbanistic and architectural patterns; but also that supervision with a sufficiently diverse training set mitigates this effect and still lets the system learn more global generic patterns that support semantic segmentation in different geographic regions not seen at all during training.

For experiments on the ISPRS Potsdam data set, we first compute three baselines. For an overview of the setup of all experiments described in the following, please refer to Table III, whereas quantitative results are given in Table IV.

1) Baseline With ISPRS Data: First, we follow the conventional semantic segmentation baseline and apply our FCN model to the ISPRS benchmark to establish a baseline with conventional hand-labeled ground truth. As a training set of

TABLE II

COMPARISON BETWEEN OUR ADAPTED FCN AND THE ORIGINAL ARCHITECTURE OF [5], FOR THREE LARGE CITY DATA SETS. NUMBERS IN BRACKETS INDICATE TRAINING TIMES FOR THE ORIGINAL FCN ARCHITECTURE OF [5] AND OURS FOR ALL DATA SETS IF TRAINED FROM SCRATCH WITHOUT ANY PRETRAINING TO FACILITATE A FAIR COMPARISON (ON A STANDARD STAND-ALONE PC WITH i7 CPU, 2.7 GHz, 64-GB RAM AND NVIDIA TITAN-X GPU WITH 12-GB RAM)

	Chicago		Paris		Zurich	
	[5] (15.7h)	Ours (10.5h)	[5] (18.3h)	Ours (7.6h)	[5] (15.5h)	Ours (6.2h)
F_1 average	0.840	**0.855**	0.774	**0.776**	0.804	**0.810**
F_1 building	0.823	**0.837**	0.821	**0.822**	**0.824**	0.823
F_1 road	0.821	**0.843**	0.741	**0.746**	0.695	0.707
F_1 background	0.849	**0.861**	**0.754**	**0.754**	**0.894**	0.891

TABLE III

OVERVIEW OF DIFFERENT EXPERIMENTAL SETUPS WE USE TO VALIDATE OUR HYPOTHESIS MADE IN SECTION I. WE ABBREVIATE BERLIN (B), ZURICH (Z), CHICAGO (C), AND POTSDAM (P). ALL ENTRIES REFER TO THE TRAINING SETUP EXCEPT THE MOST RIGHT COLUMN, WHICH INDICATES DATA USED FOR TESTING. QUANTITATIVE RESULTS FOR ALL EXPERIMENTS ARE GIVEN IN TABLE IV

		# ISPRS images	pre-training	OSM	Google images	test data
Ia	ISPRS baseline	3	no	-	-	ISPRS
Ib	ISPRS baseline pre-trained	3	yes	-	-	ISPRS
II	ISPRS gold standard	21	yes & no	-	-	ISPRS
IIIa	Google/OSM baseline Potsdam	-	no	P	P	OSM+Google
IIIb	Google/OSM baseline Potsdam+Berlin	-	yes	P, B	P, B	OSM+Google
IV	Complete substitution	21	no	P	-	ISPRS
V	Augmentation	21	yes	B, Z, C, P	B, Z, C, P	ISPRS
VI	Partial substitution	3	yes	B, Z, C, P	B, Z, C, P	ISPRS

TABLE IV

RESULTS OF EXPERIMENTS WITH THE POTSDAM DATA SET. THE THREE LEFT COLUMNS ARE AVERAGE VALUES OVER ALL CLASSES AND THE RIGHT THREE COLUMNS GIVE PER CLASS F_1-SCORES. THE *BEST RESULTS* ACROSS ALL VARIANTS ARE WRITTEN IN BOLD, THE SECOND BEST RESULTS ARE UNDERLINED, AND THE *THIRD BEST RESULTS* ARE WRITTEN IN ITALIC. ALL EXPERIMENTS (AND RUNTIMES) WERE COMPUTED ON A STANDARD STAND-ALONE PC WITH i7 CPU, 2.7 GHz, 64-GB RAM AND NVIDIA TITAN-X GPU WITH 12-GB RAM. LIKE IN TABLE III, P IS SHORT FOR POTSDAM, WHEREAS B IS SHORT FOR BERLIN

		av. F_1	av. Precision	av. Recall	F_1 Building	F_1 Road	F_1 Background	train time [h]
Ia	ISPRS baseline	0.764	0.835	0.704	0.793	0.499	0.883	16
Ib	ISPRS baseline pre-trained	0.809	0.853	0.770	0.830	0.636	*0.904*	16
II	ISPRS gold standard	0.874	**0.910**	0.841	**0.913**	0.764	**0.923**	16
IIIa	Google/OSM baseline P	0.777	0.799	0.756	0.832	0.631	0.845	16
IIIb	Google/OSM baseline P+B	0.797	0.819	0.776	0.828	0.698	0.858	32
IV	Complete substitution	0.779	0.801	0.758	0.796	0.667	0.860	16
V	Augmentation	**0.884**	0.898	**0.870**	0.900	**0.825**	0.922	78
VI	Partial substitution	*0.837*	*0.860*	*0.816*	*0.863*	*0.736*	0.899	78

realistic size, we use three completely labeled images from the ISPRS Potsdam benchmark ($3.25 \cdot 10^7$ pixels $\approx$ 27 ha). This setup *Ia* achieves 0.764 average F_1-score over the three classes if we train our FCN from scratch, i.e., weights initialized randomly as in [52] [Fig. 7(a)–(c)]. A widely used practice is to start from a pretrained model that has been learned from a very large data set, especially if the dedicated training data are limited in size. We thus compute baseline *Ib*, where we start from a model trained on the Pascal VOC benchmark and fine-tune on the three ISPRS Potsdam images. As expected, this boosts performance, to 0.809 average F_1-score [Fig. 7(d)–(f)].

2) Gold Standard With ISPRS Data: Second, we repeat the same experiment, but use all of the available training data, i.e., we train on all 21 available training images ($2.275 \cdot 10^8$ pixels$\approx$189 ha). This setup serves as a "gold standard" for what is achievable with the conventional pipeline,

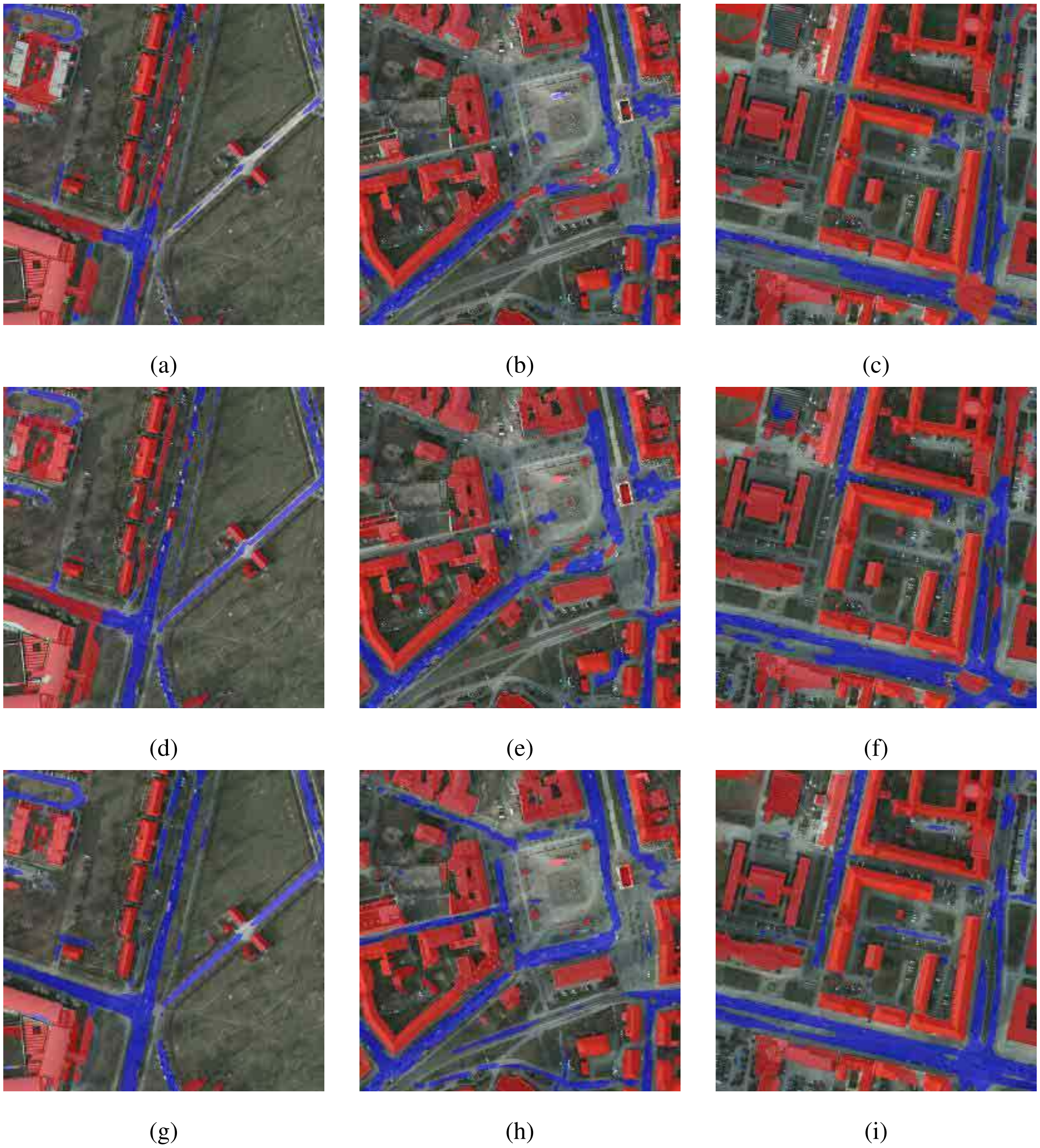

Fig. 7. Baseline experiments. (a)–(c) Baseline *Ia* trained on three ISPRS images without pretraining. (d)–(f) Baseline *Ib* trained on three ISPRS images with pretraining on Pascal VOC. (g)–(i) Gold standard *II* trained on 21 ISPRS images.

given an unusually large amount of costly high-quality training labels. It simulates a project with the luxury of >200 million hand-labeled training pixels over a medium-sized city (which will rarely be the case in practice). It achieves an F_1-score of 0.874 if trained from scratch (Fig. 7). The significant improvement of 11, respectively, 6 percent points, shows that our "standard" baselines *Ia* and *Ib* are still data limited, and can potentially be improved significantly with additional training data. As a sanity check, we also ran the same experiment with all 21 ISPRS images *and* pretraining from Pascal VOC. This marginally increases the average F_1-score to only 0.879.

We note that baseline *II* is not directly comparable with the existing benchmark entries, since we work with a reduced class nomenclature and modified ground truth, and do not evaluate on the undisclosed test set. But it lies in a plausible range, on par with or slightly below the *impervious ground* and *building* results of the competitors, who, unlike us, also use the DSM.

3) Baseline With Google Maps Images and OSM Maps: The next baseline *IIIa* trains on Google aerial images using OSM map data as ground truth. The same 189 ha as in baseline *II* are used for training, and the model achieves an F_1-score of 0.777 if tested on Google aerial images and OSM ground truth [Fig. 8(a)–(c)]. This baseline has been added as a sanity check to verify that the previously observed potential of the open data sources is also confirmed for Potsdam. We point out that the experiment is somewhat problematic and not comparable with baseline *II*, in that it inevitably confounds several effects: the drop in performance may in part be due to the larger amount of noise in the training labels; but further possible reasons include on the one hand the inferior image

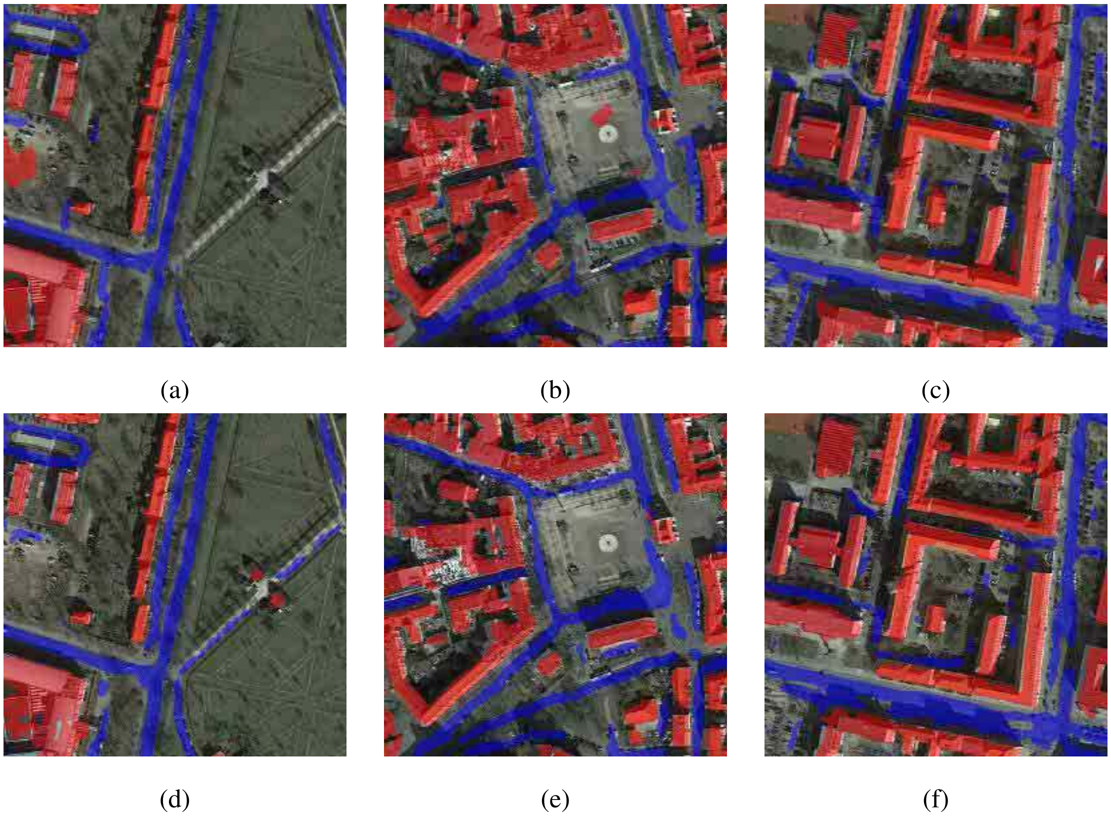

Fig. 8. Baseline experiments. (a)–(c) Baseline *IIIa* with Google Maps images and OSM Maps from only Potsdam. (d)–(f) Baseline *IIIb* with Google Maps images and OSM Maps and training on Potsdam and Berlin.

quality of the Google Maps images [see cast shadows and ortho-rectification artifacts in Fig. 8(b) and (c)] and on the other hand the noise in the OSM-based test labels.[9] Recall that the same setup achieved 0.810 for the architecturally comparable Zurich, and 0.827 for the more schematic layout of Chicago. This suggests that a part of the drop may be attributed to the smaller training set, respectively, that noisy OSM labels should be used in large quantities. To verify this assumption, we repeat the experiment, but greatly extend the training data set by adding the larger city of Berlin, which is immediately adjacent to Potsdam. This baseline *IIIb* increases performance by 2 percent points to 0.797 [Fig. 8(d)–(f)], which is only slightly below performance on Zurich (0.810). It shows that training data size is a crucial factor, and that indeed city-scale (though noisy) training data help to learn better models.

Qualitatively, one can see that the model trained on OSM has a tendency to miss bits of the road, and produces slightly less accurate and blurrier building outlines.

4) Complete Substitution of Manual Labels: Next, we evaluate the extreme setting where we do not have any high-accuracy labels and completely rely on OSM as source of training data. We thus train our FCN on the ISPRS Potsdam images, but use OSM map data as ground truth. The predictions for the ISPRS test images are then evaluated with the manual high-accuracy ground truth from the benchmark. In other words, this experiment quantifies how accurate predictions we can expect if training from OSM labels for a limited project-specific image set: since the ISPRS data set does not provide more images, one cannot augment the training set further, even though a lot of OSM data would be available. This set up achieves an F_1-score of 0.779, beating baseline *Ia* by 1.5 percent points. We conclude that *larger amounts of noisy automatically gleaned training data can indeed completely replace small amounts of highly accurate training data*, saving the associated effort and cost. The result, however, does stay 3 percent points behind baseline *Ib*, which shows that even all of Potsdam is not large enough to replace pretraining with large-scale data, which will be addressed in experiment *VI*. Compared with baseline *II*, i.e., training with equally large quantities of pixel-accurate labels, performance drops by 10 percent points. The visual comparison between baseline *II* in Fig. 7(g)–(i) and *IV* in Fig. 9(a)–(c) shows that buildings are segmented equally well, but roads deteriorate significantly. This is confirmed by the F_1-scores in Table IV. An explanation is the noise in the guessed road width (as also pointed out in [55]) in the training data (≈ 23 pixels on average, for an average road width of ≈ 100 pixels). It leads to washed-out evidence near the road boundaries, which in turn weakens the overall evidence in the case of narrow or weakly supported roads. This effect can be observed visually by comparing probability maps of *II* and *IV* in Fig. 10. Road probabilities appear much sharper at road edges for baseline *II*

[9] We also test the same model on Google aerial images with ISPRS labels, which leads to a slight performance drop to 0.759. This is not surprising, because labels have been acquired based on the ISPRS images and do not fit as accurately to the Google images.

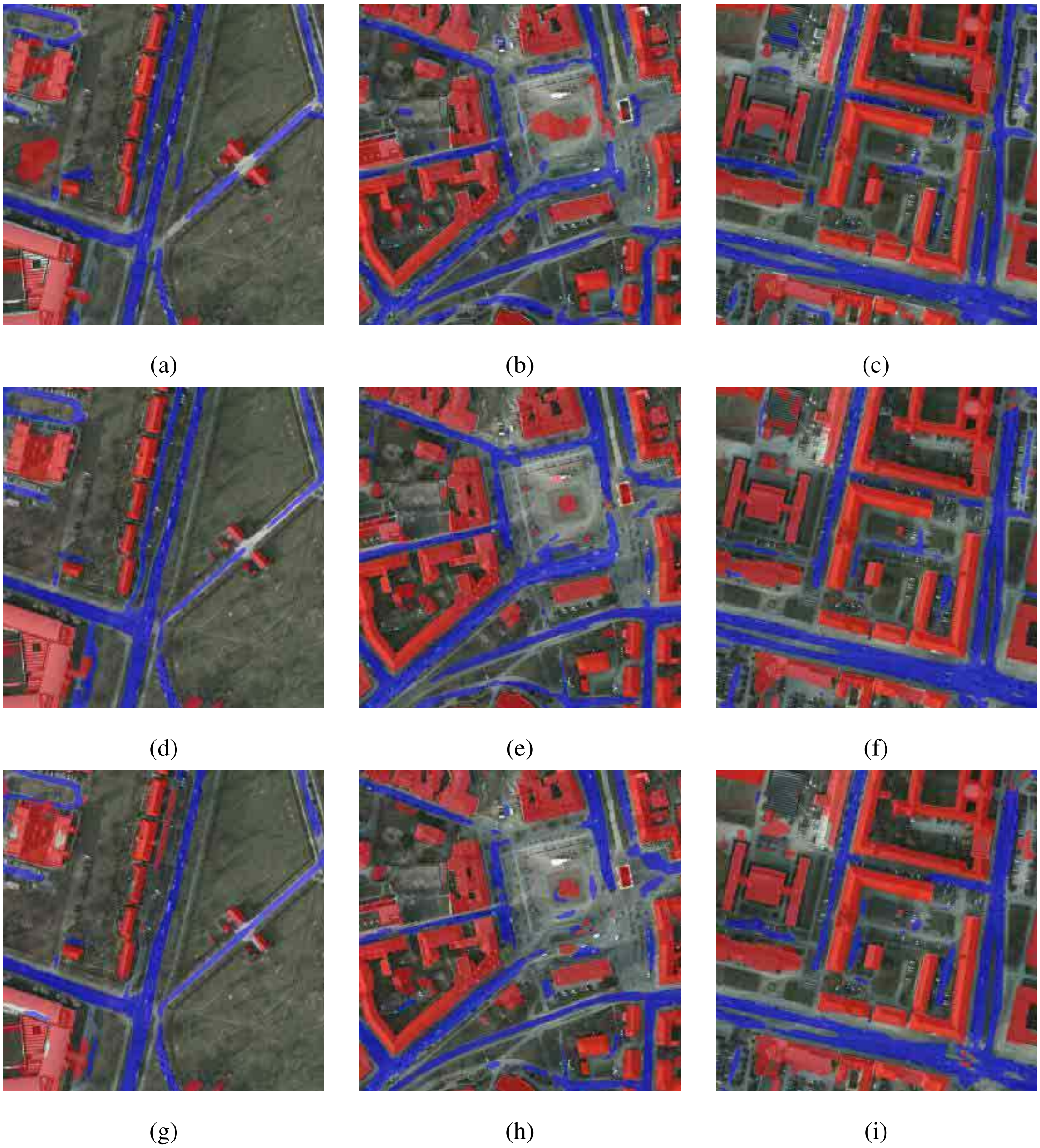

Fig. 9. (a)–(c) Complete substitution (*IV*) of manual labels, trained from scratch on ISPRS images and OSM labels of Potsdam (no pretraining). (d)–(f) Augmentation (*V*) with open data, pretrained on Chicago, Paris, Zurich, and Berlin and retrained on all 21 ISPRS training images with pixel-accurate ground truth. (g)–(i) Partial substitution (*VI*) of manual labels, pretrained on Chicago, Paris, Zurich, and Berlin and retrained on three ISPRS images with pixel-accurate ground truth.

trained with pixel-accurate ISPRS ground truth [Fig. 10(a)–(c)] compared with *IV* trained with noisy OSM ground truth [Fig. 10(d)–(f)].

5) Augmentation With Open Data: With experiment *V*, we aim to assess whether pretraining from even larger amounts of open data from other sites can further improve the gold standard *II*, by providing a sort of “generic background” for the problem, in the spirit of pretrained computer vision models such as VGG [41] or Alexnet [40]. We first train the FCN model on Google/OSM data of Chicago, Paris, Zurich, and Berlin, and use the resulting network weights as initial value, from which the model is tuned for the ISPRS data, using all the 21 training images as in baseline *II*. The pretraining boosts performance, albeit only by 1 percent point. Even if one has a comfortable amount of accurate training data at hand, it appears potentially useful to pretrain with freely available data. In future work, it may be useful to experiment with even larger amounts of open data.

A visual comparison of Figs. 7(g)–(i) and 9(d)–(f) shows small improvements for both the roads and the buildings, in all three tiles. This effect shows up quantitatively with an improvement in F_1-score of the *road* class, which reaches 0.825, up from 0.764 in baseline *II*. On the other hand, buildings are detected equally well, and no further improvement can be noticed. A possible interpretation is that complex network structures with long-range dependencies are hard to learn for the classifier, and thus more training data help. Locally well-defined compact objects of similar shape and appearance are easier to learn, so further training data do not add relevant information.

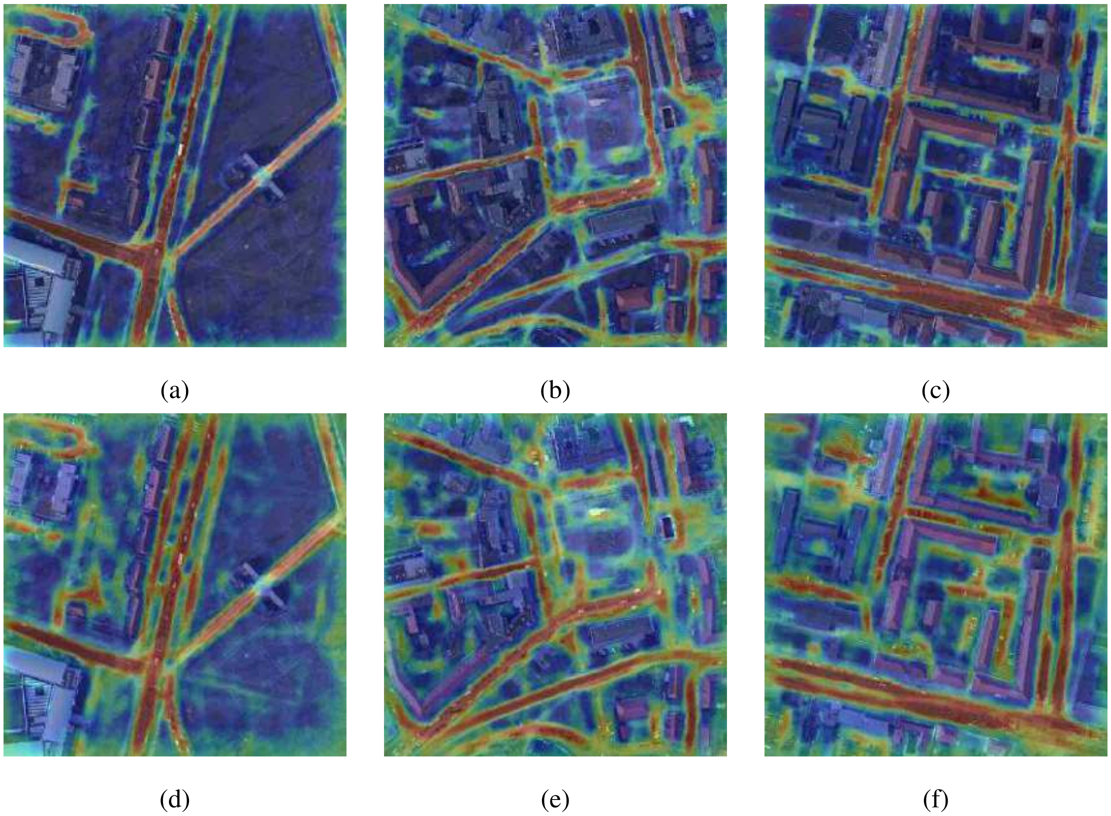

Fig. 10. Probability maps for (a)–(c) road extraction of the gold standard baseline *II* and (d)–(f) complete substitution *IV* without any manual labels. Road probabilities range from red (high) to blue (low).

6) Partial Substitution of Manual Labels: The success of pretraining in previous experiments raises the question—also asked in [50]—of whether one could reduce the annotation effort and use a smaller hand-labeled training set, in conjunction with large-scale OSM labels. An alternative view is as a domain adaptation problem, where the classifier is trained on Google Maps images, and then retargeted to ISPRS images with only few training samples. The hope is that the large amount of OSM training data would already allow the classifier to learn basic aerial image statistics and urban scene structures. Then, only a small additional training set would suffice to adapt it to the different spectral properties. In experiment *VI*, we therefore first train the FCN on the combined Google / OSM data of Chicago, Paris, Zurich, and Berlin. This part is the same as in experiment *V*. Then, we use only the small set of training images and labels from baseline *I* to tune it to the ISPRS images of Potsdam. Performance increases by 7 percent points to 0.837 over baseline *Ia*, where the model is trained from scratch on the same high-accuracy labels. We conclude that if only a limited quantity of high-quality training data is available, pretraining on free data brings even larger relative benefits, and can be recommended as general practice, which is in line with the findings reported in [50].

Importantly, experiment *VI* also outperforms baseline *Ib* by almost 3 percent points, i.e., pretraining on open geo-spatial and map data is more effective than using a generic model pretrained on random Web images from Pascal VOC. While pretraining is nowadays a standard practice, we go one step further and pretrain *with aerial images and the correct set of output labels*, generated automatically from free map data.

Compared with the gold standard baseline *II*, the performance is $\approx$ 4 percent points lower (0.837 versus 0.874). In other words, fine-tuning with a limited quantity of problem-specific high-accuracy labels compensates for a large portion ($\approx 65\%$) of the loss between experiments *II* and *IV*, with only 15 % of the labeling effort. Relative to *II*, buildings degrade most (0.863 versus 0.913). This can possibly be attributed to the different appearance of buildings due to different orthorectification. Recall that Google images were rectified with a DTM and are thus geometrically distorted, with partially visible facades. It seems that fine-tuning with only three true orthophotographs ($<$100 buildings) is not sufficient to fully adjust the model to the different projection.

Pushing the "open training data" philosophy to the extreme, one could ask whether project-specific training is necessary at all. Maybe the model learned from open data generalizes even to radiometrically different images of comparable GSD? We do not expect this to work, but as a sanity check for a "generic global" semantic segmentation model, we perform a further experiment, where we avoid domain adaption altogether. The FCN is trained on all Google aerial images plus OSM ground truth (Chicago, Paris, Zurich, Berlin, and Potsdam), and then used to predict from the ISPRS images. This achieves significantly worse results (0.645 F_1-score). A small set of images with similar radiometry is needed to adapt the classifier to the sensor properties and lighting conditions of the test set.

Finally, we respond to the questions we raised at the beginning of this section. A general consensus is that *complete substitution* of manually acquired labels achieves acceptable results. Semantic segmentation of overhead images can indeed be learned from OSM maps without any manual labeling effort albeit at the cost of reduced segmentation accuracy. *Augmentation* of manually labeled training data at very large scale reaches the best overall results. Pretraining with large-scale OSM data and publicly available images does improve segmentation of a project-specific data set of independently acquired images and labels (although only by a small margin in this case). An interesting result is that large-scale pretraining on (inaccurate) data increases recall significantly, whereas precision slightly drops (compare *II* and *V* in Table IV). *Partial substitution* of manually labeled training data with large-scale but inaccurate publicly available data works very well and seems to be a good tradeoff between manual labeling effort and segmentation performance. Indeed, pretraining with large-scale OSM labels *can* replace the vast majority of manual labels. A generic model learned from OSM data adapts very well to a specific location and data source with only little dedicated training data.

V. Conclusion

Traditionally, semantic segmentation of aerial and satellite images crucially relies on manually labeled images as training data. Generating such training data for a new project is costly and time consuming, and presents a bottleneck for automatic image analysis. The advent of powerful but data-hungry deep learning methods aggravates that situation. Here, we have explored a possible solution, namely, to exploit existing data, in our case open image and map data from the Internet for supervised learning with deep CNNs. Such training data are available in much larger quantities, but "weaker" in the sense that the images are not representative of the test images' radiometry, and labels automatically generated from external maps are noisier than dedicated ground truth annotations.

We have conducted a number of experiments that validate our hypothesis stated in Section I.

1) The sheer volume of training data can (largely) compensate for lower accuracy.
2) The large variety present in very large training sets spanning multiple different cities does improve the classifier's ability to generalize to new unseen locations (see predictions on Tokyo, Fig. 6).
3) Even if high-quality training data are available, the large volume of additional training data improves classification.
4) Large-scale (but low-accuracy) training data allow substitution of the large majority (85% in our case) of the manually annotated high-quality data.

In summary, we can state that weakly labeled training data, when used at large scale, nevertheless significantly improve segmentation performance, and improve generalization ability of the models. We found that even training only on open data, without any manual labeling, achieves reasonable (albeit far from optimal) results, if the train/test images are from the same source. Large-scale pretraining with OSM labels and publicly available aerial images, followed by domain adaptation to tune to the images at hand, significantly benefits semantic segmentation and should be used as standard practice, as long as suitable images and map data are available.

Online map data, as used in our study, is presently limited to RGB orthophotographs with unknown radiometric calibration and street map data for navigation purposes. But we are convinced that comparable training databases can be generated automatically for many problems of interest on the basis of the image and map archives of mapping agencies and satellite data providers. In fact, we are already observing a trend toward free and open data (e.g., the Landsat and MODIS archives, open geodata initiatives from several national mapping agencies, etc.).

At first glance, it seems that object classes with complex contextual relations, like our *road* class, benefit most from more training data. This intuitively makes sense, because more data are needed to learn complex long-range layout constraints from data, but more research is needed to verify and understand the effects in detail. Moreover, more studies are needed with different class nomenclatures, and more diverse data sets, covering different object scales and image resolutions. A visionary goal would be a large free publicly available "model zoo" of pretrained classifiers for the most important remote sensing applications, from which users world-wide can download suitable models and either apply them directly to their region of interest or use them as initialization for their own training.

References

[1] M. Haklay and P. Weber, "OpenStreetMap: User-generated street maps," *IEEE Pervasive Comput.*, vol. 7, no. 4, pp. 12–18, Oct. 2008.

[2] M. Haklay, "How good is volunteered geographical information? A comparative study of OpenStreetMap and ordnance survey datasets," *Environ. Planning B, Urban Anal. City Sci.*, vol. 37, no. 4, pp. 682–703, 2010.

[3] J.-F. Girres and G. Touya, "Quality assessment of the French OpenStreetMap dataset," *Trans. GIS*, vol. 14, no. 4, pp. 435–459, 2010.

[4] J. D. Wegner, S. Branson, D. Hall, K. Schindler, and P. Perona, "Cataloging public objects using aerial and street-level images—Urban trees," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 6014–6023.

[5] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. CVPR*, 2015, pp. 3431–3440.

[6] F. Rottensteiner, G. Sohn, M. Gerke, J. D. Wegner, U. Breitkopf, and J. Jung, "Results of the ISPRS benchmark on urban object detection and 3D building reconstruction," *ISPRS J. Photogram. Remote Sens.*, vol. 93, pp. 256–271, Jul. 2014.

[7] P. Fua and A. J. Hanson, "Using generic geometric models for intelligent shape extraction," in *Proc. 6th Nat. Conf. Artif. Intell.*, 1987, pp. 706–709.

[8] R. Mohan and R. Nevatia, "Using perceptual organization to extract 3D structures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 11, pp. 1121–1139, Nov. 1989.

[9] M. Herman and T. Kanade, "The 3D mosaic scene understanding system: Incremental reconstruction of 3D scenes from complex image," in *Proc. Image Understand. Workshop*, 1984, pp. 137–148.

[10] U. Weidner, "Digital surface models for building extraction," in *Automatic Extraction of Man-Made Objects From Aerial and Space Images (II)*, A. Gruen, E. P. Baltsavias, and O. Henricsson, Eds. Basel, Switzerland: Springer, 1997, pp. 193–202.

[11] A. Fischer *et al.*, "Extracting buildings from aerial images using hierarchical aggregation in 2D and 3D," *Comput. Vis. Image Understand.*, vol. 72, no. 2, pp. 185–203, 1998.

[12] M. Fischler, J. Tenenbaum, and H. Wolf, "Detection of roads and linear structures in low-resolution aerial imagery using a multisource knowledge integration technique," *Comput. Graph. Image Process.*, vol. 15, pp. 201–223, Sep. 1981.

[13] U. Stilla, "Map-aided structural analysis of aerial images," *ISPRS J. Photogram. Remote Sens.*, vol. 50, no. 4, pp. 3–10, 1995.

[14] C. Steger, C. Glock, W. Eckstein, H. Mayer, and B. Radig, "Model-based road extraction from images," in *Automatic Extraction of Man-Made Objects from Aerial and Space Images, Birkhäuser Verlag Basel*. Basel, Switzerland: Birkhäuser Verlag, 1995, pp. 275–284.

[15] T. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons," *Int. J. Comput. Vis.*, vol. 43, no. 1, pp. 29–44, 2001.

[16] C. Schmid, "Constructing models for content-based image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Dec. 2001, pp. II-39–II-45.

[17] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "TextonBoost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context," *Int. J. Comput. Vis.*, vol. 81, no. 1, pp. 2–23, 2009.

[18] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Dec. 2001, pp. I-511–I-518.

[19] P. Dollär, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," in *Proc. Brit. Mach. Vis. Conf.*, 2009, pp. 91.1–91.11.

[20] B. Fröhlich, E. Bach, I. Walde, S. Hese, C. Schmullius, and J. Denzler, "Land cover classification of satellite images using contextual information," *ISPRS Ann. Photogram., Remote Sens. Spatial Inf. Sci.*, vol. II(3–W1), pp. 1–6, May 2013.

[21] P. Tokarczyk, J. D. Wegner, S. Walk, and K. Schindler, "Features, color spaces, and boosting: New insights on semantic classification of remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 1, pp. 280–295, Jan. 2015.

[22] R. Stoica, X. Descombes, and J. Zerubia, "A Gibbs point process for road extraction from remotely sensed images," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 121–136, 2004.

[23] D. Chai, W. Förstner, and F. Lafarge, "Recovering line-networks in images by junction-point processes," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 1894–1901.

[24] M. Ortner, X. Descombes, and J. Zerubia, "Building outline extraction from digital elevation models using marked point processes," *Int. J. Comput. Vis.*, vol. 72, no. 2, pp. 107–132, 2007.

[25] Y. Verdié and F. Lafarge, "Detecting parametric objects in large scenes by Monte Carlo sampling," *Int. J. Comput. Vis.*, vol. 106, pp. 57–75, Apr. 2014.

[26] S. Kluckner, T. Mauthner, P. M. Roth, and H. Bischof, "Semantic classification in aerial imagery by integrating appearance and height information," in *Proc. ACCV*, 2009, pp. 477–488.

[27] S. Kluckner and H. Bischof, "Image-based building classification and 3D modeling with super-pixels," *Int. Arch. Photogram., Remote Sens. Spatial Inf. Sci.*, vol. 38, no. 3A, pp. 233–238, 2010.

[28] J. D. Wegner, J. Montoya, and K. Schindler, "A higher-order CRF model for road network extraction," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 1698–1705.

[29] J. Montoya, J. D. Wegner, L. Ladicky, and K. Schindler, "Mind the gap: Modeling local and global context in (road) networks," in *Proc. German Conf. Pattern Recognit.*, 2014, pp. 212–223.

[30] J. D. Wegner, J. Montoya, and K. Schindler, "Road networks as collections of minimum cost paths," *ISPRS J. Photogram. Remote Sens.*, vol. 108, pp. 128–137, Oct. 2015.

[31] M. Herold, X. Liu, and K. C. Clarke, "Spatial metrics and image texture for mapping urban land use," *Photogramm. Eng. Remote Sens.*, vol. 69, no. 9, pp. 991–1001, 2003.

[32] M. Dalla Mura, J. A. Benediktsson, B. Waske, and L. Bruzzone, "Morphological attribute profiles for the analysis of very high resolution images," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 10, pp. 3747–3762, Oct. 2010.

[33] W. Schwartz, A. Kembhavi, D. Harwood, and L. Davis, "Human detection using partial least squares analysis," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sep. 2009, pp. 24–31.

[34] S. Hussain and B. Triggs, "Feature sets and dimensionality reduction for visual object detection," in *Proc. Brit. Mach. Vis. Conf.*, 2010, pp. 112.1–112.10.

[35] F. van Coillie, L. Verbeke, and R. D. Wulf, "Feature selection by genetic algorithms in object-based classification of IKONOS imagery for forest mapping in Flanders, Belgium," *Remote Sens. Environ.*, vol. 110, pp. 476–487, Sep. 2007.

[36] Y. Rezaei, M. Mobasheri, M. V. Zoej, and M. Schaepman, "Endmember extraction using a combination of orthogonal projection and genetic algorithm," *IEEE Geosci. Remote Sens. Lett.*, vol. 9, no. 2, pp. 161–165, Feb. 2012.

[37] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biol. Cybern.*, vol. 36, no. 4, pp. 193–202, Apr. 1980.

[38] Y. LeCun *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, 1989.

[39] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[40] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.

[41] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–14.

[42] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, Aug. 2013.

[43] S. Paisitkriangkrai, J. Sherrah, P. Janney, and A. van den Hengel, "Effective semantic pixel labelling with convolutional networks and conditional random fields," in *Proc. Comput. Vis. Pattern Recognit. CVPR Workshops*, Sep. 2015, pp. 36–43.

[44] A. Lagrange *et al.*, "Benchmarking classification of earth-observation data: From learning explicit features to convolutional networks," in *Proc. Int. Geosci. Remote Sens. Symp. (IGARSS)*, 2015, pp. 4173–4176.

[45] D. Marmanis, K. Schindler, J. D. Wegner, and S. Galliani, "Semantic segmentation of aerial images with an ensemble of CNNS," in *Proc. ISPRS Ann.-ISPRS Congr.*, 2016, pp. 473–480.

[46] V. Mnih and G. E. Hinton, "Learning to detect roads in high-resolution aerial images," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 210–223.

[47] V. Mnih and G. E. Hinton, "Learning to label aerial images from noisy data," in *Proc. Int. Conf. Mach. Learn.*, 2012, pp. 1–8.

[48] G. Mättyus, S. Wang, S. Fidler, and R. Urtasun, "Enhancing road maps by parsing aerial images around the world," in *Proc. Int. Comput. Vis. Conf.*, 2015, pp. 1689–1697.

[49] V. Mnih, "Machine learning for aerial image labeling," Ph.D. dissertation, Graduate Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, 2013.

[50] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Convolutional neural networks for large-scale remote sensing image classification," *Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 645–657, Feb. 2017.

[51] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 807–814.

[52] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–359.

[53] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Sep. 2009.

[54] F. Rottensteiner, G. Sohn, M. Gerke, and J. D. Wegner, "ISPRS test project on urban classification and 3D building reconstruction," ISPRS Working Group III/4–3D Scene Anal., Hannover, Germany, Tech. Rep. 12, 2013. [Online]. Available: http://www2.isprs.org/tl_files/isprs/wg34/docs/ComplexScenes_revision_v4.pdf

[55] E. Maggiori, G. Charpiat, Y. Tarabalka, and P. Alliez. "Learning iterative processes with recurrent neural networks to correct satellite image classification maps." Unpublished paper, 2016. [Online]. Available: https://arxiv.org/abs/1608.03440

Pascal Kaiser received the M.S. degree in biology from the University of Zürich, Zürich, Switzerland, in 2011, and the M.S. degree in statistics from the Eidgenoessische Technische Hochschule Zürich, Zürich, in 2016.

Since 2016, he has been a Software Engineer with Supercomputing Systems, Zürich, where he develops and implements algorithms with a focus on machine learning, deep learning, computer vision, and statistical modeling. His research interests include deep neural networks and their application to computer vision.

Jan Dirk Wegner received the Diploma (Dipl.-Ing.) in geodesy and geoinformatics and the Ph.D. (Dr.-Ing.) degree from Leibniz Universität Hannover, Hannover, Germany, in 2007 and 2011, respectively.

He was granted an ETH Post-Doctoral Fellowship in 2011 and held a post-doctoral position from 2012 to 2015 with the Photogrammetry and Remote Sensing Group, ETH Zürich, Zürich, Switzerland, where he has been a Senior Scientist since 2015. Together with his colleagues, he runs the ISPRS benchmark challenge for 2-D pixelwise semantic segmentation, object recognition, and 3-D reconstruction using very high-resolution aerial images and airborne laser scans, and the large-scale point cloud classification benchmark that contains terrestrial laser scans of urban scenes with over one billion labeled points. His research interests include geospatial computer vision, large-scale (deep) machine learning with applications to tree inventories, and semantic 3-D reconstruction and segmentation.

Aurélien Lucchi received the M.S. degree in computer science from the INSA Lyon de France, Villeurbanne, France, and the Ph.D. degree in machine learning and computer vision from EPFL, Lausanne, Switzerland, in 2013.

He has had significant experience in industrial research including internships at Google Research as well as Microsoft Research. Since 2013, he has been a Post-Doctoral Researcher in machine learning with the Data Analytics Laboratory, ETH Zürich, Switzerland. His research interests include optimization and large scale learning as well as machine learning applications in computer vision and text analysis.

Martin Jaggi received the M.S. degree in mathematics and the Ph.D. degree in computer science from Eidgenössische Technische Hochschule Zürich, Zürich, Switzerland, in 2006 and 2011, respectively.

He was a Research Fellow with the Simons Institute, Berkeley, CA, USA, and a Post-Doctoral Researcher with the École Polytechnique, Paris, France. He is a Co-Founder of SpinningBytes, Zurich, Switzerland and the Founder of the Zurich Machine Learning and Data Science Meetup, Zurich. Since 2016, he has been a Tenure Track Assistant Professor with the École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, heading the Machine Learning and Optimization Laboratory. His research interests include machine learning, optimization, and text understanding.

Thomas Hofmann received the Diploma and Ph.D. degrees in computer science from the University of Bonn, Bonn, Germany, in 1993 and 1997, respectively.

He was a Postdoctoral Fellow with the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, from 1997 to 1998, and the University of California at Berkeley, Berkeley, CA, USA, from 1998 to 1999. He was an Assistant and an Associate Professor of Computer Science with Brown University, Providence, RI, USA, from 1999 to 2004. He was a Full Professor with TU Darmstadt, Darmstadt, Germany, from 2004 to 2005 and the Director with Fraunhofer IPSI, Darmstadt. He was the Director of Engineering at Google, Zürich, Switzerland, from 2005 to 2013, where he was involved in Search, e-commerce, and Ads Optimization. He was the Co-Founder of Recommind, London, U.K., in 2000 and 1plusX, Zurich, in 2014, companies that develop machine learning technology for user data and textual content. He is involved in the area of machine learning and artificial intelligence. Since 2014, he has been a Full Professor of computer science with ETH Zürich, Zürich, and the Co-Director with the Center of Learning Systems, Zurich, and Tübingen, Germany. His research interests include deep learning, optimization for machine learning, as well as natural language understanding and machine perception.

Dr. Hofmann is on the editorial board of the *Journal of Machine Learning Research*.

Konrad Schindler (M'05–SM'12) received the Diplomingenieur (M.Tech.) degree in photogrammetry from the Vienna University of Technology, Vienna, Austria, in 1999, and the Ph.D. degree from the Graz University of Technology, Graz, Austria, in 2003.

He was a Photogrammetric Engineer in the private industry and held researcher positions at the Computer Graphics and Vision Department, Graz University of Technology, the Digital Perception Laboratory, Monash University, Melbourne, VIC, Australia, and the Computer Vision Laboratory, ETH Zürich, Zürich, Switzerland. He was an Assistant Professor of Image Understanding with TU Darmstadt, Darmstadt, Germany, in 2009. Since 2010, he has been a Tenured Professor of Photogrammetry and Remote Sensing with ETH Zürich. His research interests include computer vision, photogrammetry, and remote sensing, with a focus on image understanding and information extraction reconstruction.

Dr. Schindler has been serving as an Associate Editor of the *Journal of Photogrammetry and Remote Sensing* of the International Society for Photogrammetry and Remote Sensing (ISPRS) since 2011, and previously served as an Associate Editor of the *Image and Vision Computing Journal* from 2011 to 2016. He was the Technical Commission President of the ISPRS from 2012 to 2016.