ROOF3D: A REAL AND SYNTHETIC DATA COLLECTION FOR INDIVIDUAL BUILDING ROOF PLANE AND BUILDING SECTIONS DETECTION

P. Schuegraf[1]*, M. Fuentes Reyes[1], Y. Xu[1], K.Bittner[1]

[1]Remote Sensing Technology Institute, German Aerospace Center (DLR), Weßling, Germany
{philipp.schuegraf,mario.fuentesreyes,yajin.xu,ksenia.bittner}@dlr.de

KEY WORDS: Building instance segmentation, roof plane instance segmentation, building vectorization, semantic segmentation, 3D reconstruction.

ABSTRACT:

Deep learning is a powerful tool to extract both individual building and roof plane polygons. But deep learning requires a large amount of labeled data. Hence, publicly available level of detail (LoD)-2 datasets are a natural choice to train fully convolutional neural networks (FCNs) models for both building section and roof plane instance segmentation. Since publicly available datasets are often automatically derived, e.g. based on laser scanning, they lack on annotation accuracy. To complement such a dataset, we introduce manually annotated and synthetically generated data. Manually annotated data is domain-specific and has a high annotation quality but is expensive to obtain. Synthetically generated data has high-quality annotations by definition, but lacks domain-specificity. Moreover, we not only detect individual building section instances, but also roof plane instances. We predict separations not only between individual buildings, but also by a class that describes the line which separates roof planes. The predicted building and roof plane instances are polygonized by a simple tree search algorithm. To achieve more regular polygons, we utilize the Douglas-Peucker polygon simplification algorithm. We describe our dataset in detail to allow comparability between successive methods. To facilitate future works in building and roof plane prediction, our Roof3D dataset is accessible at `https://github.com/dlrPHS/GPUB`.

1. INTRODUCTION

Since deep learning was introduced, it is now the state-of-the-art approach in many engineering disciplines. In remote sensing, deep learning is also the most common workflow when it comes to extracting semantic or geometric features from imagery.

1.1 Problem Statement

One of the most prominent features in urban very high-resolution (VHR) imagery are building rooftops. They are comprised of several planes, e.g., a hip roof consists of four roof planes and a gable roof of two such elements as can be seen on Figure 1. Additional major roof elements, that we consider as essential for roof-surface modelling are large dormers. Extracting rooftops and their roof planes aids applications that include 3D-reconstruction, 5G-wave-simulation, solar exposure estimation. In remote sensing imagery, roof-tops exist in a large variety of sizes, shapes and spectral appearances. Help to handle them comes from FCNs, since they are capable of learning hierarchical representations of our target objects in an automatic, data-driven style. But to train such FCN models, it is crucial to have a training dataset of **a)** sufficient variety and sizes, **b)** annotation quality and **c)** domain-relatedness. Public datasets, such as those provided by the federal governments of North-Rhine-Westphalia (NRW), Germany [1], include a large variety of annotated data, but the respective annotations have been detected automatically by a laser-scanning-based method, which leads to severe inaccuracies (see Figure 2). Using synthetic data, which is generated by a procedure as described in (Reyes et al., 2022) implies high-quality annotations of considerable quantity but leads to a domain gap (see Figure 2). Thus, mixing the synthetic data with real data is a potential approach to have both domain specificity and accurate labels. Furthermore, spectral features, such as RGB imagery, can lead to noisy predictions, since a deep learning model might confuse buildings with roads or impervious surface areas in an image. On the other hand, from digital surface model (DSM), it can be difficult to disentangle building roof-tops and high vegetation. Hence, combining both RGB and DSM as inputs to FCNs possibly helps to make roof plane instance segmentation more robust.

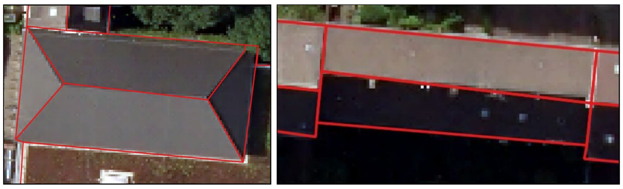

Figure 1. Visualization of a hip (left) and a gable (right) roof, consisting of several roof planes.

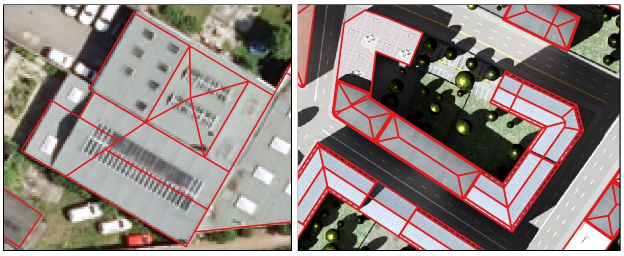

Figure 2. Visualization of erroneous annotations (left) and synthetic image + annotation (right).

* Corresponding author

[1] `https://www.opengeodata.nrw.de/produkte/geobasis`

https://doi.org/10.5194/isprs-annals-X-1-W1-2023-971-2023 | © Author(s) 2023. CC BY 4.0 License.

Name	Area	Representation	Ground Truth	Modalities
RoofWorld (Nauata and Furukawa, 2020)	-	single building	planes	RGB
RoofVec (Hensel et al., 2021)	2.00 km^2	single building	planes	RGB
ISPRS Potsdam	3.42 km^2	tiles	footprints	RGB+IR+DSM
SemCity (inst) (Roscher et al., 2020)	3.02 km^2	tiles	sections	RGB
UBC (Huang et al., 2022b)	66.12 km^2	patch	sections	RGB
Roof3D (ours)	22.40 km^2	patch	sections+planes	RGB+DSM

Table 1. Comparison of multiple building segmentation datasets. From SemCity, we only consider the tiles with instance level annotations. We make a difference between *tiles* and *patches*, where we consider *patches* to have a suitable size to feed them to a neural network during training and *tiles* to be larger than *patches*. *Patches* are usually cropped from *tiles*.

1.2 Related Work

Since roof plane extraction is an important part of LoD-2 reconstruction of buildings, most works that tackle the problem of roof plane extraction are in the realm of 3D reconstruction.

Roof-Top Extraction Several works tackle roof-top extraction. For instance, PolyMapper (Li et al., 2019) directly predicts buildings and road networks in vector format, but its performance on CrowdAI (Mohanty et al., 2020) is limited. ASIP (Li et al., 2020) easily outperforms PolyMapper. It initializes polygons by over-segmenting the image into convex cells and then refines the polygons by minimizing an energy function that describes the fidelity of each polygon to the input image and its complexity. Another method that facilitates roof-top polygonization is that of (Girard et al., 2020), where the authors train a network to predict both the building and building border class as well as a frame field, which represents the two possible tangent directions at each building border pixel. The frame field is used to regularize building borders during training and to aid the polygonization procedure. But the frame field learning method does not leverage the DSM. Several other works tackle the problem of building section instance segmentation. PolyWorld (Zorzi et al., 2022) even overcomes the performance of (Girard et al., 2020) and PolyMapper on the CrowdAI dataset (Mohanty et al., 2020) by training stacked models on several sub-tasks of building polygonization. Yet PolyWorld is not able to predict individual roof sections. That problem is described in (Schuegraf et al., 2022) and an approach that first segments satellite images into background, building and touching borders and uses morphology to refine and translate the results to instance segments. In comparison to Mask-R-CNN (He et al., 2017), the touching border-based method of (Schuegraf et al., 2022) produces seamlessly connected neighboring building sections. Hence, that method is suitable to predict individual building roof-tops, but it does not infer the roof planes.

Roof-Plane Extraction There are several studies on roof plane extraction. One work that does not incorporate learning, but relies on hand-crafted features to reconstruct building roof-tops in 3D is that of (Nex and Remondino, 2012). Their method relies on the availability of the near-infrared channel, which is not always the case. Also, this method fails when buildings are very complicated. (Arefi and Reinartz, 2013) reconstruct buildings in LoD-2 in a learning-free manner as well, leveraging both the DSM and orthorectified image. Although that approach produces regular reconstructed buildings that improve over existing 3D models, the learning-free approach relies on hand-crafted features and is therefore not robust to large variations in the input data. Deep learning offers more robust features to aid remote sensing tasks such as roof plane extraction. Sat2LoD (Gui et al., 2022) uses both learning and non-learning based roof-features to generate an LoD-2 model from input imagery and DSM. The approach requires prior information of building classification and relies on open street map (OSM). On the other hand, there exist end-to-end trainable methods. Another work that generates realistic roof-geometry outlines is RoofGAN, which is a generative adversarial network (GAN), of (Qian et al., 2020). But RoofGAN is not conditioned on images, making it impractical to use for roof plane extraction from remote sensing imagery. (Alidoost et al., 2019) use only a single overhead image to generate LoD-2 building models. Their workflow consists of first training two separate networks for height estimation and roof-line, i.e. eaves, ridge and hip extraction and then using a model based approach to obtain 3D models of buildings. Although this approach achieves regularized city models, it relies on the predicted heights, which are not reliably predicted from only an image. (Zhao et al., 2022) propose a two-stage approach, where in the first stage, a multi-task module extracts geometric primitives of roof planes. In the second stage, a graph neural network reconstructs the roof plane polygon. Their approach is suitable to predict roof plane

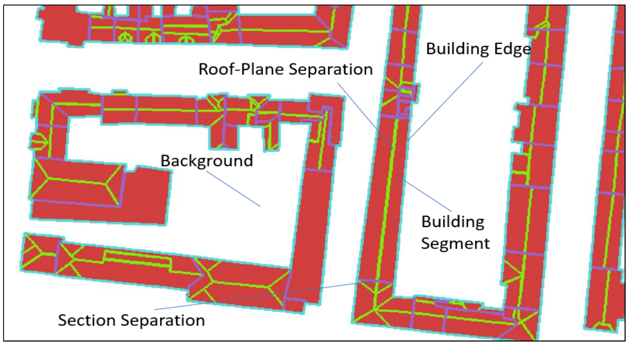

Figure 3. Visualization of the 5 classes in our segmentation problem. White is background, red is the building segment (segment), green separates roof planes (inner), purple separates building sections (section) and blue represents building edge (outer).

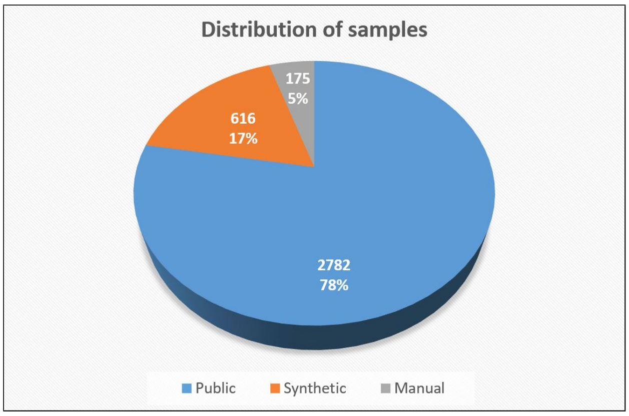

Figure 4. Distribution of samples in manual, public and synthetic subsets of our dataset.

polygons, but does lack the inclusion of the DSM.

Related Benchmark Datasets There exist multiple benchmarks for building roof reconstruction from remote sensing imagery. SemCity (Roscher et al., 2020) is desgined to facilitate research on building instance segmentation with focus on semantic segmentation and roof-part instance segmentation from multi-spectral and RGB imagery. But SemCity lacks the DSM as an input, which is also true for the urban building reconstruction (UBC) dataset (Huang et al., 2022b). In comparison to SemCity, UBC includes fine-grained roof-type information to allow LoD-2 building reconstruction. Furthermore, none of these datasets includes a large amount of buildings, which is most probably caused by relying on only the costly and limited manual annotation, which we overcome by integrating both coarse, publicly available and synthetically generated annotations with manually annotated samples. One dataseset that contains a large number of annotated RGB and DSM patches but lacks roof plane information is the Potsdam 2D semantic labelling contest dataset [2]. The City3D (Huang et al., 2022a) dataset contains 20,000 building instances in LoD-2 along with airborne lidar point-clouds. In the two works of (Nauata and Furukawa, 2020) and (Hensel et al., 2021) datasets for roof geometry extraction along with ortho-imagery are provided, but DSM data is not included.

Real and Synthetic Data in Remote Sensing The combination of real and synthetic data has been gaining attention in remote sensing recently. One such work is that of (Liu et al., 2022), where the authors use a CycleGAN to translate synthetic images to the distribution of real images. The pseudo-real images are then used for training a neural network on aircraft detection. In (Patyk et al., 2020), simulated and real data are leveraged together as well. In their work, the authors use synthetic poultry distributions and aerial imagery jointly to model poultry populations. Furthermore, (Kong et al., 2020) use synthetic data to augment real data for building segmentation. There, the authors follow the same paradigm of mixing simulated and realistic data in a single dataset as we do.

The contributions of this paper are the following:

- We annotated a single dataset that includes ground truth for both individual buildings and roof planes.
- We provide the community with a benchmark dataset which contains not only, as usual, the RGB images, but also DSMs.
- In this paper, we present a method on training a model on a combination of publicly annotated, synthetically generated and manually annotated images, in order to use complementary information from each of it to complete the task with best possible performance.

2. METHOD

Since we also introduce a baseline method for the provided dataset, we describe an approach to LoD-2 instance segmentation.

Architecture Our approach is based on semantic segmentation of an RGB image and a DSM. We use the Fuse scheme as in (Henry et al., 2021), which sums feature maps from different levels of resolution of two separate encoders at the skip-connections. Both encoders are based on ResNet-34 (He et al., 2016) and the decoder has a generic architecture with skip-connections on four levels of resolution. Our architecture is generic for bi-modal semantic segmentation tasks. The last layer of the decoder has five channels, one for the background channel, one for the LoD-2 separation line, one for the building section separation line, one for the building edge and one for the building segment class (find a visualization of the 5 classes on Figure 3).

Loss Function Next to the architecture, the loss function is of high importance for the training of deep neural networks in remote sensing. We use the weighted sum of the weighted cross entropy loss

$$\mathcal{L}_{CE}(x, y, p, w) = -\sum_{i=0}^{4} w_i \sum_{n=0}^{N} y_{in} \cdot \log(p(x_i)_n) \qquad (1)$$

and dice loss

$$\mathcal{L}_{DICE}(x, y, p) = 1 - 2 \cdot \frac{\sum_{i=0}^{4} \sum_{n=0}^{N} y_{in} \cdot p(x_i)_n}{\sum_{i=0}^{4} \sum_{n=0}^{N} y_{in} + p(x_i)_n}, \qquad (2)$$

where x is the input comprised of RGB and/or DSM, y is the ground truth raster, p are the predicted class probabilities, N is the number of pixels in a training batch and $w = [1, 4, 3, 2, 1]$ is the manually selected weight vector, to train our FCN architecture in a way that takes the imbalance of classes into account.

Post-Processing Since we want to achieve instance segmentation instead of semantic segmentation outputs, we need to post-process the semantic predictions. We use the third, fourth and fifth channel of the semantic predictions to generate building sections as in (Schuegraf et al., 2022), by using both the third and fourth channel as the watershed-lines. We dilate the watershed lines by a radius $R = 3$, to overcome the issue of small gaps in separation lines. We apply the same scheme to obtain LoD-2 sections, but combine the second, third and fourth channel as the watershed line before dilation.

Polygonization Since most remote sensing applications require that building layers are in vector format, we apply two simple steps to convert building section and plane instances to polygons, that preserve the separation between individual sections and individual planes. We use the sobel edge detector to obtain edges surrounding each individual section/plane and a contour search to convert the edge pixels to polygons. To obtain more regular polygons, we apply the Douglas-Peucker algorithm (Douglas and Peucker, 1973). But the usage of Douglas-Peucker polygon simplification on the whole instance leads to overlapping polygons or gaps at the junction of directly neighboring instances. Hence, we apply Douglas-Peucker to the separation and the result of the surroundings individually, which leads to both simplified polygons and consistent separations between them.

[2] https://www.isprs.org/education/benchmarks/UrbanSemLab/2d-sem-label-potsdam.aspx

	Train			Test		
	Buildings	Planes	Area	Buildings	Planes	Area
Public	49604	70859	17.3 km^2	-	-	-
Synthetic	2944	32649	3.6 km^2	-	-	-
Manual	1074	3609	1.5 km^2	932	1949	0.3 km^2
Total	53622	107117	22.4 km^2	932	1949	0.3 km^2

Table 2. Quantitative details of our dataset. By planes we refer to the number of roof-planes.

3. EXPERIMENTS

3.1 Data

The data we use for experimental evaluation is comprised of three parts all consisting of RGB, DSM and ground truth data. RGB images are first enhanced by setting all values below the 2nd percentile to zeros and above the 98th percentile to 255. The remaining values are stretched to the 0 to 255 range and the images are stored in 8-bit representation. Before model forward passes, the image patch is rescaled to the range $[-1, 1]$. The DSM is stored as a 32-bit float raster and before passing the DSM patch to the model, all values below the 2nd percentile are set to -1 and all values above the 95-percentile are set to 1, whereas the remaining values are stretched to the $[-1, 1]$ range. All RGB images and DSMs are resampled to $0.3\,\mathrm{m}$ ground sampling distance (GSD) using bi-cubic interpolation. In the following three paragraphs, the respective original GSD is provided. For all three sub-datasets, polygons in shapefile format are obtained by various procedures. The shapefiles are then processed by rasterizing them to roof plane instance maps. Then, the instance map is used as basis to obtain a map of the five classes: background, roof plane separation line, individual building separation line, building edge and building segment. The separation lines are similar as in (Schuegraf et al., 2022), where only only a separation line for individual buildings is segmented. In all experiments, the data areas are split in patches of size 512×512 px, with an overlap of 256 px in the horizontal and vertical directions. The dataset consists of public, synthetic and manually annotated data. The public and synthetic are used exclusively for training. The manually annotated data consists of multiple non-overlapping tiles. One of these tiles is used for testing and the others for training.

Coarsely-Annotated Public Data Since a large amount of data is available publicly on the internet, it is straightforward to use it to train deep neural networks. We use parts of a dataset from NRW, Germany. The included imagery has red-green-blue-near infrared channel representation, with original GSD of $0.1\,\mathrm{m}$, but we omit the near infrared channel in our experiments. Additionally, we use the DSM, which has an original GSD of $0.5\,\mathrm{m}$, in our dataset. We select 8 areas that are located in the city of Cologne for training. We extract the roof plane polygons from the provided CityGML.

Synthetic Data To both compensate the lack of labelled data and provide a highly accurate ground truth, we also created synthetic data resembling aerial imagery. We used a pipeline similar to the one in (Reyes et al., 2022) and a 3D model based on Paris from the ESRI platform[3]. We edited the model to include a larger density of buildings and a LoD-2 representation. Samples for flat, hip, mansard, gable and gambrel rooftops are included.

The model was exported as a Wavefront (.obj) file in two ways: as a single object containing the full scene and exporting each building as a separate object. The former is rendered with the BlenderProc pipeline (Denninger et al., 2020) to generate an image similar to what can be acquired with an aerial camera. We set the camera parameters to simulate a GSD of 30cm with an otrhographic view. Additionally, we rendered the distance map from the same point of view, which is later post-processed to a DSM. Due to its synthetic nature, it is important to mention that the simulated region does not correspond to a real place.

The second exported case, where each building is considered as a single object, is also loaded into Blender. The 3D model is further edited to remove all surfaces except for those belonging to the rooftops. This is done by filtering the object faces with respect to its texture. After the process, only the rooftop geometry remains and is exported as a Shapefile (.shp) with BlenderGIS[4]. As we mentioned above, the geometry of the buildings was limited to LoD-2 to avoid an excessive number of planes. This shapefile has a polygon for each plane belonging to the rooftops.

An additional shapefile is directly generated from the scene in CityEngine. For this case, we selected only the parcels containing buildings and exported the shapefile directly from the software. This shows a polygon for each building in the scene, unlike the previous one, where each polygon represents a plane. In total the dataset includes an aerial photo, a DSM, a plane based shapefile and a building based shapefile.

Manually-Annotated Data The gold-standard for annotation quality is still set by the human annotator (check a crop from our segmentations on Figure 5). We selected two RGB-DSM-pairs from the public data provided by NRW in the city of Cologne, that do not overlap with the areas we chose in the public data subset. We use one of the areas for evaluation of all our models and the other one in terms of the manually-annotated training data subset. As an additional training area, we selected two RGB-DSM pairs from the city of Berlin, Germany, provided by the senate administration of Berlin [5] with $0.2\,\mathrm{m}$ and $1.0\,\mathrm{m}$ original GSD respectively. We annotated roof planes and building sections according to visual inspection, leveraging QGIS (QGIS Development Team, 2009) as the annotation tool. The annotation is based on the input image and does not consider the DSM. Since in very complex situations, it is not always clear where one building ends and the next one starts in the top-view, some ambiguities remain. Our manual annotations challenge methods to annotate what is visible in the top-view.

Overall Quantitative details of the entire dataset and its parts are given on Tables 1 and 2 and Figure 4. We call our dataset Roof3D and note the following advantages of our dataset:

- Our dataset has both building instances and roof plane instances matching the building instances.

[3] https://www.arcgis.com/home/item.html?id=30d64fcf53a84be8be1d46905534f5bf

[4] https://github.com/domlysz/BlenderGIS

[5] https://www.businesslocationcenter.de/downloadportal/

https://doi.org/10.5194/isprs-annals-X-1-W1-2023-971-2023 | © Author(s) 2023. CC BY 4.0 License.

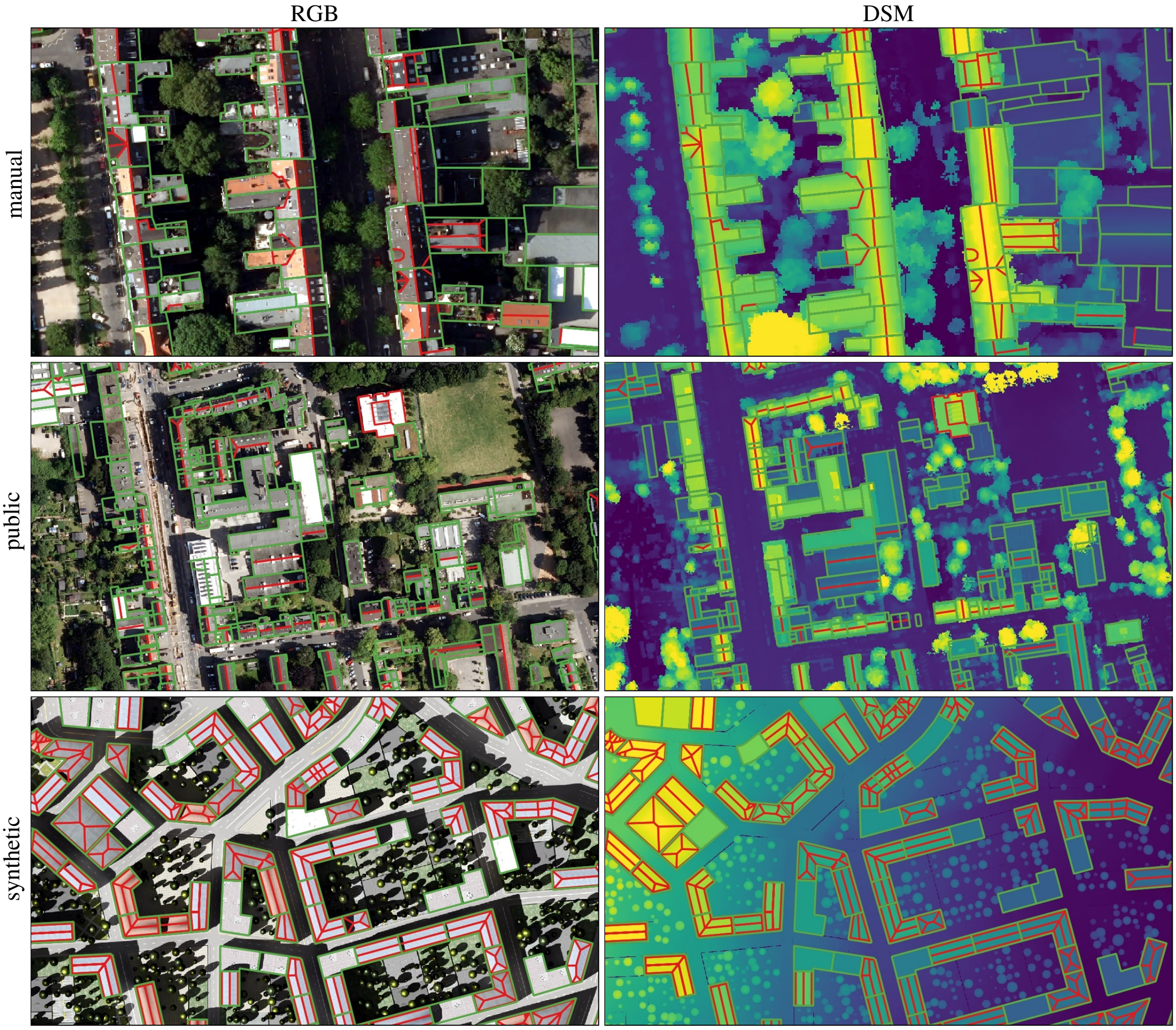

Figure 5. Visualization of examples of our annotations. Green lines represent the overlap of building section outlines and roof plane borders and red lines the remaining roof plane outlines.

- Furthermore, it includes DSM and has considerable size, which is necessary to train deep neural networks. Only UBC covers a larger area on Table 1. For the computation of the areas, we did not consider overlaps for UBC but for our dataset we did. See Figure 5 for excerpts of all parts of the dataset.

3.2 Training

We use the AdamW optimizer (Loshchilov and Hutter, 2019) to train our models with weight decay 0.0001 and an initial learning rate of 0.001. To improve convergence, we multiply the learning rate by $\gamma = 0.99$ after each epoch. Since pre-trained weights are not suitable for height features as they are included in the DSM, we initialize the learnable parameters of the network by the Xavier random initialization scheme. The model is trained for 150 epochs and we save the model with the lowest validation loss as the final model. We randomly rotate all patches in the range of $0°$-$360°$ and apply random color jitter to the RGB patch, since we want to achieve high generalization capability of the trained model.

3.3 Inference

We perform inference on the whole test area, which is a rectangular tile, by cropping patches of size 512×512 px with an overlap of 256 px in horizontal and vertical direction each. The trained model is used to infer the semantic predictions of the background, building and separation line classes. The prediction scores are averaged at the overlapping regions, which leads to 4-time inference in all areas besides the first and last 256 pixels of each row and column, where there are only 2 inferences. The four 256×256 px windows at the corners of the test area are targeted by only a single inference. The prediction of the full test area is converted to classes using the argmax classifier. In a learning-free post-processing step, the 5-class prediction is converted into two separate parts. The first part is for building section instance segmentation including the separation line between sections, building block surrounding border and building segment. The building segments gaps are filled using the predicted roof plane separations and regarding them as segment. The second part is for roof plane instance segmentation including the separation line between roof planes, the separation line between sections, the building block surround border and the segment. In both cases, the dilated line classes are regarded as watershed lines and the segment is regarded as the seed. We apply dilation with a disk of radius $R = 3$ as the structuring element. To obtain instance maps, we follow the procedure as described in (Schuegraf et al., 2022). Afterwards, simple tree search gives us polygons of building

Architecture	Modality	Public	Manual	Synthetic	IoU_{inner}	IoU_{section}	IoU_{outer}	IoU_{segment}	IoU_{mean}	OA
U-Net-ResNet34	RGB	X			0.338	0.317	0.335	0.693	0.421	0.949
U-Net-ResNet34	RGB	X	X		0.344	0.318	0.335	0.697	0.424	0.949
U-Net-ResNet34	RGB	X		X	0.342	0.309	0.330	**0.700**	0.420	**0.950**
U-Net-ResNet34	RGB	X	X	X	0.348	0.311	0.338	0.698	0.424	**0.950**
U-Net-ResNet34	DSM	X	X		0.353	0.217	0.313	0.647	0.382	0.938
U-Net-ResNet34	DSM	X	X	X	0.354	0.238	0.303	0.630	0.381	0.937
Fuse-U-Net-ResNet34	RGB+DSM	X	X		**0.366**	0.312	**0.360**	0.690	**0.432**	0.948
Fuse-U-Net-ResNet34	RGB+DSM	X	X	X	0.363	**0.322**	0.356	0.685	**0.432**	0.948

Table 3. Results of multiple architectures on the semantic segmentation metrics.

Architecture	Modality	Public	Manual	Synthetic	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AR	AR_{50}	AR_{75}	AR_S	AR_M	AR_L	$F1_{INST}$
U-Net-ResNet34	RGB	X			0.167	**0.367**	0.136	0.073	0.364	0.411	0.329	0.562	0.340	0.176	0.522	0.533	0.222
U-Net-ResNet34	RGB	X	X		**0.169**	0.361	**0.149**	**0.080**	**0.365**	0.347	0.325	0.548	0.344	0.181	0.525	**0.550**	0.222
U-Net-ResNet34	RGB	X		X	0.149	0.325	0.128	0.063	0.320	0.267	0.329	0.562	0.335	0.181	0.493	0.400	0.205
U-Net-ResNet34	RGB	X	X	X	0.168	**0.367**	0.148	0.074	0.357	**0.374**	**0.339**	0.580	0.359	0.186	**0.530**	0.517	**0.225**
U-Net-ResNet34	DSM	X	X		0.054	0.134	0.040	0.029	0.109	0.192	0.181	0.348	0.176	0.096	0.270	0.213	0.083
U-Net-ResNet34	DSM	X	X	X	0.080	0.185	0.068	0.041	0.156	0.232	0.227	0.419	0.233	0.128	0.316	0.265	0.118
Fuse-U-Net-ResNet34	RGB+DSM	X	X		0.136	0.292	0.119	0.068	0.275	0.258	0.310	0.527	0.328	0.178	0.453	0.293	0.189
Fuse-U-Net-ResNet34	RGB+DSM	X	X	X	0.153	0.330	0.137	0.068	0.309	0.319	0.333	**0.564**	**0.358**	**0.187**	0.473	0.369	0.210

Table 4. Results of multiple architectures for building section instance segmentation.

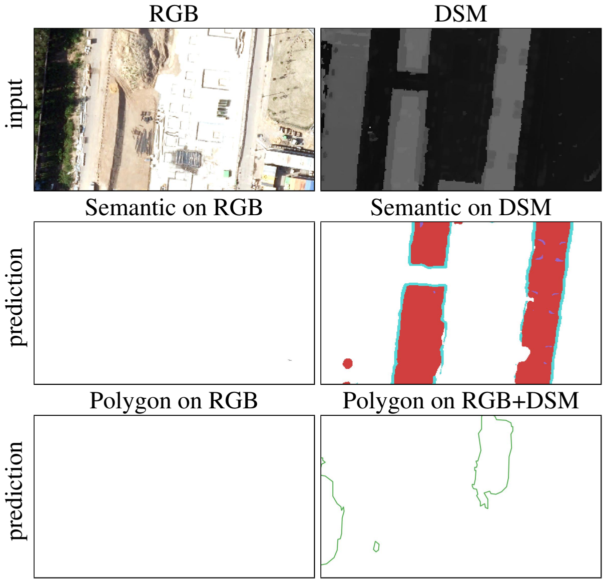

Figure 6. Visualization of the differences between the RGB and DSM and corresponding semantic (white: background, red: outer boundary, purple: segment, blue: plane separation) predictions of both single-modality networks. The green line represents a predicted section polygon. The absence of a green line indicates that no sections are detected by the model.

sections and roof planes, which are simplified using Douglas-Peucker (Douglas and Peucker, 1973) with $\epsilon = 0.5$.

3.4 Metrics

To quantitatively evaluate the predictions of the trained models, we use two kinds of metrics. Since the baseline method produces semantic segmentation maps as in-between results, we use IoU-score

$$IoU_C = \frac{TP_C}{TP_C + FP_C + FN_C}, \tag{3}$$

where $C \in \{$*background, inner, section, outer, segment*$\}$, *inner* denotes the separation between roof planes that does not belong to the separation between sections, *section* is the separation between buildings, the class *outer* is the border of the whole building block of free-standing individual buildings and *segment* are all remaining pixels belonging to buildings, TP_C is the amount of pixels belonging to class C and predicted as class C, FP_C is the amount pixels belonging to any other class but C and is predicted as C, FN_C is the amount of pixels belonging to class C but are predicted as any other class but C. To get a better overview on which experiment gives the best comprehensive result, we average the IoU of the foreground classes *inner*, *section*, *outer* and *segment* to obtain IoU_{mean}

$$IoU_{\text{mean}} = \frac{IoU_{inner} + IoU_{section} + IoU_{outer} + IoU_{segment}}{4}. \tag{4}$$

Furthermore, the overall accuracy

$$OA = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

gives us insight about the ratio of correctly classified pixels. Since most of the pixels belong to the *background* class, we expect the OA to be much higher than the IoU-scores for the line classes.

To evaluate the instance segmentation results on a polygon basis, we use the standard COCO metrics AP and AR, evaluated in $[0.5, 0.95]$ with steps of $[0.05]$ and for small, medium and large size objects, augmented by the harmonic mean of AP and AR

$$F1_{INST} = 2\frac{AP \cdot AR}{AP + AR} \tag{6}$$

.

The harmonic mean is heavily skewed towards the weaker of AP and AR. Hence, if we use it to judge the success of our experiments, we can be certain that our model performs well on both AP and AR. Since we evaluate the test region not patch wise but as a larger tile assembled from multiple overlapping patches, the standard number of detection per sample is not adequate. Instead, we take every detection of our method into account by adapting the corresponding code in the *pycocotools* package.

3.5 Comparison

To demonstrate how our dataset and method add value to the task of LoD-2 plane and building section instance segmentation, we carried out multiple experiments. For all experiments, we use the part of the dataset that comes from a public source and has coarse ground truth, since the other parts are not large enough to avoid overfitting. Furthermore, the public data is always available. We also evaluate the combinations **public +**

Architecture	Modality	Public	Manual	Synthetic	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AR	AR_{50}	AR_{75}	AR_S	AR_M	AR_L	$F1_{INST}$
U-Net-ResNet34	RGB	X			0.113	0.270	0.092	0.097	0.218	0.102	0.277	0.503	0.290	0.254	0.385	0.340	0.161
U-Net-ResNet34	RGB	X	X		**0.126**	**0.294**	**0.106**	**0.110**	0.234	0.101	**0.295**	**0.526**	**0.311**	**0.273**	**0.392**	0.300	**0.177**
U-Net-ResNet34	RGB	X		X	0.106	0.256	0.083	0.093	0.195	0.102	0.262	0.486	0.263	0.240	0.347	0.240	0.151
U-Net-ResNet34	RGB	X	X	X	0.120	0.278	0.099	0.102	0.239	0.161	0.276	0.500	0.286	0.250	0.402	**0.350**	0.167
U-Net-ResNet34	DSM	X	X		0.056	0.161	0.033	0.046	0.120	0.100	0.192	0.396	0.177	0.178	0.235	0.138	0.087
U-Net-ResNet34	DSM	X	X	X	0.068	0.181	0.045	0.061	0.110	0.018	0.211	0.420	0.206	0.207	0.232	0.033	0.103
Fuse-U-Net-ResNet34	RGB+DSM	X	X		0.112	0.266	0.090	0.092	0.228	0.103	0.270	0.497	0.281	0.247	0.357	0.140	0.158
Fuse-U-Net-ResNet34	RGB+DSM	X	X	X	0.125	0.292	0.104	0.100	**0.260**	**0.185**	0.286	0.524	0.301	0.259	0.385	0.283	0.174

Table 5. Results of multiple architectures on roof plane instance segmentation.

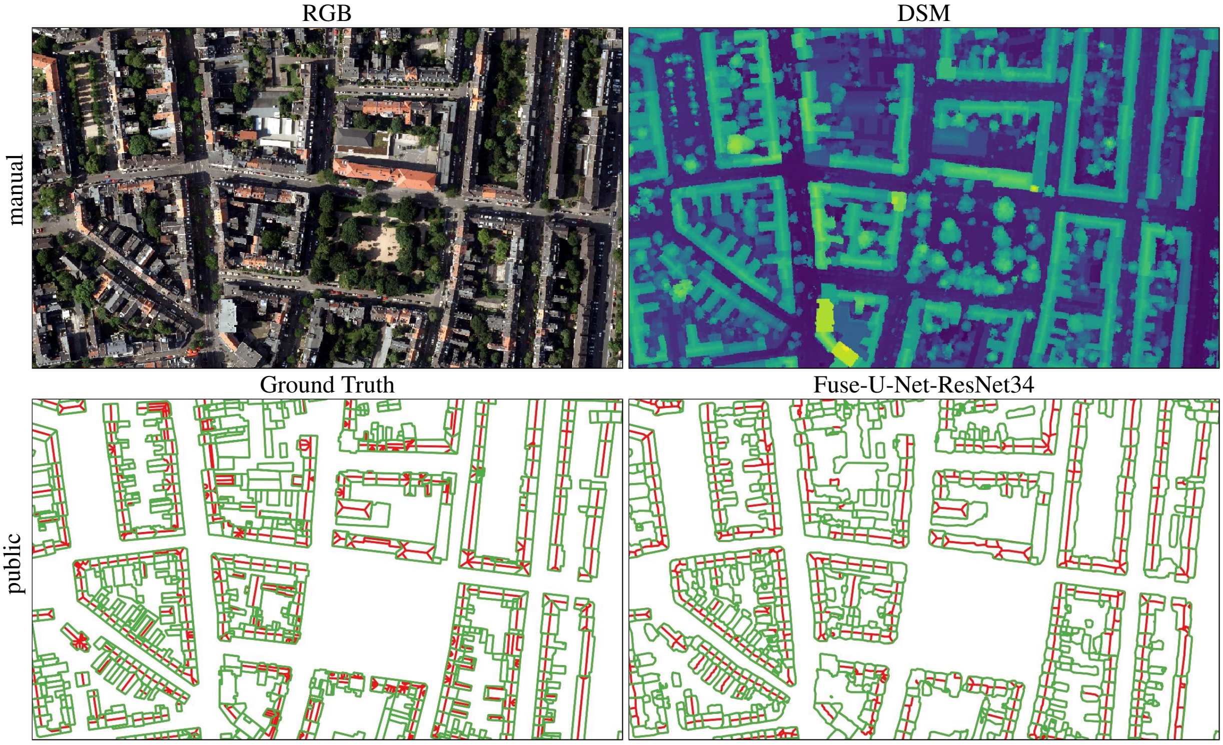

Figure 7. Visualization of a crop of our results. Green lines represent the overlap of building section outlines and roof plane borders and red lines the remaining roof plane outlines.

manual, **public + synthetic** and **public + manual + synthetic** to justify the use of synthetic data and verify the effectiveness of even a small amount of additional, hand-annotated data. In the single modality experiments, we use the U-Net-ResNet34 (Girard et al., 2020) architecture and for the experiments with bi-modal inputs we use the Fuse-U-ResNet34, since the Fuse-strategy was previously verified to be effective for building section instance segmentation by (Schuegraf et al., 2022). To show that including DSM is a valuable enrichment of the input feature space of our network, we train and evaluate models on the more promising combinations of **public + manual** and **public + manual + synthetic**. These combinations emerge to be the higher performing ones, as will be shown in the results section.

4. RESULTS

To obtain an objective perspective on the outcome of the experiments, we look at the quantitative results.

RGB On Table 3, it can be seen, that among the models that operate on the RGB modality, the combinations public + manual and public + manual + synthetic have the highest IoU_{mean} values of 0.424. Furthermore, public + manual has the largest IoU_{section} of 0.318 among the experiments on RGB and public + manual + synthetic has the highest IoU_{inner} of 0.348. Adding a small, manually annotated dataset to the public dataset improves the performance in all metrics, but adding our synthetic data improves especially the inner class. Regarding the building section instance segmentation results, the combination public + manual + synthetic achieves the highest $F1_{INST}$ score of 0.225 on Table 4. Hence, synthetic data is useful for the task at hand. On the other hand, the combination of public + synthetic performs worse with $F1_{INST}$ of 0.205. Accordingly, combining a large set of coarsely annotated data with data that has a large domain shift does not lead to an improvement, but if a small set of manually annotated data is added to the dataset as well, the performance on section segmentation improves over the other combinations. We observe similar behavior on the plane segmentation task using the RGB modality. Here, the trained model achieves an $F1_{INST}$ of 0.151 on public + synthetic, 0.167 on public + manual + synthetic and 0.177 on public + manual (compare Table 5). Therefore, at the LoD-2 plane task, the combination of public + manual is performing better than public + manual + synthetic. It is likely due to the predominant mansard roof-type in the synthetic data, which does not occur in the test area. Owing to their outstanding performance, we continue the evaluation on both tasks and the semantic segmentation metrics on only the combinations public + manual and public + manual + synthetic.

DSM Quantitatively, the models trained on the height features are performing worse than all other models. In many cases, neighboring buildings vary in spectral apperance, but not in their height profile. Furthermore, due to patterns created by the upsampling algorithm (publicly available DSMs come with

$\frac{1}{5}$ of the resolution of the aerial image) height differences can often hardly be detected because they are not much larger than the interpolation patterns. Another reason for the weaker performance of the model trained on only DSM when compared to the result of the RGB-model is that we annotated both the manual training and test areas using the RGB images (compare Figure 6).

RGB+DSM The combination of modalities RGB+DSM performs significantly worse on $F1_{INST}$ than only RGB for the public + manual subsets (0.189 vs. 0.222 on sections, 0.158 vs. 0.177 on planes) and comparable for RGB on public + manual + synthetic (0.210 vs. 0.225 on sections, 0.174 vs. 0.167 on planes). Furthermore, the two models trained on RGB+DSM outperform the other models in IoU_{mean}, both scoring 0.432, with the second best model (RGB, public+manual+synthetic) achieving 0.424. Since the performance of the Fuse-U-Net-ResNet34 trained on public+manual+synthetic performs quantitatively comparable to the U-Net-ResNet34 trained on public+manual, we verify that the difference between the two modalities leads to a performance drop of the Fuse-U-Net-ResNet34 (compare Figure 6). If we take into account the semantic segmentation metrics as well, we conclude that the Fuse-U-Net-ResNet34 is a valuable improvement over the U-Net-ResNet34 operating on only the RGB. As mentioned previously, large dormers are also regarded as roof planes. Looking at Figure 7, we observe that the Fuse-U-Net-ResNet34 does not identify the dormers that are present in the image, DSM and ground truth. Hence, our model focuses on the other parts of the roof structure.

5. CONCLUSION

We introduce a benchmark dataset for both roof plane and building section instance segmentation tasks as the basis for LoD-2 reconstruction of buildings on aerial imagery and a baseline method that is suitable for both tasks. Our presented benchmark dataset is partitioned of publicly available, manually annotated and synthetically generated ground truth. This composition shows to gain a slight advantage versus other combinations of the three subsets, although only adding synthetic to the public data does not improve segmentation performance. But we only investigated the simple mixing scheme, where the three parts are used for training in the same training dataset. Hence, we pose the question: How can the three parts be combined in a way that takes into account the size of the publicly available, the quality and domain specificity of the the manually annotated and the accuracy of the synthetically generated subsets of the dataset even better? Furthermore, our dataset includes not only an aerial image, but also the corresponding aerial DSM to introduce complementary information to the RGB image. The reader is encouraged to think of alternatives for the fusion at the skip-connection to leverage both inputs. In the future, we hope research on individual roof plane and building section segmentation is stimulated by offering Roof3D as a benchmark.

REFERENCES

Alidoost, F., Arefi, H., Tombari, F., 2019. 2D Image-To-3D Model: Knowledge-Based 3D Building Reconstruction (3DBR) Using Single Aerial Images and Convolutional Neural Networks (CNNs). *Remote Sensing*, 11, 2219 ff.

Arefi, H., Reinartz, P., 2013. Building Reconstruction Using DSM and Orthorectified Images. *Remote Sensing*, 5(4), 1681 ff.

Denninger, M., Sundermeyer, M., Winkelbauer, D., Olefir, D., Hodan, T., Zidan, Y., Elbadrawy, M., Knauer, M., Katam, H., Lodhi, A., 2020. Blenderproc: Reducing the Reality Gap with Photorealistic Rendering. *International Conference on Robotics: Sciene and Systems*.

Douglas, D. H., Peucker, T. K., 1973. Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10, 112 ff.

Girard, N., Smirnov, D., Solomon, J., Tarabalka, Y., 2020. Regularized Building Segmentation by Frame Field Learning. *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*, 1805 ff.

Gui, S., Qin, R., Tang, Y., 2022. SAT2LOD2: A Software for Automated lod-2 Building Reconstruction from Satellite-Derived Orthophoto and Digital Surface Model. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 379 ff.

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask R-CNN. *IEEE International Conference on Computer Vision*, 2980 ff.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep Residual Learning for Image Recognition. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 770–778.

Henry, C., Hellekes, J., Merkle, N., Azimi, S. M., Kurz, F., 2021. Citywide Estimation of Parking Space using Aerial Imagery and OSM Data Fusion with Deep Learning and Fine-Grained Annotation. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 479 ff.

Hensel, S., Goebbels, S., Kada, M., 2021. Building Roof Vectorization with PPGNet. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 85 ff.

Huang, J., Stoter, J., Peters, R., Nan, L., 2022a. City3D: Large-Scale Building Reconstruction from Airborne LiDAR Point Clouds. *Remote Sensing*, 14(9), 2254 ff.

Huang, X., Ren, L., Liu, C., Wang, Y., Yu, H., Schmitt, M., Hänsch, R., Sun, X., Huang, H., Mayer, H., 2022b. Urban Building Classification (UBC) – A Dataset for Individual Building Detection and Classification from Satellite Imagery. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 1412 ff.

Kong, F., Huang, B., Bradbury, K., Malof, J., 2020. The Synthinel-1 dataset: a collection of high resolution synthetic overhead imagery for building segmentation. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 1803 ff.

Li, M., Lafarge, F., Marlet, R., 2020. Approximating shapes in images with low-complexity polygons. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8630 ff.

Li, Z., Wegner, J. D., Lucchi, A., 2019. Topological Map Extraction From Overhead Images. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1715 ff.

Liu, W., Luo, B., Liu, J., 2022. Synthetic Data Augmentation Using Multiscale Attention CycleGAN for Aircraft Detection in Remote Sensing Images. *IEEE Geoscience and Remote Sensing Letters*, 19, 1 ff.

Loshchilov, I., Hutter, F., 2019. Decoupled Weight Decay Regularization. *International Conference on Learning Representations*.

Mohanty, S. P., Czakon, J., Kaczmarek, K. A., Pyskir, A., Tarasiewicz, P., Kunwar, S., Rohrbach, J., Luo, D., Prasad, M., Fleer, S. et al., 2020. Deep Learning for Understanding Satellite Imagery: An Experimental Survey. *Frontiers in Artificial Intelligence*, 3.

Nauata, N., Furukawa, Y., 2020. Vectorizing World Buildings: Planar Graph Reconstruction by Primitive Detection and Relationship Inference. *European Conference on Computer Vision*, 711 ff.

Nex, F., Remondino, F., 2012. Automatic Roof Outlines Reconstruction from Photogrammetric DSM. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 257 ff.

Patyk, K. A., McCool-Eye, M. J., South, D. D., Burdett, C. L., Maroney, S. A., Fox, A., Kuiper, G., Magzamen, S., 2020. Modelling the domestic poultry population in the United States: A novel approach leveraging remote sensing and synthetic data methods. 15.

QGIS Development Team, 2009. QGIS Geographic Information System. *Open Source Geospatial Foundation*.

Qian, Y., Zhang, H., Furukawa, Y., 2020. Roof-GAN: Learning to Generate Roof Geometry and Relations for Residential Houses. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2795 ff.

Reyes, M. F., D'Angelo, P., Fraundorfer, F., 2022. SyntCities: A Large Synthetic Remote Sensing Dataset for Disparity Estimation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15, 10087 ff.

Roscher, R., Volpi, M., Mallet, C., Drees, L., Wegner, J. D., 2020. Semcity Toulouse: A Benchmark for Building Instance Segmentation in Satellite Images. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 109 ff.

Schuegraf, P., Schnell, J., Henry, C., Bittner, K., 2022. Building Section Instance Segmentation with Combined Classical and Deep Learning Methods. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 407 ff.

Zhao, W., Persello, C., Stein, A., 2022. Extracting planar roof structures from very high resolution images using graph neural networks. *ISPRS Journal of Photogrammetry and Remote Sensing*, 187, 34 ff.

Zorzi, S., Bazrafkan, S., Habenschuss, S., Fraundorfer, F., 2022. PolyWorld: Polygonal Building Extraction With Graph Neural Networks in Satellite Images. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1848 ff.