BUILDING SECTION INSTANCE SEGMENTATION WITH COMBINED CLASSICAL AND DEEP LEARNING METHODS

Philipp Schuegraf[1] *, Julian Schnell[2], Corentin Henry[1], Ksenia Bittner[1]

[1] Remote Sensing Technology Institute, German Aerospace Center (DLR), Wessling, Germany
[2] Environmental Engineering, Technical University of Darmstadt (TU Darmstadt), Darmstadt, Germany
{philipp.schuegraf ,corentin.henry,ksenia.bittner}@dlr.de, julianchristopher.schnell@stud.tu-darmstadt.de

WG II/III

KEY WORDS: Building instance segmentation, high-resolution satellite imagery, WorldView-4, deep learning, watershed labeling, semantic segmentation.

ABSTRACT:

In big cities, the complexity of urban infrastructure is very high. In city centers, one construction can consist of several building sections of different heights or roof geometries. Most of the existing approaches detect those buildings as a single construction in the form of binary building segmentation maps or as one instance of object-oriented segmentation. However, reconstructing complex buildings consisting of several parts requires a higher level of detail. In this work, we present a methodology for individual building section instance segmentation on satellite imagery. We show that fully convolutional networks (FCNs) can tackle the issue much better than the state-of-the-art Mask-RCNN. A ground truth raster image with pixel value 1 for building sections and 2 for their touching borders was generated to train models on predicting both classes as a semantic output. The semantic outputs were then post-processed with the help of morphology and watershed labeling to generate segmentation on the instance level. The combination of a deep learning-based approach and a classical image processing algorithm allowed us to fulfill the segmentation task on the instance level and reach high-quality results with an mAP of up to 42 %.

1. INTRODUCTION

1.1 Problem Statement

Remote sensing and computer vision scientists find a big interest in building segmentation. With the world's population rising drastically and urban areas becoming denser, such applications become helpful in fields like population counting, urban planning, reconstruction, disaster monitoring and city modeling. However, building constructions are not always simple and can be described with several polygons, which in turns can represent different roof types (see Figure 1). For modern applications, it is not enough anymore to extract a row of different building roofs as one building footprint or instance. Therefore, a precise identification of different sections within one building object is a topic of interest.

In principle, the use of aerial or satellite intensity images for automatic segmentation of buildings should be sufficient. However 2D information does not reflect the real form of building rooftops which is crucial for dividing one building object on several parts in case of complex structure. To overcome this problem, the image data has to be paired with other data sources. This additional information source can be a digital surface model (DSM) - an image which represents elevation data. Height information together with intensity information provide an opportunity to find touching borders between building sections since the transition area is then very clear.

In the literature, the task of building sections separation is usually viewed as a part of 3D building reconstruction problem. Traditional methods are based on ridge lines detection from satellite normalized digital surface models (nDSMs) and intensity images (Arefi and Reinartz, 2013) or detection of the step edges from canny points on the nDSMs of light detection and ranging (LiDAR) data (Zheng et al., 2017). In this work, we propose a machine learning approach that can automatically segment building parts and touching borders between those parts. Furthermore, applying a watershed algorithm (Beucher and Meyer, 2018) we aim to extract individual instances of each building section. Thus, the decomposition of a row of different rooftop structures belonged to one building object can be viewed now as an independent task and can be utilized for various geoinformation system (GIS) applications.

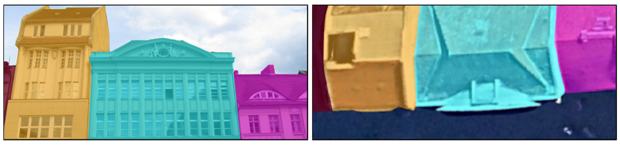

(a) Streetview (b) Topview

Figure 1. An example of typical houses in Berlin, Germany, taken from Google Earth. Each color represents a different building section, where all sections belong to the same building structure.

1.2 Related Work

Classical methods to extract building footprint on aerial or satellite imagery are based on the identification of edges and other primitive shapes typical for buildings (Huertas and Nevatia, 1988). For almost a decade, learning-based approaches like machine learning and deep learning have overtaken the state-of-the-art methodologies for remote sensing problems. Convolutional neural networks (CNNs) used for image segmentation have been developed and improved constantly since then. For

* Corresponding author

example, in 2015 researchers developed an architecture known as ResNet (He et al., 2016), a deep residual network with several options differing in depth and the number of convolution layers. This architecture's deepest version, ResNet152, reached an error rate of only 3.57 % on the ImageNet classification challenge (Russakovsky et al., 2015), better than human error on the same task (5 %) (Khan et al., 2020). In the same year, a team of researchers released models called FCN (Long et al., 2015). This pixel-wise prediction method was the first adaptation of older networks to take input of any size and output classification maps of the same size. With the skip connections in its architecture, this approach allowed for more detailed spatial information recovery and has become the basis of most state-of-the-art models. This development has also been followed up in the field of building segmentation by several studies (Bittner et al., 2018b, Schuegraf and Bittner, 2019, Khan et al., 2020). In the following years, the focus in semantic segmentation lay heavily on the backbone network architecture. Backbone networks are the subset of convolutional layers which have been originally designed for image classification task but can be used for different applications since fully connected layers are removed to keep spatial dependencies. For example, the DenseNet architecture was released in 2017 by (Huang et al., 2017) and was utilized for semantic segmentation by (Jégou et al., 2016) and in 2021, (Henry et al., 2021) proposed a novel architecture for multi-class road network segmentation based on DenseNet which includes the fusion of aerial imagery with open street map (OSM). They fuse the image and the OSM at the bottleneck of the U-shaped network and use a DenseNet121 as the encoder and a decoder, where the blocks of the encoder are mirrored. Their SkipFuse-U-DenseNet121 architecture leverages the semantic information from the OSM better than other fusion techniques and significantly outperforms the models without OSM. Recent investigations have shown that also using a DSM can have a significant impact on building detection and reconstruction tasks when using deep learning techniques (Bittner et al., 2018a, Bittner et al., 2019). In 2018, a team of scientists proposed a methodology and a model, called TernausNetV2 (Iglovikov et al., 2018), to segment buildings on the instance level by predicting not only binary building footprints, but also separation lines between buildings. With eleven multi-spectral input channels, it reached an intersection over union (IoU) score of up to 74 % on the SpaceNet dataset (Etten et al., 2019). This method is different from the most common approach in computer vision, which is a two-stage approach like in (He et al., 2017)'s work, where first, a bounding box is extracted and second, the image is segmented inside the rectangular bounding box. This has been done successfully for remote sensing instance segmentation by (Potlapally et al., 2019). An improved version of the Mask-RCNN, called Hybrid Task Cascade, has been applied to buildings by (Zhao et al., 2020). Hybrid Task Cascade iterative recycles mask and bounding box precisions in an interleaved execution procedure, together with a semantic branch, to refine the mask predictions in comparison to Mask-RCNN. (Zhao et al., 2020) additionally apply Douglas-Peucker as a post-processing step. The Douglas-Peucker algorithm simplifies the building polygons that can be obtained by edge detection, to include only the most important corners. The Hybrid Task Cascade together with the Dogulas-Peucker algorithm produces precise and geometrically sound results on a public building segmentation dataset. However, it was not shown by the authors that Hybrid Task Cascade combined with Douglas-Peucker can also segment touching building sections.

In this paper, we show that the SkipFuse-U-DenseNet121 architecture **(a)** works out of the box with a DSM and **(b)** is capable of segmenting buildings on high-resolution satellite images on a sub-instance level. We show the benefits of using a semantic segmentation network compared to a two-stage instance segmentation network like Mask-RCNN (He et al., 2017). Since the outputs of deep learning-based methods are not perfect, we use the watershed algorithm and morphological operations to post-process them, which reduces the incompleteness of the building border predictions and helps closing the gap between the building instances generated by the neural networks. We adjust pre- and post-processing steps to make the methodology less sensitive to the incompleteness of the raw predictions. The most similar work to ours is the one of (Iglovikov et al., 2018), but we focus on building sections instead of whole buildings. Our method also has parallels with the work of (Luiz Ferreira de Carvalho et al., 2021), where object borders together with a classical method are also used to predict object instances. However, differently from this work, we predict only the touching borders, not all borders, since this lays the focus on the most important borders for building instance extraction. We also apply our method to building sections instead of vehicles, which have much more homogeneous shapes than buildings.

2. METHODOLOGY

2.1 Models

In this work, the performance of several state-of-the-art neural network architectures is investigated on predicting not only the building footprint as one object, but automatically decomposing it into several parts in case of complex structures consisting of several roof types.

One of the most famous networks for instance segmentation is the Mask-RCNN architecture. It was introduced by (He et al., 2017) and its derivatives are now state-of-the-art on common, natural image instance segmentation benchmarks. It has a two-stage design, where first, bounding boxes at the object level are generated and second, a mask is generated for each bounding box. The loss function of Mask-RCNN consists of three parts:

$$\mathcal{L}_{mrcnn} = \mathcal{L}_{bbox} + \mathcal{L}_{cls} + \mathcal{L}_{mask} \tag{1}$$

where $\mathcal{L}_{bbox}$ is for bounding box regression realized by a smooth $\mathcal{L}_1$ loss

$$\mathcal{L}_{\text{smoothL1}}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 0.5 \\ |x| - 0.5 & \text{otherwise,} \end{cases} \tag{2}$$

with $x = p - y$. The variable p is the predicted center coordinate, height and width of a bounding box and y are the ground truth bounding box parameters. $\mathcal{L}_{cls}$ is for the classification task and realized via the log loss

$$\mathcal{L}_{\text{log}}(p) = -\big(y * log(p) + (1-y) * log(1-p)\big). \tag{3}$$

$\mathcal{L}_{mask}$ is for binary mask segmentation and is implemented by a per-pixel log loss as in Equation (3). By having a binary loss for the mask generation and a multinomial loss for classes separately, the mask generation and classification are decoupled, which improves the performance (He et al., 2017). However, this approach does not take into account a-priori knowledge of building sections, which do not overlap and consist of polygons. Hence, other models have to be taken into account, which perform semantic segmentation with one class being the separation class.

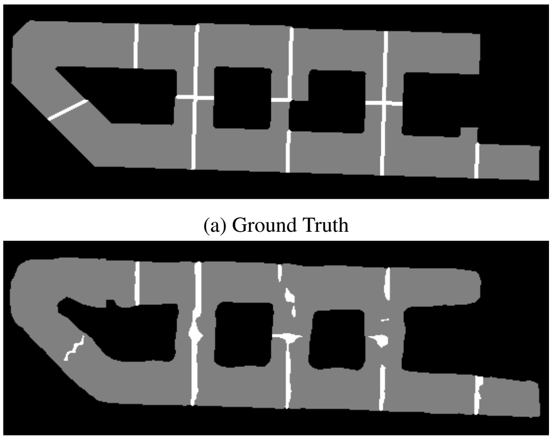
The FCN proposed by (Long et al., 2015) transforms an arbitrary classification network into a backbone for semantic segmentation. For example, due to the popularity of the ResNet50 model and its outstanding performance on classification tasks, a fully convolutional version of this model was chosen as addition and comparison. The basic construction of this network was inspired by the philosophy of the VGG net (Simonyan and Zisserman, 2015). With the help of the residual blocks design, spatial details can bypass a layer that would otherwise make the network lose accuracy and the model can therefore retain the shapes of features significantly better. The modularity of Mask-RCNN allows us to implement it with a ResNet50 backbone network. Furthermore, we exploit an FCN-ResNet50 as a baseline to show the effectiveness of the touching borders based building section instance segmentation. We train the FCN-ResNet50 and each other model in this section, other than the Mask-RCNN, using likelihood maximization via the logloss in Equation (3).

Since the FCN-ResNet50 is not designed to incorporate depth information, several models with different architectures are investigated for building segmentation. In (Schuegraf and Bittner, 2019), two U-shaped networks, one for spectral images and one for depth information are fused at the late stage, called LateFusion-U-VGG16. The U-shape originates from the stage-wise reconstruction of spatial details in the decoder, where each stage of the decoder has the same spatial resolution as a corresponding stage in the encoder. The feature maps of the encoder are also passed to the decoder through a so-called skip connection. No upsampling has to be done at the end of the U-shaped network to transfer the predictions to the image resolution. Since ResNet has outperformed VGG net on many tasks, we replace the VGG16 with a ResNet50, called LateFusion-U-ResNet50 (Bittner et al., 2019). The authors have shown that the fusion of the spectral and depth branches at the bottleneck with a common decoder is beneficial for DSM refinement using a panchromatic satellite image as a second input. Hence, the strength of this Coupled-U-ResNet50 lays in combining an image with depth information. However, Coupled-U-ResNet50 is prone to overfitting, since it uses concatenations as the input to the skip connections, which introduces a lot of additional parameters to the model. Therefore we also test SkipFuse-U-DenseNet121 that performs additions on the feature maps which flow into the skip connections, leading to significantly fewer parameters.

2.2 Pre- and Post-Processing

Outputs of deep learning based methods are never perfect, which is why post-processing them with traditional image processing methods can be a great addition to improve the results noticeably. In this case, the most significant error the outputs of all experiments can have in common is the imperfectly predicted touching borders class (see example in Figure 2) due to their tiny structure and their under-representation in comparison to building and background classes.

Since building sections do not overlap in the top-down view, their segmentation depends on the separation of sections. First, we propose to dilate the predicted touching borders with a disk-shaped kernel of 9 pixels radius to close possible holes in the predicted touching borders. The radius of 9 pixels equates to 2.7m, which is reasonably thick to close even large gaps in touching borders and thin enough to not merge too many close touching borders. We identified 9 pixels as a good radius by visual inspection. These improved borders are now subtracted from the building mask. The resulting blobs of pixels are then detected by a watershed algorithm (Roerdink and Meijster, 2003). The watershed algorithm is a traditional segmentation method, which does not include learning. It uses the smoothness of intensities in an intensity image or gradient magnitudes to separate objects by regarding the image as a topographic surface, which is flooded by water. The separated basins are then identified as objects. Usually, intensity images contain a huge number of basins that are due to noise and variations in illumination or shadows. However, in this paper, the watershed can be regarded as a very simple segmentation layer, which only takes the already segmented and separated blobs and gives them instance numbers.


(a) Ground Truth

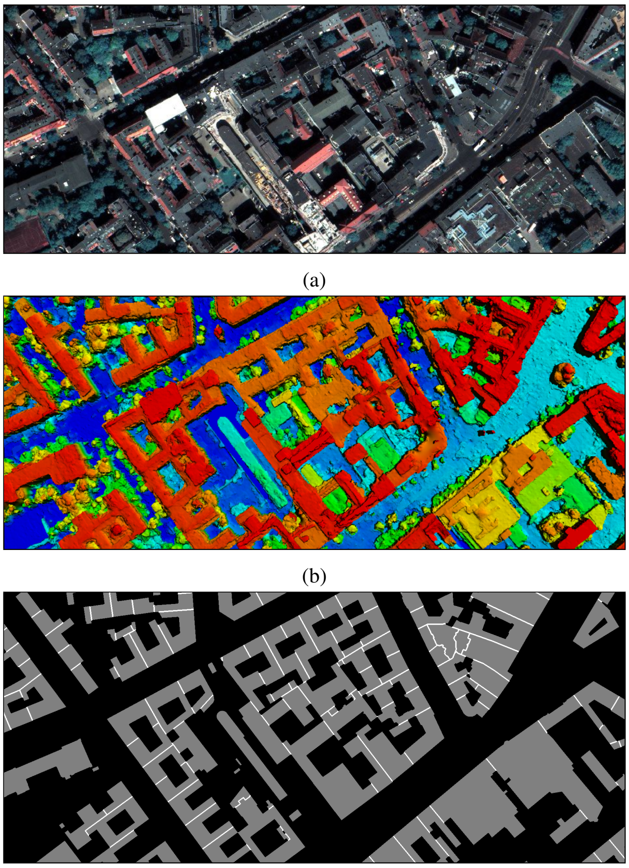
(b) FCN-ResNet50

Figure 2. One building example of (a) ground truth and (b) a raw prediction output from FCN-ResNet50 with RGB input

3. EXPERIMENTS

3.1 Dataset

The dataset consists of a pan-sharpened RGB image and an optional DSM as the input, as well as a raster image building mask as a ground truth (GT). The DSM was obtained by stereo matching of multiple different views on the same scene. Example tiles can be seen in Figure 3. The RGB image and the multi-view images which are input to the stereo matching originate from high-resolution WorldView-4 imagery and shows the city of Berlin, Germany. Its ground sampling distances (GSD) is 0.3 m. The image contains three channels (red, green, blue) and has a size of 33206×32229 pixels. The DSM was resampled to the same size and GSD as the RGB image. A building instance is defined by the coordinates of an addressed house, provided by the German Federal Agency for Cartography and Geodesy. However, adressed houses are not always visibly separated in the top-down view, which leads to possible ambiguities in the evaluation of a particular result. For visual inspection, we only take the geometric and spectral differences of neighboring houses into consideration and for quantitative evaluation, we accept inaccuracies, since we visually inspected large parts of the test area and found that it is a very rare case that different addresses



(a)

(b)

(c)

Figure 3. A sample area of the dataset, consisting of (a) RGB, (b) DSM and (c) the three-class ground truth.

are not visually distinguishable. The ground truth raster is converted to the same resolution and size as the input figures. The raster shows values 0 for background and a building raster mask with rising pixel values for each individual building, thus every pixel is linked to an individual building instance. Than, the touching borders between buildings are initially calculated by searching for neighbors for each instance. In a refinement step, morphology is applied to make touching borders pixels mutually exclusive with building pixels. Afterwards these borders are merged with a binary image of the building mask, so the result is a raster image with three values: 0 for background, 1 for building mask and 2 for touching borders.

The image is split into three regions of interest (RoI) whilst ensuring the diversity of building roofs of each RoI by visual inspection: training (∼85 %), validation (∼12.5 %) and testing (∼2.5 %). This results in 3442 patches for training, 538 for validation and 102 patches for testing each with a size of 512x512 pixels. The final step of preparing the dataset is to normalize all input data before training. Input values are rescaled to the range [-1, 1], where the minimum value of each patch is mapped to -1 and the maximum to 1.

3.2 Training Settings

Each model is trained for a duration of 100 epochs, where one epoch consists of iterations on the whole training set. After the training process to avoid overfitting, the parameters after the epoch at which the model has the lowest validation loss are loaded into the model and the model with these parameters are then used for evaluation. The learning rate was set to 0.0002 with a scheduled decrease of 10 percent of the updated learning rate (exponential decay) after every epoch.

All ResNet50 and DenseNet121 backbones are initialized with ImageNet (Deng et al., 2009) pre-trained weights. The networks' parameters are then updated with the stochastic gradient descent by the use of an Adam optimizing method (Kingma and Ba, 2015) with the first order momentum set to 0.9 and the second order momentum set to 0.999. Since the touching borders only take up 0.46 % of all pixels in the ground truth raster, the relating class was weighted up with a factor of 2 in the loss function. We found that a higher weight makes the training process less stable and a lower weight leads to a very low recall of the touching borders class. The loss for all models but the Mask-RCNN-ResNet50 is calculated with the weighted multi-class cross-entropy loss function

$$\mathcal{L}_{\text{CE}}(\hat{p}, w) = -\frac{1}{\sum_{cl=1}^{3} w_{cl}} \sum_{cl=1}^{3} \hat{y}_{cl} w_{cl} log(\hat{p}_{cl}), \qquad (4)$$

where $\hat{p}_{cl}$ is the softmax activated output of the network of class cl, w_{cl} is the respective class-weight and $\hat{y}_{cl}$ is the binary ground truth of class cl.

3.3 Experimental Design

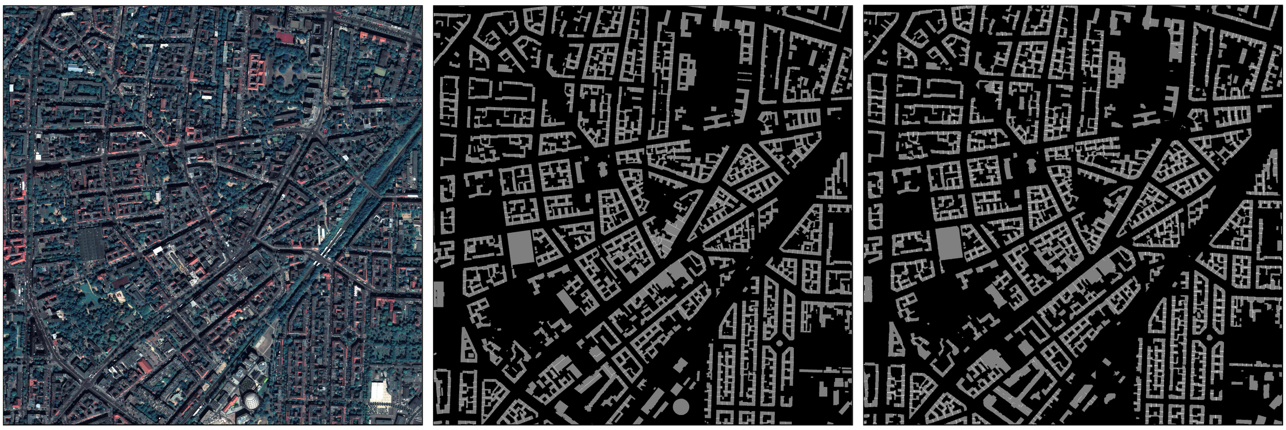
FCN-ResNet50: A ResNet50 is leveraged as the backbone combined with a simple decoder, consisting of a convolutional layer and upsampling of the feature maps of the backbone. The input is an RGB image. To show the value of auxiliary depth information, a DSM is concatenated as the fourth channel of the input for another experiment. However, concatenating the DSM is naive and therefore, two different fusion strategies are followed for comparison. Both the models with and without the DSM are trained with batch size 4 for 10 epochs without the DSM and 25 with the DSM.

Late-U-ResNet50 & Coupled-U-ResNet50: The first one is the Late fusion, where two U-shaped branches produce 30 feature maps each, both of the same size as their input, which are then concatenated and passed through three convolutional layers to produce the output. The second one utilizes the Coupled fusion, where the feature maps from two backbones are concatenated before the bottleneck and a common decoder network generates the output. Although the fusion before the bottleneck is very promising, concatenation at this semantic depth introduces millions of additional parameters. These two models are trained with batch size 2, due to their large memory consumption. The Late-U-ResNet50 is trained for 25 epochs and the Coupled-U-ResNet50 for 80 epochs.

SkipFuse-U-DenseNet121: Hence, an architecture, which fuses at the same depth, but uses summation of the feature maps instead of concatenation is evaluated. However, this network also uses a DenseNet121 instead of a ResNet50. The SkipFuse-U-DenseNet121 is trained with batch size two for 30 epochs.

U-DenseNet121 & U-ResNet50: In the following experiments, a U-DenseNet121 and a U-ResNet50 are trained on solely RGB, to see which of the encoders works better for the task at hand. The U-ResNet50 is trained with batch size 4, whereas the U-DenseNet121 is trained with batch size 2. After 10 epochs, the U-DenseNet121 peaked in validation loss. The U-ResNet50 reaches its lowest validation loss after 45 epochs.

https://doi.org/10.5194/isprs-annals-V-2-2022-407-2022



(a) RGB (b) Ground Truth (c) SkipFuse-U-DenseNet121

Figure 4. Whole test set, with (a) RGB, (b) GT and (c) output from SkipFuse-U-DenseNet121. Best viewed zoomed-in on a computer screen.

Mask-RCNN-ResNet50: Finally, a Mask-RCNN with a ResNet50 backbone is trained on RGB, since it represents a milestone in deep learning based instance segmentation. It is trained using stochastic gradient descent (SGD) with an initial learning rate of 0.01 and a scheduled decay of 5 percent of the initial learning rate. The Mask-RCNN-ResNet50 is trained with batch size 4 for 18 epochs and its loss consists of multiple losses for different tasks like bounding box detection and mask prediction.

4. RESULTS AND DISCUSSION

Outputs from the inference phase on the test data set come in patches with a size of 1024×1024 pixels and stitched together afterwards to enable the comparison between the semantic output and the ground truth mask (see Figure 4). The stitching is done by using an overlap of 40 pixels and removing a border of 20 pixels of each patch in the overlapping regions. This gives a good first overview of the performance.

The post-processing of the outputs and application of the watershed algorithm to the masks creates an image of buildings segmented on the instance level. For example in Figure 4b, each colour represents individual parts of a building, while black represents the background (0).

To evaluate the performance of experiments, several metrics are calculated for each class separately. For each class, a binary mask is computed for the ground truth and class predictions and the corresponding number of True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN) are calculated.

The precision

$$Pre_c = \frac{TP}{TP + FP} \tag{5}$$

is a measure for how good the segmentation method does not predict the negatives as positives for a particular class, the recall

$$Rec_c = \frac{TP}{TP + FN} \tag{6}$$

gives insight into how complete the pixels of a certain class are segmented, the $F1$

$$F1_c = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{7}$$

is a metric which combines precision and recall, such that the $F1$ is drawn stronger towards the lower of the two and the IoU

$$IoU_c = \frac{TP}{TP + FP + FN} \tag{8}$$

is the ratio of overlapping pixels of the prediction and ground truth over their union.

Next to the background, there are the classes building mask (BM) and touching borders (TB). All the previous metrics are listed for the class $c \in \{BM, TB\}$. Since in this work we focus on instance segmentation, the common instance metrics mean average precision (mAP) and mean average recall (mAR) are used to evaluate the models on an instance level. Both these metrics rely on computing the IoU of predicted instance masks and ground truth masks, where the mAP tends to punish predicted masks, which cannot reach a certain threshold and the mAR has a reciprocal relation with the number of ground truth instances that are not matched by any of the predicted instances in terms of a threshold. To get a balanced metric, the $F1_{IS}$ is computed similar as in Equation 8, but with the mAP and mAR instead of precision and recall. Selected metrics presented in Equations (5) to (8) are summarized in Table 1 for each model.

First, the Mask-RCNN-ResNet50 achieves the $F1_{IS}$ of 0.34, which is among the lowest of all models. The detect-then-segment approach of the Mask-RCNN was first introduced to well-separated, large objects in natural images. However, the

MODEL	INPUT	Pre_{BM}	Rec_{BM}	$F1_{BM}$	IoU_{BM}	$Prec_{TB}$	Rec_{TB}	$F1_{TB}$	IoU_{TB}	$F1_{IS}$	mAP	mAR
Mask-RCNN-ResNet50	RGB									0.33	0.27	0.43
U-DenseNet121	RGB	0.91	0.73	0.81	0.68	0.12	0.85	0.21	0.12	0.36	0.33	0.41
FCN-ResNet50	RGB	0.90	**0.74**	**0.81**	**0.69**	**0.12**	0.84	**0.22**	**0.12**	0.33	0.29	0.39
FCN-ResNet50	RGB+DSM	0.92	0.72	0.81	0.68	0.12	0.88	0.21	0.12	0.44	0.4	0.49
Late-U-ResNet50	RGB+DSM	0.91	0.71	0.8	0.67	0.12	0.84	0.2	0.11	0.35	0.3	0.41
Coupled-U-ResNet50	RGB+DSM	0.92	0.71	0.8	0.67	0.12	0.87	0.2	0.11	0.36	0.31	0.43
SkipFuse-U-DenseNet121	RGB+DSM	**0.93**	0.71	0.8	0.67	0.11	**0.92**	0.19	0.11	**0.47**	**0.42**	**0.54**
U-ResNet50	RGB	0.91	0.71	0.8	0.66	0.11	0.87	0.2	0.11	0.38	0.33	0.44

Table 1. Performance comparison of different models and input data for the building mask and touching borders classes.

MODEL	INPUT	$F1_{IS}$	mAP	mAR
Mask-RCNN-ResNet50	RGB	0.33	**0.27**	0.43
U-DenseNet121	RGB	0.09	0.06	0.15
FCN-ResNet50	RGB	0.1	0.07	0.19
FCN-ResNet50	RGB+DSM	0.17	0.13	0.25
Late-U-ResNet50	RGB+DSM	0.08	0.05	0.17
Coupled-U-ResNet50	RGB+DSM	0.12	0.08	0.25
SkipFuse-U-DenseNet121	RGB+DSM	**0.34**	0.26	**0.5**
U-ResNet50	RGB	0.11	0.07	0.23

Table 2. Quantitative results without post-processing.

touching borders method uses a-priori knowledge of building sections. For example in Figure 5, the building sections have little separations and the results of the Mask-RCNN-ResNet50 are separated in many places, whereas in the sub-instances generated by the SkipFuse-U-DenseNet121 architecture, there is no space between the sections. Hence, the Mask-RCNN is not as tailored to the task of building section instance segmentation as the touching borders method.

Next, the FCN-ResNet50 reaches a $F1_{IS}$ of 0.33 if trained on RGB and 0.4398 if trained on RGB concatenated with DSM. This shows that additional depth information, even when introduced in a naive fashion, is valuable for building segmentation.

More sophisticated fusion strategies, as the Late and Coupled fusion, do not necessarily represent an upgrade to the concatenation. The Late-U-ResNet50 and Coupled-U-ResNet50 reach an $F1_{IS}$ of 0.35 and 0.36, respectively, which is lower than the 0.44 of the FCN-ResNet50 trained on RGB+DSM. The Late fusion implies that there are two completely separate networks and the fusion is done at image resolution with concatenation and convolutional layers. This results in 279 million trainable parameters, which is much more than the 33 million trainable parameters of the FCN-ResNet50, both with and without the DSM. In deep learning, it is known that the number of parameters must be carefully adapted to a dataset's size, since large models overfit on small datasets. The Coupled-U-ResNet50 has 174 million trainable parameters, which is why we compare the Late-U-ResNet50 and the Coupled-U-ResNet50 with a model which has fewer parameters.

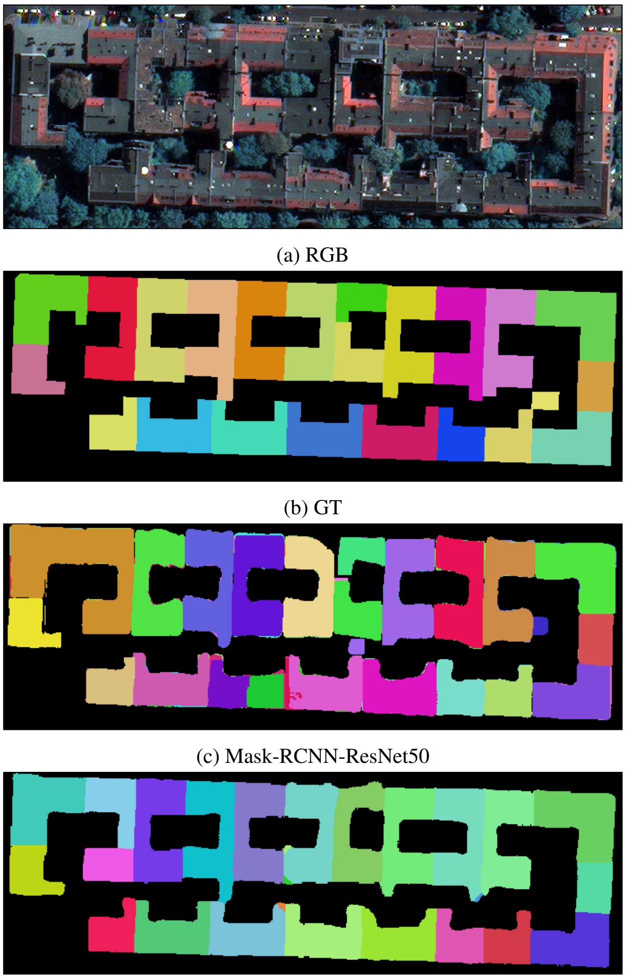
For example, the SkipFuse fusion technique has much fewer parameters, since it uses summation instead of concatenation at the skip connections. The SkipFuse-U-DenseNet121 has only 25 million parameters, reaches an $F1_{IS}$ of 0.47 and is the best performing model in our analysis in terms of instance segmentation. It does not outperform all other models on most of the metrics in Table 1, but it has the highest Rec_{TB}, which means its touching borders are the most complete and the watershed transform needs well separated regions, which is why the SkipFuse-U-DenseNet121 wins over all other models in the three instance metrics. The high Rec_{TB} can be visually understood by looking at Figure 7, where the touching borders are very complete. Furthermore, we can see in Figure 6 that an FCN-ResNet50 trained on RGB+DSM produces touching borders with a snake pattern, which the SkipFuse-U-DenseNet121 does not. Since the FCN architecture does not incorporate high-resolution geometric information from the early feature maps via skip connections, it does not perform well on small structures like touching borders. The numbers in Table 2 indicate that without the dilation, the SkipFuse-U-DenseNet121 is still the best model in the study and the FCN-ResNet50 trained on RGB and DSM is much further away from the SkipFuse-U-DenseNet121 as with post-processing. Even though the SkipFuse-U-DenseNet121 is only slightly better than the Mask-RCNN-ResNet50 if no post-processing is done, it shows that the SkipFuse-U-DenseNet121 relies less on post-processing than the other touching borders networks.

In the last experiment, the U-DenseNet121 has an $F1_{IS}$ of 0.36 and the U-ResNet50 achieves 0.38 on the same metric. The SkipFuse-U-DenseNet121 is different to the Coupled-U-ResNet50 in two major ways. It has **1**), a slightly different fusion strategy, which reduces the number of parameters and **2**) a DenseNet121 backbone instead of a ResNet50 backbone. Since the U-ResNet50 outperforms the U-DenseNet121, it is shown that the change of encoder cannot be the reason why the SkipFuse-U-DenseNet121 is so far ahead of competition and we corroborate that the fusion of RGB and DSM before the backbone, combined with summation instead of concatenation at the skip connections is suitable for building segmentation.

Since the mAP and mAR metrics are alone not enough to understand the connection between the semantic output of our model with the final instances from a quantitative point of view, we compared the result of the semantic segmentation metrics with those of the instance segmentation metrics. In Table 1 we observe a positive connection between the semantic segmentation metric $Prec_{TB}$ and the instance segmentation metric $F1_{IS}$. SkipFuse-U-DenseNet121 and FCN-ResNet50 RGB+DSM are those experiments with the highest $F1_{IS}$-scores and also the highest $Prec_{TB}$-scores. This shows that to obtain a good separation between building sections, it is most important to have complete touching borders, even if they do not have the highest $F1_{TB}$-scores.

5. CONCLUSION

Most approaches to building instance segmentation and instance segmentation in general use complex network architectures. Developed frameworks mostly present building segmentation on a semantic level only, which can be problematic when the exact number and boundaries of individual buildings are needed. We proposed a method for segmenting individual building sections on the instance level through the combination of classical and deep learning methods. The features come from RGB and DSM data. The segmentation is done through the successful detection of two classes: building mask and their touching borders. Pixel-wise predicted outputs are post-processed



(a) RGB

(b) GT

(c) Mask-RCNN-ResNet50

(d) SkipFuse-U-DenseNet121

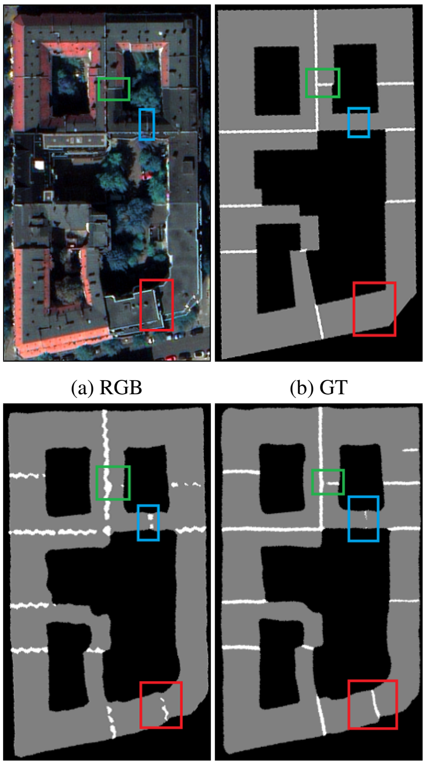
Figure 5. Comparison of the building section results of the (c) Mask-RCNN-ResNet50 trained on RGB and DSM and (d) the SkipFuse-U-DenseNet121 with the (a) RGB and (b) the ground truth. Each color represents a different building section.

with classical image processing methods (morphological operations and watershed labeling) to extract instances with the help of detected borders. Our experiments were done on high-resolution satellite imagery of city of Berlin, Germany. We show that the combination of deep learning and classical image processing methods can result in a good quality instance segmentation framework that reaches results considerably better than state-of-the-art methods. Multiple different neural networks based on the FCN and U-Net architecture proved effective on the task at hand. Our best model is the SkipFuse-U-DenseNet121, which fuses the RGB and DSM streams at the bottleneck and reduces the number of parameters by using summation instead of concatenation at the skip-connections and is reaching an mAP of 42 % and an $F1_{IS}$ of 47 %.

REFERENCES

Arefi, H., Reinartz, P., 2013. Building reconstruction using DSM and orthorectified images. *Remote Sensing*, 5(4), 1681–1703.

Beucher, S., Meyer, F., 2018. The Morphological Approach to Segmentation: The Watershed Transformation. *Mathematical Morphology in Image Processing*.



(a) RGB

(b) GT

(c) FCN-ResNet50 RGB+DSM

(d) SkipFuse-U-DenseNet121

Figure 6. Comparison of the segmentation results of the (c) FCN-ResNet50 trained on RGB and DSM and (d) the SkipFuse-U-DenseNet121 with the (a) RGB and (b) the ground truth.

Bittner, K., Adam, F., Cui, S., Körner, M., Reinartz, P., 2018a. Building footprint extraction from VHR remote sensing images combined with normalized DSMs using fused fully convolutional networks. *IEEE J. of Select. Topics in Appl. Earth Observ.s and Remote Sens.*, 11(8), 2615–2629.

Bittner, K., Adam, F., Cui, S., Körner, M., Reinartz, P., 2018b. Building Footprint Extraction From VHR Remote Sensing Images Combined With Normalized DSMs Using Fused Fully Convolutional Networks. *IEEE J. of Select. Topics in Appl. Earth Observ.s and Remote Sens.*, 11(8), 2615-2629.

Bittner, K., Körner, M., Reinartz, P., 2019. Late or earlier information fusion from depth and spectral data? large-scale digital surface model refinement by hybrid-cgan. *CVPRW*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255.

Etten, A. V., Lindenbaum, D., Bacastow, T. M., 2019. SpaceNet: A Remote Sensing Dataset and Challenge Series. arXiv:1807.01232v3.

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask r-cnn. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2980–2988.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. *CVPR*, 770–778.

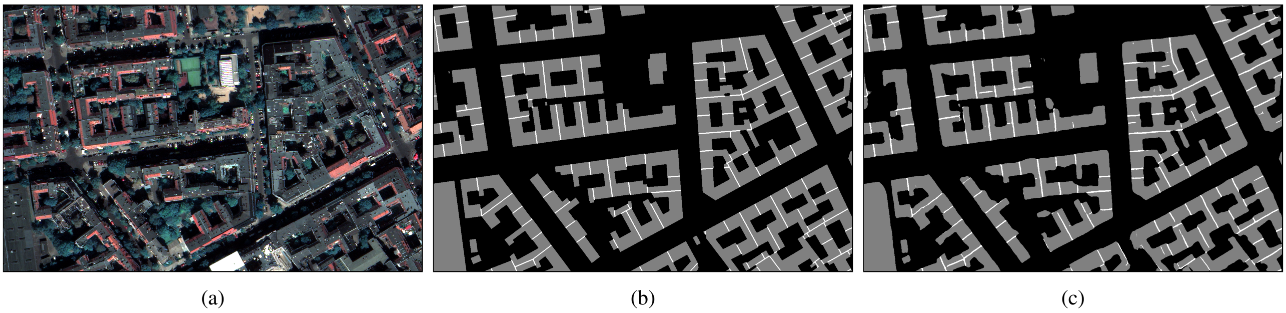


Figure 7. Test area sample showing (a) RGB, (b) ground truth and (c) raw prediction output from SkipFuse-U-DenseNet121 with RGB and DSM.

Henry, C., Hellekes, J., Merkle, N., Azimi, S. M., Franz, K., 2021. Citywide estimation of parking space using aerial imagery and osm data fusion with deep learning and fine-grained annotation. *ISPRS Archives*, XLIII-B2-2021.

Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K. Q., 2017. Densely connected convolutional networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2261–2269.

Huertas, A., Nevatia, R., 1988. Detecting buildings in aerial images. *Comput. Vision, Graph., and Image Process.*, 41(2), 131-152.

Iglovikov, V., Seferbekov, S., Buslaev, A., Shvets, A., 2018. Ternausnetv2: Fully convolutional network for instance segmentation. *CVPRW*, 233–237.

Jégou, S., Drozdzal, M., Vazquez, D., Romero, A., Bengio, Y., 2016. The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation. *arXiv e-prints*, arXiv:1611.09326.

Khan, A., Sohail, A., Zahoora, U., Qureshi, A., 2020. A survey of the recent architectures of deep convolutional neural networks. *Artificial Intell. Rev.*, 1 - 62.

Kingma, D. P., Ba, J., 2015. Adam: A method for stochastic optimization. *ICLR*.

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. *CVPR*, 3431–3440.

Luiz Ferreira de Carvalho, O., Abílio de Carvalho Júnior, O., Olino de Albuquerque, A., Castro Santana, N., Leandro Borges, D., Arnaldo Trancoso Gomes, R., Fontes Guimarães, R., 2021. Bounding Box-Free Instance Segmentation Using Semi-Supervised Learning for Generating a City-Scale Vehicle Dataset.

Potlapally, A., Chowdary, P. S. R., Raja Shekhar, S., Mishra, N., Madhuri, C. S. V. D., Prasad, A., 2019. Instance segmentation in remote sensing imagery using deep convolutional neural networks. *2019 International Conference on contemporary Computing and Informatics (IC3I)*, 117–120.

Roerdink, J., Meijster, A., 2003. The Watershed Transform: Definitions, Algorithms and Parallelization Strategies. *Fundamenta Informaticae*, 41.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., Fei-Fei, L., 2015. ImageNet Large Scale Visual Recognition Challenge. *Int. J. of Comput. Vision*, 115(3), 211-252.

Schuegraf, P., Bittner, K., 2019. Automatic Building Footprint Extraction from Multi-Resolution Remote Sensing Images Using a Hybrid FCN. *ISPRS Int. J. of Geo-Inform.*, 8(4).

Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. *ICLR*.

Zhao, W., Persello, C., Stein, A., 2020. Building instance segmentation and boundary regularization from high-resolution remote sensing images. *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*, 3916–3919.

Zheng, Y., Weng, Q., Zheng, Y., 2017. A hybrid approach for three-dimensional building reconstruction in Indianapolis from LiDAR data. *Remote Sensing*, 9(4), 310.