Building Section Instance Segmentation From Satellite Images Using Deep Learning Networks

Bachelorthesis
Julian Christopher Schnell, 2766850
Referentin TU Darmstadt, Institut für Geodäsie: Prof. Dr. Dorota Iwaszczuk
Betreuerin TU Darmstadt, Institut für Geodäsie: Emilia Lina Budde
Betreuer*Innen DLR: Ksenia Bittner, Corentin Henry

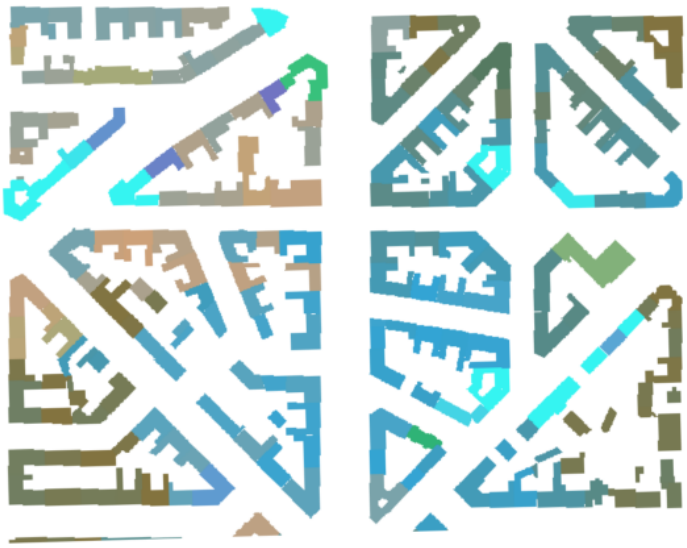
TECHNISCHE
UNIVERSITÄT
DARMSTADT

Bachelorarbeit – Aufgabenstellung

Building section instance segmentation from satellite images using deep learning

Instanzielle Segmentierung von Gebäuden aus Satellitenbildern mit Hilfe von Deep Learning Netzen

Das Ziel dieser Arbeit ist, zu untersuchen, wie Gebäude aus Satellitenbildern mit Hilfe von *deep learning networks* segmentiert werden können und die dabei verwendeten Verfahren zu vergleichen. Als zentrale Aufgabe gilt es nicht nur voneinander getrennte Gebäude einzeln zu instanziieren, sondern auch direkt miteinander verbundene Gebäude, z.B. in Gebäudeblöcken/-reihen getrennt voneinander zu segmentieren. Hierfür gibt es verschiedene Open Source Ground-truth Datensätze, wie beispielsweise auf folgendem Bild zu sehen ist:



Im Rahmen dieser Arbeit werden verschiedene neuronale Netze (z.B. Mask-RCNN) verwendet, um auf hochaufgelösten Satellitenbildern (z.B. WorldView4) eine Segmentierung durchzuführen und die Verfahren miteinander zu vergleichen. Es soll untersucht werden, ob die Deep Learning Netze Gebäudegrenzen aus der Vogelperspektive auf den Satellitenbildern „erkennen“ können. Besondere Herausforderung stellen hier geringe Unterschiede auf den Dächern dar, was in dieser Arbeit bewertet werden soll. Die Arbeit wird also darin liegen:

- die Datensätze zu laden und die Netzwerke möglicherweise an diese anzupassen
- zu untersuchen wie die Netzwerke für die Aufgabe abgeändert werden können, um die Qualität der Segmentierung zu erhöhen, z.B. durch das Verändern verschiedener Parameter
- die Ergebnisse aufzubereiten und miteinander zu vergleichen.

D. Iwaszczuk

Abstract

We present an end-to-end deep learning framework for building section instance segmentation. With the combined use of learning based approaches and classical image processing we were able to fulfil the task on World-View4 high resolution imagery and reach high quality results. We show that two well known but different deep learning models can tackle the issue with different architectures and inputs comparably. A ground truth raster image with pixel value 1 for buildings and 2 for their touching borders was generated to train the models to predict both as classes on a semantic output. Most developed frameworks present building segmentation on a semantic level only, which can be crucial when the exact number and boundaries of individual buildings is needed. In our work we post process the semantic outputs with the help of watershed labelling to generate segmentation on the instance level. The approach reaches F1-scores of up to 91.48% for buildings and 43.58% for touching borders.

Table Of Contents

List of Figures

Abbreviations

B	Building
CNN	Convolutional Neural Network
DSM	Digital Surface Model
EXP	Experiment
FCN	Fully Convolutional Network
GIS	Geographic Information System
GT	Ground Truth
IoU	Intersection over Union
RGB	Red-Green-Blue
RoI	Region of Interest
TB	Touching Borders

List of Tables

1. Introduction

There are numerous approaches to detect and segment buildings on an aerial or satellite image. Nowadays, large amounts of such images are available and often updated, since the surface on earth changes constantly. For analyzing and processing big datasets, the human's work efficiency gets worse and manual image classification much too time consuming.

Remote sensing and computer vision scientists find 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 count, urban planning, reconstruction, disaster monitoring and city modelling.

Classical methods to extract building footprints on an aerial or satellite image are based on the identification of edges and other primitive shapes buildings usually take up (Huertas, A. et al. 1988). For almost a decade, learning based approaches like machine learning and deep learning in particular have overtaken the state-of-the-art methodology for remote sensing problems. Convolutional neural networks (CNNs) used for image segmentation have been developed and improved constantly since then.

In image segmentation two methods need to be differentiated. One is to segment every pixel to a certain class but to not differentiate between individual objects which is called semantic segmentation. The other is to detect objects and their boundary boxes and give them each an individual instance, which is called instance segmentation. Another approach is to combine the two, which is called panoptic segmentation, a term first introduced in 2019 (Kirillov, A. et al. 2019).

In this thesis, the use of deep neural networks to segment buildings on a high resolution satellite image of Berlin City on an instance level is presented. Since the outputs of deep learning based methods are not perfect we use classical methods to post process them, which is a great addition to the results generated by the neural networks.

The remainder of this work will first give some background information and go through some related work to give a frameset of comparable state-of-the-art methods. Afterwards the methodology used for the task will be explained in detail. It goes on with the experiments as well as the display and evaluation of the results and finally, these results will be discussed and in the end concluded with a brief outlook on potential improvements.

2. Related Work

To make clear what comparable state-of-the-art methods regarding the given task look like, in this chapter, the evolution of deep learning networks used for image recognition are introduced.

2.1. Semantic Segmentation with Artificial Neural Networks

In the recent decade the field of deep learning has grown rapidly (Sultana, F. et al. 2020). This has also had a huge impact on remote sensing methods and image recognition/segmentation in particular. Since the rise of CNNs through the release of *AlexNet* in 2012 the architectures have changed but the basic idea of CNNs has dominated until today (Krizhevsky, A. et al. 2012). *AlexNet* is a deep convolutional network designed to tackle the *ImageNet* classification challenge and proved exemplary performance in the 2012 edition of the challenge. Three years later researchers developed a network known as *ResNet* (He, K. 2015). They present a deep residual network with several options differing in depth and the amount of convolution layers. This model's deepest *ResNet152* version only showed an error of 3.57 % on the *ImageNet* classification challenge, which is better than humans' error on the same task (5 %) (Khan A. et al. 2019). Until today this architecture proves huge significance for the design of newer models. In the same year a team of researchers released models called fully convolutional networks *(*FCNs*)*. This pixel wise prediction method was first an adaption to older networks. Networks like *AlexNet* produce non-spatial outputs, so these connected layers were casted into FCNs that take input of any size and output classification maps (Long, J. et al. 2015). This development has also had big impact on the work in this thesis. Soon after, a team of researchers in Freiburg described a new approach which they named *U-Net* referring to its architecture's shape (Ronneberger, O. et al. 2015). The network is based on FCNs, but in comparison works with less training data and yields more precise predictions, which is due to the use of many skip connections.

The whole progression of deep neural networks for semantic image segmentation can be described exemplarily by observing the evolution of *DeepLab.* The first version of this series was released in 2014 and showed state-of-the-art results on several challenges. It uses pretty much the same backbone as FCNs only with some changes in the output layers and some added complexity in the classification layers (Chen, L. et al. 2014). The core idea in *DeepLabv2,* first published in 2016 with added atrous convolution and conditional random fields stays the same. The version is extended with a backbone architecture from *ResNet* (Chen, L. et al. 2016). Just one year later this idea was updated again, with the release of the third version: *DeepLabv3* (Chen, L. et al. 2017). The atrous convolution method was changed and the conditional random fields were removed. Another year later, the same scientists released the latest version, which they named *DeepLabv3+* (Chen, L. et al. 2018). For this network the backbone was changed to *Xception* (Chollet, F. et al. 2016), since this framework has proved to bring similar results as *ResNet* with less computation power needed. The major improvement of this version was to implement the encoder-decoder design into the architecture. Encoding images into smaller feature vectors with the use of a backbone network and then decoding the features to upsample them by using convolution layers is the state-of-the-art method until today. The final architecture of the latest version is displayed in figure 2-1.

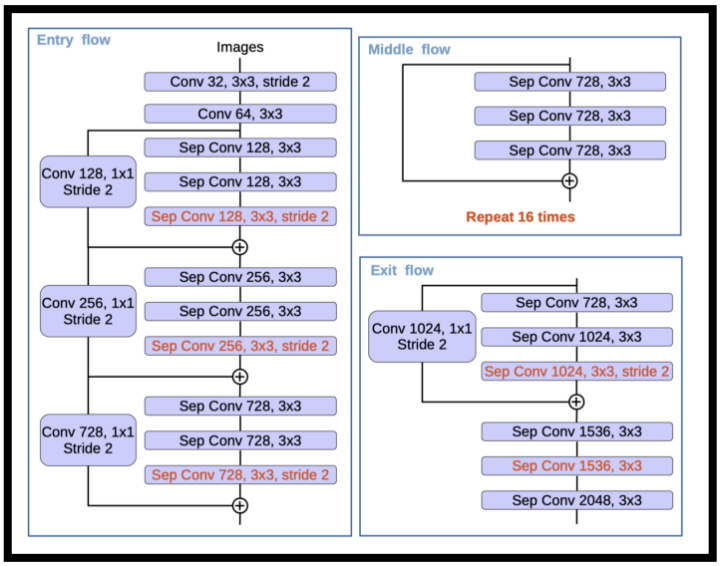


Figure 2-1: Architecture of *DeepLabV3+*

With this design and pre-training on different datasets, they were able to achieve an 82.1 % intersection over union (IoU) score on the Cityscapes dataset (Cordts, M. et al. 2016), which was a leading result at the time of the release.

2.2. State-of-the-art-Building Instance Segmentation with Learning Based Methods

In January 2018, a group of scientists proposed a novel network (*TernausNet*) to perform semantic segmentation of buildings on aerial images (Iglovikov, A. et al. 2018). With a *U-Net* architecture, including a pre-trained encoder they achieved an IoU score of up to 68.7 %, trained and tested on RGB images only. The results of this paper stay on the semantic level.

Later that year the same group of scientists published an updated version of this model (*TernausNetV2*). The method is now able to do segmentation of buildings on the instance level (Iglovikov, A. et al. 2018). The exact architecture of this network is explained later in this thesis. With an IoU score equal to 74 % with 11 input channels (RGB+panchromatic+multi-spectral), the work sets the state-of-the-art method during the time.

Another group of researchers tried building segmentation through gated graph convolutional neural networks on a very high resolution dataset (5 cm ground resolution), and reached good results with this method (Shi, Y. et al. 2019). Their best experiment reaches an F1 score of 69 % and an IoU score of 53 % for building footprint segmentation on the Planetscope dataset (Planet Team 2017).

Lastly introduced is a dataset and segmentation framework called *SkyScapes* (Azimi, S. M. et al. 2019). On a part of the dataset which contains 13 classes, including buildings, trees, cars etc. their proposed method reaches an IoU score of 51.93 % and 89.16 % on the building class. Considering that these experiments contained a lot more classes to be predicted than the ones introduced before, this is an approach claiming state-of-the-art results on the dataset.

3. Methodology

The upcoming chapter explains the methodology and framework used to carry out the experiments. It starts with the description of the dataset. Furthermore the models that are trained on the task are introduced and lastly the pre and post processing of the data is presented.

3.1. Dataset

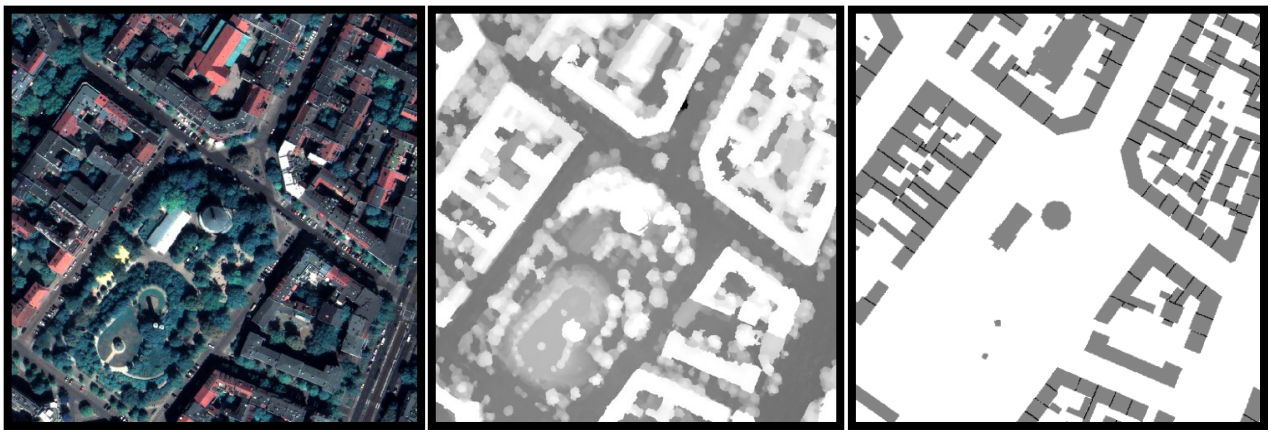


Figure 3-1: Left to right: example RGB tile, DSM tile and GT tile (buildings in grey and touching borders in black)

The dataset used for this task consists of an RGB image and an optional digital surface model (DSM) as the input, as well as a raster image building mask as ground truth (GT). Example tiles can be seen in figure 3-1. The RGB image originates from high resolution *WorldView-4* imagery and shows the city of Berlin, Germany. Its ground resolution is 0.3 m. The image contains three channels (red, green, blue) with a height of 33206 and a width of 32229 pixels. The DSM was resampled to the same size and ground resolution as the RGB image. The GT raster is converted to the same resolution and size as the input images. 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.

With the help of geo information system (GIS) applications, the touching borders (TB) between buildings were calculated. Afterwards these contours were 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 (B) and 2 for TB.

The images are then split into three RoIs (regions of interest). This is important to make sure the networks are performing well on different images than they were trained on. A standard approach is to split the dataset into a training, a validation and a testing set. In the following experiments training uses ~85 %, validation ~12.5 % and testing ~2.5 % of the image. This results in 3442 patches for training, 538 for validation and 102 patches for testing. The training dataset is used to train the models, validation set to evaluate how well the experiment works during training and the testing set in order to check how the model behaves on a completely unused part of the dataset.

The final step of preparing the dataset is to normalize all input data before training. Data normalization is very important for artificial neural networks. First the images' intensities are rescaled

by computing the histogram for all pixel values to reassign all values lower than 2 % and higher than 98 % of the quantile within this very range, so the bell shape of the histogram lies between the same minima and maxima. Afterwards the values are rescaled to the range [-1, 1].

3.2. Models

There are numerous approaches and models publicly available to possibly fulfil the task given. In the following, the two models selected for the experiments are described as well as why they are chosen for this work.

3.2.1. TernausNetV2

TernausNetV2 is an updated version of a deep convolutional network specifically designed for building segmentation. Its architecture uses a classic Encoder-Decoder approach (*U-Net*) with skip connections between all blocks of the same size (Iglovikov, A. et al. 2019)

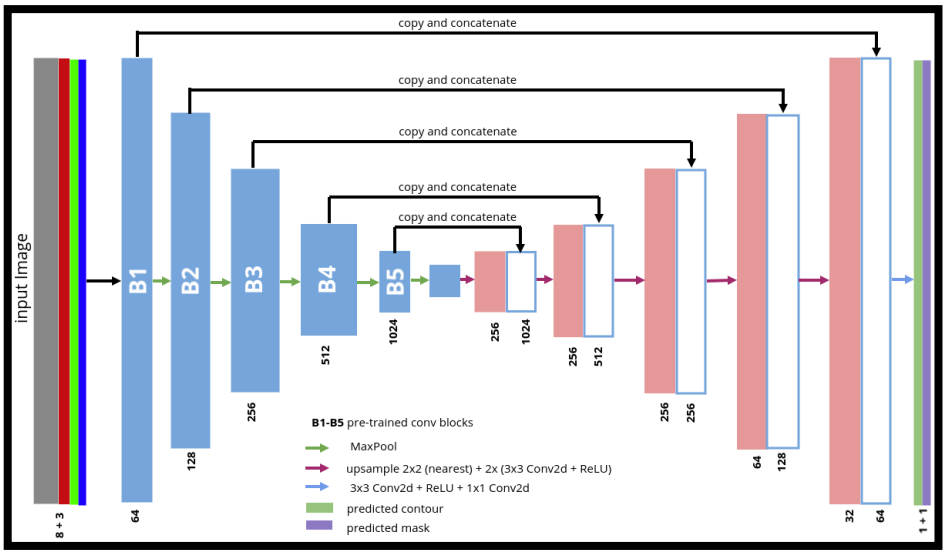


Figure 3-2: *TernausNetV2* architecture (Iglovikov, V et al. 2018)

As seen in figure 3-2 the contracting path (encoder - blue blocks on the left) follows a classic architecture of other convolutional networks. Each layer of the encoder can detect more complex patterns and semantics of the image. The expansive part (decoder - pink/white blocks on the right) does the upsampling of the feature maps each followed by a series of convolutional layers. Skip connections between encoder and decoder blocks of the same size have a positive impact on the recovery of semantics to the output. TernausNetV2 uses more skip connections than any other FCN designed for image segmentation and is therefore a great example for the task.

The model was originally optimized for a multi-channel input (11 in the proposed experiments), but is easily adjustable to work with less channels. The output generated was also adjusted. In the original experiment the model outputs two channels with the touching borders and full building mask each predicted on one. In this work the output is a one channel image with three values connected to each class (as described in chapter 4).

3.2.2. FCN-ResNet50

As mentioned before, *ResNet* was a big breakthrough in the evolution of deep neural networks. *ResNet50* is a standard deep residual network with a 50-layer architecture as seen in figure 3-3. Due to the popularity of this model and its outstanding performance on classic semantic segmentation tasks, this network was chosen as an addition and comparison. The basic construction of this network was inspired by the philosophy of *VGG* nets (Simonyan, K. et al. 2015).

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 3-3: *ResNet* architectures (He, K. et al. 2015)

With the help of the residual blocks design, spatial details can bypass a layer which would otherwise make the network lose accuracy and the model can therefore keep the shapes of features significantly better. This is why the model was chosen as a comparison, since its strengths are of a different matter.

3.3. Pre and Post Processing

The outputs of deep learning based methods are never perfect, which is why post editing them with classical 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 have in common is the imperfectly predicted touching borders class as seen in figure 3-4.

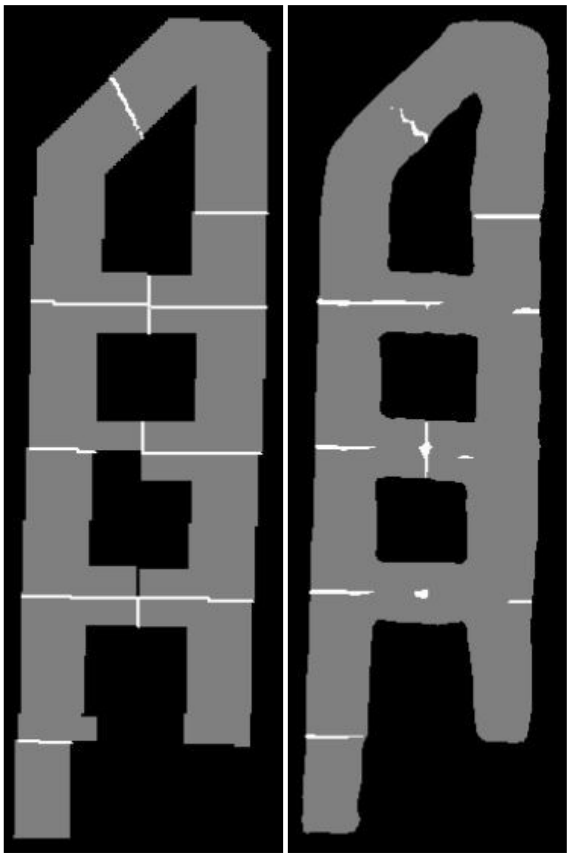


Figure 3-4: Left: GT, right: raw output

There are three different mistakes when it comes to the prediction of touching borders (class 2). The first is, that the model simply does not predict a border at all, the second is that the border is not predicted completely and the third is that a false border is predicted. The first and third error cannot be eliminated with post processing and are mistakes to be accepted and discussed. The second error is what we can almost fully eliminate with image processing methods.

In order to get results on the instance level, we need to adjust the predicted touching borders. A watershed algorithm is used to separate individual buildings (Roerdink, J.B.T.M. et al. 2001). This algorithm will classify separated blobs of pixels as one instance. First the predicted touching borders are dilated with a radius of 12 pixels in a disk shape (Van der Walt, S. et al. 2014). Now nearly all of the incomplete borders are continuous and reach from building edge to building edge as seen in figure 3-5, where the grey blobs show the dilated building borders, whereas the unprocessed ones are the black lines in the background.

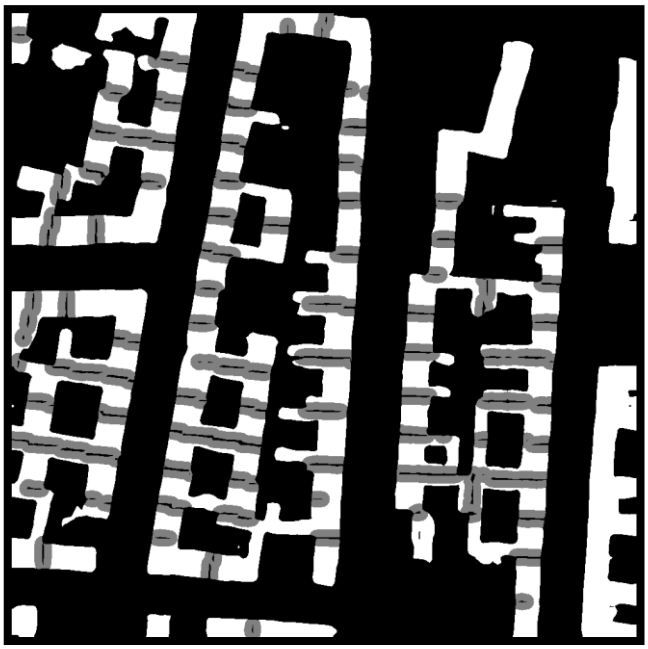


Figure 3-5: Semantic output from exp. 3 (patch 0.2) with dilated touching borders

These dilated borders are now subtracted from the building mask which will enable us to apply the watershed algorithm on now completely separated pixel groups. These are finally transferred to the original output's building mask, so the semantic masks were successfully transformed into images classified on the instance level. To evaluate the results, we generate an image with our GT mask the exact same way, but without dilation of the borders, as seen in figure 3-6.

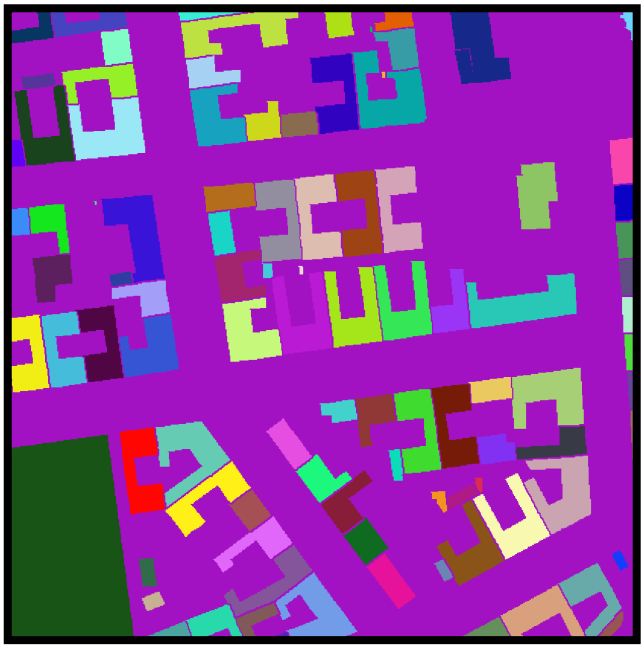


Figure 3-6: GT, instance segmented

4. Experiments

To evaluate the approach, first the training method of the different models is explained then the results are displayed and lastly quantitatively evaluated by using several metrics to compare all final experiments:

- Experiment 1: *TernausNetV2* with three channel input
- Experiment 2: *TernausNetV2* with four channel input
- Experiment 3: *FCN-ResNet50* with three channel input

4.1. Training

The models are each trained for a duration of 100 epochs, where one epoch consists of iterations on the whole training set. Training set consists of 3442 patches with a size of 512×512 pixels. Batches with the size equal to 4 are fed to the network. The learning rate was set to 0.0001 with a scheduled decrease by 10 percent after every epoch (exponential decay). The loss is calculated with a cross entropy function *L:*

$$L(p) = -(y \times log(p) + (1-y) log(1-p)) \quad (1)$$

where p is the predicted probability and y is the one hot encoded ground truth:

- Background: [1,0,0]
- Building: [0,1,0]
- Touching border: [0,0,1]

The network's parameters are then updated with the stochastic gradient descent by the use of an *Adam* optimizing method (P. Kingma, D. et al. 2017). Since the TB only take up 0.46 % of all pixels in the GT raster, the relating class was weighted up with a factor of two in the optimizer. After every epoch the model is evaluated on the validation set, and the validation loss is calculated correspondingly. The typical development of both loss-values is displayed in the graphs of figure 4-1 (both curves smoothed with factor 0.8).

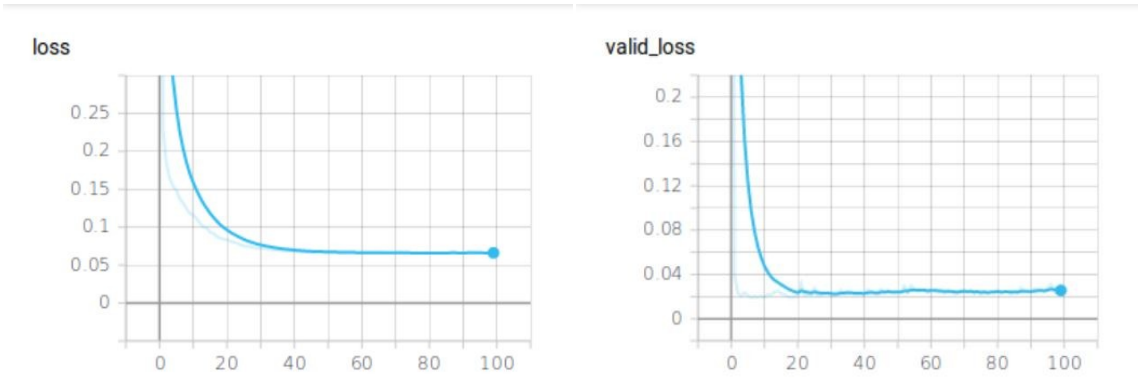


Figure 4-1: Left: loss curve, exp. 3, right: validation loss, exp. 3

After a *Softmax* function assigns decimal probabilities for each class, the final outputs are generated using an *arg max* function. This function returns the indices of the maximum values of a tensor across a dimension, where the indices are the probabilities resulting from the *Softmax*. This enables the visualization of the final three value mask.

4.2. Results

To give a quick reminder of the different steps performed by the model and algorithm, in figure 4-2 the process from RGB to segmentation on the instance level is displayed. These images come from GT and therefore show the wanted outcome of the experiments, both for the semantic and the instance level.

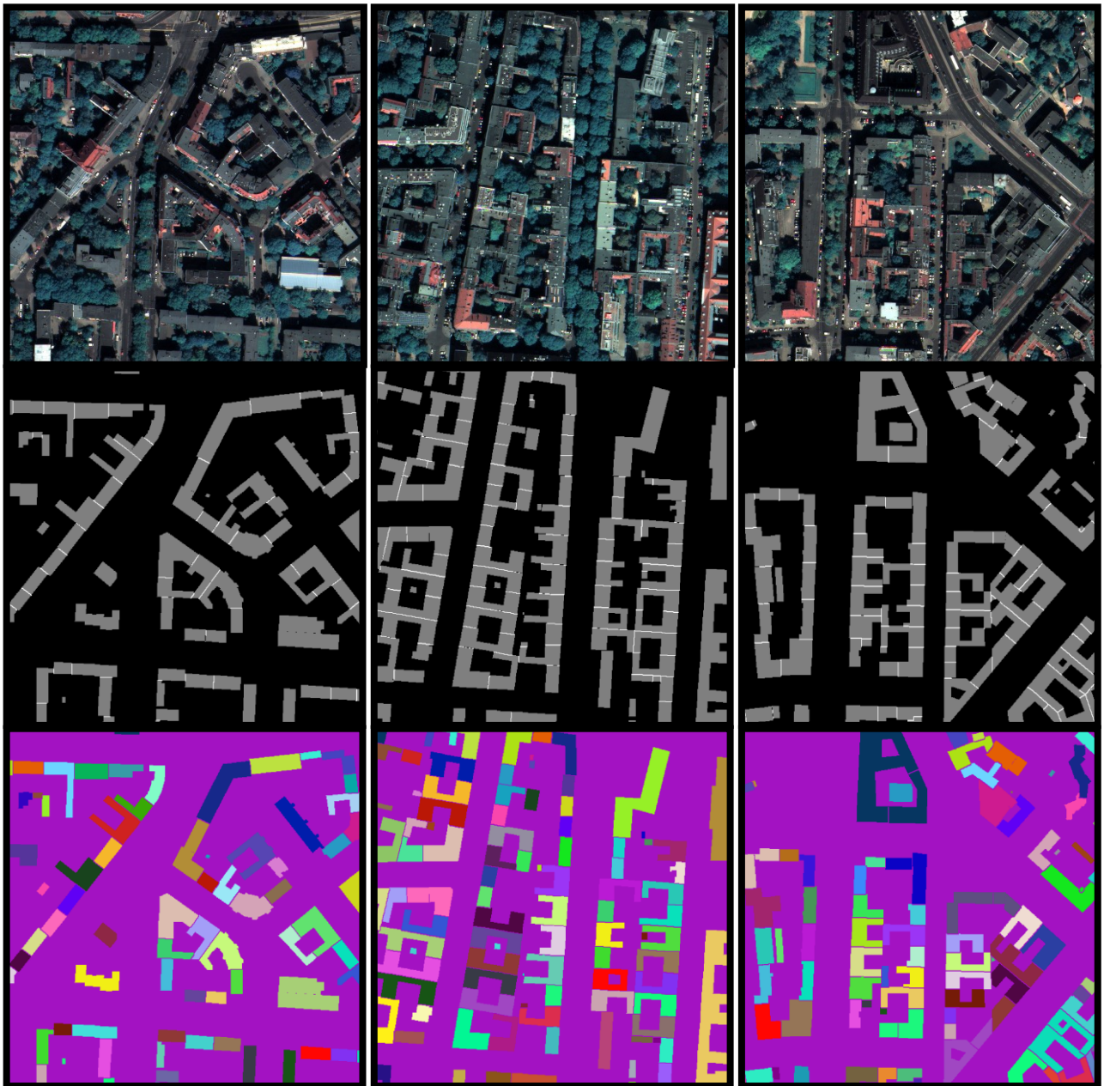


Figure 4-2: Three example patches from test set. Top to bottom: RGB, GT, instance GT

The outputs from testing the models on the test data set come in tiles/patches with a size of 1024×1024 pixels, because the graphics processing units (GPUs) available don't provide the memory to process the whole test area at once. To parallelly compare the semantic output with the GT mask, the tiles were put back together as seen in figure 4-1. This gives a good first overview of the performance.

Figure 4-1: Semantic masks (test RoI). Left: GT, right: output exp. 3

Finally, in the following figures the instance segmented building masks can be seen. The post processing of the outputs and application of the watershed algorithm to the masks as explained in chapter 3.3 creates an image of buildings segmented on the instance level. Each colour represents an instance/an individual building, while purple represents the background (0).

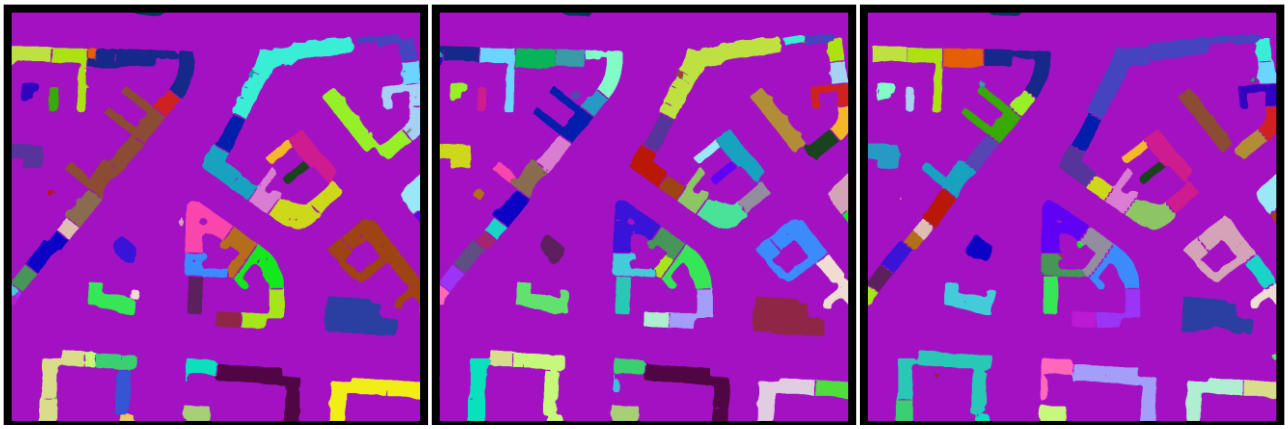


Figure 4-2: Results patch 0.0. Left to right: exp. 1-3

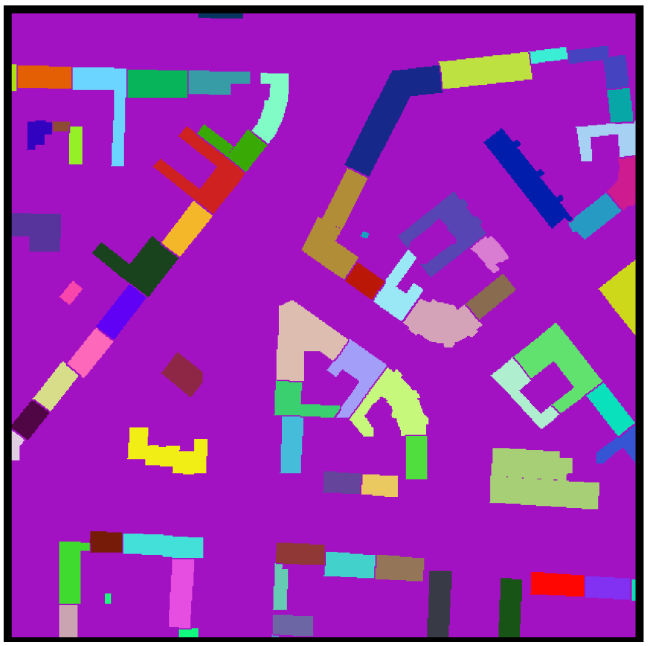


Figure 4-3: instance GT, patch 0.0

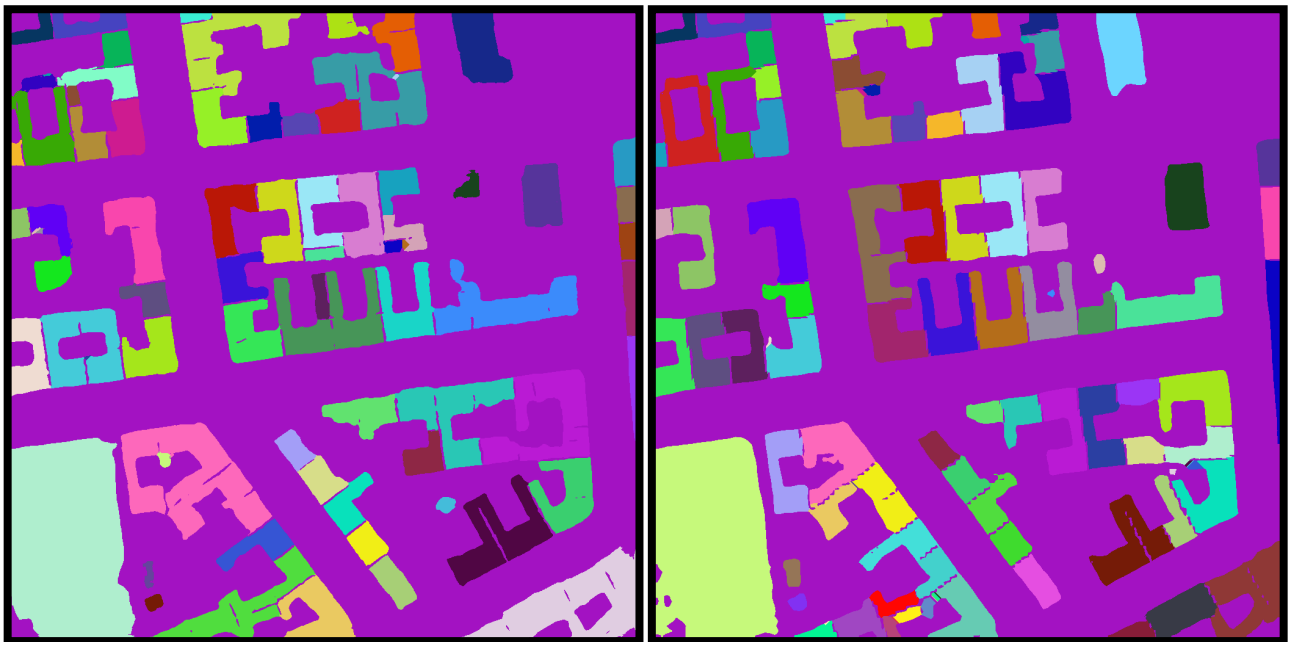


Figure 4-4: Results patch 2.1. Left: exp. 2, right: exp. 3

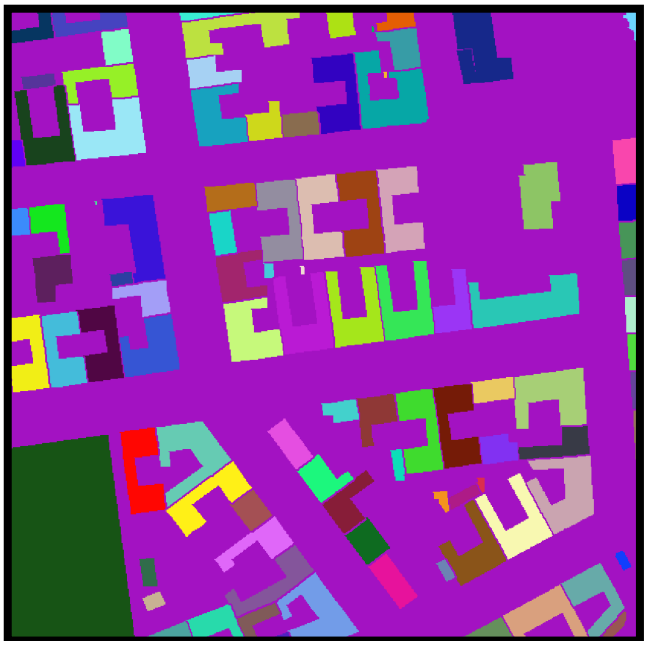


Figure 4-5: instance GT, patch 2.1

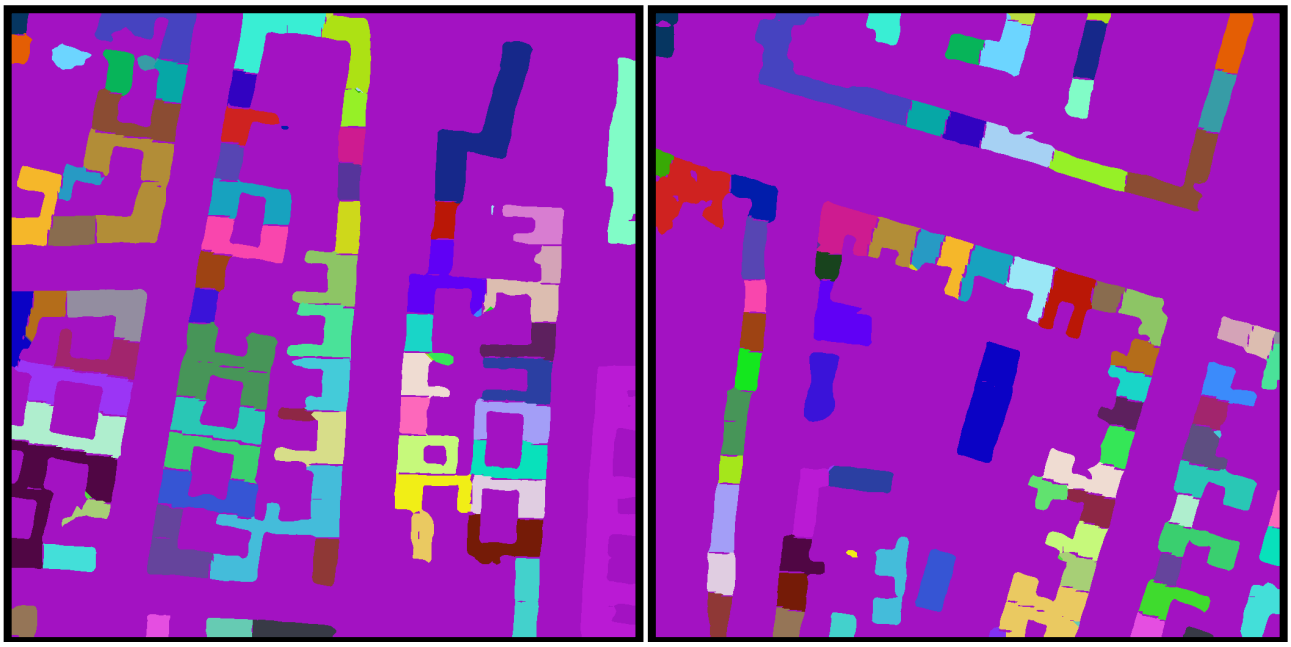


Figure 4-6: Left: patch 0.2 exp. 3, right: patch 0.4 exp.3

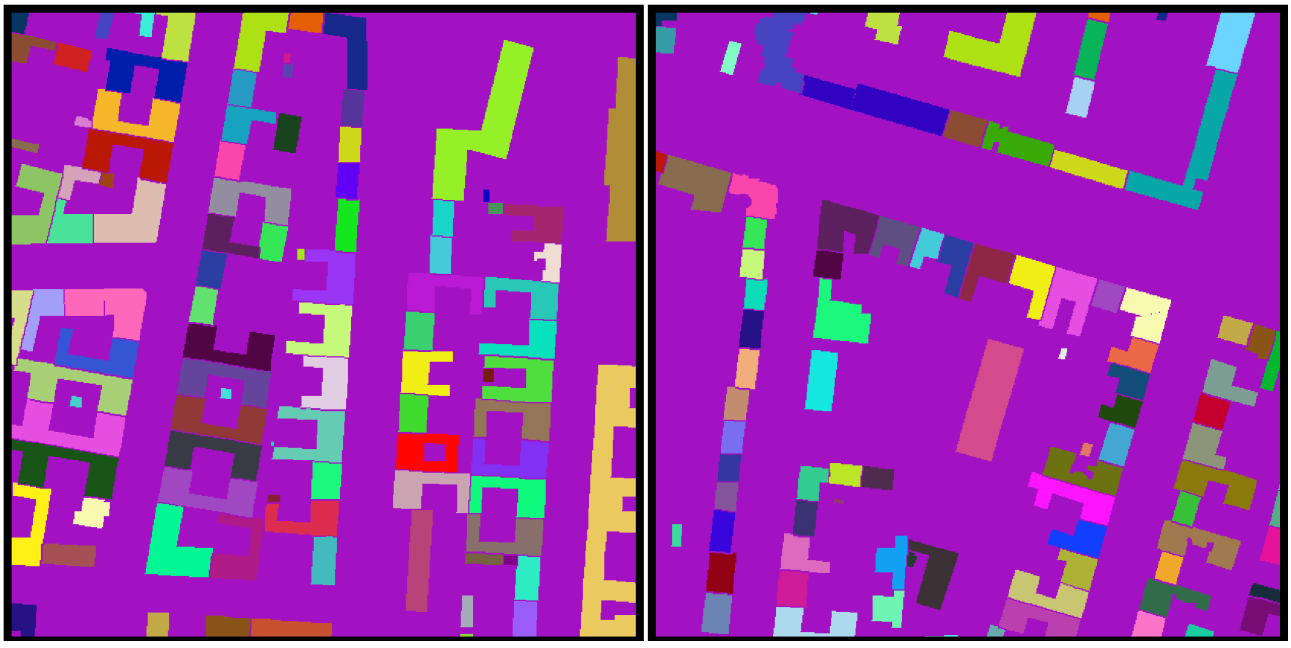


Figure 4-7: Left: patch 0.2 GT, right: patch 0.4 GT

4.3. Evaluation and Comparison

		Predicted	
Actual		**Negative**	**Positive**
	Negative	True Negative	False Positive
	Positive	False Negative	True Positive

Table 4-1: Confusion Matrix

To evaluate the outcome of an experiment in semantic segmentation several metrics are used. From the experiments' outputs of the test set each binary mask is extracted to be evaluated individually. The two binary masks from the output are compared to the two binary masks of the GT image. To do this the confusion matrix as seen in table 4-1 is generated. Afterwards, these values are taken to calculate metrics that express the performance of the models.

Each pixel gets classified as one of the following groups. True Positives are the number of pixels that belong to the class and are classified such, False Positives the ones that do not belong to the class but are classified such, True Negatives the ones that do not belong to the class and are not classified such and last there are False Negatives, which are all pixels that belong to the class but are not classified such.

$$\textbf{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positives} \tag{2}$$

$$\textbf{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negatives} \tag{3}$$

$$\textbf{F1} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

The average precisions and recalls (equation 2 and 3) of all test image tiles are taken to get the mean F1-measure (equation 4), which is a widely used score for quality estimation in image detection (Kirillov, A et al. 2019).

Building Mask	Average Precision	Average Recall	Average F1
TernausNetV2 – 3C	90.12 %	90.25 %	90.16 %
TernausNetV2 – 4C	90.53 %	90.63 %	90.55 %
ResNet50 – 3C	91.73 %	91.84 %	**91.48 %**

Table 4-2: Evaluation metrics, building class

In table 4-2, the metrics for the class 1 (building) are displayed for all three experiments. *FCN-ResNet50* with RGB input only performs best, with a F1-score of 91.48 %.

Touching Borders	Average Precision	Average Recall	Average F1
TernausNetV2 – 3C	43.54 %	31.90 %	36.68 %
TernausNetV2 – 4C	48.49 %	40.03 %	**43.58 %**
ResNet50 – 3C	44.52 %	35.05 %	35.33 %

Table 4-3: Evaluation metrics, touching borders

In table 4-3 we can see the same metrics for class 2 (touching borders) and can conclude that here *TernausNetV2* with RGB+DSM as input channels performs the best in a quantitative manner with a F1-score of 43.58 %. The significantly lower score for class 2 is to be expected since the class is much harder to learn due to its share of pixels and its complexity.

Another common method to evaluate the outcome of an image segmentation experiment is the IoU-score. This score indicates the percent overlap of the target mask and the prediction output.

IoU-score	
Exp. 1	60 %
Exp. 2	66 %
Exp. 3	62 %

Table 4-4: IoU-scores exp. 1-3

In table 4-4 the IoU scores for all three experiments are displayed. It is the average score calculated on the whole validation set after the last epoch of training. The outcome is similar to the outcome of the evaluation described previously. Experiment 2 performs best with an IoU of 66 %.

5. Discussion

TernausNetV2 performs well with 3 channels, but DSM is a useful addition to the input, which makes sense, since buildings are next to trees the main thing to elevate from the ground in urban areas. Also touching borders between two buildings are often noticeable through the different height of two bordering buildings as seen in figure 5-1.

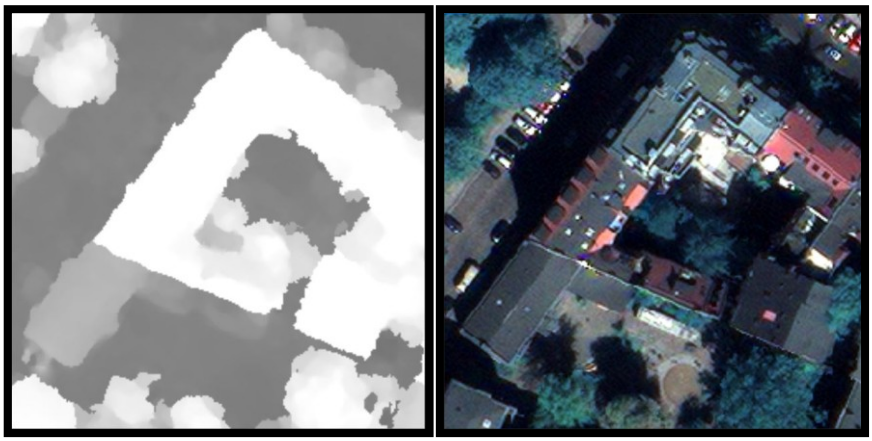


Figure 5-1: Left: DSM cutout, right: matching RGB cutout

As technological development goes on aerial and satellite images are publicly available for the whole surface of the earth, DSMs aren't, especially not in such high resolutions. Other works have shown that fusing DSMs into training can make an essential difference.

Qualitative evaluation leads to the conclusion that the instance segmentation resulting from experiment 3 (*FCN-ResNet50*) come out in the highest quality. The comparison of these to the ground truth instances show the most similarities and the shapes of the buildings are the cleanest with the least outliers.

The outcome of the experiments results in what was expected from the different models. *TernausNetV2* performs very well in quantitative manner, while the architecture of the *FCN-ResNet50* puts out cleaner shapes and therefore better results in a qualitative manner.

The models perform best for the most common architectures of buildings, which are mostly residential houses that show typical similarities in shape and roof structure as seen in figure 5-2.

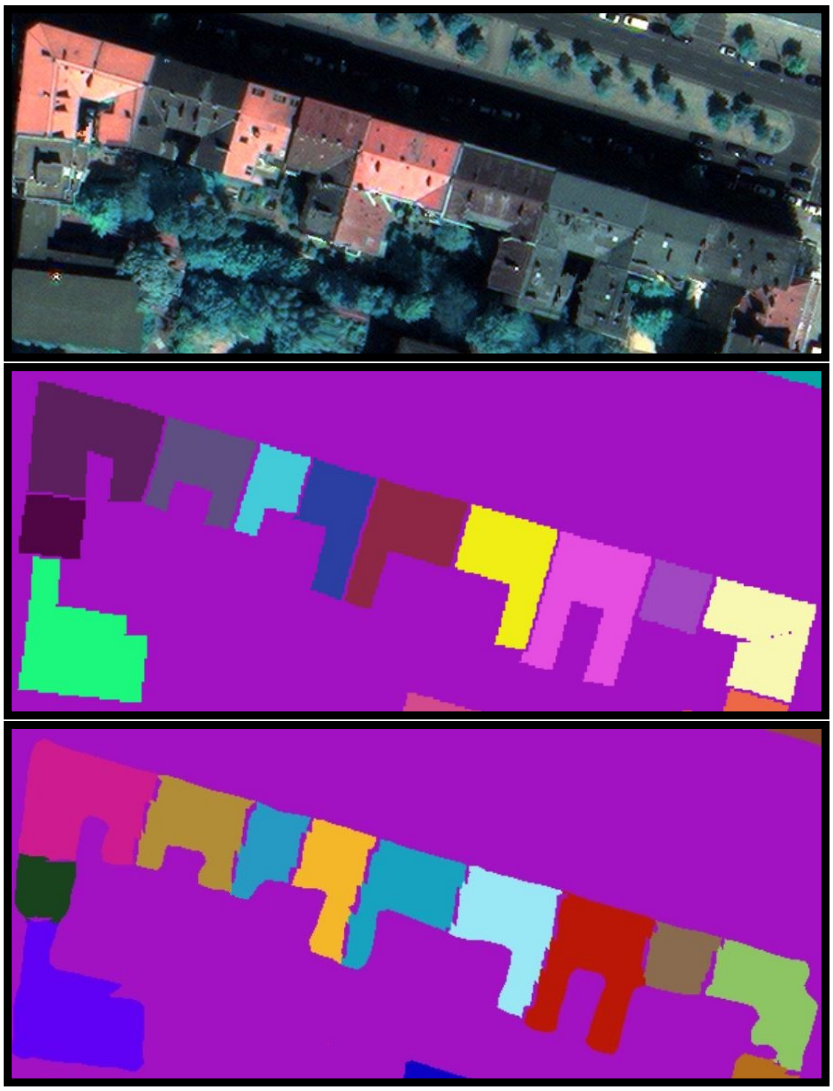


Figure 5-2: Top to bottom: RGB, GT, output exp. 3

Usual mistakes happen when the building is partly covered by a tree or the shadow of another building. Generally the segmentation does not work well for conditions like buildings in construction, smaller huts, industrial areas or buildings with a unique design as seen in figure 5-3.

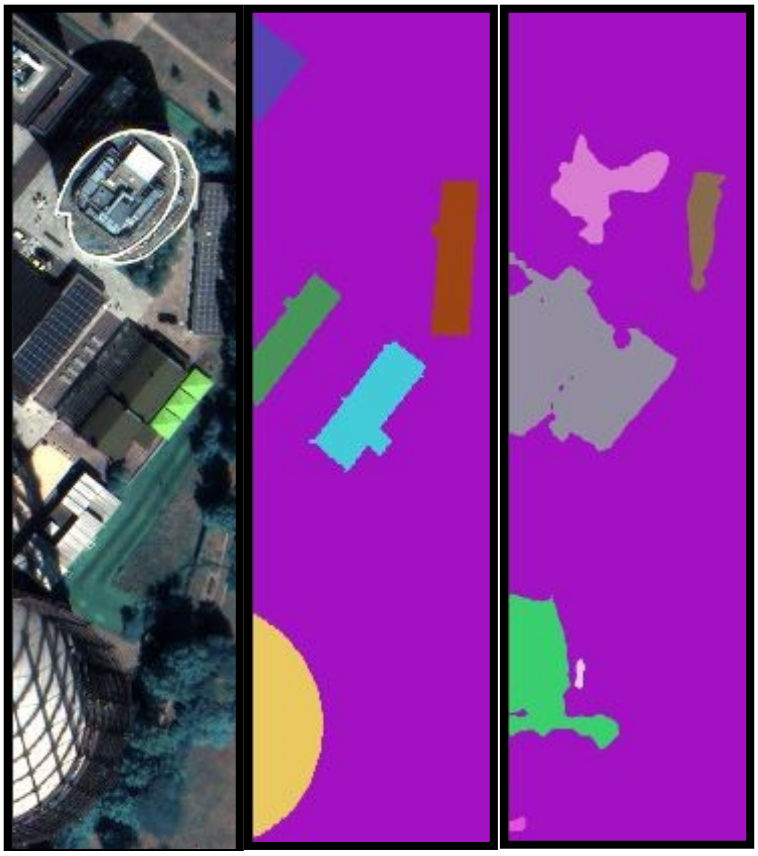


Figure 5-3: Left to right: RGB, GT, output exp. 3

From inspecting figure 5-3, the assumption can be made that the dataset is sometimes wrong or does not match the input images completely. In the top part of the cutout there seems to be a round building that does not appear on the GT image.

6. Conclusion

It was shown that the combination of learning based methods with classical methods of image analysis can provide building instance segmentation on high resolution satellite imagery of high quality. The results reached similar evaluation metrics as the reference state-of-the-art experiments, even with less training data. Two neural networks of different designs and functionality both proved to be working well for the task.

As generalization is a well known problem faced in remote sensing tasks, bigger datasets from different areas could make the models' accuracy higher. For now it is expected that the models would get good results for images of central European cities or regions with similar climate and therefore construction methods only, since otherwise the buildings' architecture would be drastically different and deep learning methods are very sensitive to such changes. With the use of training data from completely different cities or even rural areas, the models could end up working well on a more global scale.

With deep learning methods and computation power changing progressively, the work of this thesis leaves ideas and potential to be further improved. Newer architectures and recent developments like panoptic segmentation methods will have a big impact on the quality of image classification.

7. References

Azimi, S. M.; Henry, C.; Sommer, L.; Schumann, A.; Vig, E. (2019): SkyScapes – Fine-Grained Semantic Understanding of Aerial Scenes

Bittner, K.; Körner, M.; Reinartz, P. (2015): Late or Earlier Information Fusion from Depth and Spectral Data

Chen, L.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A. L. (2014): Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs

Chen, L.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A. L. (2016): DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs

Chen, L.; Papandreou, G; Schroff, F.; Adam, H. (2017): Rethinking Atrous Convolution for Semantic Image Segmentation

Chen, L.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. (2018): Encoder-Decoder with Atrous Seperable Convolution for Semantic Image Segmentation

Chollet, F. (2016): Xception: Deep Learning with Depthwise Separable Convolutions

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. (2016): The Cityscapes Dataset for Semantic Urban Scene Understanding

He, K.; Zhang, X.; Ren, S.; Sun, J. (2015): Deep Residual Learning for Image Recognition

Huertas, A.; Nevetia, R. (1988): Detecting Buildings in Aerial Images

Iglovikov, V.; Shvets, A. (2018): U-Net with VGG-11 Encoder Pre-Trained on ImageNet for Image Segmentation

Iglovikov, V.; Seferbekov, S.; Buslaev, A. (2018): TernausNetV2: Fully Convolutional Network for Instance Segmentation

Khan, A.; Sohail, A.; Zahoora, U.; Qureshi, A. S. (2019): A survey of the Recent Architectures of deep Convolutional Neural Networks

Kingma, D. P.; Ba, J. (2017): Adam: A Method for Stochastic Optimization

Kirillov, A.; He, K.; Girshick, R.; Rother, C.; Dollár, P. (2019): Panoptic segmentation

Krizhevsky, A.; Sutskever, I.; Hinton, G. E. (2012): ImageNet Classification with Deep Convolutional Neural Networks

Long, J.; Shelhamer, E.; Darrell, T. (2015): Fully Convolutional Networks for Semantic Segmentation

O' Mahony, N.; Campbell, S.; Carvalho, A.; Harapanahalli, S.; Hernandez, G. V.; Krpalkova, L.; Riordan, D.; Walsh, J. (2019): Deep Learning vs. Traditional Computer Vision

Planet Team (2017): Planet Application Program Interface: In Space for Life on Earth

Roerdink, J.B.T.M.; Meijster A (2001): The Watershed Transform: Definitions, Algorithms and Parallelization Strategies

Ronneberger, O.; Fischer, P.; Brox, T. (2015): U-Net: Convolutional Networks for Biomedical Image Segmentation

Shi, Y.; Li, Q.; Zhu, X. X. (2019): Building Segmentation through a Gated Graph Convolutional Neural Network with Deep Structured Feature Embedding

Simonyan, K.; Zisserman, A (2015): Very Deep Convolutional Networks for Large-Scale Image Recognition

Sultana, F.; Sufian, A.; Dutta, P. (2020): Evolution of Image Segmentation using Deep Convolutional Neural Networks: A Survey

Van der Walt, S.; Schönberger, J. L.; Nunez-Iglesias, J.; Boulogne, F.; Warner, J. D.; Yagar, N.; Gouillart, E.; Yu, T. (2014): Scikit-Image: Image Processing in Python

Xiong, Y.; Liao, R.; Zhao, H.; Hu, R.; Bai, M.; Yumer, E.; Urtasun, R. (2019): UPSNet: A Unified Panoptic Segmentation Network

Zhang, A.; Liu, X.; Gros, A.; Tiecke, T. (2017): Building Detection from Satellite Images on a Global Scale

Abschlussarbeit von

Herr Julian Christopher Schnell
Matrikelnummer: 2766850

Erklärung zur Abschlussarbeit gemäß § 22, Abs. 7 APB

Hiermit versichere ich, die vorliegende Abschlussarbeit ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben.

Alle Stellen, die aus den Quellen entnommen wurden, sind als solche kenntlich gemacht worden.

Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 29.03.2021

Unterschrift