Fakultät für
Informatik und
Mathematik

HOCHSCHULE
FÜR ANGEWANDTE
WISSENSCHAFTEN
MÜNCHEN

Automatic building footprint extraction from multiple remote sensing images with different spatial resolution using a hybrid fully convolutional neural network

Philipp Schuegraf

Bachelorarbeit Scientific Computing

Examiner:
Prof. Dr. E. Eich-Soellner, University of Applied Sciences Munich

Supvervisor:
K. Bittner, German Aerospace Center

02.01.2019

Declaration of Authorship

Philipp Schuegraf, born on 24.05.1994 (IC7, WS 18/2019)

I hereby declare that the bachelor thesis submitted is my own unaided work. All direct or indirect sources used are acknowledged as references. I am aware that the bachelor thesis in digital form can be examined for the use of unauthorized aid and in order to determine whether the bachelor thesis as a whole or parts incorporated in it may be deemed as plagiarism. For the comparison of my work with existing sources I agree that it shall be entered in a database where it shall also remain after examination, to enable comparison with future bachelor theses submitted. Further rights of reproduction and usage, however, are not granted here. This paper was not previously presented to another examination board and has not been published.

Munich, 02.01.2019

. . . . . .

Signature

Abstract

We present an end-to-end deep learning framework for the integration of depth and spectral information for the task of building footprint extraction. Technical developements made it possible to supply large-scale satellite image coverage. This poses the challenge of efficient discovery of imagery. One very important task in applications like urban planning and reconstruction is to automatically extract building footprints. Recently, deep neural networks where extended from image classificators to image segmentators, allowing to densely predict semantic labels. We show that a UNet enhances the boundary quality in building footprint extraction compared to a FCN4s. Usually, satellites provide a high-resolution panchromatic image, but only a low-resolution multispectral image. We tackle this issue by using a residual neural network block to fuse both and then feed them to a UNet. In a parallel stream, a stereo Digital Surface Model (DSM) is also processed by a UNet. Our approach achieves 81.7 % Intersection over Union and has only one eighth of the number of free parameters of the state of the art approach [2].

Acknowledgements

Table of Contents

List of Figures

List of Tables

Abbreviations

CNN	Convolutional Neural Network
DEM	Digital Elevation Model
DSM	Digital Surface Model
$FCRF$	Fully Connected Conditonal Random Field
FCN	Fully Convolutional Neural Network
GPU	Graphics Processing Unit
GSD	Ground Sampling Distance
IoU	Intersection over Union
$LiDAR$	Light Detection and Ranging
$NDVI$	Normalized Difference Vegetation Index
OSM	Open Street Map
$ReLU$	Rectified Linear Unit
SGD	Stochastic Gradient Descent
VHR	Very High Resolution

1 | Introduction

Nowadays, large amounts of high resolution satelite imagery is available, offering a huge potential to extract semantic meaning from them. One of the most challenging and important tasks in the analysis of remote sensing imagery is the acurate identification of building rooftops. Several remote sensing applications make use of this information, among them are urban planning and reconstruction, disaster monitoring, 3D city modeling, etc. Although it is possible to manually delineate the buildings footprints, this is very time consuming and becomes infeasible when trying to cover larger areas, which also change over time. Therefore, the developement of algorithms for automatic building detection is an active research area.

The most common ways to extract buildings identify edges and other primitives in spectral images [8]. Also, the improvement of building polygons was investigated by [6]. Even with very accurate polygons, flat objects can have similar shapes and can therefore not be distinguished from buildings. Therefore, the upcoming of depth images led to new approaches in building extraction by offering to use height patterns [3]. Since depth images, like DSMs do not include all characteristics of buildings, such as their spectral appearance, later work was done to fuse the information from either LiDAR or stereo DSMs and spectral images [4], [18], [20]. This approach relies on giving accurate parameters to the algorithms, such as the minimum building height and the maximum NDVI of a building.

Since the latest ascent of deep learning methods, methods which are relying on prior models are not state of the art for wide fields anymore. Rather, there is a trend in many remote sensing applications to solve larger parts of the tasks at hand with learning based approaches, deep learning in particular. Deep learning was only used for a narrow range of applications such as document recognition in LeCun et al. 1998 until Krizhevsky et al. 2012 made a groundbreaking attempt on using deep convolutional neural networks for image classification. [10] show that CNNs can be trained on huge databases using efficient GPU implementation of the convolution operation by parallelising it. Since then, many new architectures have emerged while pushing the state of the art in image classification even further, such as [19] and [7].

Two reasons why CNNs beat comparing methods by a huge margin are (1) that they do not rely on manual feature extraction and (2) that they are translationally equivariant, meaning that features are extracted independently from their location in an image.

Advances in re-purposing CNNs for semantic segmentation make it possible to densely classify images [12]. In this work, we investigate the suitedness of different deep learning-based approaches to the fusion of DSM and spectral images. We use the fused data to classify every input pixel as building or non-building.

The remainder of this work first gives an overview of different approaches for buildings footprint extraction in chapter 2. In chapter 3, the proposed deep learning method is explained. To show the effectiveness and efficiency of our approach, we give insight in the carried out experiments in chapter 4 and present the results and a brief discussion of them in chapter 5.

2 | Related Work

A lot of research effort has gone into developing algorithms for building footprint extraction. There are two main classes of approaches, in which these algorithtms can be divided. We first give a brief introduction to approaches which are not based on deep learning and then we discuss some methodologies developed for semantic segmentation tasks in the field of remote sensing that are based on deep learning.

2.1 Feature Engineering based Approaches

Classical methods derive geometrical models from the analysis of building properties by human experts. In Huertas and Nevatia 1988 , the authors assume that buildings are characterized by rectangular shapes and combinations of them, as well as the existence of shadows, which can be used to distinguish non-building from building outlines. This approach yields building polygons, which often consist of jagged lines. To obtain more realistic footprint boundaries, Guercke and Sester 2011 use the Hough-Transformation to detect lines in a predetermined building footprint, then use lines corresponding to peaks in Hough space to construct a refined polygon.

Furthermore, to extract buildings more accurate and robust against varia-

tions in building appearance, datasets with both depth images and spectral images where used. Rottensteiner et al. **(year?)** use Dempster-Shafer theory to fuse multiple features obtained from LiDAR DSM and multispectral aerial imagery for building detection. These features include the normalised difference of the near-infrared and the red band (NDVI), the rise of objects from the ground (nDSM) and measures for the roughness of objects. From the rise, objects which are assumed to be lower than buildings are detected. The roughness measure and the NDVI mainly seperate trees from buildings.

Ekhtari et al. 2009 also use a LiDAR nDSM and refine the boundaries of the resulting building mask, using a WorldView image. First, an initial building mask is generated from the nDSM and is than reduced to its rough edges. Next, edges are detected in the spectral image, which are not limited to building outlines and are often discontinous. To filter out non-building edges, the edges from the spectral image are masked by the edges from the depth based building edges. Finally, to eliminate the discontinouities, polygons are fitted to the masked edges.

Turlapaty et al. 2012 compute the depth information by fusing spaceborne multi-angular imagery. They also use a multispectral image and a PAN image, which are fused by pansharpening. Afterwards, the NDVI is calculated from the pansharpened image. Statistical properties of these data sources are fed to a support vector machine, which classifies each pixel as building or non-building.

Although they work for certain areas and building appearances, a big drawback of hand-crafted methods is that the models are not applicable for many complex building structures. Furthermore, they rely on identifying relevant features by human interactors.

2.2 Deep Learning based Approaches

In contrast, deep learning methods pass the task of extracting features through a neural network model going through a data-based optimization process. Based on FCNs, Marmanis et al. 2016 apply an ensemble of networks, where each network is pretrained on different large databases of media images, and finetuned on remote sensing images of ground sampling distance 0.1 m. The class probabilities of their multi-class semantic segmentation task are then averaged to obtain the final output probabilities. They empirically show, that for remote sensing imagery, the models trained for computer vision tasks generalize well. Fixing the weights of the lower layers during early training and later making them learnable brought main improvements of computational costs as the loss doesn't need to be backpropagated through the lower layers. The authors not only use as input the spectral images but also a DEM which contains height information of vegetation and construction. The choice for DEM over nDSM (i.e. nDSM = DSM - DTM) reduced pre-processing.

Maggiori et al. 2017 , improve their results by multi-scale processing and finetuning, by applying FCNs to remote sensing imagery. Multi-scale processing captures the contextual information, due to a large receptive field, as well as the localization of the extracted features. To obtain this, they both downsample the input image to $\frac{1}{4}$ of the input resolution in one convolutional branch, while keeping the input resolution intact in the other convolutional branch. Then, after upsampling the downsampled branch to input resolution, both branches are added and an activation function is applied elementwise. The trade-off here is to use less convolutional layers in the full-resolution branch to reduce the number of parameters. Further, Maggiori et al. 2017 divide the training process into two stages. First, the model is trained on inexact OSM ground truth. Second, a refining stage is applied on the training

dataset with hand labeled ground truth. This improves their result by 50% IoU compared to the experiment before the refining stage. In this work, no hand labeled data is used for training, since the ground truth is already of high quality.

Bittner et al. 2017 extract building footprints using nDSM as data source. Their results show, that using height information is valuable for that task. FCRFs are used as a post-processing step to enhance local context of the prediction maps, which improves fine details in the building mask. Compared to Bittner et al. 2017 , we do not use FCRF, because it would make tuning of extra hyperparameters necessary and only yields small improvements.

The FCN8s architecture works well in computer vision, where image features are usually large and well seperated. But building footprints in remote sensing imagery can be of complex structure and dramatically vary in geometrical and spectral appearence. To tackle this issue, Bittner et al. 2018 adapt the FCN8s architecture by inserting an extra skip connection and makingthe final upsampling factor equal to 4. This architecture is then evaluated on VHR remote sensing imagery with ground sampling distance 0.5 m for the task of building footprint mask generation. Furthermore, in this work, the effect of using multiple data sources was studied based on the observations of Marmanis et al. 2016 . The best result is showen when pansharpened RGB, PAN and nDSM images are trained as three FCN4s networks. The three networks are concancatenated and three convolutional layers are applied at the end. This makes the network learn a joint information from each datasource. Despite the high quality results, using nDSMs and pansharpened RGB is not optimal because it requires preprocessing. Also, the FCN architecture has a comparatively huge amount of learnable parameters. This leads to a small batch size and hampers training speed and inference speed. As the spectral data shares many common features, using two seperate network branches for them is redundant.

Iglovikov and Shvets 2018 have successfully adapted the UNet architecture to remote sensing imagery. The difference of this approach to FCN4s and FCN8s is that it uses more skip connections. Image resolution is recovered at the last skip connection. Although Iglovikov and Shvets 2018 achieved high quality results, they do not utilize enough of the available data sources, which leaves potential for improvement.

3 | Methodology

For image classification, CNNs are the state of the art. If a neural network has layers with learned convolutional filters, it is called a CNN. In contrast to fully-connected layers, in convolutional layers the neurons of layer l are only connected to the neurons of a spatial window W of layer $l-1$. Their output is a vector of weighted sums, where for each component of the vector, the weights are different. For each component,

$$c^l = \sum_{i \in W} w_i c_i^{l-1} + b_i, \tag{3.1}$$

where the weights w_i and bias b_i of each component are equal across a layer. This reduces the number of parameters and introduces translational equivariance Goodfellow et al. 2016 . Since convolution is a linear operator, nonlinearities cannot be modeled by them. To introduce nonlinearity, an activation function is applied. In CNNs, ReLU is the standard choice for the activation function, which follows convolutional layers. ReLU works like a gate, which lets only positive values unchanged but sets all negative values to zero [15]. During training of the FCN architectures, dropout layers set units in the first two convolutionalized fully-connected layers to zero with probability of 0.5 for each unit. Dropout forces the parameters to a region in the parameterspace, where units can function independently. This decorrelates

the units and thus, decreases overfitting. To increase their receptive field, CNNs often use maximum pooling layers. This way, a larger spatial context can be encoded. The last layers of a CNN are usually fully connected to convert the feature maps into scores, which are fed to the softmax function to obtain class probabilities.

The key idea in deep learning is to stack multiple layers on top of each other, which enables the model to learn more abstract features. In lower layers it is easy to interpret the trained convolution kernels, because they usually correspond to simple geometrical abstractions like edges and corners. As information flows deeper inside the network, it becomes harder to interpret the meaning of the features. Because the abstraction increases in each layer, it is important to choose a model depth which suits the complexity of a specific problem. If the network is to shallow, the necessary abstraction cannot be achieved. In contrast, if the network is to deep, the network would need to learn the identity function to avoid exaggerated abstraction, which is hard for many networks [7].

Using back-propagation, the gradient $\nabla_w L$ of a loss L with respect to the weights w and the bias b, to which we will refer to as gradient, is computed. The gradient is usually evaluated for mini-batches instead of the whole training dataset (a) to get more frequent weight updates and (b) for efficiency. The magnitude of the gradient is not definitely connected to the localization of a minimum, which is why an empirically determined learning rate α is used to rescale the parameter update. Weight decay prevents the parameters from becoming excessively large. This leads to solutions with more balanced parameters which increase the effective capacity of the model. It is a good praxis to perform weight decay as a regularization technique, where a norm $||w||$ of the parameters is added to the loss function L for a binary classification task

$$L(p) = -(ylog(p) + (1-y)log(1-p)), \tag{3.2}$$

where y and p refer to the true label of an input x and the conditional probability $P(y = 1|x)$. The conditional probability p is approximated by the softmax function $\sigma(x)$, which is computed by

$$\sigma(x)_j = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}} \tag{3.3}$$

for j = 1,...,K, where $\sigma(x)_j$ and x_j refer to the j-th element of the softmax and the input vector respectively. Other common loss functions are the sigmoid function and the Euclidian distance. When using the sigmoid function, the gradient may saturate and thus, slow down optimization.

To rescale the weight decay, it is multiplied by a hyperparameter η, such that

$$L(p) = -(ylog(p) + (1-y_i)log(1-p)) + \eta||w||. \tag{3.4}$$

Additionally, to avoid the training algorithm to oscilate around local optimal solutions, momentum can be introduced. Let $g^{(i)}$ be the gradient of the loss function with respect to the parameters at iteration i and μ be the momentum hyperparameter, then the parameter update $\Delta w^{(i)}$ at iteration i is computed by

$$\Delta w^{(i)} = (1-\mu)\alpha g^{(i)} + \mu \Delta w^{(i-1)}. \tag{3.5}$$

3.1 FCNs

Fully connected layers drop the localization of the features, which is of high importance for semantic segmentation. Long et al. 2015 have pointed out, that fully connected layers are equivalent to convolutions with kernel size equal to the image size. Applying this kernel over image borders is the same as to stride the kernel. They use this fact to convolutionalize existing image classification networks, which makes them suitable for arbitrary sized input images and allows extending these architectures for effective pixelwise classification. For semantic segmentation, the extracted feature maps can be upsampled to the input dimension. The authors experimentally show, that utilizing a decoder module which gradually increases image dimensions and concatenates feature maps of corresponding scale improves over upsampling the last feature map of the encoder. They propose an architecture called FCN8s, which, as final step, upsamples the feature maps by a factor of eight to get to original image size. In particular, gradually upsampling the feature map improves the localization of features. The VGG16 architecture proved to be a reasonable choice as the encoder of the FCN architecture for various remote sensing data sources Bittner et al. 2018 .

One important aspect of FCNs is their output stride s, i.e. $s = \frac{resultion_{input}}{resolution_{output}}$. Decreasing s, the number of parameters rises and extra skip connections from lower layers with higher resolution could be introduced, which is why potentially more high frequency information is fed to the predicted masks. This refines the boundaries in the class probability maps.

3.2 Transposed Convolution

The resulting feature maps from a convolutionalized CNN have been downsampled by strided convolutions and/or pooling. To increase the resolution, transposed convolution is used. Transposed convolution works with kernels, which are applied to each pixel in the input by multiplying each kernel element by the value of the pixel. The stride s determines by how far the outputs of two neighboring inputs are shifted relative to each other. When it is lower than the kernel size, outputs at neighboring pixels overlap. The stride also determines the output resolution. The kernel weights are learnable, which distinguishes transposed convolution from upsampling by interpolation with fixed weights.

3.3 Pansharpening with CNNs

To find new ways to utilize the high spectral resolution of the multi-spectral and the high spatial resolution of the PAN image, we take a look at recent successfull approaches to the pansharpening problem. Rao et al. 2017 apply three convolutional layers to the PAN image. The number of output channels of the third convolution is three. It is added to the downsampled multi-spectral image and then fed to the euclidean loss. Yang et al. 2017 upsample the low-resolution multi-spectral image by the four and concatenate it with the pan image before feeding the concatenated channels to ResNet.

To reduce the amount of preprocessing, we (a) use DSM instead of normalized DSM and (b) combine the PAN and the multispectral image automatically in an end-to-end fashion similar to Rao et al. 2017 . We also use eight multispectral bands instead of only rgb, because we want our model to learn from a broader range of the electromagnetic spectrum with higher spectral

resolution. Furthermore, we will adapt UNet architecture because the extra skip connections provide more detailed local information to the final building footprint mask. For the encoder part of the UNet we will use (a) the first five layers of VGG16 to be able to use the ImageNet-pretrained weights and (b) one additional layer to learn the task specific features.

3.4 Fusing DSM, PAN and Multispectral Images

In pansharpening, the aim is to generate a multispectral image with high spatial resolution. The deep learning approach to this problem is learning a transformation from a low-resolution multispectral image and a high-resolution PAN image to a high-resolution multispectral image. This requires processing them in a common stream. We fuse the PAN and multispectral branches of our network at an early stage, by applying transposed convolution to the multispectral image to upsample it by four. The PAN image is fed to a shallow three-layer CNN to get eight feature maps of the PAN image. The output of the transposed convolution and the CNN are added. This approach is very similar to Rao et al. 2017 . They use the of the interpolated multispectral and the PAN, because only the residuals, which are sparse, are left to learn for the network. This potentially makes learning the fusion of the spectral data easier. We will compare this strategy to upsampling the multispectral image by four and concatenating it with the PAN image as proposed by Yang et al. 2017 , the late fusion approach of Bittner et al. 2018 , but with a low-resolution RGB, which is upsampled by a transposed convolution and the strategy of Bittner et al. 2018 . Similar to Bittner et al. 2018 , we have one stream each for DSM and spectral data and fuse them using a three layer network which automatically learns the recognize the individual contributions of spectral and depth information for extracting

the buildings. Each of the two streams is a UNet.

Figure 3.1 visualizes the four compared ways of fusing the spectral images. FCN_PAN and FCN_MS respectively represent a fully convolutional network for PAN and multispectral images.

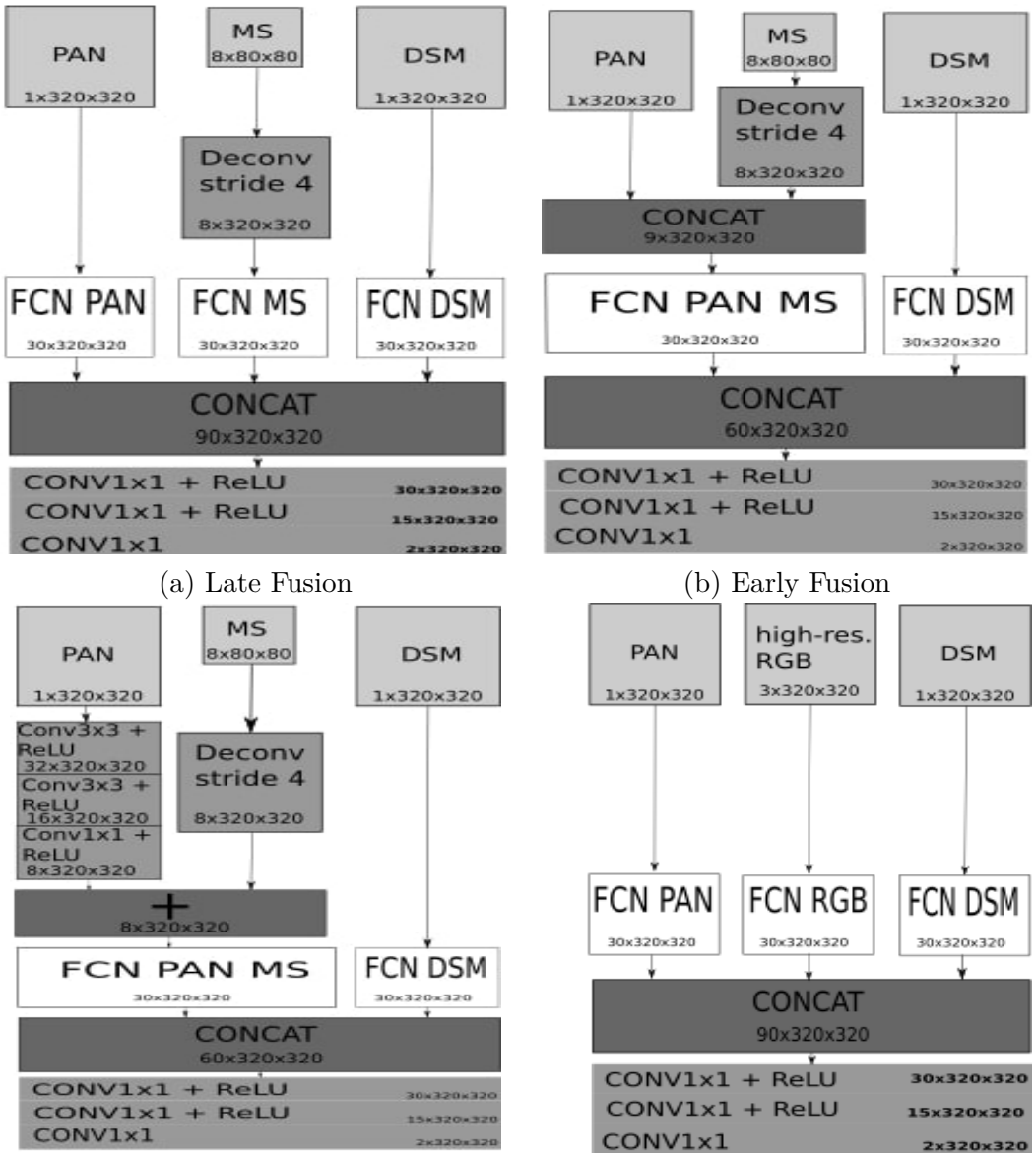
3.5 UNet

Ronneberger et al. 2015 proposed the UNet archtitecture, which shows state of the art results on biomedical semantic segmentation. As the FCN4s and FCN8s, it consists of an encoder and a decoder module. The encoder uses max pooling to utilize multi-scale context, whereas the decoder uses transposed convolutions and skip connections. which consist of cropping the feature map of the desired resolution and concatenating it with the output of the transposed convolution, followed by convolutional layers. In contrast to FCNs with less skip connections, UNet recovers image resolution at the last skip connection.

This concept is generic because the encoder can be any FCN. As VGG16 has shown good results in Bittner et al. 2017 , Marmanis et al. 2016 and Bittner et al. 2018 , we use it's first five layers and then put a sixth layer with 512 3x3-kernels on top of it to learn the features specific for building footprint extraction. Compared to VGG16, this approach dramatically decreases the number of parameters in the network.

3.6 Transfer Learning

For many architectures, there exist models, which have been trained on huge databases for many iterations. To reduce computational cost and excessive



(a) Late Fusion (b) Early Fusion

(c) Pansharpening Fusion (d) FCN4s fused

Figure 3.1: Four different fusion strategies.

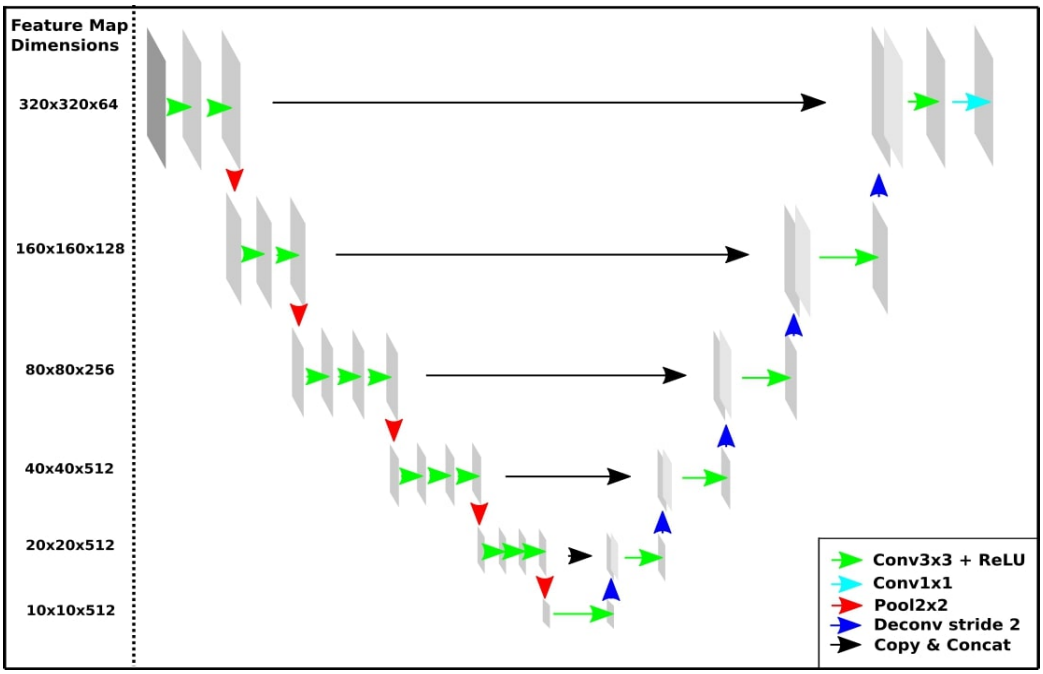


Figure 3.2: The proposed adapted UNet architecture.

hyperparameter tuning, one can use these models and fine tune them for the task at hand with comparably few iterations. ImageNet pretrained models have successfully been finetuned for semantic segmentation of remote sensing imagery by Marmanis et al. 2016 . Building on these results, we will use the weights of the first five layers of the VGG16 network pretrained on the ImageNet database only for our multispectral and PAN data, as the features learned by the pretrained model are specific for spectral data and are not exhaustively describing the 3D information in DSM data.

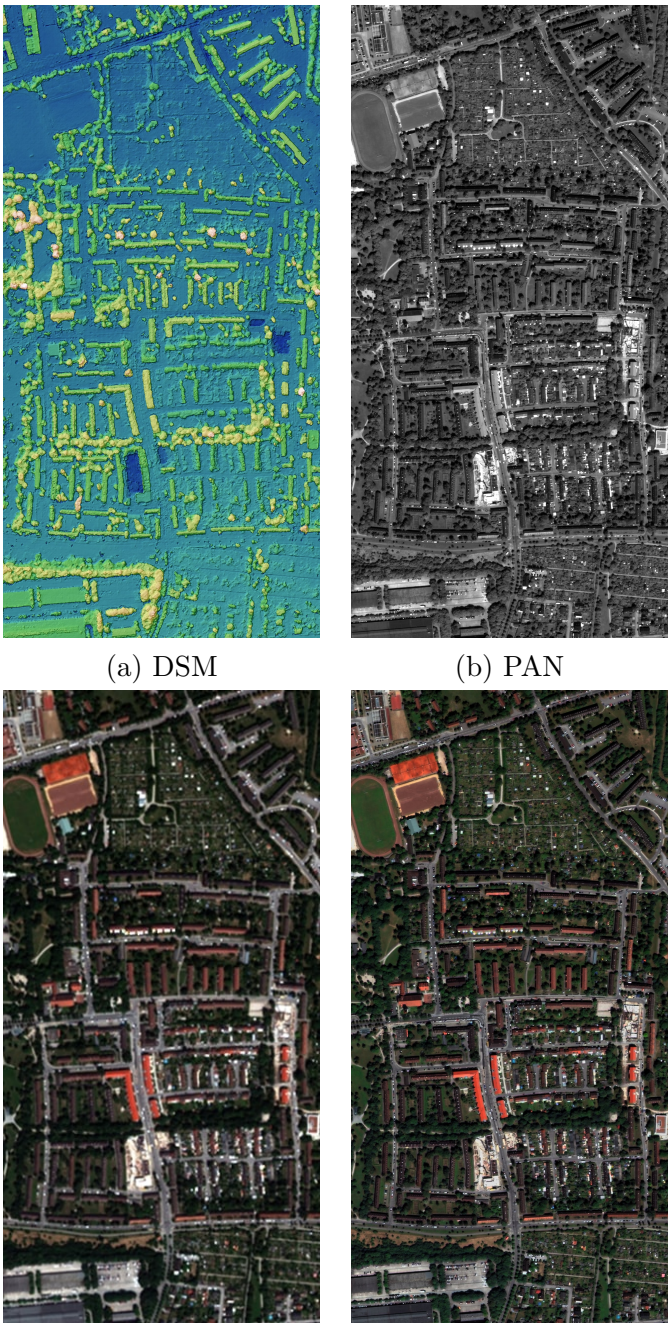
3.7 Network Architecture

Our final architecture consists of three stages. We (1) fuse the PAN and multispectral images using transposed convolution and a three layer network,

(2) feed the fused spectral features and the DSM image to parallel branches of our proposed UNet and (3) concatenate both branches by appyling a three layer network to find which features suit the ground truth more.

4 | Study Area and Experiments

To validate our approach, we use WorldView-2 imagery of Munich, Germany. The training data consists ofDSM, PAN (both 0.5 m GSD) and MS images (8 channels, 2 m GSD) reorganized into a collection of 32500 patches with a size of 320x320 px and overlap 160x160 px, where 20% are kept back for validation. A 1280x2560 px area, which doesn't overlap with the training data, is used for testing. The satellite images are orthorectified, because we want to obtain building footprints that appear as if they are view from nadir. In order to show the generalization capability of our model, we include small parts from WorldView-2 imagery of urban areas of Tunis, Tunisia. To compensate for the missing ground truth in this area, we use building footprints from OSM. However, there are only few areas, which are densely covered by OSM building footprint data. The test areas are aquired by selecting rectangles where as well high quality DSM data and OSM building footprints are available.

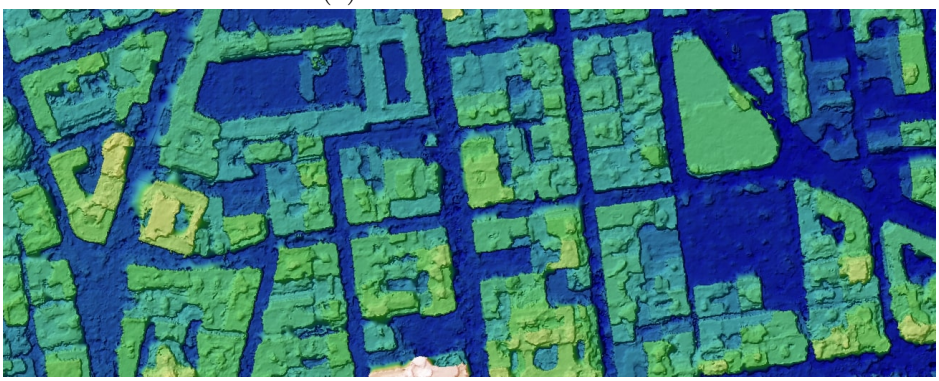


(a) DSM (b) PAN

(c) RGB (d) pansharpened RGB

Figure 4.1: Test area in Munich, Germany. DSM image is color-shaded for better visualization.

(a) low-resolution RGB



(b) DSM

(c) PAN

Figure 4.2: Patch of the test area in Tunis, Tunisia used for visual inspection. DSM image is color-shaded for better visualization.

4.1 Image Preprocessing

For artificial neural networks, it is important to have inputs of equal mean and scale. Therefore, we subtract the mean of the whole training data from each patch during training and while testing, we subtract the mean of the test area. Then we rescale our data to the range $[-1, 1]$, because it prevents the neural network model to learn different ranges and increases the resolution of the numbers, that are broadcast to the network, compared to the interval $[0, 1]$, assuming the same type of machine numbers. The raw DSM data has many outliers, which are caused by the generation process. These outliers badly influence the minimum and maximum of our data, which are important. Also, since outliers affect the mean, they push the true values to a tight region of the range, which decreases their distuingishability. Therefore, we roughly extract the lowest true value from the histogram and set all values below to this value. The high outliers are far less then the low ones, so we did not change them.

4.2 Implementation and Training Details

Building on the code developed by Bittner et al. 2018 , we implemented our UNet network on top of the *Caffe* deep learning framework. For training, we use SGD with momentum and weight decay. An epoch consists of iteratively feeding mini-batches to the network, computing the gradients and updating the weights, until each patch in the training data has been processed once by the network. Depending on the batch size, the number of iterations in one epoch varies. Due to the memory limit of 12 GB on the used NVDIDIA TITAN X (Pascal) GPU, batch size is limited and was chosen as big as possible for each network. The learning rate is multiplied by 0.1 every 4 epochs. All training hyperparameters are empirically chosen. We use early

FCN4s				UNET
FUSED	LATE	EARLY CONCAT	PSFUSION	
10	10	8	9	8

Table 4.1: Number of epochs used to train different architectures.

stopping and give the number of epochs for every trained architecture in 4.1.

Training with patches of side length 320 px, boundary effects can arise. Therefore, our patches overlap by 160 px in each dimension. During inference, the network output is averaged in the overlapping regions. As pointed out in 3.6, we use pretrained weights for the spectral branches and train the DSM branch from scratch. To balance the training progress, we increase the learning rate in the layers 1-5 in the DSM branch by factor 10. We initialize all weights of convolutional layers, which are not filled with pretrained weigths by uniformly sampled random numbers from the range $\left[-\frac{1}{N}, \frac{1}{N}\right]$, where N is the number of neurons for that layer. The transposed convolution layers are initialized by bilinear weights, that is, in the beginning of the training, these layers perform bilinear upsampling to their input.

4.3 Comparison with Alternative Methods

To compare the network developed in Section 3.7 to other architectures by means of how well it makes use of the available data and for efficiency, we first train and test the FCN4s fused as in Bittner et al. 2018 with pansharpened RGB, PAN and DSM data. Because of the computational burden of generating nDSMs for large areas, we train it with DSM instead of nDSM. In order to demonstrate that FCNs can integrate spectral and geometric information independently, we directly deploy the FCN4s fused on low-resolution RGB, PAN and DSM data. Since multispectral information is not limited to the RGB channels, we use five more of the available spectral bands, cov-

ering the range from 400 nm (Coastal) to 1040 nm (Near-IR2), and show that this increases the building prediction performance of the FCN4s compared to using only RGB. Furthermore, we compare different approaches on fusing spectral and depth information, as described in Section 3.4, to show that pansharpening fusion can be used alternatively to late fusion and give better results than early concatenation. Finally, we demonstrate that using UNet instead of FCN4s results in straighter building outlines, higher scores on several metrics and reduces the number of parameters in our network significantly.

5 | Results and Discussion

To evaluate our approach, we test different architectures in three stages. First, we compare the building footprints generated by different models by their appearance. Then, for every model we use several metrics to evaluate them. Last, we test our approach on an entirely new area, to examine its generalization capacity.

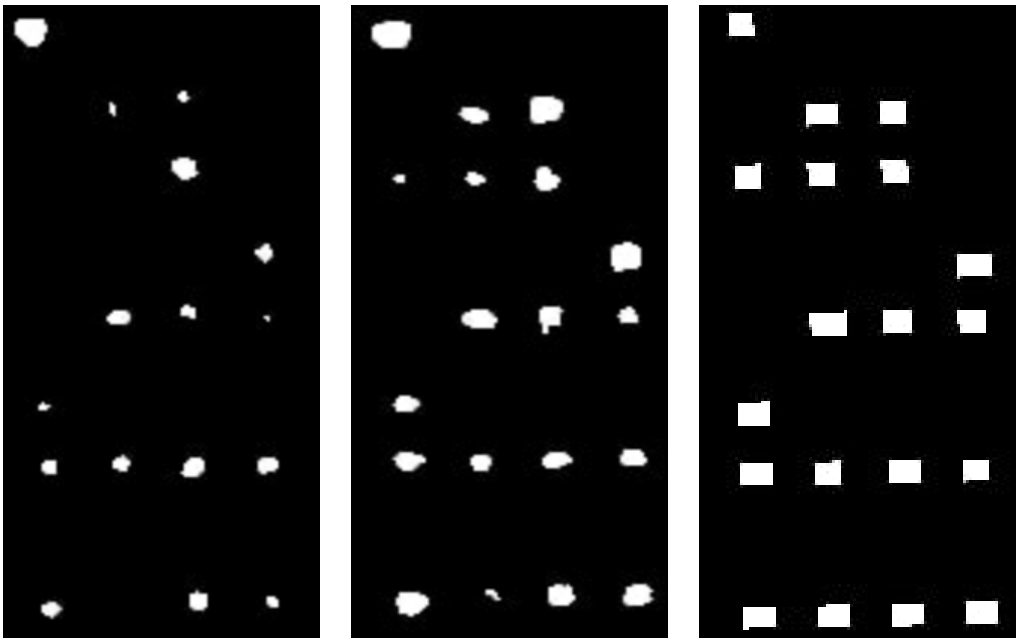
5.1 Qualitative Evaluation

5.1.1 FCN4s with Low Resolution RGB

The preprocessing used to obtain a high-resolution RGB image is computationally intensive and we, therefore, decided to directly pass the low-resolution RGB image to a deconvolution layer, before feeding its output to the FCN4s. In Figure 5.1, both generated building footprints and the ground truth are compared. The low resolution approach results in too small building footprints and the outlines are smoother (see Figure 5.2). But it still provides a high quality mask and includes less preprocessing.

(a) Low-resolution RGB (b) High-resolution RGB (c) Ground truth

Figure 5.1: Generated masks of high vs. low resolution RGB based FCN4s.



(a) Low-res. RGB. (b) High-res. RGB. (c) Ground truth.

Figure 5.2: Detailed comparison of high vs. low resolution RGB based FCN4s. In (a), some small buildings are missing here or are smaller than in the ground truth.

5.1.2 FCN4s with Multispectral Image

(a) RGB (b) Multispectral (c) Ground truth

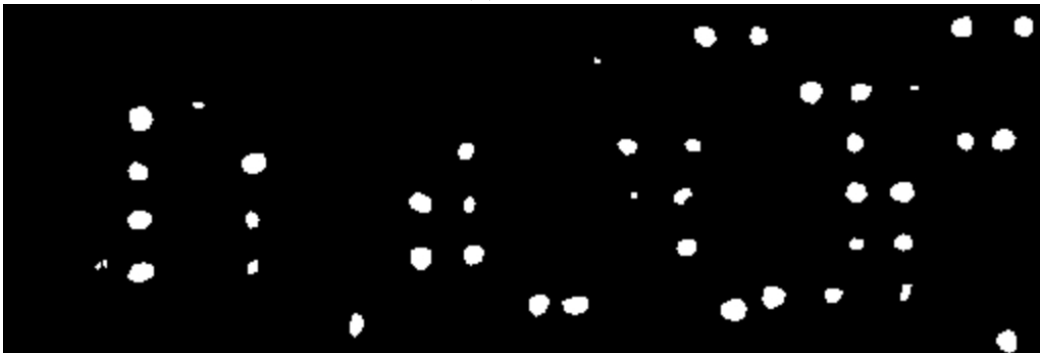
Figure 5.3: Generated masks of RGB vs. multispectral based FCN4s.

To make the network more powerfull, we replace the RGB image with an eight-channel multispectral image. In the middle of the upper part of Figures 5.3 a) and b), the multispectral based FCN4s detects larger parts of the small building footprints. Comparing both footprints as a whole with the ground truth, using multispectral information only slightly increases the number of parameters (see Table 5.2) of the overall network but yields more complete building footprints.

5.1.3 FCN4s with Pan-sharpening Fusion

We now focus on fusing our available data. See Figure 5.5 for reference on the resulting masks. Early concatenation leads to a less complete mask, which

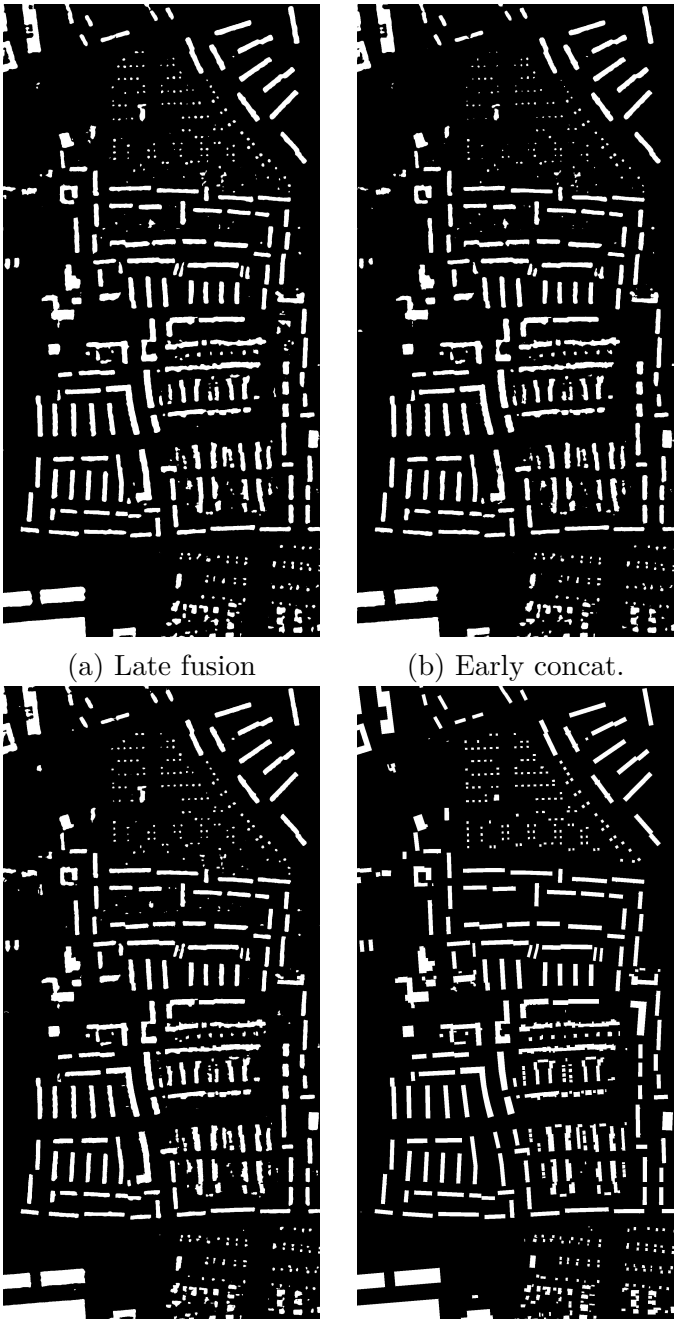
(a) RGB.



(b) MS.

(c) Ground truth.

Figure 5.4: Detailed comparison of multispectral vs. RGB based FCN4s. In (a), some small buildings are missing here or are smaller than in the ground truth. Result (b) is also incomplete, but includes much more information than (a).



(a) Late fusion (b) Early concat.

(c) Pan-sharpening fusion (d) Ground truth

Figure 5.5: Generated masks of three different fusion strategies.

can be seen in the area of small buildings in the top part (compare Figure 5.6. Here, both the late fusion and the pan-sharpening fusion give more complete results. In the masks generated by the FCN4s we see additional structures in the top left area, which are not present in the ground truth. This building has a glass roof built on a hash-like structure. As we can see in Figure 5.6, the information is present in our data. It is noteworthy, that with a far smaller number of parameters the FCN4s pan-sharpening fusion architecture performs very similar to the one with late fusion and also with the reference architecture from Bittner et al. 2018 which uses a high resolution RGB image.

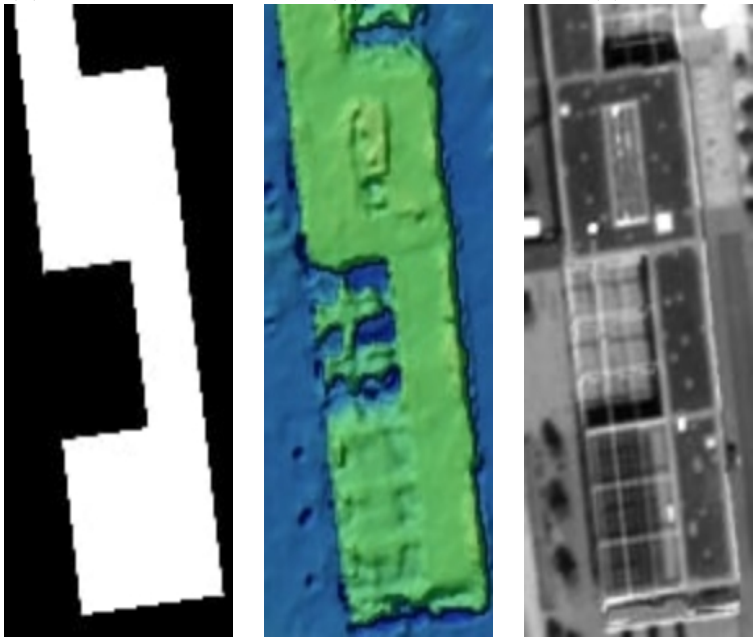
5.1.4 UNet

Since it is very important for building footprint masks to have as straight as possible outlines, we aim to utilize more of the high resolution content of our input images, by applying the UNet architecture to our data. As visualized in figure 5.7, most of the building outlines produced by the UNet are straighter then those produced by the FCN4s. Furthermore, the UNet's resulting building outlines are less bumpy (see Figure 5.8). In the middle part of the right side, there is a footprint in the ground truth, which is covered only very little by the FCN4s and with many gaps by the UNet. During gathering of the image signals, this building was under construction. There are small structures visible in the data, which are very similar to those in a finished building.

5.1.5 UNet vs. FCN4s fused

To show the improvements over the recent state of the art technique, we visualize the differences in the generated building masks of UNet with pan-

(a) Early concat.. (b) Late (c) p.-s.

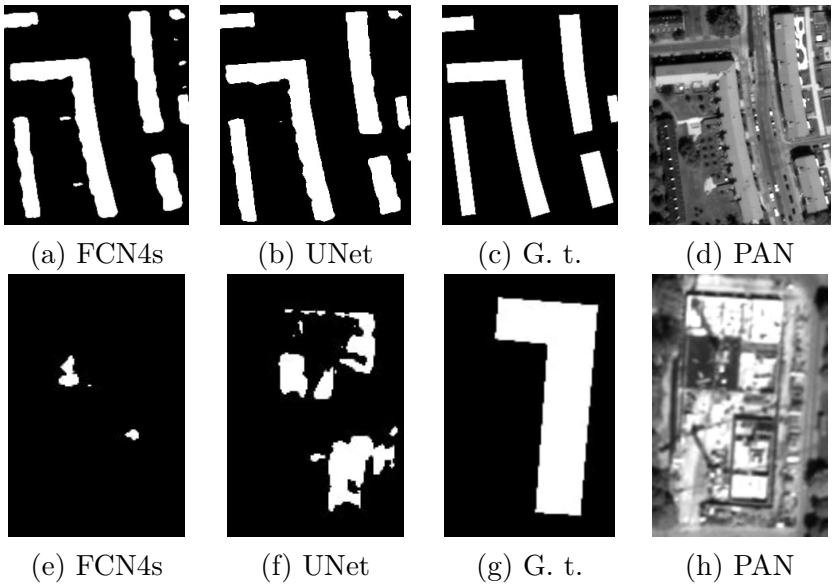


(d) G.t.. (e) DSM (f) PAN

Figure 5.6: Detailed comparison of different fusion strategies. DSM image is color-shaded for better visualization.

(a) FCN4s (b) UNet (c) Ground truth

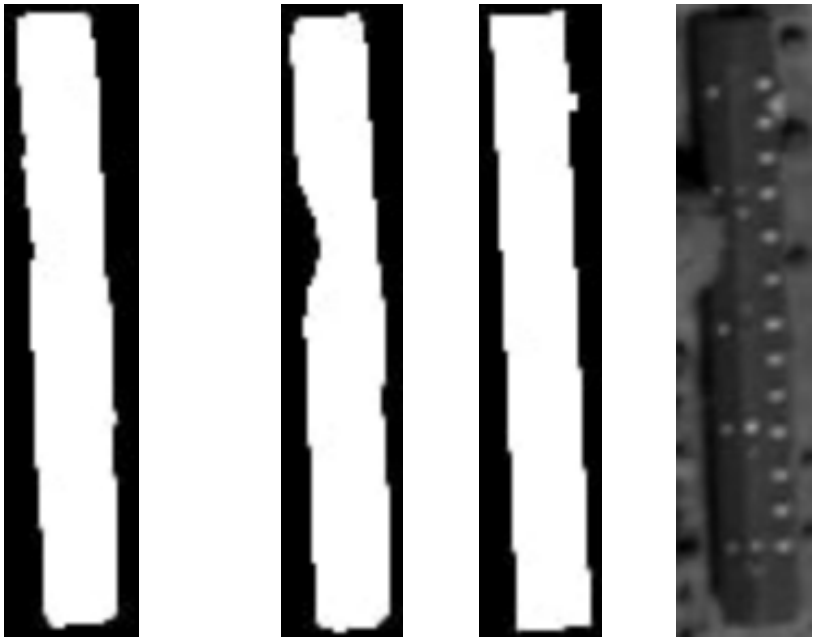
Figure 5.7: Generated masks of FCN4s vs. UNet.



(a) FCN4s (b) UNet (c) G. t. (d) PAN

(e) FCN4s (f) UNet (g) G. t. (h) PAN

Figure 5.8: Detailed comparison of FCN4s and UNet. In (a), (b), (c) and (d) we can see, that the boundaries of the UNet are slightly sharper than those of the FCN4s. In (e), (f), (g) and (h) the results in the area of a building under construction are illustrated. The UNet captured more details of an incomplete building.



(a) UNet p.-s. fusion (b) FCN4s fused (c) PAN (d) Ground truth

Figure 5.9: Detailed comparison of FCN4s fused and UNet with pan-sharpening fusion.

(a) UNet p.-s. fusion (b) FCN4s fused

(c) Ground truth

Figure 5.10: Detailed comparison of FCN4s fused and UNet with pan-sharpening fusion.

	RGB		Multispectral				Metrics		
FCN type	high res, fused	low res, fused	**low res, fused**	late fus	concat fus	**ps fus**	Acc.	IoU	F_1
FCN4s	x						**97.2**	80.3	**89.1**
FCN4s		x					96.8	77.2	87.1
FCN4s			x	x			96.9	78.5	88.0
FCN4s			x		x		96.8	77.5	87.3
FCN4s			x			x	**97.2**	80.1	**88.9**
UNet			**x**			**x**	**97.4**	**81.7**	**89.1**

Table 5.1: Quantitative results of the examined architectures, given in percent.

sharpening fusion and FCN4s fused, respectively. In Figure 5.9 we zoomed very closely on building footprints generated by the FCN4s fused and the UNet with pan-sharpening fusion and compare them to the groundtruth. The boundaries are much sharper in the results of the UNet with pan-sharpening fusion. In the PAN, we can see that there is a tree that covers some part of the building. The UNet reconstructs the boundary. Despite it does not have the information where the true outlines are from the data, it produces a rather rectangular shape. The reason might be, that the UNet has learned better than the FCN4s, that buildings often are rectangular. In Figure 5.10, we make the observation, that the UNet produces less bumpy outlines than the FCN4s, which are more similar to the ground truth.

5.2 Quantitative Evaluation

We use the Accuracy, which is common in semantic segmentation, as well as the IoU and the F_1-measure, which are suitable for binary classification tasks, to quantitatively evaluate the experiments. In the case of binary classification,

architecture	number parameters	time forward-pass
FCN4s fused	403.205.772	0.100553875923 s
FCN4s fused low-res. RGB	403.205.964	0.100716901302 s
FCN4s late low-res. MS	403.209.164	0.100480812907 s
FCN4s early concat.	268.807.592	**0.0667015542984 s**
FCN4s pan-sh. fusion	268.812.040	**0.0677197296619 s**
UNet pan-sh. fusion	**56.185.288**	**0.0735983946323 s**

Table 5.2: Evaluation results regarding the efficiency. Inference on a NVIDIA GeForce Titan X with Maxwell architecture.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}, \tag{5.1}$$

$$IoU = \frac{TP}{TP + FP + FN}, \tag{5.2}$$

$$F_1 = \frac{2TP}{2TP + FP + FN}, \tag{5.3}$$

where TP is the number of pixels that belong to building and are classified such, FP the number of pixels which do not belong to buildings but are classified as such, TN the number of pixels classified as non-building, that belong to non-buildings and FN the number of pixels classified as non-buildings but belong to building. The F_1-measure is the special case of the F_β-measure, where $\beta = 1$. It can be derived as the harmonic mean of the *precision* and *recall* metrics, where the *precision* is low if FP is high and vice-versa and *recall* is low if FN is high and vice-versa. Thus, the F_1-measure expresses, how correct and at the same time complete the predicted building footprints are. In our data, the number of building pixels is much smaller than non-building pixels. The IoU is the hardest of these three to obtain a high score from, because the number of building pixels is much smaller than non-building pixels and the IoU can only be high if TP is high.

We list the results of evaluated metrics of the FCN4s fused as in Bittner et al. 2018 , the FCN4s fused with low-resolution RGB and extra deconvolutional layer in the beginning, the FCN4s fused with multispectral data, the FCN4s with three different fusing strategies and the UNet in Table 5.1 and additionally the number of parameters as well as the inference speed in Table 5.2. From the statistics in Table 5.1, we recognize that the architectures which use a pan-sharpened RGB, or implicitly produce a pseudo pan-sharpened multispectral image, are significantly worse on IoU, but still let to comparable performance on the other two metrics. Since the used pretrained models are trained on sharp RGB images, it might be advantegous that these architectures feed the fused information from both spectral images to the model. Furthermore, we note that the UNet performs slightly better than the FCN4s. For the UNet, the lower number of layers and kernels in the 14th layer did not influence the performance but the extra skip connections resulted in small perfomance improvements. The obtained results are very accurate, which points out that (a) no post-processing is necessary and (b) DSM is a suitable substitute for nDSM.

5.3 Model Generalization Capability

To study the model's capacity to extract the key features distinguishing buildings from non-buildings, we employ it on data from Tunis, Tunisia. Different than in Munich data, the Tunis images contain more complex rooftop textures. Further challenges on this dataset are the very high grade of detail of the building outlines and the high variations in building density. The Tunis test data was directly passed to the system that was only trained on Munich data. In Figure 4.2 we can see, that the resulting building footprints are much more detailed than the ground truth obtained from OSM. Many details and even some complete buildings, that we can determine in Figure 4.2, are

missing in the OSM data, but are present in the predicted mask. By visual inspection, the model generalizes well on the Tunis images. It captures fine details and highly complex building structures and operates independently from building density. The statistical evaluation yielded an $Accuracy$ of 79.0 % an IoU of 46.4 % and an F_1 measure of 61.8 %. The huge dip in performance, about 35 % lower IoU, is due to the incomplete and partially incorrect ground truth obtained from OSM.

5.4 Discussion

First, the experiments carried out on the fusion strategy showed that early concatenation does not perform as good as the other two approaches. The approach implies concatenating one PAN channel with eight multispectral channels. Therefore, the amount of information proceeded to the next layer is strongly imbalanced, having only a small proportion of the PAN's high spatial resolution. Despite late fusion performs better than early concatentating, it does not take into account that PAN and multispectral images share many common features. E.g., a shape typical for a building might be recognized by the network by its geometrical appearance in both image signals. The pansharpening fusion avoids this redundancy and balances the proportions of information proceeded to the next layer by both images.

Furthermore, the applied preprocessing involved normalisation of the images had a huge influence on the results. During developement, tests with not equally scaled images as those in the training set showed poor results. Artifacts were introduced by the network, which are hard to interpret from the given images. This shows that the network learned scale specific. Also, it is very important to apply the same rescaling method to all data sources. Even though the network could learn to balance differently scaled data, this takes an extra effort, hampering the training process. Statistical issues also

showed in the data itself. The generation process of stereo DSM images can produce outliers, which influence the histogram. Ignoring the outliers leads to misbehavior during training and testing, because it affects the mean value. Subtracting a mean which is excessively high due to outliers, pushes the true values to a small range of numbers, whereas we want the true values to share the whole range of possible values.

Also, the performance of a model should not only be evaluated based on metrics, e.g. IoU, but also based on the number of free parameters, which are adapted by SGD steps. The larger this number, the longer the training of the network takes. This is due to the fact that a smaller model can use larger mini-batches, which makes the gradient estimations more accurate and increases the exploitation of the potential for parallelisation on a GPU. Also, a lower number of parameters corresponds to faster forward passes in training and testing.

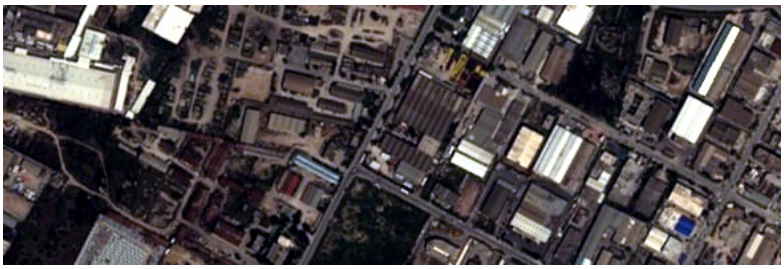
When comparing our results to those in other papers, it is important to be aware of the respective training dataset. In general, larger training datasets can cover a greater amount of buildings and variety of building appearances, allowing the network to produce scores with higher certainty and generalize better. Even if the amount of training data is high, bad quality of the ground truth can influence the results and causes uncertainty on incorrectly labeled features, or can make testing difficult, as seen in Section 5.3

During training, validation gives a hint on how the training process serves to improve the models performance on unseen data, which is important because overfitting can occur when training for to many iterations. On the other hand, if one does not train sufficiently long, the model might not adapt well enough to the training data and for this reason performs bad on the test data.

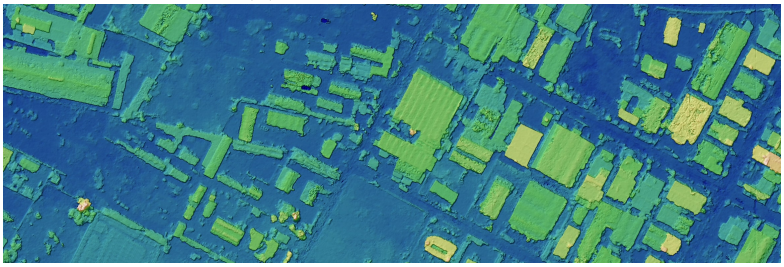
6 | Conclusion

We adapted UNet to VHR remote sensing imagery for the task of building footprint extraction and showed that it can provide building masks of high quality. Furthermore, we presented a method to fuse depth and spectral information based on CNNs. The used architecture provides an end-to-end framework for semantic segmentation, which performs well on the task of building footprint extraction from WorldView2 images. The trained system was tested on unseen urban areas in Munich, Germany and Tunis, Tunisia. It produces masks with sharper edges and has less parameters than the reference architecture. Furthermore, it works with DSM and low-resolution multispectral images. The performance of the proposed architecture does not depend on simple or reoccuring shapes, but segments complex and very small building structures accurately. Some of the remaining noise and incaccuracies in the generated building masks is often due to trees covering whole buildings, ongoing construction work or building structures of complexity, which are challenging even for humans to distinguish from non-buildings. Although the improvement in quality of our method is small, it still excels the performance of the reference architecture, while having far less parameters and higher inference speed. Therefore we believe, that the presented method has a great potential to efficiently exploit mixed datasets of remote sensing imagery for building footprint extraction.

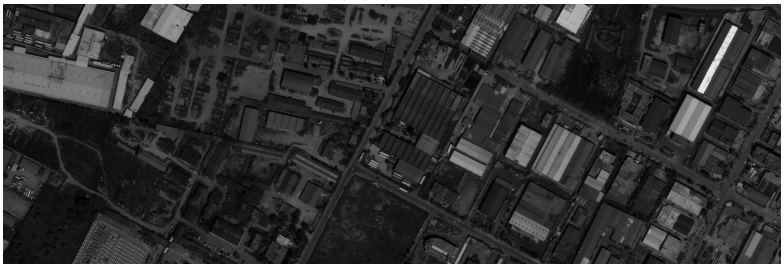
Appendix 1: Testarea Tunis



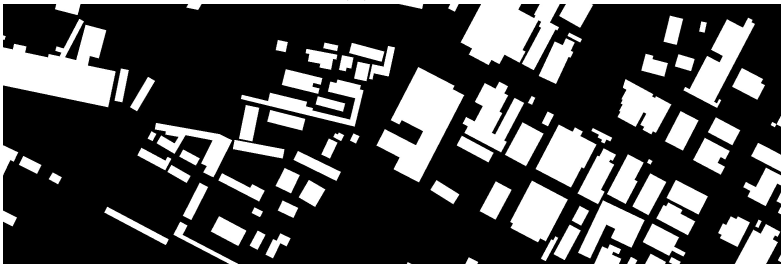
(a) low-resolution RGB



(b) DSM

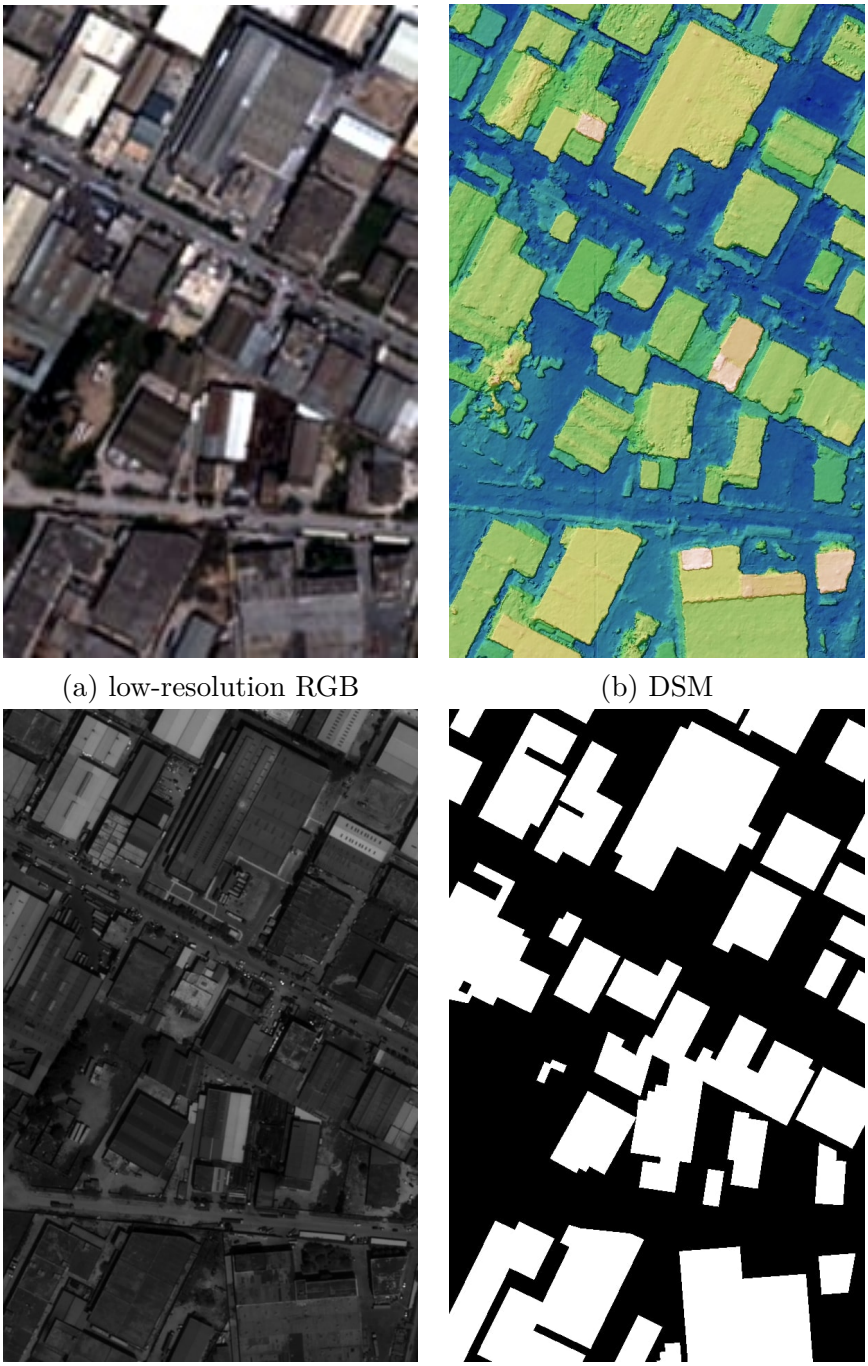


(c) PAN



(d) OSM building footprints

Figure 6.1: Patch 2 of Testarea in Tunis.

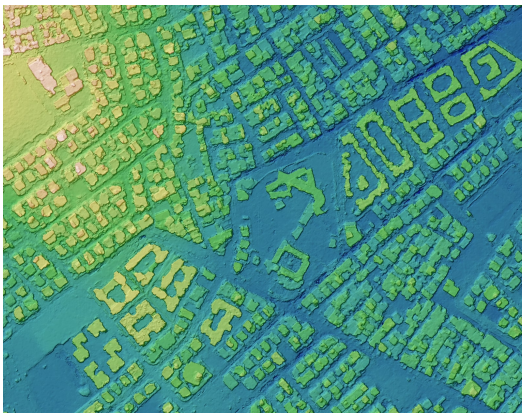


(a) low-resolution RGB (b) DSM

(c) PAN (d) OSM building footprints

Figure 6.2: Patch 3 of Testarea in Tunis.

(a) low-resolution RGB

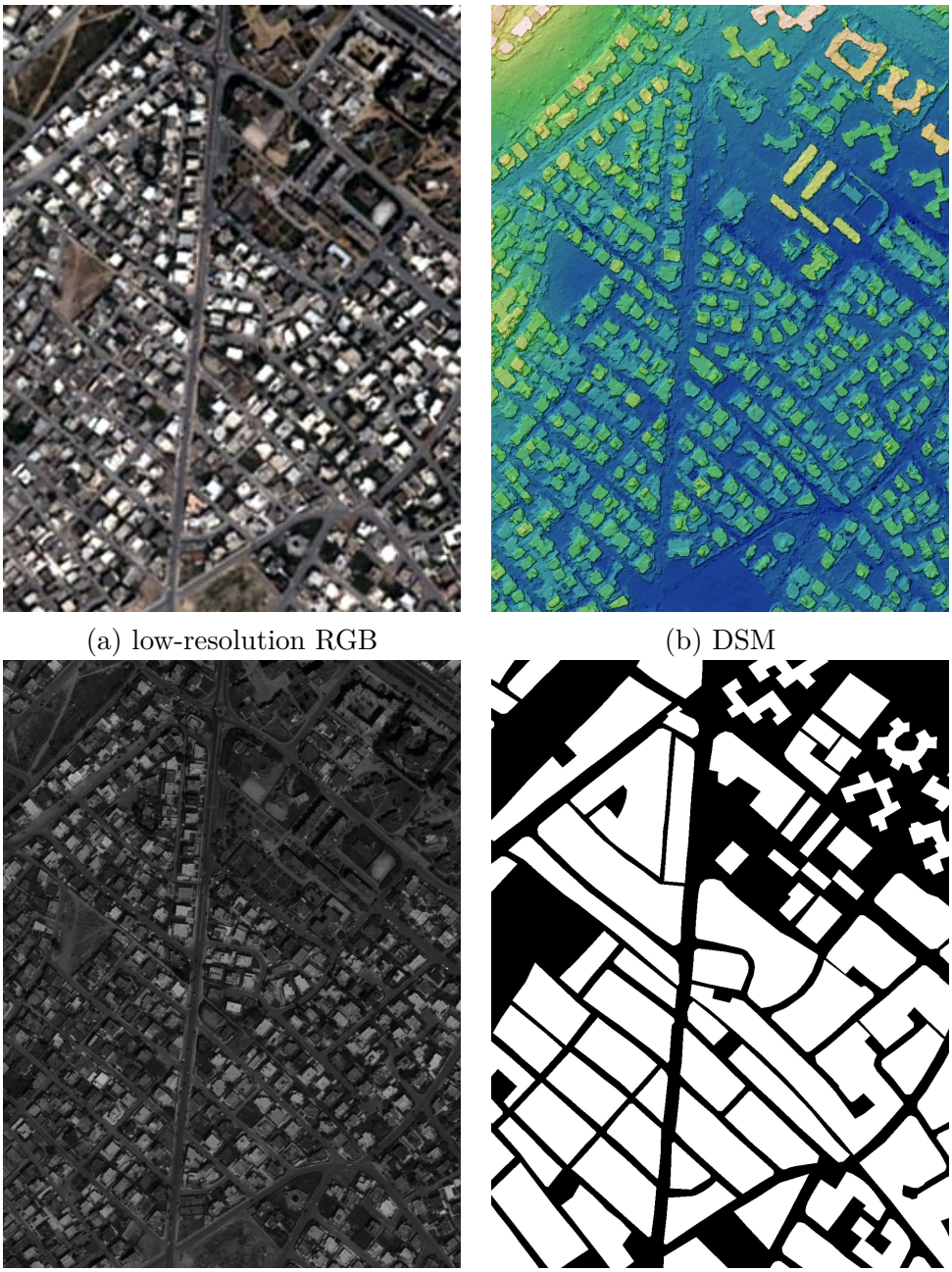


(b) DSM

(c) PAN

(d) OSM building footprints

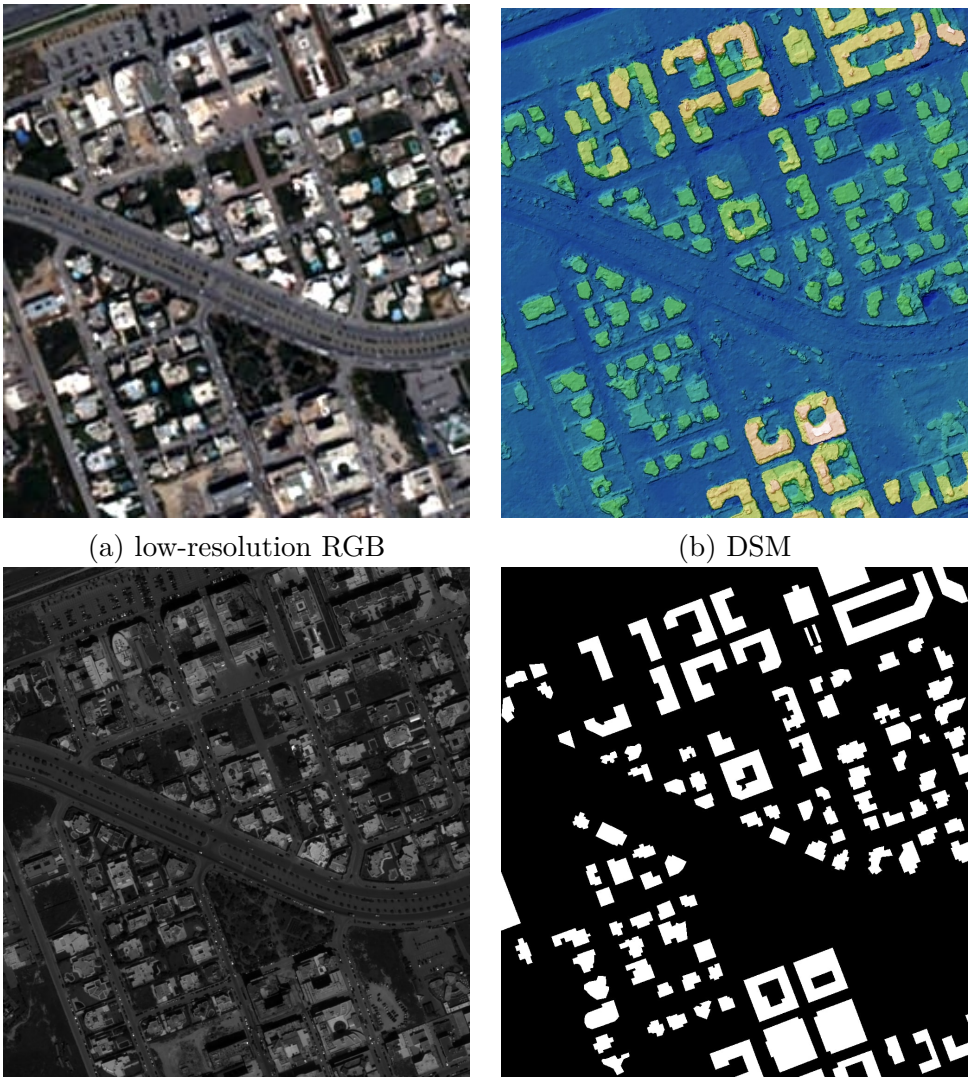
Figure 6.3: Patch 4 of Testarea in Tunis.



(a) low-resolution RGB (b) DSM

(c) PAN (d) OSM building footprints

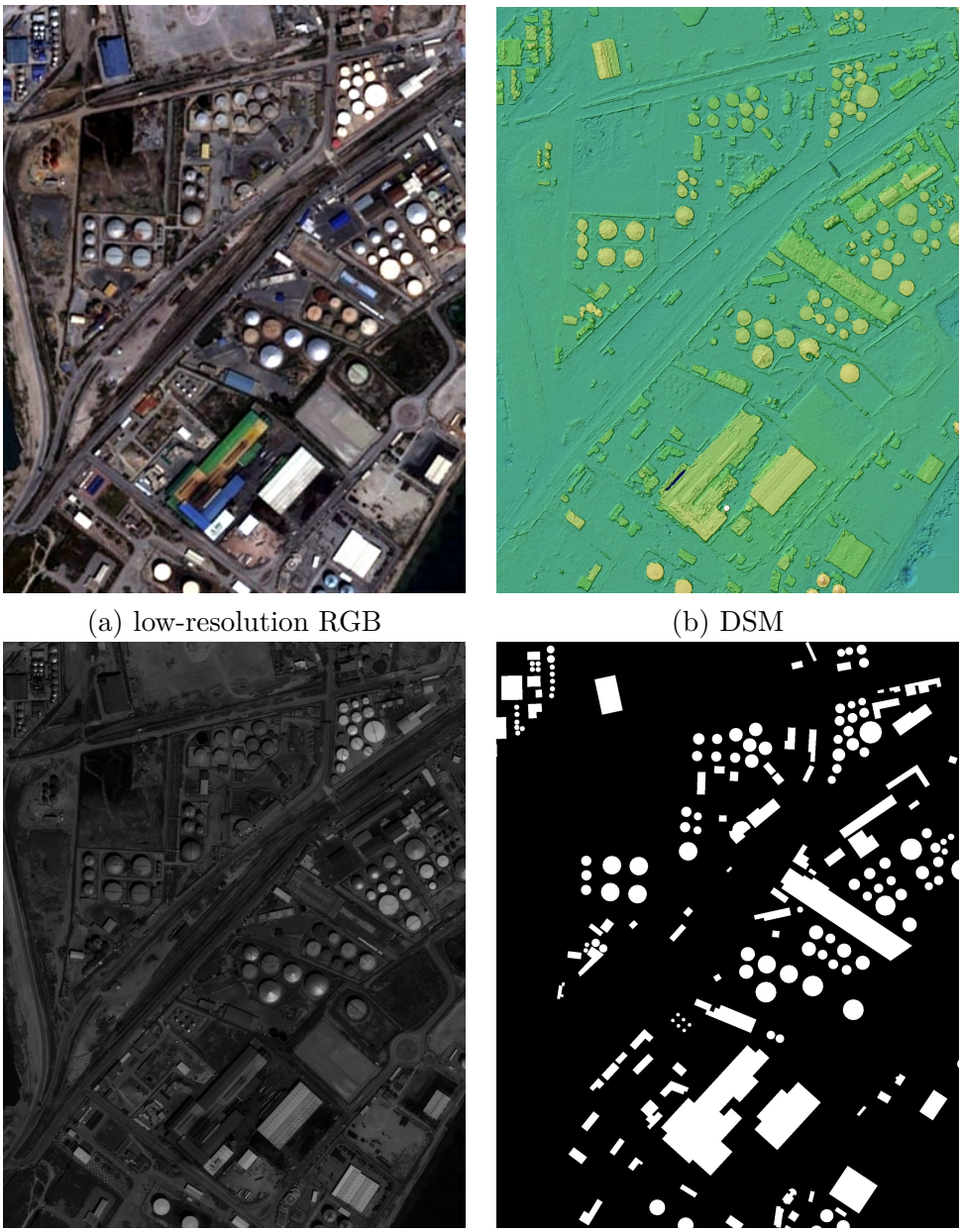
Figure 6.4: Patch 5 of Testarea in Tunis.



(a) low-resolution RGB (b) DSM

(c) PAN (d) OSM building footprints

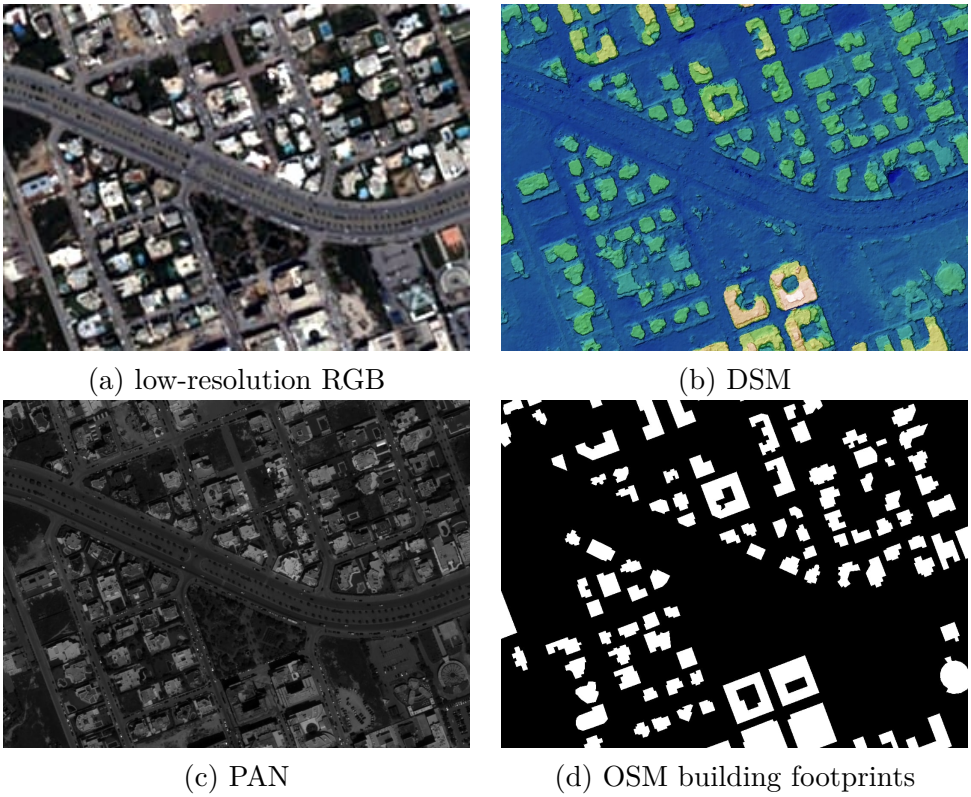
Figure 6.5: Patch 6 of Testarea in Tunis.



(a) low-resolution RGB (b) DSM

(c) PAN (d) OSM building footprints

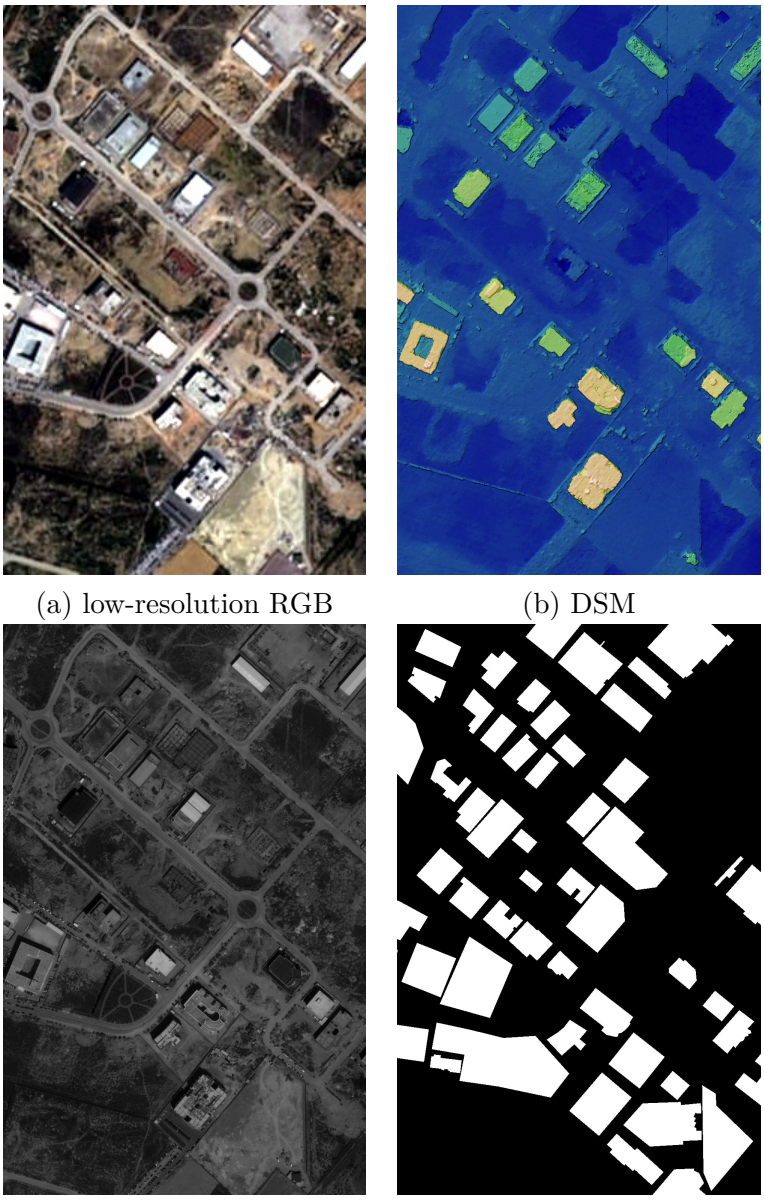
Figure 6.6: Patch 7 of Testarea in Tunis.



(a) low-resolution RGB (b) DSM

(c) PAN (d) OSM building footprints

Figure 6.7: Patch 8 of Testarea in Tunis.



(a) low-resolution RGB (b) DSM

(c) PAN (d) OSM building footprints

Figure 6.8: Patch 9 of Testarea in Tunis.

References

[1] K. Bittner, S. Cui, and P. Reinartz. *Building extraction from remote sensing data using fully convolutional networks.* The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLII-1/W1, 2017, 2017.

[2] K. Bittner, F. Adam, S. Cui, M. Körner, and P. Reinartz. *Building footprint extraction from VHR remote sensing images combined with normalized DSMs using fused fully convolutional networks.* IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2018.

[3] M. Brédif, O. Tournaire, B. Vallet, and N. Champion. *Extracting polygonal building footprints from digital surfce models: A fully-automatic global optimization framework.* ISPRS journal of photogrammetry and remote sensing, vol 77, pp. 57-65, 2013.

[4] N. Ekhtari, M. R. Sahebi, M. J. V. Zoej, and A. Mohammadzadeh. *Automatic building extraction from LIDAR digital elevation models and WorldView imagery.* Journal of Applied Remote Sensing, 2009.

[5] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning.* MIT Press, deeplearningbook.org, 2016.

[6] R. Guercke and M. Sester. *Building Footprint Simplification Based on Hough Transform and Least Squares Adjustment.* Proceedings of the 14th workshop of the ICA commission on generalisation and multiple representation, Paris, 2011.

[7] K. He, X. Zhang, S. Ren, and J. Sun. *Deep Residual Learning for Image Recognition.* eprint arXiv:1512.03385, 2015.

[8] A. Huertas and R. Nevatia. *Detecting Buildings in Aerial Images.* Computer Vision, Graphics and Image Processing 41, 131-152, 1988.

[9] V. Iglovikov and A. Shvets. *TernausNet: U-Net with VGG11 Encoder Pre-Trained on ImageNet for Image Segmentation.* eprint arXiv:1801.05746, 2018.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. *Imagenet Classification with Deep Convolutional Neural Networks.* Advances in Neural Information Processing, 2012.

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. *Gradient-Based Learning Applied to Document Recognition.* Proceedings of the IEEE, 1998.

[12] J. Long, E. Shelhamer, and T. Darell. *Fully convolutional networks for semantic segmentation.* Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431-3440, 2015.

[13] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez. *Convolutional neural networks for large-scale remote sensing image classification.* IEEE Transactions on Geoscience and Remote Sensing, 2017.

[14] D. Marmanis, J. D. Wegner, S. Galliani, K. Schindler, M. Datcu, and U. Stilla. *Semantic segmentation of aerial images with an enselmble of cnns.* ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2016, vol. 3, pp. 473-480, 2016, 2016.

[15] V. Nair and G. Hinton. *Rectified linear units improve restricted boltzmann machines.* ICML'10 Proceedings of the 27th International Conference on International Conference on Machine Learning Pages 807-814, 2010.

[16] Y. Rao, L. He, and J. Zhu. *A residual convolutional neural network for pan-shaprening.* 2017 International Workshop on Remote Sensing with Intelligent Processing (RSIP), 2017.

[17] O. Ronneberger, P. Fischer, and T. Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation.* Medical Image Computing and Computer-Assisted Intervention (MICCAI), Springer, LNCS, Vol.9351: 234–241, 2015, 2015.

[18] F. Rottensteiner, J. Trinder, S. Clode, K. Kubik, and B. Lovell. *A Hybrid Approach for Building Extraction From Spaceborne Multi-Angular Optical Imagery.* IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing.

[19] K. Simonyan and A. Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition.* eprint arXiv:1409.1556, 2014.

[20] A. Turlapaty, Q. Du, B. Gokaraju, and N. H. Younan. *A Hybrid Approach fo rBuilding Extraction From Spaceborne Multi-Angular Optical Imagery.* IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2012.

[21] J. Yang, X. Fu, Y. Hu, Y. Huang, X. Ding, and J. Paisley. *PanNet: A deep network architecture for pan-sharpening.* 2017 IEEE International Conference on Computer Vision (ICCV), 2017.