TRAINING A FULLY CONVOLUTIONAL NEURAL NETWORK WITH INCOMPLETE, IMPERFECT AND IMBALANCED GROUND TRUTH ON ROOF-TYPE SEGMENTATION

PHILIPP SCHUEGRAF

Professor: Prof. Dr. David Spieler
Supervisor: Dr. Ksenia Bittner
University: University of Applied Sciences Munich
Department: Department for Mathematics and Computer Science
Cooperation Partner: German Aerospace Center, Earth Observation Center

Date of submission: 01.04.2021

ABSTRACT

Nowadays, satellites constantly supply world-wide coverage of large-scale, Very High-Resolution (VHR) satellite imagery. The interpretation of such imagery is very expensive if done by a human. However, modern deep learning methods automatically extract semantically meaningful features for image interpretation if trained on a set of input-output pairs of high quality. In 3D reconstruction, the automatic prediction of the roof-type is an open problem. Even though some research has been done to predict the roof-type, either the number of classes was limited to flat and non-flat [1], or the acquisition of the ground truth was done by manually labeling many buildings [2]. But roof-type information is publicly available through the internet, such as contained in the CityGML [3] dataset of Berlin, Germany. On the other hand, such datasets have only very few samples of some classes, contain mislabeling and are incomplete. But there are methods for dealing with class-imbalance, such as the focal loss [4] and inverse frequency weights and recently, an adaption of the loss function in deep learning has been proposed, which makes the training of an Fully Convolutional Neural Network (FCN) more robust to errors in the ground truth [5]. Furthermore, Semi-Supervised Learning (SSL) was extended from classification to semantic segmentation. For example, Virtual Adversarial Training (VAT) was evaluated for dense, pixel-wise classification on a benchmark dataset [6]. In this thesis, these solutions are assembled into a combined loss $\mathcal{L}_{COM}$ to train a DeepLabv3+ [7] for roof-type segmentation on an imbalanced, imperfect and incomplete training dataset. The proposed method achieves considerable improvements and successfully predicts the roof-type in many cases. But it also fails in some cases, which are visualized and discussed.

ACKNOWLEDGEMENTS

I want to thank Prof. Dr. David Spieler, who always encouraged me on my path during the last two years. He was my supervisor during my masters program and I would like to thank him for his support in academic and organizational affairs. I also want to thank Dr. Ksenia Bittner, who took great responsibility in supporting me and the success of our common project, which I feel we approached together as a team. I also want to thank her for introducing me to deep learning and remote sensing during my bachelors thesis in 2018 and during this masters thesis. Next, I want to thank Corentin Henry, whose expert knowledge in the field of deep learning for remote sensing and his never ending stream of ideas were a valuable resource of progress and motivation during the completion of this thesis. Furthermore, I want to appreciate the support of Prof. Dr. Gerta Köster during the first year of my master program. At the beginning of my masters program, my research topic was far away from remote sensing. But the advice she gave me exceeds the boundaries of any specific scientific field. I also want to thank my entire family, starting with my mother Angelika, whose love and support during my entire life made it possible for me to strive for the goals in life I really care about. I also want to thank my father Werner whose support, original thought, continuous encouragement and emphasize on family cohesion are of great importance in my life. My siblings Ben, Paula and Karla are a great enrichment for me. My girlfriend Elisabeth brings joy and love to my life everyday.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

CE Cross-Entropy

CNN Convolutional Neural Network

DSM Digital Surface Model

FCN Fully Convolutional Neural Network

GAN Generative Adversarial Network

GSD Ground Sampling Distance

IoU Intersection over Union

KL Kullback-Leibler

LDS Local Distributional Smoothness

LOD Level of Detail

NIR Near Infrared

OSM Open Street Map

SOA State-of-the-Art

SVM Support Vector Machine

SSL Semi-Supervised Learning

VHR Very High-Resolution

VAT Virtual Adversarial Training

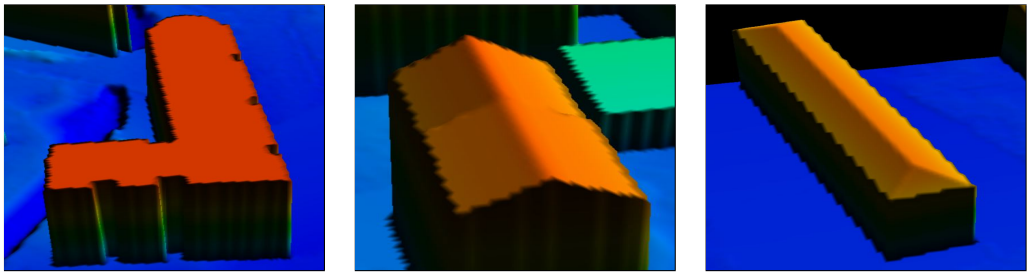
WV-1 World-View-1

WV-4 World-View-4

1 INTRODUCTION

Nowadays, the biggest potential in the engineering discipline of remote sensing lays in the rise of deep learning methods to make efficient and effective use of the vast amount of both airborne and space-borne data. Space-borne data, from satellites such as the WorldView spectral imaging devices, is especially cost-efficient, since they can orbit the earth non-stop for many years and supply a constant stream of Very High-Resolution (VHR) imagery. For example, multi-view panchromatic images of a scene can be used to create a Digital Surface Model (DSM) and the combination of a high-resolution panchromatic image and a multi-spectral image can yield a high-spatial and spectral-resolution pan-sharpened multi-spectral image. The huge amount of image data from satellites poses a big challenge, which is to extract semantic meaning from these images. Several engineering fields require semantic information annotated to space-borne images to produce value from them. These fields include urban planning and reconstruction, disaster monitoring and 3D city modeling.

Modern deep learning methods are the State-of-the-Art (SOA) for many computer vision tasks, such as image classification, semantic segmentation and instance segmentation. It showed, that the success in computer vision transfers to the field of remote sensing [8]. Although computer vision benchmarks are performed on multi-media data, which is very different to space-borne imagery, the spatial and spectral feature extraction mechanism of recent deep learning methods also works in remote sensing. But there are also show-stoppers for deep learning methods. Since most of these methods are supervised, they require huge amounts of labeled data to achieve high quality results. But different to the acquisition of imagery, it is expensive in time and money to manually label data for large areas to obtain a good ground truth for deep learning. There are also labeled datasets in remote sensing, which are big, but are labeled incompletely. For example, there is CityGML [3] data of Berlin, where only about 30 % of the buildings are labeled with a roof-type (compare Figure 6). Furthermore, roof-type data is often highly imbalanced (see Figure 6). Some roof-types occur much less frequent then others, which causes standard deep learning approaches to ignore minority classes. Another big challenge is incorrectly labeled data (compare Figure 2). Many buildings are labeled incorrectly due to time difference between the acquisition of the satellite image and the annotation of the roof-type, imperfect pre-processing and human error. Furthermore, the prediction of classes that are characterized by the geometric appearance, such as roof types in satellite images, requires the input to the deep



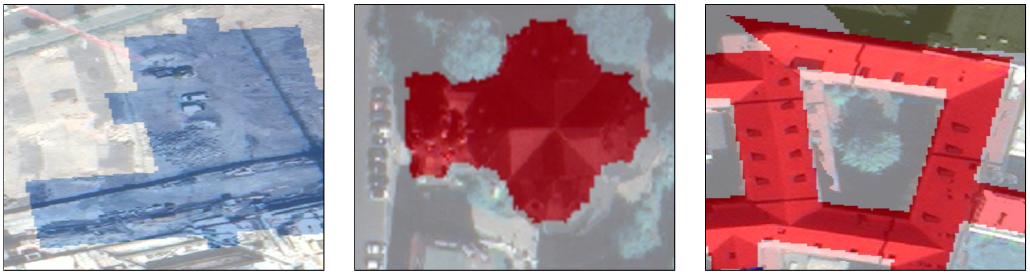
(a) mono-plane **(b)** gable **(c)** hip

Figure 1: The roof-types mono-plane, gable and hip. In **(a)**, a mono-plane roof is shown. Mono-plane roofs can also have an inclination. In **(b)**, there is a gable roof. In **(c)**, a hip roof is shown. The hip class also includes half-hip buildings, with only one triangular end, and the other end as in the gable class. The mono-plane class is characterized by a single plane with an arbitrary angle of less than 90°. The gable class is characterized by a ridge line and equal angles of the neighboring planes. The hip class is characterized by a ridge line and one or two isosceles triangles.

learning model to contain these features. Otherwise, the model's predictive power is strongly limited. This thesis proposes a method to tackle the three issues **1)** class imbalance, **2)** incorrectly labeled data and **3)** incompletely labeled data.

There have been deep learning based methods before for roof-type classification from different image sources [9], [10], [2], [1]. Most of this work was done on classifying instances of buildings, which requires instance localization information of the buildings [9], [10], [2]. Also, semantic segmentation of roof-types was performed, but only for the three classes like background, flat and non-flat [1]. This thesis closes the gap between semantic segmentation of three classes and building roof-type classification with many classes, attempting to segment the five classes background, mono-plane, gable, hip and other (see Figure 1). The other class is any other class than background, mono-plane, gable or hip, or a composition of multiple roof-types.

The following chapters are organized as follows: Chapter 2 gives an overview of previous work on roof-type classification and semantic segmentation, semi-supervised learning, class imbalance and incorrectly labeled data. Chapter 3 introduces the key concepts of the evaluated methods. Chapter 4 shows the evaluation procedure. Chapter 5 explains the data preparation, the carried out experiments and the metrics used to quantitatively judge their success. Chapter 6 presents the results and 7 concludes the thesis. Finally, Chapter 8 gives an outlook on possible further directions of this research.



(a) **(b)** **(c)**

Figure 2: A visualization of some incorrect labels in the roof-type segmentation ground truth. In the background, there is an RGB image of a scene and in the foreground, there is a polygon in either blue or red, representing the label of the building in the ground truth. Blue color represents the unknown building class and red color the flat building class. In **(a)**, an unknown building was annotated, but in the RGB image, the corresponding image is demolished. In **(b)** and **(c)**, clearly non-flat buildings are annotated as flat. In **(c)**, the roof-type annotation does not overlap completely with the building in the RGB image.

2 RELATED WORK

2.1 BUILDING ROOF-TYPE CLASSIFICATION

To this day, only very little work was done on the classification of building roof-types. However, some efforts have been made to accomplish this hard task.

Partovi et al. [2] investigate the potential of pretrained Convolutional Neural Networks (CNNs) for roof-type classification and apply two different methods to pan-sharpened RGB images. One of the methods refines an ImageNet pretrained CNN [11] by hand-labeled building roof patches and the other one uses deep features from a hidden layer of such a CNN and passes these features to an Support Vector Machine (SVM). The roof-type classes are flat, gable, half-hip, hip, pyramid, mansard and complex.

In Axelsson et al. [9] an ensemble of ten CNNs is trained to predict the building roof-types flat and non-flat. As an input, different combinations of spectral and height information is used, which include Near Infrared (NIR), red, green and height. The combination of red, green and height scored the best classification accuracy, whereas the combination of NIR, red and green was only slightly worse. Passing only height information to the CNNs was significantly less accurate.

Alidoost and Arefi [10] propose a CNN-based, automatic roof-type segmentation algorithm. The segmentation is done by first extracting building tiles by building mask generation and than classifying each building pixel to a roof-type. They use both RGB and DSM data as the input. The training is carried out by first fine-tuning an ImageNet-pretrained CNN on the RGB data and than further fine-tuning the model with the DSM data. The extracted deep features are fused by selecting the class with the maximum score among them.

Another approach by Bittner et al. [1] even achieved promising results on roof-type segmentation, assigning to each pixel of a half-meter resolution DSM either of the classes background, flat roof or non-flat roof in a multi-task setting.

However, none of these works addresses the segmentation of more than two roof-types and the background class.

2.2 CLASS IMBALANCE

Class imbalance is a problem in deep learning, because standard loss functions, such as the Cross-Entropy (CE) loss, can be minimized by always favoring the dominant class.

In object detection, the location of an object in an image is predicted by a model. The locations where no object is, are the dominant class, in this case. Lin et al. [4] propose the focal loss function, which is a scaled CE loss function, where the scale depends on how confident the model is about its prediction. If the model is highly confident, the pixel's influence on the parameter update is scaled by a real power of 1 minus the confidence. The confidence based scaling together with a weighting parameter leads to a high increase in performance in object detection.

Class-imbalance was also previously addressed in remote sensing. In road segmentation from aerial imagery, the number of non-road pixels is much higher than that of road pixels. Henry et al. [5] use an adapted version of the dice loss, to better balance the influence of each pixel in training, such that road pixels do not get ignored by the training process.

2.3 SEMI-SUPERVISED LEARNING AND ADVERSARIAL TRAINING

Missing annotations are another great challenge in roof-type datasets and machine learning in general. Semi-Supervised Learning (SSL) together with consistency regularization have been investigated by many researchers.

Yalniz et al. [12] present an approach on semi-supervised learning for large-scale image classification. In their work, they propose to use a student-teacher approach, where a teacher is trained on the labeled dataset and the labels of the unlabeled dataset are inferred by the teacher. Then, the student model is trained on both the labeled dataset and the originally unlabeled dataset with the inferred labels.

Goodfellow et al. [13] reason about why small, worst-case perturbations of inputs to classifiers can lead to highly confident but incorrect classification results and propose a method called adversarial training to make classifiers robust against such adversarial attacks. Even better, the adversarial training has a regularizing effect on the classifier, such that its test accuracy increases.

In their work, Bai and Urtasun [14] use a workflow consisting of Fully Convolutional Neural Networks (FCNs) and a CNN to regularize building boundaries in an adversarial setting. They use masks generated by a Mask-RCNN and ideal masks from Open Street Map (OSM) as inputs and generate refined masks as an in-between product in the adversarial setting, where the discriminator classifies whether the regularized mask is a real mask or a fake one. The refined building footprints are more regular than

those produced by the Mask-RCNN and also have a higher Intersection over Union (IoU). Zorzi and Fraundorfer [15] observed a similar effect.

2.3.1 *Virtual Adversarial Training*

Adversarial training as in Goodfellow et al. [13] requires knowledge of the class label. Since this is not always given, Virtual Adversarial Training (VAT) was developed by Miyato et al. [16] and gradually extended and improved [17], [18].

In Miyato et al. [16] the authors use VAT to improve the Local Distributional Smoothness (LDS) of CNN (see Subsection 3.1.1). This way, the model becomes more robust against adversarial attacks and generalizes better to previously unseen data. VAT has relatively low computational cost. To compute the approximated gradient of the LDS for a neural network, only two pairs of forward and backward propagations are necessary. VAT, compared to adversarial training, generates adversarial examples without using the label information and therefore is applicable to the semi-supervised setting. Miyato et al. [17] apply VAT on supervised and semi-supervised benchmark tasks in the text domain. But in the text domain, one-hot encoded word representations are used. That is why it is inappropriate to apply small perturbations to the input directly. Instead, Miyato et al. [17] add adversarial and virtual adversarial perturbations to the word embeddings. Miyato et al. [18] improved VAT, such that only two forward and backward passes are needed to compute the gradient of the virtual adversarial loss. The authors apply the entropy minimization principle, which makes their approach to VAT achieve state-of-the-art performance for semi-supervised learning tasks.

Most of the work in the research area of SSL was contributed to classification tasks, whereas semi-supervised semantic segmentation was also recently studied more [19], [20], [21], [6].

In French et al. [6], VAT was extended to semantic segmentation by regarding the adversarial consistency regularization loss as a spatial map and averaging the LDS values over the map. However, in this thesis, the consistency loss is computed by the outputs of only one neural network, where in French et al. [6], a student-teacher approach is used.

2.4 IMPERFECT GROUND TRUTH

The quality of the ground truth is very important for the performance of deep learning methods. Handling imperfect ground truth in deep learning for remote sensing is a very young, but promising field.

In Henry et al. [5], the problem of imperfect ground truth is tackled by replacing the usual CE loss by a soft-bootstrapped adapted dice loss, which replaces the ground truth by a weighted average of the ground truth and

a model's predictions. That way, the learning target is shifted, such that it does not ignore what the model has already learned. Furthermore, they apply artificial noise to the ground truth, to rectify the topological noise. With the rectified noise, the model becomes more robust to such noise. Combining the soft-bootstrapped adapted dice loss with the artificial noise leads to a strong improvement in quantitative evaluation of a deep learning based approach to road segmentation from aerial imagery.

3 BACKGROUND

3.1 INTRODUCTION TO SEMI-SUPERVISED LEARNING

Deep Neural Networks are nowadays very popular in applications with labeled datasets $\mathcal{D}_l$. In the case of fully labeled datasets, the learning procedure is called supervised. Since it is often hard to acquire labels for data, the size $|\mathcal{D}_l|$ of labeled datasets is limited. Nowadays, the acquisition of huge unlabeled datasets $\mathcal{D}_u$ is usually easy. Therefore it is possible that learning based on both labeled and unlabeled datasets $\mathcal{D} = \mathcal{D}_l \cup \mathcal{D}_u$ is an improvement over learning on only on $\mathcal{D}_l$. In this section, the conditions under which such methods work are examined and different approaches are presented. This section is based on the surveys of Ouali et al. [22] and Engelen and Hoos [23].

Learning with a dataset $\mathcal{D}$ which contains both labeled and unlabeled data is called SSL. The labeled portion $\mathcal{D}_l$ can be very small compared to the total amount of data. This is due to the relative hardness of creating labels compared to the acquisition of huge datasets. The aim of SSL is to leverage the unlabeled portion $\mathcal{D}_u$ to improve the performance over a fully supervised algorithm using only $\mathcal{D}_l$. In general, $\mathcal{D}_l$ gives information about the conditional density $p(y|x)$, where y is the true label of a data sample $x \sim p(x)$, coming from the data distribution $p(x)$ and $\mathcal{D}_u$ supplies inside into the structure of $p(x)$. The conditional density function $p(y|x)$ is what is intended to be inferred by a parametric prediction function f_θ with trainable parameters θ. In supervised learning, only $p(y|x)$ is approximated, whereas in SSL, $p(x)$ is leveraged to incorporate information of the density distribution of the data, which can improve the decision boundary.

SSL requires several assumptions to hold. These assumptions are especially important for the generalization from the training data to unseen test cases. These are assumptions are

- **(a) The Smoothness Assumption** Two close points x_1 and x_2, which lie in a high-density region of the data, have close corresponding outputs y_1 and y_2 [24].
- **(b) The Cluster Assumption** Two points x_1 and x_2 in the same cluster are of the same class [24].
- **(c) The Manifold Assumption** The high-dimensional data is located on a low-dimensional manifold [24].

Details about these assumptions are given in Chapelle et al. [24].

The main question here is when it improves performance to incorporate unlabeled data. The question has no easy answer. Instead, as usual in machine learning, SSL is yet another approach to given problems, which has to be evaluated and compared to supervised methods but is not an automatic upgrade to them.

Intrinsically SSL-methods have in common, that they optimize a loss function which include terms for labeled and unlabeled data. They extend supervised methods by including unlabeled data in the objective function.

Perturbation-based methods use the smoothness-assumption of SSL. If an input is locally perturbed, than the output should not greatly vary. This robustness does not rely on the true label of the data points, which allows the incorporation of unlabeled data. One extension to a supervised method exploiting the smoothness-assumption would be to add a distance measure of the outputs of a data point and its perturbed version. Neural networks allow the incorporation of unlabeled data by an additional unsupervised loss term, which makes it straightforward to extend them to the SSL setting.

3.1.1 *Virtual Adversarial Training*

Perturbation-based methods apply general perturbations to each input, to augment the data, pushing the trained model to have smooth outputs in random directions, which can improve the generalization performance. This way, the direction in the input space, where the model of the label probability $p(y|x)$ is most sensitive to, which is called the adversarial direction, might be neglected. Inspired by the work of Goodfellow et al. [13], where the model is trained to be invariant to the input in the adversarial direction, Miyato et al. [18] propose VAT as a regularization technique, that trains the model to be invariant to the input in the adversarial and random directions. VAT is called virtual, because the adversarial direction is approximated from unlabeled data, which puts it into the realm of SSL. To train the model to be identically smooth around each data sample, VAT smooths the output distribution in its most adversarial direction [22]. Given the data sample x, we want to obtain the adversarial perturbation r_{adv}, which alters the model's prediction most [22]. First, a Gaussian noise r is sampled, equally dimensioned as the input x and then its gradient $grad_r$ of a loss between $f_\theta(x)$ and $f_\theta(x+r)$, where f_θ is the model parameterized by θ, is computed with respect to x. As the distance measure, the Kullback-Leibler (KL) divergence

$$d_{KL}(P,Q) = \sum_{x \in X} P(x) \times \log \frac{P(x)}{Q(x)}, \tag{1}$$

where P and Q are random distributions and X is the dataset, is applied. Then, r_{adv} is computed by normalizing and scaling $grad_r$ by ϵ [22]. The computation is summarized by the following sequence of computations:

$$r \sim \mathcal{N}(0, \frac{\xi}{\sqrt{dim(x)I}}), \tag{2}$$

$$grad_r = \nabla d_{KL}(f_\theta(x), f_\theta(x+r)), \tag{3}$$

$$r_{adv} = \epsilon \frac{grad_r}{||grad_r||}, \tag{4}$$

where $\mathcal{N}(0, \kappa)$ is the zero-centered normal distribution with the normalized co-variance matrix κ.

These steps are the first iteration of the approximation of r_{adv}, which can be improved by setting $r_{adv} = r$ and then repeating the computations in Equations 3 and 4. Given the heavy computational load of this procedure, including a forward and a backward pass through the model, only the first iteration will be used in practice. Next, the unsupervised loss is computed as the KL divergence between the prediction of the model for the unperturbed and the prediction of the model for the perturbed input

$$\mathcal{L}_u(x) = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}_u} d_{KL}(f_\theta(x), f_\theta(x + r_{adv})), \tag{5}$$

which is scaled by α and added to the classification loss $\mathcal{L}_l$ to compute the loss

$$\mathcal{L} = \mathcal{L}_l + \alpha \mathcal{L}_u \tag{6}$$

VAT can easily be extended to semantic segmentation by averaging $\mathcal{L}_u$ over all pixels of the predicted class scores.

3.2 FULLY CONVOLUTIONAL NEURAL NETWORKS

FCNs are the SOA for semantic segmentation in general and for remote sensing problems. In this Section, their main structural components, training and architecture are emphasized.

3.2.1 *Structure of Fully Convolutional Neural Networks*

FCN's core components are convolutional layers and transposed convolutional layers. The convolutional layers are linear, parametric transformations which extract features of different levels of abstraction [25]. Convolutional layers apply multi-dimensional kernels to multi-channel 2D-arrays c^{l-1} to extract features. The output of a convolution layer

$$c^l_{m,n} = \sigma\Big(\sum_{i=-q}^{q}\sum_{j=-r}^{r} w_{i,j} c^{l-1}_{m+i,n+j} + b_{i,j}\Big) \tag{7}$$

is a weighted sum, where $q = \frac{H-1}{2}$, $r = \frac{W-1}{2}$, w are matrices of weights with height H and width W, and b is the bias. Because convolution decreases the size of its input, padding is applied if the input dimension shall be preserved. In each convolutional layer, $\frac{H-1}{2}$ zeros at the borders of the input can be padded to preserve the size of the input in the feature map. Transposed convolutional layers are also parameterized through kernels and work very similar to convolutional layers. Transposed convolutional layers often have a stride to increase the resolution of their input. As it is not easy to dimension the convolutional and transposed convolutional layers properly, often the number of channels is chosen to be rather too big than too small and then, regularization is applied to avoid overfitting. Here, overfitting is caused by an FCN which has a very high capacity, such that it adapts almost perfectly to the training data but performs poorly on unseen data.

As mentioned earlier, to make an FCN applicable to tasks with a non-linear dependence of the output on the input, non-linear activation functions are applied to the output of the convolutional and transposed convolutional layers. A common activation function is $\tanh : \mathbb{R} \to [-1, 1]$. Its range makes it particularly useful if the output needs to be positive and negative. In some regions in its domain, the derivative of the $\tanh$ function goes to zero, which is unfavorable in gradient-based training procedures [25]. Thus, the rectified linear unit ReLU, which has a constant, non-zero derivative for positive inputs, is a common choice in FCN architectures.

To increase their receptive field, FCNs often use maximum pooling layers which sub-sample their input. In this way, the spatial context is aggregated and objects covering larger areas can be recognized even with small kernels, i. e. if H and W are small.

To obtain class probabilities, the scores $f_\theta(x)_i$ are fed to the $\mathrm{softmax}$ function

$$\sigma(f_\theta(x)) = \frac{e^{f_\theta(x)_i}}{\sum_{j=1}^{K} e^{f_\theta(x)_j}} \tag{8}$$

for $i = 1, ..., K$, where $\sigma(f_\theta(x))_i$ refers to the i-th element of the $softmax$ function, K is the number of classes. To compute the classification output per pixel, the $argmax$ function is applied, assigning to each pixel

$$i_{max} = \arg\max_i(\sigma(f_\theta(x))). \tag{9}$$

3.2.2 *Training of Fully Convolutional Neural Networks*

For training a FCN, a loss function (see 3.2.3) $\mathcal{L}$ is applied to the $softmax$-output $\sigma(f_\theta(x))$. Using back-propagation, the gradient $\nabla_{w,b}\mathcal{L}$ of the loss with respect to the weights w and the bias b is computed. $\mathcal{L}$ measures the dissimilarity between the output of a neural network and the corresponding label. Training is the search for a locally optimal point in parameter-space with respect to $\mathcal{L}$. 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 re-scale the parameter update. To avoid the training algorithm to oscillate around local optimal solutions, momentum can be introduced. Let $g^{(i)}$ be $\nabla_{w,b}\mathcal{L}$ at iteration i and μ be the momentum hyper-parameter, which controls how much the gradient of the weight update of the previous iteration contributes to the current weight update, then the parameter update $\Delta\theta^{(i)}$ is computed by

$$\Delta\theta^{(i)} = (1-\mu)\alpha g^{(i)} + \mu\Delta\theta^{(i-1)}. \tag{10}$$

The training of a differentiable FCN $\hat{f}$ is commonly done by **(i.)** computing the gradient $\nabla_\theta\mathcal{L}(\hat{f}(x), y, \theta)$ of a loss function $\mathcal{L}(\hat{f}(x), y, \theta)$ with respect to the FCN's parameters θ and **(ii.)** moving θ in the parameter space in the direction of the gradient computed in **(i.)**. If this gradient becomes too large, the optimization algorithm might diverge. As a mean against that, gradient norm clipping reduces the magnitude of the gradient, such that it does not become bigger than a threshold. In gradient based optimizers, the back-propagation algorithm [26] (for and introduction to back-propagation, see Goodfellow et al. [25], p. 197 ff.) is used. It utilizes the chain-rule of calculus to compute the gradient in **(i.)**. In mini-batch optimization algorithms, there is a trade-off between the quality of the gradient estimate and the computational efficiency. In most cases, the gradient is averaged over a mini-batch of size N_{bs}, where $1 \leqslant N_{bs} \leqslant N_{samples}$, where $N_{samples}$ is the number of rows of samples in the training data. In the optimization algorithm ADAM [27], the first and second order moments are used, which incorporates higher-order information into $\Delta\theta$ along the path of preceding locations in the parameter space. Incorporating higher order information is an effective means to reduce oscillation around a local minimum, because

it heuristically scales the parameter updates such that their magnitudes are linked to the distance to the local minimum.

3.2.3 *Loss Functions*

This thesis includes the empirical exploration of different loss functions for different purposes. These loss functions are the CE loss

$$\mathcal{L}_{CE}(x, y, p) = -\sum_{b=1}^{N_{BS}} \sum_{i=1}^{N_{cls}} y_{bi} \log(p_{bi}), \tag{11}$$

the soft-bootstrapped CE loss [5]

$$\mathcal{L}_{SBCE}(x, y, p, \beta) = -\sum_{b=1}^{N_{bs}} \sum_{i=1}^{N_{cl}} (\beta y_{bi} + (1-\beta) p_{bi}) \log(p_{bi}), \tag{12}$$

the focal loss [4]

$$\mathcal{L}_{FOC}(x, y, p, \gamma) = -\sum_{b=1}^{N_{bs}} \sum_{i=1}^{N_{cl}} (1-p_{bi})^{\gamma} y_{bi} \log(p_{bi}), \tag{13}$$

and the weighted loss

$$\mathcal{L}_{BAL}(x, y, p, w, \lambda) = \frac{1}{\Sigma W} \sum_{b=1}^{N_{bs}} \sum_{i=1}^{N_{cl}} w_{bi} \mathcal{L}(x, y, p, \lambda)_{bi}, \tag{14}$$

where $p_{bi} = \sigma(f_\theta(x_b))_i$, f_θ is a neural network parameterized by θ, x_b is a sample input to the neural network, $y_{bi} \in \{0, 1\}$ is a element of the one-hot-encoded label vector y, w_{bi} is the loss weight of class i and at batch index b, β and γ are control hyper-parameters, $\lambda \in \{\beta, \gamma\}$, $\mathcal{L}(...)_{bi}$ is either $\mathcal{L}_{CE}$, $\mathcal{L}_{SBCE}$ or $\mathcal{L}_{FOC}$, σ is the $softmax$ activation function (see Equation 8) and $\Sigma W = \sum_{b=1}^{N_{bs}} \sum_{i=1}^{N_{cl}} w_{bi}$ to make $\mathcal{L}_{BAL}$ a weighted average of loss values.

The generic loss function in Equation 14 is applicable to classification tasks, but can be extended to the semantic segmentation loss

$$\mathcal{L}_{SEG}(x, y, p, w, \lambda) = \frac{1}{HW} \sum_{r=1}^{H} \sum_{c=1}^{W} \mathcal{L}(x, \hat{y}_{rc}, \hat{p}_{rc}, w, \lambda), \tag{15}$$

where $\hat{p}$ and $\hat{y}$ are tensors, which contain for each pixel in a two dimensional grid a class-wise $softmax$ output ($\hat{p}$) or a one-hot-encoded matrix

of dimension $N_{bs} \times N_{cl}$ ($\hat{y}$), by summation over the vertical index r, where $0 < r \leqslant H$ and over the horizontal index c, where $0 < c \leqslant W$ (see Equation 15). Two more loss functions, which are specialized on semantic segmentation are the adapted dice loss [5]

$$\mathcal{L}_{DICE}(x, y, p) = \frac{1}{N_{cl}} \sum_{i=1}^{N_{cl}} (1 - \frac{1 + 2\sum_{u=1}^{N_{pix}} p_{iu} y_{iu}}{1 + \sum_{u=1}^{N_{pix}} p_{iu}^2 + y_{iu}^2}) \tag{16}$$

and the soft bootstrapped adapted dice loss [5]

$$\mathcal{L}_{SBD}(x, y, p, \beta) = \frac{1}{N_{cl}} \sum_{i=1}^{N_{cl}} (1 - \frac{1 + 2\sum_{u=1}^{N_{pix}} p_{iu}(\beta y_{iu} + (1-\beta)p_{iu})}{1 + \sum_{u=1}^{N_{pix}} p_{iu}^2 + (\beta y_{iu} + (1-\beta)p_{iu})^2}), \tag{17}$$

where p_{iu} and y_{iu} are 0 if pixel u in a batch belongs to class if $j \neq i$ and 1 otherwise and $N_{pix} = N_{bs}HW$ is the number of pixels in a given batch. Since $\mathcal{L}_{DICE}$ and $\mathcal{L}_{SBD}$ are averaged over the class dimension, weighting of the classes can also easily applied, by multiplying each member of the sum by an arbitrary weight and dividing by ΣW instead of N_{cl}.

It is a big problem in deep learning if the ground truth contains many missed classifications. Based on a large enough data set and model that generalizes well, a model can be robust against errors in the ground truth, but only to an extent. The soft-bootstrapped CE loss $\mathcal{L}_{SBCE}$ [5] and the soft-bootstrapped dice loss $\mathcal{L}_{SBD}$ [5] take into account that the prediction of the model, based on the previous training iterations, might be more accurate than the ground truth by replacing the ground truth with a linear combination of y and p. The parameter β regulates the contribution of the ground truth to that term. Note that $\mathcal{L}_{SBCE}(x, y, p, 1.0) = \mathcal{L}_{CE}(x, y, p)$ There is also a hard-bootstrapped loss [5], which replaces the probability vector p by the one-hot-encoded vector $\hat{p}$. But in Henry et al. [5], the soft version produced better results in road segmentation from aerial imagery.

Also, the CE loss $\mathcal{L}_{CE}$ is commonly used in semantic segmentation task. But due imbalance of the ground truth, it leads to a model which focuses only on the strongly represented classes and neglects the others. The focal loss $\mathcal{L}_{FOC}$ makes the model learn in dependence of its confidence. The more confident the model is, the less the value of $\mathcal{L}_{FOC}$ is, such that rare classes are emphasized, because they usually do not have high confidences when training is done with $\mathcal{L}_{CE}$. The hyper-parameter γ controls by how much the underrepresented classes are emphasized. For example, $\gamma = 0$ corresponds to $\mathcal{L}_{CE}$. The dice loss $\mathcal{L}_{DICE}$ also puts an emphasis on the underrepresented classes since its magnitude for each class is independent from the number of pixels in the batch belonging to the class but depends on the ratios of pixels in the batch which overlap with the ground truth for each class.

Another strategy to balance the classes during training is the application of loss weights (see Equation 14). The loss weights can be costumed (see Subsection 3.2.4) to empirically produce well-balanced results.

3.2.4 *Loss Weights*

There are infinitely many ways to compute loss weights for $\mathcal{L}_{BAL}$. In this thesis, the square-root inverse

$$(w_{root})_{bi} = \frac{1}{\sqrt{freq_i}}, \tag{18}$$

inverse

$$(w_{inv})_{bi} = \frac{1}{freq_i}, \tag{19}$$

and squared inverse weights

$$(w_{sqr})_{bi} = \frac{1}{freq_i^2}, \tag{20}$$

are compared. The three strategies are based on the frequency of class i

$$freq_i = \frac{cl_i}{\sum j = 1^{N_{cl}} cl_j}, \tag{21}$$

where cl_i is, in the case of semantic segmentation, the number of pixels corresponding to class i in the ground truth. For both strategies, the quality of the parameter updates during optimization depends on the distribution of classes in each batch. Therefore, increasing N_{bs} makes the distribution of classes in each optimization step better represent the distribution of classes in the training data, from which the loss weights are computed.

3.2.5 *Network Architectures*

The choice of the network architecture can have an impact on the performance of deep learning algorithm. The ResNet101-DeepLabv3+ and Dense Unet architectures are examples of FCNs, which have been successfully applied to remote sensing problems [1], [5].

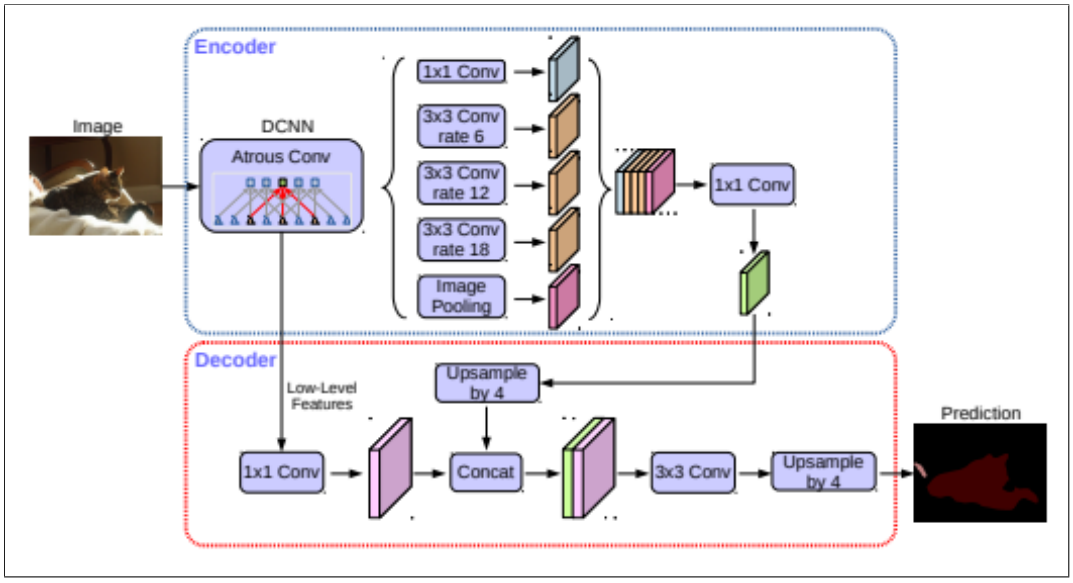


Figure 3: The Figure was taken from Chen et al. [7]. It represents the encoder decoder architecture of the DeepLabv3+ FCN.

3.2.5.1 *ResNet101-DeepLabv3+*

Equation 7 shows how the standard convolution is computed. The ResNet-architecture [28] is a CNN, which relies on the convolution operation. In the ResNet101-DeepLabv3+-architecture [7], the convolution is adapted, such that its receptive field is increased. This is done by applying holes to the convolutional filter. This version of the convolution is called atrous convolution ("trous" is french for the english word holes). Atrous convolution

$$c^l_{m,n} = \sigma\Big(\sum_{i=-q}^{q}\sum_{j=-r}^{r} w_{i,j}\, c^{l-1}_{m+ai,n+aj} + b_{i,j}\Big) \tag{22}$$

adds the hyperparameter a to the convolution. This way, the number of parameters in the convolution does not change, but the receptive field can be increased by setting $a > 1$. Furthermore, ResNet101-DeepLabv3+ uses batch normalization (see [25]) to make the training faster. It does so by normalizing the output of the convolution, before applying the non-linearity. The new mean and standard deviation of the output of the convolution are learnable parameters in the architecture. This makes it easier for the next layer to adapt its parameters, such that they take into account the distribution of its input. Another technique used in the ResNet101-DeepLabv3+ is the atrous spatial pyramid pooling, where atrous convolutions with multiple different values for a, a convolution with a scalar filter and global average pooling are applied to the deep features of the ResNet101, without the fully connected layers, in parallel, concatenated and then passed through another convolutional layer with a scalar filter. The purpose of

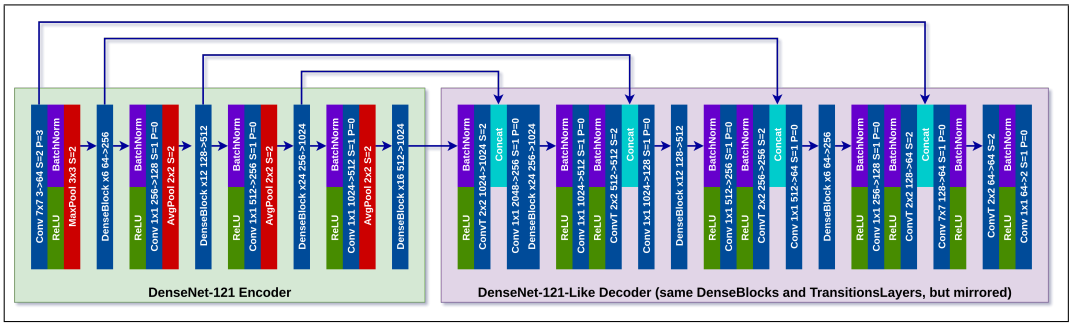


Figure 4: The Figure was taken from Henry et al. [5]. It shows, how in the Dense Unet architecture, the encoder and decoder are connected by skip connections on each level of spatial resolution.

atrous spatial pyramid pooling is to combine deep features, with a high semantic expressiveness, at different receptive fields to further improve these features value for discrimination between classes. To bring the features to the same spatial resolution as the input, a small decoder module (see Figur 3) is applied, that incorporates lower level features from the ResNet101 at a high spatial resolution to improve the boundaries of the segmentation result. ResNet101-DeepLabv3+ does not apply transposed convolution. Instead it combines convolution with bilinear upsampling.

3.2.5.2 *Dense Unet*

Henry et al. [5] developed a symmetrical Unet architecture, which is based on DenseNet [29]. DenseNet follows the finding that CNNs do not suffer the vanishing gradient problem if they have short connections from the input layer to intermediate and output layers. In DenseNet, every layer is connected with every other layer by such short connections. They successfully extended DenseNet to an Dense Unet, by using DenseNet as the encoder part of the FCN and using a decoder which, on every level of resolution, has symmetrical kernel sizes and number of filters, as well as connections as the encoder. In their work, they used the Dense Unet for road segmentation from aerial imagery. In this thesis, the Dense Unet is applied to roof-type segmentation from space-borne imagery.

4 METHODS

4.1 NETWORK ARCHITECTURE

As the network architecture in this thesis, the DeepLabv3+ [7] FCN is utilized with a single input channel and five output channels. DeepLabv3+ is a common architecture for semantic segmentation task. It lays its focus on the field of view for robust class separation. It produces segments with realistic shape, but does not focus as much on the fine-grained object localization as the DenseUnet [5] architecture, which has a deeper decoder with skip connections from all levels of resolution in the encoder. With its decoder, the DenseUnet can separate segments on a small scale, but for roof-type segmentation in an urban environment, the segments are usually bigger than roads on aerial images, where DenseUnet proved to be particularly effective for road segmentation.

4.2 LOSS FUNCTION

The loss function is the key element in this method for semantic segmentation of roof-types in complex urban environments with incomplete, imperfect and imbalanced ground truth. The combined loss function

$$\mathcal{L}_{COM} = \mathcal{L}_{SUP} + \mathcal{L}_{UNS} \tag{23}$$

is a sum of the supervised loss $\mathcal{L}_{SUP}$ and the unsupervised loss $\mathcal{L}_{UNS}$. $\mathcal{L}_{SUP}$ depends on the output of the neural network and the corresponding ground truth, whereas $\mathcal{L}_{UNS}$ depends only on the output of the neural network. Both $\mathcal{L}_{SUP}$ and $\mathcal{L}_{UNS}$ correspond to $\mathcal{L}$ in $\mathcal{L}_{SEG}$ in Equation 15. This means that they both receive a matrix of dimension $N_{bs} \times N_{cl}$ for p and y, respectively, but are than plugged into $\mathcal{L}_{SEG}$ (see Equation 15) to receive a scalar value for $\mathcal{L}_{COM}$. More details on the two pixel-wise losses are provided in the following two subsections.

4.2.1 *Supervised Loss*

The supervised loss has to handle the problems of imbalanced data and imperfect ground truth. To achieve this goal, $\mathcal{L}_{SBCE}$, $\mathcal{L}_{FOC}$ and $\mathcal{L}_{BAL}$ (see Equations 12, 13, 14) are combined, such that

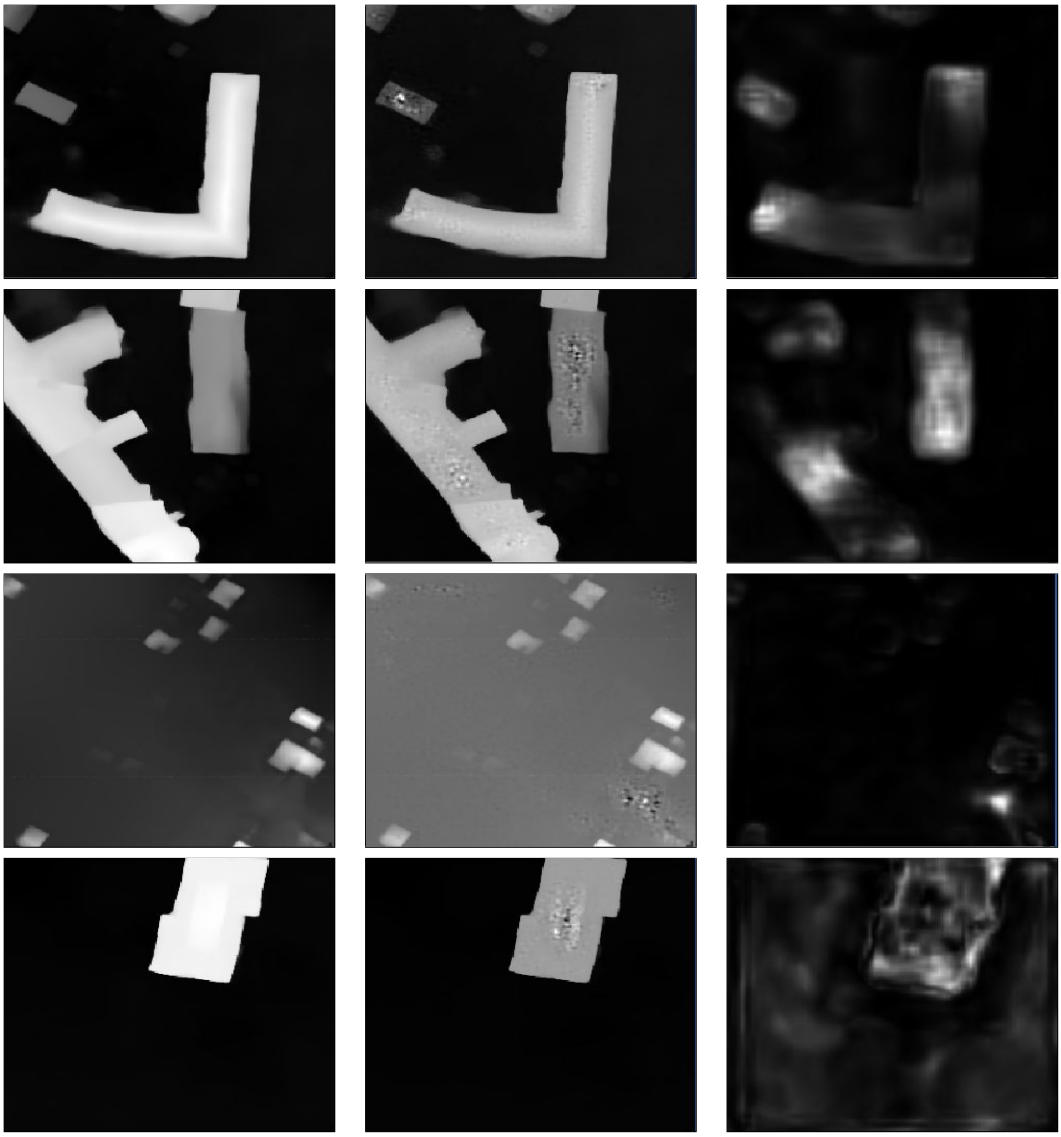
$$\mathcal{L}_{SUP}(x, y, p, \beta, \gamma) = \frac{-1}{\Sigma W} \sum_{b=1}^{N_{bs}} \sum_{i=1}^{N_{cl}} (\hat{w}_{inv})_{bi} \nu_\gamma m_\beta \log(p_{bi}), \tag{24}$$

where $\nu_\gamma = (1 - p_{bi})^\gamma$ and $m_\beta = (\beta y_{bi} + (1 - \beta) p_{bi})$. This choice of the loss includes two design choices to balances classes, which are weighting each member of the sum by $(w_{inv})_{bi}$ and making the influence of each pixel dependent on how confident the network is about its prediction p_{bi}. To better handle imperfect ground truth, the target of the loss function, which is usually the ground truth, is replaced by a combination of the ground truth and the networks predictions, to incorporate the networks knowledge. Since β controls the influence of the ground truth on the loss and the class weight depends on the class of the target, $(\hat{w}_{inv})_{bi}$ is the combination of the networks class prediction and the ground truth, weighted by β.

4.2.2 *Unsupervised Loss*

To leverage many unlabeled buildings in the given dataset, an unsupervised loss $\mathcal{L}_u$ (see Subsection 3.1.1 and Equation 5) is scaled by $\alpha > 0$ and added to $\mathcal{L}_{SUP}$ to build the final loss $\mathcal{L}_{COM}$. This is a regularization loss which smooths the output of the neural network, especially in the adversarial direction. This is accomplished by generating adversarial examples, which can be understood as an attack on the network with the purpose to change the output of the network as much as possible, by adding barely visible noise to the input. Since in this thesis, VAT is used to regularize the network, the invisibility of the noise is not stressed. In Figure 5, the effect of the perturbation in the adversarial direction is visualized. At the ends of the biggest building in the first row in column (b), the noise is the strongest. Since the triangles at the end of that building, which are visible in the first row in column (a), are characteristic for hip roofs, it makes sense that the noise r_{adv}, which represents the vulnerability of the network toward changes in the output, is strong at the triangles. In the rows two and four, the noise is located mainly at the middle of the buildings, where, possibly, the absence or presence of a ridge line would lead to a switch in output from mono-plane to gable. The locations of the noise correspond to how VAT alters the decision boundaries in the input space. It makes the network focus on the characteristic features, which can lead to a more robust FCN. Since the adversarial direction depends on the state of the parameters of

the network, it is important to say, that these findings are all in regard of a trained network. The adversarial noise of a network, which is initialized by random parameters, before training, is not meaningful, because the network outputs are random.



(a) x **(b)** $x + r_{adv}$ **(c)** $\mathcal{L}_u(x)$

Figure 5: A visualization of VAT for roof-type segmentation. In column **(a)**, the input height image is shown in gray-scale visualization. In column **(b)**, the perturbed version of the input, which alters the output of the network most, is visualized. In column **(c)**, the pixel-wise output of $\mathcal{L}_u$ is shown. In the first, second and fourth row, the noise r_{adv} is strongest on some buildings. In the third row, r_{adv} is very strong in the bottom right corner. The loss $\mathcal{L}_u$ is the highest on some parts of the roofs in the first, second and fourth row and has its maximum in the bottom right corner in the third row.

5 EXPERIMENTS

5.1 DATA PREPARATION

5.1.1 *Ground Truth*

As the ground truth, a roof-type map was extracted from CityGML data of Berlin, Germany. The roof-types included in the CityGML data are flat, monopitch, skip pent, gable, hip, half-hip, mansard, pavilion, cone, copula, shed, arch, pyramid and complex. Since most of these roof-types occur only very rare, the number of classes was reduced. The class shed, which represents a roof that is flat but has an inclination can hardly be distinguished from the flat roof and is therefore merged with the flat class into the mono-plane class. The classes hip and half-hip are merged into the hip class, gable is another class and all other roof-types are merged into the other/complex class. There are two more types of pixels, which are background and unknown. The background pixels correspond to pixels where no building is located and the unknown ones represent the lack of information of the roof-type in the CityGML data. The resolution of the ground truth map is matched to the 0.3 m of the input data from WV-4. In Figure 6, a small crop from the ground truth is visualized. Besides the background, unknown roofs are dominant in this Figure. The same is true for the complete ground truth. The area for quantitative and qualitative evaluation in the experiments is visualized in Figure 8. In this area, the ground truth was manually corrected and completed, such that the metrics evaluated over this area are an accurate measure of the performance of the different experiments in Section 5.2. This area was selected to make sure that each of the five classes background, mono-plane, gable, hip and other is included with a not too small number of buildings. The corresponding distribution of pixels per class is given in Table 4. Another part of the data is held back during training for validation. This area was selected to be large enough to contain buildings of all considered classes with some variety and the distribution of pixels in this area is given in Table 5.

5.1.2 *Input Images*

To represent building roofs, such that they can be discriminated well from each other, a refined DSM is produced. First, a DSM is created by semi-global matching of multiple WV-4, panchromatic images with a Ground Sampling

Distance (GSD) of 0.3 m. But in Figure 9 **(a)**, the quality of the stereo DSM is not good enough to predict the roof-type, since the building in that image could be confused with a mono-plane or gable roof. The 3D model in Figure 9 **(c)** represents the geometry of a hip roof much better. But the CityGML data is usually not available for any arbitrary city or region in the world, whereas the stereo DSM can easily be acquired from space-borne imagery. So, instead of using either the stereo or the Level of Detail (LOD) DSM, a refinement process is carried out on the input area. This step is done by the method of Bittner et al. [30], where a Generative Adversarial Network (GAN) is trained to refine a stereo DSM, using the LOD as the ground truth. The training of the GAN is performed on a dataset which has no overlap with the dataset, that is used for training, validation or testing in roof type segmentation in this thesis. This separation is necessary, since otherwise the segmentation results might not be as good on previously unseen data, where the refinement step is not training on that data. The refined DSM in Figure 9 **(b)** is smoother than the stereo DSM, does not include vegetation and does not have the hole of the stereo DSM. But still, the geometry is not a clearly visible as in the LOD and a strong bump replaces the hole. To get more information about the refinement process, the reader is referred to Bittner et al. [30]. The patches are generated by first selecting a random index on a grid, and then cropping a window of height and width of 256 pixels each. This width and height corresponds to almost 77 m, such that each covers an area of around 5900 m^2. This is large enough to contain most of the buildings completely. The patch size together with the random shifting of the window to extract the patch, makes sure that during training, most of the buildings are seen completely by the network. The patches are normalized to the interval $[-1, 1]$ using the minimum and maximum value of each single patch. This makes the values symmetric about zero and independent of absolute height values, which leads to a greater generalization capacity of the method, since the absolute height values of Berlin do not apply to e. g. Munich, Germany, which lays several hundred meters higher in altitude. During testing, the height and width of the input was not changed, but the cropping of the patches was done by overlapping neighboring patches by 128 pixels.

5.1.3 *Data Configurations*

For the different experiments, different splittings and degrees of annotations are used. In Figure 10, the splitting is visualized and the subareas are given the names TRAIN1, TRAIN2, TRAIN3, TRAIN4, TRAIN5, VALIDATION and TEST.

5.2 COMPARISON OF APPROACHES

To evaluate the methods studied in this thesis, several experiments are performed. For all experiments, training is done for 200 epochs, where for the first 100 epochs the learning rate is constantly 2×10^{-4} and for the last 100 epoch, the learning rate is decayed linearly to 0. The optimizer Adam (see Subsection 3.2.2) is used with the momentum parameters 0.5 and 0.999. These hyper-parameters where chosen by informal search. Since the available data is limited to a small area, data augmentation is used to increase the variety of the input data. To augment the data, random vertical and horizontal shifting is applied, each with a probability of 50%. Also, the window to crop a patch from satellite image is shifted to the right and down by a random number between 0 and the width and height of the patch. After every 10000 patches during training, that is 313 optimization steps, validation is done on the validation data, which is neither included in the training data, nor the test data. The validation is not used for early stopping, but to monitor the development of the generalization during training, based on one metric commonly for all experiments, regardless of the different loss functions applied during training, to increase comparability of the different losses. During validation and testing pixels of unknown roof-types are ignored. In the training, unlabeled pixels are also ignored in the supervised loss. In the test phase, the network output is averaged on overlapping areas of neighboring patches and the final class is determined by the argmax of the logits.

5.2.1 *Class-Balancing*

In Table 1, the distribution of the pixels over the classes of the data configuration *WV4* shows that by far most of the pixels belong to the background class. Less than 1 % of the pixels belong to the gable or hip classes. That means that in training, the model is presented some classes orders of magnitudes more often and is therefore prone to ignoring the gable and hip class. In the first step of the experimental evaluation, the focus lays on finding a balancing strategy, which leads to results, which have good quality for each of the classes, not only for the strongly represented classes.

5.2.1.1 *Balancing with different Loss Functions*

As the baseline DLv3+_CE for this batch of experiments, a DeepLabv3+ (see Subsubsection 3.2.5.1) is trained without any balancing strategy, using the semantic segmentation CE loss (see Equations 11, 15). Further experiments are done on the focal loss (see Equation 13), dice loss (see Equation 16) and class weights in the dice, focal and CE loss. For these experiments, the FCN is DeepLabv3+ and the batch size was chosen to be

32, since a higher number of pixels in each batch better represents the distribution of pixels over the classes in the training data. The configuration of the other hyper-parameters can be found in Table 6, specifically the rows DLv3+_CE, DLv3+_CEInvW, DLv3+_CEIRootW, DLv3+_CESqrW, DLv3+_FOC05InvW, DLv3+_FOC10InvW, DLv3+_FOC20InvW, DLv3+_DICE, DLv3+_DICEInvW, DLv3+_DICERootW and DLv3+_DICESqrW.

5.2.1.2 *Balancing by Annotation*

To evaluate the influence of more annotations of the weak classes, an additional area, which has already many annotated buildings of the strong classes, is improved by manually labeling some of the gable and hip roofs. Also, some buildings of the classes gable and hip, which where unlabeled before, but lay in the area of the original training data, are annotated. The new, improved distribution of the classes in the training data can be seen in Table 2 (compare with Table 1) and is visualized in Figure 7 (compare with Figure 6). The ratio of gable and hip labels is increased from 0.26% to 0.40% and 0.07% to 0.23%, respectively and visually there are many more hip and gable roofs in the improved ground truth (see Figures 6, 7). With this better balanced data, a DeepLabv3+ is trained with batch size 32. The experiment DLv3+_CE_impr differs from DLv3+_CE only in the data configuration, which is $WV4_{\text{impr}}$ instead of $WV4$.

5.2.2 *DeepLabv3+ vs. DenseUnet*

In the next step, DeepLabv3+ is compared to the DenseUnet (see Subsubsection 3.2.5.2). The two architecture focus on different aspects of the semantic segmentation. DeepLabv3+ dedicates most of his parameters to get a high semantic resolution in the bottleneck, whereas DenseUnet focuses more on the spatial resolution of the segments, by incorporating much more higher resolution features in the decoder. For this experiments, the batch size was chosen to be 32 for the DeepLabv3+ and 16 for the DenseUnet, since the DenseUnet has higher memory requirements during training. The loss function is chosen to be the focal loss with $\gamma = 1$ and inverse frequency weights are applied. For this comparison, the improved ground truth ($WV4_{\text{impr}}$, see Table 2) is used. The other hyperparameters of the experiments DLv3+_FOC10InvW_impr and DUN_FOC10InvW_impr are given in Table 6.

5.2.3 *Imperfect Ground Truth*

To evaluate how a bootstrapped loss can lead to results which are less dependent on the noise in the training data, the soft-bootstrapped

focal loss $\mathcal{L}_{SBFOC}$ with inverse frequency weights is used as the optimization target with five different values $\{0.5, 0.7, 0.8, 0.9, 0.99\}$ for β, $\gamma = 1$ and the results are compared for these values. The lower the value of the β with the best results, the lower the quality of the ground truth is. This experiment is performed on the improved ground truth $WV4_{impr}$ with the other hyper-parameters as in the rows DLv3+_SB99FOC10InvW_impr, DLv3+_SB90FOC10InvW_impr, DLv3+_SB80FOC10InvW_impr, DLv3+_SB70FOC10InvW_impr and DLv3+_SB50FOC10InvW_impr of Table 6.

5.2.4 *Unlabeled Data*

Since many of the buildings in the ground truth are still unlabeled, another experiment is performed by applying VAT as in Subsection 4.2.2. First, a baseline, DLv3+_SB99FOC10InvW_impr2, is trained on $WV4_{unl}$ with the same loss and hyperparameters as DLv3+_SB99FOC10InvW_impr, with batch size 16. With the experiments DLv3+_SB99FOC10InvW_impr2_vat, DLv3+_SB99FOC10InvW_impr2_vat2, DLv3+_SB99FOC10InvW_impr2_vat3 and DLv3+_SB99FOC10InvW_impr2_vat4, the effect of the integration of unlabeled data is investigated. The batch size for the labeled batch is 16 and also 16 for the unlabeled batch, which is used to compute the virtual adversarial segmentation loss. The labeled and the unlabeled batch are different from each other and are sampled according to the data configuration $WV4_{unl}$. Since the unlabeled portion of $WV4_{unl}$ is about twice as large as the labeled portion, the model will go through twice as many iterations of the full labeled data as of the unlabeled data. The hyper-parameters of the supervised loss are $\gamma = 1$ and $\mathtt{beta} = 0.99$. For the unsupervised part of the loss, $\mathcal{L}_u$, $\epsilon = 10$, $\xi = 10$ and α is in $\{1, 10, 100, 500\}$.

5.3 METRICS

For the quantitative evaluation in this thesis, several metric are used. It is important, that not a single metric can be the only one used for quantitative evaluation, since different metrics describe different properties of the results. True positive TP_i, is the number of samples of class i which are also classified to class i, true negative TN_i is the number of samples of any class but i and are not classified as class i, false positive FP_i is the number of samples of any class but i but are classified as class i and false negative FN_i is the number of samples of class i which are not classified as i. The first metric used is the accuracy

$$acc = \frac{\sum_{i=1}^{N_{cl}} (TP_i + TN_i)}{N_{test}}, \tag{25}$$

where N_{cl} is the number of classes of the task and N_{test} is the number of pixels in the test set. But the accuracy is not suitable to judge the success of testing on imbalanced data, because a class that is big enough can reward a model which predicts only this class for all pixels. For instance, a class which contains 99% of the pixels, gives such a model an accuracy of 99%. The precision

$$prec_i = \frac{TP_i}{TP_i + FP_i}, \tag{26}$$

recall

$$rec_i = \frac{TP_i}{TP_i + FN_i}, \tag{27}$$

specifity

$$spec_i = \frac{TN_i}{TN_i + FP_i}, \tag{28}$$

F_1-score

$$(F_1)_i = 2 \times \frac{prec_i \times rec_i}{prec_i + rec_i} \tag{29}$$

and balanced accuracy

$$(acc_b)_i = \frac{spec_i + rec_i}{2} \tag{30}$$

are computed as vectors with one component for each class and are common metrics for semantic segmentation. They do not have the downside of the accuracy in terms of imbalanced test data. The precision of class i, $prec_i$, shows how well the model avoids producing false positives, whereas the recall of class i, rec_i shows, how well the model avoids false negatives. The specificity of class i, $spec_i$, similar as the precision, punishes any false positives, but favors true negatives over true positives. The F_1-score $(F_1)_i$ and the balanced accuracy $(acc_b)_i$ each consider two of precision, recall and negativity. The F_1-score is the harmonic mean of precision and recall and is therefore drawn towards the minimum of $prec_i, rec_i$, what makes this metric a strong indicator for a balanced model. The balanced accuracy $(acc_b)_i$ is the arithmetic mean of specificity and recall, such that it increases with the amount of true positives and true negatives, but is not as strongly influenced by a single one of the false positives and false negatives being large, when compared to the F_1-score. To judge the quantitative result of each experiment, the arithmetic mean over the classes F_1 of $(F_1)_i$ and acc_b of $(acc_b)_i$ is given.

	background	flat	gable	hip	complex/other	unknown	Σ
ABS	670528308	62811577	2253788	581249	30368636	89089922	855633480
REL	0,7837	0,0734	0,0026	0,0007	0,0355	0,1041	1,0000

Table 1: The distribution of the pixels per class in the training data (data configuration *WV4*). In row ABS, the number of pixels belonging to each of the 5 classes and the unknown number of pixels is given. In the row REL, the ratio of pixels belonging to each class is given. In the column Σ, the sum of each row is given.

	background	flat	gable	hip	complex/other	unknown	Σ
ABS	728876721	67017759	3644476	2160572	32103489	88792664	922595681
REL	0,7900	0,0726	0,0040	0,0023	0,0348	0,0962	1,0000

Table 2: The distribution of the pixels per class in the training data with additional annotated gable and hip roofs (data configuration $WV4_{\mathtt{impr}}$). In row ABS, the number of pixels belonging to each of the 5 classes and the unknown number of pixels is given. In the row REL, the ratio of pixels belonging to each class is given. In the column Σ, the sum of each row is given.

Sub-Areas		background	flat	gable	hip	complex/other	unknown	Σ
TRAIN	ABS	214980240	15483050	3303708	1995147	5929601	19570455	261262201
1,5	REL	0,8229	0,0593	0,0126	0,0076	0,0227	0,0749	1
TRAIN	ABS	0.07490733	48450713	3256524	1594476	23274881	64456563	661333480
2,3,4	REL	0,7867	0,0733	0,0049	0,0024	0,0352	0,0975	1
ALL	ABS	735280563	63933763	6560232	3589623	29204482	84027018	922595681
	REL	0,797	0,0693	0,0071	0,0039	0,0317	0,0911	1

Table 3: The distribution of the pixels per class in the training data configuration $WV4_{\mathtt{unl}}$. In row ABS, the number of pixels belonging to each of the 5 classes and the unknown number of pixels is given. In the row REL, the ratio of pixels belonging to each class is given. In the column Σ, the sum of each row is given.

	background	flat	gable	hip	complex/other	unknown	Σ
ABS	4788047	1269177	433769	87877	966609	5961	7551440
REL	0,6341	0,1681	0,0574	0,0116	0,1280	0,0008	1,0000

Table 4: The distribution of the pixels per class in the test data. In row ABS, the number of pixels belonging to each of the 5 classes and the unknown number of pixels is given. In the row REL, the ratio of pixels belonging to each class is given. In the column Σ, the sum of each row is given.

	background	flat	gable	hip	complex/other	unknown	Σ
ABS	46896205	10187112	511995	58895	5339230	6084643	69078080
REL	0,6789	0,1475	0,0074	0,0009	0,0773	0,0880	1,0000

Table 5: The distribution of the pixels per class in the validation data. In row ABS, the number of pixels belonging to each of the 5 classes and the unknown number of pixels is given. In the row REL, the ratio of pixels belonging to each class is given. In the column Σ, the sum of each row is given.

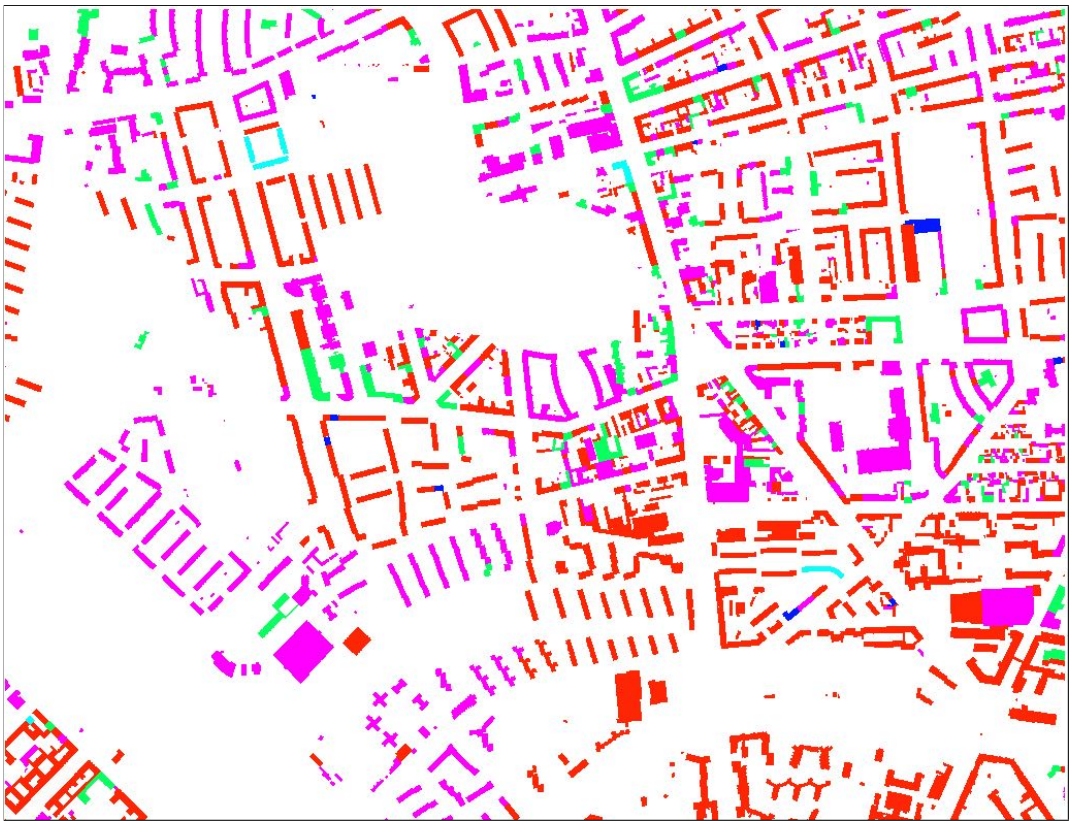


Figure 6: A small area of the ground truth. **White:** background, **magenta:** mono-plane, **blue:** gable, **cyan:** hip, **green:** other, **red:** unknown.

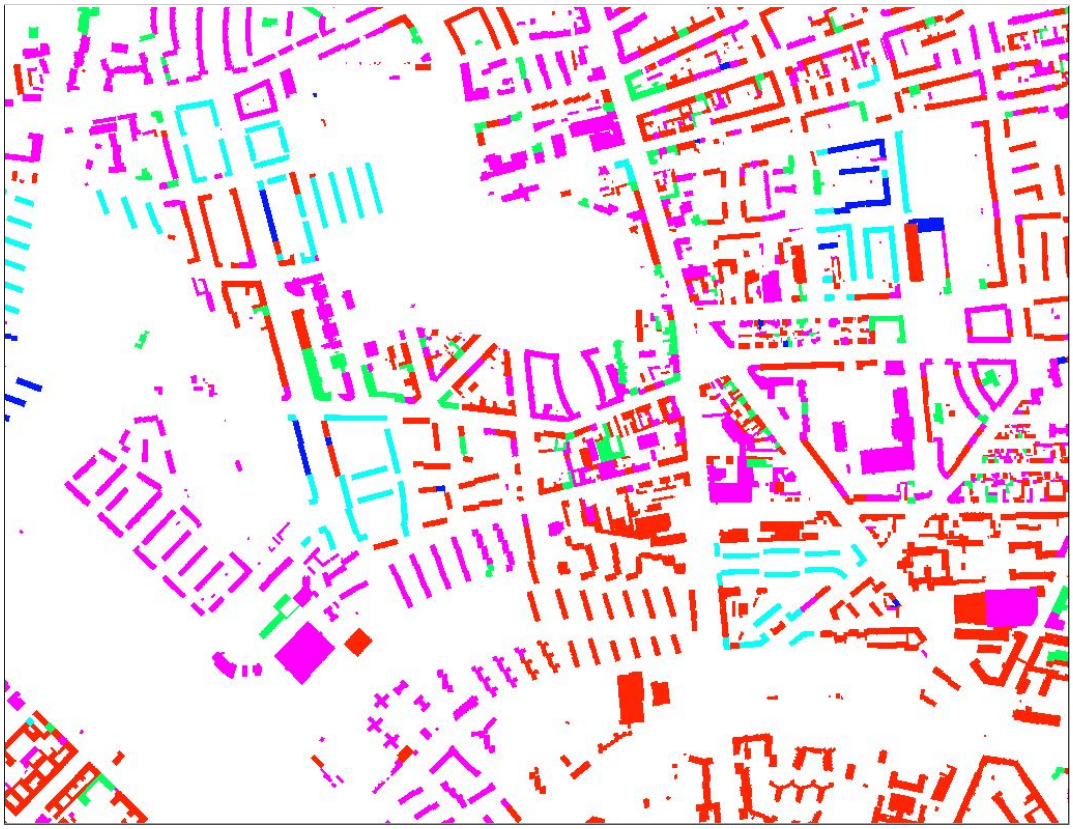
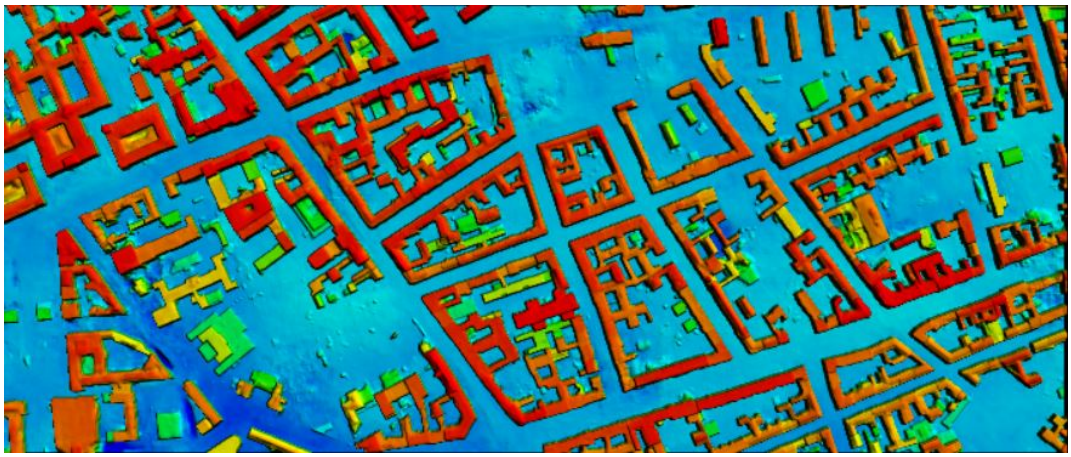


Figure 7: A small area of the ground truth with additionally labeled gable and hip roofs. **White:** background, **magenta:** mono-plane, **blue:** gable, **cyan:** hip, **green:** other, **red:** unknown.

Name	Data	Arch	BS	BS_unl	Weights	Loss	Gamma	Beta	Vat	Alpha	Instance Masking
DLv3+_CE	$WV4$	DeepLabv3+	32	-	-	$\mathcal{L}_{CE}$	-	-	No	-	No
DLv3+_CEInvW	$WV4$	DeepLabv3+	32	-	w_{inv}	$\mathcal{L}_{CE}$	-	-	No	-	No
DLv3+_CERootW	$WV4$	DeepLabv3+	32	-	w_{root}	$\mathcal{L}_{CE}$	-	-	No	-	No
DLv3+_CESqrW	$WV4$	DeepLabv3+	32	-	w_{sqr}	$\mathcal{L}_{CE}$	-	-	No	-	No
DLv3+_FOC05InvW	$WV4$	DeepLabv3+	32	-	w_{inv}	$\mathcal{L}_{FOC}$	0.5	-	No	-	No
DLv3+_FOC10InvW	$WV4$	DeepLabv3+	32	-	w_{inv}	$\mathcal{L}_{FOC}$	1.0	-	No	-	No
DLv3+_FOC20InvW	$WV4$	DeepLabv3+	32	-	w_{inv}	$\mathcal{L}_{FOC}$	2.0	-	No	-	No
DLv3+_DICE	$WV4$	DeepLabv3+	32	-	-	$\mathcal{L}_{DICE}$	-	-	No	-	No
DLv3+_DICEInvW	$WV4$	DeepLabv3+	32	-	w_{inv}	$\mathcal{L}_{DICE}$	-	-	No	-	No
DLv3+_DICERootW	$WV4$	DeepLabv3+	32	-	w_{root}	$\mathcal{L}_{DICE}$	-	-	No	-	No
DLv3+_DICESqrW	$WV4$	DeepLabv3+	32	-	w_{sqr}	$\mathcal{L}_{DICE}$	-	-	No	-	No
DLv3+_CE_impr	$WV4_{impr}$	DeepLabv3+	32	-	-	$\mathcal{L}_{CE}$	-	-	No	-	No
DLv3+_FOCE10InvW_improved	$WV4_{impr}$	DeepLabv3+	32	-	w_{inv}	$\mathcal{L}_{FOC}$	1.0	-	No	-	No
DUN_FOC10InvW_imroved	$WV4_{impr}$	DenseUnet	32	-	w_{inv}	$\mathcal{L}_{FOC}$	1.0	-	No	-	No
DLv3+_SB99FOC10InvW_improved	$WV4_{impr}$	DeepLabv3+	32	-	w_{inv}	$\mathcal{L}_{SBFOC}$	1.0	0.99	No	-	No
DLv3+_SB90FOC10InvW_improved	$WV4_{impr}$	DeepLabv3+	32	-	w_{inv}	$\mathcal{L}_{SBFOC}$	1.0	0.9	No	-	No
DLv3+_SB80FOC10InvW_improved	$WV4_{impr}$	DeepLabv3+	32	-	w_{inv}	$\mathcal{L}_{SBFOC}$	1.0	0.8	No	-	No
DLv3+_SB70FOC10InvW_improved	$WV4_{impr}$	DeepLabv3+	32	-	w_{inv}	$\mathcal{L}_{SBFOC}$	1.0	0.7	No	-	No
DLv3+_SB50FOC10InvW_improved	$WV4_{impr}$	DeepLabv3+	32	-	w_{inv}	$\mathcal{L}_{SBFOC}$	1.0	0.5	No	-	No
DLv3+_SB99FOC10InvW_impr2	$WV4_{unl}$	DeepLabv3+	16	16	w_{inv}	$\mathcal{L}_{SBFOC}$	1.0	0.99	Yes	1	No
DLv3+_SB99FOC10InvW_impr2_vat1	$WV4_{unl}$	DeepLabv3+	16	16	w_{inv}	$\mathcal{L}_{SBFOC}$	1.0	0.99	Yes	1	No
DLv3+_SB99FOC10InvW_impr2_vat2	$WV4_{unl}$	DeepLabv3+	16	16	w_{inv}	$\mathcal{L}_{SBFOC}$	1.0	0.99	Yes	10	No
DLv3+_SB99FOC10InvW_impr2_vat3	$WV4_{unl}$	DeepLabv3+	16	16	w_{inv}	$\mathcal{L}_{SBFOC}$	1.0	0.99	Yes	100	No
DLv3+_SB99FOC10InvW_impr_vat4	$WV4_{unl}$	DeepLabv3+	16	16	w_{inv}	$\mathcal{L}_{SBFOC}$	1.0	0.99	Yes	500	No

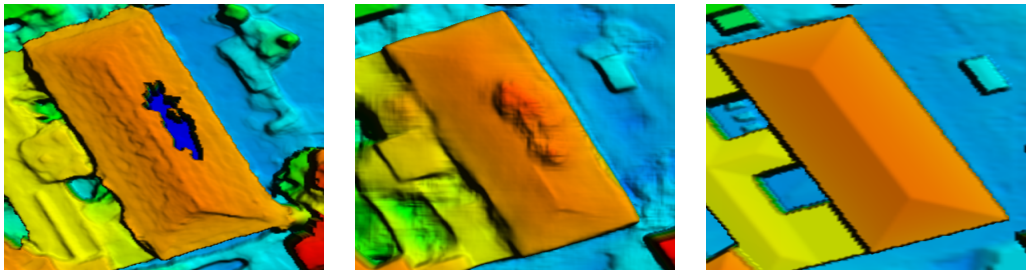
Table 6: The table with the configuration of the experiments in this thesis. The other hyperparameters, which do not vary across experiments are given in Section 5.2. The column Loss refers to the $\mathcal{L}$ in Equation 15, since all the experiments are on semantic segmentation.

(a) Ground Truth Testarea



(b) Refined DSM Testarea

Figure 8: In this figure, the testarea for quantitative and qualitative evaluation in Section 5.2 is visualized. In **(a)**, the ground truth for this area can be seen. **White:** background, **magenta:** mono-plane, **blue:** gable, **cyan:** hip, **green:** other, **red:** unknown. In **(b)**, the refined DSM is visualized with colors ranging from dark-blue to green to red, with red as the highest value.



(a) stereo **(b)** refined **(c)** lod

Figure 9: A comparison of three different DSMs from the same scene. In **(a)**, a hole is visible on the roof, which stems from the semi-global matching. In **(b)**, the refined version does not have the hole, but an artefact is visible in the area of the hole. In **(c)**, the planes have neither artefacts nor holes. The main structure of the roof geometry is visible in all three images, but much clearer in the LOD, as compared to the stereo and refined DSM.

Configuration	Sub-Areas Labeled	Sub-Areas Unlabeled	Level of Annotation
$WV4$	TRAIN1, 2, 3, 4	-	1
$WV4_{\text{impr}}$	TRAIN1, 2, 3, 4, 5	-	2
$WV4_{\text{unl}}$	TRAIN1, 5	TRAIN2, 3, 4	3

Table 7: Three different configurations of the training data, each including a subset of {TRAIN1, TRAIN2, TRAIN3, TRAIN4, TRAIN5}. The level of annotation indicates how much extra effort was invested to label buildings. In level of annotation 1, no more labeled buildings are included than in the original CityGML data of Berlin. In level of annotation 2, additional buildings are labeled, based on level of annotation 1. In level of annotation 3, additional buildings are labeled, based on level of annotation 2. See Figure 10 for a visualization of the locations of the different areas.

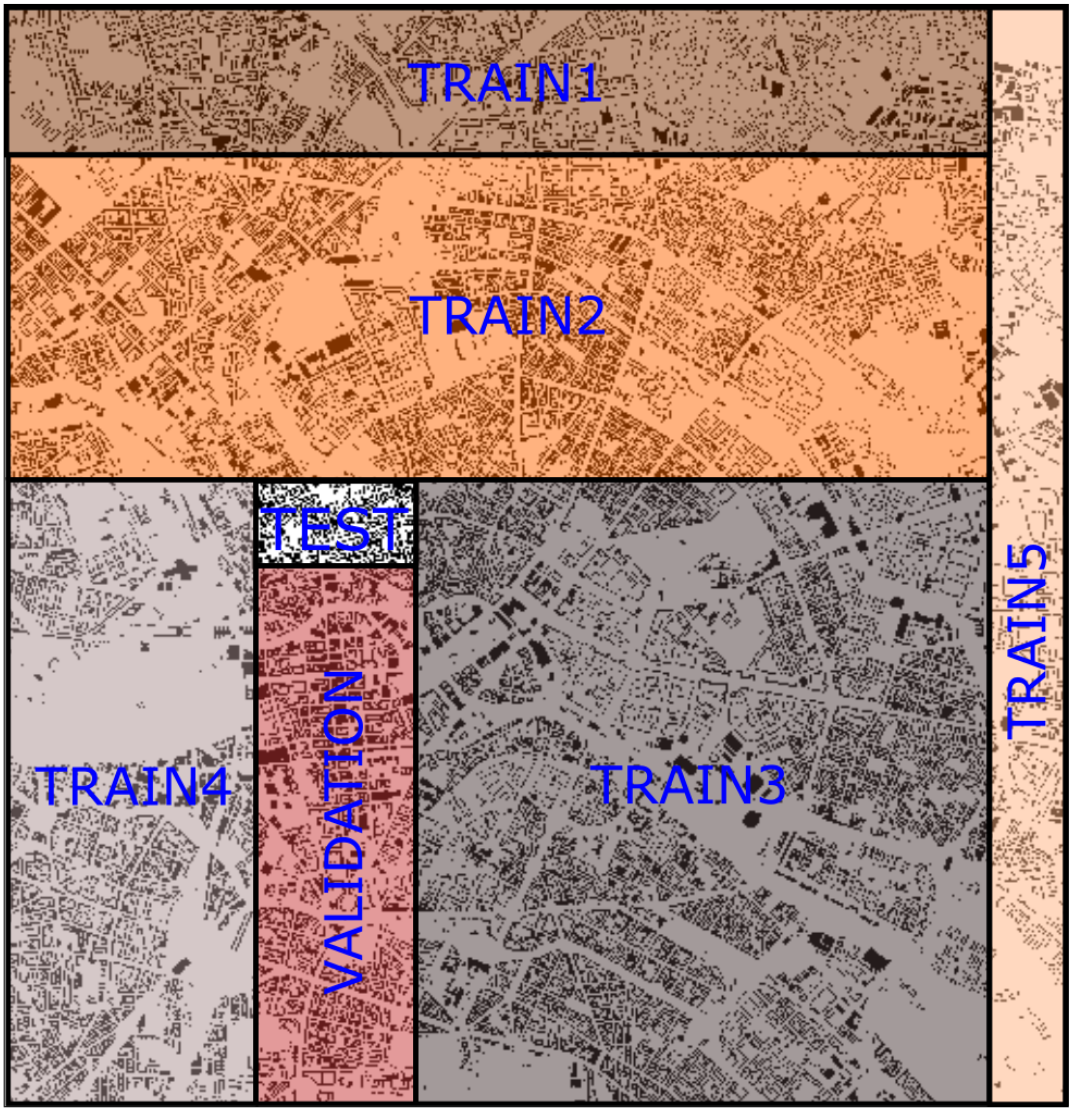


Figure 10: The study area is split into seven sub-areas. These areas are five areas for training, one area for validation and one area for testing. In this figure, the splitting is visualized schematically.

6 RESULTS & DISCUSSION

6.1 CLASS-BALANCING

In Subsection 5.2.1, twelve experiments are described. The values of the metrics (see Section 5.3) for these experiments are given in Table 8 and a visual comparison of the results is presented in Figure 11. The baseline experiment DLv3+_CE achieves an $(F_1)_i$ of 1.3 and 2.1 for the minority classes like gable and hip. In Figure 11 (a), gable and hip are assigned almost nowhere, despite these classes are clearly visible in the ground truth in Figure 11 (m). This confirms the assumption, that the CE loss is not suitable to train an FCN for roof-type segmentation on the five selected classes on a data set with a class distribution as in Table 1. Among the three experiments DLv3+_CEInvW, DLv3+_CERootW and DLv3+_CESqrW, DLv3+_CEInvW achieves the best performance quantitatively in Table 8 and looks most similar to the ground truth in Figure 11. The rec_i values for gable and hip are much lower in DLv3+_CERootW than in DLv3+_CERInvW and much higher in DLv3+_CESqrW than in DLv3+_CEInvW. Also, $prec_i$ for gable and hip is much lower in DLv3+_CESqrW than in DLv3+_CEInvW. This shows, that class weights $(w_{root})_{bi}$ do not favor the underrepresented classes enough and class weights $(w_{sqr})_{bi}$ strongly overemphasize the underrepresented classes, which leads to many false positives for the minority classes.

The visual results of the experiments DLv3+_FOC05InvW, DLv3+_FOC10InvW and DLv3+_FOC20InvW in Figures 11 (e), (f) and (g) look very similar in many places. But quantitatively, DLv3+_FOC10InvW has the highest acc_b and F_1 scores of the three. The comparison of the results on the hip roof in the bottom right of the Figures 11 (e), (f), (g) and (m) shows, that in (f) and (g) the roof is segmented with gable (blue) and other (green) in (e), whereas this roof is a hip roof in the ground truth in (m). Since hip roofs have a section of a ridge line (compare Figure 1), they are more similar to gable than to any other class. Therefore, the segmentation as gable is favored over any other segmentation. This indicates that the variation of γ in DLv3+_FOC05InvW, DLv3+_FOC10InvW and DLv3+_FOC20InvW has an impact on the overall performance. Comparing DLv3+_FOC10InvW (Figure 11 (f)) and DLv3+_InvW (Figure 11 (b)) visually, the segments of DLv3+_InvW look slightly more homogeneous, but DLv3+_InvW has better average $(F_1)_i$ scores for the minority classes. Visually, when looking at almost square building below the three parallel hip roofs at the top middle position, DLv3+_FOC10InvW classifies most

of this buildings pixels as gable, which matches the ground truth and DLv3+_CEInvW segments the roof as mono-plane. Therefore, the focal loss leads to more balanced results than the CE loss.

Different than for the experiments with the CE loss (DLv3+_CE, DLv3+_CEInvW, DLv3+_CERootW and DLv3+_CESqrW), for the experiments with the DICE loss (DLv3+_DICE, DLv3+_DICEInvW, DLv3+_DICERootW and DLv3+_DICESqrW), the application of class weights to the class-wise outputs of the DICE loss leads to a degradation in performance, both quantitatively (see Table 8) and qualitatively (see Figures 11 (h), (i), (j), (k)). It is also apparent, that in the visual output of DLv3+_DICE, many buildings are classified as background, which is also true for DLv3+_DICERootW. Comparing the results of the experiments with the DICE loss to DLv3+_FOC10InvW, it shows, that the focal loss is more suitable to balance classes than the DICE loss.

Comparing the results of the experiments DLv3+_CE and DLv3+_CE_impr (compare Table 8 and Figures 11 (a) and (l), the results of DLv3+_CE_impr has a higher mean balanced accuracy acc_b and a much higher mean F_1. The same is true for the minority classes. Especially the recall of the hip class is increased from 1.1 to 23.3. This also expresses in the visual results in Figure 11 (l), where the three parallel hip roofs in the top middle position are correctly segmented. In Figure 11 (a), the hip class does not show. These results indicate that the additionally labeled hip roof in the data configuration $WV4_{impr}$ lead to an improvement in the balancing of the output distribution of an FCN trained with the cross entropy loss without balancing strategies. Although the results of e. g. DLv3+_FOC10InvW have better mean and per-class metrics, the three parallel hip roofs in the top middle position are less over-smoothed in Figure 11 (l) than in Figure 11 (f). This might be caused by the weights applied to each class, which lay in different orders of magnitude and make the training more difficult. Therefore, improving the ground truth by balancing it is the strategy which ultimately leads to the best overall results.

6.2 DEEPLABV3+ VS. DENSEUNET

Quantitatively, the masks generated by the DeepLabv3+ as trained in DLv3+_FOC10InvW_impr and the DenseUnet as trained in DUN_FOC10InvW_impr are very similar, but DLv3+_FOC10InvW_impr has a significant advantage in the F_1 score for the hip class, which exceeds the one of DUN_FOC10InvW_impr by 11% (compare Table 9). Visually, the roof-type segments in Figure 12 (b) are over-smoothed when compared to those in the ground truth in Figure 12 (a), but are more consistent, that is, more homogeneous. In Figure 12 (c) the shapes of the roof segments look more similar to those in the ground truth, but are less

homogeneous. The differences in the level of homogeneity and accuracy of roof boundaries between the two networks most likely stem from their different decoding strategy. The DenseUnet arcitecture (see Subsubsection 3.2.5.2), uses feature maps from each level of resolution in the encoder to incorporate spatial information into the decoder, which leads to a very detailed separation of segments, for both the separation of background and roof and the separation of different roof-types. The DeepLabv3+ architecture (compare Subsubsection 3.2.5.1) only uses one connection from feature maps with low spatial, but high semantic resolution in the end of the encoder to improve feature localization. Also, DeepLabv3+ uses atrous spatial pyramid pooling to increase the field of view, which leads to consistent roof segments, as the network incorporates more of the surrounding pixels to classify each pixel. For roof-type segmentation, the consistency of the segments is more important, since the outline can be improved by post-processing using building footprints.

6.3 IMPERFECT GROUND TRUTH

In the CityGML data of Berlin, the roof-type is assigned incorrectly in many places. This noise makes it harder for an FCN to learn to segment roof-types. Next to balancing the roof-type imbalance, the loss $\mathcal{L}_{SUP}$ (see Equation 24) takes into account, that the training is robust to inconsistencies to a degree, and replaces the ground truth by a combination of the ground truth and the networks predictions. In Table 10, the quantitative results for different values of β in $\mathcal{L}_{SUP}$ are given. For $\beta = 0.99$ and $\beta = 0.9$, the mean F_1 score is the highest at 58.5%. The mean acc_b is the highest for $\beta = 0.99$. The visual comparison of the results of DLv3+_SB99FOC10InvW_impr, DLv3+_SB90FOC10InvW_impr, DLv3+_SB80FOC10InvW_impr, DLv3+_SB70FOC10InvW_impr and DLv3+_SB50FOC10InvW_impr in Figures 13 (b), (c), (d), (e) and (f) all look very similar (INCLUDE PLACES WHERE THEY LOOK SIMILAR) but as β increases, the gable roofs in Figure 13 (a) are detected less often, which matches the decreasing value of rec_i for the gable class. These results suggest that the bootstrapping may negatively effect the class balancing. If the network sees too few gable and hip roofs at the beginning of the training the bootstrapped class weight $\hat{w}_{inv}$ in Equation 24 is biased toward the more frequent mono-plane and other class, because the networks predictions do not include the gable and hip class. But comparing DLv3+_SB99FOC10InvW_impr to DLv3+_FOC10InvW_impr, the bootstrapped version of the focal loss achieves higher acc_b, F_1 and $(F_1)_i$ for the gable and hip class. This shows, that leveraging the networks intrinsic robustness to noise in the ground truth improves the generalization capacity of the DeepLabv3+.

6.4 UNLABELED DATA

The quantitative results for the experiments on VAT for roof-type segmentation (see Subsection 5.2.4) are presented in Table 11. In the experiment DLv3+_SB99FOC10InvW_impr2_vat2, the influence of $\mathcal{L}_u$ was scaled by $\alpha = 10$. This leads to an improvement of 3.2% in F_1 over the fully supervised DLv3+_SB99FOC10InvW_impr2, which corresponds to $\alpha = 0$. The boost in performance by adding an unsupervised loss is not surprising, since this regularization term allows the DeepLabv3+ to learn features from data it would not see in the fully supervised setting. For $\alpha > 10$, the F_1 becomes even lower than for $\alpha = 0$. This indicates, that if the influence of $\mathcal{L}_u$ is to high, the learning of $\mathcal{L}_{SUP}$ is degraded. Among the experiments with $\alpha > 0$, $\alpha = 10$ has the highest acc_b and F_1. Next, the results of the experiment DLv3+_SB99FOC10InvW_impr2_vat2 are set in context with the corresponding pairs of input and ground truth.

EXAMPLE 1 Example 1 is visualized in Figure 14. The comparison in the Figure shows that, due to the availability of the most important features in the input, the network has learned to predict the hip class correctly. The Figure shows a building without other buildings in direct neighborhood, which might make it easier for the network to predict the hip class correctly, since hip roofs are often without direct neighborhood in the training data (compare Figure 7).

EXAMPLE 2 In Figure 15, a hip roof is visualized. The ridge line is clearly visible in the input and the prediction of the network is correct and matches with the ground truth. This example shows, that the network has learned to recognize a gable roof, even if the inclination of the planes is low.

EXAMPLE 3 Figure 16 shows an example in the test area, where the CityGML data disagrees with the RGB and the input. The input to the network features multiple different mono-plane roofs, which pixels are all predicted to be mono-plane. But since there are multiple mono-plane roofs, the building is entirely annotated as other. Two possible, non-excluding interpretations for the network's predictions are **i.** that each individual mono-plane roof was segmented correctly or **ii.** that the division into the classes and therefore the modeling of the problem is not optimal.

EXAMPLE 4 Figure 17 shows another place where the network correctly identifies the gable class. In Example 4, gable roofs with complex features or their respective surfaces are included and the network robustly segments these roofs as gable.

EXAMPLE 5 In Figure 18, a hip roof without direct neighbors is correctly segmented. The resultant segment is much smoother than both the ground truth and the input, which is due to the specific decoder of the DeepLabv3+, which does not incorporate spatial information from early layers in the encoder.

EXAMPLE 6 Example 6 is visualized in Figure 19. The Figure shows a building with multiple mono-planes and that the network predicts a mix of mono-plane and other, which is certainly a better prediction than the annotation hip, which is derived from the LOD. This example shows, that the method described in Chapter 4 produces roof-type maps which, in some places, provides more accurate roof-types than the CityGML data.

EXAMPLE 7 In Figure 20, an example of a hip roof, which was not predicted correctly by the network, is shown. There are two reasons described in the Figure, which makes it hard for the network to identify the hip roof correctly. **i.** the triangles at the ends of the roof have only very little inclination and their boundaries are very weak. Hence, even though the main characteristics of a hip roof are featured in the input, they are not strong enough for the network to identify. **ii.**, the ridge line is corrupted by a bump in the middle of the building. This bump stems from the stereo DSM and could not be fixed completely by the refinement process (compare Figure 9). This shows, that the network is only robust to noise in the input to a certain degree.

EXAMPLE 8 In Example 8, the case of a gable roof, which is segmented as a hip roof is studied (see Figure 21). Only what is visible in Figure 21 is the basis for the incorrect prediction of the network, since the building is located at the lower end of the test area (compare Figure 8). Therefore, it is not possible, that the network sees a triangular shape at the other end of the building. The input features the important ridge line, which is a characteristic of both gable and hip roofs. This shows, that the network can not always distinguish if a ridge line belongs to a gable, or a hip roof. Thus, the decision boundary between gable and hip in the space of the high-dimensional input of the network sometimes crosses a high-density area, which indicates, that the smoothness assumption in Section 3.1 is not completely satisfied.

EXAMPLE 9 In Figure 22, a hip roof in direct neighborhood to mono-plane roofs is visualized. The separation of the buildings in the CityGML data, which is represented by the annotation of these two buildings as gable and other, is neither intuitive nor optimal for 3D reconstruction. The prediction includes mostly gable and other roofs. The network does not recognize the hip characteristics, but segments the hip roof mostly as gable. The gable

segment even exceed the lower boundary of the hip roofs and covers the mono-plane roofs. This Example shows, that the network **i.** confuses hip and gable roofs and **ii.** extends segments over reasonable boundaries.

Name	Class	$prec_i$	rec_i	$spec_i$	$(acc_b)_i$	$(F_1)_i$	acc_b	F_1
DLv3+_CE	gable	**46,4**	0,7	**100**	50,3	1,3	65,8	39,1
	hip	**39,1**	1,1	**100**	50,5	2,1		
DLv3+_CEInvW	gable	32,4	38,8	95,1	**66,9**	**35,3**	**75,7**	**54,3**
	hip	26	33,3	98,9	**66,1**	**29,2**		
DLv3+_CERootW	gable	40,4	11,2	99	17,6	17,6	70	45,8
	hip	12,4	1,9	99,8	3,4	3,4		
DLv3+_CESqrW	gable	9,2	**86,6**	48,2	67,4	16,7	56,7	12,2
	hip	1,8	**37,1**	75,8	56,5	3,4		
DLv3+_FOC05InvW	gable	26,2	32,7	**94,4**	63,5	29,1	74,4	52,6
	hip	**32,9**	33	**99,2**	66,1	33		
DLv3+_FOC10InvW	gable	**28,1**	**36,4**	94,3	**65,4**	**31,7**	**75,5**	**53,8**
	hip	29,5	**38,4**	98,9	**68,6**	**33,4**		
DLv3+_FOC20InvW	gable	24,5	34,9	93,4	64,2	28,8	74	51,7
	hip	27,8	32	99	65,5	29,8		
DLv3+_DICE	gable	10,8	0,6	99,7	50,2	1,2	64,2	**44,2**
	hip	**95,5**	**21,5**	**100**	**60,8**	**35,1**		
DLv3+_DICEInvW	gable	**61**	4,9	99,8	52,4	9,1	**64,7**	36,7
	hip	0	0	**100**	0	0		
DLv3+_DICERootW	gable	65	0,7	**100**	50,4	1,5	64,3	39,1
	hip	80,1	2,1	**100**	51,1	4,2		
DLv3+_DICESqrW	gable	49,2	**8,1**	99,5	**53,8**	**13,9**	62,8	36
	hip	56,2	3,5	**100**	51,7	6,6		
DLv3+_CE_impr	gable	76,5	2,9	99,9	51,4	5,6	69,2	47,7
	hip	81,3	23,3	99,9	61,6	36,2		

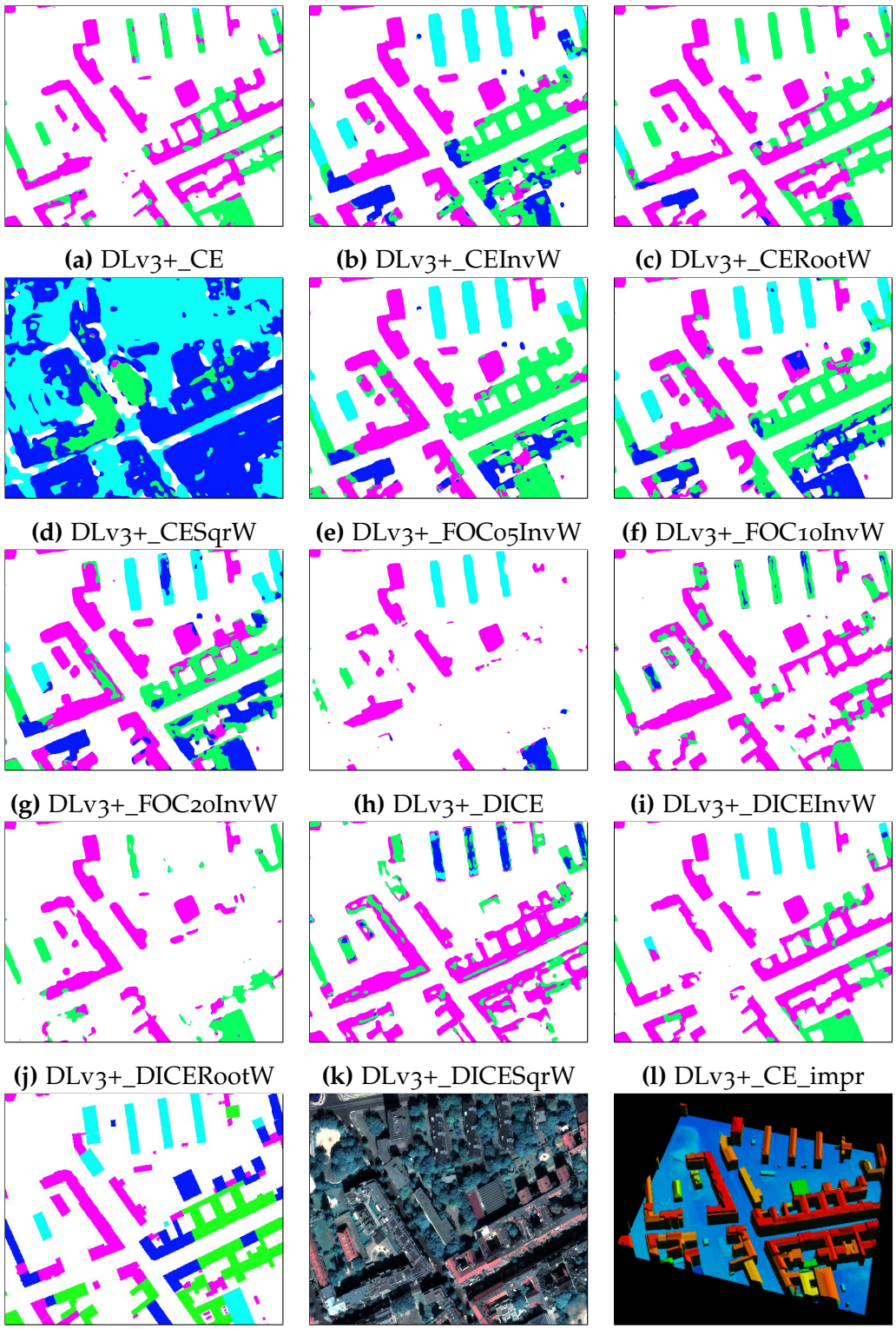
Table 8: The quantitative results of the Experiments on class-balancing from Subsection 5.2.1. The per-class metrics are only listed for the classes gable and hip, since these are the underrepresented classes and hard to learn from the given data. The overall metrics acc_b and F_1 are the arithmetic mean over all five classes.

Name	Class	$prec_i$	rec_i	$spec_i$	$(acc_b)_i$	$(F_1)_i$	acc_b	F_1
DLv3+_FOC10InvW_impr	background	98,2	85,7	97,3	91,5	91,6	75,3	55,9
	flat	**59,1**	52,7	**92,6**	72,7	55,7		
	gable	37,9	36,1	96,4	66,3	37		
	hip	**56**	**35**	99,7	**67,3**	**43,1**		
	other	40,3	**74**	83,9	**79**	52,2		
DUN_FOC10InvW_impr	background	98,3	86,3	97,4	91,8	91,9	75,1	55,6
	flat	54,8	**70,6**	88,3	**79,4**	**61,7**		
	gable	**47,1**	34	97,7	65,8	39,5		
	hip	42,3	25,9	99,6	62,7	32,1		
	other	**45,2**	62,9	**88,8**	75,9	52,6		

Table 9: The quantitative results of the Experiments on architecture from Subsection 5.2.2. The per-class metrics are listed for the classes background, mono-plane, gable, hip and other.The overall metrics acc_b and F_1 are the arithmetic mean over all five classes.

Name	Class	$prec_i$	rec_i	$spec_i$	$(acc_b)_i$	$(F_1)_i$	acc_b	F_1
DLv3+_SB99FOC10InvW_impr	gable	46,4	**41,3**	97,1	**69,2**	**43,7**	**76,9**	**58,5**
	hip	49,6	**40,4**	99,5	**69,9**	44,5		
DLv3+_SB90FOC10InvW_impr	gable	45,1	36,4	97,3	66,8	40,3	75,8	**58,5**
	hip	61,7	36,4	99,7	68,1	**45,8**		
DLv3+_SB80FOC10InvW_impr	gable	49,7	18,7	98,8	58,8	27,1	73,9	55,9
	hip	72,2	28,7	99,9	64,3	41,2		
DLv3+_SB70FOC10InvW_impr	gable	**58,7**	19,3	99,2	59,2	29	73,3	55,2
	hip	63,3	25,2	99,8	62,5	36,1		
DLv3+_SB50FOC10InvW_impr	gable	55,9	11,3	**99,5**	55,4	18,9	72,1	53,5
	hip	**90,7**	23,3	**100**	61,6	37,1		

Table 10: The quantitative results of the Experiments on imperfect ground truth from Section 5.2.3. The per-class metrics are only listed for the classes gable and hip, since these are the underrepresented classes and hard to learn from the given data. The overall metrics acc_b and F_1 are the arithmetic mean over all five classes.



(a) DLv3+_CE **(b)** DLv3+_CEInvW **(c)** DLv3+_CERootW

(d) DLv3+_CESqrW **(e)** DLv3+_FOC05InvW **(f)** DLv3+_FOC10InvW

(g) DLv3+_FOC20InvW **(h)** DLv3+_DICE **(i)** DLv3+_DICEInvW

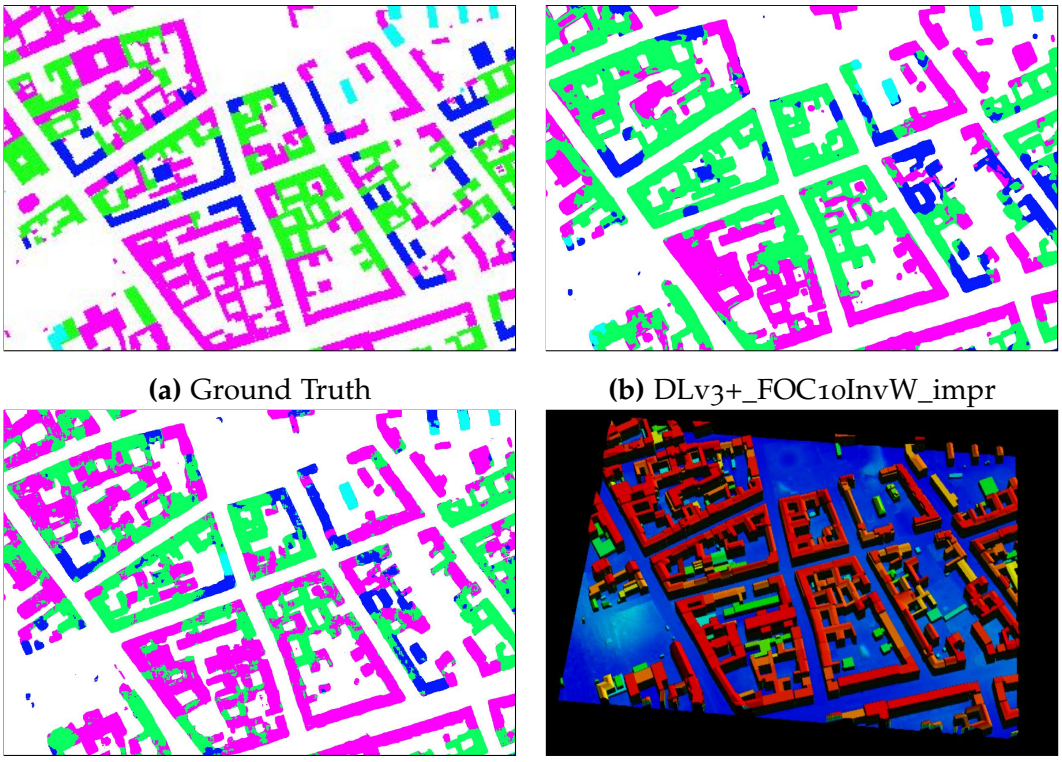
(j) DLv3+_DICERootW **(k)** DLv3+_DICESqrW **(l)** DLv3+_CE_impr

(m) Ground Truth **(n)** RGB **(o)** LOD

Figure 11: A visualization of a small area from the results of the experiments on class-balancing over the test area. **White:** background, **magenta:** monoplane, **blue:** gable, **cyan:** hip, **green:** other. Best viewed zoomed in. In **(a)**, **(i)**, **(j)**, almost none of the pixels where assigned to the minority classes gable and hip. In **(d)**, the minority classes have many false positives when compared to the ground truth in **(m)** and the buildings are not separated from the ground truth. In **(b)**, **(c)**, **(e)**, **(f)**, **(g)**, **(h)**, **(k)** and **(l)**, all classes are present and the buildings are separated from the background.

Name	Class	$prec_i$	rec_i	$spec_i$	$(acc_b)_i$	$(F_1)_i$	acc_b	F_1
DLv3+_SB99FOC10InvW_impr2	gable	**48**	42,1	97,5	67,9	42,7	73,6	52,5
	hip	41,4	**38,4**	**99,7**	60	27,3		
DLv3+_SB90FOC10InvW_impr2_vat	gable	45,1	44,5	96,7	70,6	**44,8**	74,8	54,5
	hip	37,9	27,8	99,5	63,3	32,1		
DLv3+_SB99FOC10InvW_impr2_vat2	gable	52	38,6	97,8	68,2	44,3	**75,3**	**55,7**
	hip	38,8	31,6	99,4	65,5	**34,8**		
DLv3+_SB99FOC10InvW_impr2_vat3	gable	33,2	**65**	92	**78,5**	44	74,1	51,2
	hip	**42,9**	31,8	99,5	**65,7**	36,5		
DLv3+_SB99FOC10InvW_impr2_vat4	gable	45,4	31,6	**97,7**	64,6	37,2	66,3	43,5
	hip	16,3	18,3	98,9	58,6	17,3		

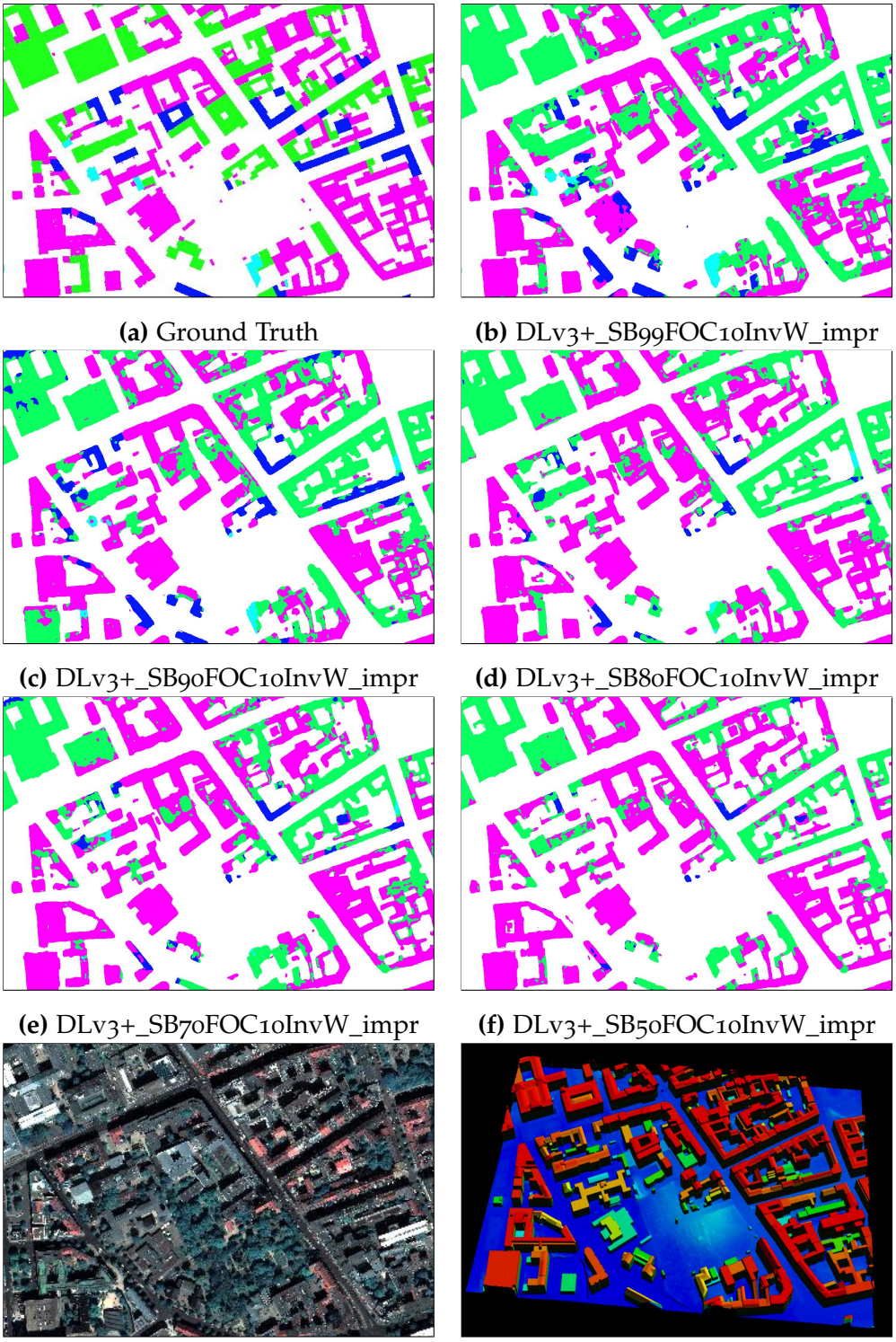
Table 11: The quantitative results of the Experiments from Section 5.2.4. The per-class metrics are only listed for the classes gable and hip, since these are the underrepresented classes and hard to learn from the given data. The overall metrics acc_b and F_1 are the arithmetic mean over all five classes.



(a) Ground Truth **(b)** DLv3+_FOC10InvW_impr

(c) DUN_FOC10InvW_impr **(d)** LOD

Figure 12: A visualization of a small area from the results of the experiments on architecture over the test area. **White:** background, **magenta:** mono-plane, **blue:** gable, **cyan:** hip, **green:** other. Best viewed zoomed in. In **(a)**, the ground truth is shown. In **(b)**, the segments look homogeneous in many places, but the boundaries of the building segments exceeds the boundaries in the ground truth. In **(c)**, the shapes of the roof boundaries are very similar to the ground truth, but the segments are not homogeneous in many places.



(a) Ground Truth

(b) DLv3+_SB99FOC10InvW_impr

(c) DLv3+_SB90FOC10InvW_impr

(d) DLv3+_SB80FOC10InvW_impr

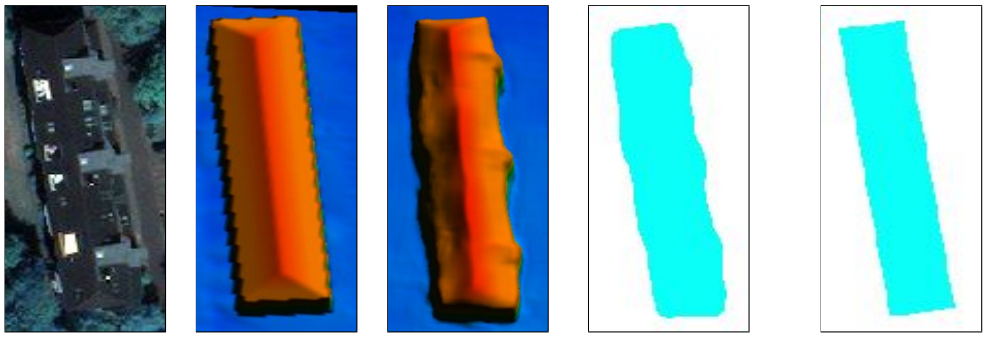
(e) DLv3+_SB70FOC10InvW_impr

(f) DLv3+_SB50FOC10InvW_impr

(g) RGB

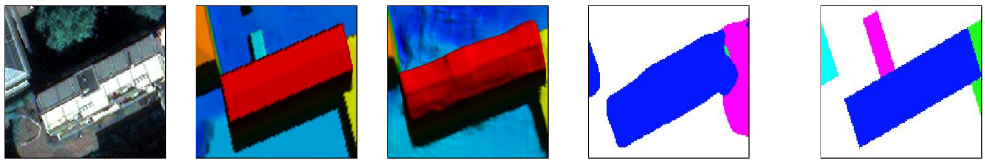
(h) LOD

Figure 13: A visualization of a small area from the results of the experiments on imperfect ground truth over the test area. **White:** background, **magenta:** mono-plane, **blue:** gable, **cyan:** hip, **green:** other. Best viewed zoomed in.



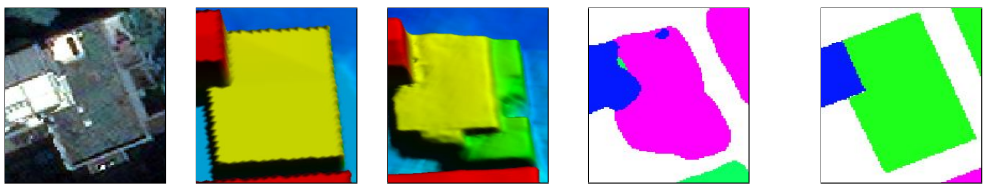
(a) RGB **(b)** LOD **(c)** input **(d)** prediction **(e)** Ground Truth

Figure 14: Visualization of a building in the test area. In **(a)** and **(b)**, the characteristics of a hip roof are clearly visible. In **(c)**, the main characteristics of a hip roof are degraded but still featured. The prediction in **(d)** shows a hip roof (cyan), which matches the ground truth. Best viewed zoomed in.



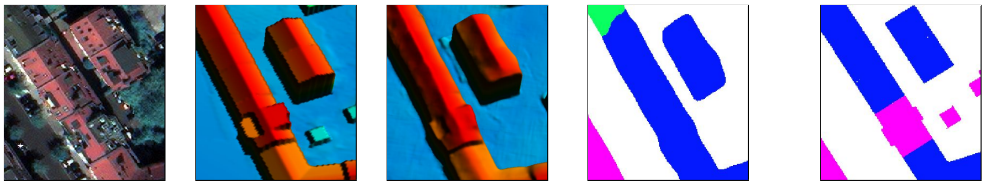
(a) RGB **(b)** LOD **(c)** input **(d)** prediction **(e)** Ground Truth

Figure 15: Visualization of a building in the test area. In **(a)** and **(b)**, the characteristics of a gable roof are clearly visible. In **(c)**, the main characteristics of a gable roof are also featured. The prediction in **(d)** shows a gable roof (blue), which matches the ground truth. Best viewed zoomed in.



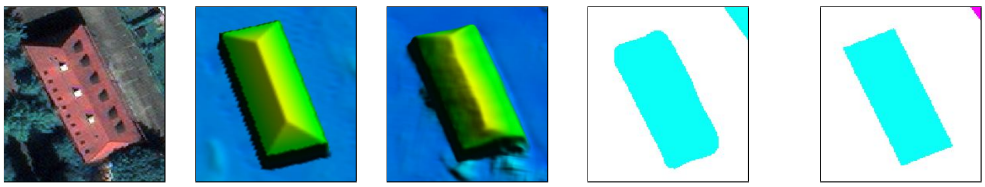
(a) RGB **(b)** LOD **(c)** input **(d)** prediction **(e)** Ground Truth

Figure 16: Visualization of a building in the test area. In **(a)**, **(b)** and **(c)**, the characteristics of a mono-plane roof are clearly visible. In **(a)** and **(c)**, there are multiple different mono-plane roofs, which all belong to the same building, which is why the building was labeled as class other (green) in **(e)**. The prediction in **(d)** shows, that the network has labeled the entire building as mono-plane. Best viewed zoomed in.



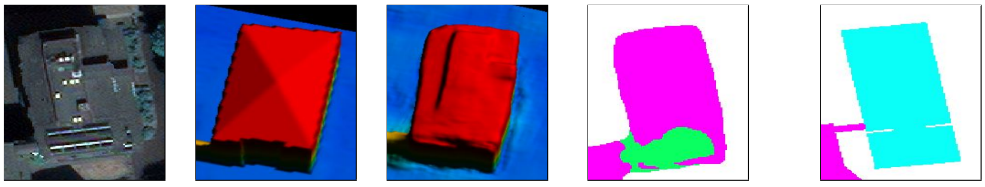
(a) RGB **(b)** LOD **(c)** input **(d)** prediction **(e)** Ground Truth

Figure 17: Visualization of a building in the test area. In **(a)**, **(b)** and **(c)**, two gable roofs are visible. There is a small part in **(b)**, where the longer building is flat on the right half of the ridge line. This part is labeled as mono-plane (magenta) in **(e)**. In **(d)**, the network assigns gable to both of the buildings, including the part with the flat shares. Best viewed zoomed in.



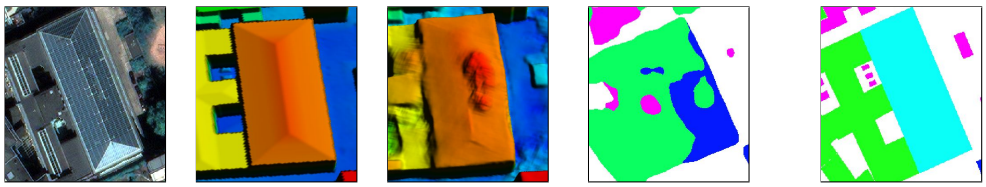
(a) RGB **(b)** LOD **(c)** input **(d)** prediction **(e)** Ground Truth

Figure 18: Visualization of a building in the test area. In **(a)** and **(b)**, the characteristics of a hip roof are clearly visible. In **(c)**, the main characteristics of a hip roof are degraded but still featured. The prediction in **(d)** shows a hip roof (cyan), which matches the ground truth. Best viewed zoomed in.



(a) RGB **(b)** LOD **(c)** input **(d)** prediction **(e)** Ground Truth

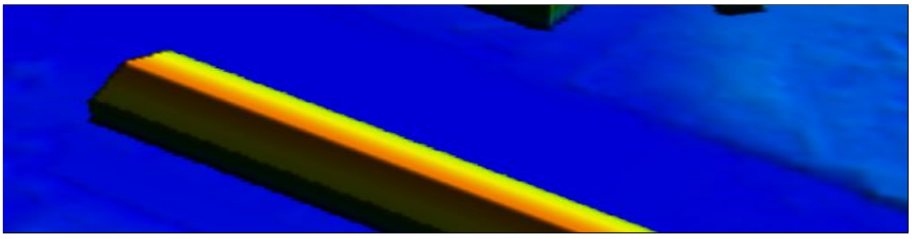
Figure 19: Visualization of a building in the test area. In **(a)** and **(c)**, the building features multiple levels of flat roofs, such that the building belongs into the other class. In **(b)**, the building features characteristics of a tent roof, which is very similar to the hip roof and was therefore annotated as hip (cyan) in **(e)**. The network's prediction in **(d)** is a mix of mono-plane and other. Best viewed zoomed in.



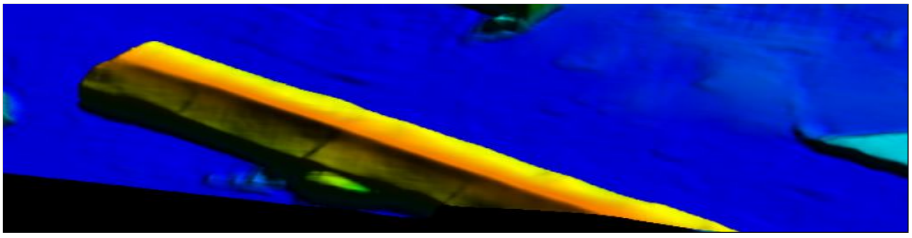
(a) RGB **(b)** LOD **(c)** input **(d)** prediction **(e)** Ground Truth

Figure 20: Visualization of a building in the test area. In **(a)** and **(b)**, the characteristics of a hip roof are visible and in **(e)**, the building is labeled as hip (cyan). In **(c)**, the triangles of the hip roof are still visible, but might not be clear enough. Furthermore, there is a huge bump visible in **(c)**, which is also visible in Figure 9 **(b)**. In **(d)**, the output of the network assigns a mix of gable (blue) and other (green) to the building. Best viewed zoomed in.

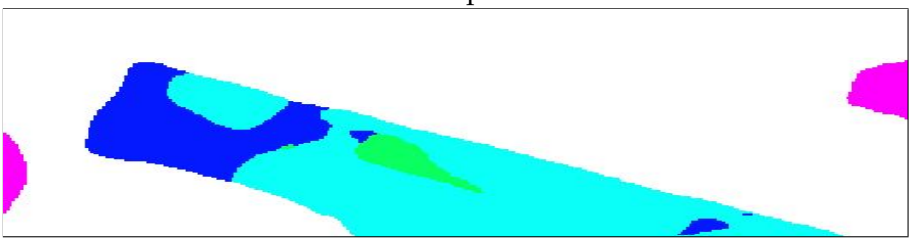
(a) RGB



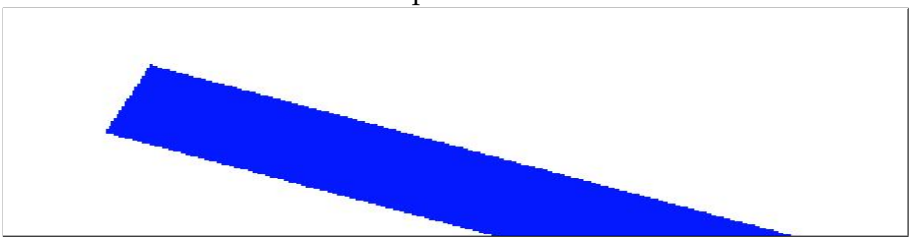
(b) LOD



(c) input

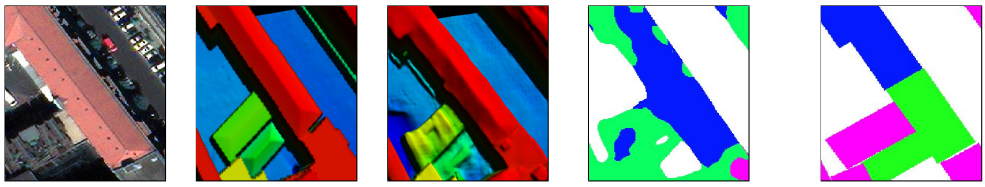


(d) prediction



(e) Ground Truth

Figure 21: Visualization of a building in the test area. In **(a)**, **(b)** and **(c)**, a gable roof is visible. In **(e)**, the building is annotated as gable (blue). In **(b)**, the network predicts the building to have characteristics of gable (blue), other (green) and hip (cyan), but the hip class dominates. Best viewed zoomed in.



(a) RGB **(b)** LOD **(c)** input **(d)** prediction **(e)** Ground Truth

Figure 22: Visualization of a building in the test area. In **(a)** and **(b)**, the biggest buildings has a triangular end and a ridge line and a smaller flat building is next to the triangular end. In **(c)**, the triangle barely visible. In **(e)**, the hip (cyan) roof and the mono-plane (magenta) building are not divided into a hip and a mono-plane roof, but a gable (blue) and an other (green) roof. In the predictions in **(d)**, the hip roof is labeled as gable and the mono-plane roof is labeled as a mix of other and mono-plane. Best viewed zoomed in.

7 CONCLUSION

Modern satellites, such as World-View-1 (WV-1) and World-View-4 (WV-4), supply a huge amount of Very High-Resolution (VHR) satellite imagery. The quality of these images is high enough to visualize various features of buildings. For instance, stereo Digital Surface Models (DSMs) can be improved by deep learning methods to obtain a refined DSM. In 3D reconstruction, these features are nowadays exploited by deep learning methods. One of the open problems in 3D reconstruction is the classification of the roof-type of a building instance. As a step towards achieving this goal, dense classification of roof-types can be an intermediate step and later be combined with instance-level methods. In this thesis, a method for dealing with imbalanced, imperfect and incomplete ground truth during the training of an Fully Convolutional Neural Network (FCN) is evaluated. In this Chapter, the findings of the experiments, which are described in Chapter 5 and evaluated in Chapter 6, are summarized.

- Weighting in the Cross-Entropy (CE) loss balances the classes better than no weighting. Especially the inverse frequency weighting encourages the model to not neglect the minority classes like gable and hip, which it otherwise does. Furthermore, the focal loss brings a visually visible improvement and puts more emphasize on the rarest minority class like hip than the CE loss. Also, adding buildings of the minority classes to the training data improves the segmentation of minority classes like gable and hip.

- For tackling the problem of imperfect ground truth, the soft-bootstrapped CE loss improves the qualitative and quantitative results for roof-type segmentation. Especially the minority classes profit from this error-robust loss function, which is connected to the assumption that errors weigh much heavier if only few examples of a class are available.

- For a very small labeled, and a large unlabeled dataset, Virtual Adversarial Training (VAT) improves the performance of a DeepLabv3+ on roof-type segmentation.

- The proposed method is successful in many places, but fails to predict the correct roof-type in many other cases, e. g. if the input data has defects. In many cases, the two minority classes gable and hip are confused due to common features.

Finally, deep learning with imbalanced, imperfect and incomplete data can only work in certain boundaries, which are explored in this thesis. To obtain significantly better results in roof-type segmentation, a training dataset must be used which has a higher quality than the one described in Chapters 1 and 5.

8 FUTURE WORK

Roof-type segmentation's goal is to provide highly accurate predictions of the roof-type for 3D building reconstruction. With the results presented in this thesis, this can not be accomplished (see Chapter 7). To improve the roof-type segmentation, there are multiple possible solutions, which are based on the findings of this thesis.

8.1 EXTRA ANNOTATIONS

As the experiments DLv3+_CEInvW_impr and DLv3+_SB99FOC10InvW_impr2_vat2 in Chapters 5 and 6 showed, small improvements in the ground truth, in terms of balance and ratio of labeled data can instantly improve the quality of the roof-type segments. Therefore, the investment of labeling more buildings pays off, especially when combined with VAT, which can regularize an FCN, such that it learns from a small labeled and a big unlabeled dataset to achieve an increase in performance if compared with the fully supervised experiment DLv3+_SB99FOC10InvW_impr2 and a comparable performance as an FCN trained on a much greater labeled dataset as in DLv3+_SB99FOC10InvW_impr. Therefore, increasing the density of labels in an area comparable to TRAIN1 and TRAIN5 in Figure 10, might lead to a big increase in performance.

8.2 IMPROVED VALIDATION DATA

The quality of the validation data in this thesis was not sufficient to explain the performance of the FCN on the test set, since the validation data labels are incomplete an erroneous, as was the rest of the labels before a small test area was corrected and completed and training data was extended for the minority classes gable and hip. To optimize the hyper-parameter search, and especially the number of epochs, an improved validation ground truth could lead to a boost in performance.

8.3 INCORPORATING SPECTRAL DATA

As described in Chapter 6 the proposed methods often fails due to missing features in the input height data. Spectral data often includes features, which are not visible in height data. A combination of multiple data sources,

even of multiple resolutions, can increase the performance of FCNs for roof-type segmentation which Bittner et al. [30] and Schuegraf and Bittner [31] have verified in their work. This raises the question if a combination of multiple data sources can lead to an improved performance in roof-type segmentation. The results of Peng et al. [21] show that Deep Co-Training can improve the performance of semi-supervised semantic segmentation by using multiple different input representations. However, instead of adversarial examples as in Peng et al. [21], Deep Co-Training might be most suitable for roof-type segmentation with a spectral input in addition to the height input.

8.4 INSTANCE SEGMENTATION

To reconstruct cities in 3D, the roof-type has to be known for building instances, not pixels. The Mask-RCNN architecture [32] can separate instances from each other. This can be leveraged for 3D reconstruction in many ways. Two of them are **(i.)** computing a building instance map and combining it with the roof-type segmentation results and **(ii.)** directly classify the roof-types within the Mask-RCNN network.

8.5 MULTI-TASK LEARNING

In their work Bittner et al. [1] show that training a common encoder together with one decoder for DSM refinement and roof-type segmentation can improve the performance on both tasks. Also, the method proposed in Bittner et al. [1] does not need a refined DSM as input, which improves the quality of the features if the ground truth for the refinement is limited because it overlaps with the ground truth for the roof-type segmentation, compared to separately training on a small part of the area with the ground truth for refinement and then using the refined DSM for roof-type segmentation. Hence, a multi-task approach could further improve the roof-type segmentation results, even for multiple imbalanced, incomplete and erroneous classes.

8.6 IMPROVED SEMI-SUPERVISED SEMANTIC SEGMENTATION

The choice of the algorithm for semi-supervised semantic segmentation in this thesis was made based on the intuition that a network, which is robust to attacks, generalizes better. However, there are other approaches to semi-supervised learning, which achieve better metrics on benchmark datasets [19], [21], [20], [6]. An exhaustive study, using relatively small, but corrected

and complete, ground truth data, with additional unlabeled data, might lead to solid improvements in roof-type segmentation.

BIBLIOGRAPHY

[1] Ksenia Bittner, Marco Körner, Friedrich Fraundorfer, and Peter Reinartz. "Multi-Task cGAN for Simultaneous Spaceborne DSM Refinement and Roof-Type Classification." In: *MDPI Remote Sensing* (2019). DOI: 10.3390/rs11111262.

[2] Tahmineh Partovi, Friedrich Fraundorfer, Seyedmajid Azimi, Dimitrios Marmanis, and Peter Reinartz. "Roof Type Selection based on patch-based classsification using deep learning for high Resolution Satellite Imagery." In: *ISPRS Archives* (2017). DOI: 10.5194/isprs-archives-XLII-1-W1-653-2017.

[3] Thomas Heinrich Kolbe, Gerhard Gröger, and Lutz Plümer. 2005. DOI: 10.1007/3-540-27468-5_63. URL: https://www.researchgate.net/publication/237443110_CityGML_-_Interoperable_access_to_3D_city_models.

[4] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. "Focal Loss for Dense Object Detection." In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (2017), pp. 2980–2988. URL: https://openaccess.thecvf.com/content_iccv_2017/html/Lin_Focal_Loss_for_ICCV_2017_paper.html.

[5] Corentin Henry, Friedrich Fraundorfer, and Eleonora Vig. "Aerial Road Segmentation in the Presence of Topological Label Noise." In: *International Conference on Pattern Recognition* (2021).

[6] Geoff French, Samuli Laine, Timo Aila, Michal Mackievicz, and Graham Finalyson. "Semi-Supervised Semantic Segmentation needs Strong, Varied Perturbations." In: (2020).

[7] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. "Encoder-Decoder with Atrous Separable Convolution for Semantic Segmentation." In: (2018). URL: https://arxiv.org/pdf/1802.02611.pdf.

[8] Lei Ma, Yu Liu, Xueliang Zhang, Yuanxin Ye, Gaofei Yin, and Brian Alan Johnson. "Deep Learning in Remote Sensing Applications: A meta-analysis and Review." In: *ISPRS Jornal of Photogrammetry and Remote Sensing* 152 (2019), pp. 166–177. DOI: 10.1016/j.isprsjprs.2019.04.015. URL: https://www.researchgate.net/publication/332718482_Deep_learning_in_remote_sensing_applications_A_meta-analysis_and_review.

[9] Maria Axelsson, Ulf Soderman, Andreas Berg, and Thomas Lithen. "Roof Type Classification Using Deep Convolutional Neural Networks on Low Resolution Photogrammetric Point Clouds From Aerial Imagery." In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018. DOI: 10.1109/ICASSP.2018.8461740.

[10] Fatemeh Alidoost and Hossein Arefi. "A CNN-based approach for Automatic Building Detection and Recognition of Roof Types using a Single Aerial Image." In: *PFG-Journal of Photogrammetry, Remote Sensing and Geoinformation Science* 86 (Dec. 2018), pp. 235–248. ISSN: 5-6.

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

[12] I. Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. "Billion-scale semi-supervised learning for image classification." In: (2019). URL: https://www.researchgate.net/publication/332832081_Billion-scale_semi-supervised_learning_for_image_classification/stats.

[13] I. Goodfellow, J. Shlens, and C. Szegedy. "Explaining and Harnessing Adversarial Examples." In: *ICLR 2015* (2015). URL: https://arxiv.org/pdf/1412.6572.pdf.

[14] Min Bai and Raquel Urtasun. "Deep Watershed Transform for Instance Segmentation." In: *IEEE Conference on Computer VIsion and Patter Recognition*. 2017. DOI: 10.1109/CVPR.2017.305.

[15] Stefano Zorzi and Friedrich Fraundorfer. "Regularization of Building Boundaries in Satellite Images Using Adversarial and Regularized Losses." In: *IGARSS*. 2019. DOI: 10.1109/IGARSS.2019.8900337.

[16] Takeru Miyato, Shin-Ichi Maeda, Masanori Koyama, Ken Nakae, and Shin Ishii. "Distributional Smoothing with VIrtual Adversarial Training." In: *ICLR 2016*. 2016.

[17] Takeru Miyato, Andrew M. Dai, and Ian Goodfellow. "Adversarial Training Methods for Semi-Supervised Text Classification." In: *ICLR 2017*. 2017.

[18] Takeru Miyato, Shin-Ichi Maeda, Masanori Koyama, and Shin Ishii. "Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.8 (Aug. 2019), pp. 1979–1993. DOI: 10.1109/TPAMI.2018.2858821.

[19] Seunghoon Hong, Hyeonwoo Noh, and Bohyung Han. "Decoupled Deep Neural Network for Semi-Supervised Semantic Segmentation." In: (2015).

[20] Wei-Chih Hung, Yi-Hsuan Tsai, Yan-Ting Liou, Yen-Yu Lin, and Ming-Hsuan Yang. *Adversarial Learning for Semi-Supervised Semantic Segmentation*. 2018. URL: https://arxiv.org/abs/1802.07934.

[21] Jizong Peng, Guillermo Estrada, Marco Pedersoli, and Christian Desrosiers. "Deep Co-Training for Semi-Supervised Image Segmentation." In: *Pattern Recognition* (2020). URL: https://www.sciencedirect.com/science/article/abs/pii/S0031320320300741.

[22] Yassine Ouali, Céline Hudelot, and Myriam Tami. *An Overview of Deep Semi-Supervised Learning*. 2020. URL: https://arxiv.org/abs/2006.05278.

[23] Jesper E. van Engelen and Holger H. Hoos. "A survey on semi-supervised learning." In: *Machine Learning* (2020). DOI: https://doi.org/10.1007/s10994-019-05855-6.

[24] Olvier Chapelle, Bernhard Shölkopf, and Alexander Zien. *Semi-Supervised Learning*. MIT Press, 2006.

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

[26] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "Learning Representations by Back-Propagating Errors." In: *Nature* 323 (Oct. 1986), pp. 533–536. ISSN: 6088. DOI: 10.1038/323533a0.

[27] D. P. Kingma and J. L. Ba. *ADAM: A method for stochastic optimization*. Conference Paper ICLR 2015, 2015.

[28] Kaiming He, Xiangyu Zhan, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition." In: *IEEE Conference on Computer Vision* (2016). DOI: 10.1109/CVPR.2016.90. URL: https://ieeexplore.ieee.org/document/7780459.

[29] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. "Densely Connected Convolutional Networks." In: *Conference on Computer Vision and Pattern Recognition* (July 2017). DOI: 10.1109/CVPR.2017.243.

[30] 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.

[31] Philipp Schuegraf and Ksenia Bittner. "Automatic Building Footprint Extraction from Multi-Resolution Remote Sensing Images Using a Hybrid FCN." In: *International Journal of Geo-Information* (2019).

[32] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. "Mask R-CNN." In: 2017. DOI: 10.1109/ICCV.2017.322.

DECLARATION

I declare that I completed this thesis on my own and that information which has been directly or indirectly taken from other sources has been noted as such. Neither this nor a similar work has been presented to an examination committee.

Rosenheim, 01.04.2021

Philipp Schuegraf