International Journal of Geo-Information

MDPI

Article

Learning Cartographic Building Generalization with Deep Convolutional Neural Networks

Yu Feng *, Frank Thiemann and Monika Sester

Institute of Cartography and Geoinformatics, Leibniz University Hannover, Appelstraße 9a, 30167 Hannover, Germany; frank.thiemann@ikg.uni-hannover.de (F.T.); monika.sester@ikg.uni-hannover.de (M.S.)

* Correspondence: yu.feng@ikg.uni-hannover.de; Tel.: +49-511-762-19437

Received: 16 April 2019; Accepted: 27 May 2019; Published: 30 May 2019

Abstract: Cartographic generalization is a problem, which poses interesting challenges to automation. Whereas plenty of algorithms have been developed for the different sub-problems of generalization (e.g., simplification, displacement, aggregation), there are still cases, which are not generalized adequately or in a satisfactory way. The main problem is the interplay between different operators. In those cases the human operator is the benchmark, who is able to design an aesthetic and correct representation of the physical reality. Deep learning methods have shown tremendous success for interpretation problems for which algorithmic methods have deficits. A prominent example is the classification and interpretation of images, where deep learning approaches outperform traditional computer vision methods. In both domains-computer vision and cartography-humans are able to produce good solutions. A prerequisite for the application of deep learning is the availability of many representative training examples for the situation to be learned. As this is given in cartography (there are many existing map series), the idea in this paper is to employ deep convolutional neural networks (DCNNs) for cartographic generalizations tasks, especially for the task of building generalization. Three network architectures, namely U-net, residual U-net and generative adversarial network (GAN), are evaluated both quantitatively and qualitatively in this paper. They are compared based on their performance on this task at target map scales 1:10,000, 1:15,000 and 1:25,000, respectively. The results indicate that deep learning models can successfully learn cartographic generalization operations in one single model in an implicit way. The residual U-net outperforms the others and achieved the best generalization performance.

Keywords: cartography; map generalization; deep convolutional neural networks; geometry simplification

1. Introduction

Cartographic generalization is the process of generating smaller scale representations from large scale spatial data. This process is being conducted by cartographers by applying different operators, such as selection, simplification or displacement, which—in sum—lead to a simpler and a more clear representation of the spatial scene at a smaller scale. There are many powerful automatic solutions for individual operators. The main challenge today lies in mastering the interplay between different operators, for which e.g., optimization approaches, rule based approaches, or agent based approaches are used. Here, however, the benchmark is still the human operator, who is able to design an aesthetic and correct representation of the physical reality by a careful orchestration of several operators.

Deep learning methods have shown impressive success for interpretation problems for which algorithmic methods are difficult. A prominent example is the classification and interpretation of images, where deep learning approaches outperform traditional computer vision methods. In both domains-computer vision and cartography-humans are very well able to produce good solutions.

Hence, the idea in this paper is to employ deep convolutional neural networks (DCNNs) for cartographic generalizations tasks, especially for the task of building generalization. Many deep learning approaches are based on supervised learning, i.e., they require example data with given input–output pairs. Fortunately, in map generalization, many training data sets are available from the existing map series.

Thus, the idea is straightforward to utilize the deep learning approaches on rasterized building maps for the generalization problem. The neural network architectures applied in this paper are similar to the models used for semantic segmentation [1]. These methods do not provide one classification for a given image patch, but lead to a classification of each pixel. Thus, the output is again an image, however with a label for each pixel. In this paper we adopt this idea. In our case, both the input and output are binary raster images.

The paper presents initial experiences with a straightforward implementation and adaption of some of the well-known existing network architectures, namely U-net, residual U-net and generative adversarial network (GAN). The goal is to explore the potential of deep learning for a cartographic problem.

This paper is organized as follows. The next section is a summary of the state-of-the-art approaches for cartographic generalization. Related works regarding deep neural networks are introduced. In Section 3, the details of the data preparation, implementation and training the models will be reported. In Section 4, an in-depth analysis and discussion of the results will be given both quantitatively and qualitatively. In the last section, we conclude and give an outlook on future work.

2. State of the Art

The automation of building map generalization has been studied by many researchers in recent years—mainly coming from cartography, geoinformatics, and computer science, especially computational geometry (e.g., [2]). The methods are often based on geometric considerations; the integration of operators is relying on optimization (e.g., [3]), rule sets (e.g., [4]), or agent based methods (e.g., [5]).

The approaches are being mainly applied in vector space, however, there are also methods using rasterized representations of the spatial scene. Examples for this kind of operations are aggregation, typification (e.g., [6]), displacement (e.g., [7]), and also building generalization (e.g., [8,9]). A recent approach [10] has applied superpixel segmentation and clustering to simplify and aggregate polygonal features.

Building generalization involves different elements, such as selection (according to size, type or usage), aggregation (in order to close small gaps) and simplification of the building outline. During this process, the typical shape of a building has to be preserved or even enhanced. This often involves rectifying nearly right angles and enforcing parallel lines in the building footprint, which, for instance, can be achieved using a set of rules (e.g., [3,11]). The factors controlling the generalization depend on the scale, and the area of the building, as well as small structures (facade lengths). When moving to even smaller scales, these parameters would lead to an elimination of most of the buildings, therefore, then typification is applied, i.e., the replacement of the buildings by a building template (mostly square or rectangle) while preserving their spatial arrangement. This operation is applied at scales 1:40,000 and smaller.

There have been early attempts to use machine learning to extract cartographic rules from given examples. One goal was to learn suitable parameters of operations (e.g., [12]). In their work, the authors observed a human cartographer in order to learn his actions. In a similar way, Mustière [13] aimed at identifying optimal sequences of operators using machine learning. Sester [14] tried to extract spatial knowledge from given spatial data. While these approaches were very interesting, they mainly remained proofs of concepts.

Deep learning as a new paradigm has re-emerged in recent years, triggered by the now available computational power (especially exploiting GPUs)—allowing us to also design very deep (many

layers) and complex networks—and large quantities of available training data. The success in image interpretation was much influenced by the development of new modelling schemes like CNN [15]. They constrain the number of connections in the network to local environments, thus mimicking the human visual system with it perceptual fields, but also constraining the relations of neurons based on neighborhood principles. Such models have been applied in many different disciplines to classify images, e.g., digit or characters recognition (e.g., [16]), information retrieval of certain scenes (e.g., [17,18]). In further, the fully convolutional networks [19] and its subsequent network architectures achieved to assign class label to each pixel—which is called semantic segmentation. Such methods are being applied in very relevant tasks such as intelligent security systems, autonomous driving and health-care, as well as in topographic mapping, such as semantic segmentation of aerial images (e.g., [20,21]), road extraction from remote sensing images [22].

In the generalization domain, recently machine learning has been proposed for different applications. Xu et al. [23] used an deep autoencoder network to asses the quality of building footprint data by learning the characteristics of quality from OpenStreetMap (OSM) and authoritative data. Zhou and Li [24] compared different machine learning approaches concerning their capability to select important road links for road network generalization. Lee et al. [25] used different machine learning methods to classify buildings as a prior step for their generalization. Cheng et al. [26] propose a Neural Network to detect and learn local situations in building ground plans, which have to be replaced by simpler situations. An approach for generalization of lines from traffic trajectories using deep learning has been proposed by Thiemann et al. [27].

To the best of our knowledge, the approaches proposed in this paper are the first attempts to develop end-to-end solutions for cartographic building generalization using DCNNs. The models do not only learn the simplification of building outlines, but also the aggregation and selection of buildings in one model at the same time. This is an advantage as opposed to most existing solution, which are often focusing on the automation of ONE operation (e.g., simplification of outline); other approaches need dedicated, explicit interaction rules to describe the interplay between different operations (e.g., agent based approaches). Our models do not learn the choice of operations e.g., elimination, simplification, or aggregation as in [25], but generate the generalization results directly, so that the interplay between operators does not need to be considered.

3. Approach

Building generalization is composed of different sub-processes: small buildings are eliminated (selection), small parts of the building outline are eliminated, the outline is simplified (simplification), neighboring objects can be merged (aggregation), too-small buildings can be enlarged (enhancement), buildings are displaced (displacement) and finally, groups of buildings may be replaced by another group, however, with less objects (typification) [28]. The approach presented in this paper aims at an end-to-end training scheme, where an input and a target output are given, and the system has to learn the operations in between. Thus, the expectation is that the system learns all these generalization operations in a holistic way.

In order to train the neural networks, examples have to be provided in terms of corresponding input and output data-pairs. Those data sets are available from existing maps, where situations before and after generalization are depicted. In our approach, we rely on an image based approach, which means that the data is prepared in terms of raster images. One straightforward option would be to use image patches around individual buildings as training data; however, we decided to use the whole map as such, which is cut into regular image patches of given size $b \times b$. When selecting an appropriate size for b, this approach ensures, that adequate context around each building is implicitly used.

In this paper, three network architectures are applied for building generalization. This is a substantial extension of a previous work [29], where only one architecture was employed. The idea of that paper was inspired by the work from Simo-Serra [30], where sketch drawings were simplified

using a deep convolutional neural network. In another attempt of generalization using deep learning, a similar deep neural network structure has been applied to the problem data aggregation, namely to extract a road map from car trajectories [27].

The models evaluated in this paper originate from algorithms used in computer vision, namely U-net [31], residual U-net [32] and GAN [33]. Some modifications have been made to adapt the networks to our current task. Proper parameters were selected according to their learning behaviour on the validation set. Finally, end-to-end models for building generalization were trained for the building maps at three target scales, 1:10,000, 1:15,000 and 1:25,000 respectively.

3.1. U-net

The U-net [31] architecture is an encoder-decoder structure with skip connections, which was at first applied for medical image segmentation with binary labels. Many applications have used this architecture, such as the simplification of sketch drawings [30] or the classification of airborne laser scanning points [34]. The U-net network is symmetric and consists of three parts (shown in Figure 1), down-sampling (left side), bottleneck (middle) and up-sampling (right side) paths. While up-sampling, each up-sampled layer is concatenated with the layer with the same size on the down-sampling path. This step is known as skip connection. Skip connections intend to provide local information while up-sampling. The models are able to generate less blurring results [35].

This is also the basic network architecture used in our previous work [29]. In that work, two modifications have been made compared with the original U-net [31]. Zero-padding was applied at each convolution layer to achieve the same output size as the input image. Down-sampling was conducted using a convolutional layer with stride 2 instead of using max-pooling [36]. The architecture of the network is shown in Figure 1. In that study, the network architecture has shown that it can generate promising results for the building generalization task. In this paper, we intend to improve it by exploring two other architectures. In order to illustrate the improvements, this architecture is used as a baseline.

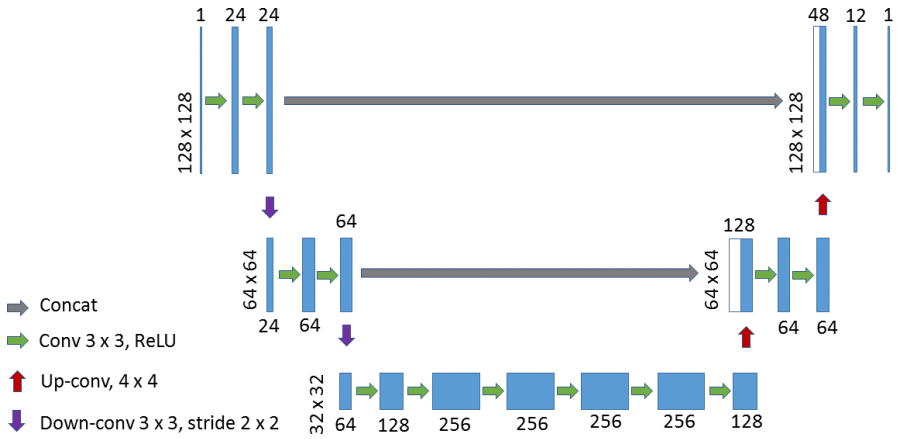

Figure 1. Model architecture for building generalization network.

3.2. Residual U-net

Since building generalization mostly affects the boundary of the buildings, the pixel-wise difference between the input and output images are very small. In order to improve the generalization results, models with higher capacity need to be built. One of the solutions is to build deeper models, however, previous research already proved that simply increasing the number of convolutional layers does not improve the performance of CNNs [37]. Degradation problems may occur. Therefore, the full pre-activation residual unit proposed for residual neural network [37] was used to overcome such problems.

Following this idea, we replaced the convolutional layers in U-net by the residual unit, which consists of batch normalization (BN), ReLU (Rectified Linear Unit) activation and one convolutional layer (as shown in Figure 2). The input of the residual unit is skip connected with the layer after convolution layers with addition. In this way, only the residuals between input and output of the residual unit are learned. By combining the residual units to U-net, the capacity of the model increased and the performance of the model should be improved. A similar idea has been successfully applied for road extraction from aerial images with binary labels [32]. Such architectures show better performance compared with using a simple U-net architecture.

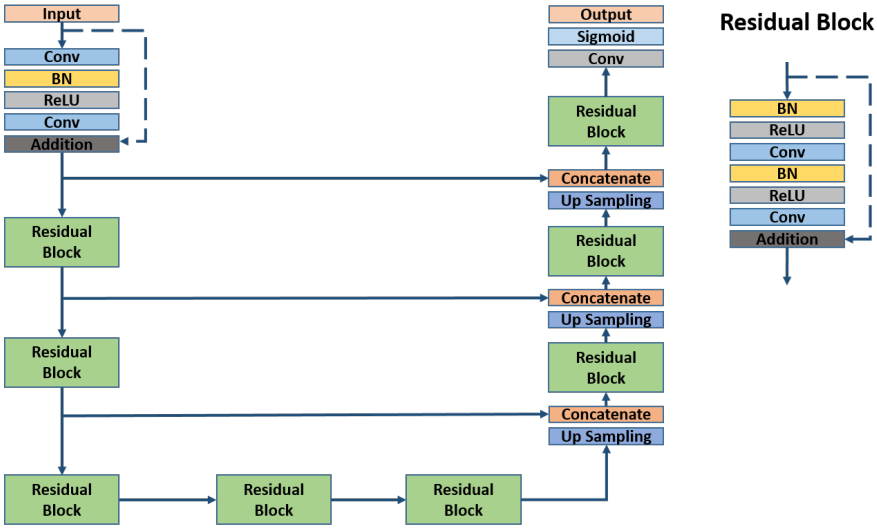

Figure 2. Residual U-net for building generalization.

3.3. Generative Adversarial Network (GAN)

The results from our previous study [29] show that the parallelism and rectangularity of the building patterns in some cases can not be preserved adequately. Another neural network architecture, which has great potential to tackle this problem, is the GAN [33].

This kind of architecture iteratively trains two neural networks, a generator network and a discriminator network. The generator G is used to capture the data distribution and generates samples similar to those drawn from training data. The discriminator D estimates the probability that a sample comes from the real data, rather than being the outputs of generator. The discriminator provides an adversarial loss, which could force the generated images towards the target manifold [38]. Such kinds of network architectures are currently often used for style transfer [38–40], image super-resolution [41], image inpainting, etc. The aim is often to generate photos as realistic as possible. However, for building generalization, we are not only aiming at generating simplified raster building maps at different scales, but the outputs should also look like the buildings in the maps. In this case, the term realistic for GANs means to achieve better rectangularity and parallelism of the building shapes.

Since large amounts of paired input and target images are available for this task, a supervised adversarial training [42] was applied. It is similar to the image-to-image translation problem [35], the supervised loss from the generator was combined with an adversarial loss from discriminator to optimize the model performance. The generator learns a mapping from input image x to output image y. This supervised loss $\mathcal{L}(G(x), y^*)$ can be any standard loss, such as L1, L2 or cross-entropy, which measures the difference between the model output $G(x)$ and target output y^*.

At the same time, the discriminator learns a model which can compute the probability $D(G(x))$ that an input y comes from the real data, rather than being the output of a generator with the following objective function

$$\mathcal{L}_{GAN}(G,D) = \mathbb{E}_{(x,y^*)\sim\rho_{x,y}}\left[\log D(y^*) + \log(1 - D(G(x)))\right]. \quad (1)$$

The final objective function to be optimized is

$$\min_G \max_D \mathcal{L}_{GAN}(G,D) + \lambda\mathcal{L}(G(x),y^*), \quad (2)$$

where λ is a hyper-parameter which generates a weighted combination of the generator loss and adversarial loss. When λ is too high, it performs similar to only using the generator. On the contrary, the model generates realistic outputs without correlation to the inputs.

In this work, the residual U-net is used as generator. Many generative networks, such as CartoonGAN [38], CycleGAN [39], also involve residual units to improve their performance. Therefore, we still consider that the residual blocks is a good choice for the generator. In general, there are two popular architecture for the discriminator. The early one, ImageGAN (as shown in Figure 3), only predicts an overall score, whether the input of the discriminator is real or fake (e.g., GAN [33], cGAN [43], WGAN [44]). Another very popular approach is PatchGAN [35] (as shown in Figure 4) which classifies small patches into real or fake (e.g., Pix2pix [35], CycleGAN [39], DiscoGAN [45]). In our experiment, these two types of discriminators are compared. We trained the PatchGAN structure with a patch size 32 pixel to provide the adversarial loss.

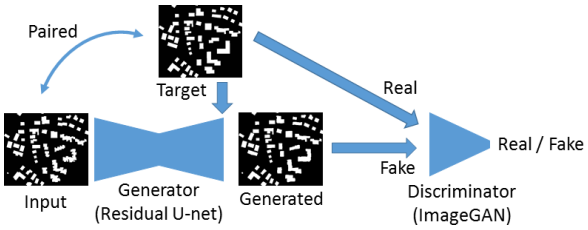

Figure 3. ImageGAN as generative adversarial network (GAN) discriminator.

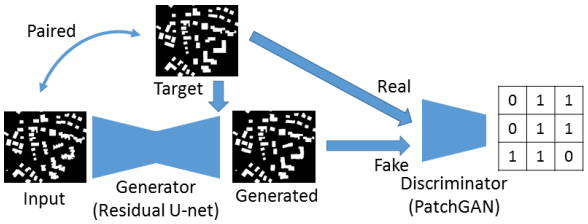

Figure 4. PatchGAN as GAN discriminator.

4. Experiments

In this section, details of the experiment are explained in the following two aspects: preparation of training/validation data set and training of the deep neural networks. Proper hyper-parameters are chosen according to their performance on the validation set.

4.1. Data Preparation

Building polygons available at OpenStreetMap https://www.openstreetmap.org (OSM) were used as input data for training our network. This data had an approximate scale of 1:5,000. We generalized the buildings using the software CHANGE [11]. Three target map scales, 1:10,000, 1:15,000 and 1:25,000, were calculated respectively. The target scale determined different parameters, such as the minimum length of a facade element (3 m, 4.5 m, 7.5 m) or the minimum area (9, 20, 56 m^2) to be preserved. After the generalization with CHANGE, buildings in the original layer and in the target layers were available. As both the input and output data was in vector form, they were rasterized in 0.5 m $\times$ 0.5 m grids. This grid size ensured that the details of the building ground plan are preserved during the rasterization process.

For a selected area in Stuttgart, Germany (shown in Figure 5), a raster building map (42,800 pixel × 35,000 pixel) was generated. This raster image was then partitioned into tiles (128 pixel × 128 pixel) without overlaps (example shown in Figure 6). The image tiles without any building patterns were removed. In total, 31,760 tiles were selected as training set and 3528 tiles (about 10%) were used for validation. In the same way, the corresponding image tiles were generated by CHANGE for all three target map scales.

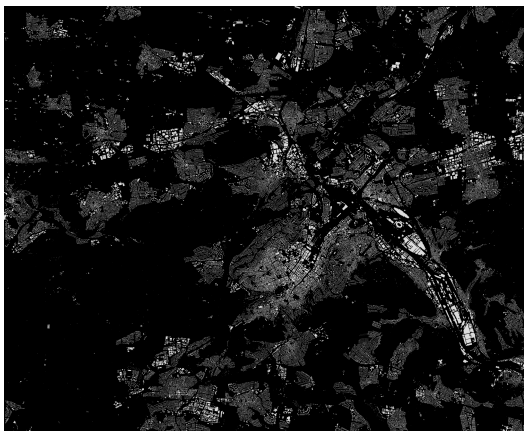

Figure 5. Raster image generated from OpenStreetMap (OSM) in the area of Stuttgart, Germany.

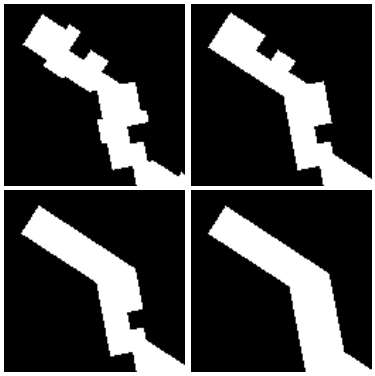

Figure 6. Example of one 128px × 128px map tile from OSM (**upper left**), at scale 1:10,000 (**upper right**), 1:15,000 (**bottom left**), and 1:25,000 (**bottom right**).

4.2. Training the Network

The neural networks were implemented and trained using Keras https://keras.io/ with Tensorflow [46] backend. Early stopping was applied to limit the training process to a manageable scale. When the loss function value on the validation set did not improve over three epochs, the training process stopped automatically. Only the model, which performed the best on the validation set, was used for further evaluation. Data augmentation with randomly given rotation angles was used to avoid duplicated training inputs from epoch to epoch. Other data augmentation operations, such as shift, rescale, were not applied because they are not proper for this task. Due to limitations of our computational resources, we limited the mini-batch size to 16. All experiments in this paper were performed on a PC with Intel Core i7-4790 CPU, 16 GB RAM and one NVIDIA GeForce Titan X GPU.

Training neural networks normally involves tuning many hyper-parameters. Since the same validation set was used during the training of the models, we observed the learning curves to select the proper hyper-parameter for training the networks. Only few previous research focused on learning shape simplification (e.g., sketch drawing simplification [30]) similar to this work, therefore there was little guidance available for training the models. Two parameters were identified which were important for the performance of such models. One was the loss function, which implied the object function to be minimized. Another was the choice of the optimizer as well as its learning rate, which influencd the

training efficiency significantly. Since the GAN used in this work was applied to the residual U-net as generator, no separate experiment has been conducted to compare GAN with the others with respect to loss function and optimizer.

However, GANs are hard to train, in particular, the balance between the discriminator and generator [42]. As stated in Equation (2), λ controls this balance. When λ is too high, it performs similar to only using the generator. On the contrary, the model generated realistic output without correlation with the inputs. Therefore, we further tried both the ImageGAN and PatchGAN as discriminator and figured out the proper value for this coefficient.

4.2.1. Loss Function

The choice of the loss function significantly influences the generalization results. For training the deep neural networks with a label image as output, generally the following three common choices are taken as loss function: L1 loss, which minimizes mean absolute error (MAE), L2 loss, which minimizes mean square error (MSE), and cross entropy loss. Since the data in this task is binary labeled, binary cross entropy loss should be a proper choice. However, for a similar work, sketch drawing simplification [30,42] MSE was used; a work on image-to-image translation [35] mentioned L1 loss could encourage less blurring than the L2 loss. Additionally, many other works regarding semantic segmentation have proved DICE to be efficient. The DICE loss is defined for binary labels as below,

$$\mathcal{L}(y, y^*) = 1 - \frac{2\sum_{i=1}^{N} y_i y_i^*}{\sum_{i=1}^{N} y_i + \sum_{i=1}^{N} y_i^*}, \tag{3}$$

where y is the generated output and y^* is the target output. N is the number of examples in the mini-batch. Therefore, all these four loss functions were evaluated. Experiments were conducted for all the three target map scales respectively to select a proper loss function for this task.

According to Figure 7, the residual U-net using L2 loss outperforms the models using other loss functions for all three scales. Compared with our previous work using U-net (represented by the blue lines), the models with residual blocks in general learn much faster and achieve much better performance on the validation dataset. The models using cross entropy loss and DICE loss perform only slightly worse than using L2 loss. Therefore, L2 loss is chosen as the loss function for the residual U-net.

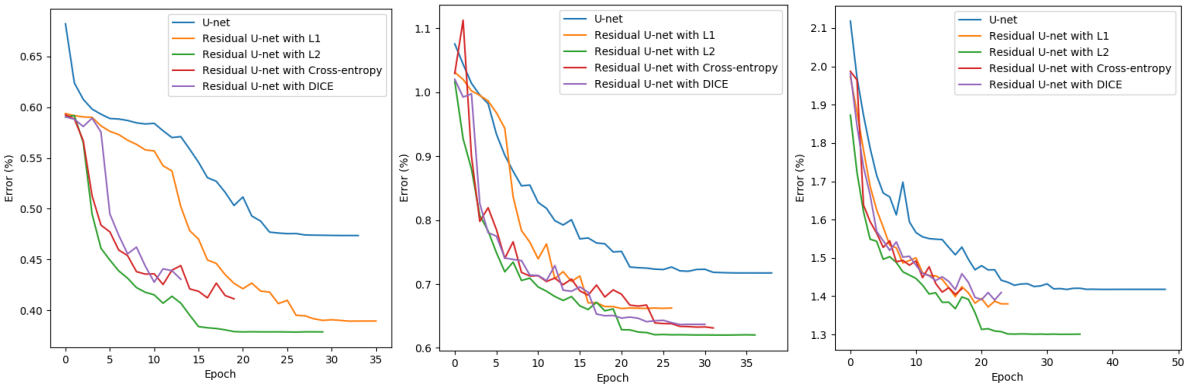

Figure 7. Learning curves of network architectures using different loss functions at target scale 1:10,000 (**left**), 1:15,000 (**middle**) and 1:25,000 (**right**) based on validation set.

4.2.2. Optimizer and Learning Rate

The choice of optimizer and learning rate influences the learning efficiency and model performance significantly. Since only limited computational resources were available, three optimizers: RMSprop [47], Adam [48], Adadelta [49] were tested in this work. Three learning rates were tested for Adam and only the default learning rate in Keras was used for RMSprop and Adadelta.

According to the results (shown in Figure 8), Adam in general performed relatively better, especially for the case at map scale 1:25,000. Among the learning rates we tested for Adam, 0.0004 outperforms the others or at least similar to the best for all the three map scales. Although Adadelta can dynamically adapt itself and needs no manual tuning of a learning rate [49], unfortunately, the model trained for the target scale 1:10,000 failed to converge. For the other two map scales, it performs similar to our baseline method. Therefore, we identified that Adam with a learning rate of 0.0004 should be the proper choice for this task.

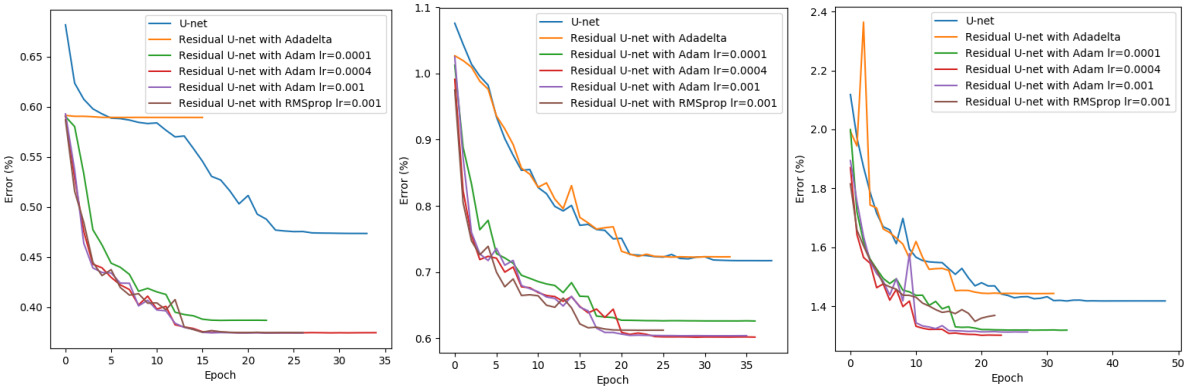

Figure 8. Learning curves of network architectures using different optimizer at target scale 1:10,000 (**left**), 1:15,000 (**middle**) and 1:25,000 (**right**) based on validation set.

4.2.3. GAN: Discriminator and Weighting Factor

As described in Section 3.3, two possible types of discriminators, namely ImageGAN and PatchGAN, were evaluated. According to Equation (2), the weighting factor λ plays an important role while training. In the further material, three possible values, 1000, 100 and 10 were observed. Then, we compared the learning curves of different GANs with the baseline U-net and the best performing residual U-net.

In Figure 9, we observed that none of the GAN achieved a performance similar to the residual U-net. However, most of them achieved a result better than the baseline method. At least, based on the qualitative evaluation on the validation set, adding a discriminator shows no significant benefit for the current task. Compared with PatchGAN, ImageGAN can reach a more stable state at the end and achieved also a relatively better error rate. We also observed that the larger the weighting factor λ, the better performance on the validation set. The training time of adversarial networks and also the need of computational resources were considerably higher than training a residual U-net. Since the results of the ImageGAN as discriminator with a λ of 100 and 1000 do not differ significantly, we investigated the model performance further with the model trained with λ of 100 to observe the benefit of using an adversarial loss.

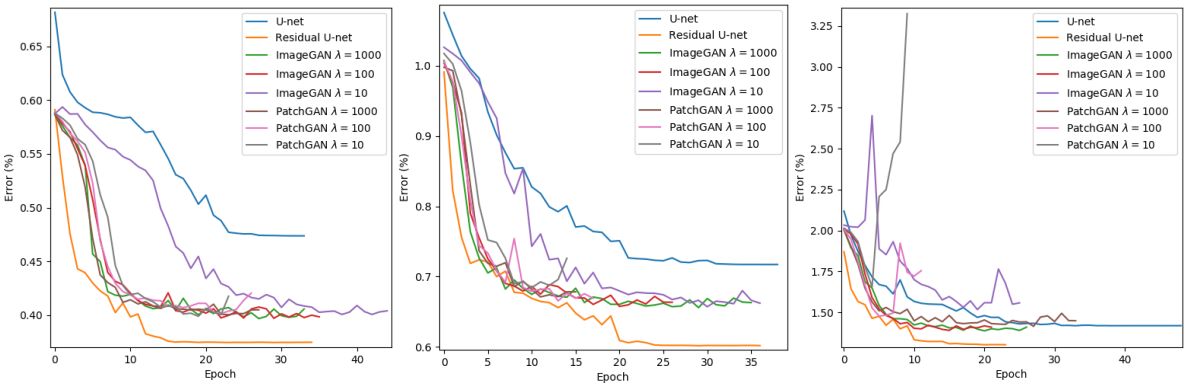

Figure 9. Learning curves of GANs using different discriminators and weighting factors at target scale 1:10,000 (**left**), 1:15,000 (**middle**) and 1:25,000 (**right**) based on validation set.

5. Results and Discussion

In this section, results from the previously described three neural network architectures for building generalization are presented in the following five case studies. An 1.2 km × 1.2 km area outside our training data set was selected and then evaluated quantitatively for the three architectures at the three target map scales. Additionally, two randomly selected small areas outside the training area were evaluated at a specific target map scale 1:15,000 to demonstrate its ability to simplify and aggregate. Qualitatively, specific buildings with different orientations were generalized and a visual comparison was provided. The results at three target scale will be presented with best performed models from each neural network architecture. After that, we present the results where the best-performing residual U-net was applied on a large extent of data for a more detailed visual inspection. Finally, the observations will be summarized and discussed in the last subsection.

5.1. Quantitative Analyses on a Single Independent Test Area at Three Target Map Scales

An area to the south of our study area was selected, which does not have any overlap with our training data. Raster images of 2400 pixel × 2400 pixel at all three target scales were generated from rasterized CHANGE outputs. They are used as the target images for a quantitative evaluation of the models. The models were mainly evaluated based on the error rate, because we want to focus on the pixels where wrong predictions were made.

$$Error\ rate = 1 - Accuracy = \frac{FP + FN}{TP + TN + FP + FN}, \tag{4}$$

where TP = true positive, which indicates the amount of building pixels correctly predicted as building; FP = false positive, which indicates the amount of non-building pixels wrongly predicted as building; TN = true negative, which indicates the amount of non-building pixels correctly predicted as non-building; FN = false negative, which indicates the amount of building pixels wrongly predicted as non-building. Since the difference between the input and target image is small, the error rate of a direct use input compared with target is also listed (Table 1). In further, Table 2 provides the precision, recall and f1-score for the positive class.

Table 1. Error rate at three target map scale for the independent test set.

	Input vs. Target	U-net	Residual U-net	GAN
1:25,000	2.5172%	1.3574%	**1.1589%**	1.2942%
1:15,000	1.3551%	0.7138%	**0.4899%**	0.6137%
1:10,000	0.7396%	0.4956%	**0.3156%**	0.3536%

Table 2. Precision/Recall/F1-score on the positive class at three target map scale for the independent test set.

	Input vs. Target			U-net			Residual U-net			GAN		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
1:25,000	93.94%	93.02%	93.48%	96.42%	96.59%	96.50%	**97.19%**	**96.83%**	**97.01%**	96.76%	96.56%	96.66%
1:15,000	96.84%	96.13%	96.49%	98.19%	98.12%	98.15%	**98.82%**	**98.65%**	**98.73%**	98.55%	98.28%	98.41%
1:10,000	98.20%	97.95%	98.08%	98.77%	98.65%	98.71%	**99.20%**	**99.16%**	**99.18%**	99.14%	99.02%	99.08%

From the results above, it is obvious that all the three methods were able to generate generalized outputs which were closer to the target output rather than the rasterized original building vectors. However, the residual U-net outperformed the other two approaches at all target scales. Even though GAN can not produce the best results, however, it is still better than the baseline approach U-net.

5.2. Experiments with the Specific Map Scale 1:15,000

The following two tests have been conducted for building maps at target scale 1:15,000 (see Figure 10). Test 1 investigates the simplification of certain building structures. The three encircled areas in Figure 10 visualize the generalization capabilities of the U-net model: *A* shows that the intrusions at the corners are filled; *B* shows that the F-shaped building is simplified to a rectangle-similar to the target output produced by CHANGE. The area encircled in *C* visualized that the system was capable of simplifying very complex building outlines: the many extrusions and intrusions were replaced by a smooth outline. Still, the outlines of U-net were not exactly straight lines, nor were the corners exact right angles, as given in the target output.

Then, we compared these three encircled areas with the corresponding parts of the predictions from residual U-net and GAN: the outlines were much closer to straight lines and the corners of the buildings were better preserved. The improvement by introducing residual blocks to the U-nets was obvious.

Still, there were some very small extrusions and intrusions. Note, however, that these visualizations were enlarged substantially in order to show the generalization effect appropriately. To illustrate the real size, Figure 11 shows input image, target image and the predictions from three neural networks in the the dedicated target scale 1:15,000. It can be stated that the effect of generalization of our prediction was clearly positive: as opposed to the original image (left) scaled down to 1:15,000, the forms were simpler, no tiny details disturb the visual impression.

Test 2 in Figure 10 investigated the aggregation where neighbouring objects can be merged. The input and target images were presented together with the predictions from three models. Clearly, the buildings close to each other were merged (see, e.g., the squared block in the lower left corner). The church in *D* was generalized to a better representation compared with the target output from CHANGE. Compared with the prediction from U-net, the results from residual U-net achieved better straightness of the building borders and preserved some of the corners clearly.

In addition to the visual inspection, quantitative tests have also been conducted. For the two independent test sets mentioned above, we evaluated the target output with input image and the predictions from the models based on the error rate (Table 3) and the corresponding precision, recall and f1-score on the positive class (Table 4). The results show that the models did generate predictions much similar to the target output. The residual U-net outperformed the other two models, which was similar to the observation in the previous case study.

Table 3. Error rate for independent test sets at map scale 1:15,000.

	Input vs. Target	U-net	Residual U-net	GAN
Test 1	2.3587%	1.1153%	**0.8850%**	0.9752%
Test 2	1.9261%	0.8975%	**0.7150%**	0.8191%

Table 4. Precision/recall/F1-score on the positive class for independent test sets at map scale 1:15,000.

	Input vs. Target			U-net			Residual U-net			GAN		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Test 1	94.74%	93.13%	93.93%	97.32%	97.05%	97.19%	**97.99%**	**97.55%**	**97.77%**	97.83%	97.24%	97.54%
Test 2	97.60%	95.42%	96.50%	98.51%	98.25%	98.38%	**99.01%**	**98.41%**	**98.71%**	98.92%	98.12%	98.52%

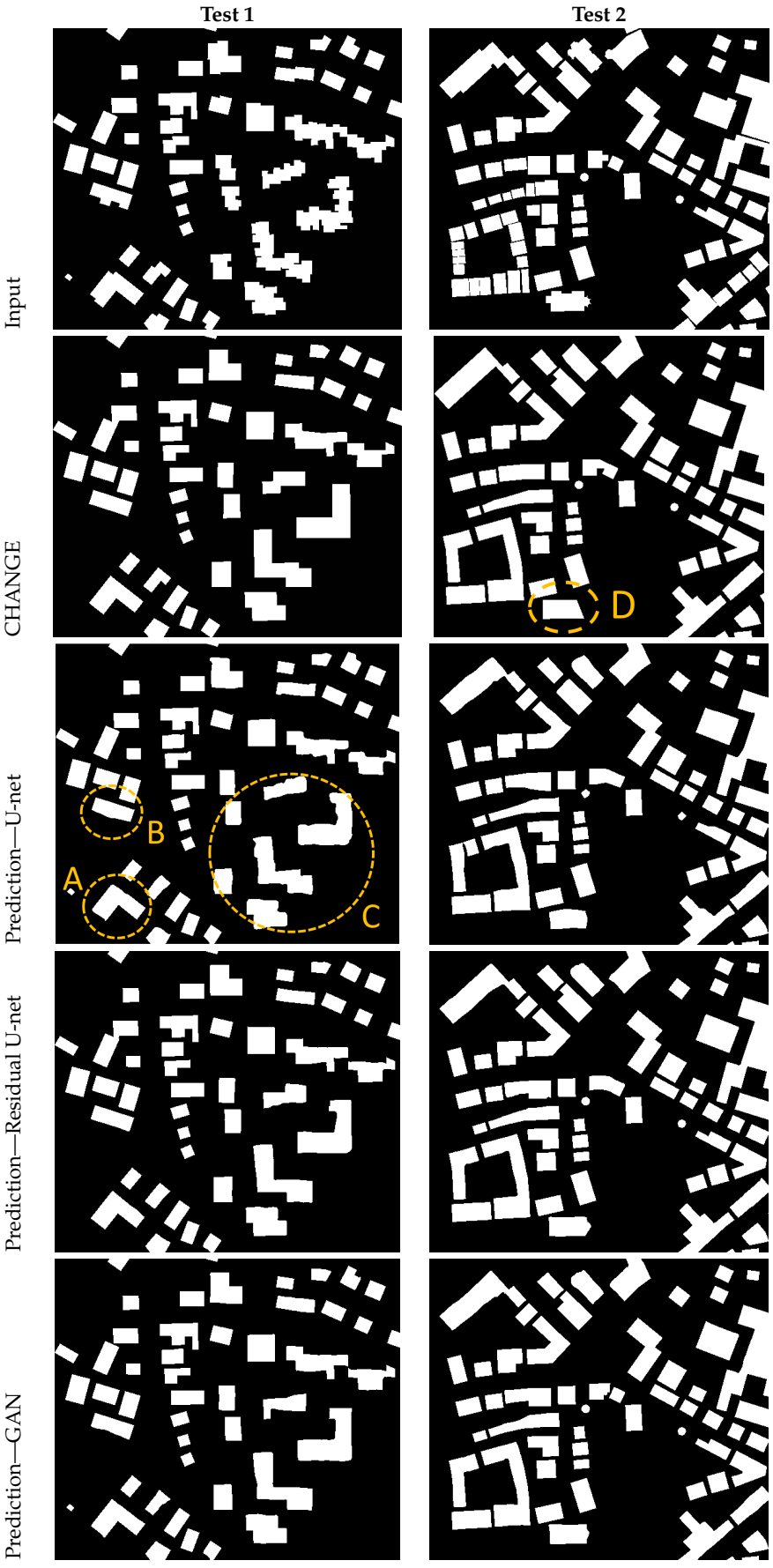

Figure 10. Evaluation of different models on independent test at map scale 1:15,000.

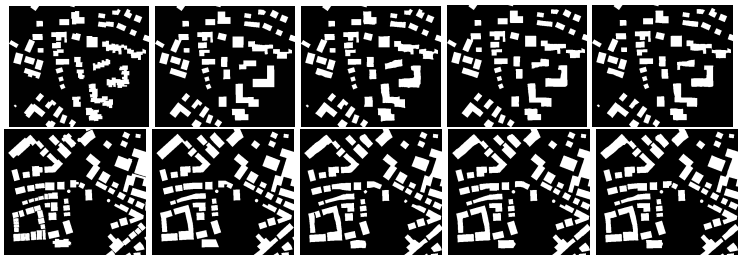

Figure 11. Buildings shown at original scale 1:15,000, from top to bottom: tests 1 and 2; from left to right: input image, target image, U-net prediction, residual U-net prediction, and GAN prediction.

5.3. Experiments with Specific Buildings and Orientations

In order to evaluate the behaviour of the network with respect to typical building shapes, a test series of buildings has been produced. All the buildings have offsets and extrusions of different extents. In this test, also the potential dependency on the orientation was investigated by rotating the shapes in four different directions. The results are shown in Figures 12 and 13. Please note that the size of the buildings is different—the longest dimension of buildings is 80m in Figure 12 and 25 m in Figure 13. The figures visualize the original image on the top and the generalization results in below.

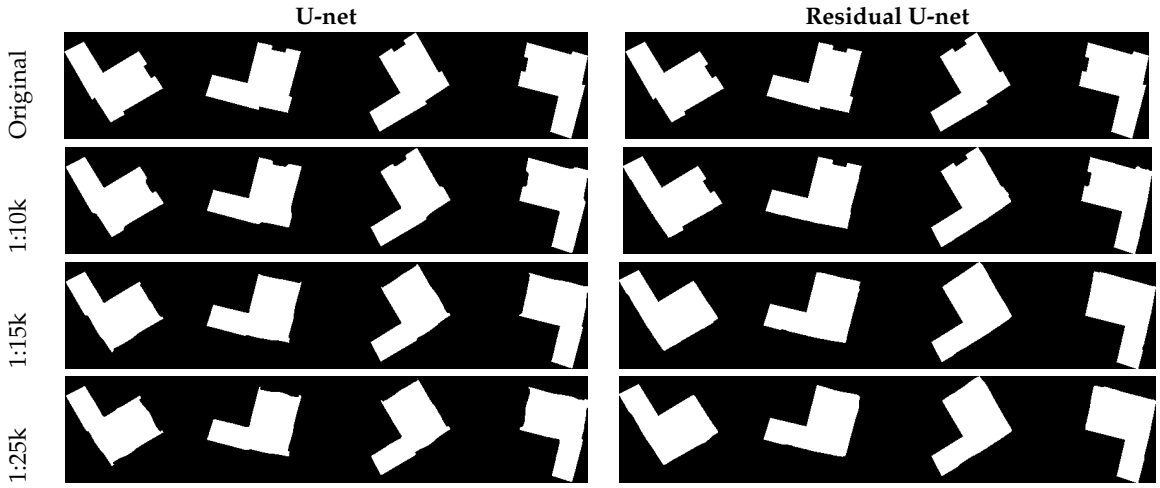

Figure 12. Evaluation of different models on buildings with orientations at three target map scales.

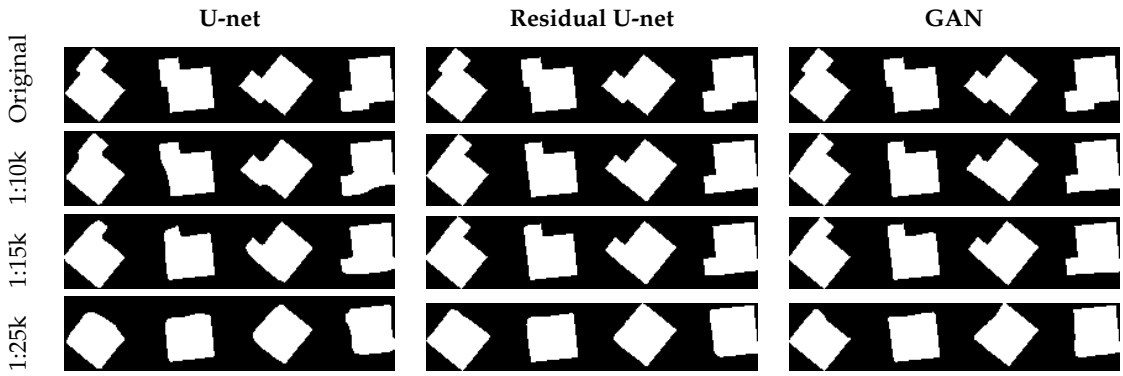

Figure 13. Evaluation of different models on buildings with orientations at three target map scales.

It can be observed that the extrusions and intrusions are continuously eliminated with increasing scale. It is also clear, that the modifications of the building outlines are very subtle for the larger scales, and get more and more visible for the smaller scales. In Figure 12 the offsets vanish with smaller scales, so does the annex in Figure 13. The purpose of this experiment was mainly to check, if there is a dependency on the orientation of the buildings. This is not the case—obviously, the network has

learned enough examples of extrusions, intrusions and offsets in different orientations, and is able to generalize them appropriately. A general observation can be made: compared with U-net and GAN, the residual U-net can generate buildings with much better preserved straight lines and corners.

5.4. Map Series in Different Scales

The following visualizations show the same spatial extent at three different target scales (Figure 14). In the top row, the solutions using CHANGE are given, whereas in the following rows the predictions with the three deep neural networks are given. For a detailed comparison, the same spatial extents at all three scales are visualized with images in the same size. Figure 15 shows the generalization results at the three target map scales in their respective size.

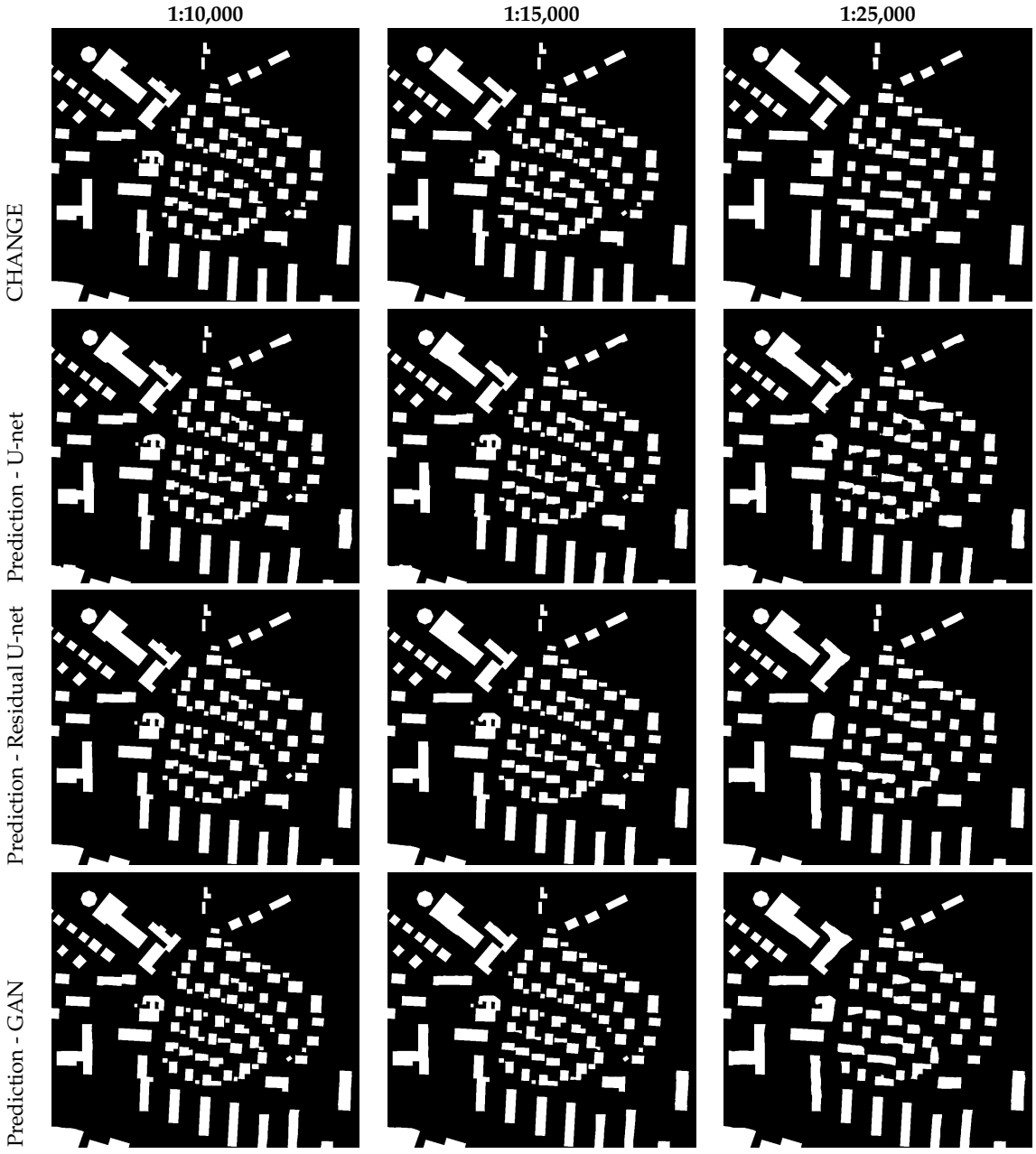

Figure 14. Comparison of building map generalized with CHANGE (**top**) and predicted representation with deep neural networks.

From the results presented above, it is apparent that the models were able to both simplify existing outlines, eliminate too small buildings, and also merge adjacent buildings; i.e., the models contained a combination of different individual operators. The models involving residual blocks showed significantly better results compared with U-net. As for the generalization results at original target size scale (Figure 15), the building outlines were simpler, no tiny extrusions and intrusions disturbed the visual impression.

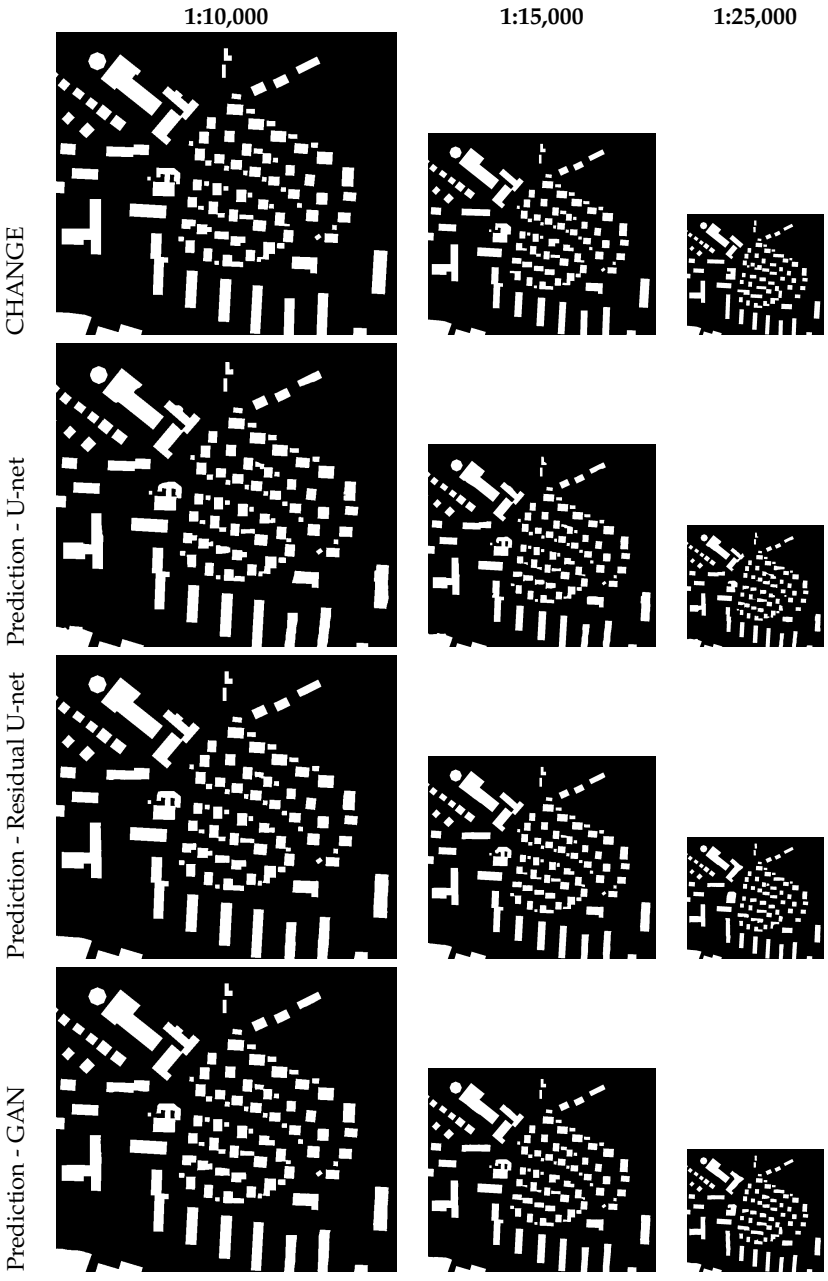

Figure 15. Comparison of building map generalized with CHANGE (**top**) and predicted representation with deep neural networks. Visualization is at the original target scale size.

5.5. Results with a Large Extent of the Data Set

Figure 16 shows the result of a larger area in the target scales 1:10,000, 1:15,000 and 1:25,000, where the same spatial extents are visualized with images in the same size. The extent comprised both residential buildings and industrial buildings, as well as an inner city area. It can be seen that the residual U-net was able to produce appropriate simplifications for the different building

types, i.e., rectangular shaped residential buildings and irregular shaped buildings in the city center, where different individual buildings were merged. It was also visible that small buildings have been eliminated in the smaller scales.

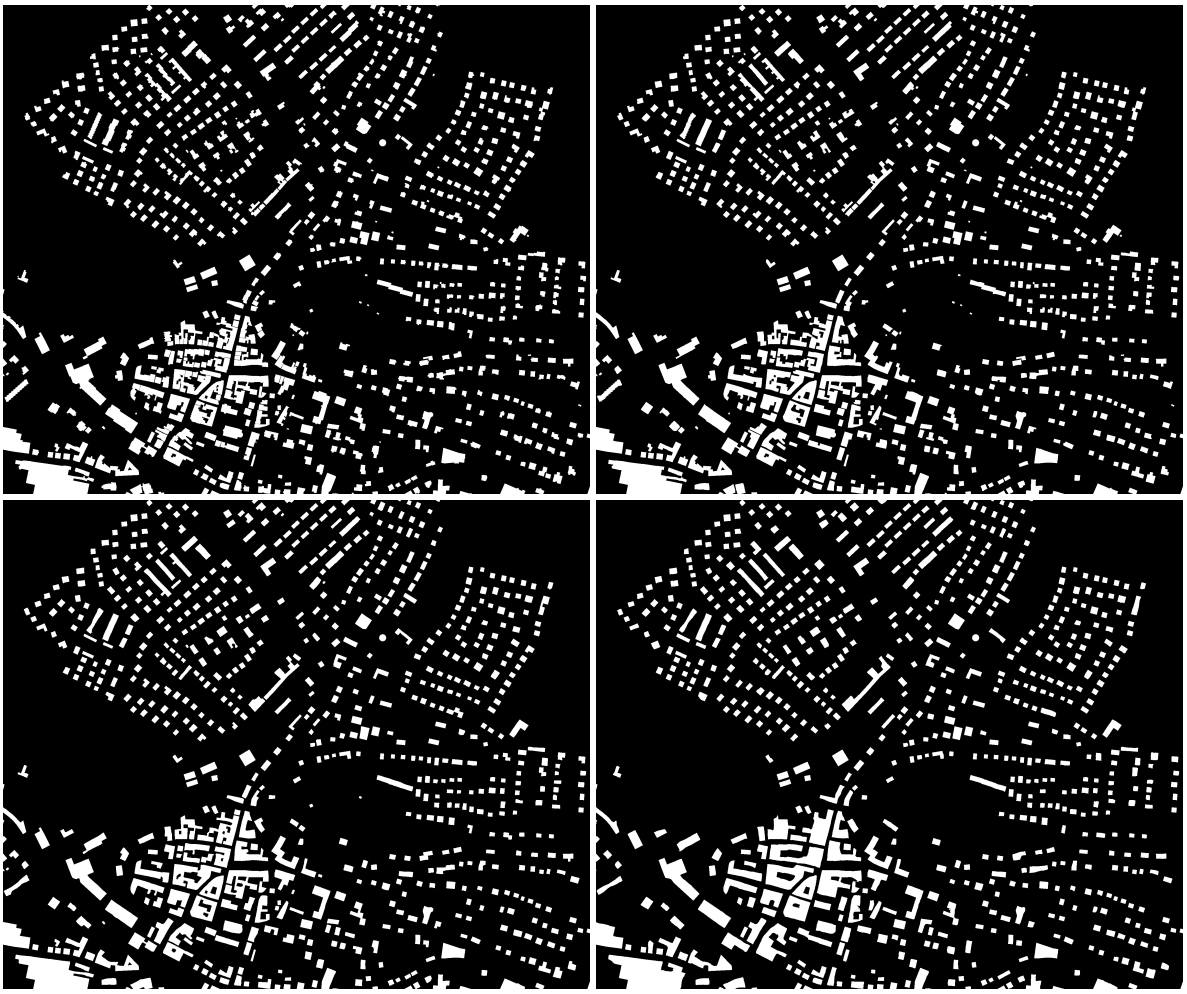

Figure 16. Original map scale (**upper left**), generalization results from residual U-net at map scales 1:10,000 (**upper right**), 1:15,000 (**bottom left**) and 1:25,000 (**bottom right**)—residential and industrial buildings, inner city.

5.6. Discussion

In the case studies shown above the predictions of the neural networks are evaluated both quantitatively and qualitatively. A visual inspection of the results clearly indicated that the networks have learned the simplification of the buildings for the respective scales: small buildings are eliminated in smaller scales; close, neighboring buildings are aggregated, and outlines are simplified by eliminating small extrusions and indentations. This can nicely be observed in Figure 10 test 1, where the buildings with the complicated ground plans in the right part of the image have been simplified to L-shaped outlines.

Our baseline solution U-net can already provide valid simplification for buildings. After combination with the residual units, a significant improvement has been achieved both in the quantitative test and visual inspection. By further introducing a GAN-like discriminator, unfortunately, the network did not improve its performance in the quantitative evaluation. As observed and stated in the work of image-to-image translation [35], GANs are effective at generating outputs of highly detailed or photographic objects and scenes, however, they cannot achieve a better performance than simply using L1 regression for the semantic segmentation task. Since a similar network architecture

for semantic segmentation was used in this work, the previous observation can also be confirmed that simply introducing GAN-like discriminators does not necessarily improve the prediction results in a quantitative evaluation.

Qualitatively, the residual U-net and GAN achieved relatively similar results based on our visual inspection. However, visual inspection is subjective, only a systematic assessment by human map readers can provide a valid comparison. Furthermore, training the GANs is more computational expensive than training a residual U-net. Therefore, we conclude that introducing GAN-like discriminators, such as ImageGAN or PatchGAN, do not significantly contribute to the model performance.

Even though the residual U-net can produce predictions with much clearer outlines, it is still sometimes not as perfect as the classic solutions in vector space. The characteristic features of some of the individual buildings, namely rectangularity and parallelism are not always well preserved or enhanced. In many cases, the models do not generate exact straight lines. These small defects could be cleaned up in a post-processing step, using e.g., an approach proposed by Sester and Neidhart [50], which explicitly takes these constraints into account and optimizes it in an adjustment process.

The visualizations showing the original sizes of the target scales (e.g., Figures 11 and 15) reveal that the results of the predictions can be used for presentation graphics: the generalization—i.e., simplification of the outline, reducing the content, as well as preserving the overall structure of the buildings and building structure—is clearly achieved with the approach.

6. Conclusions

In summary, this work was intended to be a proof-of-concept whether applying or slightly modifying existing DCNNs could achieve satisfying generalization results. The results indicate that this was successful. In this paper, three neural network architectures, namely U-net, residual U-net and GAN, have been evaluated on the cartographic building generalization task. The results show that all these networks could learn the generalization operations in one single model in an implicit way. The residual U-net, which combines the benefits of U-net's skip connection and ResNet's residual unit [51], outperforms the other two types of models. GAN was applied and tested with two types of discriminators and multiple weighting factors. Unfortunately, it could not significantly enforce the generator to produce generalized building patterns with better parallelism and rectangularity as we had hoped. Therefore, the residual U-net is considered as a proper solution for this task.

However, there are several avenues to go in the near future: obviously, the simple L2 loss does not strongly enforce that characteristic building shapes are preserved (such as parallelism, rectangularity). Future work will be devoted to investigating other loss functions or adaptations on current loss function, which could be applied to enforce such constraints. Also, other generalization functions will be tested, such as typification or displacement.

This work focused on learning the cartographic generalization operations in raster space, which needs a conversion between vector and raster. At the moment, we are investigating the ability to generate building maps at different map scales from high resolution areal images, where image segmentation models directly generate outputs in raster. These raster maps are then fed into the learned models to produce building maps in different scales. On the other hand, we will also explore the use of deep learning in vector space by creating a representation of the building outline, which can be represented or simplified using approaches such as long short-term memory (LSTM) [52], or a graph convolutional neural network [53]. A possible representation could be a vocabulary, which was proposed for a streaming generalization approach [54].

Author Contributions: conceptualization, M.S., F.T. and Y.F.; methodology and software, Y.F. and M.S.; validation, Y.F. and M.S.; data curation, F.T. and M.S.; writing—original draft preparation, Y.F. and M.S.; writing—review and editing, Y.F., M.S. and F.T.; visualization, Y.F. and M.S.; supervision, M.S.

Funding: The publication of this article was funded by the Open Access Fund of the Leibniz Universität Hannover.

Acknowledgments: We gratefully acknowledge the support of NVIDIA Corporation with the donation of a GeForce Titan X GPU used for this research.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CNN	convolutional neural network
DCNN	deep convolutional neural network
GAN	generative adversarial network
GPU	graphics processing unit
LSTM	long short-term memory
MAE	mean absolute error
MSE	mean squared error
OSM	OpenStreetMap
ResNet	deep residual network

References

1. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv* **2015**, arXiv:1511.00561.
2. Mackaness, W.A.; Ruas, A.; Sarjakoski, L.T. *Generalisation of Geographic Information: Cartographic Modelling and Applications*; Elsevier: Amsterdam, The Netherlands, 2011.
3. Sester, M. Optimization approaches for generalization and data abstraction. *Int. J. Geogr. Inf. Sci.* **2005**, *19*, 871–897. [CrossRef]
4. Stoter, J.; Post, M.; van Altena, V.; Nijhuis, R.; Bruns, B. Fully automated generalization of a 1:50 k map from 1:10 k data. *Cartogr. Geogr. Inf. Sci.* **2014**, *41*, 1–13. [CrossRef]
5. Renard, J.; Gaffuri, J.; Duchêne, C.; Touya, G. Automated generalisation results using the agent-based platform CartAGen. In Proceedings of the 25th International Cartographic Conference (ICC'11), Paris, France, 3–8 July 2011.
6. Müller, J.C.; Zeshen, W. Area-patch generalisation: A competitive approach. *Cartogr. J.* **1992**, *29*, 137–144. [CrossRef]
7. Jäger, E. Investigations on automated feature displacement for small scale maps in raster format. In Proceedings of the 15th Conference of the ICA, Bournemouth, UK, 22 September–1 October 1991; pp. 245–256.
8. Damen, J.; van Kreveld, M.; Spaan, B. High quality building generalization by extending the morphological operators. In Proceedings of the 11th ICA Workshop on Generalization and Multiple Representation, Montpellier, France, 20–21 June 2008; pp. 1–12.
9. Li, Z.; Yan, H.; Ai, T.; Chen, J. Automated building generalization based on urban morphology and Gestalt theory. *Int. J. Geogr. Inf. Sci.* **2004**, *18*, 513–534. [CrossRef]
10. Shen, Y.; Ai, T.; Li, W.; Yang, M.; Feng, Y. A polygon aggregation method with global feature preservation using superpixel segmentation. *Comput. Environ. Urban Syst.* **2019**, *75*, 117–131. [CrossRef]
11. Powitz, B. Computer-assisted generalization—An important software tool for gis. *Int. Arch. Photogramm. Remote Sens.* **1993**, *29*, 664–673.
12. Weibel, R.; Keller, S.F.; Reichenbacher, T. Overcoming the knowledge acquisition bottleneck in map generalization: The role of interactive systems and computational intelligence. In *COSIT*; Springer: Berlin/Heidelberg, Germany, 1995; pp. 139–156.
13. Mustière, S. GALBE: Adaptive Generalisation. The need for an Adaptive Process for Automated Generalisation, an Example on Roads. In Proceedings of the Geographic Information Systems PlaNet, Lisbon, Portugal, 7–11 September 1998.
14. Sester, M. Knowledge acquisition for the automatic interpretation of spatial data. *Int. J. Geogr. Inf. Sci.* **2000**, *114*, 1–24. [CrossRef]
15. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]

16. Zhong, Z.; Jin, L.; Xie, Z. High performance offline handwritten chinese character recognition using googlenet and directional feature maps. In Proceedings of the International Conference on Document Analysis and Recognition (ICDAR 2015); Nancy, France, 23–26 August 2015; pp. 846–850.
17. Feng, Y.; Sester, M. Extraction of pluvial flood relevant volunteered geographic information (VGI) by deep learning from user generated texts and photos. *ISPRS Int. J. Geo-Inf.* **2018**, *7*, 39. [CrossRef]
18. Feng, Y.; Shebotnov, S.; Brenner, C.; Sester, M. Ensembled convolutional neural network models for retrieving flood relevant tweets. In Proceedings of the MediaEval 2018 Workshop, Sophia Antipolis, France, 29–31 October 2018; p. 27.
19. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
20. Marmanis, D.; Schindler, K.; Wegner, J.D.; Galliani, S.; Datcu, M.; Stilla, U. Classification with an edge: Improving semantic image segmentation with boundary detection. *ISPRS J. Photogramm. Remote Sens.* **2018**, *135*, 158–172. [CrossRef]
21. Chen, K.; Fu, K.; Yan, M.; Gao, X.; Sun, X.; Wei, X. Semantic Segmentation of Aerial Images with Shuffling Convolutional Neural Networks. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 173–177. [CrossRef]
22. Wang, W.; Yang, N.; Zhang, Y.; Wang, F.; Cao, T.; Eklund, P. A review of road extraction from remote sensing images. *J. Traffic Transp. Eng.* **2016**, *3*, 271–282. [CrossRef]
23. Xu, Y.; Chen, Z.; Xie, Z.; Wu, L. Quality assessment of building footprint data using a deep autoencoder network. *Int. J. Geogr. Inf. Sci.* **2017**, *31*, 1929–1951. [CrossRef]
24. Zhou, Q.; Li, Z. A comparative study of various supervised learning approaches to selective omission in a road network. *Cartogr. J.* **2017**, *54*, 254–264. [CrossRef]
25. Lee, J.; Jang, H.; Yang, J.; Yu, K. Machine Learning Classification of Buildings for Map Generalization. *ISPRS Int. J. Geo-Inf.* **2017**, *6*, 309. [CrossRef]
26. Cheng, B.; Liu, Q.; Li, X.; Wang, Y. Building simplification using backpropagation neural networks: A combination of cartographers' expertise and raster-based local perception. *Gisci. Remote Sens.* **2013**, *50*, 527–542. [CrossRef]
27. Thiemann, F.; Cakir, S.; Politz, F. Generalisierung mittels Deep Learning (CNN) am Beispiel der Straßenextraktion aus GPS-Trajektorien. In *38. Wissenschaftlich-Technische Jahrestagung der DGPF und PFGK18 Tagung, Muenchen Publikationen der DGPF, Band 27*; Deutsche Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation (DGPF) e.V.: Munich, Germany, 2018.
28. Regnauld, N.; McMaster, R. A Synoptic View of Generalisation Operators. In *Generalisation of Geographic Information*; Mackaness, W., Ruas, A., Sarjakoski, L., Eds.; Elsevier Science B.V.: Amsterdam, The Netherlands, 2007; pp. 37–66.
29. Sester, M.; Feng, Y.; Thiemann, F. Building generalization using deep learning. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2018**, *XLII-4*, 565–572. [CrossRef]
30. Simo-Serra, E.; Iizuka, S.; Sasaki, K.; Ishikawa, H. Learning to Simplify: Fully Convolutional Networks for Rough Sketch Cleanup. *ACM Trans. Graph.* **2016**, *35*, 121. [CrossRef]
31. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer: Cham, Switzerland, 2015; pp. 234–241.
32. Zhang, Z.; Liu, Q.; Wang, Y. Road Extraction by Deep Residual UNet. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 749–753. [CrossRef]
33. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems 27*; Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2014; pp. 2672–2680.
34. Politz, F.; Sester, M. Exploring ALS and DIM data for semantic segmentation using CNNs. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2018**, *42*, 347–354. [CrossRef]
35. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-To-Image Translation With Conditional Adversarial Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5967–5976.

36. Springenberg, J.T.; Dosovitskiy, A.; Brox, T.; Riedmiller, M. Striving for simplicity: The all convolutional net. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
37. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity mappings in deep residual networks. In Proceedings of the 14th European Conference on Computer Vision, Cham, The Netherlands, 8–16 October 2016; Springer: Berlin, Germany, 2016; pp. 630–645.
38. Chen, Y.; Lai, Y.K.; Liu, Y.J. Cartoongan: Generative adversarial networks for photo cartoonization. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 18–23 June 2018 pp. 9465–9474.
39. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2242–2251.
40. Peters, T.; Brenner, C. Conditional adversarial networks for multimodal photo-realistic point cloud rendering. In Proceedings of the Spatial Big Data and Machine Learning in GIScience, Workshop at GIScience 2018, Melbourne, Australia, 28 August 2018; pp. 48–53.
41. Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4681–4690.
42. Simo-Serra, E.; Iizuka, S.; Ishikawa, H. Mastering sketching: Adversarial augmentation for structured prediction. *ACM Trans. Graph. (TOG)* **2018**, *37*, 11. [CrossRef]
43. Mirza, M.; Osindero, S. Conditional generative adversarial nets. *arXiv* **2014**, arXiv:1411.1784.
44. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein gan. *arXiv* **2017**, arXiv:1701.07875.
45. Kim, T.; Cha, M.; Kim, H.; Lee, J. K.; Kim, J. Learning to discover cross-domain relations with generative adversarial networks. In Proceedings of the 34th International Conference on Machine Learning; Sydney, Australia, 6–11 August 2017; Volume 70, pp. 1857–1865.
46. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. Tensorflow: A system for large-scale machine learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), Savannah, GA, USA, 2–4 November 2016; pp. 265–283.
47. Hinton, G.; Srivastava, N.; Swersky, K. Lecture 6a Overview of Mini-Batch Gradient Descent. Coursera Course Lectures: Neural Networks for Machine Learning. 2012. Available online: http://www.cs.toronto.edu/~hinton/coursera/lecture6/lec6.pdf (accessed on 30 May 2019).
48. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
49. Zeiler, M.D. ADADELTA: An adaptive learning rate method. *arXiv* **2012**, arXiv:1212.5701.
50. Sester, M.; Neidhart, H. Reconstruction of building ground plans from laser scanner data. In Proceedings of the 11th AGILE International Conference on Geographic Information Science 2008, Girona, Spain, 4–8 August 2008.
51. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA , 27–30 June 2016; pp. 770–778.
52. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
53. Yan, X.; Ai, T.; Yang, M.; Yin, H. A graph convolutional neural network for classification of building patterns using spatial vector data. *ISPRS J. Photogramm. Remote Sens.* **2019**, *150*, 259–273. [CrossRef]
54. Sester, M.; Brenner, C. Continuous generalization for visualization on small mobile devices. In Proceedings of the 11th International Symposium on Spatial Data Handling (SDH'04), Leicester, UK, 23–25 August 2004; pp. 355–368.