








# Deep Neural Network Regression for Normalized Digital Surface Model Generation With Sentinel-2 Imagery

Konstantin Müller , Robert Leppich , Christian Geiß , *Member, IEEE*, Vanessa Borst , Patrick Aravena Pelizari , Samuel Kounev , *Member, IEEE*, and Hannes Taubenböck 

**Abstract**—In recent history, normalized digital surface models (nDSMs) have been constantly gaining importance as a means to solve large-scale geographic problems. High-resolution surface models are precious, as they can provide detailed information for a specific area. However, measurements with a high resolution are time consuming and costly. Only a few approaches exist to create high-resolution nDSMs for extensive areas. This article explores approaches to extract high-resolution nDSMs from low-resolution Sentinel-2 data, allowing us to derive large-scale models. We thereby utilize the advantages of Sentinel 2 being open access, having global coverage, and providing steady updates through a high repetition rate. Several deep learning models are trained to overcome the gap in producing high-resolution surface maps from low-resolution input data. With U-Net as a base architecture, we extend the capabilities of our model by integrating tailored multiscale encoders with differently sized kernels in the convolution as well as conformed self-attention inside the skip connection gates. Using pixelwise regression, our U-Net base models can achieve a mean height error of approximately 2 m. Moreover, through our enhancements to the model architecture, we reduce the model error by more than 7%.

**Index Terms**—Deep learning, multiscale encoder, sentinel, surface model.

## I. INTRODUCTION

HEIGHT information as provided by normalized digital surface models (nDSMs) is important for various application fields related, for example, to energy consumption optimization [1], population assessment [2], natural hazard risk assessment [3], [4], or urban planning [5], among others. Both spaceborne remote sensing and airborne remote sensing are

well-established sources to derive nDSM data. The former includes missions such as TanDEM-X (TDM), a radar interferometer, which delivers a digital surface model (DSM) with an unprecedented spatial resolution of 0.4 arcseconds at a global level [6], [7], which seamlessly processes into an nDSM. The latter comprises remotely sensed data from two or more viewing directions or light detection and ranging (LiDAR) measurements to compute nDSM data with a spatial resolution of up to centimeters [8]. However, nDSM data with a fairly high spatial resolution ( $\leq 10$  m) exist solely with spatially fragmented coverage or cannot be queried globally on an open-source basis. The realization of flight campaigns is costly, and the time continuous monitoring of areas to capture changes in the land surface requires further resources.

In contrast to this, we focus on the estimation of nDSM data from globally available Sentinel-2 imagery. Thereby, we introduce a methodology to predict high-resolution height images ( $0.5 \times 0.5$  m) from Sentinel-2. We utilize the multispectral bands of Sentinel-2 with a resolution of  $10 \times 10$  m and investigate whether deep neural networks can overcome this jump in scale of a factor of 20. To this end, we propose a tailored convolutional neural network (CNN) regression architecture, building upon the well-known U-Net model [9], which has proven to perform well across many scientific fields including the interpretation of satellite images. In this context, Zhang et al. [10] have proposed several improved versions of U-Net used for building segmentation in urban areas. Many others have used it not only for urban but also for countryside mapping tasks [11], [12]. The U-Net architecture has mostly been used for segmentation, differentiating between multiple classes of objects. However, U-Net can also be used for pixelwise regression, naturally also in the case of satellite imagery [13].

Our proposed neural network architecture is based on several substantial extensions and enhancements of U-Net: Besides residual connections, we propose a multiscale encoder to tackle the problem that artifacts occurring on the surface differ in size and shape. The multiscale encoder consists of two parallel encoders with different kernel sizes. With this newly used technique of duplicating the encoder with differently sized kernels (i.e., fanning the encoder), we are able to learn differently sized features in both encoders separately. In addition, we integrate attention mechanisms when merging the different encoders, which enhances our model even more. By integrating these

Manuscript received 3 April 2023; revised 25 May 2023 and 2 July 2023; accepted 18 July 2023. Date of publication 21 July 2023; date of current version 22 September 2023. (*Corresponding author: Konstantin Müller.*)

Konstantin Müller, Robert Leppich, Vanessa Borst, and Samuel Kounev are with the Department of Computer Science, Julius-Maximilians-Universität Würzburg, 97070 Würzburg, Germany (e-mail: konstantinfinn.mueller@gmx.de; robert.leppich@uni-wuerzburg.de; vanessa.borst@uni-wuerzburg.de; samuel.kounev@uni-wuerzburg.de).

Christian Geiß and Patrick Aravena Pelizari are with the German Remote Sensing Data Center, German Aerospace Center, 82234 Weßling, Germany (e-mail: christian.geiss@dlr.de; Patrick.AravenaPelizari@dlr.de).

Hannes Taubenböck is with the German Remote Sensing Data Center, German Aerospace Center, 82234 Weßling, Germany, and also with the Earth Observation Research Cluster, Julius-Maximilians-Universität Würzburg, 97070 Würzburg, Germany (e-mail: hannes.taubenboeck@dlr.de).

Digital Object Identifier 10.1109/JSTARS.2023.3297710

new elements, we not only achieve better overall performance with respect to the considered numerical metrics but also attain more coherent outputs regarding the visual representation of the results. The latter can then be combined via a mosaic to generate nDSM data for larger areas.

The rest of this article is organized as follows. Section II presents related work. Section III details the proposed modeling methodology. Section IV presents the deployed datasets and explains the experimental setup. Section V provides experimental results as a validation of our modeling approach, merged with a detailed evaluation and discussion of the results. Finally, Section VI concludes this article.

## II. RELATED WORK

In the past, methods have been proposed to estimate surface height data such as height information from more abundant optical and synthetic aperture radar (SAR) acquisitions. The estimation problem has been dominantly modeled as a supervised learning problem. The goal is to find a mapping between an incoming vector (i.e., more ubiquitously available optical and SAR acquisitions) and an observable output (i.e., a training set including nDSM data derived over spatially limited areas). In particular, height estimations from single aerial images with a very high spatial resolution ( $< 1$  m) have been the focus of numerous method-oriented works. The underlying problem has also been mapped to a very well known image translation task [14], that is, monocular depth estimation [15], [16]. Ghamisi and Yokoya [17] deploy conditional generative adversarial networks whose architecture is based on an encoder–decoder network with skip connections (generator) and penalizing structures at the scale of image patches (discriminator). To learn an image-to-nDSM translation rule, the network is trained on scenes where both the nDSM and the optical data are available. Subsequently, the trained network is utilized to estimate elevation information on a single optical image target scene. Paoletti et al. [18] propose an unpaired model (i.e., not requiring aligned pairs of optical nDSM data) based on variational autoencoders and generative adversarial networks to perform image-to-image translation. Besides the aforementioned generative models, discriminative models have also been heavily deployed. This also constitutes the field in which our model takes place. However, first, Amirko-lae and Arefi [19] deploy a CNN-based network for estimating height information. They employ an encoder–decoder network, where the encoding part is using a deep residual network and the decoding part is designed to map the abstract feature maps into height images. Xing et al. [20] design a gated feature aggregation module to effectively combine low- and high-level features for height estimation. Li et al. [21] propose to divide height values into spacing-increasing intervals and model the regression problem as an ordinal regression problem, using also an ordinal loss for network training. Beyond, Recla and Schmitt [22] deploy very high spatial resolution SAR intensity data for DSM generation, and the authors of [23], [24], [25], and [26] model the estimation problem as a multitask optimization objective. They present dedicated networks whereby semantic segmentation is integrated into the nDSM regression task. This enables the use

of a shared backbone network that can extract complementary features from each objective to improve the performance of the individual task.

In parallel to the above, approaches have been developed that focus on large-scale height estimates based on supervised learning methods: Geiß et al. [27] use automatically compiled built-up height information either from TDM surface height data [28], [29] or from cadastral sources and design a multi-strategy ensemble regression approach to map built-up heights on an urban neighborhood scale from Sentinel-2 features. Subsequently, also a multitasking model is designed to jointly estimate built-up height and built-up density in a beneficial manner [30]. Li et al. [31] use features computed from various optical, SAR, and ancillary geospatial datasets and regress building height with an ordinary random forest approach for  $1 \times 1$  km grid cells across Europe, the USA, and China. Frantz et al. [32] combine Sentinel-1 and Sentinel-2 time-series data with very high resolution 3-D building models, mapping building heights for Germany on a 10-m grid with an ordinary support vector regression model. Cao and Huang [33] build upon ZY-3 multi-view images to estimate building height at a spatial resolution of 2.5 m for 42 Chinese cities with a network that internalizes multispectral, multiview, and multitask properties. Concerned with the natural environment, Lang et al. [34] employ a CNN to regress countrywide canopy height for Gabon and Switzerland from Sentinel-2 images, using reference values obtained from airborne LiDAR scans and photogrammetric stereo matching as training data.

## III. MODELING APPROACH

In this section, we present our U-Net-based architecture. Taking the original U-Net model of Ronneberger et al. [9] as a starting point, we propose several enhancements to improve its ability to detect and predict artifacts and their respective heights. In total, we propose and evaluate three different variants of the architecture: a baseline U-Net with only small modifications (V1), a multiscale fanned U-Net (V2), and an attention-enriched multiscale fanned U-Net (V3). Thereby, one architecture builds upon and extends the previous one in a progressive way. For illustration, Fig. 1 shows the final model, that is, the attention-enriched fanned U-Net.

### A. Baseline U-Net (V1)

The architecture V1 refers to a baseline U-Net similar to the one proposed by Ronneberger et al. [9], that is, it follows an encoder–decoder structure with skip connections between corresponding levels of the encoder and the decoder path. By means of these shortcuts, valuable spatial information from all encoding stages is available during the upscaling process. In more detail, the network uses double convolution blocks with a kernel size of  $3 \times 3$  at each level of the encoder to derive the respective feature maps, where the current level output of the two blocks is saved and passed along the skip connection to the corresponding decoder block. Whenever we perform such a convolution operation, we use a *reflection padding* of size one for the incoming image or feature map in order to keep the spatial

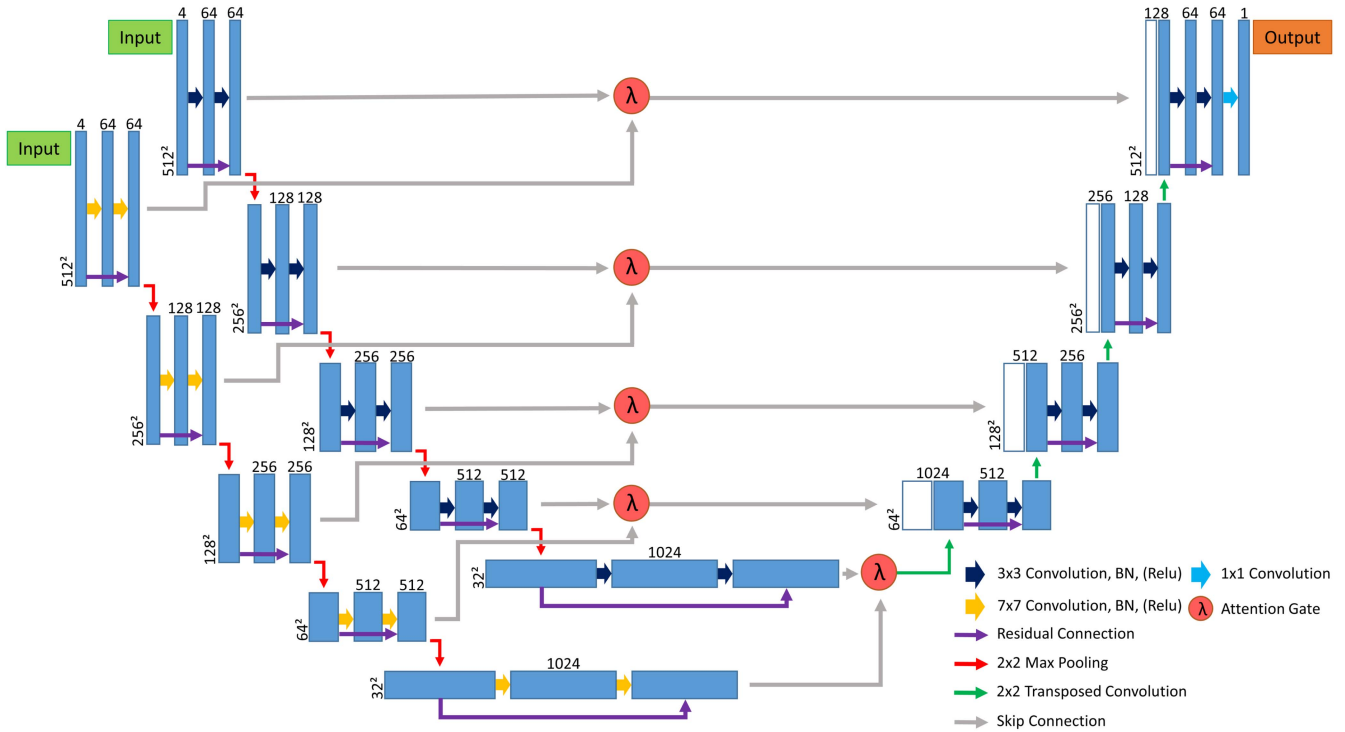


Fig. 1. Schematic diagram of the proposed fanned U-Net with soft attention gates along the skip connections between the encoder and the decoder part of the network as well as at the bottleneck. The contraction path consists of two parallel encoders with different kernel sizes, where each set of convolution blocks, BN, and ReLU activations is supported by an additional residual connection depicted in purple. In the expansion path, the decoder first fuses the information of the corresponding outputs of each of the two encoder components by means of a soft attention gate, and afterward, it upsamples the feature maps with a transposed convolution while halving the number of channels. At each decoder level, the information from the level below and the output of the soft attention gate belonging to the corresponding encoder level is concatenated and further processed by two convolution-BN-ReLU sequences. Again, additional residual connections allow the network to bypass these two blocks. After the double convolution, a transposed convolution increases the size of the feature maps while reducing the number of channels by half. The feature maps are displayed as blue rectangles with the channel amount being indicated on the top and the spatial dimensions being displayed on the bottom left.

dimensions and avoid shrinking. Furthermore, we employ batch normalization (BN) and a rectified linear unit (ReLU) activation after every convolution layer. After every encoder component containing the two previously described convolution-BN-ReLU blocks, a max-pooling layer with kernel size  $2 \times 2$  follows.

At the bottleneck (i.e., after the last encoder layer), the up-scaling process starts. Here, a transposed convolution layer with a  $2 \times 2$  kernel and a stride of two is employed for increasing the spatial dimensions of the feature maps while halving the number of channels. In addition, a set of two  $3 \times 3$  convolution blocks follows after each upscaling step. Repeating this operation flow several times, the size of the feature maps is successively increased up to the original input size. In the end, an additional  $1 \times 1$  convolution layer is used after the two final convolution blocks. This aims at attaining a single-channel output feature map, which is required for performing pixelwise regression.

Despite the similarities to the original U-Net version, a major difference regarding our architecture lies in the introduction of additional residual connections, which we place around the double convolution blocks at each level of both the encoder and the decoder, as depicted in Fig. 2. According to He et al. [35], such residual blocks allow the network to become more complex and deeper without losing reference to earlier learned features by offering the possibility to bypass one or more blocks via identity skip connections. Driven by the successes achieved by

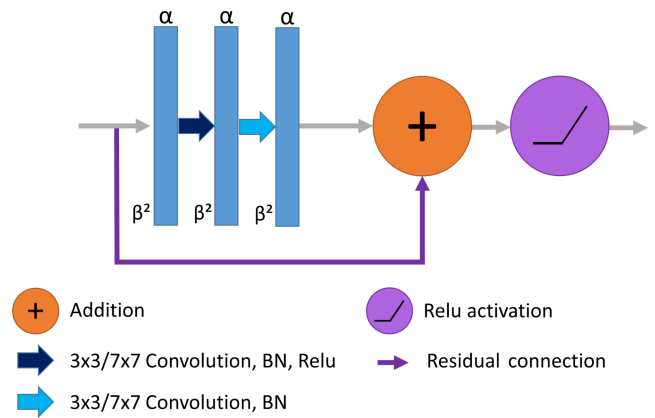


Fig. 2. Structural representation of a residual block. Each block uses a skip connection to allow the feature maps to bypass the subsequent convolutional operations, where the copied and unchanged maps are added to the convolutional output afterward. While for the first convolution, BN and a ReLU activation follow directly, for the second, the ReLU function is applied only after the addition.  $\alpha$  denotes the amount of channels and  $\beta$  the width/height of the tensor.

employing this concept in deep neural networks developed in many different scientific domains, we save the incoming feature maps before the double convolution block and add them again to its output after the last BN operation but before the final ReLU activation (cf. Fig. 2).

### B. Fanned U-Net (V2)

To tackle the problem of many different-sized and shaped artifacts that occur on the earth’s surface, we introduce a multiscale encoder (V2), that is, we fan the encoder part out. To this end, we mirror the encoder described above, where the mirrored version differs from the original one in that a different kernel size of  $7 \times 7$  is used for the convolution filters. Moreover, for the bigger kernels, a *reflection padding* of size 3 is employed, while the padding size of one is maintained for the encoder path with  $(3 \times 3)$ -sized kernels.

To connect both encoder components, we combine their learned feature maps by adding them up. This applies to the horizontal skip connections from each encoder level to the corresponding decoder level as well as to the bottleneck, that is, we also add the feature maps of the two encoder paths after the last encoder layer. Instead of an addition, we also implemented concatenation as a merge operation, but this did not lead to model improvements and was, thus, not further considered.

The concept of using several parallel encoders was proposed earlier by us in the context of time-series data [36]. With this approach, more different features can be learned by enabling the network to better extract patterns of different resolutions or sizes. In our context, it can be imagined as using several different scan sizes for attaining different abstraction levels, with the  $3 \times 3$  kernels focusing on local patterns (i.e., smaller objects like corners) and the  $7 \times 7$  kernels capturing more global patterns (i.e., larger objects like longer axes of buildings). Each encoder component alone would either oversee and, thus, not learn smaller artifacts (only  $7 \times 7$  kernels) or fail to account for the entire object (only  $3 \times 3$  kernels).

### C. Attention Enriched Fanned U-Net (V3)

In recent years, the concept of attention has become increasingly popular. Since the initial introduction of the attention mechanism by Bahdanau et al. [37], the concept has been continuously developed (e.g., multihead attention proposed by Vaswani et al. [38]) and has become more and more established in the deep learning domain. During the learning process, attention helps the network to identify and focus on important parts of the input image. In our case, the model will greatly benefit from taking into account important features such as buildings and vegetation more than less important ones such as bare areas or flat fields.

Technically, the adding operations of fanned U-Net (V2), that is, the ones along the horizontal skip connections of each encoder–decoder level as well as at the bottleneck, are replaced by the so-called attention gates. In this way, we can assign importance weights to the features at each encoding stage. A graphical illustration of the attention-enriched version of the model (V3) is shown in Fig. 1, while Fig. 3 provides details about the realization of the attention gates themselves.

Analogously to the previous multiscale version (V2), each attention gate first adds up the learned feature maps that are passed along the two skip connections from both the encoders. The result is cached via a skip connection, and in parallel, it is further processed as follows: First, a ReLU function is

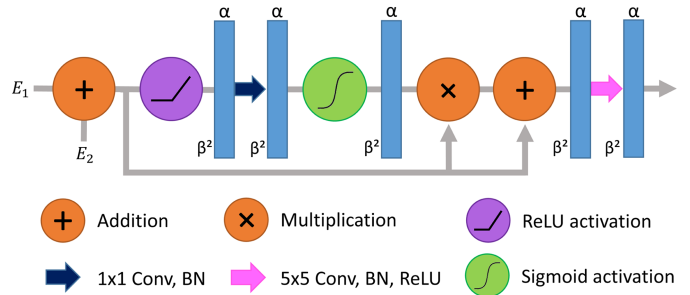


Fig. 3. Lambda attention gate consisting of a sequence of several layers. The gate takes the outputs of the double convolution block of both encoders ( $E_1$  and  $E_2$ ) as input, adds them up, and retains the result for later uses in the form of a skip connection. In parallel, the result is further processed by a ReLU function, a  $1 \times 1$  convolution with BN, and a Sigmoid activation. Afterward, the gate performs a multiplication based on the attention weights to merge both paths, that is, the saved result of the addition and the one of the further processing pipeline. The skip connection is then used again to re-add the original feature maps to the attention-generated output. Finally, a final  $5 \times 5$  convolution is performed. The amount of channels  $\alpha$  is indicated on top of the blue rectangles, and the size  $\beta$  (i.e., the width and height of the feature maps) is at the bottom left.

applied, followed by a  $1 \times 1$  convolution to break down the number of channels to one. For retrieving the attention weights, a Sigmoid activation function is then employed to distinguish between high and low importance while avoiding infinitely high values. After a transposed convolution to restore the original amount of channels, we merge both paths by multiplying the output of the processing path with the result of the previous summation. Finally, a BN layer is used to complete the attention gate. Overall, the attention-driven combination of the learned feature maps of both encoder components aims to highlight important artifacts while suppressing activations of less relevant unoccupied fields.

## IV. EXPERIMENTAL SETUP

This section starts by introducing the two main datasets used for the experimental evaluation of our modeling approach; it then provides a detailed description of the conducted experiments, i.e., the applied preprocessing steps and the experimental setup. Afterward, the evaluation results are presented in the next section, and the overall potential of the proposed methodology is discussed.

### A. Datasets

In this work, two different datasets are mainly used, namely, an nDSM dataset of North Rhine-Westphalia (NRW) and a matching Sentinel-2 dataset of the same region.

- 1) *NRW nDSM data*: We acquire the nDSM data for NRW from the open data platform of the Institute of Information and Technology of North Rhine-Westphalia [39]. The dataset contains 35.860 tiles with a size of  $1 \text{ km}^2$ , where each tile has a resolution of  $2000 \text{ px} \times 2000 \text{ px}$ , that is, one pixel covers an area of  $0.5 \text{ m} \times 0.5 \text{ m}$ . More specifically, the data were obtained as the difference between an image-based nDSM and an airborne laser scanner-based



digital terrain model. In total, the calculated nDSM yields a horizontal and a vertical error of  $\pm 5$  dm. For detailed information, we refer to [40].

- 2) *NRW multispectral Sentinel-2 data*: The Sentinel-2 data are provided by the German Aerospace Center. To consistently account for the highest spatial resolution provided by Sentinel-2, we only use the four  $10 \times 10$  m bands, i.e., 2 (blue), 3 (green), 4 (red), and 8a (NIR) from all 13 bands available. The dataset contains Level 3A surface reflectance derived from the WASP processor [41], which, in turn, utilizes Level 2A products. The latter consists of surface reflectance corrected for atmospheric effects and clouds by means of the MAJA processor [42]. Given that each of these four tiles has a size of  $10\,980 \text{ px} \times 10\,980 \text{ px}$ , this results in an area coverage of more than  $12.000 \text{ km}^2$  per tile.

### B. Preprocessing

Before the data can be inserted tile by tile into the neural network, we apply several preprocessing steps, where the downloaded nDSM data serve as the baseline, and the corresponding Sentinel bands are retrieved during the rest of the preprocessing process: First, the Sentinel-2 and the nDSM data are aligned with respect to their coordinate system. Subsequently, we split each of the ( $2000 \text{ px} \times 2000 \text{ px}$ )-sized nDSM tiles into 16 smaller pieces of size  $500 \text{ px} \times 500 \text{ px}$  to decrease the computational effort on the hardware side.

In the next step, the corresponding Sentinel excerpts have to be identified for each of these 16 subtiles per nDSM tile. Apart from the location based on the coordinates, the matching of the time component is important. Since the nDSM tiles were measured independently every month from 2016 to 2022, we need to choose the Sentinel excerpts for every nDSM subtile according to the nDSM measurement time stamps in order to achieve a matching for every month. This is necessary since an nDSM tile measured in a particular month (e.g., August) would not match the vegetation heights of the Sentinel data for another month (e.g., January). After searching for temporally and spatially matched Sentinel sections for a given  $500 \text{ px} \times 500 \text{ px}$  nDSM subtile, we retrieve the Sentinel data belonging to band 2 (blue), 3 (green), 4 (red), and 8a (NIR), while the remaining Sentinel bands are ignored. At this point, one could argue to use more channels to have more features available in the original input. However, Lang et al. [34] found that using only the combination of the R, G, B, and NIR bands performs almost equally well if not often better than using all bands in terms of vegetation. Furthermore, the remaining bands of Sentinel provide only even coarser resolutions than  $10 \text{ m} \times 10 \text{ m}$ , leading to the fact that only very coarse objects would also benefit from additional channels, while computational efforts increase exponentially. However, the resolution of the selected Sentinel bands ( $10 \text{ m} \times 10 \text{ m}$ ) differs from the one of the nDSM data ( $0.5 \text{ m} \times 0.5 \text{ m}$ ). Thus, the lower resolution Sentinel data have to be upscaled to the higher spatial resolution of the nDSM tiles. For that purpose, we use spline interpolation with nearest neighbor resampling order [43], as we work with discrete data. Besides this, we also

experimented with bilinear interpolation and cubic convolution; however, the nearest-neighbor-based approach worked best in preserving realistic transitions between different elevated pixels of the image.

In this way, we obtain a set of quadruples consisting of a ( $500 \text{ px} \times 500 \text{ px}$ )-sized nDSM subtile and the four corresponding ( $500 \text{ px} \times 500 \text{ px}$ )-sized sentinel sections, each of the four having a resolution of ( $0.5 \text{ m} \times 0.5 \text{ m}$ ) after resampling. In the next preprocessing step, we then mirror the edges of each quintuple element by six pixels on each side, yielding tiles with a size of  $512 \text{ px} \times 512 \text{ px}$ , that is, power-of-two-sized input images, which are subsequently processed independently. This approach follows the overlap-tile strategy of Ronneberger et al. [9], who proposed this method to be able to seamlessly segment arbitrary large images and prevent the loss of pixels when applying pooling operations at deeper levels of the network. However, losses and results will be calculated using only the original  $500 \text{ px} \times 500 \text{ px}$  image size through center cropping.

Combining all these steps, we obtain 520.368 spatially and temporally aligned quintuples, each containing a high-resolution nDSM tile and four matching Sentinel crops. We store each tuple into a compressed.npz file, which makes the data manageable for further processing.

In the last preprocessing step, the 520.368 extracted quintuples were subjected to some correction routines, filtering out 5.828 of them. Hence, the final number of quintuples available for the subsequent training, validation, and testing amounts to 514.540. In more detail, we first filtered out measurement and transformation errors in the nDSM dataset itself by searching for nonlogical patterns like tiles containing only zero values, values within an unrealistic range (difference between the min and the max value of more than 400), too small values (at least one value  $< -50$ ), or a too high proportion of negative pixel values (i.e., more than 20% of values  $< -12$ ). Notably, for the tiles that contain negative values in a ratio below the 20% threshold, all negative values are set to 0 on-the-fly during processing by the neural network.

Apart from the constraints mentioned above, we filter out quadruples containing nonrectangular subtiles; that is, we remove nDSM data without a complete match in the Sentinel tiles. This may occur in the case when the nDSM coordinates are on the edge of the Sentinel tiles, and therefore, the nDSM tile can only be partially matched. Finally, further errors are sorted out manually by visually inspecting suspicious images.

As a summary for this section, we include Fig. 4. It sums up the steps we take from matching the nDSM with the Sentinel-2 data to the final processed and filtered dataset.

### C. Implementation Details

Regarding the technical realization, we use the Python-based PyTorch framework on the software side combined with a partitioned NVIDIA A100 graphics card on the hardware side, with 40 GB of dedicated memory available for the experiments. For evaluating and tuning our network, several metrics are utilized. In this work, we mainly use the mean absolute error (MAE) and the root-mean-square error (RMSE) metrics, given that

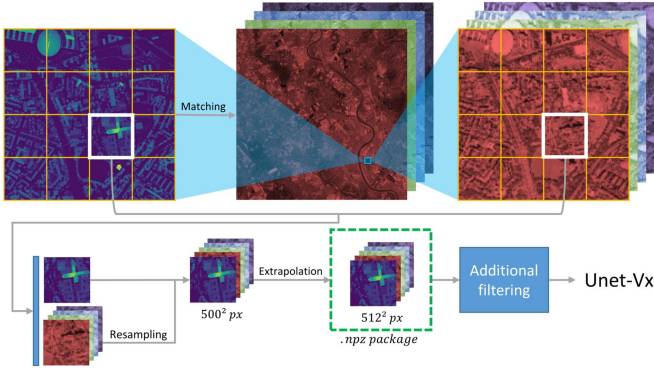


Fig. 4. Preprocessing chain that we use to acquire our samples to feed into the network. From the upper left on, we start by matching an nDSM tile spatially and timewise to a larger Sentinel scene. After cropping its respective part, we split the tiles into 16 smaller pieces. We then resample the Sentinel-2 crops to match the pixel size of the nDSM and join all layers to a quintuple. Moving on, we extrapolate the quintuple in order to receive a width and height of a power of two. With additional filtering of value-specific errors, we then feed the .npz packages into either U-Net-V1, U-Net-V2, or U-Net-V3 (U-Net-Vx).

they are widely used in related work (see, e.g., [26] and [44]). Their mathematical definition is given in (1), where  $x$  denotes a pixel of our prediction and  $y$  represents the ground truth value. By means of the index parameter  $i$ , we iterate over the whole image consisting of  $N$  pixels in total. With respect to the implementation, the TorchMetrics library (domain “regression”) is used [45]

$$\begin{aligned} \text{MAE} &= \frac{1}{N} \sum_{i=1}^N |x_i - y_i| \\ \text{RMSE} &= \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - y_i)^2}. \end{aligned} \quad (1)$$

In addition to these two metrics, we use the structural similarity index (SSIM) [46], again using the TorchMetrics library (domain “image”) for the implementation [47]. While neither MAE nor RMSE indicates whether pixels belong to the same region or object but rather considers each pixel independently, the SSIM explicitly takes this into account via a sliding window approach. By analyzing individual structures within each window, human-perceivable visual features and arrangements can be captured. With the goal of creating elevation maps that are easily recognizable and processable by humans, we systematically test the different approaches to maximize these accuracy measures.

In more detail, the SSIM performs each computation based on a window with index  $i$ , as defined in (2). Here, we choose a window size of 5, which corresponds to the average of the  $(3 \times 3)$ - and  $(7 \times 7)$ -sized convolution kernels of the two encoder components used. In the formula,  $\mu_{i_{x/y}}$  and  $\sigma_{i_{x/y}}$  denote the local mean and standard deviation of the respective windows  $x$  and  $y$  at position  $i$ , respectively,  $\sigma_{i_{xy}}$  represents the covariance between both windows, and  $C1$  and  $C2$  are small constants to stabilize the division with small denominators

$$\text{SSIM} = \frac{(2\mu_{i_x}\mu_{i_y} + C1)(2\sigma_{i_{xy}} + C2)}{(\mu_{i_x}^2 + \mu_{i_y}^2 + C1)(\sigma_{i_x}^2 + \sigma_{i_y}^2 + C2)}. \quad (2)$$

To further evaluate the performance of our network, we calculate the zero-mean normalized cross-correlation (ZNCC) [48]. Unfortunately, despite being used in an increasing amount of papers, there are no ZNCC implementations in standard libraries yet. Thus, we implement it from scratch, following the definition of Ghamisi and Yokoya [17] given in (3). In our approach, we add a safety factor  $\epsilon$  to avoid encountering any division by 0

$$\text{ZNCC} = \frac{1}{n} \sum_{i=1}^n \frac{1}{\sigma_x \sigma_y + \epsilon} (x_i - \mu_x)(y_i - \mu_y). \quad (3)$$

Finally, to obtain a metric stable against outliers and hard-to-predict tiles, we also calculate the absolute median error [MedAE; (4)] for each image in the test set

$$\text{MedAE} = \text{median} \left( |x_1 - y_1|, \dots, |x_N - y_N| \right). \quad (4)$$

Again,  $x$  and  $y$  both denote the prediction and ground truth value at position  $i$ , which iterates from index 1 to the number of pixels  $N$ . To derive a single metric score for the entire test set, we finally average the respective medians of the images contained within.

As a loss function, we selected the  $L1Loss$  function as implemented in Python [49], whose definition equals the MAE metric shown in (1). We also added other metrics like the structural similarity in the form of a combined loss function instead of solely relying on MAE. However, these metrics were found to provide worse results. Finally, we use 70% of the dataset for training, 20% for validation, and 10% for testing purposes. Assigning the quadruples randomly to the different subsets, we ensure an equal distribution of the different area types (urban, suburban, countryside, industrial, and so on). During training, we use an early stopping approach based on the validation loss with a patience of five training and validation alternations. Moreover, PyTorch [50] is employed as our optimizer, using an initial learning rate of 5e-06 and a weight decay of 5e-04. We provide our code of the implementation details and the neural network architectures via GitHub at [51]. For completeness, we train the famous Adabins architecture by Bhat et al. [52] as a comparison to our models on our dataset. In Section I, we introduced our problem also as one of the fields of depth estimation. Since Adabins is in this context and has an encoder–decoder structure similar to U-Net as its baseline, it suits best for comparison. However, a few tweaks are necessary to fit our data into the model. We modify the first convolution layer to handle a fourth image channel (NIR) by increasing its channel input from three to four. Furthermore, Adabins does not allow a one-on-one comparison regarding the size of its input and output images, so we upsample the outcome  $(256 \times 256)$  to the original input size of  $(512 \times 512)$  in a bilinear fashion. Finally, due to hardware limitations, we set the number of adaptive bins to 25.

With the setup described, we finally input our preprocessed Sentinel-2 patches with a size of  $512 \times 512$  pixels and the corresponding four image channels with a  $10 \times 10$  m resolution of Sentinel-2 (R, G, B, NIR). We receive an nDSM prediction and compare it via the loss function to its matching nDSM ground truth tile.

TABLE I  
RESULTING MEAN OF THE EARLY STOPPING SELECTED EPOCH OF THE METRICS AND INDICES FOR V1, V2, V3, AND THE ADABINS COMPARISON

Metric/Index	V1–Baseline	V2–Multiscale	V3–Attention	Adabins
Training set				
MAE ↓	2.175	2.027	<b>2.003</b>	3.089
RMSE ↓	4.799	4.566	<b>4.530</b>	6.230
SSIM ↑	0.587	0.601	<b>0.604</b>	0.588
ZNCC ↑	0.612	0.642	<b>0.646</b>	0.601
Validation set				
MAE ↓	2.226	2.081	<b>2.066</b>	3.083
RMSE ↓	4.841	<b>4.602</b>	<b>4.602</b>	6.226
SSIM ↑	0.590	0.599	<b>0.602</b>	0.590
ZNCC ↑	0.611	<b>0.646</b>	0.645	0.606
Test set				
MAE ↓	2.222	2.079	<b>2.065</b>	3.073
RMSE ↓	4.843	4.610	<b>4.608</b>	6.210
SSIM ↑	0.590	0.598	<b>0.602</b>	0.590
ZNCC ↑	0.608	<b>0.644</b>	0.643	0.605
MedAE ↓	1.062	0.961	<b>0.940</b>	2.044

All values are rounded to the 3rd decimal. Furthermore, we indicate the direction of improvement for each row with little arrows. We mark in bold the best scores for each set and metric, respectively.

## V. EVALUATION AND DISCUSSION

In this section, we outline the results of our experiments with all three versions of our proposed model.

In Table I, we present the results of all three model variants with the train, validation, and test dataset. We calculated the MAE, RMSE, SSIM, and ZNCC metrics for all experiments. For the test run, we additionally calculated MedAE, which we discuss later. Variant V1, as a plain U-Net model with minor adaptations for our use case, serves as the baseline model in our experiments. As presented in Table I, it achieves an MAE of 2.2 m on the test set. Meanwhile, our modified versions of U-Net, V2 and V3, improve on all metrics in comparison to the baseline model. With them, the absolute difference in the accuracy of which they predict elevation is almost exactly 2 m. Therefore, already, the numerical point of view underlines the fundamental validity of our methodology in the application domain of surface model creation. For a visual impression, we depict several tiles of different land cover types in Fig. 5.

However, besides excellent results, we observe general challenges for all model variants. Mostly, the network cannot predict small single-surface objects such as transmission towers or stand-alone trees. This is due to the fact that with low input data resolution of  $10 \times 10$  m, such structures are only present in the ground truth data, given that it is of high resolution. Hence, those few but special structures are ignored by our model not seeing an indicator for a structure in the input, leading to a significant performance decrease in our metrics. This is underlined by the fact that the MedAE error values in Table I are only half the size of the original MAE, stressing the impact of this case. For completeness, we depict the mentioned “pinpoint” problem in Fig. 6.

### A. Multiscale Improvement

Another challenge we observe is the prediction of large human-made structures. These mostly include large industrial halls. Extensive vegetation across large areas instead is detected seamlessly. The multiscale V2 version of U-Net shows enhancements consistently across all metrics. Moreover, despite the overall better numeric results, the multiscale approach mainly supports the visual domain realm. Given the usage of larger kernels, V2 predicts areas of larger contiguous heights more accurately. Here, the bigger kernels come into play, as they can learn larger scaled features. In addition, through bigger convolution areas, more pixels are combined into one mass. This clears out further disturbances we observe in V1, as large height areas form tissue-like bugs. We depict such a scenario in Fig. 7.

Nonetheless, we still discover the need for our U-Net-based model to detect large human-made structures in a more reliable manner, even predicting with V2. In some cases, there is no significant difference in the spectral features between human-made flat-roofed structures and natural structures. To be more precise, we suspect the spectral features of flat-roofed large buildings to be similar to those of a ground area, mainly when greening was used for the roof. Furthermore, the roof spectral responses match the ones of asphalt. This leads to false classifications of areas as ground or flat vegetation and, thus, to false height estimations. To overcome this issue, using different bands with better reflection properties in materials of mentioned structures could increase the classification performance of the network. Another solution to this issue would be incorporating auxiliary data, such as cadastral data.

### B. Attention Effect

To further support the height prediction accuracy of the V2 multiscale encoder, we implemented self-attention inside of the skip connections, resulting in V3. As clearly observable in Table I, V3 exhibits better performance over almost every metric in every dataset compared to V2. Despite the improved numerical realm, we again observe struggles in areas with large human-made buildings, as discussed previously.

We, therefore, again depict the example already described in Fig. 7. However, this time, we put aside the prediction of V3, as shown in Fig. 8. Furthermore, we provide an attention map (with 256 channels) taken out of the V3 network in order to see what the model is focusing on and why it struggles.

As shown, the attention mechanism is heavily biased by the spectral properties of the input. The above example depicts a clear difference in the roof’s material as the lower part is metal, and the upper part consists of dark, rough roof plates. Hence, the focus is on only the lower part of the building, preventing our large  $7 \times 7$  encoder kernels from capturing the elevated area.

### C. External Comparison

In comparison to our models, Adabins performs well regarding the SSIM and ZNCC values, even outperforming V1 in some cases. However, it lacks behind in the absolute and squared meter



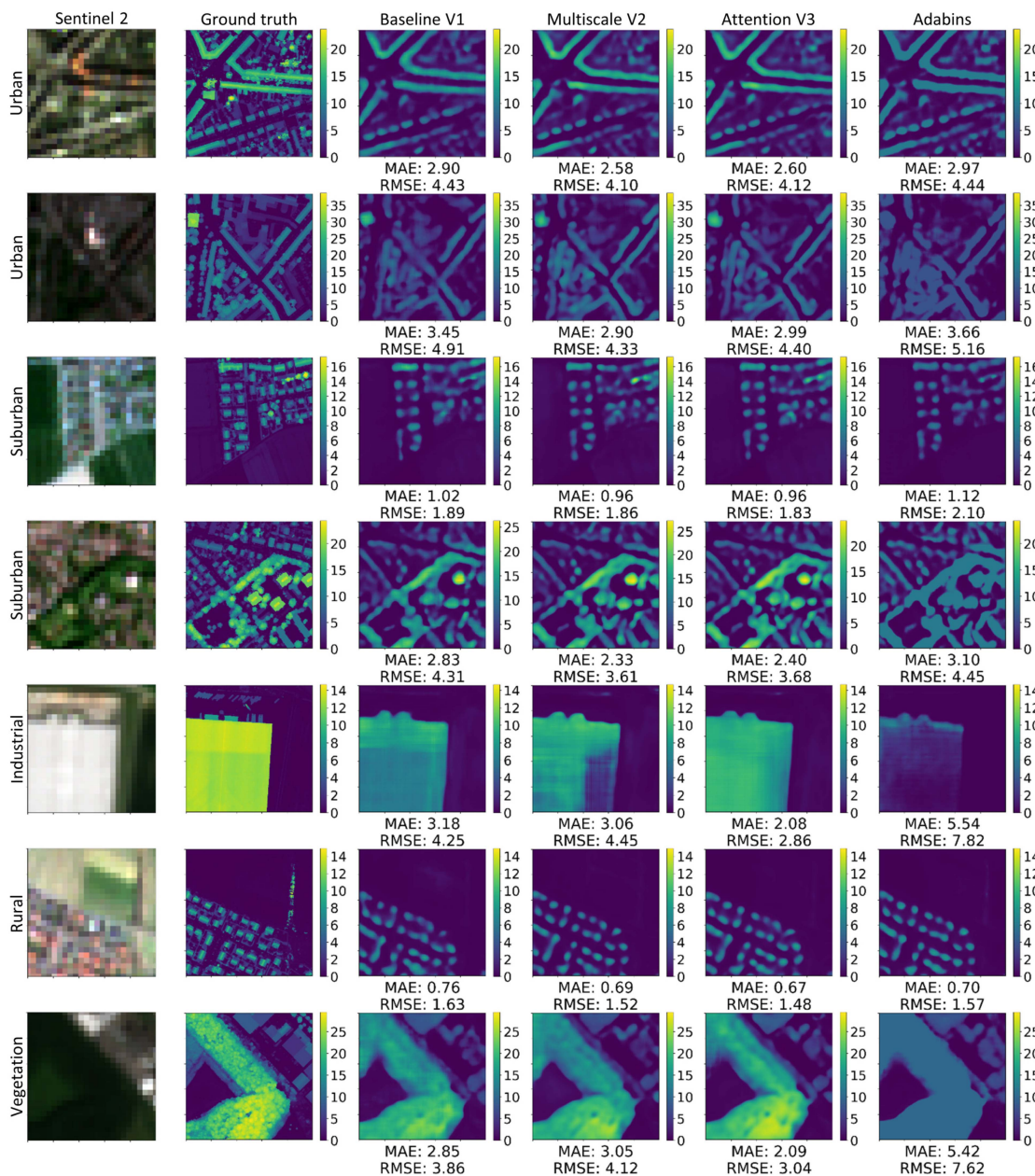


Fig. 5. We depict the predictions of our model variants V1-baseline, V2-multiscale, and V3-attention with their input data and ground truth tile, respectively. We thereby depict imagery from different types of landscapes. Note that all three models are producing decent results despite the low-resolution input data. From left to right: Input Sentinel-2 data ( $10 \times 10$  m), ground truth nDSM ( $0.5 \times 0.5$  m) image as well as the V1, V2, V3, and Adabins prediction. All predictions also have a spatial resolution of  $0.5 \times 0.5$  m.

error metrics. The adaptive thresholding module of Adabins seems so to complement the similarity of the input and output. Nevertheless, V2 and V3 also clearly perform better not only in the case of the raw height information but also on both similarity metrics SSIM and ZNCC. Again, this shows the power of our multiscale and attention components. The numerical deficit of the MAE and RMSE values especially unfolds in the visual domain, as shown in Fig. 5. Here, the network predicts too low values in almost all examples. Moreover, through its adaptive binning module, it seems not to differentiate between height changes within one shape itself. One here could argue that we only use 25 bins when building the architecture due to

computational limits, as the standard is 100 bins in the original implementation. Anyhow, even with 25 bins, the model took 25% longer to train per epoch than our computationally heaviest version V3-attention. We conclude that just increasing the bins without tweaking the architecture itself more is not reasonable.

#### D. Morphological Class Differentiation

As shown in Table I, V2 performed almost as good on the MAE metric as V3, only lacking roughly 0.02 m in the absolute error domain. Moreover, also better or equal scores can be achieved with the ZNCC metric in the training and testing



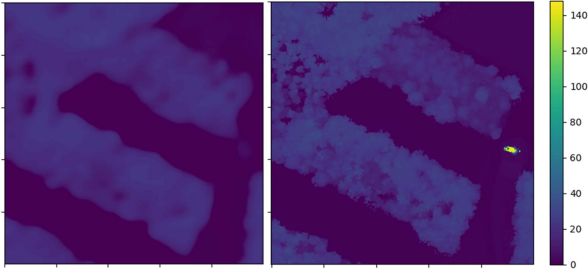


Fig. 6. Prediction for a vegetation area with a “pinpoint” structure, which could be a transmission tower. On the right-hand side, it shows the ground truth containing the tower with around 140 m. The prediction of V1 on the left side thereby yields high error metric values. This is due to the fact that the input is originally sampled with  $10 \times 10$  m, which means that it might actually not even contain the pinpoint structure. Clearly, V2 and V3 will produce similar results, as they are fed with the same input.

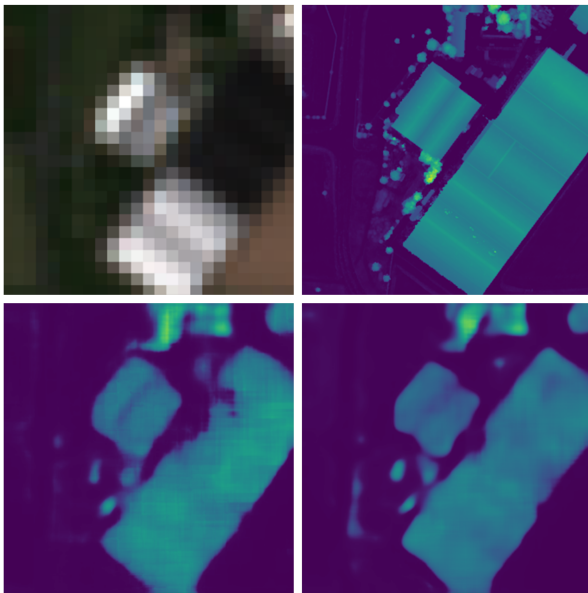


Fig. 7. Upper row is filled with the input data (left) and the ground truth (right). Then, the lower left picture depicts Base U-Net (V1) prediction on a larger building. It misses out on parts of it and creates tissue-like disturbances. On the contrary, the lower right image shows multiscale U-Net prediction on the same tile. Due to bigger kernels included in the structure, it clears out tissue-like disturbances. In addition, it is able to learn larger features allowing it to predict bigger geometric forms. In this example, it predicts the long lines (edges of the building) correctly. However, V1 discontinues the line and falls apart.

set. Considering a reduced amount of two million trainable parameters and a decreased training time of 6 h per epoch of V2 compared to V3 revealed the effectiveness of our multiscale encoder. To furthermore investigate on the behavior that one version of either V2 or V3 is not clearly outperforming the other, we differentiate our result calculations via discrete classes, which describe the morphology of the environment. We, thus, measure each metric for the individual classes, respectively.

Therefore, we separate the height and density of the ground truth tiles into three classes, respectively. For the actual separation, we calculate the distribution of both and use the Jenks natural breaks [53] algorithm to figure out where to set up boundaries for class distinction. We thereby use the implementation in Python of [54]. The distribution of the height  $\hat{h}$  for each tile is



Fig. 8. We depict (from left to right) the input for a large industrial hall in the east of Krefeld, the prediction of V2-attention, the attention map taken (256 channels), and the Google Maps reference image.

defined as a 0.95 quantile, as described in the following equation:

$$\hat{h} = Q_{0.95}(target). \quad (5)$$

For the distribution of the build-up density  $\hat{d}$ , we calculate all pixels being elevated above 1 m over all pixels of the image. The threshold of 1 m is used to decrease fluctuations in very small vegetation heights or agricultural fields. Moving human-made objects like cars are already filtered out of the provided nDSM. We depict a more detailed formula as follows:

$$\hat{d} = \frac{\sum_{i=1}^N (height(i) > 1)}{N}. \quad (6)$$

We denote the amount of all pixels in the target as  $N$  and the height value of a pixel at position  $i$  as  $height(i)$ . As a result, we receive nine different classes out of the combinations of low, medium, high density, and low, medium, high height. We can then calculate our results for each class separately and depict the difference between V2 and V3 in a heatmap in Fig. 9. As the metrics have different ranges, we utilize the  $z$ -score (standard-score) [55] for comparing the single metrics in one heatmap. This allows a fair comparison of different model variants and metrics. In the figure, positive values state an improvement of the V2 model over the V3. Moreover, we depict our formula used for each metric, respectively, as follows:

$$z_{m,c} = \frac{(x_{m,c} - \mu_m)}{\sigma_m}. \quad (7)$$

We denote our  $z$ -score for metric  $m$  and class  $c$  as the value of the metric minus its mean divided by the standard deviation.

Here, we observe a clear picture from the heatmap in Fig. 9. With a lower height density of the tile, V2 slightly improves over V3. However, with increasing height density, V3 outperforms V2. Therefore, overall, the attention mechanism can do more

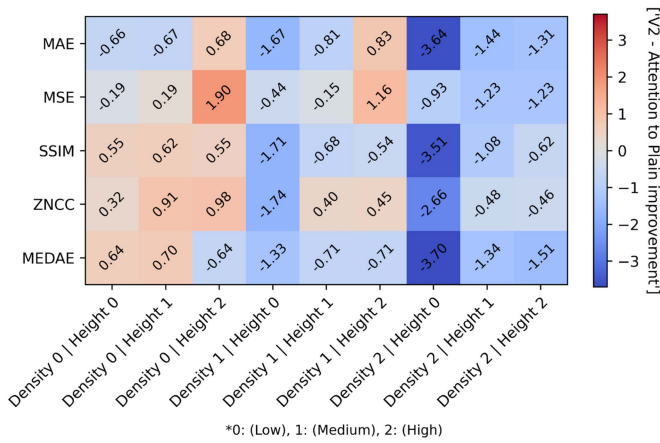


Fig. 9. Metrics calculated on the result set and differentiated by the combinations of the three different density and height borders, yielding nine classes in total. A positive value indicates an improvement in this specific class and metric of V2 over V3.

work, as with higher density, more areas are available to focus on if spectral signals are not disrupting the encoder, as shown previously in Fig. 8. Furthermore, we obtain V3 scores best if the tile is of the class “Height 0” (Low). V2 shows the smallest improvements in those cases, and V3 the strongest. The height class distribution is determined by the 95th quantile  $\hat{h}$ . This yields the fact that if the height is low ( $\hat{h} < 8.187$  m, defined by the natural Jenks algorithm), the attention mechanism detects low-level surface structures that V2 is not able to capture. Hence, this effect enlarges with higher density, resulting in even stronger numeric improvements.

## VI. CONCLUSION

Our models proved to be able to predict high-resolution nDSM images from low-resolution optical satellite data. Indeed, deep neural networks are capable to handle this scalene jump. Furthermore, our results revealed the effectiveness of both our developed multiscale encoder and our attention gates in the application domain of neural-network-driven nDSM creation. As mentioned, with better generalization, we pursue the goal of building up an nDSM on national scale through low-resolution input data. This map can further be improved through the use of auxiliary data. Especially, cadastral footprint data would be of great interest for data fusion opportunities to fix building outlines for future works. This would also help even smaller individual objects to be detected by the network, although it is not visible in the Sentinel-2 data. However, implementing auxiliary data into our methodology is not trivial and will require careful revision of multiple challenges. Hence, this task would be a predestined future work manuscript. Furthermore, since common resampling methods did not improve our results, one could investigate on supersampling by artificial intelligence algorithms to help overcome the dilemma of the coarse input resolution. As an exemplary workflow, we suggest [56]. Furthermore, as mentioned in Section V, an adaptive threshold module as proposed in the Adabins comparison model could potentially be beneficial for the similarity of the input and output

images. Given more computational power to handle a higher number of bins, one could also be attached to our multiscale and attention-enhanced models.

To conclude our work, despite its struggles with certain building types, our developed models proved to provide very good results, especially from the viewpoint of the input it gets. Finally, to once more demonstrate the potential of our models, we reflect back to Fig. 5, depicting our results for different types of land cover.

## REFERENCES

- [1] C. Geiß et al., “Remote sensing-based characterization of settlement structures for assessing local potential of district heat,” *Remote Sens.*, vol. 3, no. 7, pp. 1447–1471, 2011.
- [2] M. Sapena, M. Kühnl, M. Wurm, J. E. Patino, J. C. Duque, and H. Taubenböck, “Empiric recommendations for population disaggregation under different data scenarios,” *PLoS One*, vol. 17, no. 9, pp. 1–29, 2022, doi: [10.1371/journal.pone.0274504](https://doi.org/10.1371/journal.pone.0274504).
- [3] C. Geiß et al., “Benefits of global Earth observation missions for exposure estimation and earthquake loss modelling—Evidence from Santiago de Chile, Chile,” *Nat. Hazards*, pp. 1–26, 2022, doi: [10.1007/s11069-022-05672-6](https://doi.org/10.1007/s11069-022-05672-6).
- [4] C. Geiß et al., “Estimation of seismic building structural types using multi-sensor remote sensing and machine learning techniques,” *ISPRS J. Photogrammetry Remote Sens.*, vol. 104, pp. 175–188, 2015.
- [5] B. Stone and M. Rodgers, “Urban form and thermal efficiency: How the design of cities influences the urban heat island effect,” *J. Amer. Plan. Assoc.*, vol. 67, pp. 186–198, 2001.
- [6] G. Krieger et al., “TanDEM-X: A satellite formation for high-resolution SAR interferometry,” *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 11, pp. 3317–3341, Nov. 2007.
- [7] C. Geiß, M. Wurm, M. Breunig, A. Felbier, and H. Taubenböck, “Normalization of TanDEM-X DSM data in urban environments with morphological filters,” *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 8, pp. 4348–4362, Aug. 2015.
- [8] B. Sirmacek, H. Taubenböck, P. Reinartz, and M. Ehlers, “Performance evaluation for 3-D city model generation of six different DSMs from air- and spaceborne sensors,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 1, pp. 59–70, Feb. 2012.
- [9] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, 2015, pp. 234–241.
- [10] P. Zhang, Y. Ke, Z. Zhang, M. Wang, P. Li, and S. Zhang, “Urban land use and land cover classification using novel deep learning models based on high spatial resolution satellite imagery,” *Sensors*, vol. 18, no. 11, 2018, Art. no. 3717.
- [11] Z. Han et al., “Comparing fully deep convolutional neural networks for land cover classification with high-spatial-resolution Gaofen-2 images,” *ISPRS Int. J. Geo-Inf.*, vol. 9, no. 8, 2020, Art. no. 478.
- [12] A. Stojan, V. Poulain, J. Inglada, V. Poughon, and D. Derksen, “Land cover maps production with high resolution satellite image time series and convolutional neural networks: Adaptations and limits for operational systems,” *Remote Sens.*, vol. 11, no. 17, 2019, Art. no. 1986.
- [13] W. Yao, Z. Zeng, C. Lian, and H. Tang, “Pixel-wise regression using U-Net and its application on pansharpening,” *Neurocomputing*, vol. 312, pp. 364–371, 2018.
- [14] X. Liu, D. Hong, J. Chanussot, B. Zhao, and P. Ghamisi, “Modality translation in remote sensing time series,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5401614.
- [15] D. Eigen, C. Puhrsch, and R. Fergus, “Depth map prediction from a single image using a multi-scale deep network,” in *Proc. 28th Annu. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2366–2374.
- [16] C. Godard, O. M. Aodha, M. Firman, and G. J. Brostow, “Digging into self-supervised monocular depth estimation,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2018, pp. 3827–3837.
- [17] P. Ghamisi and N. Yokoya, “IMG2DSM: Height simulation from single imagery using conditional generative adversarial net,” *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 5, pp. 794–798, May 2018.
- [18] M. E. Paoletti, J. M. Haut, P. Ghamisi, N. Yokoya, J. Plaza, and A. Plaza, “U-IMG2DSM: Unpaired simulation of digital surface models with generative adversarial networks,” *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 7, pp. 1288–1292, Jul. 2021.

- [19] H. A. Amirkolae and H. Arefi, "Height estimation from single aerial images using a deep convolutional encoder-decoder network," *ISPRS J. Photogrammetry Remote Sens.*, vol. 149, pp. 50–66, 2019.
- [20] S. Xing, Q. Dong, and Z. Hu, "Gated feature aggregation for height estimation from single aerial images," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 6003705.
- [21] X. Li, M. Wang, and Y. Fang, "Height estimation from single aerial images using a deep ordinal regression network," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 6000205.
- [22] M. Recla and M. Schmitt, "Deep-learning-based single-image height reconstruction from very-high-resolution SAR intensity data," *ISPRS J. Photogrammetry Remote Sens.*, vol. 183, pp. 496–509, 2021.
- [23] M. Carvalho, B. L. Saux, P. Trouvé-Peloux, F. Champagnat, and A. Almansa, "Multitask learning of height and semantics from aerial images," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 8, pp. 1391–1395, Aug. 2020.
- [24] W. Liu, W. Zhang, X. Sun, Z. Guo, and K. Fu, "HECR-Net: Height-embedding context reassembly network for semantic segmentation in aerial images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 9117–9131, 2021.
- [25] W. Liu, X. Sun, W. Zhang, Z. Guo, and K. Fu, "Associatively segmenting semantics and estimating height from monocular remote-sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5624317.
- [26] J. Lu and Q. Hu, "Semantic joint monocular remote sensing image digital surface model reconstruction based on feature multiplexing and inpainting," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4411015.
- [27] C. Geiß, H. Schrade, P. A. Pelizari, and H. Taubenböck, "Multistrategy ensemble regression for mapping of built-up density and height with Sentinel-2 data," *ISPRS J. Photogrammetry Remote Sens.*, vol. 170, pp. 57–71, 2020.
- [28] C. Geiß, P. A. Pelizari, S. Bauer, A. Schmitt, and H. Taubenböck, "Automatic training set compilation with multisource geodata for STM generation from the TanDEM-X DSM," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 3, pp. 456–460, Mar. 2020.
- [29] C. Geiß et al., "Large-area characterization of urban morphology—Mapping of built-up height and density using TanDEM-X and Sentinel-2 data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 8, pp. 2912–2927, Aug. 2019.
- [30] C. Geiß, E. Brzoska, P. A. Pelizari, S. Lautenbach, and H. Taubenböck, "Multi-target regressor chains with repetitive permutation scheme for characterization of built environments with remote sensing," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 106, 2022, Art. no. 102657.
- [31] M. Li, E. Koks, H. Taubenböck, and J. van Vliet, "Continental-scale mapping and analysis of 3D building structure," *Remote Sens. Environ.*, vol. 245, 2020, Art. no. 111859.
- [32] D. Frantz et al., "National-scale mapping of building height using Sentinel-1 and Sentinel-2 time series," *Remote Sens. Environ.*, vol. 252, 2021, Art. no. 112128.
- [33] Y. Cao and X. Huang, "A deep learning method for building height estimation using high-resolution multi-view imagery over urban areas: A case study of 42 Chinese cities," *Remote Sens. Environ.*, vol. 264, 2021, Art. no. 112590.
- [34] N. Lang, K. Schindler, and J. D. Wegner, "Country-wide high-resolution vegetation height mapping with Sentinel-2," *Remote Sens. Environ.*, vol. 233, 2019, Art. no. 111347.
- [35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 770–778.
- [36] R. Leppich, "Pre-training of deep transformer encoders for time series representation models," M.S. thesis, Dept. Comput. Sci., Julius-Maximilians-Universität Würzburg, Würzburg, Germany, 2021.
- [37] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. 3rd Int. Conf. Learn. Representations*, Y. Bengio and Y. LeCun, Eds., San Diego, CA, USA, 2015.
- [38] A. Vaswani et al., "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.
- [39] Institute of Information and Technology of North Rhine-Westphalia, "Normalized digital surface model 50," 2016–2022. [Online]. Available: <https://www.opengeodata.nrw.de/produkte/geobasis/hm/>
- [40] Institute of Information and Technology of North Rhine-Westphalia, "Normalized digital surface model—Product description," 2016–2022. [Online]. Available: [https://www.bezreg-koeln.nrw.de/brk\\_internet/geobasis/hoehenmodelle/digitale\\_oberflaechenmodelle/normalisiertes\\_digitales\\_oberflaechenmodell/index.html](https://www.bezreg-koeln.nrw.de/brk_internet/geobasis/hoehenmodelle/digitale_oberflaechenmodelle/normalisiertes_digitales_oberflaechenmodell/index.html)
- [41] *Sentinel-2 MMSI—Level 3A (MAJA/WASP Tiles)*, German Aerosp. Center, Cologne, Germany, 2022. [Online]. Available: <https://geoservice.dlr.de/data-assets/4hcq6dkgkj648.html>
- [42] *Sentinel-2 MSI—Level 2A (MAJA/WASP Tiles)*, German Aerosp. Center, Cologne, Germany, 2022. [Online]. Available: <https://geoservice.dlr.de/data-assets/ifczsszckcp63.html>
- [43] The SciPy Community, "scipy.ndimage.zoom—Documentation," 2022. [Online]. Available: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.ndimage.zoom.html>
- [44] C.-J. Liu, V. A. Krylov, P. Kane, G. Kavanagh, and R. Dahyot, "IM2ELEVATION: Building height estimation from single-view aerial imagery," *Remote Sens.*, vol. 12, p. 2719, Aug. 2020, doi: [10.3390/rs12172719](https://doi.org/10.3390/rs12172719).
- [45] *Torchmetrics*, PyTorch Lightning, New York, NY, USA, 2022. [Online]. Available: <https://torchmetrics.readthedocs.io/en/stable/>
- [46] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
- [47] scikit image, "Structural similarity index." Accessed: Jan. 15, 2022. [Online]. Available: [https://torchmetrics.readthedocs.io/en/stable/image/structural\\_similarity.html](https://torchmetrics.readthedocs.io/en/stable/image/structural_similarity.html)
- [48] L. Di Stefano, S. Mattoccia, and F. Tombari, "ZNCC-based template matching using bounded partial correlation," *Pattern Recognit. Lett.*, vol. 26, no. 14, pp. 2129–2134, 2005.
- [49] PyTorch, "L1Loss," 2022. [Online]. Available: <https://pytorch.org/docs/stable/generated/torch.nn.L1Loss.html>
- [50] PyTorch, "Adam," 2022. [Online]. Available: <https://pytorch.org/docs/stable/generated/torch.optim.Adam.html>
- [51] KonstiDE, "DSMCreation," 2023. [Online]. Available: <https://github.com/KonstiDE/DSMCreation>
- [52] S. F. Bhat, I. Alhashim, and P. Wonka, "AdaBins: Depth estimation using adaptive bins," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 4008–4017.
- [53] G. Jenks, *Optimal Data Classification for Choropleth Maps* (Occasional Paper). Lawrence, KS, USA: Univ. Kansas, 1977.
- [54] M. Viry, "jenkspy 0.3.2," 2022. [Online]. Available: <http://github.com/mthh/jenkspy>
- [55] J. Urbano, H. Lima, and A. Hanjalic, "A new perspective on score standardization," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 1061–1064, doi: [10.1145/3331184.3331315](https://doi.org/10.1145/3331184.3331315).
- [56] T. Lu, J. Wang, Y. Zhang, Z. Wang, and J. Jiang, "Satellite image super-resolution via multi-scale residual deep neural network," *Remote Sens.*, vol. 11, no. 13, 2019, Art. no. 1588.



**Konstantin Müller** received the B.Sc. degree in deep learning architectures for satellite imagery in 2021 from the Department of Computer Science, Julius-Maximilians-Universität Würzburg (JMU), Würzburg, Germany, where he is currently working toward the M.Sc. degree in the eagle program with the Department of Remote Sensing under the EAGLE (Earth Observation and Geoanalysis of the Living Environment) project.

He did a semester abroad in Karlskrona, Sweden. He was a Software Engineer in Leipzig, Germany.

His research interests include the combination of data science, deep learning, and geospatial data.



**Robert Leppich** received the bachelor's and master's degrees in computer science from the Julius-Maximilians-Universität Würzburg (JMU), Würzburg, Germany, in 2018 and 2021, respectively.

He is currently a Doctoral Researcher with the Chair of Software Engineering, Julius-Maximilians-Universität Würzburg (JMU). His research interests include machine-learning- and deep-learning-based time-series analysis with a particular focus on natural time series in medicine and sports science.





**Christian Geiß** (Member, IEEE) received the M.Sc. degree in applied geoinformatics from the Paris Lodron University of Salzburg, Salzburg, Austria, in 2010, and the Ph.D. degree (Dr. rer. nat.) in geography from the Humboldt University of Berlin, Berlin, Germany, in 2014. He is currently working toward a Habilitation Project in geography with the Julius-Maximilians-Universität Würzburg (JMU), Würzburg, Germany, with the focus on “Collective Sensing Techniques and Artificial Intelligence for Natural Hazard Risk and Impact Assessment.”

Since 2010, he has been with the German Remote Sensing Data Center, German Aerospace Center, Cologne, Germany, where he is currently the Head of the research group “georisks.” In 2017, he was a Visiting Scholar with the Centre for Risk in the Built Environment, University of Cambridge, Cambridge, U.K. Since 2023, he has been a Professor of “georisk assessment with earth observation techniques” with the Department of Geography, University of Bonn, Bonn, Germany. His research interests include the development of machine learning methods for the interpretation of earth observation data, multimodal remote sensing of the built environment, exposure and vulnerability assessment in the context of natural hazards, such as earthquakes, tsunamis, and floods, as well as techniques for automated damage assessment after natural disasters.



**Vanessa Borst** received the B.Sc. and M.Sc. degrees in computer science from the Julius-Maximilians-Universität Würzburg (JMU), Würzburg, Germany, in 2018 and 2021, respectively.

During her master’s studies, she focused on “Intelligent Systems” and also spent a semester abroad with the University of the Basque Country, Bilbao, Spain. Since October 2021, she has been a Doctoral Researcher with the Chair of Software Engineering, Julius-Maximilians-Universität Würzburg (JMU), where she works on computer-aided disease

diagnosis and prognosis through the clinical use of artificial intelligence. Her research interests include the automated analysis of medical images and time series, with a particular focus on the development and application of deep learning techniques.



**Patrick Aravena Pelizari** received the M.Sc. degree in physical geography/environmental systems from Ludwig-Maximilians University, Munich, Germany, in 2013.

In 2013, he joined the German Remote Sensing Data Center, German Aerospace Center, Oberpfaffenhofen, Germany, where he is currently a Member of the research group “georisks.” As a Research Scientist, he has contributed to projects in the fields of humanitarian relief and natural disaster management by developing machine learning algorithms to extract

related geoinformation from Earth observation data. His current research interests include the vulnerability-related characterization of built environments exposed to natural hazards with multimodal geoinformation data (remote and in situ sensing) and deep learning methods.



**Samuel Kounev** (Member, IEEE) received the M.Sc. degree in mathematics and computer science from the University of Sofia, Sofia, Bulgaria, in 2000, and the Ph.D. (Dr.-Ing.) degree in computer science from the Technical University of Darmstadt, Darmstadt, Germany, in 2005.

He is currently a Full Professor and the Chair of Software Engineering with the University of Würzburg, Würzburg, Germany. He has recently coauthored the first textbook on Systems Benchmarking (Springer, 2020). In the area of benchmarking,

he founded the SPEC Research Group, a consortium within the Standard Performance Evaluation Corporation, providing a platform for collaborative research efforts in the area of quantitative system evaluation and analysis. His research is inspired by the vision of self-aware computing systems, to which he has been one of the major contributors shaping its development. His research interests include developing novel methods, techniques, and tools for the engineering of software for building dependable, efficient, and resilient distributed systems, including cloud-based systems, cyber-physical systems, and scientific computing applications.

Prof. Kounev is an Associate Editor for *Performance Evaluation*. He co-founded the ACM/SPEC International Conference on Performance Engineering (ICPE) in 2010 and the IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS) in 2020, for which he has also been serving on the Steering Committees.



**Hannes Taubenböck** received the Diploma from the Ludwig-Maximilians University München, Munich, Germany, in 2004, and the Ph.D. degree (Dr.rer.nat.) and Habilitation from the Julius-Maximilians-Universität Würzburg (JMU), Würzburg, Germany, in 2008 and 2022, respectively, all in geography.

He is currently the Head of the Department of “Georisks and Civil Security,” German Remote Sensing Data Center (DFD), German Aerospace Center (DLR), Cologne, Germany. He is also a Professor with the Chair “Global Urbanization and Remote

Sensing,” Julius-Maximilians-Universität Würzburg (JMU). In 2005, he joined the DFD of the DLR, Weßling, Germany. After a postdoctoral research phase with the Julius Maximilian University of Würzburg from 2007 to 2010, he returned in 2010 to DFD, DLR as a Scientific Employee, where he was the Head of the “City and Society” team from 2013 to 2022. He became a Full Professor with the Julius-Maximilians-Universität Würzburg (JMU), in 2022. His research interests include urban remote sensing topics, from the development of algorithms for information extraction to value adding to classification products for new findings in urban geography.