Combining parallel trained U-Nets for Surface Model Generation with Sentinel-2 Imagery

Konstantin Müller

***Abstract*—Satellite driven surface models constantly gained importance to solve large scale geographic problems of cluster detection, natural catastrophes and various prediction and planning tasks. Furthermore, it is very valuable if high resoluted surface models are available, as they contain a higher information density for specific areas. However, high resolution measurements for generating such are time consuming and not economical. Until today, only a few approaches exist to create high resolution digital surface models for extensive areas. This paper investigates deeper into generating high resolution digital surface models using low resoluted Sentinel-2 imagery. For this, several deep learning models are trained with additional surface data. We thereby utilize the advantages of Sentinel-2 data being open access, having global coverage and providing steady updates through a high repetition rate. Moreover, we investigate on the usage of a Dual-Network approach, where we parallelly train two neural networks contributing to one result. With this we are able to predict the surface height with pixel-wise regression, better than with traditional single network structures; close to three meters in the mean domain. Lastly, we improve visual performance with this technique.**

***Index Terms*—Sentinel-2, Deep Neural Network, U-Net, Surface Model Generation, Dual-Network**

I. Introduction

DIGITAL surface models (DSMs) play an important role in major current decision making. They provide a high information density while still being visually understandable. Hence, they can fasten processes in natural risk assessment [1], urban planning [2] or energy consumption estimations [3]. Especially, high resolution ones, are of great benefit for those challenges. However, such resoluted equal or below one meter are only yet available for single cities or other smaller selected areas of interest. Moreover, the current state of the art literature only refers to higher resolution input data [4], [5]. This paper proposes a methodology to generate high resolution surface models out of low resolution Sentinel-2 imagery. Using this data, we utilize Sentinels temporal resolution, global coverage and open access availability. As a base work, [6] is used. Here, normalized digital surface models (nDSMs) were generated by a deep neural network using Sentinel-2 input tiles and pixel-wise regression. Anyhow, several challenges as inaccurate height estimation and blurred shape prediction of surface objects were detected (Section 5.5, [6]). The biggest change we apply to this, is a dual network approach. We think that splitting the process of shape and height estimation will decrease the overall error, hence separate neural networks can focus on separate features. Furthermore, we imply aiming for sharpened visual results of the shape of surface artifacts, because a separate structure is predicting it. Also, our base work reported shape inconsistency and blurred regressional results as the one major problem. Regardless, this may be limited to some degree by the low input resolution of the Sentinel-2 input data, we still think that improvements can be achieved. As a network architecture of the models we use U-Net [7]. U-Net, traditionally a segmentation network, has proven itself in the field of remote sensing; [8], [9] or [10]. It provides proper results in both binary and multi-class segmentation. Nevertheless, [11] and certainly [6] showed, that also pixel-wise regression can be done with it. Through the just mentioned split approach we create a segmentation and regression problem with the shape and height prediction respectively. We so think that U-Net will be beneficial for the combination of them. Hence, we propose a network where the shape of surface artifacts is predicted by an additional clone network contributing to the result of our baseline network from [6]. Afterwards, we present many metrics validating our approach and investigate on the improvement over the traditional combined model used in our base work. For conclusion, we illustrate our results in a comparative fashion, before further potential and possibilities are discussed.

II. Proposed Methodology

For producing high resolution surface maps, we train to two deep learning model versions based on U-Net; one traditional, combined version and one newly developed split variant. In the following sections II-A and II-B we propose our full architecture based on U-Net used for predicting surface maps. Parallel to that, Figure 1 depicts the complete architecture of the split system. Note that this version fully builds on top of the baseline model one from [6] with an additional clone network. However, later in this work, we train both a combined (baseline) and a newly developed split model. Thereby, the combined version also embraces the model of Figure 1, but without the clone network and the connection components to the baseline. Furthermore, we omit certain elements like residual connections [12] and use less level of depth in both variants to save computational resources. Nevertheless, they were originally implemented in our basework (Section 4.2.2, [6]).

A. Combined architecture

The U-Net architecture consists of an encoder-decoder structure. Its input in most cases represents a multi-channel image (often RGB) with a certain width and height. First, the encoder starts to contract the image to its most important features. It thereby learns them through its double convolution layers with a kernel size of 3×3 (dark blue arrows). In the first layer, we increase the channels to 64, determining

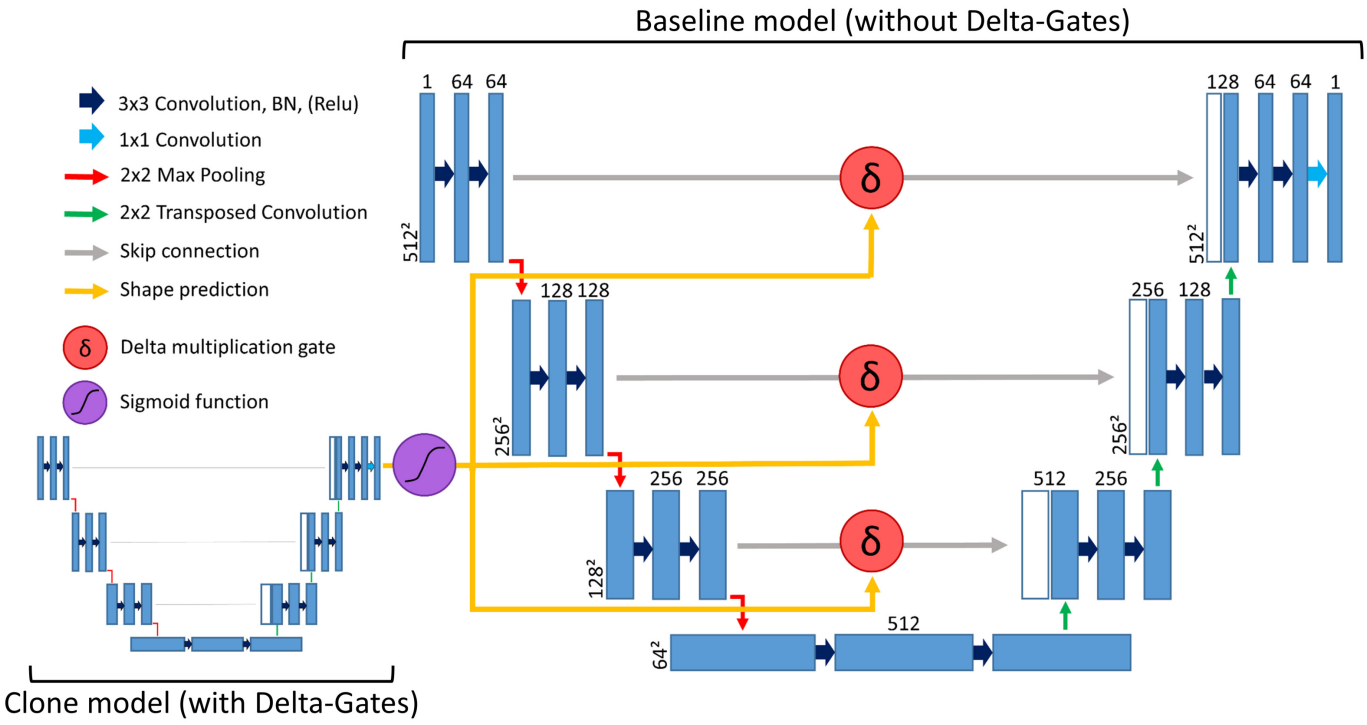

Figure 1. The figure depicts a schematic diagram of the split U-Net in detail. Note, that we denote the height and width of the image-tensor at the lower left part of one and the amount of feature channels at the top. In the described encoder-decoder system, convolution together with batch-normalization layers, relu activations and max poolings components are used to contract the input image (left side of the baseline and clone structure). Thereby, in each level, the width and height is halved, while the feature channels double. In the following expansion of the decoder path, the image is up-sampled. In each level of the architecture, transposed convolution is applied doubling the size again (right side of the baseline and clone structure). During this process, the skip connection containing spatial information is concatenated to the upcoming upsampled image from the decoder in each level of depth. This is followed by another double convolution-BN-ReLU sequence. Lastly, the image is back to the original size, but has too much feature channels for a form of displaying a value map of just the heights. Hence, a final convolution with the result of one image channel is applied performing pixel-wise regression. This concludes the combined U-Net structure. Besides the encoder-decoder system, it also shows the shape predicting clone used in the split U-Net version. The clone is doing a pixel-wise segmentation map of the artifacts from whether an artifact exists on a pixels location (1) or not (0). It trains in parallel to the main model and supports it with this map. After a Sigmoid function, this spatial information is added to it via the delta gates in each skip connection.

how many features can be learned simultaneously in this convolution block. Following every single convolution component, a batch-normalization (BN) component and a ReLU activation is applied. Note that for each convolution, we use a padding equal to 1, to preserve the size of the image-tensor. Although the original U-Net paper [7] does not do this, it is a popular tweak in many implementations. After each level of feature extraction (double convolution), 2×2 max pooling is applied to reduce the pixel to its most important pixels (red arrows). With this, spatial information is reduced, however, the information density in deeper layers grows. Moreover, with higher information density, more important features can be learned. As so, we double the feature channel size each level. Furthermore, because we later on will need the spatial information to up-sample our contracted image again, we save skip connections after each double convolutional setup. The bottom layer now represents a small image with very high density information, also having the maximal feature channel amount of 512 in the system. To achieve an output shaped exactly as the input, we step by step ascend the layers, re-adding our saved spatial information to it. To do so, as a first step, a transposed convolution with a kernel size of 2×2 is applied to it. Doing so, we double the width and height of the image (green arrows), matching the size one layer above. We concatenate it with the incoming skip connection, and apply double convolution. With the double convolution, we also half the feature channels again. Continuing this routine, we reach the top layer again with 64 channels and the original input width and height. Again through the padding in each convolution, we only use widths and heights of a potency of two, scaling the bottom layer exactly back to the input size. This will allow for a $1:1$ comparison of the in- and output. Lastly, a final convolution is applied to set the image channels to 1, finally performing pixel-wise regression for the image and predicting the actual height values.

B. Split U-Net architecture

The split U-Net architecture fully adapts the combined version. However, it uses a clone of itself, to predict only the shape of the objects on the surface via a segmentation with 1 (surface object) and 0 (no surface object). The produced segmentation is then used, to support the learned spatial information in the baseline structure. Figure 2 depicts how

exactly the shape U-Nets output is combined with the height U-Nets skip connection.

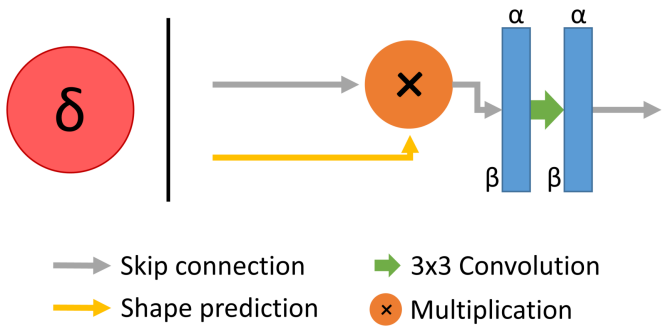

Figure 2. The figure visualizes the implementation of a Delta-Gate. A the skip connection (grey) of the baseline model is multiplied with the shape probability values (orange). The outcome is met with a 3×3 convolution to further process the just combined information. The convolution inherits the amount of image channels α and the size of the tensor β from the skip connection.

Firstly note, that a Sigmoid layer is applied at the end of our shape model, as shown in Figure 1. The output so represents values between 0 and 1 weighted as probabilities of how sure the network is, whether a pixel is or is not part of a surface object. By multiplying this probability and the spatial skip connection, we give each pixel of the skip connection this importance. Onwards, the result of the multiplication is conducted by a 3×3 kernel-sized convolution. This processes the weighted data which is then carried on, as the modified new skip connection for the upsampling part of the expansion path. Note that the output of the shape model inserted into every level of our height model is always sized 512×512. However, the size of the skip connection varies throughout the depth levels. We solve this by resizing the incoming shape tensor to the desired width and height.

III. Experimental Setup

A. Datasets

Two datasets will mainly be used in this work. Sentinel-2 imagery serves as the input data, while a normalized digital surface model (nDSM) is for ground truth validation. Both data sets are provided by DLR. The following subsections give a brief introduction into their source and their pre-processing applied from our side to fit neural network requirements.

- **Surface model data:** We are provided with nDSM data from measurements from the 15th to the 23rd of July 2017. Its shows the German city of Munich with its surrounding suburban, sometimes more rural area. The rasterized polygon depicts $310.7km^2$ of geometrical height information. Thereby, it has a pixel size of $0.5m \times 0.5m$ which was acquired by aerial lidar measurements.

- **Sentinel-2 data:** The Sentinel-2 data is provided already with a Sen2Cor [13] correction, providing level 2A imagery. The data so is freed from atmospheric, terrain and cirrus artifacts. The initial tile contains the blue (2nd), green (3rd), red (4th) and near infrared (8) band of Sentinel-2A with a spatial resolution of $10 \times 10m$ across those bands. The measurement of the sentinel imagery took place on the 16th February 2016. Note that the time slots of the acquisitions of both the nDSM and sentinel tiles differ by several months which could lead to incompatibilities in vegetation areas in both the shape and height case. However, we are building upon a certain work, the usage of the originally provided data preserves comparability.

B. Pre-Procsessing

In order to be able to input satellite data into a neural network, a tile based strategy is arguably the state of the art at the moment. This means that we grid our datasets into smaller crops to meet the deep learning model requirements in terms of input size. Furthermore, cropping is done in a way that each tile of the input has a corresponding tile with the same location and extent in the supervision dataset. This provides easy $1:1$ comparability between the sets. Most sources, also previous mentioned [4], [5] as well as our base work [6] utilize this approach. We here cut all tiles to a size of $512 \times 512px$ and stack them together in small data-packs easier processable for the computer and handier for later analysis.

Note that the extent of a specific pixel size would here differ from a crop of sentinel data and a crop of nDSM data, as they do not match in spatial resolution. To fix this, we up-sample the sentinel data two the nDSM matching resolution of $0.5 \times 0.5m$. As a side effect, the output of our model is also of half a meter, because the shape of the output is equal to the shape of the input in width and height of an image.

For this paper, thus, we work with different networks in parallel, the data-packs differ for each network. The combined model will receive a packet containing all four Sentinel-2 bands and the ground truth nDSM. For the split approach including the shape predicting clone model, also a shape ground truth is required providing segmentational ground truth data. We therefore apply a range from 0 to 1 to our data. One thereby represents an arbitrary surface object equal or greater a threshold τ. The rest of the pixels, smaller than τ, get zero. The selection of τ is crucial as it defines the surface objects the tile contains. Picking a too high threshold results in a false shape ground truth, because artifacts below higher τ would be represented as zero. They are just "swallowed". Selecting a very small threshold would result in an unclear shape tile, as each and every small artifact would be captured. To solve this threshold selection we train ten different versions of the split U-Net. We utilize different values of τ from 1 to 10. Equation 1 depicts a clearer mathematical visualization of how the segmentation ground truth is produced. Thereby, i denotes the x, y pixel position in the image. If the height at i is lower then the current used threshold, the value for the binary ground truth will be zero (not counted as a surface object), one elsewise (counted as a surface object).

$$height(i) = \begin{cases} 0, & \text{if } height(i) < \tau \\ 1, & \text{otherwise} \end{cases} \quad (1)$$

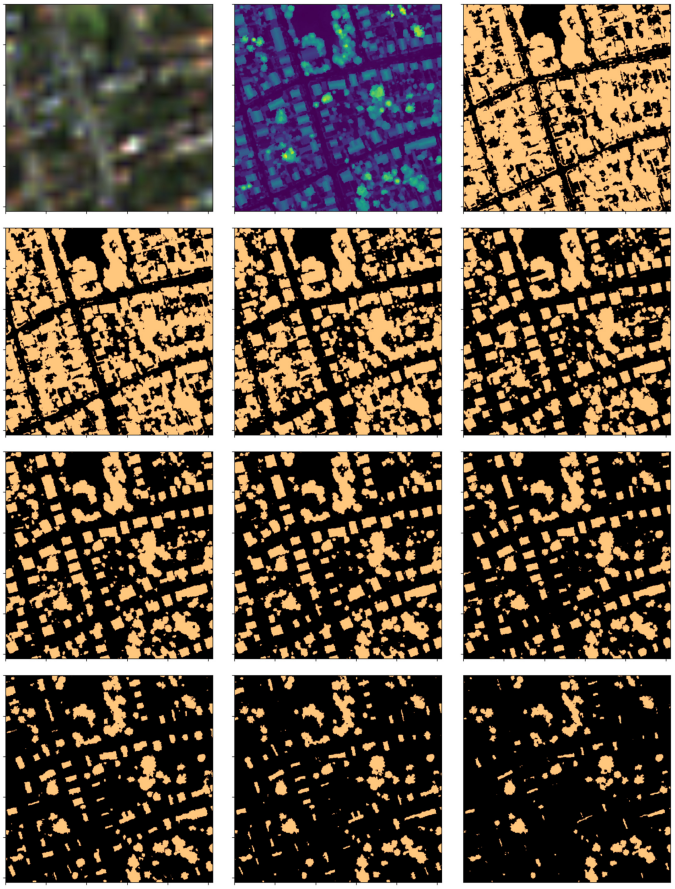

Figure 3. We here depict (in row order) a RGB composition of the input data, the height ground truth and the shape grounds truths. For latter ones, the threshold τ increases for pixels that set to zero (black). Note, how the shape of artifacts (ocher) thereby decreases respectively.

We illustrate different shape segmentations with the original nDSM ground truth in Figure 3. Note that just taking available cadastral footprints is not possible to replace the just described routine of producing a surface object segmentation map. Surely the use of such auxiliary data would be of great use here to build up strongly validated ground truth tiles. However, this work investigates not on the usage of different data sources to solve height map prediction, but the split neural network approach as its main focus. Furthermore, again, we keep comparability to our basework and also predict the whole environment, not just the buildings.

C. Implementation details

Both models are trained on the PyTorch [14] framework and Python 3.8 [15]. For this, we utilize partitions of a NVIDIA A100 and, for testing purpose,s a NVIDIA GTX 1060 GPU. Training took around 5 minutes for the combined and 15 minutes for the split network per epoch. On average, both models, in various tested parameters, took around 30 to 40 epochs to finish. They stop by a validation-loss based early stopping of five. Moreover, we split our tiles into 60% training, 30% validation and 10% data. We here chose a high amount of validation data, as we want to validate improvements as much as possible. As a loss function, we use the popular L1Loss [16] for height predictions in the baseline model. For back-propagating shape predictions, the prominent binary cross entropy loss [17] is applied. Lastly, we use Adam [18] as our optimizer with pytorchs grad scaler AMP [19]. As already mentioned, the height prediction architecture of both systems is exactly the same. We want to investigate only on the possible improvement brought from the clone model and strongly validate it. To further support the validation of the models, various metrics are collected: Firstly, the height predictions, we utilize commonly used regression indices like the mean absolute error (MAE) and the mean squared error (MSE). Both metrics are depicted in detail in Equation 2.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|x_i - y_i| \qquad MSE = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2 \tag{2}$$

We denote x and y as the prediction and the ground truth at position i in all pixels N. For the MSE, we square the calculated error to put more weight on higher height differences of the prediction and the ground truth.

For further evaluation we introduce the structural similarity index measure (SSIM). Here, also depicted in Equation 3, a window at a positional index i is slid over the image in a convolutional fashion. For each window, we calculate the local mean and standard deviation $\mu_{i_{x/y}}$ and $\sigma_{i_{x/y}}$ of again the prediction x and the ground truth y. Additionally, to numerically stabilize calculations, small constants are implemented , i.e. $C1$ and $C2$.

$$SSIM = \frac{(2\mu_{i_x}\mu_{i_y} + C1)(2\sigma_{i_{xy}} + C2)}{(\mu_{i_x}^2 + \mu_{i_y}^2 + C1)(\sigma_{i_x}^2 + \sigma_{i_y}^2 + C2)} \tag{3}$$

Hence the height prediction is a regression and the shape prediction a segmentation, we cant apply the same metrics. Therefore, we here stick with the commonly used Accuracy to validate our model. We depict the accuracy metric in detail in Equation 4. Thereby, we switch our denotation for the following metric equations to the system. Now, we mark the difference between the pixels of the prediction and ground truth as true-, false-positive or true-, false-negative represented by TP, FP and TN, FN (i.e. [20]).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

Furthermore, we conduct the popular F1-Score from our measurement of the recall and precision metric; typical ones for segmentation challenges. We define the F1-Score as described in Equation 5.

$$F1 = 2 \star \frac{Recall \times Precision}{Recall + Precision} = \frac{2TP}{2TP + FP + FN} \tag{5}$$

All metrics and indices we utilize are implemented, by the TorchMetrics [21] library. Additionally, we provide the full code of this work at Github [22].

IV. Results

This section is dedicated to present the numerical results. All metrics (MAE, MSE and SSIM) are combined in Table IV. Within it, we present the results of the combined version as well as all variants of the split models with $i < \tau = 0m$; τ ranging from 0 to 10. Hence, we get a direct comparison of the model variances. Note that we do not include metrics for the shape predicting clone model as they are only used for validating the functionality of it. We will discuss them separately later in this section.

Just by looking at the numerical domain we already can observe a few things. Firstly, it is uncommon, that the MSE is much higher than the square of the MAE. This indicates insufficiencies in our dataset as for example outliers and very poorly predicted tiles immensely increase the MSE, as MSE assigns more weight to bigger errors. However, we observe the same behaviour in our base work [6]. Additionally, when comparing it to the basework on a numerical basis, we here do produce values worse the numbers of it, lacking roughly $0.2m$ in the MAE and mostly over $10m$ in the MSE domain. Moreover, the validation results in our work here are constantly better than the training ones, which is unusual for deep neural networks in the image segmentation domain.

Furthermore, we obtain the best scores across all metrics not from the baseline model without the clone, but from the newly proposed variants with $\tau < 4m$ and $\tau < 5m$. This clearly indicates an improvement through our newly developed shape-predicting clone and reveals the effectiveness of taking on the shape prediction additionally in a separate network, as both the combined and the split baseline networks are identical.

For a visual impression of what our newly developed dual network approach is capable of, we depict an example of suburban scene processed by it. We produced a prediction and an error map from our result set in Figure 4. For the prediction itself, we use the variant with a threshold of $\tau < 4m$, thus, yields the most best scores. Note, how despite the very low input resolution, our network is still capable of capturing the main details of the image and the arrangement of the depicted surface objects.

Regardless mostly excellent results, we also encountered challenges. Besides the fact that captured surface objects are in blotchy shape, tiles containing large, human-made surface objects, are only detected partially. Hereby, both the combined and the split approach were not able to detect them consistently and without any gaps, yielding very high MSE values. We show an example of such a tile with a large mall near Munich-Pasing in Figure 5.

Lastly, as already mentioned at the beginning of this section, we shortly consider the metrics of the segmentation network. The values of those are not depicted in Table IV, as we cannot state a comparison with our combined model version with them. Moreover, they do not provide useful information for a discussion, the combined approach just does not use them.

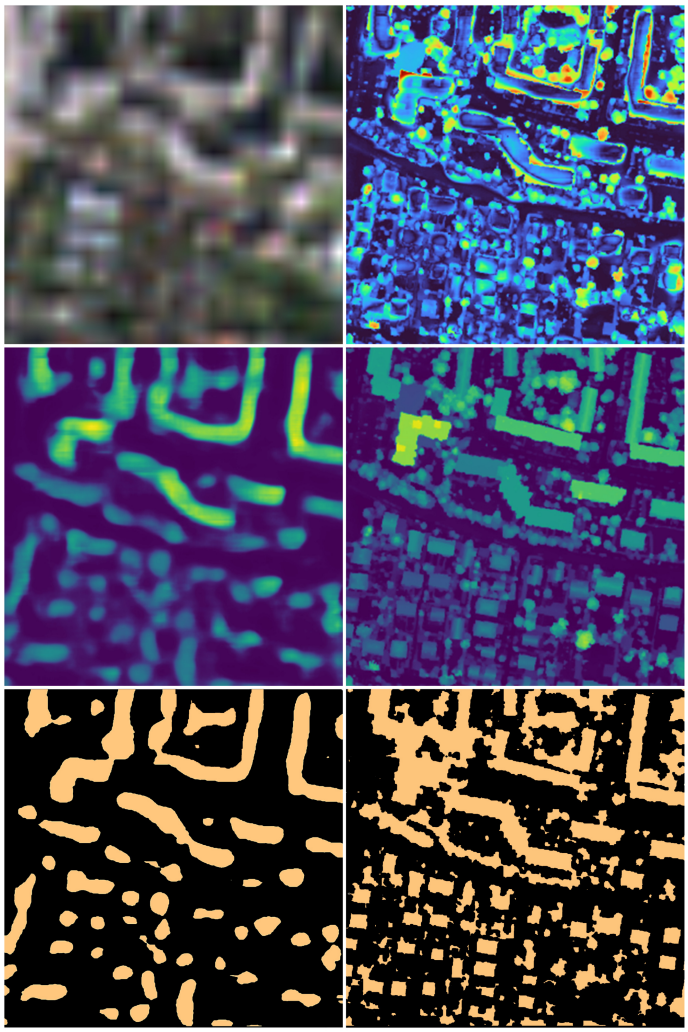

Figure 4. We here row-wise (left to right) depict the $10 \times 10m$ resoluted Sentinel-2 input file and an error map produced via taking the absolute error between our height-prediction and its ground truth respectively. The error map denotes high differences in red, lower ones in purple and light blue. We follow up with the height prediction and its ground truth before also showing the shape prediction of the clone network and its ground truth. Not that our predictions match the ground truth resolution of $0.5 \times 0.5m$.

However, for completeness, we depict our metrics for the clone model an Figure 6. Still, we shortly discuss them in Section V.

V. Discussion

This section is dedicated to discuss our findings made in Section IV.

At first, we mentioned that the values produced in this work, do not topple the ones of our basework [6]. This is due to the fact, that with less time and lower computational resources available, we omitted architectural improvements originally implemented. For example, we did not utilize residual connections or more image channels as mentioned in Section II. Moreover, we obtain lower MAE values, even though our loss function $L1Loss$ focuses on decreasing the MAE, while [6] uses a loss focusing down MSE values. This underlines the potential of the architectural changes made

Table I
THE TABLE DEPICTS THE MOST IMPORTANT MEASURED INDICES MEANED OVER THE WHOLE TRAINING, VALIDATION AND TEST SET. THE MODEL RESPECTIVELY IS THE BEST ONE MEASURED BY THE VALIDATION LOSS. THE LOSS (L1LOSS) IS IDENTICAL TO THE MEAN ABSOLUTE ERROR (MAE). WE MARK THE BEST INDEX IN BOLD. AND DENOTE THE DIRECTION OF THE METRICS SCORING BETTER WITH AN ARROW (↑, ↓)

Metric/Index	**Combined**	$\tau < 1$	$\tau < 2$	$\tau < 3$	$\tau < 4$	$\tau < 5$	$\tau < 6$	$\tau < 7$	$\tau < 8$	$\tau < 9$	$\tau < 10$
					Training (2630 tiles)						
MAE / L1Loss ↓	3.407	3.413	3.459	3.444	3.114	**3.102**	3.421	3.389	3.200	3.370	3.401
MSE ↓	32.630	33.165	34.270	33.778	28.056	**27.852**	33.427	32.623	29.403	32.474	33.135
SSIM ↑	0.492	0.497	0.494	0.494	0.507	**0.508**	0.495	0.495	0.500	0.491	0.493
					Validation (1316 tiles)						
MAE / L1Loss ↓	3.122	3.083	3.121	3.153	**3.010**	3.039	3.115	3.071	3.138	3.105	3.103
MSE ↓	28.519	27.341	28.546	29.536	**26.599**	27.598	28.382	27.238	28.565	27.885	28.178
SSIM ↑	0.516	0.516	0.518	0.516	0.523	**0.524**	0.519	0.519	0.517	0.517	0.511
					Test (438 tiles)						
MAE / L1Loss ↓	3.075	3.044	3.066	3.112	**2.952**	3.010	3.057	3.020	3.092	3.061	3.054
MSE ↓	27.960	27.127	28.040	29.190	**26.032**	27.416	27.878	26.867	28.160	27.640	27.791
SSIM ↑	0.458	0.460	0.464	0.460	**0.470**	0.468	0.463	0.462	0.464	0.460	0.460

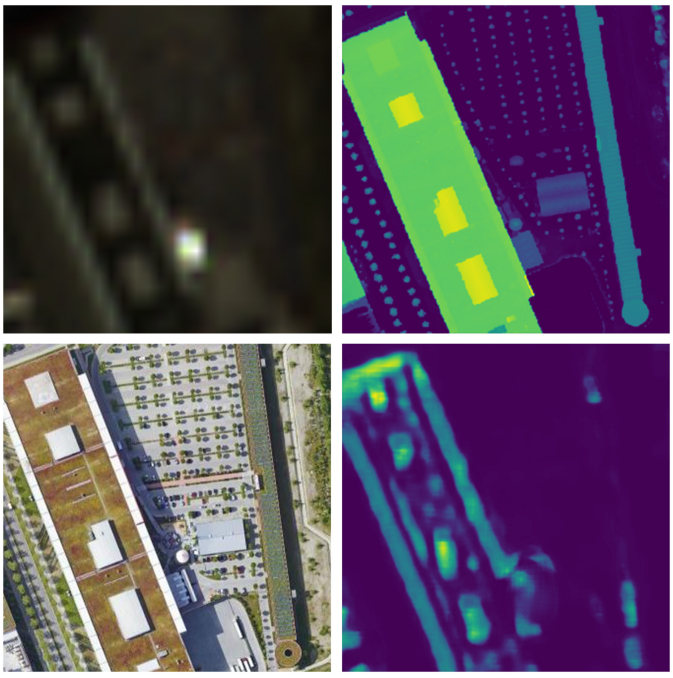

Figure 5. The Figure shows the Sentinel input data (upper left), the ground truth (upper right), the prediction (lower right) and the corresponding Google Maps reference (lower left) of a large suburban mall in Munich-Pasing (GPS coordinates: 48° 8' 1.2984" N 11° 24' 45.414" E). Here, the prediction was made by our newly developed split approach with $\tau < 4m$. As depicted, the flat roof of the hall could not be fully captured.

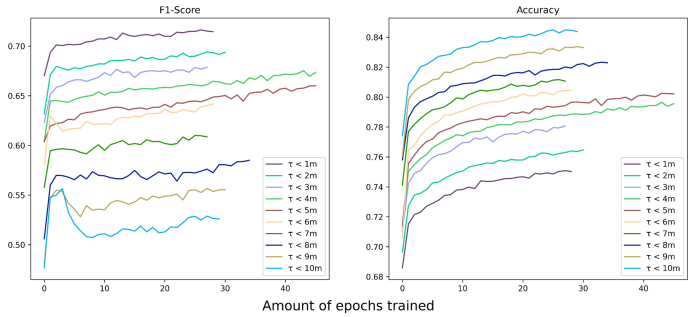

Figure 6. We here depict the F1 (left) and Accuracy (right) metric for validating our shape-predicting clone of the network over all epochs.

in [6]. Furthermore, our values found in the validation set, are better than those of the training one. The reason for it is the different distribution of the data files. As this work is dedicated to only investigate on the procedure of predicting nDSMs, we examine on our newly developed, split architecture. To prove our internal structure of it, it is important to have a high amount of validation data, for claiming actual improvements.

With the results in Table IV we clearly state that our dual network approach shows improvements over all metrics in comparison to the baseline version. Mostly, the best scores are achieved by the variants containing a threshold of $\tau < 4mor5m$. This confirms the previous thought that a threshold too small holds too much detail, while a threshold too high provides not enough. Thus, the perfect combination lies in the center. For a visual comparison, we depict Figure 7 showing the prediction of the input data, the ground truth, and both predictions of the combined and the split network.

As shown, the dual network approach (right side) outperforms the traditional combined approach (left side). We thought, that the improvement of the shape would be more visible. However, as we looked through the result tile, the major improvements were made in case of predicting the height itself. Although, the shape prediction was outsourced into a separate network. We so think that outsourcing this does not directly complement the shape itself, but gives the height prediction more room to work inside the baseline model. As we stated in Section I, by splitting up the tasks of predicting height and shape, better results were achieved. Nonetheless, of course some limitations stay, as for example a course

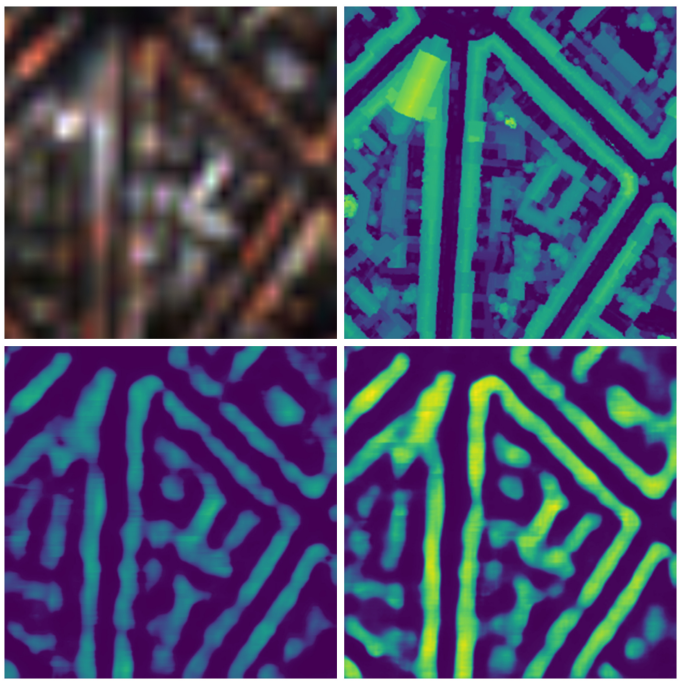

Figure 7. This figure depicts the Sentinel-2 input data tile ($10 \times 10m$) with its corresponding ground truth (upper row) and the both prediction of the combined network (lower left) and our proposed dual network approach (lower right). Thereby, the split network is more concise regarding the height and shape of the surface objects. This results in an numerical advantage in the MAE of roughly $1.1m$ for the split variant in this example.

poor input resolution still prevents us from capturing detailed surface objects and predict very defined shapes for smaller detected surface objects.

However our split approach is superior in both the numerical and visual realm, we encounter challenges, as already shown previously in Figure 5. This can have multiple reasons, however, the main factor is the material, the roof is made out of and its spectral properties. For smaller, residential buildings, roofs are often made out of roof bricks or roof tiles, yielding a specific spectral signature our network can adapt to predict them as a house and denote a typical height. However, for larger scale buildings, roofs are usually made out of concrete or flat, dark roof plates. If the spectral property of such is too similar to for example asphalt or parking lots made out of concrete, the model can no longer differentiate between a roof and flat space on the ground. Hence, the height will be close to zero. This effect is amplified when roofs are greened, as given in Figure 5. Our network can only relate objects with a clear distinguishable spectral signature to a certain height. However, greened space is predicted with a ground like height, as ground areas like meadows and lawns match the spectral properties.

For our clone network, we conducted the F1-Score and the Accurary in Figure 6 in Section IV. Although they are not part of the actual comparison between our two architectures, they are still an important part of the validation of the clone model. We so want to shortly discuss the findings on those two metrics: If one observes the results of the F1-Score and the Accuracy, we get a linear relation between τ and the score for the F1-, and an inverted linear relation for the Accuracy-Score. Firstly, for the F1 case, this makes sense due to the fact that with more details being shown with a decreasing threshold τ, we are able to predict more surface-objects than non-surface objects, complementing F1. This is due to the fact that the score does not account for TN pixels and puts a double weight on TP. Hence, with a lower τ, the F1 metric is increasing. Vice versa, with less details, F1 achieves lower scores, cause there are no surface objects to predict. Meanwhile, Accuracy implements all cases. This leads to the inverted linear relationship, as with an increasing threshold of τ, there is just less to predict. That means, more TN pixels are pushing up the accuracy.

Lastly, for the discussion, we want to further evaluate on the challenge of comparing our work to current state of the art of deep neural network DSM creations. When comparing our work to recent literature, e.g. [5], [4], our networks lacks big chunks in numerical and visual results. However. the resolution of our used input, is worse by around a factor of 14 than the ones used in the cited literature. Moreover, one can not apply this factor onto the metrics measured by us and compare them to those in the presented literature. This is due to the fact, that with more details to capture through better resolution, also more errors will be obtained, just because of the nature of seeing more in the input image that needs to be detected by the network. Nevertheless, our input data has the clear advantage of being open access, frequently updated and available on a global scale. This leads us to propose to in general use our model for index mapping (e.g. [23], [24], for only vegetation: [25]), in where we could assign height or density classes to our predicted tile, and then use it for a national wide country height mapping. Although, one could argue that we then do not need to predict to a resolution up to $0.5 \times 0.5m$. However, it is clear, that the finer resolution we achieve, the closer we get to the actual truth. Hence, we can also predict indices for index mapping more accurately. Nonetheless, if doing so, we would need more data from different locations to achieve better generalization over larger areas for our model.

VI. Conclusion and Outlook

In conclusion, both approaches, the combined one and our newly developed split approach manage to create decent nDSM tiles with main details clearly being differentiable. Moreover, our dual network approach achieved better results in both the numerical and visual domain. When selecting the threshold for additional ground truth prediction τ, we showed directly by Table IV that cutting out details can help the network focus on the important surface structure, as long as we preserve enough. We so propose to further investigate on such dual network structures, as they might increase the performance in other tasks besides only DSM creation. We lastly think, that our developed methodology can be used for index mapping across larger areas, thus, our approach predicting rectangular tiles would be easy transferable to such a task. As a wrap up, we show the potential of our dual network approach in comparison to the combined architecture

in Figure 8. We present the different height predictions for also the split approach using $\tau < 1m$ and $\tau < 10m$ to visually compare them to the best model with $\tau < 4m$.

Acknowledgment

The author personally thanks R. Leppich for sharing computational resources on the university GPU cluster.

List of Abbreviations

Abbreviations used:

DSM	Digital Surface Model
nDSM	Normalized Digital Surface Model
BN	Batch Normalization
MAE	Mean Absolute Error
MSE	Mean Squared Error
SSIM	Structural Similarity Index Measure
TP	True Positive (pixel)
TN	True Negative (pixel)
FP	False Positive (pixel)
FN	False Negative (pixel)

References

[1] C. Geiß, H. Taubenböck, M. Wurm, *et al.*, "Remote sensing-based characterization of settlement structures for assessing local potential of district heat," *Remote Sensing*, vol. 3, no. 7, pp. 1447–1471, 2011.

[2] B. Stone and M. Rodgers, "Urban form and thermal efficiency: How the design of cities influences the urban heat island effect," *Journal of the American Planning Association*, vol. 67, pp. 186–198, Jun. 2001. DOI: 10.1080/01944360108976228.

[3] C. Geiß, P. Priesmeier, P. A. Pelizari, *et al.*, "Benefits of global earth observation missions for exposure estimation and earthquake loss modelling–evidence from santiago de chile, chile," 2011.

[4] C.-J. Liu, V. A. Krylov, P. Kane, G. Kavanagh, and R. Dahyot, "Im2elevation: Building height estimation from single-view aerial imagery," *Remote Sensing*, vol. 12, p. 2719, Aug. 2020. DOI: 10.3390/rs12172719.

[5] L. M. andXiao Xiang Zhu, "Im2height: Height estimation from single monocular imagery via fully residual convolutional-deconvolutional network," *CoRR*, vol. abs/1802.10249, 2018. arXiv: 1802.10249. [Online]. Available: http://arxiv.org/abs/1802.10249.

[6] K. Müller, "Creating high resolution normalized digital surface models through supervised deep learning using sentinel-2 data," Bachelor, Julius-Maximilians-Universität Würzburg, 2021. [Online]. Available: https://elib.dlr.de/144693/.

[7] P. F. Olaf Ronneberger and T. Brox, "U-net: Convolutional networks for biomedicalimage segmentation," *Computer Science Department and BIOSS Centre for Biological Signalling Studies, University of Freiburg, Germany*, 2017. DOI: https://doi.org/10.1007/978-3-319-24574-4_28.

[8] E. Irwansyah, Y. Heryadi, and A. Agung, "Semantic image segmentation for building detection in urban area with aerial photograph image using u-net models," Mar. 2021. DOI: 10.1109/AGERS51788.2020.9452773.

[9] T. J. Yi, "Semantic segmentation of aerial imagery using u-nets," Mar. 2020. [Online]. Available: https://scholar.afit.edu/cgi/viewcontent.cgi?article=4594&context=etd, 07.09.2021.

[10] P. Ulmas and I. Liiv, "Segmentation of satellite imagery using u-net models for land cover classification," vol. abs/2003.02899, 2020.

[11] 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, ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2018.05.103. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231218307008.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Jun. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[13] E. S. A. (ESA). "Sen2cor." (), [Online]. Available: http://step.esa.int/main/snap-supported-plugins/sen2cor/. 26.06.2021.

[14] PyTorch. "From research to production." (), [Online]. Available: https://pytorch.org/. 27.07.2021.

[15] P. S. Foundation. "Python." (), [Online]. Available: https://www.python.org/. 04.08.2021.

[16] PyTorch. "L1loss." (), [Online]. Available: https://pytorch.org/docs/stable/generated/torch.nn.L1Loss.html. 06.08.2021.

[17] PyTorch. "Crossentropyloss." (), [Online]. Available: https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html. 05.08.2021.

[18] PyTorch. "Adam." (), [Online]. Available: https://pytorch.org/docs/stable/generated/torch.optim.Adam.html#torch.optim.Adam. 27.07.2021.

[19] PyTorch. "Automatic mixed precision package - torch.cuda.amp." (), [Online]. Available: https://pytorch.org/docs/stable/amp.html. 10.08.2021.

[20] G. Developers. "Classification: True vs. false and positive vs. negative." (), [Online]. Available: https://developers.google.com/machine-learning/crash-course/classification/true-false-positive-negative?hl=en. 10.02.2023.

[21] P. L. et al. "Torchmetrics." (2022), [Online]. Available: https://torchmetrics.readthedocs.io/en/stable/.

[22] K. Müller. "Scientificwriting repository." (), [Online]. Available: https://github.com/KonstiDE/ScientificWriting/tree/master.

[23] D. Frantz, F. Schug, A. Okujeni, *et al.*, "National-scale mapping of building height using sentinel-1 and sentinel-2 time series," *Remote Sensing of Environment*, vol. 252, p. 112 128, 2021, ISSN: 0034-4257. DOI: https://doi.org/10.1016/j.rse.2020.112128. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0034425720305010.

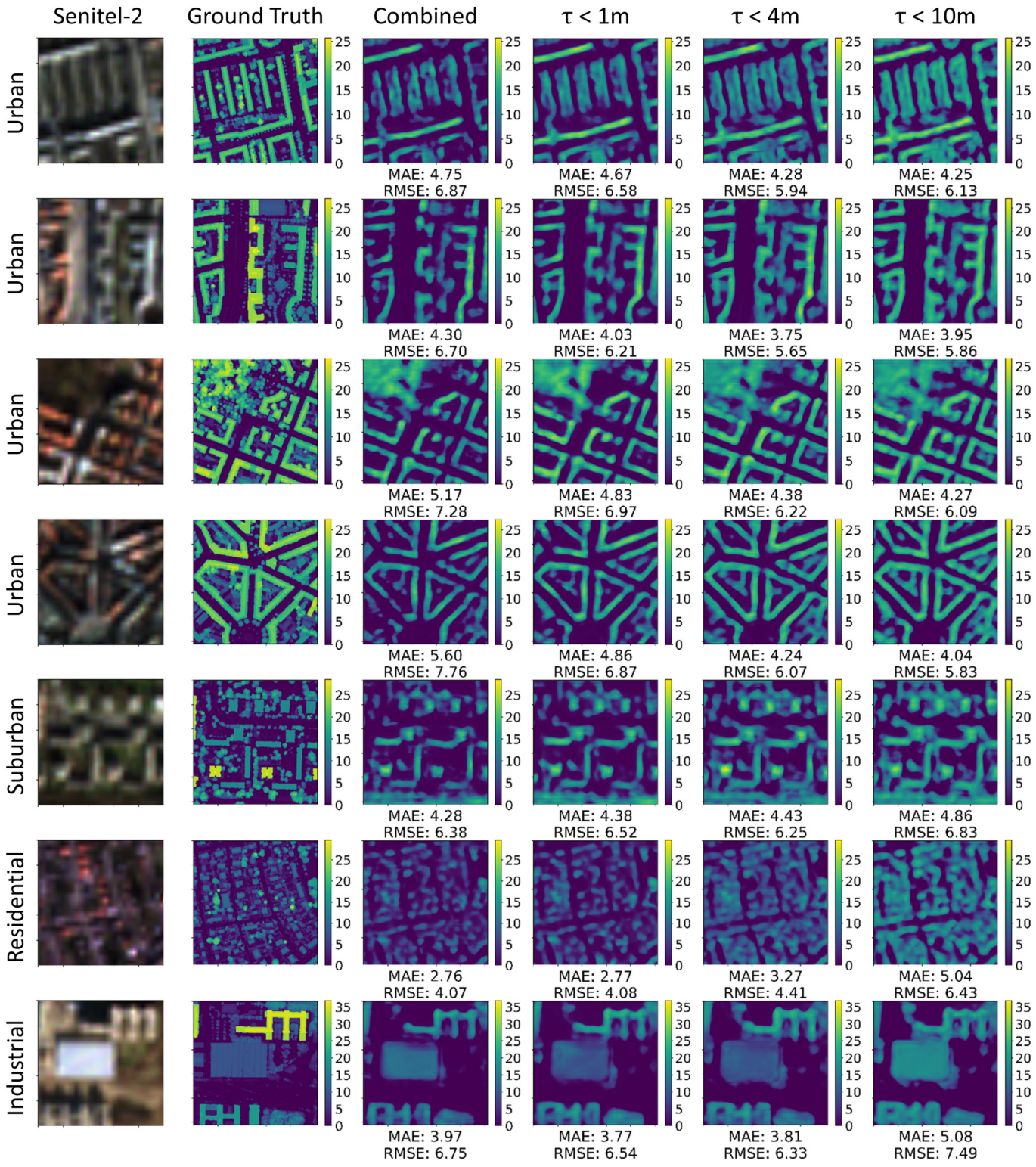

Figure 8. The Figure depicts (from left to right) the Sentinel-2 input data, the nDSM ground truth, and the prediction of 3 split approach variant utilizing a shape ground truth with either $\tau < 1m$, $4m$ *or* $10m$ for different tiles.

[24] C. Geiß, T. Leichtle, M. Wurm, *et al.*, "Large-area characterization of urban morphology—mapping of built-up height and density using tandem-x and sentinel-2 data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 8, pp. 2912–2927, 2019. DOI: 10.1109/JSTARS.2019.2917755.

[25] N. Lang and J. Wegner, "Country-wide high-resolution vegetation height mapping with sentinel-2," *Remote Sensing of Environment*, vol. 233, p. 111 347, Nov. 2019. DOI: 10.1016/j.rse.2019.111347.