TREES HEIGHTS PREDICTION BY CONVOLUTION NEURAL NETWORKS (CNN)

It is known, that the higher the frequencies of radio signals, the more their propagation is affected by obstacles' influence. It means any kind of object between the transmitter and receiver works as a barrier deteriorating the radio signal propagation. Therefore, such impact should be considered in case of 5G networks with a frequency range from 450 MHz to 6 GHz and from 24.25 GHz to 52.6 GHz. The vegetation presented by an array of crowns of trees and every single vegetation unit and bushes must be embedded into the mapping models.

As a rule, a contemporary city involves many parks, squares, recreation areas, tree plantations, etc. In some cities, the buildings are entirely shrouded in foliage.

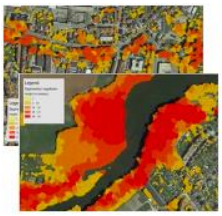
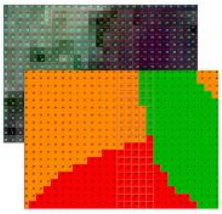
Thus, there is a task to correctly identify the vegetation and map it in the obstacle models with the required accuracy and details. 5G networks deployed without bearing in mind the 3D trees do not correspond to the 5G planning demands and lead to lower connectivity. On the other hand, 3D Tree models ensure an accurate and reliable 5G network planning process allowing the placing of a required number of base stations for stable signal and full coverage.

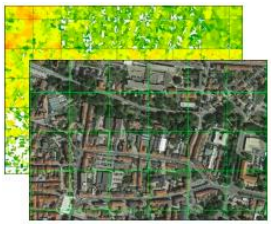
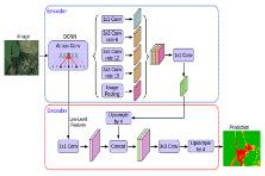
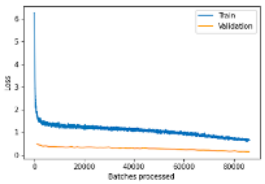
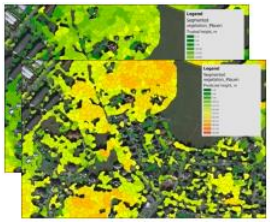
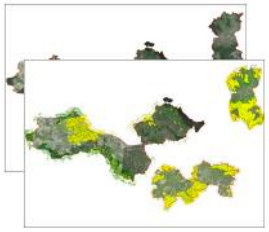
One of our projects was the creation of 3D models of 146 cities in Germany for 5G network deployment for the national operator. Accordingly, 3D Tree models should be produced for more than 1,500 sq.km with the high quality and accuracy required for 5G planning.

The best and most reliable sources for deriving accurate vegetation heights are LIDAR data and stereo aerial photos. But these data are usually enough expensive and not always available for the coverage of large areas.

For this project, our R&D department created the technology of a previously trained **Convolutional Neural Network (CNN)** model trained with available sources for two cities – aerial images and LIDAR data for Bautzen city and Plauen city - for further deployment of the obtained results for other cities.

THE WORKFLOW OF TREES HEIGHTS MEASUREMENT

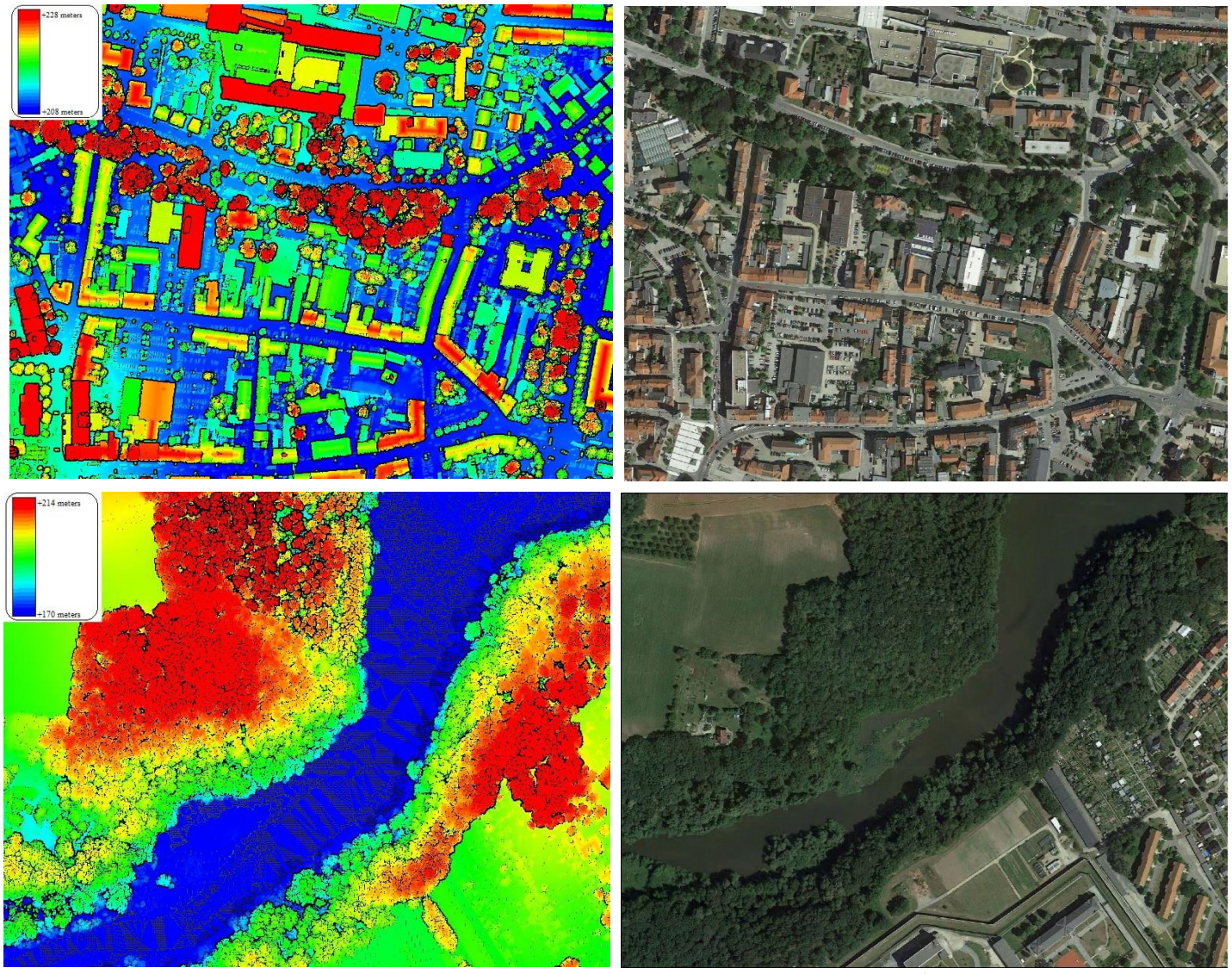
					
STEP 1 Input data	**STEP 2** Identification of different types of vegetation by Semantic Segmentation methods	**STEP 3** Post Processing results and creation of vector polygons for vegetation	**STEP 4** Segmentation of vegetation	**STEP 5** LIDAR point cloud data	**STEP 6** Problem Statement

				
STEP 7 Training set creation	**STEP 8** Choosing of the model architecture and its further training	**STEP 9** Model training	**STEP 10** Cross-validation and quality assessment according to the CNN model	**STEP 11** Model deployment for 146 cities

Step 1: Our inputs

We used LIDAR data and orthophoto of aerial imageries with corresponding metadata for Bautzen and Plauen cities for our model training and deployment for other 100+ cities in Germany

Preview of LIDAR POINT CLOUD (a) AND AERIAL IMAGERY (b) for the same part of Bautzen city



a b

Step 2: Identification of different types of vegetation by Semantic Segmentation methods

The process of vegetation feature extraction from imageries is based on the segmentation and classification tools of ArcGIS Pro and its API for Deep Learning. As a result of image processing, we got a thematic raster with two vegetation classes: “Trees” and “Bushes”. Then their shapes and spatial characteristics were grouped into the objects that represent all vegetation on the ground.

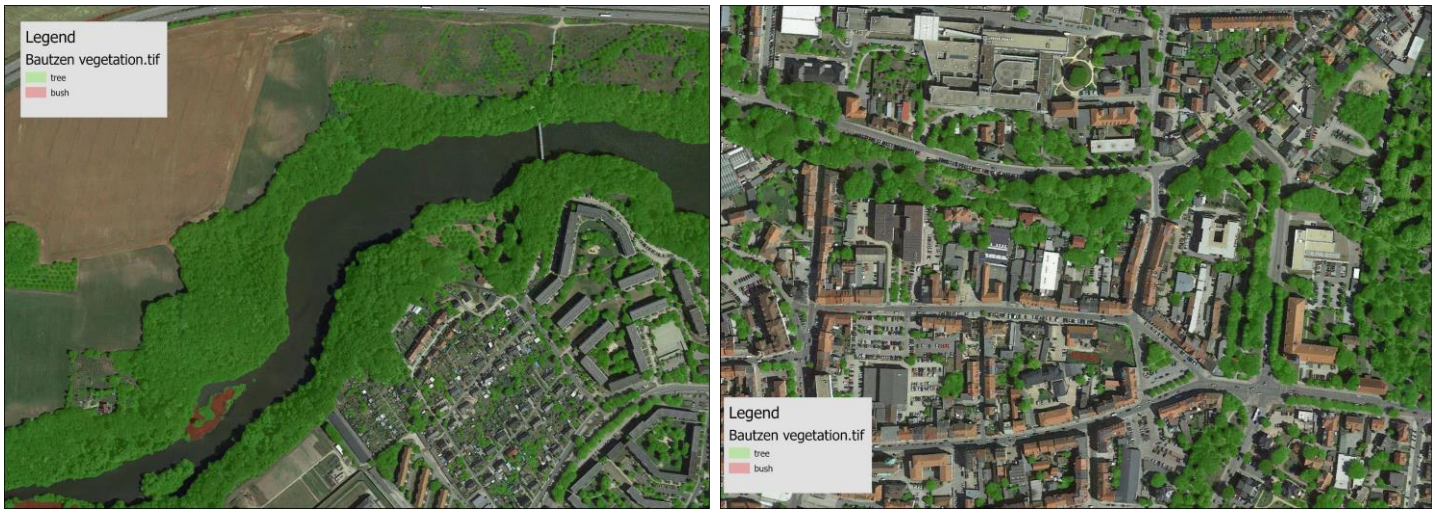


Figure 1. Thematic raster with two vegetation classes (Trees and Bushes), Bautzen city

Step 3: Thematic raster post-processing and creation of vegetation vector polygons

Thematic raster was converted into GIS vector and post-processed to obtain precise topology of vegetation outlines

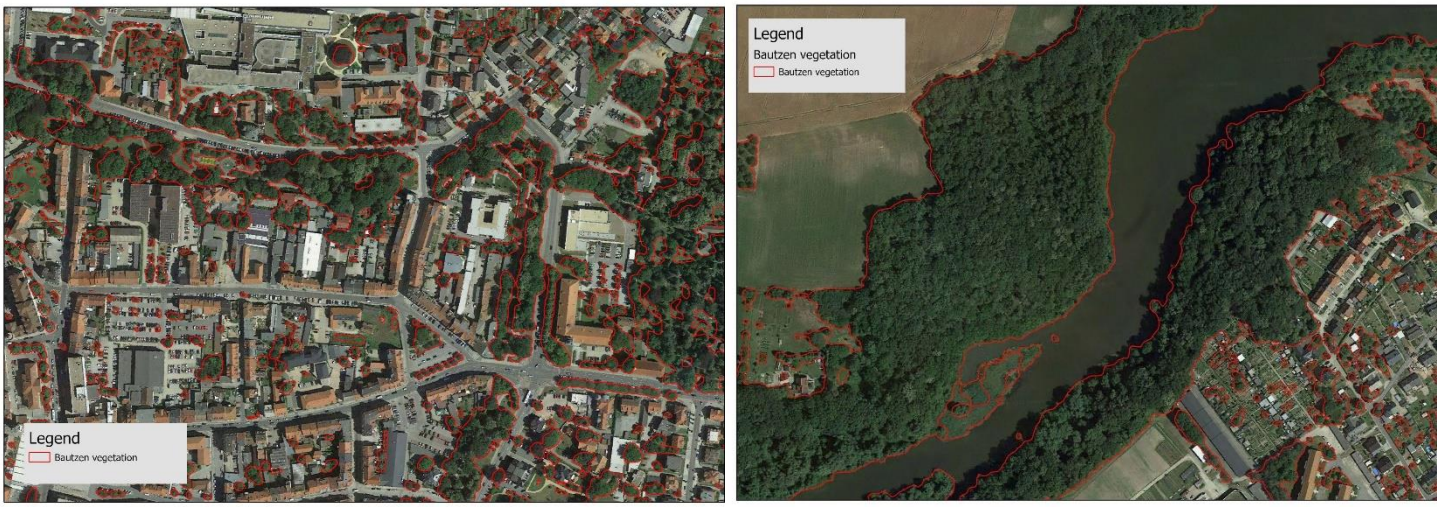


Figure 2. Thematic raster with processed vector, Bautzen city

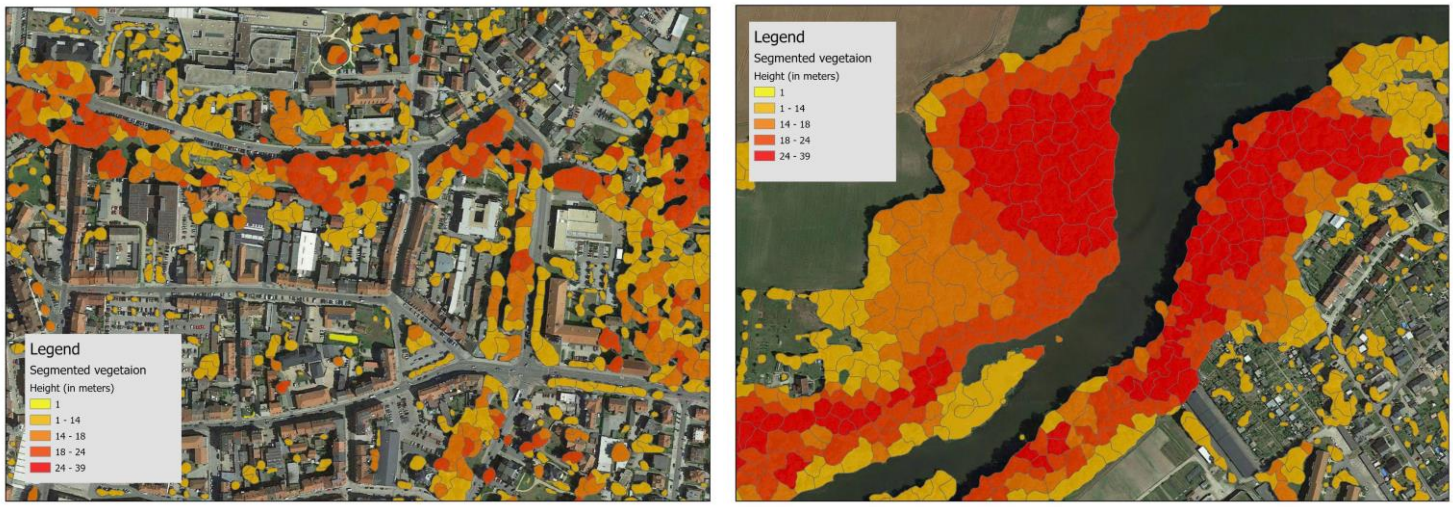
Step 4: Segmentation of vegetation

Then, vegetation vector polygons were segmented into small polygons by geometrical methods based on the Tissen polygons technique (see Fig.3). The applied procedures keep the topology of segmented vector polygons accurate and correct, providing the same or very close height of segmented polygons as in the initial vegetation array.

Figure 3. Thematic raster with segmented vector polygons, Bautzen city

Step 5: LIDAR Points Cloud data

LIDAR data contains heights of above-ground features including all obstacles like buildings, trees, etc. So, if LIDAR data are available for the specific area they can be used for vegetation height estimation and measurement.
LIDAR Points Cloud data were processed within our project for Bautzen city and then, the obtained heights were assigned to the corresponding segments of the vegetation polygons.



(4a) (4b)

Figure 4. The obtained heights of the vegetation polygons for urban (a) and forest areas (b)

Step 6: The Task Statement

The next step is the creation of the new thematic raster from the segmented vegetation vector with the height attribute obtained in the previous step.
The new thematic raster should represent the vegetation height value in each pixel where "Null" correspond to pixels of the aerial image without vegetation and *pixels with values more than 0 describe vegetation heights in every point.* The thematic raster must strictly match the aerial imagery in each pixel, both spatially (XY coordinates) and thematically (vegetation). For our project, we used aerial imagery of Baurzen city to derive our new thematic raster, so the output data do not have spatial shifts and season-based artifacts like deforestation or new trees.

In terms of Computer Vision (CV) we have RGB image and a Labeled raster with values that are considered as desired classes of vegetation. These data are the basis for further training and testing by various machine learning methods.

Figure 5 below demonstrates the used aerial imagery (RGB image) where the given values mean Brightness for Green, Blue, and Red bands correspondingly and a thematic (labeled) raster with values of vegetation heights (i.e. desired classes).

To summarize, the task is to classify each pixel on the image by the set of predefined classes. Based on our previous experience, we used semantic segmentation methods by Convolutional Neural Networks (CNN) as the most suitable solution.

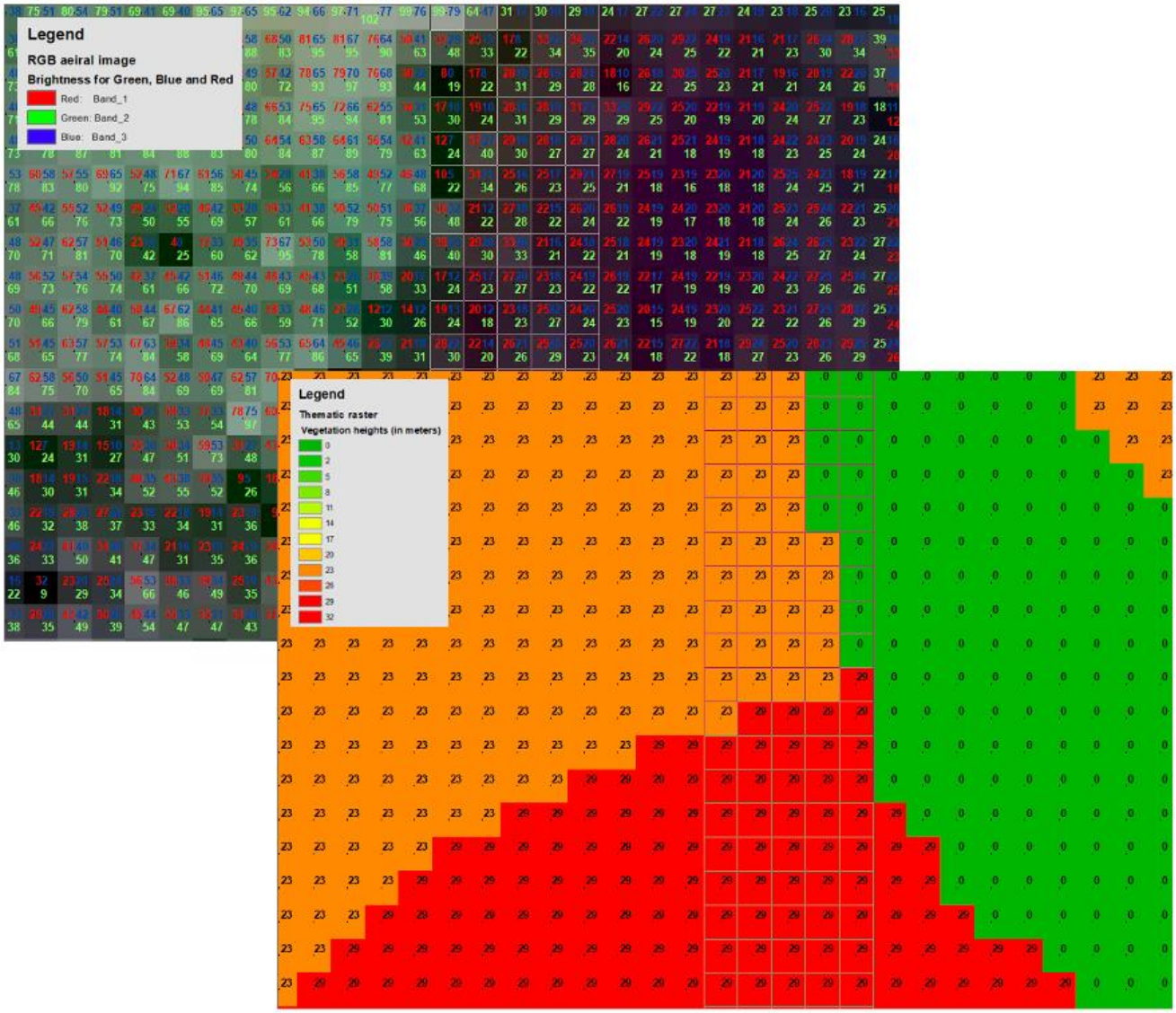


Figure 5. Vegetation heights according to the aerial imagery and output thematic raster

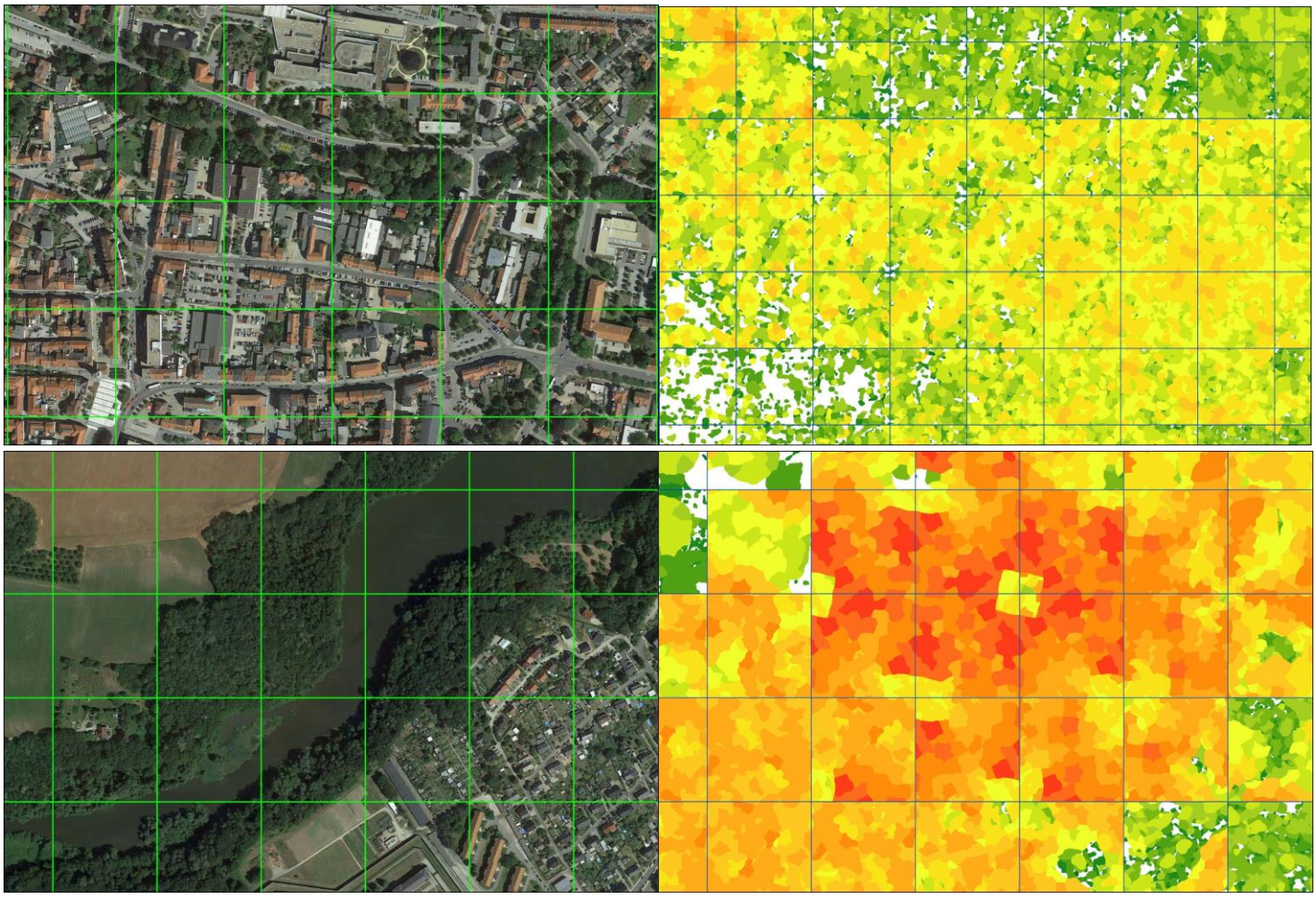
Step 7: Training set creation

The aerial imageries (orthophotos) are presented as 8-bit images with 3 RGB bands with a huge XY pixel size. To make their processing more effective we divided the aerial image (and labeled raster) into a tiles package in standard metadata format for Deep learning frameworks and CUDA. We defined the size of each tile as 512×512 pixels.

The resulting Training set we have got includes:

- Set of small images
- The corresponding labeled images
- Metadata describing these images for ArcGIS and Pytorch API.

The pictures below demonstrate the part of our training dataset: small images obtained from aerial imagery (left side) and labeled image obtained from the thematic elevation raster (right side).



(6a) (6b)

Figure 6. The resulting dataset on the orthophoto (7a) and the elevation thematic raster (7b)

Additionally, we implemented into the Training set a number of transformations (rotation, height, and width displacement). That allows the model to make more realistic generalizations. Since the model never sees the same image twice, the implemented rotation methods (90º, 180º, and 270º) around the vertical axis complement the training set with new realistic images.

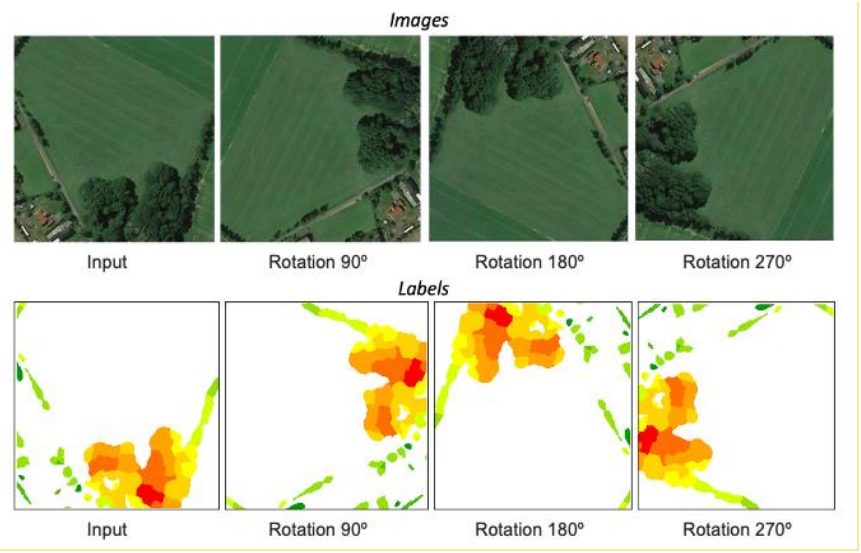


Figure 7. Examples of the images and labels rotation methods.

The resulting training set consists of 18030 images, labeled tiles of 512x512x3 size, and the desired 40 classes.
All tiles were divided into two subsets for training and testing (see Fig. 8). During training, these datasets are shuffled automatically and divided into two groups, which include the training set (90%) and the cross-validation set (10%). Over several cycles of training and cross-validation the hyperparameters are determined and optimized.

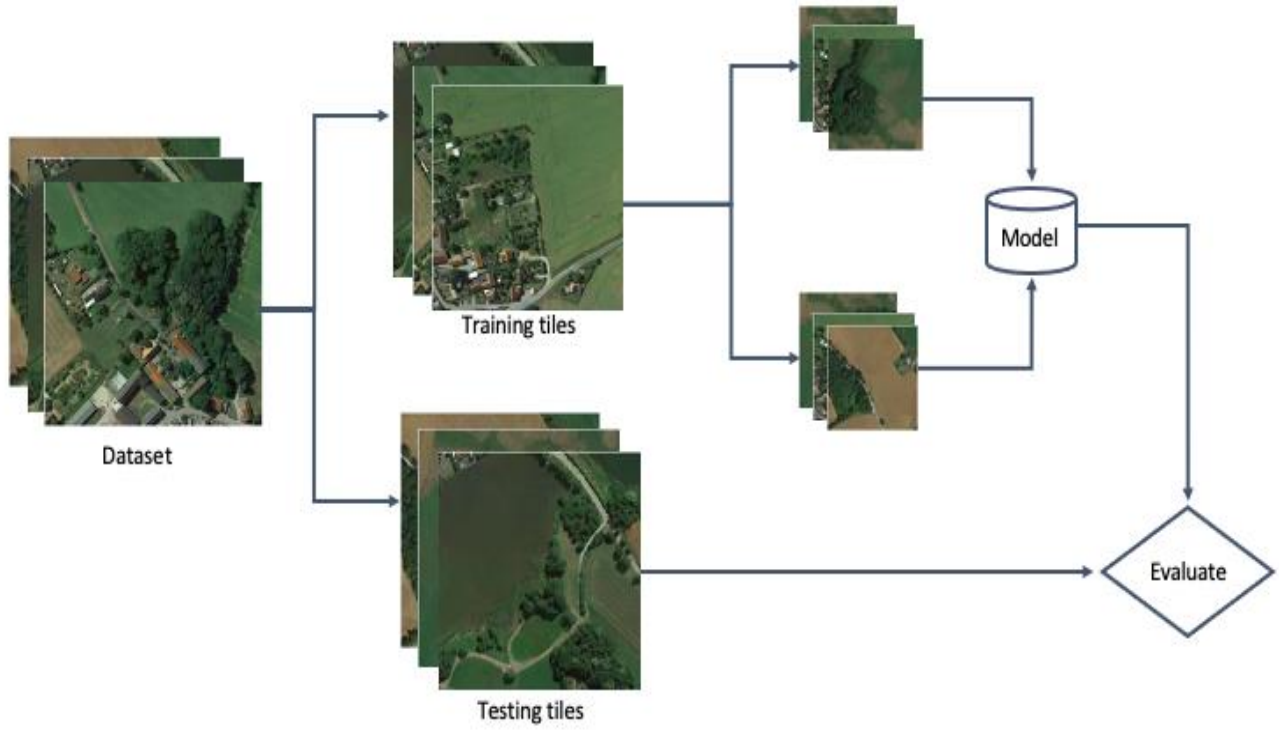


Figure 8. Training set

Step 8: Model architecture

We chose Deeplab v3+ CNN architecture (its structure is shown in Figure 9). This model is very similar to the classical U-net model, but the main difference is the usage of Atrous Convolution instead classical one.
The Atrous Convolution uses computer memory very effectively, works well with objects' edges, and can encode multi-scale contextual information.

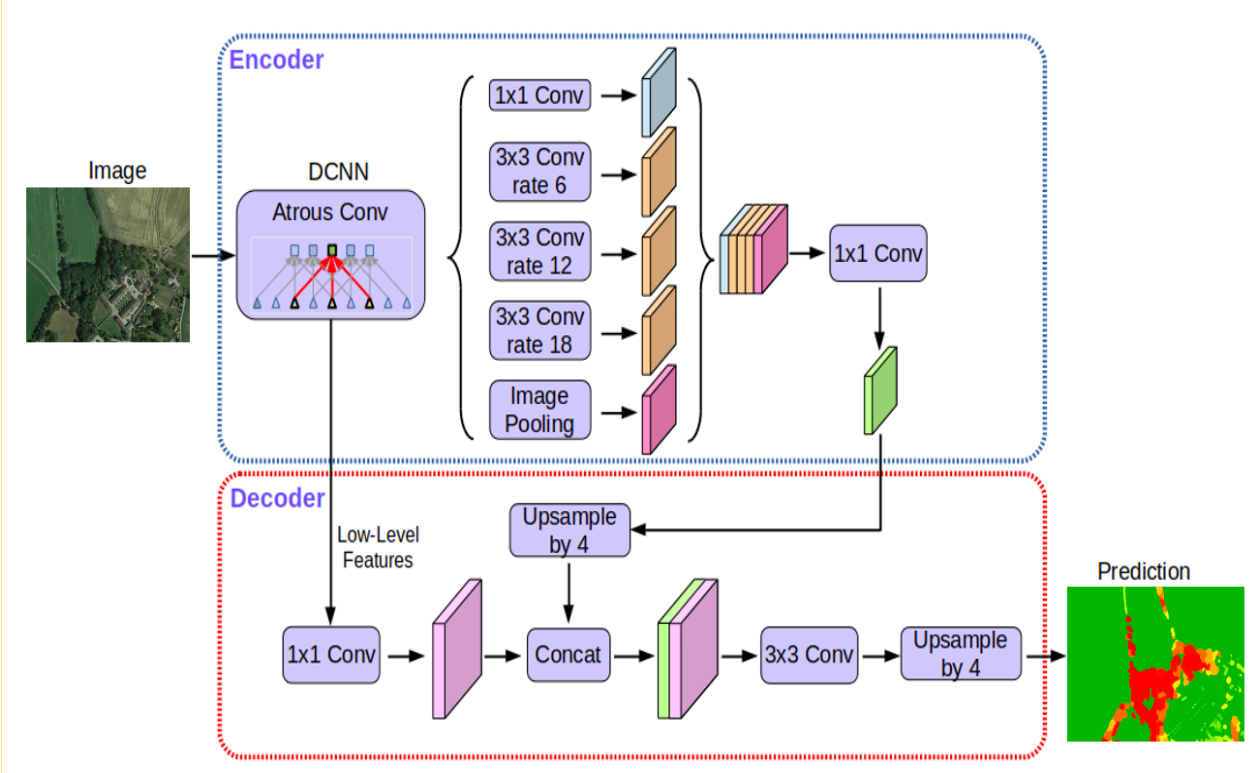


Figure 9. Structure of the model architecture
(taken from the open data source: https://ai.googleblog.com/2018/03/semantic-image-segmentation-with.html and elaborated by the authors)

Step 9: Model training

The ArcGIS API for Python and its arcgis.learn module can use PyTorch (an open-source machine learning framework based on the Python programming language and the Torch library) to create a prototype and deploy neural networks. Additionally, several useful for data scientists packages like Scikit-learn, Numpy, Pandas, etc., integrated into the ArcGIS ecosystem can also be used.

The CNN model is saved as *.pth + *.emd files that contain the model architecture and fitted weights.
The training process is the improvement of the weights by reducing the gap between the losses on the training and validation data sets after each batch completion.

The learning process is based on graphics processing unit (GPU) calculations with the usage of the parallel computing platform CUDA (Nvidia). The software divides the data processing task into several small parts and sends them to the GPU. Then the GPU performs parallel calculations of these small portions at higher speeds. This algorithm speeds up the model learning procedure from weeks and months to several days compared to the CPU calculations.

During the training process, we evaluate its quality with Loss Function charts on the Training and Validation sets. These charts provide us with enough information about Training set quality, it's balancing, and fitting.

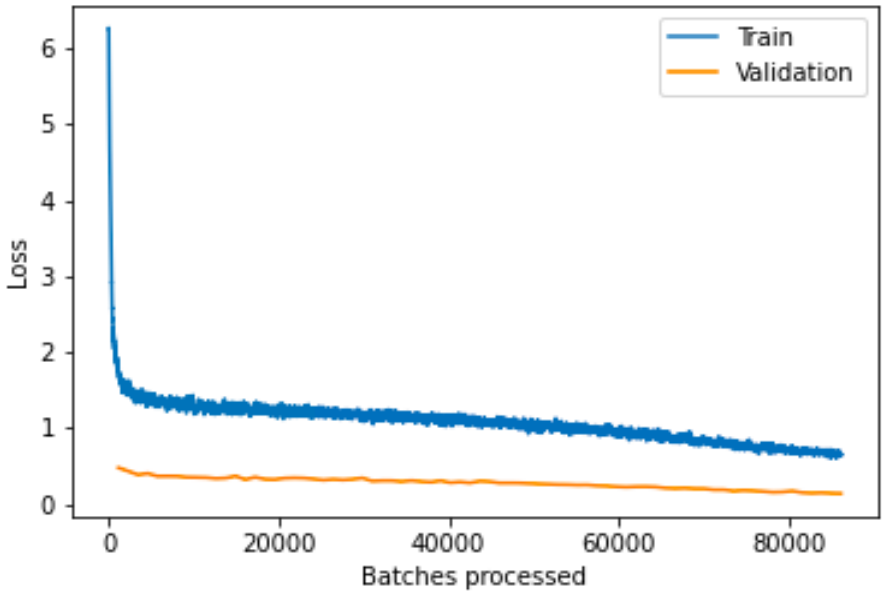


Figure 10. Loss Function chart

But charts don't provide us with information about the quality of pixels classification which is usually evaluated through per-class metrics (metrics and accuracy values for some classes are represented in the table below).

	NoData	**2**	**5**	**8**	**11**	**14**	**17**	**20**	**23**	**26**	**29**	**32**	**35**	**40**
Precision	0,98	0,90	0,78	0,81	0,79	0,81	0,79	0,80	0,80	0,77	0,75	0,74	0,70	0,74
Recall	0,99	0,87	0,72	0,76	0,77	0,74	0,78	0,80	0,81	0,80	0,71	0,69	0,71	0,59
f1	0,99	0,89	0,75	0,78	0,78	0,77	0,79	0,80	0,81	0,78	0,72	0,71	0,71	0,66

For example, class 11 (i.e. set of pixels of 11m height) was classified with a precision 79% on the test data.
However, taking into account the other landscape, the same accuracy couldn't be achieved in other cities we should process. The task of the model testing on other cities to receive the objective value of the classification accuracy is performed in the next step.

Step 10: Cross Validation and Quality Estimation by Fitted CNN Model

To obtain an objective assessment, it is necessary to test the model on other data and under different conditions using the Cross-Validation technique.

Once the model is fitted, it is ready to be used on the sizeable aerial image. First, the image is divided into packages of tiles, and then these tiles are passed to the model predict function, which returns a set of labeled tiles of images and provides a predicted class for each pixel. And finally, the predicted labeled tiles are stitched to create a complete predicted labeled image for the entire input image.

Since we had LIDAR data for Plauen city available, we used them to evaluate the effectiveness of the presented CNN model. The vegetation heights derived from LIDAR data are considered as **trusted vegetation heights** to compare with predicted ones.

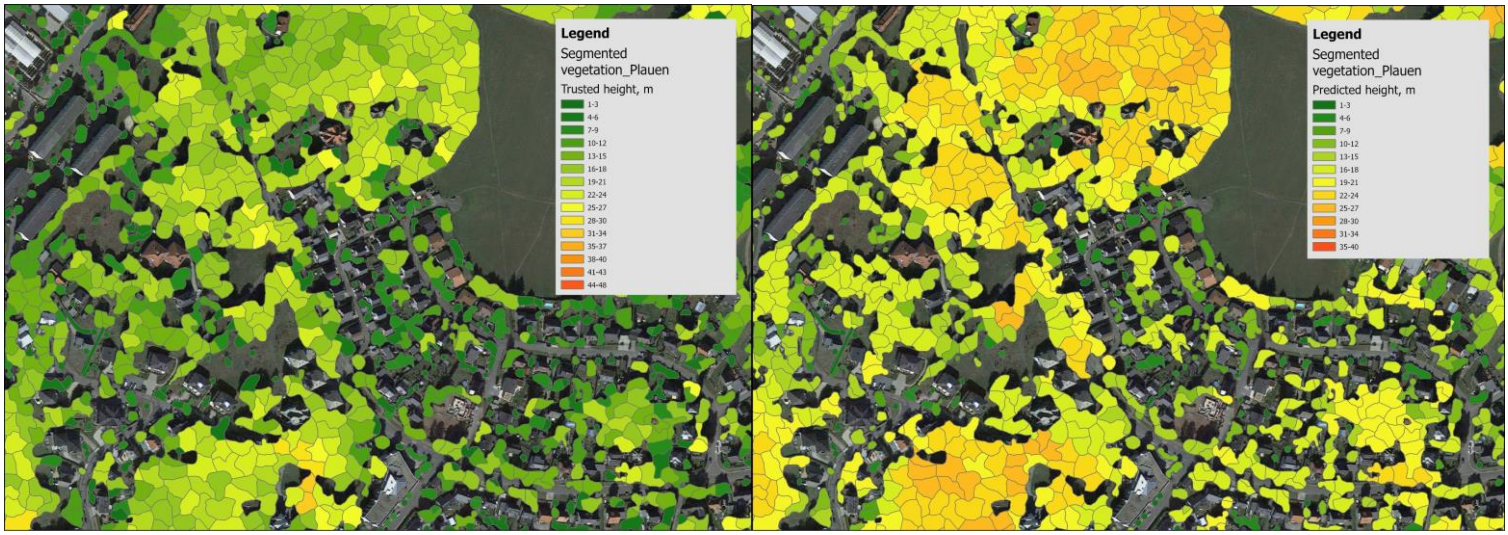


Figure 11. True height (a) and predicted height (b), Plauen city

After applying the Cross-Validation technique, we received the following metrics for the quality of pixels classification:

Difference value	Percent from the total processed pixels
± 3 m	60.99 %
4-5 m	15.2 %
6-7 m	13 %
more than 7 m	12 %

The averaged calculated difference between predicted and true values of tree heights (MAE) is 3.9m.

So, we see that the obtained results demonstrate the high level of correlation between predicted and true (trusted) tree heights

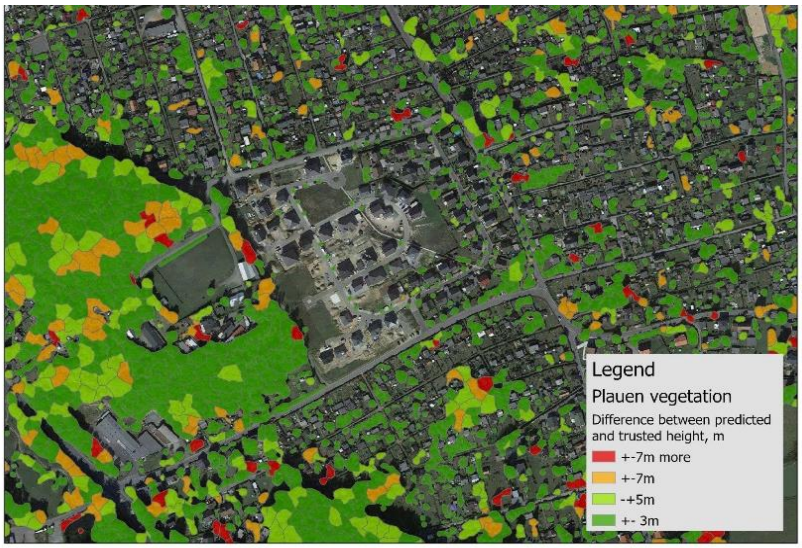


Figure 12.
Calculated difference values between predicted and true tree heights

Step 11: Model deployment for 100+ cities

Thus, we obtained the trained CNN model that performed well in real working conditions on the territory of Germany. We applied it to the other 146 cities of this project, where only aerial photos were available as source materials.

Figure 13 shows the result of processing one of these cities by our CNN model.

Implementing new technology allowed us to dramatically speed up the production process of the 3D vegetation layer without losing the quality and accuracy of the heights.
The previously fitted model can be applied to projects with similar vegetation types, of course, under the conditions of preliminary analysis and validation of its accuracy.

To process areas with other vegetation types, a new CNN model can be created based on the already used algorithm and the presented pipeline.

In any case, convolutional neural networks are promising and powerful instruments for the automation of 3D data extraction from aerial imagery.

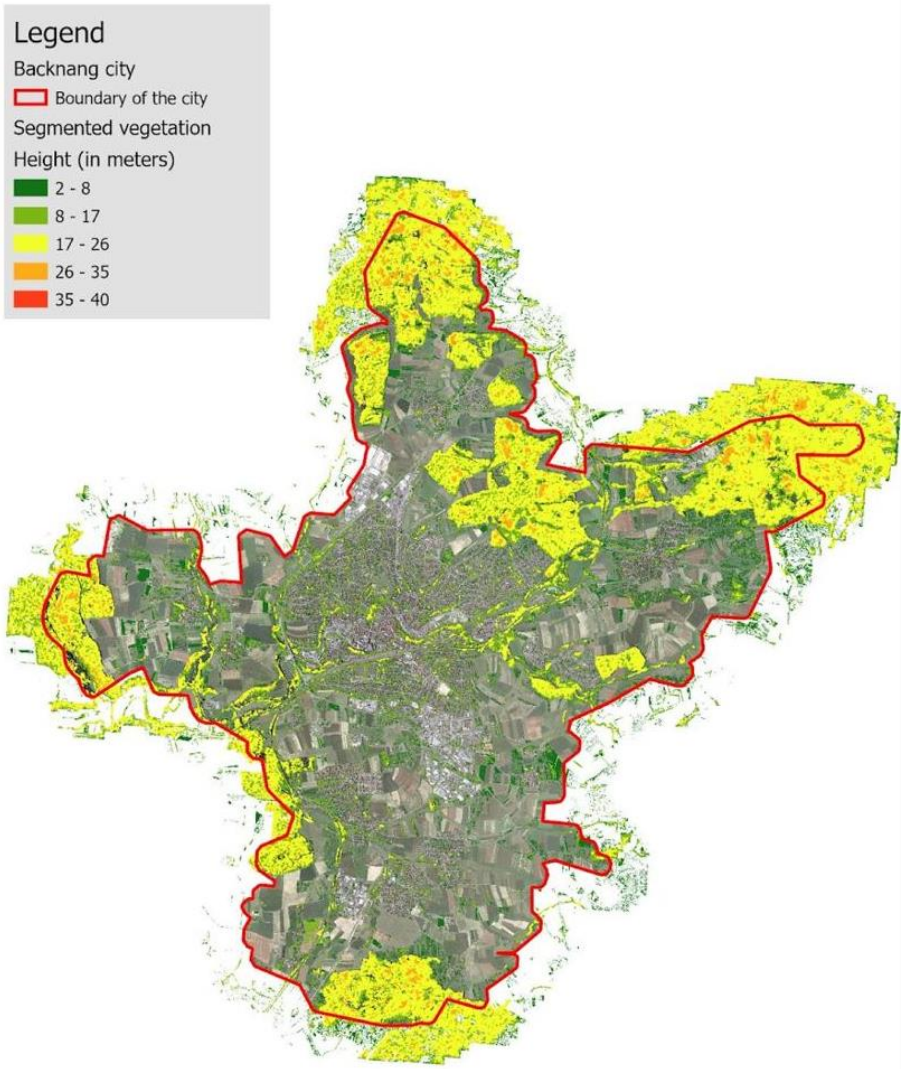


Figure 13. Backnang city with extracted vegetation heights