Comparing Mask R-CNN and Mask2Former architectures for individual tree crown delineation

Sonja Ruschhaupt[1], Jonas Troles[1] and Ute Schmid[1]

Abstract: Weather anomalies caused by the anthropogenic climate crisis challenge environmental workers such as foresters with an increasing number of responsibilities. In addition to providing attractive conditions for new generations of workers, deep-learning tools can ease processes for better efficiency. Tree instance segmentation has potential for many functionalities that support arborists and foresters by detecting and classifying singular trees. In the past, Mask R-CNN was primarily applied due to its outstanding performance on the 2016 COCO dataset challenge. As an alternative, we suggest Mask2Former, which outperforms Mask R-CNN on the COCO dataset. Additionally, we test whether additional digital canopy height model data can improve training. While the latter is shown to have no, if not a negative impact on the results, Mask2Former indeed outperforms Mask R-CNN in tree instance segmentation by up to 3.8%. Our code is publicly available.[2]

Keywords: individual tree crown delineation, instance segmentation, Mask2Former

1 Introduction

With the ongoing emission of greenhouse gases, the anthropogenic climate crisis continues to intensify, disrupting ecosystems worldwide. From 2018 to 2020, central Europe experienced a multi-year drought that resulted in extensive tree mortality [Ob21], and scientists warn of such events recurring [Ra22]. As the list of climate anomalies grows, environmental workers such as foresters face an increasing number of responsibilities. Despite this, Bavaria has seen little to no increase in forestry workers between 2007 and 2022 [Ba24]. To address this, foresters should be equipped with tools that allow them to fulfill tasks more efficiently, as the workload will likely continue to grow.

Computer vision enables the automatic deduction of data on woodlands from imagery taken by unmanned aerial vehicles (UAVs). Thus, foresters can potentially reduce time spent on tedious, repetitive tasks of taking inventory, allowing them to invest more effort in, e.g., targeted preservation work. Instance segmentation, in particular, enables functionalities that require the detection and classification of individual trees. In the past, Mask R-

[1] Otto Friedrich University Bamberg, Cognitive Systems, An der Weberei 5, 96049 Bamberg, firstname.lastname@uni-bamberg.de
[2] https://gitlab.rz.uni-bamberg.de/cogsys/public/bakim/masktreeformer

CNN [He17] was primarily used for tree instance segmentation, as its architecture outperformed the winners of each task in the 2016 challenge on the COCO dataset. However, since then, many new architectures have been developed.

In this paper, we suggest Mask2Former [Ch22] as an alternative to Mask R-CNN in tree instance segmentation. We further investigate whether tree height information as an additional input to RGB imagery provides a benefit for training. To do so, Section 2 serves to give an overview of the works on tree instance segmentation to date, as well as to introduce the architecture of Mask2Former. Section 3 describes our own approach, Section 4 its realization and evaluation. We conclude our work in Section 5 with a prospect of new research questions.

2 State of research

This section will outline the current state of the art in tree instance segmentation and provide a description of Mask2Former, a newer instance segmentation tool that has yet to be utilized for this purpose.

2.1 Instance segmentation tools to support foresters

Instance segmentation enables the detection and classification of multiple individual objects within an image, taking into account the shape of each object. In this process, each pixel in the image is assigned to a specific object using a pixel mask. In the context of tree detection, instance segmentation is done on orthomosaics, which are mosaics of orthogonal photographs to create an aerial view of an area. The segmentation provides many benefits: [Br20] suggest tree instance segmentation to take inventory of tropical forests. [Ch20] work on detecting dead trees to help prevent the spread of uncontrolled forest fires. [Tr23] use instance segmentation to provide tools to arborists and foresters for efficient task planning. They utilize the work of [Ba23] on Detectree2, who research tree instance segmentation to monitor the health of upper-canopy trees, which play an important role in carbon storage. All of the research mentioned utilizes Mask R-CNN [He17], a deep-learning architecture commonly used in tree instance segmentation, as it is known for its performance on the COCO dataset [HB20]. Traditional approaches exist as well but have been shown to be outperformed significantly by deep learning models [Fa24].

Despite the benefits of instance segmentation in forestry, few studies have focused on individual tree crown delineation in this field, possibly due to challenges in obtaining suitable training data [Ka21]. According to [Sc20], another obstacle could be the similarity in crown characteristics between branches and actual tree crowns, posing difficulties for differentiation. They argue that RGB orthomosaics may lack sufficient detail for effective instance segmentation, suggesting the need for additional data such as tree stem locations

or ancillary remote sensing data. In this paper, we provide additional information in the form of tree height information (see Section 3.2).

2.2 Mask2Former

Mask R-CNN is primarily used for instance segmentation tasks for individual tree crown delineation. However, the *Masked-attention Mask Transformer* (Mask2Former) by [Ch22] outperforms Mask R-CNN in instance segmentation when applied to the COCO dataset. Mask2Former is a multi-purpose image segmentation tool for all segmentation tasks of panoptic, instance and semantic segmentation. The architecture, which can be seen in Figure 1, consists of three components: an interchangeable backbone, a pixel decoder, and a transformer decoder. The backbone serves to extract low-resolution features from the input image. These features are passed to the pixel decoder which gradually upscales the input to form a feature pyramid of four layers. The transformer decoder is composed of L transformer sets, each of which consists of three transformer units. Each of the units receives one layer of the feature pyramid, starting with the smallest. To receive mask prediction, the output of the fourth and final pixel decoder layer is multiplied by the output of each transformer unit for mask prediction, which in turn can be fed into the next transformer unit.

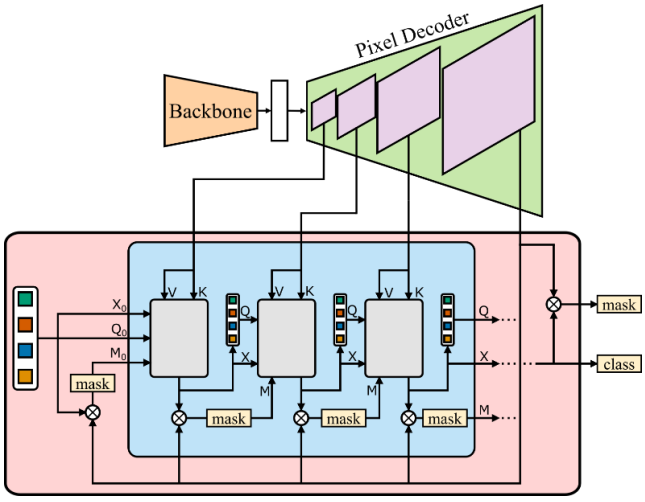


Fig. 1: The architecture and information flow of Mask2Former. Layer indices l are left out with the exception of $l = 0$. The transformer decoder (light red) consists of an L-times repetition of three transformer units (see Fig. 2). Only a single three-unit set is depicted here (light blue). See Section 2.2 for a more thorough description. Transformed from [Ch22]

The structure of a three-layer transformer unit, as used in Mask2Former, can be seen in Figure 2. It varies from a standard transformer [Va17] in that cross-attention is applied before self-attention. Furthermore, an attention mask is utilized in cross-attention. The transformer unit takes four pieces of information as input: the attention mask M, the image features, the query features $\boldsymbol{Q}$ and the output $\boldsymbol{X}$ of the previous transformer unit. The query features $\boldsymbol{Q}$ are calculated from the output $\boldsymbol{X}$ of the previous layer and the query embedding. $\boldsymbol{Q}_0$ is initialized randomly and $\boldsymbol{X}_0 = \boldsymbol{Q}_0$. The mask input M is calculated from the last pixel decoder layer and $\boldsymbol{X}$. Lastly, the image features are taken from the corresponding pixel decoder layer of the feature pyramid, as mentioned before. From them, values $\boldsymbol{V}$ and keys $\boldsymbol{K}$ are derived. The masked cross-attention layer's output is calculated as such:

$$\boldsymbol{X}_l = \text{softmax}(\boldsymbol{M}_{l-1} + \boldsymbol{Q}_l \boldsymbol{K}_l^T)\boldsymbol{V}_l + \boldsymbol{X}_{l-1}\,, \tag{1}$$

where l is the index of the cross-attention layer and $l-1$ that of the fully connected layer coming before it. The mask $\boldsymbol{M}$ is

$$\boldsymbol{M}_{l-1}(x,y) = \begin{cases} 0, & \text{if } M_{l-1}(x,y) = 1 \\ -\infty, & \text{else} \end{cases} \tag{2}$$

thus setting softmax to zero for areas not included in the mask.

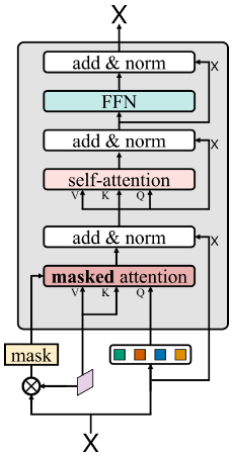


Fig. 2: A three-layer transformer unit as used in Mask2Former. Detailed description in Section 2.2. Transformed from [Ch22]

3 Own approach

With Mask R-CNN being the first choice in individual tree crown delineation and Mask2Former outperforming the architecture on the COCO-Dataset, it follows naturally to apply Mask2Former to tree detection. We hypothesize:

Mask2Former performs better than Mask R-CNN in tree instance segmentation. (1)

As mentioned in Section 2.1, RGB orthomosaics on their own may not provide enough information for achieving usable instance segmentation results [Sc20]. During the labeling process, [Tr24] found that our experts benefit from additional height data. Therefore, we provide Mask R-CNN and Mask2Former with data from the digital crown height model (DCHM) and hypothesize:

The additional data from the DCHM will improve model performance. (2)

A digital surface model (DSM) derived from the orthomosaic generation process in combination with a digital terrain model (DTM) provided by the Bavarian Agency for Digitisation, High-Speed Internet and Surveying was used to create the DCHM. The height is encoded in values from 0 to 255, such that 0 marks the ground and 255 is equivalent to the height of the tallest tree in the dataset. The information is passed to the model via the alpha channel of the PNG image. Henceforth, "RGBA" will be used to describe input including, and "RGB" not including height information. Both Mask2Former and Mask R-CNN can handle a fourth input channel without issue, only requiring minor configurational changes, which will be described in Section 4.1.

4 Realization and evaluation

The Mask2Former model was tested against Mask R-CNN by applying Detectree2's configuration and weights [Ba23]. The dataset used is the Stadtwald and Tretzendorf areas of the BAMFOREST dataset by [Tr24] with a tilesize of 1024 x 1024 px. We apply a training/validation/test split of 7053/1988/1621 images. The exact split can be found in the publication of the BAMFOREST dataset [Tr24]. As all subsets use data from different areas of the same regions of interest, the issue of spatial autocorrelation applies and will be discussed in Section 4.3.

4.1 Parameters

Generally, the training is aborted after not improving AP_{50} for 15 epochs, or after training for 50 epochs. At 50%, 75% and 87.5% of epochs, the learning rate is reduced by a factor of 0.1. We use a batch size of four due to being limited to a singular GPU (a GeForce RTX

3090 with a memory of 24GB). Aside from the necessary changes of hyperparameters that relate to pre-trained weights mentioned further below, the configurations were applied as they were given.

Mask2Former: [Ch22] provide multiple Mask2Former configurations, for instance, segmentation on COCO-like datasets. Here, the configuration for the ResNet50 backbone is used with a learning rate of $5e^{-5}$, which was chosen by trial and error. Training diverged with the suggested learning rate of $1e^{-4}$. Detectron2's ImageNet pre-trained ResNet50 weights are used. As the pre-trained weights for the first backbone layer do not include an alpha channel, this weight is randomized for RGBA training according to Mask2Former's implementation [Ch22]. Due to the random initialization of the backbone for RGBA training, we deactivated the backbone multiplier, which otherwise would reduce the effects of training on the backbone weights.

Mask R-CNN: To evaluate Mask R-CNN, we apply [Ba23] Detectree2 configuration to Mask2Former's trainer, thus only switching out the meta-architecture from Mask2Former to Mask R-CNN, but keeping all other architectural components such as optimizer, data loader and evaluator. We use a ResNet101 backbone as Detectree2 did, in contrast to Mask2Former. Detectree2's hyperparameters are not given as a configuration file but rather set as the default parameters of the model. Thus, they were extracted from the code itself. For the most part, Detectree2's configuration was applied as is. Detectree2's 2023 weights for random resizing were used for pre-training. However, as with the ImageNet pre-training used for Mask2Former, Detectree2 was trained on three-channel input data. The pre-training for the first layer cannot be applied to RGBA training. Therefore, it is randomized and not frozen for RGBA training, and no backbone multiplier is applied.

4.2 Evaluation

Figures 3 and 4 show the loss curve of Mask R-CNN and Mask2Former respectively. At epoch 25, the loss drops slightly for all training configurations due to the learning rate schedule. The change in learning rate is also evident in a rise of AP_{50}, as can be seen in Figure 5, which will be discussed in detail further below.

As loss is acquired differently for both models, an inter-model comparison offers no information on the models. Intra-model comparison, however, shows that in both cases the graph of RGB training runs below that of RGBA training, therefore indicating that height information has a negative influence on tree instance segmentation. This aligns with the AP_{50} evaluation curve for all four training sessions, shown in Figure 5. While the tail of RGB training and RGBA training for both Mask R-CNN and Mask2Former level off at approximately the same value, the maximum AP_{50}^{RGB} is higher than the maximum AP_{50}^{RGBA} for both architectures. The same accounts for the AP_{50} values resulting from testing the models on the test set, listed in Table 1. RGB training moreover reaches its best model

before RGBA training: for both architectures at epoch 26. RGBA training reaches its final model at epoch 32 for Mask2Former and only at epoch 40 for Mask R-CNN. The overall results are within close range of another: RGB training outperforms RGBA training by $\Delta_{intra}^{M2F} = 1.18\%$ for Mask2Former and by $\Delta_{intra}^{D2} = 0.87\%$ for Mask R-CNN. To reduce the risk of comparing models hitting local maxima, t-tests based on multiple runs with random initializations of weights would be necessary.

As for inter-comparison, Mask2Former shows a better performance than Mask R-CNN in both RGB and RGBA training. Mask R-CNN's RGBA training is outperformed by $\Delta_{inter}^{RGBA} = 3.53\%$, its RGB training by $\Delta_{inter}^{RGB} = 3.83\%$. While our Mask R-CNN performs worse than [Fa24]'s Mask R-CNN, trained on the same dataset, our Mask2Former shows equal performance with a deviation of +0.1% on the AP50 score [Fa24]. This difference may be caused by the different tilesize used for training. With a tilesize of 2048 x 2048, Mask2Former might perform notably better than [Fa24]'s Mask R-CNN. This makes Mask2Former a promising alternative candidate for tree instance segmentation. However, even though Mask2Former trained on RGB data shows the best AP_{50} performance, this difference is not necessarily evident or beneficial to human users. Figure 6 shows the input file and the ground truth of one tile taken from the testing set as well as the instance segmentation put out by the four models. While the inter-model performance difference is quite large, with nearly 4%, doing multiple training runs to perform t-tests and check for statistical significance was outside the scope of this study.

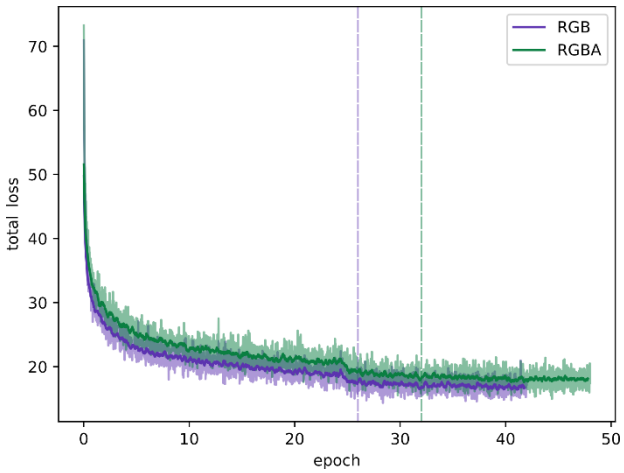


Fig. 3: Loss over epoch for including (RGBA) and not including (RGB) DCHM-data when using Mask2Former. True values in opaque, smoothed values in saturated color for better visibility of temporal progress. Lines mark epochs of best AP_{50} performance

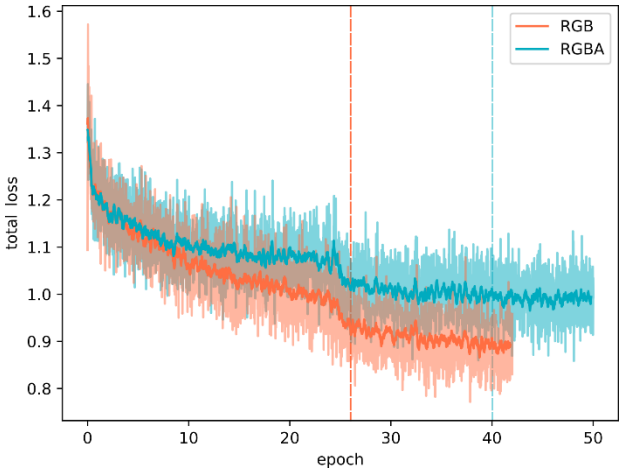


Fig. 4: Loss over epoch for including (RGBA) and not including (RGB) DCHM-data when using Mask R-CNN. True values in opaque, smoothed values in saturated color for better visibility of temporal progress. Lines mark epoch of best AP_{50} performance

.

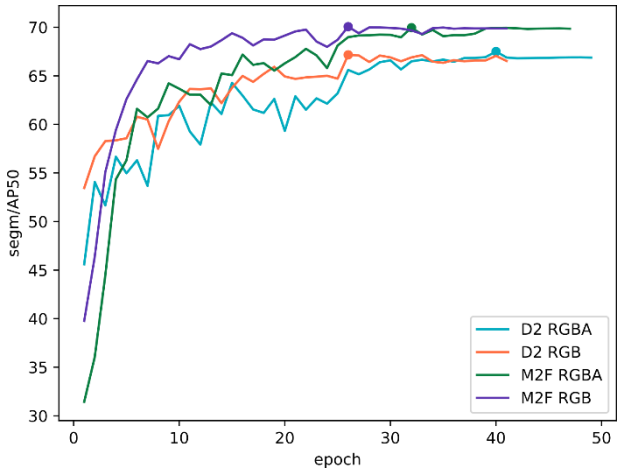


Fig. 5: AP_{50} for instance segmentation over epoch on the evaluation set for including (RGBA) and not including (RGB) DCHM-data when using Mask R-CNN (D2) and Mask2Former (M2F). Points mark epochs of best AP_{50} performance

Segm / AP_{50} in %	RGB	RGBA	Δ_{intra}
Mask2Former	69.05	67.87	1.18
Mask R-CNN	65.22	64.34	0.87
Δ_{inter}	3.83	3.53	

Tab. 1: AP_{50} on instance segmentation for Mask2Former and Mask R-CNN, including (RGBA) and not including (RGB) height information, and differences in performance by model and input

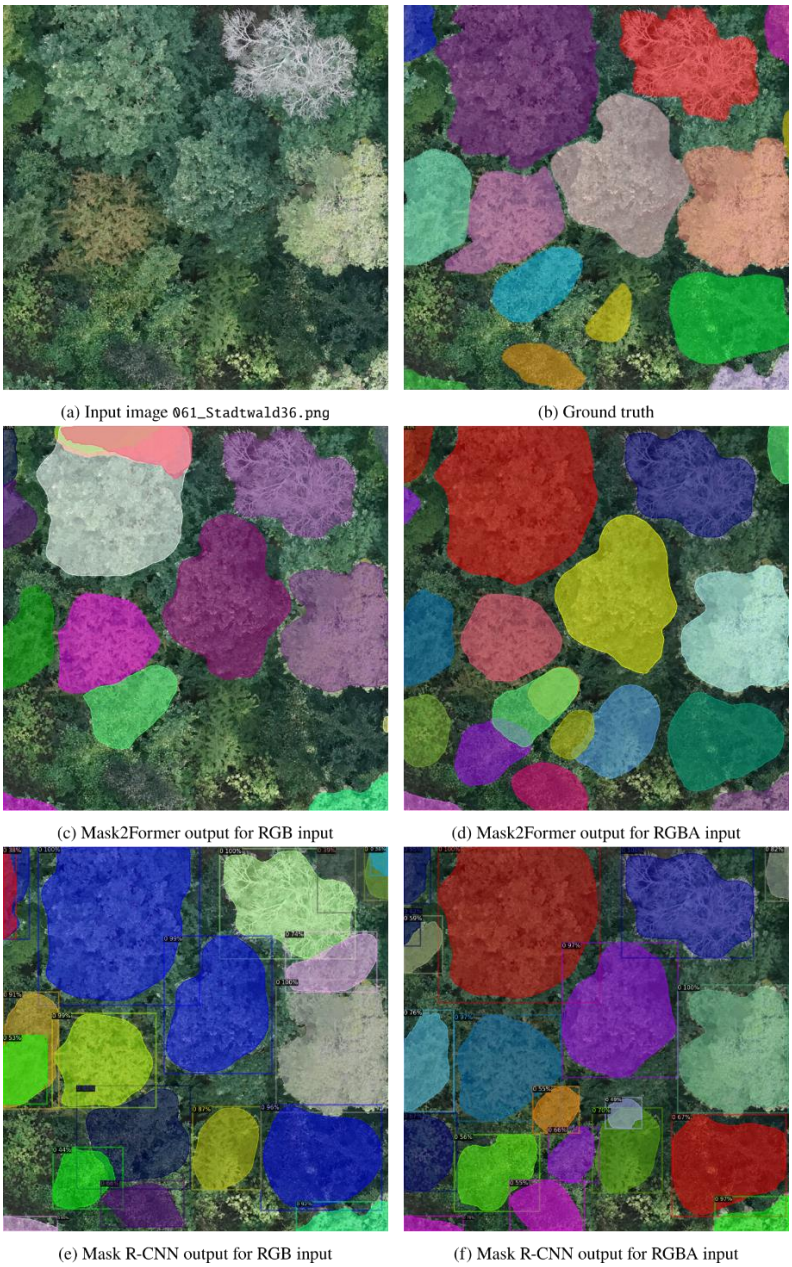
4.3 Limitations

Although the results are promising, several inherent limitations complicate the interpretation of the experiment.

Resources: Without a significance test of the models' performance, our analysis holds less weight. Conducting such a test is only hindered by time constraints put on this paper, as a training run for one model on a single GPU can take over 24h. A dependable t-test would need 25 trainings runs for each model and configuration. While Mask2Former itself was implemented to handle multi-GPU training, our early stopping functionality has yet to be tested in this aspect.

Parameters: The hyperparameters were taken from the configuration files that were provided with the models [Ba23; Ch22]. Only those that relate to pre-trained weights mentioned in Section 4.1 were set accordingly. None of any attempted tweaking of parameters yielded an improvement of model performance for either model. Neither Cheng et al. nor Ball et al. give information on their procedure to find the optimal parameters for the respective models. Searching the hyperparameter space to optimize the models is, however, outside the scope of this paper and our resources.

Dataset: Spatial autocorrelation implies that measurements of variables with spatial dependencies show more similarity (or distinction) when taken closer together compared to those taken further apart [Le93]. This characteristic within our dataset suggests that the model trained on this data has limited applicability. Specifically, the model may perform less effectively on forests in different locations, even if the species composition is similar. Therefore, while our work may benefit local foresters, it is not easily generalizable beyond the immediate area. To address this, training on a more diverse dataset comprising images from different forests captured using various devices could enhance the model's robustness. However, acquiring such data for tree instance segmentation is challenging due to the expertise required for accurate labeling [Ka21].



(a) Input image `061_Stadtwald36.png`

(b) Ground truth

(c) Mask2Former output for RGB input

(d) Mask2Former output for RGBA input

(e) Mask R-CNN output for RGB input

(f) Mask R-CNN output for RGBA input

Fig. 6: Input and output for Mask2Former and Mask R-CNN

5 Conclusion and future work

In this paper, we suggested Mask2Former [Ch22] as an alternative to the commonly used Mask R-CNN [He17] for tree instance segmentation. We further applied height information in an effort to improve training results. While height information had no, if not a negative impact on results, Mask2Former was shown to outperform Mask R-CNN by up to 3.8%. This promising outcome warrants a deeper investigation into Mask2Former, in particular conducting significance tests to validate our findings. Analysis to learn whether Mask2Former requires more resources than Mask R-CNN and, if so, whether these resources are worth the performance increase, is further required. Other studies suggest that a larger tilesize improves model performance. Therefore, a training of Mask2Former with a larger tilesize should be conducted. Lastly, parameter optimization might further enhance Mask2Former's performance on tree instance segmentation. Our work shows that the task of individual tree crown delineation can still be improved. Applied to software products, our findings can support the work of arborists by generating detailed information on forests and improving decision-making and communication with political stakeholders. This is one crucial step to make the effects of the anthropogenic climate crisis more manageable.

Bibliography

[Ba23] Ball, J. G. C. et al.: Accurate delineation of individual tree crowns in tropical forests from aerial RGB imagery using Mask R-CNN. Remote Sensing in Ecology and Conservation, 9/23, S. 641-655, 2023.

[Ba24] Bayerische Landesanstalt für Wald und Forstwirtschaft: Beschäftigte und Umsätze, https://www.lwf.bayern.de/forsttechnik-holz/betriebswirtschaft/050299/index.php, Stand: 02.10.2024.

[Br20] Braga, J. R. G. et al.: Tree Crown Delineation Algorithm Based on a Convolutional Neural Network. Remote Sensing, 12/20, S. 1288, 2020.

[Ch19] Chen, K. et al.: Hybrid Task Cascade for Instance Segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). S. 4969-4978, 2019.

[Ch20] Chiang, C. et al.: Deep Learning-Based Automated Forest Health Diagnosis From Aerial Images. IEEE Access, 8/20, S. 144064-144076, 2020.

[Ch22] Cheng, B. et al.: Masked-Attention Mask Transformer for Universal Image Segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). S. 1290-1299, 2022.

[Fa24] Fan, W. et al.: Comparing Deep Learning and MCWST Approaches for Individual Tree Crown Segmentation. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, X-1-2024/24, S. 67-73, 2024

[HB20] Hafiz, A. M.; Bhat, G. M.: A survey on instance segmentation: state of the art. International Journal of Multimedia Information Retrieval, 9/20, S. 171-189, 2020.

[He17] He, K. et al.: Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). S. 2980-2988, 2017.

[Ka21] Kattenborn, T. et al.: Review on Convolutional Neural Networks (CNN) in Vegetation Remote Sensing. ISPRS Journal of Photogrammetry and Remote Sensing, 173/21, S. 24-49, 2021.

[Le93] Legendre, P.: Spatial Autocorrelation: Trouble or New Paradigm?. Ecology, 74/93, S. 1659-1673, 1993.

[Ob21] Obladen, N. et al.: Tree mortality of European beech and Norway spruce induced by 2018-2019 hot droughts in central Germany. Agricultural and Forest Meteorology, 307/11, S. 108482, 2021.

[Ra22] Rakovec, O. et al.: The 2018–2020 Multi-Year Drought Sets a New Benchmark in Europe. Earth's Future, 10/22, S. e2021EF002394, 2022.

[Sc20] Schiefer, F. et al.: Mapping forest tree species in high resolution UAV-based RGB-imagery by means of convolutional neural networks. ISPRS Journal of Photogrammetry and Remote Sensing, 170/20, S. 205-215, 2020.

[Tr23] Troles, J. et al.: Task Planning Support for Arborists and Foresters: Comparing Deep Learning Approaches for Tree Inventory and Tree Vitality Assessment Based on UAV-Data. In: Innovations for Community Services, S. 103-122, 2023.

[Tr24] Troles, J. et al.: BAMFORESTS: Bamberg Benchmark Forest Dataset of Individual Tree Crowns in Very-High-Resolution UAV Images. Remote Sensing, 16/24, S. 1935, 2024.

[Va17] Vaswani, A. et al.: Attention is All you Need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. S. 6000-6010, 2017.

[Wu19] Wu, Y. et al.: Detectron2. https://github.com/facebookresearch/detectron2, 2019.