# Supplementary Materials for
# Iterative Next Boundary Detection for Instance Segmentation of Tree Rings in Microscopy Images of Shrub Cross Sections

## 1. Influence of Hyperparameters

We additionally evaluate the role of the hyperparameters for INBD and Multicut. Important hyperparameters for INBD are the angular density $\alpha$ that controls the angular resolution $M$ and the number of iterations in one training epoch $n$. The results of our evaluations are presented in Figure 1.

The performance boost of iterative training diminishes and might even have a detrimental effect after 3 iterations. Contrary to our expectations and in contrast to other computer vision tasks like image classification, increasing the angular resolution has a negative effect on the detection performance, we attribute to a lower field of view.

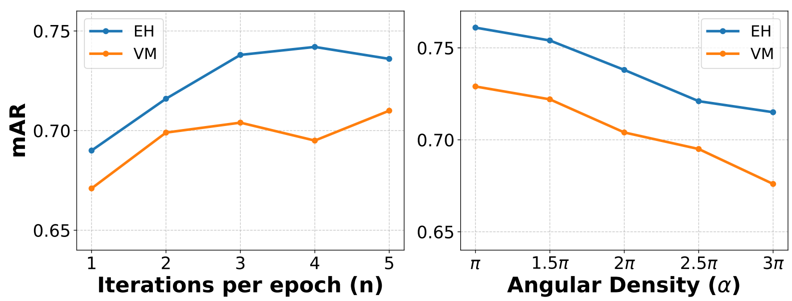



Figure 1. Influence of INBD hyperparameters on the detection performance.

For Multicut we have found that the smoothing factor for the watershed seed map (referred to as `sigma_seeds` in the PlantSeg source code) can be crucial and has to be tuned specifically to the plant species as shown in Figure 2. The results in the main paper show only the best values for each subset.

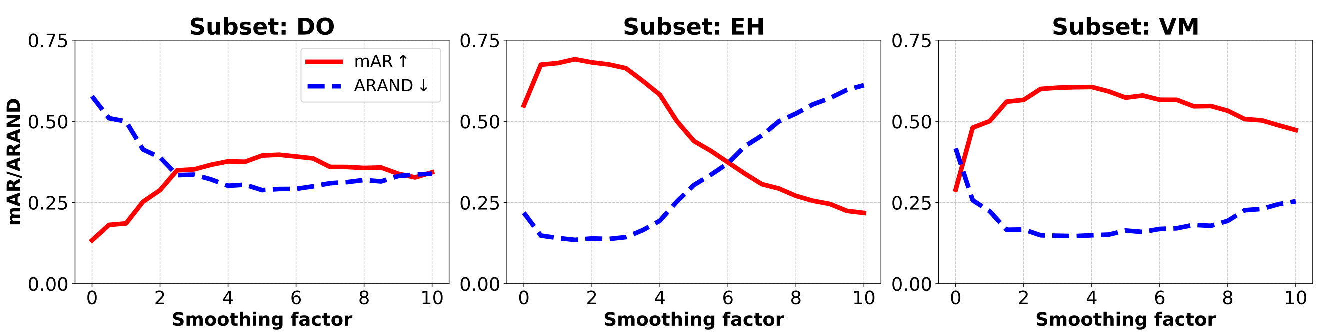
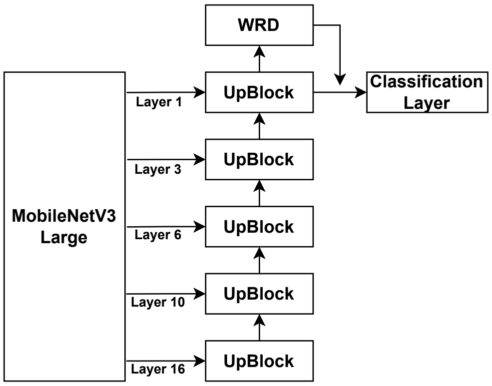



Figure 2. Influence of the Multicut watershed seed map smoothing factor on the detection performance.




(a) High level overview

| Operator | Kernel | # of Channels |
|---|---|---|
| `Upsample (x2)` | | x |
| `Concatenation` | | x+y |
| `Conv2D` | (1,1) | y |
| `CirularConv2D` | (3,3) | y |
| `InstanceNorm` | | |
| `ReLU` | | |

(b) INBD network UpBlock (data flow from top to bottom)

| Operator | Kernel/Stride |
|---|---|
| `MaxPool2D` | (2,1) |
| `Conv2D` | (1,1) |
| `InstanceNorm` | |
| `ReLU` | |
| `MaxPool2D` | (2,1) |
| `Conv2D` | (1,1) |
| `InstanceNorm` | |
| `ReLU` | |
| `Conv2D` | (1,1) |

(c) Wedging Ring Detection (WRD) Module

Figure 3. INBD network architecture

## 2. Network Architecture

For better reproducibility, we report more details on the used network architectures in Figure 3, however we note that our method is not dependent on this specific architecture, other segmentation networks should work as well. For all our experiments we have used a network architecture based on U-Net with a pretrained MobileNetV3-Large [1] backbone as implemented in torchvision (v0.11). This backbone was chosen as a compromise between prediction performance and speed: the high image resolution puts limits on the network size for both training and inference on an end user's device. Circular convolutions [2] are also used in the backbone and the circularity only applies to the angular axis, not to the radial one.

## 3. INBD with Cartesian Coordinates

INBD can in theory work with Cartesian coordinates as well, with the advantage that it is significantly easier to implement. We also evaluate how well this alternative performs. For this, we use the same architecture except with standard convolutions and without WRD. In each iteration step $i$ this network receives the outputs of the 3-class segmentation network $f$ as well as all previously detected rings and it is trained to segment the next ring $i+1$, akin to the our main method, but working on full images and not on polar grids. A basic result and comparison with polar coordinates can be found in Table 3 of the main paper. We observe that this alternative is prone to nonconvexities, an example is shown in Figure 5c. Polar coordinates on the other hand impose a prior on the shape, ensuring that it is coherent and (quasi-)convex.

We note that the image resolution has some influence on the overall detection performance: a high resolution allows for recognition of very indistinct boundaries but comes at the cost of a lower field of view which is needed for long-range dependencies and a consistent ring segmentation. An evaluation of the influence is shown in Figure 4. For our dataset, the optimal resolution lies around $768\times768$ pixels.

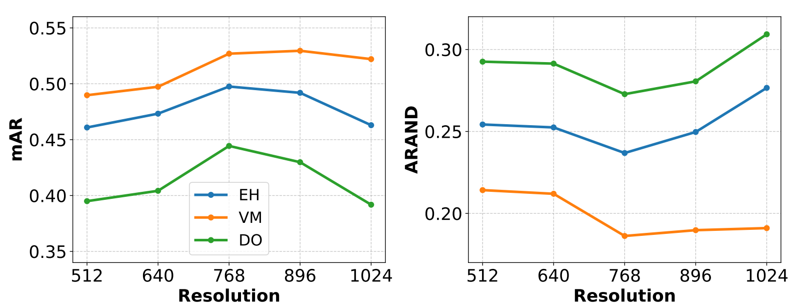
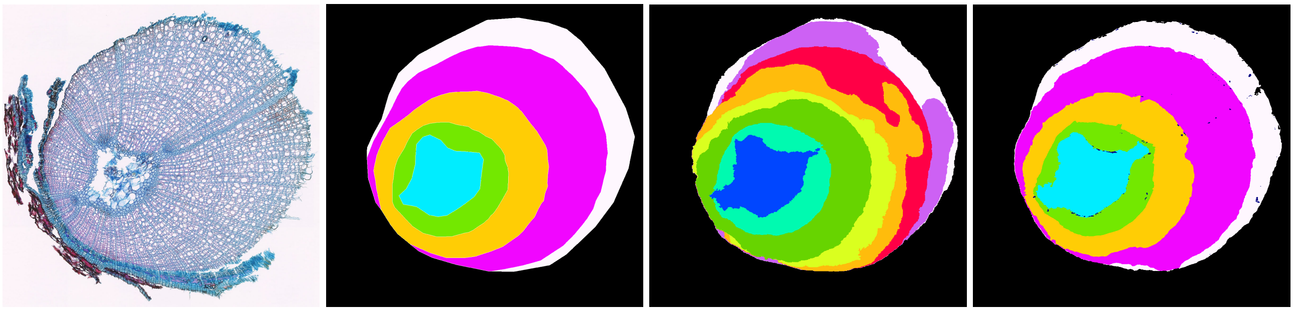
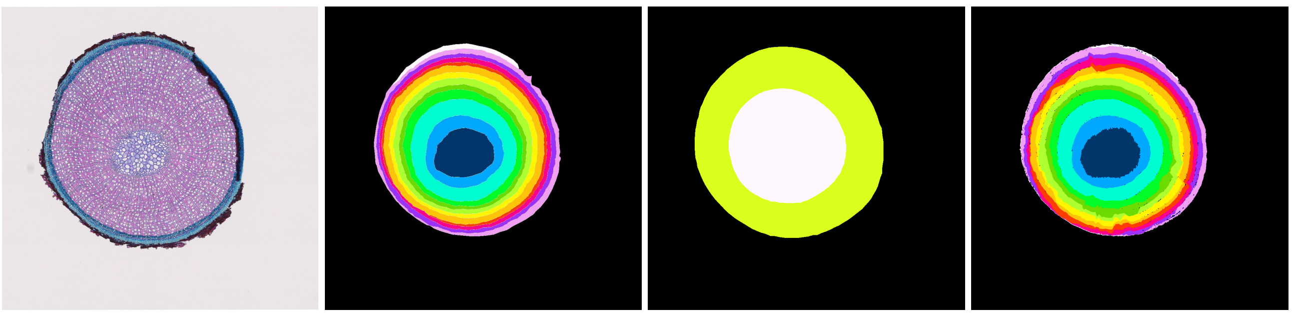



Figure 4. Influence of the image resolution on the performance of the Cartesian baseline



(a) Input (b) Annotation (c) INBD with Cartesian coordinates (d) INBD with polar coordinates

Figure 5. An image from the DO subset and comparison of INBD with Cartesian and polar coordinates. Note the typical nonconvex artifact on the orange ring in 5c.



(a) Input (b) Annotation (c) Deep Snake (d) INBD

Figure 6. An image from the VM subset and comparison of INBD with Deep Snake. Deep Snake inherently struggles detecting concentric objects. The result of INBD is better, correctly estimating the number of rings but still too inaccurate for further processing.

## 4. Additional Images

A comparison with Deep Snake can be seen in Figure 6. More failure cases are shown in Figure 7 which shows the need for more research into this direction. For better understanding of the application background, Figure 8 shows collected branch samples and the landscape where they were collected.

## References


[1] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019. 2

[2] Sida Peng, Wen Jiang, Huaijin Pi, Xiuli Li, Hujun Bao, and Xiaowei Zhou. Deep snake for real-time instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8533–8542, 2020. 2

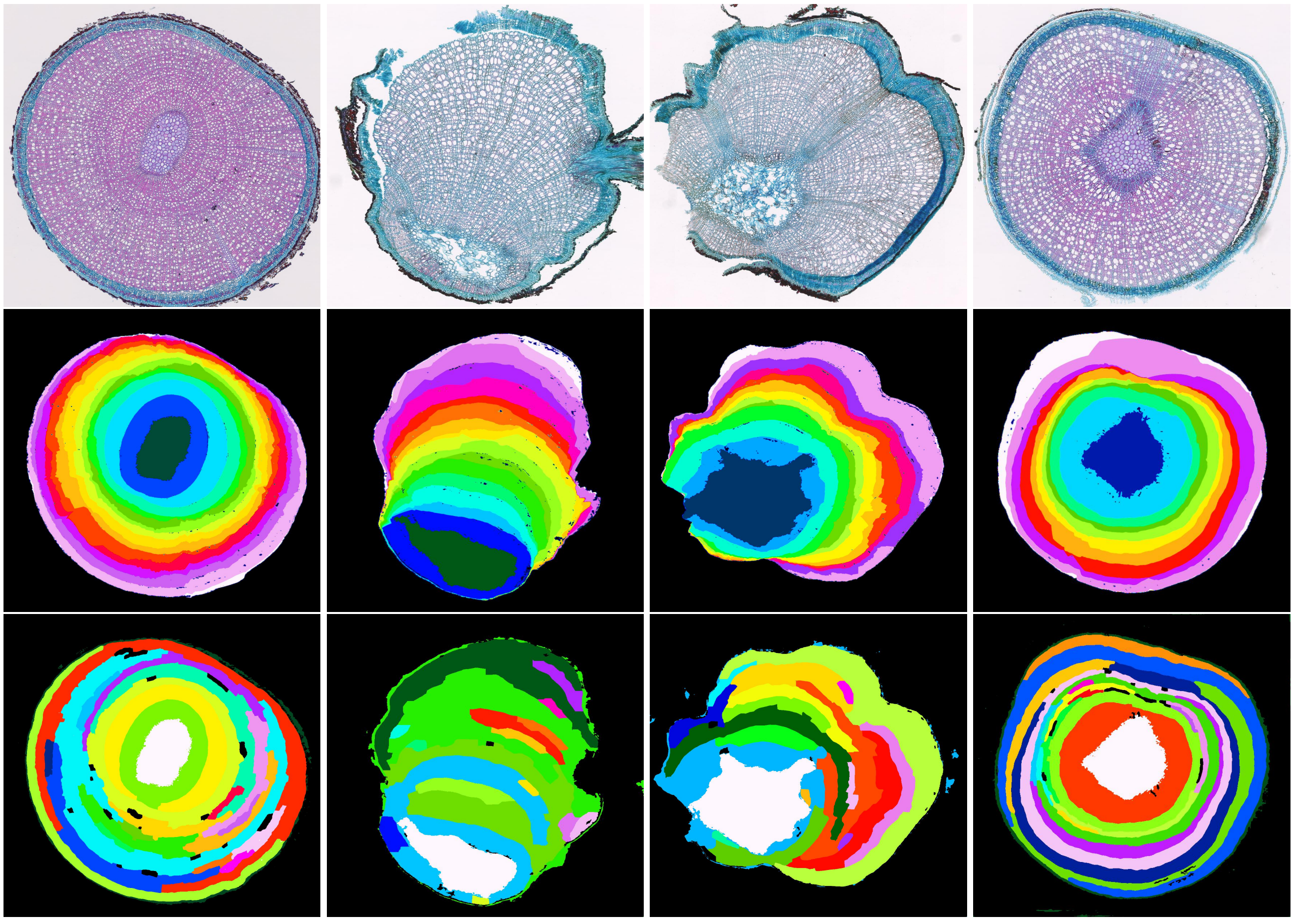


Figure 7. Example images on which none of the compared methods (center: INBD, bottom: Multicut) perform sufficiently well. These images are not in our published dataset because annotators were also not able to fully annotate them.

Figure 8. Images of branch samples (Dryas octopetala) from our dataset and the landscapes where samples were collected.