# A COMPARISON OF STATE-OF-THE-ART NETWORK ARCHITECTURES FOR INSTANCE-SEGMENTATION IN FOREST ENVIRONMENTS.

Lukas Michiels[1]*, Manuel Westermann[1], Benjamin Kazenwadel[1], Chris Geiger[2], Marcus Geimer[1]

*[1] Karlsruhe Institute of Technology / Institute of Mobile Machines, Rintheimer Querallee 2, 76131 Karlsruhe, Germany*
*2 Hohenloher Spezial-Maschinenbau GmbH (HSM), Im Greut 10, D-74635 Neu-Kupfer*

* Corresponding author: E-mail address: lukas.michiels@kit.edu

## ABSTRACT

Research and development have increasingly focused on automating mobile machines to reduce the negative influence of labor shortages and high labor costs. Object detection is a key requirement for the automation of mobile machines. The transfer of the developed methods to the environment of mobile machines, e.g. a forest, a building site, or in mining, is challenging. Objects of the same class can have significantly different phenotypes and the surroundings cannot be controlled, weather as well as lighting conditions can change. Neural networks are the state-of-the-art method for detecting and classifying objects for image sensors. The required datasets as well as network architectures mastering object detection across different forest areas have not yet been presented. We collected two datasets, *MobimaWoodlands* and *MobimaSkidRoads*, one with a handheld camera and one captured while driving on skid roads in different areas and in different seasons. Three network architectures for the instance segmentation with two different backbones were trained on the two datasets to segment stems, trees, and stumps. In a subsequent step, the trained networks were evaluated on two public datasets which have not been used in the training process. With an adapted training pipeline, we achieved a similar accuracy with a slight decrease in the AP of 0.1 on one of the unknown datasets with similar tree specimens. On the second unknown dataset, the AP decreased more significantly by 0.3. The findings highlight that generalization over various forest areas is possible, even in demanding outdoor settings. However, the portability to unknown domains cannot be guaranteed especially if different tree species are present, which continues to be an issue in many applications.

***Keywords:*** Object Detection, Forest, Neural Networks, Instance Segmentation

## 1. INTRODUCTION

Object detection is one of the key requirements for the automation of mobile working machines [1]. In recent years, neuronal networks have become the state-of-the-art approach [2]. Object detection tasks are generally divided into three subdomains: object classification, semantic segmentation, and instance segmentation. Object classification generally provides knowledge of whether an object of a specific class is present in the picture; semantic segmentation provides the class for every object; and instance segmentation separates each instance of the same object class [3].

Various works have already evaluated partial aspects of instance segmentation with neural networks in forest environments, such as object detection or instance segmentation of individual object categories. Two main challenges remain regarding instance segmentation in forest environments. The distinction between individual instances and the background (stuff, [4]) is often ambiguous. In many cases no distinct differentiation between individual trees (foreground) and trees in the background (stuff) is possible. Additionally, most research in the computer vision area is focused on the COCO Dataset ([5]) and only a handful of public datasets for forestry environments are available [6], [7]. Datasets are key aspects of object detection, especially in environments with many varying objects. An optimal algorithm can recognize all possible variations of the specific objects (perfect generalization). When the evaluation dataset deviates from the training dataset, e.g., if a tree species is not included in the training dataset, the algorithm needs to be able to extrapolate the features of the object class (e.g. trees) to recognize other possible variations. Encountering data outside the training dataset's scope (out-of-distribution data) is inevitable in forest environments. A dataset featuring four seasons, sunny and cloudy days, coniferous, deciduous, and mixed forests for three different age classes already has at least $4 \cdot 2 \cdot 3 \cdot 3 = 72$ different parameter combinations. In integrating different tree species and additional objects, the number of parameter combinations increases rapidly, and a huge number of images are already required to cover the parameter space, even without considering the different phenotypes of the same tree species. The same issue comes up regarding the bias of different object classes. Generating a forest dataset with an equal number of human instances and tree instances is unfeasible. In consequence, object recognition for autonomous mobile working machines requires robustness regarding biased and incomplete training datasets.

In recent years, many new architectures have been presented. Mostly the improved accuracy comes with increased complexity demanding higher computational power. In the environment of mobile machines, transferability to unknown areas is often more important than accuracy on one collected training dataset. Computational power is costly, and it is in dispute whether increasing the complexity is worthwhile for the application in mobile machines. In this paper, we compared three state-of-the-art instance segmentation architectures on two new datasets, *MobimaWoodlands*, and *MobimaSkidRoads,* as well as their performance when applied to a completely unknown dataset from a different area. The validation dataset consisted of images that had been separated from the dataset and had not been used in the training of the network. However, they share the same tree variations as already encountered in the training data set. In contrast to the validation data, we focused on the transferability of the trained networks to an unknown forest area with different tree variations and a different environment. For this purpose, we used the public *FinnWoodlands* and *CanaTree100* datasets [6], [7]. The implemented training pipeline focus on transferability of the trained models instead on maximizing the mean average precision (mAP) over a single dataset.

The paper is structured as follows: First, the state if the art is reviewed in Section 2. Section 3 presents the dataset and the object classes. In Section 4 the setup and the training pipeline are described. Section 5 evaluates the training results and transferability to other datasets before Section 6 concludes the paper and gives an outlook on further approaches.

## 2. STATE OF THE ART

### 2.1. Evaluation Metrics

Several evaluation metrics are known to evaluate the quality of the predictions generated by the instance segmentation algorithms [8]. We focus on the Intersection over Union (IoU) and the Average Precision (AP).

#### *Intersection over Union*

The intersection over union (IoU) is a metric of the accuracy of individual bounding boxes and masks. This metric, also known as the Jaccard index, is calculated from the predicted label $B_{pr}$ and the ground truth $B_{gt}$ according to the following equation [9]:

$$IoU = \frac{B_{pr} \cap B_{gt}}{B_{pr} \cup B_{gt}} \tag{1}$$

A predicted mask/bounding box was a correct detection, if its IoU with the ground truth was higher than a specified threshold. The IoU threshold is given as percent, e.g. $AP_{50}$ for an $IoU > 0.5$

### *Average Precision and Recall*

Based on the IoU with regard to the ground truth, the following cases were separated:

- True positive (TP): Correct detection of an object.
- False negative (FN): Not detected existing (labeled) object.
- False positive (FP): Incorrect detection of an object that does not exist or a misaligned detection of an existing object.

The true-negative case of a correctly undetected annotation is omitted in the context of object detection due to the large number of theoretically possible bounding boxes that should not be detected.

Based on these cases, the precision and the recall are defined as:

$$Precision = \frac{TP}{TP + FP}, \qquad Recall = \frac{TP}{TP + FN} \tag{2}$$

The precision is a metric for the performance of the network with regard to the detected objects. The recall is, on the other side, a metric for the performance with regard to all existing objects. Both the precision and the recall depend on the IoU threshold for true positives. If the IoU threshold is small, the recall is typically larger while the precision decreases. The precision-recall curve is constructed from all recall and precision values for an IoU threshold from zero to one. The interpolated area under the precision-recall curve for a given IoU threshold is denoted by Average Precision (AP) [8]. The $AP_{0.5:0.95}$ is the average of the AP for the IoU thresholds from 0.5 to 0.95, $IoU \in \{0.50, 0.55, \ldots, 0.95\}$ [9]. The mean Average-Precision (mAP) is the arithmetic mean of all AP values of each class.

## 2.2. Network Architectures

Three different network architectures, Mask R-CNN, Cascade Mask-RCNN, and Mask2Former were regarded in the following work and explained in detail. The Mask R-CNN architectures are two-stage architectures with a region proposal network and a region of interest head. The Mask2Former architecture on the other hand is a single-stage transformer-based architecture. All architectures require a backbone network for feature recognition. In the state of the art, further architectures were used in forest environments, e.g., Rotated Mask R-CNN, Yolact++ and EfficientPS which will not be discussed in detail.

### *Mask R-CNN*

Mask R-CNN ([10]) is a two-stage detector architecture developed as an extension of the original Faster R-CNN architecture [11]. The first stage consists of the backbone (e.g. a fully convolutional network) and a region proposal network, which determines the regions of interest. In the second stage, the algorithm predicts the class, a bounding box, and a binary mask for each region of interest. In addition to the Faster R-CNN architecture, Mask R-CNN has additional convolutional layers for mask generation. Figure 1 depicts an overview of the Mask-RCNN architecture. In the second stage, all regions of interest can be evaluated in parallel.

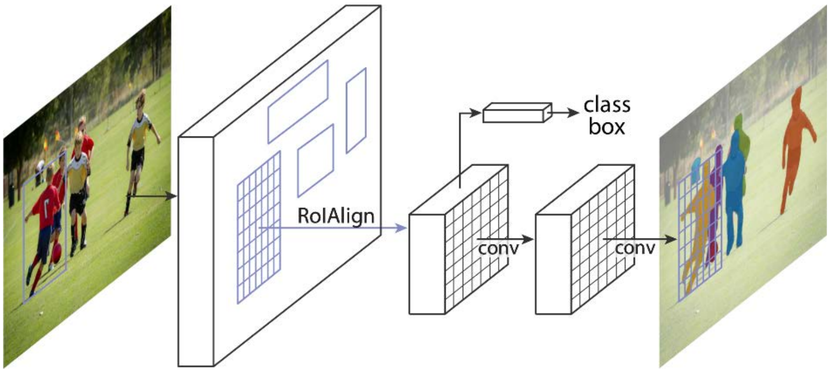


**Figure 1: The Mask R-CNN framework for instance segmentation [10].**

### *Cascade Mask-RCNN*

Cascade Mask R-CNN is an extension to the original Mask R-CNN framework to improve high Intersection over Union (IoU) threshold detections [12]. Cascade Mask R-CNN is a two-stage architecture composed of a series of detectors in the second stage. The detectors are trained in parallel, each with the detector output of the previous. Each subsequent detector has a higher IoU threshold than the previous. This adapted architecture should improve the hypotheses' quality and guarantee a positive training set for each detector. Additionally, it aims to minimize overfitting. The first stage is identical to the Mask R-CNN architectures, using identical backbones.

### *Mask2Former*

Mask2Former is a transformer-based universal algorithm for image segmentation [13]. In contrast to the other two architectures presented, it utilizes transformers for the segmentation tasks and can be theoretically used to perform panoptic, instance, and semantic segmentation without retraining. The algorithm consists of three modules: the backbone, the pixel decoder, and the transformer decoder. The backbone architecture (e.g. Swin or Resnet) extracts low-resolution feature maps from the input image. These are then upscaled to high-resolution feature masks by a pixel decoder. The transformer decoder uses the features of the pixel decoder to generate object queries. The output masks are a combination of the feature maps from the pixel decoder and the object queries from the transformer decoder.

## 2.3. Feature Backbones

All the presented instance segmentation architectures have in common that they use a backbone for feature extraction. There are a vast number of different backbones available, each with advantages and drawbacks. In this study, we focus on two backbones, the residual network ResNet presented by He et al., with a depth of 101 layers [14] (R101) and the Swin-T backbone [15] of Liu et al. The Swin-T backbone is a shifted window transform backbone. The Swin-T backbone addresses the challenge of high resolutions and different feature scales by a hierarchical transformer architecture. The image is partitioned into regular windows and the self-attention is computed within each window. In the next layers, the windows are shifted. This architecture aims to provide flexibility for different feature scales while maintaining linear computational complexity with respect to image size.

## 2.4. Applications in Forestry

The first application of instance segmentation for forestry equipment focused on log detection for autonomous grasping with a forwarder grapple. For this purpose, Fortin et al. recorded the TimberSeg 1.0 dataset, consisting of 220 images [16]. They used Mask R-CNN, Rotated Mask R-CNN, and Mask2Former for the segmentation. The lowest accuracy was achieved by Mask R-

CNN, followed by Rotated Mask R-CNN with a 12 percentage points higher mean Average Precision (mAP) of 31.38 %. By far the best results were achieved by Mask2Former with a mAP of 57.53%. The algorithms showed good robustness against changing external influences such as snow, glare, and darkness. Geiger et al. investigated the partial automation of the loading process [17]. The YOLACT++ architecture was used to detect and segment the stems before gripping. The trained network reached a mAP over all classes of 56.65.

Deep learning and image classification with neuronal networks were studied by Liu et. Al in [18]. Liu et al. classified tree specimens and stock volume to provide a more efficient and faster alternative to conventional ground surveys. Grondin et al. created the synthetic dataset *SYNTHTHREE43K* consisting of 43,000 images, and the *CanaTree100* dataset, consisting of 100 RGB and depth images of Canadian forests in different weather conditions [7], [19]. The annotated masks and bounding boxes include trees and other objects such as stumps and grass. In addition to the masks and bounding boxes, the labels also include the position of the cut, the diameter, and the inclination. They used Mask R-CNN and Cascade Mask R-CNN with different backbones for the instance segmentation. After initial training on synthetic images in [19], the model was adapted to the real images in [7]. The Cascade Mask R-CNN algorithm gave better results than Mask R-CNN on all backbones, however, both algorithms reached high mean average precision of AP-50 > 85 and AP-50:95 > 60 for both bounding boxes and segmentation. In the second step, the trained model was applied to an unknown Portuguese dataset. A significant degradation of the results was observed.

Lagos et al. aimed to create a dataset for data-driven methods in forest environments [6]. Their *FinnWoodlands* dataset contains 300 RGB stereo images, point clouds, and sparse depth maps. They provide manual annotations for semantic, instance, and panoptic segmentation. The instance categories include three types of trees and the obstacles: “Lake”, “Ground” and “Track”. In addition to providing the data, they evaluated the instance segmentation with Mask R-CNN and EfficientPS where they achieved an AP-50 of 28% and 50% respectively.

## 3. DATASETS

For this study, two annotated datasets of forest areas have been created. The *MobimaWoodlands* (doi:10.35097/1749) dataset consists of two subsets, *MobimaWoodlands/Winter* and *MobimaWoodlands/Summer*, with 126 images each. Both subsets are captured manually with handheld cameras. This dataset features typically middle-European mixed forests in summer and in winter. The summer subset is in 16:9 landscape format, while the winter subset is in 4:3 portrait orientation. The second dataset, *MobimaSkidroads* (doi:10.35097/1750), consists of 293 images captured while driving on a skid or forest road with an industrial camera mounted on a vehicle. The dataset is completely in 16:9 landscape format and includes mixed and coniferous forests in summer and winter. Example images with annotations are displayed in Figure 2.

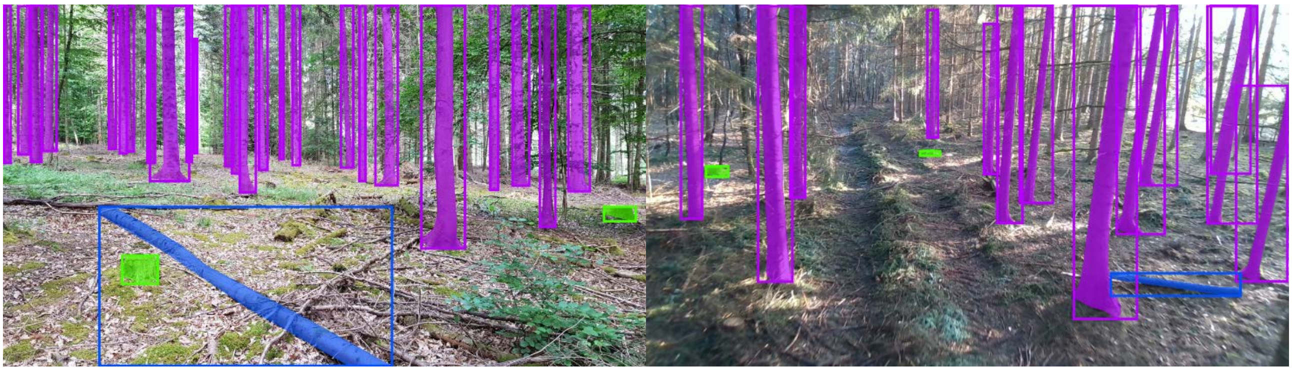


**Figure 2: Example images with annotations (blue: stems, purple: trees, green: stumps) of *MobimaWoodlands* (a) and *MobimaSkidRoads* (b)**

The datasets include mask and bounding box annotations for the five object classes described in Table 1. The number of objects per class is given in Table 2.

**Table 1: Object classes**

| Class | Description |
|---|---|
| Stems | Trunks and cutted trees with a diameter of approx. $\geq 10\ cm$ |
| Trees | Trees with a diameter of approx. $\geq 10\ cm$ |
| Stumps | Tree stumps |
| Obstacles | Non-passable objects not being subject to any of the other groups (e.g. stones, raised hides, way signs) |
| Persons | Complete or partially visible humans |

The datasets were randomly divided into a training set and a validation set by a ratio of 0.8/0.2. The number of objects per class shows a significant bias. The number of annotated tree instances exceeds largely the other classes, while obstacles and persons are almost not present. However, unbiased datasets are challenging as the number of trees exceeds the number of stems, stumps, and especially humans present in a forest.

**Table 2: Number of objects per class in the training/validation sets**

| Class | *MobimaWoodlands/ Winter* | *MobimaWoodlands/ Summer* | *MobimaSkidRoads* |
|---|---|---|---|
| Stems | 175/34 | 177/54 | 83/34 |
| Trees | 425/108 | 697/232 | 2762/672 |
| Stumps | 13/1 | 53/17 | 67/19 |
| Obstacles | 6/0 | 7/11 | 32/1 |
| Persons | 0/0 | 10/1 | 0/0 |
| Images | *100/26* | *100/26* | *234/59* |

## 4. TRAINING SETUP

Three different network architectures, Mask R-CNN, Cascade Mask-RCNN, and Mask2Former were implemented in the following comparison. The architecture implementations from the PyTorch-based library MMdetection were employed [20]. In this study, the Swin-T backbone is used for each architecture except for the Mask R-CNN architecture, where we compared the results of the R101 backbone with those of the more complex Swin-T backbone.

All backbones were initialized with pre-trained weights from the COCO dataset. The weights were obtained from the MMDetection library. Training detection backbones is time-consuming; pre-trained backbones allow fast transfer learning on new datasets and minimize the computational time for training adapted detectors. Additionally, the first stages of the backbone layers have been frozen, as studies indicate that retraining the complete backbone has no significant benefit [21]. For the Mask R-CNN and Cascade Mask R-CNN architectures, the first backbone stage was frozen. For the training of the Mask2Former architecture, all backbone stages were frozen to reduce the number of trainable parameters.

In addition to the two published datasets, we included the *Mobimalogs* dataset from [17] in our training. To prevent a possible bias due to the different dataset sizes, the data loader created a mini-batch of 5 samples with a sample ratio of [1,1,2,1] from *MobimaWoodlands/Winter*, *MobimaWoodlands/Summer*, *MobimaSkidRoads*, and *MobimaLogs* respectively.

The training results largely depend on the training pipeline. Typically, a random image section is extracted from the base image at each training epoch. This section can be a part of the image or, with a certain probability, be the complete image. In contrast to this approach, the training pipeline depicted in Figure 3 uses a random section of each image but never the complete image to prevent overfitting of the networks to the surrounding area. Each image was resized first to a maximal size randomly sampled from the interval $l_{px} \in [1280, 3072]\, px$ while conserving the aspect ratio. In a subsequent step, a random section with size: $width\ x\ height = 800x800px$ was taken from the resized image. The section was flipped horizontally with a probability of $50\%$. Hence, only image parts with a ratio of $0.625x$ to $0.26x$ of the original image were utilized in the training process. This training pipeline reduced the performance of the trained networks when trained on a single dataset but increased it when different datasets with various tree aspects were present.

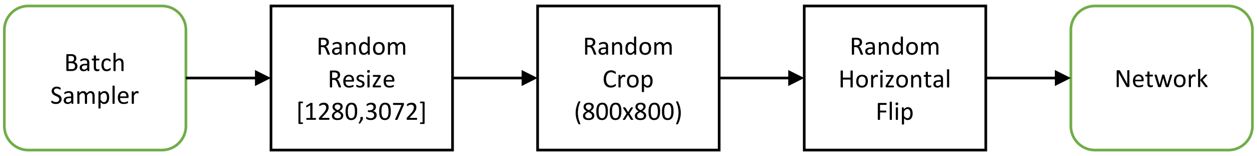


**Figure 3: Training Pipeline**

Due to the large bias in the datasets, evaluating the mAP would largely overrepresent the influence of the obstacle and person instances. In contrast to most studies evaluating the mAP, we evaluate the AP on a per-class basis.

## 5. RESULTS

The trained architectures are evaluated on the validation sets of the training datasets and the complete dataset of unknown forest areas represented by the *CanaTree100* and the *FinnWoodlands* datasets.

### 5.1. Training Datasets

At first, the trained network architecture was evaluated on the validation sets. Figure 4 depicts the AP on a per-class basis for the validation set of *MobimaWoodlands/Summer* for all four network architectures and the backbone (in parentheses). In the whole validation dataset, only one person is present, and all networks predicted the person with at least an IoU of 0.5. On the other hand, the random separation has put many of the obstacles in the validation set, and therefore all architectures struggle with the recognition of the obstacles, especially as this class includes various object types. This example illustrates, that the mean Average Precision is not a suitable metric when working with biased datasets. All four architectures perform similarly on the tree class, with a small benefit for the Mask2Former architectures on stem and stump classes.

The results for all datasets are given in Table 3. The highest value for each class and dataset is displayed in bold. Both the MaskRCNN (Swin) and the Mask2Former perform better on some classes for the same datasets. In total, transformer backbones and the transformer architecture of Mask2Former did not lead to improvements regarding the average precision. In combination with the high AP levels of an IoU threshold of 0.5, this leads to the conclusion that the overall performance of the networks is not limited by the architecture but by the quality of the input datasets. Especially, the distinction between trees in the foreground, which are labeled, and trees in the background, which are not labeled, is ambiguous and therefore varies between images and datasets.

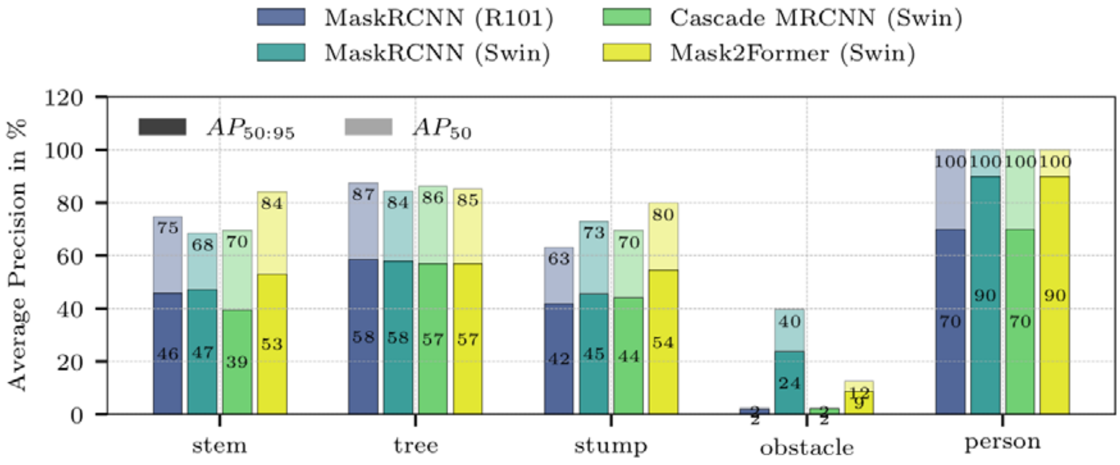


**Figure 4: Average Precision ($AP_{50}$ stacked on $AP_{50:95}$) of all architectures on the *MobimaWoodlands/Summer* validation set**

## 5.2. Generalization

In the next step, the trained networks were evaluated on the dataset *CanaTree100* ([7]) and *FinnWoodlands* ([6]). As these datasets had not been used for the training, the networks were evaluated on the complete sets, including both, the training and the validation set. For the other datasets, that had been used for the training, the networks were evaluated on the validation set and not on the training set. The generalization of the networks was analyzed for the tree class, as the other classes are not consistent across all datasets. Figure 5 shows the AP of the tree class for all datasets. On the *CanaTree100* dataset, the AP remained in the same range of $AP_{50:95} \in [0.42,0.58]$ with no significant variation between the architectures. On the *FinnWoodlands* dataset, however, the AP of all architectures diminished significantly.

**Table 3: Average Precision per class for each dataset (*bold: best AP per dataset*)**

| | *Stem* | *Tree* | *Stump* | *Obst.* | *Person* | *Stem* | *Tree* | *Stump* | *Obst.* | *Person* |
|---|---|---|---|---|---|---|---|---|---|---|
| ***MW/Winter*** | *MaskRCNN (R101)* | | | | | *MaskRCNN (Swin)* | | | | |
| *Box* $AP_{50:95}$ | 0.62 | 0.49 | 0.70 | - | - | 0.63 | 0.47 | 0.80 | - | - |
| *Segm* $AP_{50:95}$ | 0.39 | 0.48 | **0.80** | - | - | 0.42 | 0.45 | **0.80** | - | - |
| | *Cascade MRCNN (Swin)* | | | | | *Mask2Former (Swin)* | | | | |
| *Box* $AP_{50:95}$ | 0.65 | 0.48 | **0.90** | - | - | **0.78** | **0.51** | 0.80 | - | - |
| *Segm* $AP_{50:95}$ | 0.44 | 0.44 | **0.80** | - | - | **0.71** | **0.51** | 0.70 | - | - |
| ***MW/Summer*** | *MaskRCNN (R101)* | | | | | *MaskRCNN (Swin)* | | | | |
| *Box* $AP_{50:95}$ | 0.54 | **0.56** | 0.39 | 0.01 | 0.70 | **0.56** | **0.56** | 0.41 | **0.28** | **1.00** |
| *Segm* $AP_{50:95}$ | 0.46 | **0.58** | 0.42 | 0.02 | 0.70 | 0.47 | **0.58** | 0.45 | **0.24** | **0.90** |
| | *Cascade MRCNN (Swin)* | | | | | *Mask2Former (Swin)* | | | | |
| *Box* $AP_{50:95}$ | 0.47 | 0.57 | 0.42 | 0.02 | 0.80 | 0.55 | 0.50 | **0.49** | 0.08 | **1.00** |
| *Segm* $AP_{50:95}$ | 0.39 | 0.57 | 0.44 | 0.02 | 0.70 | **0.53** | 0.57 | **0.54** | 0.09 | **0.90** |
| ***MSkidRoads*** | *MaskRCNN (R101)* | | | | | *MaskRCNN (Swin)* | | | | |
| *Box* $AP_{50:95}$ | 0.26 | 0.60 | 0.30 | **0.90** | - | 0.30 | 0.59 | **0.37** | **0.90** | - |
| *Segm* $AP_{50:95}$ | 0.12 | 0.47 | 0.30 | **0.80** | - | 0.15 | 0.47 | **0.35** | **0.80** | - |
| | *Cascade MRCNN (Swin)* | | | | | *Mask2Former (Swin)* | | | | |
| *Box* $AP_{50:95}$ | 0.26 | **0.62** | 0.32 | **0.90** | - | **0.34** | 0.47 | 0.35 | 0.70 | - |
| *Segm* $AP_{50:95}$ | 0.12 | **0.48** | 0.31 | **0.80** | - | **0.30** | 0.42 | 0.34 | 0.70 | - |

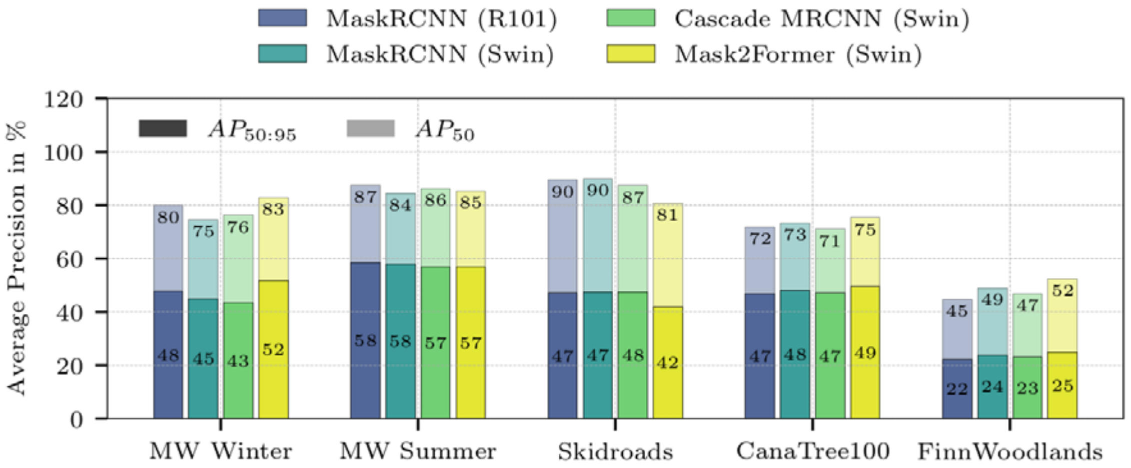
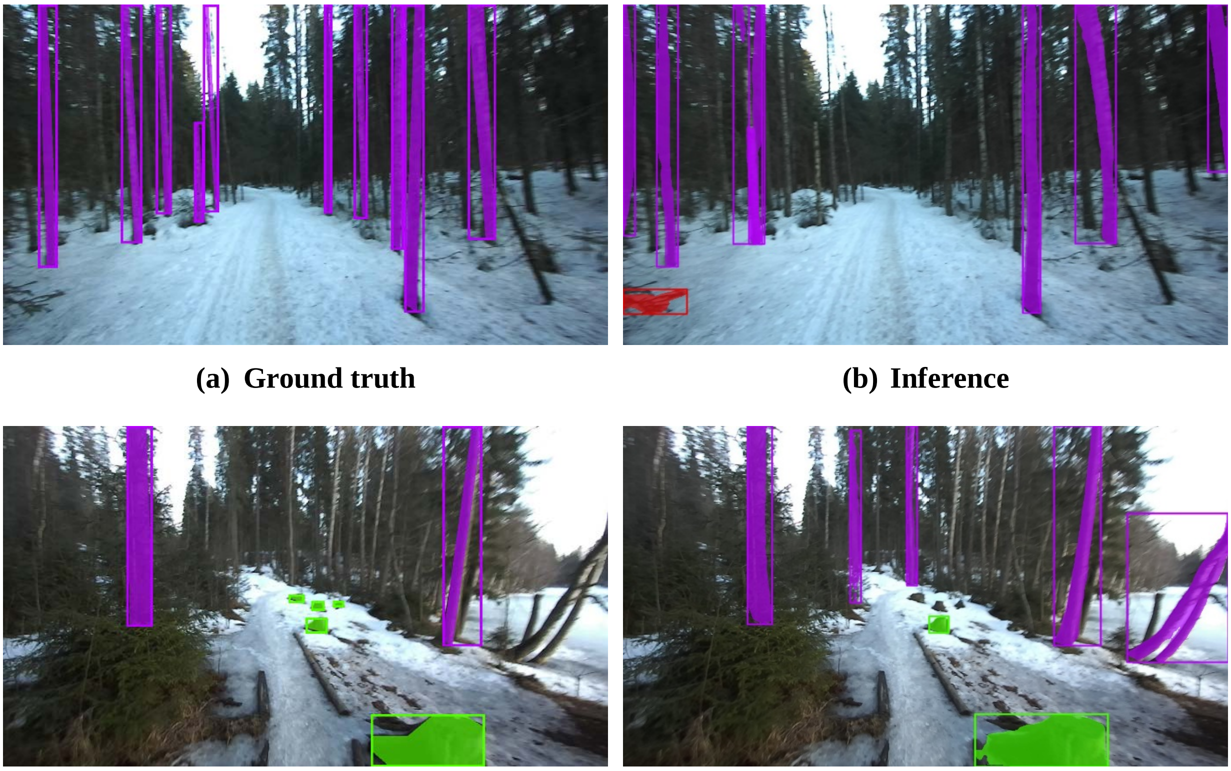


**Figure 5: Average Precision ($AP_{50}$ stacked on $AP_{50:95}$) of all architectures for the *Tree* object class**

To explain the difference between the CanaTree100 and the FinnWoodlands results, three possible explanations are likely. The CanaTree100 dataset is probably more similar to our datasets and therefore requires less generalization of the network. Additionally, the FinnWoodlands dataset includes many birch trees which are not present in our datasets and therefore unknown to the networks. Figure 6 (a) and (b) illustrate False Negative detection of birch trees. Finally, missing labels reduce the AP, as seen in the example of Figure 6 (c) and (d). The network correctly identifies tree instances that were not labeled as trees. This ambiguity of the ground truth labels remains a major challenge when creating datasets and training image detectors for trees with different datasets.



(a) Ground truth

(b) Inference

(c) Ground truth

(d) Inference

**Figure 6: Inference examples of the Mask2Former architecture on the FinnWoodlands dataset (blue: stems, purple: trees, green: stumps, red: obstacles).**

The distribution of True Positives, False Negatives, and False Positives is displayed in Figure 7. The previous assumption that the remaining error on the training datasets is due to unlabeled trees, which are recognized as False Positives, is supported by the high percentage of False Positives, especially regarding the Mask2Former architecture. On the *FinnWoodlands* dataset, the number of False Positives detections is not larger than on the other datasets. On the contrary, many False Negatives were not detected, indicating that especially the undetected (birch) trees were responsible for the decreased AP.

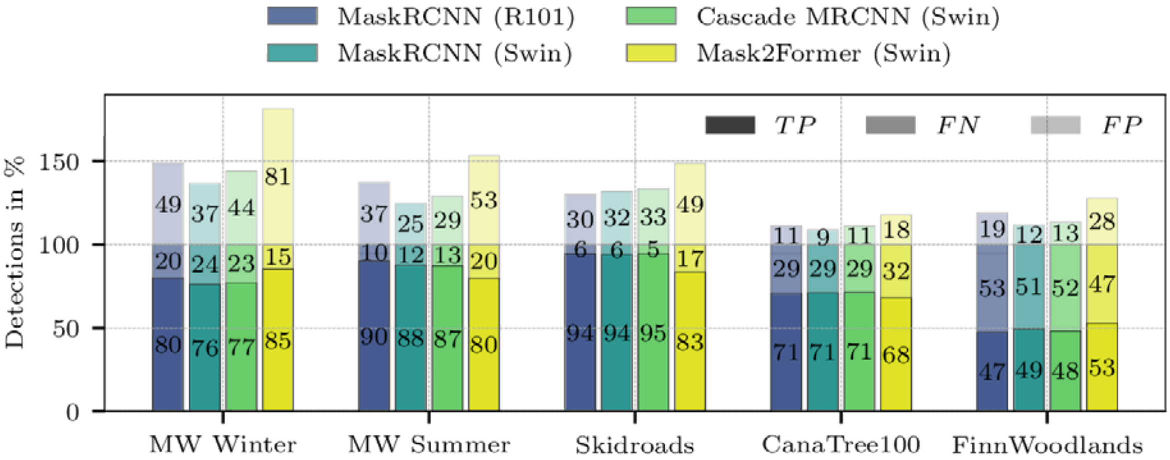


**Figure 7: Error causes of tree classes (stacked: True Positive (IoU > 0.5), False Negative, False Positive, 100% = TP+FN)**

## 6. CONCLUSION

In this paper, we trained four instance segmentation architectures with two different backbones on two newly generated datasets. All four architectures achieved a high Average Precision for the three primary object classes: stems, trees, and stumps. For IoU thresholds of 0.5 an Average Precision of above 0.80 is reached for the tree class with less than 10% False Negative tree detections. A qualitative evaluation of the detection results indicates that most of the remaining false detections are due to incorrect labels. Deciding whether a tree is in the foreground and should be labeled as a tree or whether it is part of the background is ambiguous and therefore prone to errors. Similar issues arise for stems and stumps, whether they are still distinguished objects or part of the forest floor. In a subsequent step, the trained networks were evaluated on two public datasets *CanaTree100* and *FinnWoodlands*. The generalization of the network was analyzed on the tree class as the other classes are inconsistent across the datasets. On the *CanaTree100* dataset, the $AP_{50}$ decreased only about 0.1 for all architectures. On the *FinnWoodlands* dataset, however, the $AP_{50}$ decreased by 0.3 points, and the detection result had around 50% false negatives detections. The high number of False Negatives could be explained by the fact that some of the tree specimens of the *FinnWoodlands* were not present in the training dataset. The findings indicate, that adapted training procedures and large datasets improve the transferability of the networks to unknown forest areas. However, the performance cannot be guaranteed, especially if the training dataset is from a different vegetation zone. More complex network architectures and new transformer architectures as well as transformer backbones, however, did not lead to better detection and generalization results than the Mask R-CNN architecture.

In the future, different approaches could be beneficial to further improve the generalization of object detectors in forest environments. The amount of available training data from different locations, representing different phenotypes and environmental conditions, needs to be increased and the quality of these datasets has to be ensured. New approaches, including stuff classes for background, could be beneficial to solve the ambiguity of background objects. Generative adversarial networks on the other hand could provide fictional images to train the detector on various object phenotypes and environmental conditions.

## ACKNOWLEDGMENT

We like to thank ForstBW and Matthias Gauß for the possibility of collecting parts of the dataset on their skid roads. This publication is part of a project funded by the Federal Ministry of Education and Research (BMBF) under funding number 01IS23018 A-C.

## REFERENCES

[1] T. Hellström, P. Lärkeryd, T. Nordfjell, and O. Ringdahl, *Autonomous Forest Machines - past, present and future*. 2008.

[2] R. Sharma, M. Saqib, C. T. Lin, and M. Blumenstein, “A Survey on Object Instance Segmentation,” *SN COMPUT. SCI.*, vol. 3, no. 6, p. 499, Sep. 2022, doi: 10.1007/s42979-022-01407-3.

[3] O. Elharrouss, S. Al-Maadeed, N. Subramanian, N. Ottakath, N. Almaadeed, and Y. Himeur, “Panoptic Segmentation: A Review.” arXiv, Nov. 19, 2021. doi: 10.48550/arXiv.2111.10250.

[4] G. Heitz and D. Koller, “Learning Spatial Context: Using Stuff to Find Things,” in *Computer Vision – ECCV 2008*, D. Forsyth, P. Torr, and A. Zisserman, Eds., in Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2008, pp. 30–43. doi: 10.1007/978-3-540-88682-2_4.

[5] T.-Y. Lin *et al.*, “Microsoft COCO: Common Objects in Context.” arXiv, Feb. 20, 2015. doi: 10.48550/arXiv.1405.0312.

[6] J. Lagos, U. Lempiö, and E. Rahtu, "FinnWoodlands Dataset," in *Image Analysis*, R. Gade, M. Felsberg, and J.-K. Kämäräinen, Eds., in Lecture Notes in Computer Science. Cham: Springer Nature Switzerland, 2023, pp. 95–110. doi: 10.1007/978-3-031-31435-3_7.

[7] V. Grondin, J.-M. Fortin, F. Pomerleau, and P. Giguère, "Tree detection and diameter estimation based on deep learning," *Forestry: An International Journal of Forest Research*, vol. 96, no. 2, pp. 264–276, Apr. 2023, doi: 10.1093/forestry/cpac043.

[8] R. Padilla, S. L. Netto, and E. A. Da Silva, "A survey on performance metrics for object-detection algorithms," in *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, IEEE, 2020, pp. 237–242.

[9] R. Szeliski, *Computer vision: algorithms and applications*. Springer Nature, 2022.

[10] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE international conference on computer vision (ICCV)*, Oct. 2017.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., Curran Associates, Inc., 2015.

[12] Z. Cai and N. Vasconcelos, "Cascade R-CNN: High Quality Object Detection and Instance Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 5, pp. 1483–1498, May 2021, doi: 10.1109/TPAMI.2019.2956516.

[13] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 1290–1299.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[15] Z. Liu *et al.*, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9992–10002, Oct. 2021, doi: 10.1109/ICCV48922.2021.00986.

[16] J.-M. Fortin, O. Gamache, V. Grondin, F. Pomerleau, and P. Giguère, "Instance Segmentation for Autonomous Log Grasping in Forestry Operations," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2022, pp. 6064–6071. doi: 10.1109/IROS47612.2022.9982286.

[17] C. Geiger, M. Weißenböck, and M. Geimer, "Assistance System for an Automatic Loading Process," in *Proceedings of The Joint Annual 43rd Annual Meeting of Council on Forest Engineering (COFE) & the 53rd International Symposium on Forestry Mechanization (FORMEC)*, Online, 2021, pp. 5–7.

[18] J. Liu, X. Wang, and T. Wang, "Classification of tree species and stock volume estimation in ground forest images using Deep Learning," *Computers and Electronics in Agriculture*, vol. 166, p. 105012, Nov. 2019, doi: 10.1016/j.compag.2019.105012.

[19] V. Grondin, F. Pomerleau, and P. Giguère, "Training Deep Learning Algorithms on Synthetic Forest Images for Tree Detection," May 2022.

[20] K. Chen *et al.*, "MMDetection: Open MMLab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.

[21] S. Hinterstoisser, V. Lepetit, P. Wohlhart, and K. Konolige, "On Pre-Trained Image Features and Synthetic Images for Deep Learning," presented at the Proceedings of the European Conference on Computer Vision (ECCV) Workshops, 2018, pp. 0–0.