This WACV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

TimberVision: A Multi-Task Dataset and Framework for Log-Component Segmentation and Tracking in Autonomous Forestry Operations

Daniel Steininger Julia Simon Andreas Trondl Markus Murschitz
AIT Austrian Institute of Technology (Center for Vision, Automation & Control)
{daniel.steininger,julia.simon,andreas.trondl.fl,markus.murschitz}@ait.ac.at

Abstract

Timber represents an increasingly valuable and versatile resource. However, forestry operations such as harvesting, handling and measuring logs still require substantial human labor in remote environments posing significant safety risks. Progressively automating these tasks has the potential of increasing their efficiency as well as safety, but requires an accurate detection of individual logs as well as live trees and their context. Although initial approaches have been proposed for this challenging application domain, specialized data and algorithms are still too scarce to develop robust solutions. To mitigate this gap, we introduce the TimberVision dataset, consisting of more than 2k annotated RGB images containing a total of 51k trunk components including cut and lateral surfaces, thereby surpassing any existing dataset in this domain in terms of both quantity and detail by a large margin. Based on this data, we conduct a series of ablation experiments for oriented object detection and instance segmentation and evaluate the influence of multiple scene parameters on model performance. We introduce a generic framework to fuse the components detected by our models for both tasks into unified trunk representations. Furthermore, we automatically derive geometric properties and apply multi-object tracking to further enhance robustness. Our detection and tracking approach provides highly descriptive and accurate trunk representations solely from RGB image data, even under challenging environmental conditions. Our solution is suitable for a wide range of application scenarios and can be readily combined with other sensor modalities.

1. Introduction

Wood has been a valuable resource since prehistoric times, gaining further importance since the industrial age due to its multitude of uses. Today, it is used not only for furniture, building materials and paper, but also fabric, insulation, bio-gasoline and many other products. While the processing of wood is often highly automated, the early stages of its value chain still require significant human labor and involve considerable risk, as forestry workers are among the group with the highest fatality rates per full-time employee [33]. Automating operations such as cutting, processing and handling logs as well as forest inventory holds the potential to increase both safety and efficiency. Several works have focused on automating harvesters and forwarders, but due to the high complexity of the tasks, these systems either operate only in ideal testing scenarios [22] or experience dangerous failures due to the lack of visual input data [15, 37]. In our opinion, these limitations mostly result from the lack of a modern, scalable and efficient machine-learning solution for identifying logs along with the training data it requires. Our objective is to develop a dataset and approach addressing the autonomous or assisted harvesting, loading and measuring of logs, i.e. cut stems and branches. A detection approach has to provide not only their position but ideally also their orientation and contour as well as the positions of their cut and lateral surfaces, as visualized by the annotation examples in Fig. 1. This information is essential for deriving geometric cues such as middle axis and boundaries. We aim to infer these features exclusively from

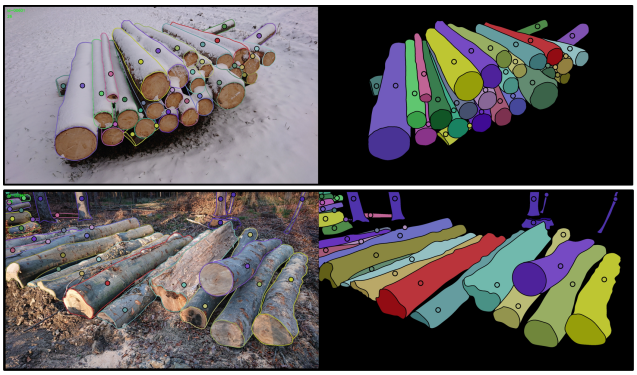

Figure 1. Representative examples of semi-automatically generated annotations for instance segmentation of multiple trunk components in the TimberVision dataset.

RGB data to provide an easily accessible solution, which can be used on its own or combined with 3D information to estimate the center of gravity required for precisely grasping and therefore efficiently manipulating each log.

Learning-based vision approaches are considered a key technology for autonomous operations in unstructured environments. However, covering all application scenarios from natural forests to more ordered saw-mill environments requires significant efforts in acquiring and preparing relevant image data, as well as a specialized approach for training and fusing multiple learning tasks. To address these research objectives, we propose the following contributions:

- We provide a novel multi-task image dataset and efficient annotation pipeline focusing on logs and their components in various application scenarios along with multiple annotated scene parameters.[1]
- We conduct comprehensive ablation experiments for real-time-capable oriented object detection (OOD) and instance segmentation (ISEG) and analyze the impact of scene configurations on model performance.
- We introduce an extensible framework for fusing both learning tasks, correlating individual components into unified object representations, deriving their geometric information and tracking them over time.

2. Related work

The degree of automation in forestry operations is still low, partially due to the lack of training and test data capturing the required variability for robustly detecting the organic shapes of tree trunks in this highly unstructured environment. Some existing approaches heavily rely on 3D data captured by LIDAR sensors, either using model-based cylinder fitting [31, 34] or learning-based approaches [4, 21, 42] for detecting live trees. While they perform well for inventory applications in uncluttered forests, they are usually not applicable for log piles, dead trees or dense vegetation. Similarly, satellite or aerial imagery [8, 9] does not provide the perspective required for handling individual logs. Synthetically generating data, as in [12], holds the advantage of high quantity, but requires a lot of effort to bridge the gap to real imagery. The most readily available data are ground-based RGB images. There are multiple navigation-focused datasets containing trees, either in the context of urban scenarios [5, 18, 24, 30, 46] or unstructured outdoor environments [16, 23, 43]. While all of them provide segmentation masks, they, too, provide only live trees as one of multiple classes. The number of datasets exclusively focusing on the detection of tree trunks in RGB images is very limited. The ForTrunkDet dataset [6] provides axis-aligned bounding-box annotations for live trees on 2.9k RGB and thermal images, but does not contain segmentation masks or cut logs. CanaTree100 [11] contains 100 images with segmentation masks for 920 live trees. The TimberSeg dataset [10], on the other hand, focuses exclusively on cut logs, omitting live trees. It consists of 220 images with segmentation masks for 2500 trunks and is therefore the most closely related to our application domain. However, in images with log piles, the annotation only includes their top layer, thereby limiting the applicability for tasks such as log counting, volume estimation or handling multiple logs at once. None of the existing datasets differentiates between individual trunk components such as cut and lateral surfaces or includes both live and cut trees. Furthermore, there are few datasets, even in other domains, incorporating individual meta-parameters, such as daytime [38] or weather [32], and even fewer providing a comprehensive analysis of data variability and model robustness [40, 49]. Our TimberVision dataset incorporates all these features and additionally contains the largest number of instance-segmentation masks in real RGB images in the domain of forestry operations.

Object detection is one of the first learning tasks addressed by many established learning-based architectures [3, 26, 27, 35]. While most of them use axis-aligned bounding boxes, recent works such as DOTA [45, 47] show the advantages of oriented bounding boxes (OBBs) for elongated objects, which provide a significantly better representation for detecting and tracking cut logs with arbitrary poses. Similarly, pixel-wise class assignment can be conducted using semantic segmentation [28, 48], but the more recent development of instance segmentation [13] provides separated contours of individual objects and is therefore more relevant for our purpose. Panoptic segmentation [19, 25] could provide the most descriptive representation, but requires a full annotation of all visible classes, most of which are not relevant for our application. For both selected tasks, there are approaches based on convolutional neural networks [17, 44] as well as vision transformers [41, 51], with the former still providing competitive accuracy and the advantages of faster inference along with more efficient deployment.

3. The TimberVision dataset

To address the data gap in the domain of forestry operations, we gathered a comprehensive set of relevant images from multiple sources and designed a process to efficiently derive scene parameters and annotations, as detailed below.

3.1. Image acquisition

Since the amount of publicly available image data suitable for tree-trunk detection is limited, we recorded our own data in two stages. Initially, we focused on forests accessible by public transportation and hiking, mainly around Vienna, but also including other areas of Austria, Slovakia

[1] Dataset, code and models are available for academic use at https://github.com/timbervision/timbervision

and the Czech Republic. We captured about 3k images with a total of eleven sensors (including smartphones, an SLR camera and a UAV) on more than 40 separate days between December 2021 and June 2024, thereby covering a wide range of environmental and lighting conditions as well as seasonal effects. Resolutions range from 1280x720 to 3000x2000 pixels with a median of 2016x1512. Each day typically includes multiple sites containing different numbers and arrangements of logs as well as varying backgrounds. Later on, we complemented the data with images recorded by a ZED 2 camera while operating a crane throughout different loading and harvesting scenarios. While the first setup is designed to maximize the variety of log appearances and environmental conditions, the latter is more closely related to specific application scenarios. In both cases, we verified that the published images do not contain any sensitive personal information.

3.2. Scene parameters

Efficiently choosing a representative but feasible set of samples for further annotation and later evaluating the performance of trained models in specific scenarios both require a high-level description of scene properties and context. Beside a binary tag for images containing snow, we defined four parameters and assigned them one of three intensities for each image, as visualized in Fig. 2:

- **Entropy** describes the arrangement of logs varying from parallel stacks to unstructured wood piles.
- **Quantity** defines the number of depicted trunks, with 8 and 30 marking the minimum counts for *Mid* and *High* settings, respectively.
- **Distance** roughly categorizes the offset between the majority of trunks and the camera, thereby indicating the number of truncated logs at image borders. Images completely filled with logs are labeled as *Low*, while *High* indicates the visibility of entire instances.
- **Irregularity** refers to the shape of trunks, ranging from approximately cylindrical to highly uneven.

3.3. Instance annotation

Since annotations are conducted in-house, we aim to derive them for a maximum of different learning tasks with as little manual effort as possible. With open-source tools as the preferred solution, Scalabel's [39] lane-detection setup turned out to be a good fit for our requirements. As visualized in Fig. 3, it allows the definition of trunk components, including their surface discontinuities, as poly-lines based on only a few selected points. Each instance is assigned a label differentiating between lateral *Edges* separating logs

Figure 2. Representative image samples for Low, Mid and High intensities of annotated scene parameters. The color bars show their distributions across the dataset.

from the each other or the background, visible *Section Areas* as well as *Section Lines* denoting the visible borders of cross-sections facing the other way. Additional points unambiguously denote areas covered by each trunk between its constituent lines. Fitting all markers and lines facilitates annotation of instances down to a width of 8 pixels. As a final step, each component is assigned an ID unique to the trunk instance it belongs to or ID 0 for live trees and rooted stumps. For selected sequences, IDs are set consistent over time to allow the evaluation of multi-object tracking.

Based on these manual point annotations, detailed two-dimensional representations for multiple tasks are derived automatically. Annotated *Edges* and *Section Lines* are interpolated as smooth splines, while *Section Areas* are used for fitting either ellipses for regular logs or closed splines for more complex forms. These boundaries combined with the area markers form comprehensive representations of three types of trunk components, which can be converted into arbitrary formats. *Side* surfaces represent only the lateral area of trunks, *Cut* surfaces their cross sections and *Bound*aries their back-facing sections. *Trunks* can include multiple of these components. As a final step, the generated annotations are verified by human annotators. In the future, we plan to use our own results or foundation models such as Segment Anything [20] or CLIPSeg [29] for generating pre-annotations and thereby further increasing efficiency.

3.4. Dataset statistics

We annotated more than 2k images containing about 26k trunks categorized into six subsets by their origin and de-

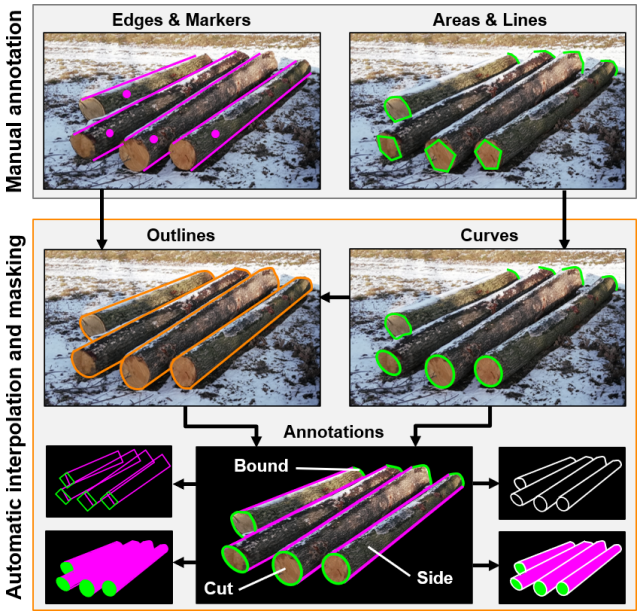

Figure 3. Overview of our annotation pipeline automatically deriving multi-task annotations from manual point annotations.

Subset	Rec	Images	Trunks	Comp
Core	71	1,415	17,999	33,445
Loading	18	220	3,853	8,782
Harvesting	10	45	825	1,720
OpenSource	42	42	776	1,634
Tracking	*13*	*266*	*1,900*	*3,708*
TimberSeg*	*10*	*35*	*940*	*2,049*
	164	**2,023**	**26,293**	**51,338**

Table 1. TimberVision dataset statistics. **Images** are clustered into **Rec**ording sessions based on timestamps and GPS tags, ensuring a difference of at least two minutes and one hundred meters between them. **Trunks** can consist of up to three **Comp**onent classes.

picted scenarios, as listed in Tab. 1. *Core* contains the images captured in forests and other outdoor locations. It is complemented by extensions consisting of *Loading* and *Harvesting* scenarios with visible machinery and third-party *OpenSource* data. *Tracking* consists of keyframes evenly sampled from video sequences at 2 FPS and annotated with consistent object IDs over time.

Tab. 2 relates our dataset to the most similar existing works. The most relevant synthetic dataset is SynthTree43k. However, like CanaTree100, it focuses on live trees, which are only a secondary target of our work. The closest match in terms of scenarios and input modalities is TimberSeg, which, however, provides only a fraction of the image and instance quantities in TimberVision and less detailed annotations, as only the top layers of log piles and no

Dataset	Images	Trunks	C	R	L
SynthTree43k [12]	43k	162k	-	-	-
CanaTree100 [11]	100	920	-	✓	-
TimberSeg [10]	220	2.5k	-	✓	✓
TimberVision	**2k**	**26.3k**	✓	✓	✓

Table 2. Comparison of TimberVision to existing forestry datasets. Beside the numbers of annotated **Images** and **Trunks**, we rate the inclusion of **C**omponent annotations, **R**eal rather than synthetic images and cut **L**ogs.

individual trunk components are included. To evaluate the generalization of our models, we created new annotations following our policy for a representative selection of TimberSeg images (denoted as *TimberSeg** in Tab. 1). More details regarding annotation compatibility and dataset statistics can be found in the supplementary material. To the best of our knowledge, TimberVision surpasses all current real-image trunk-segmentation datasets in terms of quantity as well as annotation depth by a large margin.

4. Methodology

Based on our dataset, we perform comprehensive ablation experiments for the tasks of OOD and ISEG. To combine their advantages, we demonstrate a generic approach for fusing their output into unified trunk representations and subsequently tracking them.

Learning experiments We build our experiments on the YOLOv8.2 framework [17], which provides an established pipeline for training OOD and ISEG, as well as real-time multi-object trackers. We use a random dataset split of 70/15/15 (train/val/test) applied separately for each subset in Tab. 1. To avoid overlapping areas between splits, we ensure that images of the same recording session are in one split. The *TimberSeg** and *Tracking* images are excluded from training and exclusively used for testing generalization and tracking performance, respectively.[2] For our cross-dataset fine-tuning experiments based on TimberSeg and CanaTree100, we use five-fold validation to compensate for lower image quantities with random 70/15/15 splits and the original provided cross-validation folds, respectively. Experiments are conducted on an NVIDIA Titan RTX using YOLOv8.2 pre-trained models for initialization and adaptive batch sizes. The remaining hyper-parameters as well as data augmentation adhere to the default settings, except for a reduced weighting of distribution focal loss of 0.01, as this was found to improve convergence. The best models are selected after training for 500 epochs.

[2]The test set as well as a part of the *Tracking* set are not publicly available to ensure fair benchmarking of community results.

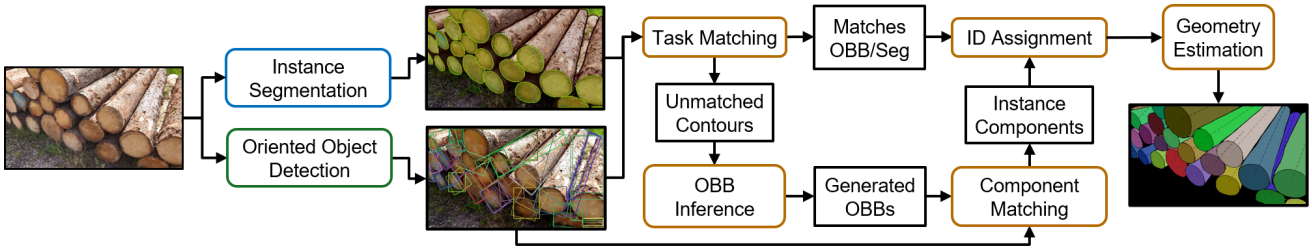

Figure 4. Overview of our task-fusion algorithm deriving unified trunk representations from OOD and ISEG outputs for individual component classes. Processing steps are denoted by rounded and intermediate results by square boxes.

Task fusion and multi-object tracking Our generic approach for fusing OOD and ISEG results of multiple components into robust unified object representations is presented in Fig. 4 and demonstrated for the classes of our dataset. It consists of two matching steps for identifying correspondences between both model outputs in the form of OBBs and segmentation contours, as well as between different components belonging to the same object. Both are conducted using linear sum assignment and differ only by the distance metric to be minimized. For task matching, we generate OBBs enveloping each contour and use their overlap with detected ones. Component matching is based solely on OBBs (either detected or generated) to maximize efficiency and requires dedicated metrics for each class combination. For associating *Cuts* and *Sides*, for instance, we use the largest relative overlap between any of the *Side*'s OBB lines with the *Cut*'s area. Combining all correspondences and eliminating unmatched detections results in unified objects representations consisting of up to one instance of each component represented by an OBB and an optional contour each. For the resulting *Trunks* without both *Bound* and *Cut* components, geometric information is inferred from the remaining classes' relative poses and dimensions, ensuring they are assigned exactly two endpoints.

To preserve object identities over time and increase robustness, we integrate the ByteTrack [50] and Bot-SORT [1] implementations of YOLOv8.2 into our framework. Instead of raw detections, we track new OBBs for unified *Trunks* enveloping all their components, thereby preserving geometric properties and optimizing real-time performance.

5. Experimental results

This chapter describes the evaluation of our ablation and learning experiments, including a comprehensive analysis of relevant impact factors on model performance, as well as fusion and tracking results.

Evaluation protocol and metrics Regarding our learning experiments and task-fusion algorithm, results are reported solely on the test set using the same input resolutions as during training and an empirically derived confidence threshold of 0.4. We report the challenging $mAP^{50\text{-}95}$ metric, introduced by the MS COCO benchmark [27], which averages multiple Intersection-over-Union (*IoU*) thresholds, as opposed to the still widely used mAP^{50} metric [7] yielding higher, but less descriptive results based on a single threshold. If not stated otherwise, *mAP* scores for ISEG are reported for masks only and not for their less relevant axis-aligned bounding boxes. For fusion results, the same metric becomes even more challenging as it is applied on overall OBBs of unified trunk representations, which only fit the ground-truth if all individual components are accurately detected as well as correctly matched to each other. Since fusion *mAP* scores are therefore not directly comparable to individual model results with multiple classes, we provide additional context in the form of precision and recall.

Tracking performance is evaluated on the dedicated *Tracking* subset using the established py-motmetrics [14]. In this case, we explicitly take into account that each trunk can consist of multiple components and rate both segmentation performance and their assignment to trunks by implementing a custom component-wise Intersection-over-Union (IoU_c) score for object masks.

$$IoU_c(G, P) = \frac{\sum_{l \in L} |G_l \cap P_l|}{\sum_{l \in L} |G_l \cup P_l|}, \tag{1}$$

where G_l and P_l are the ground-truth and prediction masks for component l, and L is the set of available components (i.e. a subset of the labels *Side*, *Cut* and *Bound* in this case). To quantify all aspects of tracking based on this matching score, we discuss Multi-Object-Tracking Accuracy (*MOTA*) [2] as well as *IDF1* Score [36]. Furthermore, we report ID Precision (*IDP*), ID Recall (*IDR*) and mean IoU_c over all correct detections ($mIoU_c$).

5.1. Ablation experiments

Class ablation As a preliminary step, we conduct multiple experiments with class combinations ranging from the most fine grained with *Side*, *Cut* and *Bound* as separate classes, to the coarsest combining all of them in a single *Trunk* class. Results for multiple model variants with different input sizes are presented in Fig. 6 and show con-

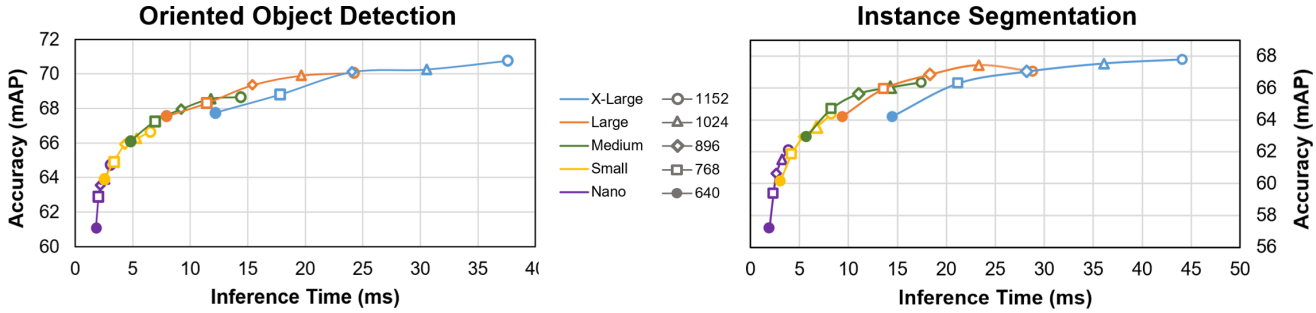

Figure 5. Accuracy as mean class $mAP^{50\text{-}95}$ and average inference time on test set for multiple model capacities and image sizes.

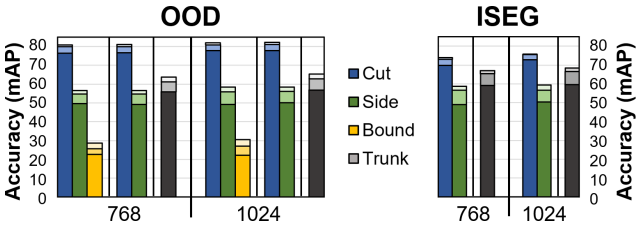

Figure 6. Ablation results as $mAP^{50\text{-}95}$ on test set for three class combinations and two input sizes. For ISEG, *Bound* is excluded due to its inherent overlap with *Side*. The dark-to-light bar colors denote model capacities *Nano*, *Medium* and *X-Large*, respectively.

		Base	Tracking	TimberSeg*
OOD	**Cut**	81.8	80.6	67.4
	Side	58.0	66.0	51.7
ISEG	**Cut**	76.0	73.3	56.6
	Side	58.9	67.3	48.4

Table 3. Model performance as $mAP^{50\text{-}95}$ for the classes *Cut* and *Side*. Results are reported for *Large* models trained and evaluated on the same image resolution of 1024 pixels on our test (*Base*) and *Tracking* sets and selected images from TimberSeg.

sistent trends for both tasks. While *mAP* scores for the *Side* class are slightly lower than those of opaque *Trunk* instances, the values for *Cut* are significantly higher. Since multiple classes also provide more detailed information for post-processing, training two classes is clearly superior to a single-class setup. Another obvious result, however, is the inferior performance of the *Bound* class, which usually covers very small image areas. Following these indications, we rely on a combination of *Cut* and *Side* for subsequent experiments, and infer boundary information geometrically.

Model ablation To find the optimal setup, our ablation study includes five image sizes and all standard model capacities, as visible in Fig. 5. As expected, accuracy and inference time increase with both parameters. For smaller models, accuracy increases stronger than inference time with higher resolutions, while large models show inverse behavior. Based on the analysis, *Large* models with an input size of 1024 pixels provide the best trade-off for our purposes and are used during further evaluation.

5.2. Performance evaluation

Model performance Tab. 3 reports accuracy on multiple test sets described in Sec. 3.4. Given the variability of input data, OOD achieves promising results on both of our own test sets, especially for the *Cut* class, which has the most distinct appearance and shape in typical surroundings. Segmentation masks are a more fine-grained representation with slightly lower overall accuracy, but a less distinct gap between the performance of individual classes. Regarding generalization to the images extracted from TimberSeg, our models achieve similar results to the main test set for some scenarios but perform significantly worse for others resulting in overall lower accuracy. Especially nighttime harvesting scenarios with perspectives and tree types highly differing from our recordings, present a limitation for ISEG to be addressed by the next image-acquisition iteration.

To assess the range of practical operating conditions, we evaluate the impact of scene parameters defined in Sec. 3.2 on model performance, as summarized in Fig. 7. As opposed to previous evaluations conducted exclusively on the test set, we now include the validation set as well to ensure a statistically significant number of samples for each parameter. In general, the more visually distinct *Cut* class remains more robust than *Side* against all scene configurations. Especially in case of high trunk quantities which are typically arranged as log piles, *Cuts* tend to be less occluded and well delineated. Regarding tasks, ISEG is better suited for detecting irregularly shaped logs, while OOD is preferable for large quantities of instances. Furthermore, the difference between classes is less pronounced for ISEG, which may be attributed to the larger *Side* instances producing a higher recall than small *Cuts* for this more challenging task. Overall, the results indicate that the models complement each other and perform reasonably well even in challenging scenarios.

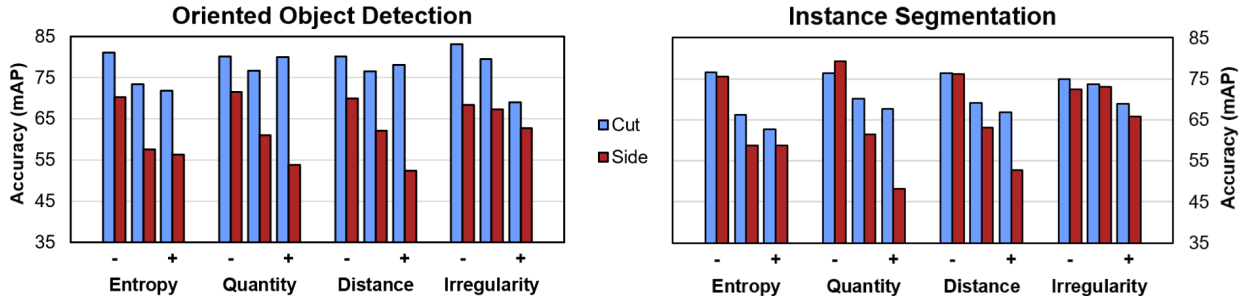

Figure 7. Model performance for different scene-parameter intensities as *mAP* on validation and test splits.

	IDF1	IDP	IDR	MOTA	mIoU_c
ByteTrack	71.1	85.0	61.2	63.8	**85.8**
Bot-SORT	72.5	**86.6**	62.4	63.7	85.4
Optimized	**72.9**	86.0	**63.2**	**65.2**	85.1
Opt | 10 fps	66.5	81.1	56.4	57.8	84.4

Table 4. Mean MOT results for default and optimized configurations on all *Tracking* sequences at a default frame rate of 30 fps.

	10	50	100	300	Best
TS_COCO	2.0	21.6	30.5	41.3	$\mathbf{46.8}_{\pm 3.9}$
$\text{TS}_\text{TimberVision}$	33.2	44.7	46.5	49.4	$\mathbf{52.1}_{\pm 4.5}$
CT_COCO	2.5	37.8	46.8	52.9	$\mathbf{59.9}_{\pm 2.8}$
$\text{CT}_\text{TimberVision}$	31.7	51.3	52.8	56.5	$\mathbf{60.4}_{\pm 2.5}$

Table 5. Evaluation of fine-tuning ISEG models pre-trained with MS COCO and our TimberVision dataset on **T**imber**S**eg [10] and **C**ana**T**ree100 [11] using *Large* models with an input size of 1024. $mAP^{50\text{-}95}$ is reported on the validation set after different numbers of epochs and on the test set for the **Best** resulting models along with standard deviations from five-fold validation for the latter.

Fusion and tracking Overall trunk OBBs generated by our fusion algorithm with the selected models yield an *mAP* of 57.5 on the test set. Given the challenging task of performing both detection and component matching successfully, this score is naturally lower than those for individual models. However, in combination with a high precision and recall of 84.3 and 72.9, respectively, it still indicates that components are rarely matched incorrectly, since a wrong assignment between *Side* and *Cut* instances to the same trunk would drastically distort its overall bounding box. The *mAP* therefore merely suggests that fused trunks might not fit the ground truth as accurately as the results of individual models. This limitation, however, is balanced by the significantly more detailed description of each instance's components and geometric properties.

The evaluation of multi-object tracking in Tab. 4 gives a comprehensive impression of detection, component-matching and association quality for our unified *Trunk* instances. Default configurations already yield encouraging results for both *MOTA* and *IDF1* with Bot-SORT showing only marginally superior performance. Precision is higher than recall, suggesting that both delineation and matching of individual trunk components are consistently accurate, but a relevant number of ground-truth samples remains undetected. This is especially true for recordings containing high numbers of thin live trees in the background. Piles of logs partially occluding each other, on the other hand, can be delineated and associated with relatively high precision. Based on these insights, we optimize the Bot-SORT configuration with a lower *new_track_thresh* of 0.05 and higher *match_thresh* of 0.9, which increases *IDF1* and *MOTA* by 0.4 and 1.5, respectively, while also increasing the number of mostly tracked instances according to the Clear-MOT definition from 50.9% to 52.0%. Even reducing the detection and tracking frame rate to one third does not drastically decrease performance, further illustrating the application range of our entire pipeline.

Cross-dataset experiments To demonstrate the benefits of our dataset and models for other similar domains, we use our final ISEG model as a basis for fine-tuning on both TimberSeg and CanaTree100, which focus on detecting the top layers of log piles and live trees in forest scenes, respectively. Results in comparison to the same approach based on a generic YOLOv8.2 model pre-trained on MS COCO [27] are summarized in Tab. 5. While even overall results on the test set are slightly superior for our model, its main advantage is significantly faster convergence on the new training data. This indicates our models' potential as a basis for fine-tuning on other smaller forestry datasets.

5.3. Discussion and limitations

Fig. 8 shows a selection of representative fusion results. The combination of OOD and ISEG achieves highly robust detections, even in challenging scenarios, as indicated by the scene-parameter-based evaluation. The approach even generalizes to other datasets such as TimberSeg, as visible

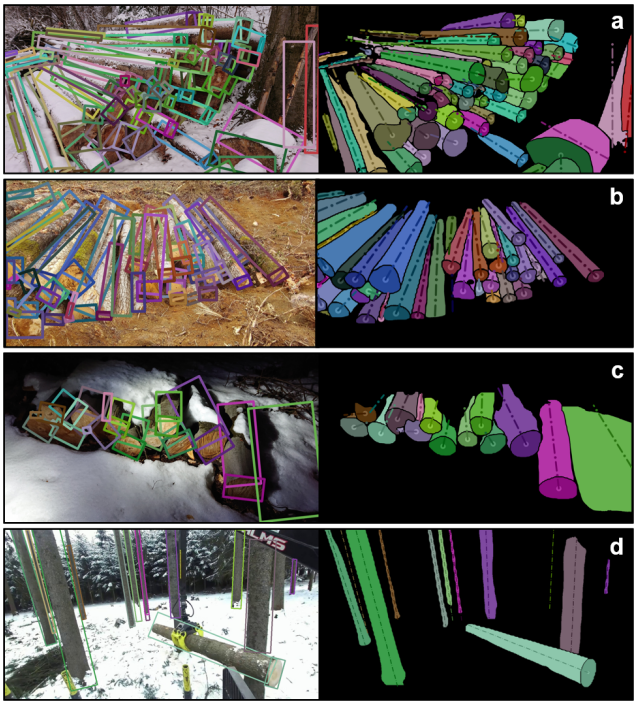

Figure 8. Representative fusion results on test images. The left side shows detected and derived OBBs. The right side shows ISEG results along with center points of cut surfaces and middle axes estimated by our fusion algorithm. Colors identify individual trunk instances, with lighter and darker shades of the same hue corresponding to associated *Side* and *Cut* components, respectively.

in Fig. 8(b). Supporting the quantitative analysis, *Cut* and *Side* components are delineated with high accuracy even if they are covered by snow or debarked, with the visually more distinct *Cut* class usually producing slightly more precise contours. Occlusions are handled reasonably well and only split detections into multiple parts if they affect large areas. Components are correctly assigned to trunk instances in most cases, even for cluttered scenes and large log piles. Furthermore, the results support our thesis that deriving trunk boundaries from inherent geometric properties yields superior performance compared to explicitly detecting them in the form of a *Bound* class, as the middle axes of logs can successfully be inferred for most instances of sufficient visible length. However, there are still a few corner cases to be resolved regarding small or irregularly formed trunks, as visible in the lower right corner of Fig. 8(a). As in this case, wrong component assignment often leads to incorrect estimation of geometric properties, which is the only relevant drawback of this method.

Overall, false detections are rare and mainly occur in challenging scenarios covered by few training samples such as the combination of darkness and snow in Fig. 8(c). Furthermore, the detection of live trees performs well in the foreground of images, while very thin instances in the far background, as in Fig. 8(d), present a more challenging target and can negatively impact quantitative scores due to their inconsistent detection. However, most practical applications, such as harvesting, loading and inventory tasks, require reliable detections mainly in the near and mid-range, which are comprehensively covered by our approach. More detailed quantitative and qualitative results on all subsets can be found in the supplementary material.

While our results based on the well-established YOLOv8.2 framework introduce a solid baseline for OOD and ISEG, it will be interesting to see the performance of other architectures in the future to identify any potential inherent limitations or biases of this approach. Furthermore, while the incorporation of 3D data would be beneficial for certain applications, we intentionally limited our input data to RGB images to evaluate the potential of this affordable and easily accessible modality by itself before combining it with other sensors.

6. Conclusion

Detecting and tracking trunk instances and their components is an essential prerequisite for efficiently handling them during harvesting, loading and measuring tasks. We developed a novel dataset for forestry operations containing annotations for more than 51k log components in real-world images, thereby significantly surpassing all existing similar datasets in terms of quantity and detail. We trained multiple models for the tasks of oriented object detection and instance segmentation and thoroughly evaluated them in different scenarios. Furthermore, we provide a novel fusion and multi-object-tracking framework combining both tasks for real-time perception of trunks and their components as well as geometric properties such as boundaries and middle axes. Our approach detects and delineates trunk components with high accuracy even under challenging conditions, fuses them into unified representations and precisely tracks them across image sequences.

To further improve performance, we plan to integrate additional tracking algorithms and re-identification, as well as to infer segmentation masks directly from oriented instead of axis-aligned bounding boxes. Furthermore, our log representation already includes all relevant information for 3D projection and can therefore be readily combined with depth data to further increase robustness and descriptiveness. By publicly providing the dataset and fusion framework, we aim to extend and develop both in cooperation with the scientific community.

Acknowledgement. We would like to thank our colleagues Christian Zinner, Mario Niedermeyer, Verena Widhalm, Marlene Glawischnig and Vanessa Klugsberger for their valuable contributions during image acquisition.

References

[1] Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. Bot-sort: Robust associations multi-pedestrian tracking. *arXiv preprint arXiv:2206.14651*, 2022. 5

[2] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008. 5

[3] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 2

[4] Steven W Chen, Guilherme V Nardari, Elijah S Lee, Chao Qu, Xu Liu, Roseli Ap Francelin Romero, and Vijay Kumar. Sloam: Semantic lidar odometry and mapping for forest inventory. *IEEE Robotics and Automation Letters*, 5(2):612–619, 2020. 2

[5] Kwanghun Choi, Wontaek Lim, Byungwoo Chang, Jinah Jeong, Inyoo Kim, Chan-Ryul Park, and Dongwook W Ko. An automatic approach for tree species detection and profile estimation of urban street trees using deep learning and google street view images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 190:165–180, 2022. 2

[6] Daniel Queirós da Silva, Filipe Neves Dos Santos, Armando Jorge Sousa, and Vítor Filipe. Visible and thermal image-based trunk detection with deep learning for forestry mobile robotics. *Journal of Imaging*, 7(9):176, 2021. 2

[7] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision (IJCV)*, 88(2):303–338, 2010. 5

[8] Adnan Firoze, Bedrich Benes, and Daniel Aliaga. Urban tree generator: spatio-temporal and generative deep learning for urban tree localization and modeling. *The Visual Computer*, 38(9):3327–3339, 2022. 2

[9] Adnan Firoze, Cameron Wingren, Raymond A Yeh, Bedrich Benes, and Daniel Aliaga. Tree instance segmentation with temporal contour graph. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2193–2202, 2023. 2

[10] Jean-Michel Fortin, Olivier Gamache, Vincent Grondin, François Pomerleau, and Philippe Giguère. Instance segmentation for autonomous log grasping in forestry operations. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 6064–6071. IEEE, 2022. 2, 4, 7

[11] Vincent Grondin, Jean-Michel Fortin, François Pomerleau, and Philippe Giguère. Tree detection and diameter estimation based on deep learning. *Forestry*, 96(2):264–276, 2023. 2, 4, 7

[12] Vincent Grondin, François Pomerleau, and Philippe Giguère. Training deep learning algorithms on synthetic forest images for tree detection. In *Workshop in Innovation in Forestry Robotics: Research and Industry Adoption (ICRA)*, 2022. 2, 4

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2961–2969, 2017. 2

[14] Christoph Heindl and Jack Valmadre. py-motmetrics. *Code repository https://github.com/cheind/py-motmetrics*, 2024. 5

[15] Edo Jelavic, Dominic Jud, Pascal Egli, and Marco Hutter. Towards autonomous robotic precision harvesting: Mapping, localization, planning and control for a legged tree harvester. *arXiv preprint arXiv:2104.10110*, 2021. 1

[16] Peng Jiang, Philip Osteen, Maggie Wigness, and Srikanth Saripalli. Rellis-3d dataset: Data, benchmarks and analysis. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1110–1116. IEEE, 2021. 2

[17] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics YOLO. https://github.com/ultralytics/ultralytics, 2024. Accessed: 2024-07-15. 2, 4

[18] Danilo Samuel Jodas, Sergio Brazolin, Takashi Yojo, Reinaldo Araujo De Lima, Giuliana Del Nero Velasco, Aline Ribeiro Machado, and Joao Paulo Papa. A deep learning-based approach for tree trunk segmentation. In *2021 34th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 370–377. IEEE, 2021. 2

[19] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019. 2

[20] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 3

[21] Sean Krisanski, Mohammad Sadegh Taskhiri, Susana Gonzalez Aracil, David Herries, and Paul Turner. Sensor agnostic semantic segmentation of structurally diverse and complex forest point clouds using deep learning. *Remote Sensing*, 13(8):1413, 2021. 2

[22] Pedro La Hera, Omar Mendoza-Trejo, Ola Lindroos, Håkan Lideskog, Torbjörn Lindbäck, Saira Latif, Songyu Li, and Magnus Karlberg. Exploring the feasibility of autonomous forestry operations: Results from the first experimental unmanned machine. *Journal of Field Robotics*, 41(4):942–965, 2024. 1

[23] Juan Lagos, Urho Lempiö, and Esa Rahtu. Finnwoodlands dataset. In *Scandinavian Conference on Image Analysis*, pages 95–110. Springer, 2023. 2

[24] Qiujie Li and Yu Yan. Street tree segmentation from mobile laser scanning data using deep learning-based image instance segmentation. *Urban Forestry & Urban Greening*, 92:128200, 2024. 2

[25] Yanwei Li, Hengshuang Zhao, Xiaojuan Qi, Liwei Wang, Zeming Li, Jian Sun, and Jiaya Jia. Fully convolutional networks for panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 214–223, 2021. 2

[26] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988, 2017. 2

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014. 2, 5, 7

[28] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 2

[29] Timo Lüddecke and Alexander Ecker. Image segmentation using text and image prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7086–7096, 2022. 3

[30] Stefanie Lumnitz, Tahia Devisscher, Jerome R Mayaud, Valentina Radic, Nicholas C Coops, and Verena C Griess. Mapping trees along urban street networks with deep learning and street-level imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 175:144–157, 2021. 2

[31] Meher V. R. Malladi, Tiziano Guadagnino, Luca Lobefaro, Matias Mattamala, Holger Griess, Janine Schweier, Nived Chebrolu, Maurice Fallon, Jens Behley, and Cyrill Stachniss. Tree instance segmentation and traits estimation for forestry environments exploiting lidar data collected by mobile robots. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024. 2

[32] Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S Ecker, Matthias Bethge, and Wieland Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. *arXiv preprint arXiv:1907.07484*, 2019. 2

[33] U.S. Bureau of Labor Statistics. National census of fatal occupational injuries in 2022. https://www.bls.gov/news.release/pdf/cfoi.pdf, 2023. Accessed: 2024-07-15. 1

[34] Alexander Proudman, Milad Ramezani, Sundara Tejaswi Digumarti, Nived Chebrolu, and Maurice Fallon. Towards real-time forest inventory using handheld lidar. *Robotics and Autonomous Systems*, 157:104240, 2022. 2

[35] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016. 2

[36] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 17–35. Springer, 2016. 5

[37] Juergen Rossmann, Michael Schluse, Christian Schlette, Arno Buecken, Petra Krahwinkler, and Markus Emde. Realization of a highly accurate mobile robot system for multi purpose precision forestry applications. In *2009 International Conference on Advanced Robotics*, pages 1–6. IEEE, 2009. 1

[38] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7374–7383, 2019. 2

[39] Scalabel open-source web annotation tool. https://github.com/scalabel/scalabel. Accessed: 2024-07-15. 3

[40] Daniel Steininger, Andreas Kriegler, Wolfgang Pointner, Verena Widhalm, Julia Simon, and Oliver Zendel. Towards scene understanding for autonomous operations on airport aprons. In *Proceedings of the Asian Conference on Computer Vision*, pages 147–163, 2022. 2

[41] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7262–7272, 2021. 2

[42] Jiamin Wang, Xinxin Chen, Lin Cao, Feng An, Bangqian Chen, Lianfeng Xue, and Ting Yun. Individual rubber tree segmentation based on ground-based lidar data and faster r-cnn of deep learning. *Forests*, 10(9):793, 2019. 2

[43] Maggie Wigness, Sungmin Eum, John G Rogers, David Han, and Heesung Kwon. A rugd dataset for autonomous navigation and visual perception in unstructured outdoor environments. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5000–5007. IEEE, 2019. 2

[44] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. Accessed: 2024-07-15. 2

[45] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3974–3983, 2018. 2

[46] Qian Xie, Dawei Li, Zhenghao Yu, Jun Zhou, and Jun Wang. Detecting trees in street images via deep learning with attention module. *IEEE Transactions on Instrumentation and Measurement*, 69(8):5395–5406, 2019. 2

[47] Xingxing Xie, Gong Cheng, Jiabao Wang, Xiwen Yao, and Junwei Han. Oriented r-cnn for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3520–3529, 2021. 2

[48] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 472–480, 2017. 2

[49] Oliver Zendel, Katrin Honauer, Markus Murschitz, Daniel Steininger, and Gustavo Fernandez Dominguez. Wilddash-creating hazard-aware benchmarks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 402–416, 2018. 2

[50] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 1–21. Springer, 2022. 5

[51] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 2