Detection of Knots in Oak Wood Planks: Instance versus Semantic segmentation

1st Evalds Urtans
Riga Technical University
Riga, Latvia
evalds.urtans@rtu.lv

2nd Karlis Bumanis
SIA Meza un koksnes produktu petniecibas un attistibas instituts
Jelgava, Latvia
karlis.bumanis@e-koks.lv

3rd Valters Vecins
Riga Technical University
Riga, Latvia
vecins.valters@edu.rtu.lv

4th Maris Ancans
Riga Technical University
Riga, Latvia
maris.ancans@edu.rtu.lv

5th Aiga Andrijanova
Riga Technical University
Riga, Latvia
aiga.andrijanova@edu.rtu.lv

6th Marcis Teodors Upenieks
Riga Technical University
Riga, Latvia
marcis.teodors.upenieks@edu.rtu.lv

7th Kristofers Volkovs
Ventspils University
Ventspils, Latvia
kristofers.volkovs@venta.lv

***Abstract*—In this study, we present a new dataset of knot-covered oak planks. It contains 1500 images that have 1 to 11 knots per image, along with mask and bounding-box annotations. The data set was evaluated using deep machine learning methods, and it has been found that instance segmentation models are superior in this task, achieving 59% Box-IoU versus 49% Box-IoU using semantic segmentation. Instance segmentation performed better to detect knots by segmenting instances with an accuracy of 90%, while semantic segmentation detected konts with an accuracy of 89%.**

***Keywords*—wood surface defects, wood industry, wood processing, wood quality control process, wood defects dataset, instance segmentation, semantic segmentation, deep learning, dataset**

I. Introduction

In recent years, there has been an increase in research interest in the domain of instance segmentation and semantic segmentation models. Although these methods are often used interchangeably because they often accomplish the same goal, there has rarely been a direct comparison between both methods on the same data set under controlled experimental conditions. This research compares these fundamentally different methods using domain-specific dataset. It also introduces a novel data set called 'FSCC oak knots', which consists of high-quality photos of wooden oak boards that contain knots. Wood knots are a term known in the woodworking industry that describes defects or blemishes in the wood and the knots are caused by the natural growth of the tree. In the woodworking industry, every step of the manufacturing process affects the use and cost of the material. Defects, such as knots, reduce its aesthetic value and the heterogeneity of the material [1]. It plays an important role in the mechanical properties of wood, such as strength. Traditionally, these defects have been detected by manual labor of skilled but often biased workers, and recent studies have shown that manual inspections achieve low precision [2]. In addition to the data set, this research has an open-source implementation [1] of data processing scripts and training scripts written in PyTorch.

[1]https://github.com/evaldsurtans/FSCC-Oak-Knots-Dataset

II. Related work

Publicly available wood defect datasets are very limited; currently there are only a handful of such datasets, for example, LSIDWS [3], WOOD [4], Wood Spieces [5] and Kaggle Wood Textures [2]. Most of the data sets used in the wood defects research domain are either private or require contacting a particular individual to possibly acquire said data sets. Some data sets are available for the classification of defects in other industrial materials, including kaggle steel defects [3], magnetic tile defects [6], CrackForest data set [7], TILDA textile texture database [4], etc. Furthermore, there are different technology classification datasets, such as DTD [8] or Kylberg texture dataset [9], which do not have the labels needed to perform semantic or instance segmentation. This indicates a certain interest in the topic of segmentation and classification of various material defects. However, this field still severely lacks publicly available qualitative datasets. For models, the latest research uses smaller datasets of wood defects that were made for the purpose of the particular research. Often, for instance segmentation models such as SSD [10], Faster R-CNN [2], Mask R-CNN [11] and YOLO [12], [13] are used. Classification models that do not provide defect localization have also been studied, such as FCN based methods [14], [15] or novel PCA feature fussion methods [16].

A. Dataset

The data set contains 1500 high-quality oak plank photos with controlled lightening that have been photographed from the same distance. The images are colored RGB photos with a resolution of 500x500 pixels. The data set has been divided into 1200 samples for training and 300 samples for testing. Data have been collected using professional wooden plank scanning equipment and metdology that has been described in previous research [11]. All samples contain at least one knot,

[2]https://www.kaggle.com/datasets/edhenrivi/wood-samples

[3]https://www.kaggle.com/competitions/severstal-steel-defect-detection/overview

[4]http:// lmb.informatik.uni-freiburg.de/resources/datasets/tilda.en.html

and many samples contain more knots per image, as shown in Fig. 1. Most samples contain one knot per image, which is also the median count of knots per image, and the maximum number of knots per image is 11.

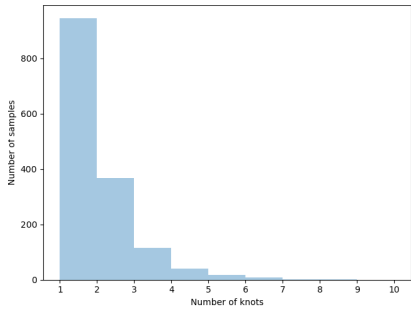

Fig. 1: Number of knots per image

Most samples contain pixels labeled knots between 0.1-40% of the image, but there are also some images that contain larger knots, as shown in Fig. 2. The median area size of knots per sample is 1.54% of the image. Visual examples of knots with the smallest area are shown in Fig. 3. Similarly, examples of knots with the largest area are shown in Fig. 4.

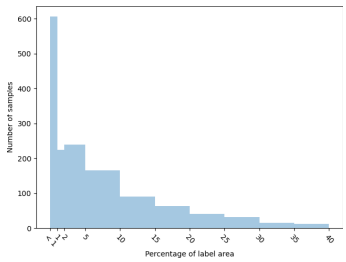

Fig. 2: Percentage of pixels labelled as knot per image

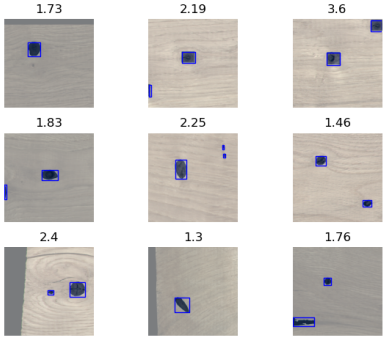

Fig. 3: Random selection of samples with smallest area size of knots below 5% of whole image. Number above image denotes the area in percentage.

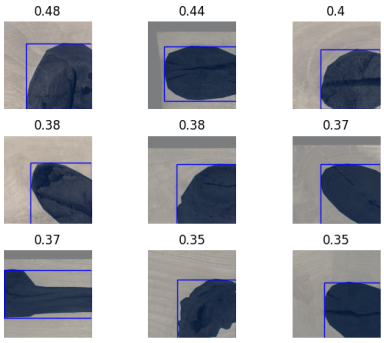

Fig. 4: Random selection of samples with largest area size of knots above 25% of whole image. Number above image denotes the area in percentage.

Examples of images with the highest number of knots are shown in Fig. 5. As can be seen in the examples, bias of laboratoryler is present, as in some samples labeler might choose to label multiple small knots as one larger knot or mark each of them separately. Finally, the data set is publicly available for use in research under CCL [5].

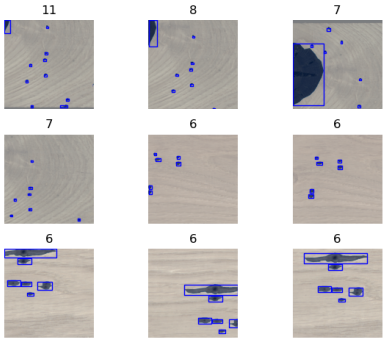

Fig. 5: Samples with highest number of knots

III. Methodology

For evaluation, two metrics have been chosen, IoU (intersection over union) for pixel area and Box-IoU for bounding boxes. The first metric IoU Equation (1) for pixel area have been used for semantic segmentation models because this group of models produces per-pixel masks as the output, but Box-IoU has been used for both semantic and instance segmentation models. In equation Equation (1) TP denotes True Positive, FP denotes False Positive, FN denotes False Negative, y denotes the ground truth label of a pixel, $\hat{y}$ denotes the predicted label of a pixel, and ϵ denotes the smoothing constant for numerical stability.

$$IoU = \frac{1}{N} \sum \frac{TP + \epsilon}{TP + FP + FN + \epsilon} = \quad (1)$$

$$\frac{1}{N} \sum \frac{y \odot \hat{y} + \epsilon}{y \odot \hat{y} + (1 - y) \odot \hat{y} + y \odot (1 - \hat{y}) + \epsilon} \quad (2)$$

[5]http://share.yellowrobot.xyz/1651675667-llu

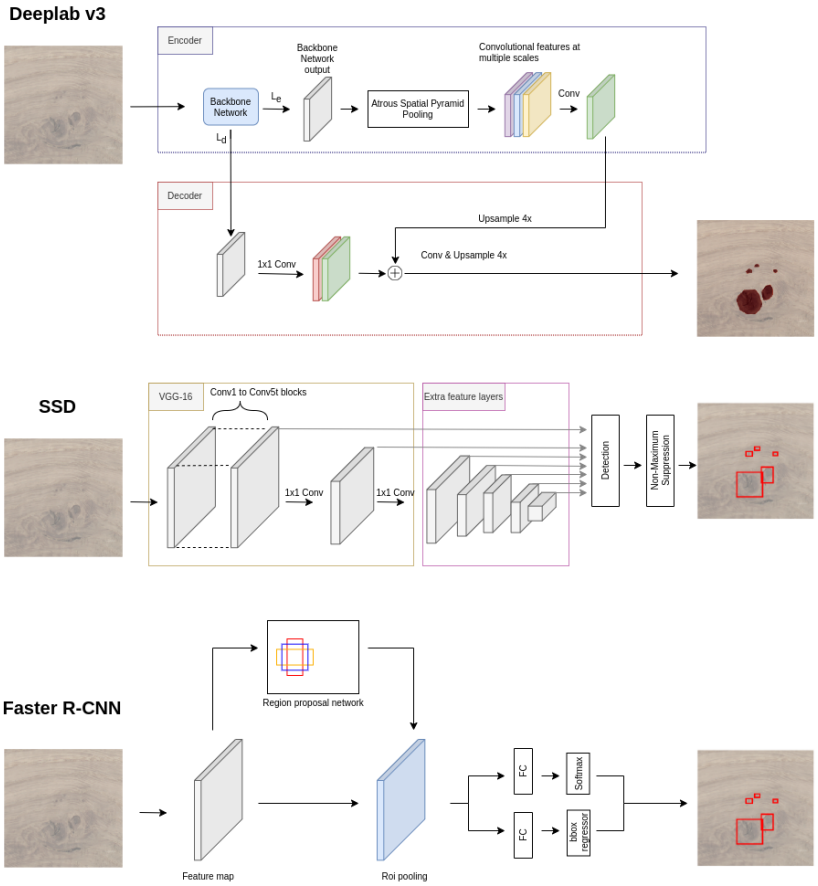

Fig. 6: Different types of archtectures used in training

Models have also been evaluated using the accuracy metric, which is calculated as the classification metric to measure how many predictions overlap with the actual positions of the knots; any bounding box that overlaps is classified correctly. This mimics the human labrourer who usually identifies only the location of the knot and not the exact shape of a knot which is subject to interpretation. Only the best metrics found during the entire training run have been used, and only those runs with loss function convergence have been used. The box-IoU measures the intersection area between two bounding boxes with a score of 0 to 1, with 1 being the perfect bounding box intersection and 0 without intersection Fig. 7. Equations Equation (3) show steps to calculate the Box-IoU metric. Since the intersection area is nothing more than a rectangle, it is calculated using the x and y coordinates of the bounding boxes that make up this area. In (x_2, y_2) and (x_3, y_3) coordinates belong to Box_1 and Box_2 accordingly. Then the total area covered by both bounding boxes is calculated using their respective coordinates. Finally, the intersected area is divided by the total area outside the overlapping region from both bounding boxes to produce a Box-IoU score.

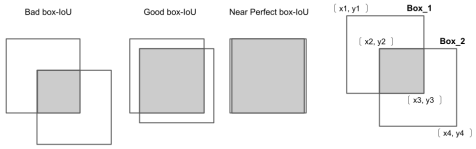

Fig. 7: Bounding box IoU working principle.

Experiments have been repeated iteratively multiple times by running subsets of grid search of hyper-parameters to find the best performing model. All experiments used the maximum number of samples per batch that could fit in GPU and different learing rates, class weights, and model architectures.

$$Intersec = (x_3 - x_2) * (y_3 - y_2) \tag{3}$$

$$Box1\_Area = (x_3 - x_1) * (y_3 - y_1) \tag{4}$$

$$Box1\_Area = (x_3 - x_1) * (y_3 - y_1) \tag{5}$$

$$Box2\_Area = (x_4 - x_2) * (y_4 - y_2) \tag{6}$$

$$\text{Box IoU} = \frac{Intersec}{Box1\_Area + Box2\_Area - Intersec} \tag{7}$$

The study included the following architectures that use semantic segmentation: FCN ResNet, [17], DeepLab v3 ResNet [18], DeepLab v3 MobileNet [19], and Lite R-ASPP MobileNet [20]. The study included the following architectures that use instance segmentation: Faster R-CNN [21] with different backbones, FCOS [22], RetinaNet [23] and SSD [24]. The main architectures used are shown in Fig. 6. Semantic segmentation models were trained using weighted BCE. While instance segmentation models used composite losses consisting of objectiveness loss, box regression loss, region-proposal network loss, and classifier loss. For semantic segmentation, the bounding boxes were calculated using the contour detection algorithm in the predicted class maps [25]. For instance segmentation models confidence threshold of 0.5 for predicted bounding boxes was used. If two or more boxes overlapped with mote than 0.1 Box-IoU then only bounding boxes with the highest confidence score were left in the results. This was done to avoid artificially improving the results for multiple bounding boxes at the same predicted position. All experiments used a set of random image augmentations with probability opf 90% in the training data set. These included horizontal flip, vertical flip, scale, rotation, and color jitter.

TABLE I: Comparison of performance between instance and semantic segmentation methods using the FSCC oak knot dataset

Model	**Method**	**Pixel IoU**	**Box IoU**	**Acc.**
Faster R-CNN	Instance segmentation	N/A	**0.59**	**0.90**
FCOS	Instance segmentation	N/A	0.51	0.77
SSD VGG-16	Instance segmentation	N/A	0.27	0.67
RetinaNet	Instance segmentation	N/A	0.49	0.72
DeepLab V3 ResNet-101	Semantic segmentation	**0.51**	**0.49**	**0.89**
DeepLab V3 ResNet-50	Semantic segmentation	0.45	0.40	0.86
DeepLab V3 MobileNet	Semantic segmentation	0.40	0.37	0.72
Lite R-ASPP MobileNet	Semantic segmentation	0.42	0.37	0.73
FCN ResNet-50	Semantic segmentation	0.42	0.43	0.81

IV. Experiments

The results show a considerable difference between instance segmentation and semantic segmentation models, as seen in Table I. Instance segmentation loss functions and model architectures are significantly more complex and outperform much simpler semantic segmentation models using the Box IoU metric. Instance segmentation models with some exceptions like Mask R-CNN [26] output only a bounding box without mask of the detected object. This is the reason why instance segmentation models do not contain Pixel IoU values.

In Table I it is seen that Faster R-CNN model achieved both the highest accuracy and the Box-IoU metric of all models tested in this data set. The lowest was obtained for both accuracy and Box IuU metrics with the SSD VGG-16 model. From the semantic segmentation models, the Peak Pixel IoU metric was achieved with DeepLab V3 ResNet-101 and the lowest value for this metric was obtained by Lite R-ASPP MobileNet and FCN ResNet-50.

Instance segmentation and semantic segmention models reached the convergence of loss function well below 100 epochs, but instance segmentation models achieved a much more stable improvement in Box-IoU and accuracy metrics. Also, as seen in Fig. 8 and Fig. 9 convergence of loss function, for instance segmentation was much faster than for semantic segmentation. Sensitivity to the details of textures might, and interpretation of human labelers might have a significant effect on results between both methods. As seen in the examples below, semantic segmentation predictions try to estimate the shape of a knot, which often has not been very precisely drawn by labelers. Thus, a more practical metric might be the detection of overlapped bounding boxes and the higher accuracy of the detection of a knot at a specific location, rather than the precise shape of it. On the left Fig. 8 are shown the loss function and the metric plots. On the right mask, the ground truth and the predicted mask are seen. On the left Fig. 9 are shown both the loss function and the Box-IoU plots during training for the test and training sets. On the right there are samples with both true and predicted bounding boxes, and labeled masks are seen.

V. Further research

This study did not include all semantic segmentation models such as UNet ++ [27] or UNet 3+ [28] which could improve the results for this group of models. Models that combine instance and semantic segmentation should also be examined [26]. Furthermore, this study did not include all losses that could have a significant effect on the results functions used in this domain, such as focal loss [23], DICE loss [29], Tversky loss [30], etc. The results could also be improved by studying transfer learning techniques from a pre-training model using other texture or wood defect datasets. Furthermore, more complex data enhancement could be explored to further improve the results.

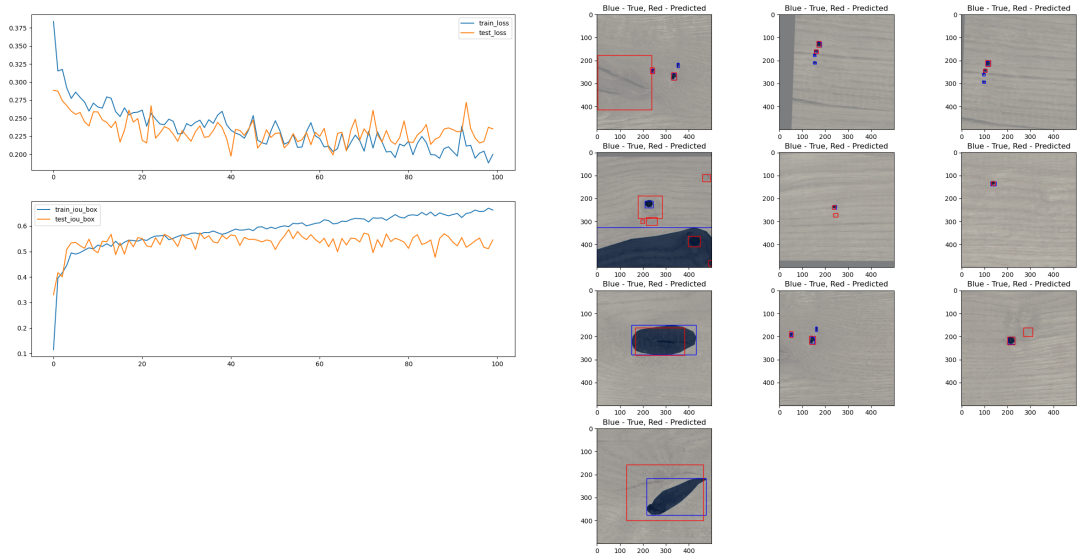

Fig. 8: Example of training process and results using instance segmentation model

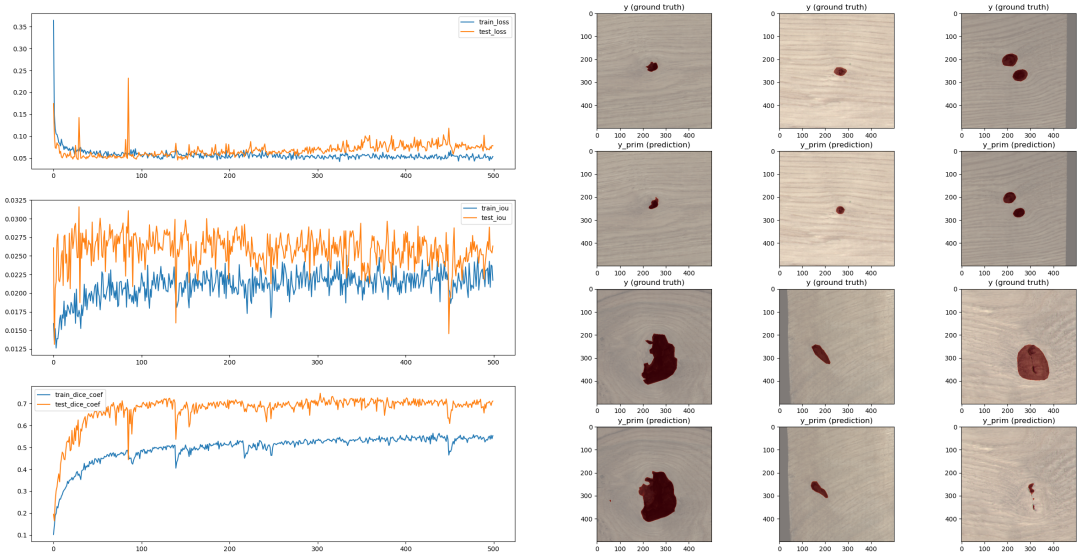

Fig. 9: Example of training process and results using semantic segmentation model

VI. Conclusions

In this article, we compared several instance and semantic segmentation models and found that instance segmentation models performed better on this task. The results demonstrated the applicability of both methods to the task of locating and segmenting wood defects. The limitation of this method is the need to manually label images to train models, which makes it difficult to error and bias the labeler. Another limitation is that the Box-IoU for semantic segmentation was calculated using bounding boxes derived from the contour detection

algorithm, which reduces the precision of the output. Instance segmentation models achieved 59 % Box-IoU versus 49 % Box-IoU semantic segmentation, which is surprising because the semantic segmentation models have simpler architecture and loss functions, which should produce more robust results in all experiments, but could not achieve better results in the same controlled setting. Instance segmentation was also able to detect knots with an accuracy of 90%, while semantic segmentation detected konts with an accuracy of 89 %. It can be concluded that instance segmentation is better at detecting boundries of a shape of a knot. The results achieved in this work establish a benchmark for follow-up studies and are probably not the highest achievable accuracy.

ACKNOWLEDGEMENTS

According to contract No. 1.2.1.1/18/A/004 between 'Forest Sector Competence Center' Ltd. and the Central Finance and Contracting Agency, concluded on 17 April 2019, the study is carried out by 'Meža un koksnes produktu pētniecības un attīstības institūts' Ltd. with support from the European Regional Development Fund (ERDF) within the framework of the project 'Forest Sector Competence Center'. Research has been completed with support from the High Performance Computing Center of Riga Technical University, which provided 12 nVidia K40 GPUs and 8 nVidia V100 GPUs.

REFERENCES

[1] O. Broman , M. Fredriksson, "Wood material features and technical defects that affect yield in a finger joint production process," *Wood Material Science & Engineering*, vol. 7, pp. 167 – 175, 2012.

[2] A. Urbonas, V. Raudonis, R. Maskeliūnas, , R. Damaeviius, "Automated identification of wood veneer surface defects using faster region-based convolutional neural network with data augmentation and transfer learning," *Applied Sciences*, vol. 9, p. 4898, 2019.

[3] P. Kodytek, A. Bodzas, , P. Bilik, "A large-scale image dataset of wood surface defects for automated vision-based quality control processes," *F1000Research*, 2021.

[4] O. Silvén, M. Niskanen, , H. Kauppinen, "Wood inspection with non-supervised clustering," *Machine Vision and Applications*, vol. 13, pp. 275–285, 2003.

[5] P. Barmpoutis, K. Dimitropoulos, , I. Barboutis, "Wood species recognition through multidimensional texture analysis," *Comput. Electron. Agric.*, vol. 144, pp. 241–248, 2018.

[6] Y. Huang, C. Qiu, , K. Yuan, "Surface defect saliency of magnetic tile," *The Visual Computer*, vol. 36, pp. 85–96, 2018.

[7] L. Cui, Z. Qi, Z. Chen, F. Meng, , Y. Shi, "Pavement distress detection using random decision forests," in *International Conference on Data Science*. Springer, 2015, pp. 95–102.

[8] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , , A. Vedaldi, "Describing textures in the wild," in *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[9] G. Kylberg, "The kylberg texture dataset v. 1.0," Centre for Image Analysis, Swedish University of Agricultural Sciences and Uppsala University, Uppsala, Sweden, External report (Blue series) 35, September 2011. http://www.cb.uu.se/~gustaf/texture/

[10] F. Ding, Z. Zhuang, Y. Liu, D. Jiang, X. Yan, , Z. Wang, "Detecting defects on solid wood panels based on an improved ssd algorithm," *Sensors (Basel, Switzerland)*, vol. 20, 2020.

[11] G. K. P. R. G. Romanovskis, K. Bumanis, "Location and identification of oak wood defect by deep learning," vol. 20, 2021, pp. 1837–1842.

[12] B. Wang, C. Yang, Y. Ding, , G. Qin, "Detection of wood surface defects based on improved yolov3 algorithm," *BioResources*, 2021.

[13] D. J. V. Lopes, G. dos Santos Bobadilha, , K. M. Grebner, "A fast and robust artificial intelligence technique for wood knot detection," *Bioresources*, vol. 15, pp. 9351–9361, 2020.

[14] M. Gao, D. Qi, H. Mu, , J. Chen, "A transfer residual neural network based on resnet-34 for detection of wood knot defects," *Forests*, 2021.

[15] M. Gao, P. Song, F. Wang, J. Liu, A. Mandelis, , D. Qi, "A novel deep convolutional neural network based on resnet-18 and transfer learning for detection of wood knot defects," *J. Sensors*, vol. 2021, pp. 4 428 964:1–4 428 964:16, 2021.

[16] Y. Zhang, C. Xu, C. Li, H. Yu, , J. Cao, "Wood defect detection method with pca feature fusion and compressed sensing," *Journal of Forestry Research*, vol. 26, pp. 745–751, 2015.

[17] K. He, X. Zhang, S. Ren, , J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[18] L.-C. Chen, G. Papandreou, F. Schroff, , H. Adam, "Rethinking atrous convolution for semantic image segmentation," *ArXiv*, vol. abs/1706.05587, 2017.

[19] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, , H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *ArXiv*, vol. abs/1704.04861, 2017.

[20] A. G. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, , H. Adam, "Searching for mobilenetv3," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1314–1324, 2019.

[21] S. Ren, K. He, R. B. Girshick, , J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 1137–1149, 2015.

[22] Z. Tian, C. Shen, H. Chen, , T. He, "Fcos: Fully convolutional one-stage object detection," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9626–9635, 2019.

[23] T.-Y. Lin, P. Goyal, R. B. Girshick, K. He, , P. Dollár, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, pp. 318–327, 2020.

[24] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C.-Y. Fu, , A. C. Berg, "Ssd: Single shot multibox detector," in *ECCV*, 2016.

[25] S. Suzuki , K. Abe, "Topological structural analysis of digitized binary images by border following," *Comput. Vis. Graph. Image Process.*, vol. 30, pp. 32–46, 1985.

[26] K. He, G. Gkioxari, P. Dollár, , R. B. Girshick, "Mask r-cnn," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, pp. 386–397, 2020.

[27] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, , J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support : 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, held in conjunction with MICCAI 2018, Granada, Spain, S...*, vol. 11045, pp. 3–11, 2018.

[28] H. Huang, L. Lin, R. Tong, H. Hu, Q. Zhang, Y. Iwamoto, X. Han, Y.-W. Chen, , J. Wu, "Unet 3+: A full-scale connected unet for medical image segmentation," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1055–1059, 2020.

[29] C. H. Sudre, W. Li, T. K. M. Vercauteren, S. Ourselin, , M. J. Cardoso, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," *Deep learning in medical image analysis and multimodal learning for clinical decision support : Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017, held in conjunction with MICCAI 2017 Quebec City, QC,...*, vol. 2017, pp. 240–248, 2017.

[30] S. S. M. Salehi, D. Erdoğmuş, , A. Gholipour, "Tversky loss function for image segmentation using 3d fully convolutional deep networks," *ArXiv*, vol. abs/1706.05721, 2017.

[31] T. He, Y. Liu, C. Xu, X. Zhou, Z. Hu, , J. Fan, "A fully convolutional neural network for wood defect location and identification," *IEEE Access*, vol. 7, pp. 123 453–123 462, 2019.