LOCATION AND IDENTIFICATION OF OAK WOOD DEFECT BY DEEP LEARNING

Gints Romanovskis[1], Karlis Bumanis[2], Gundars Kulikovskis[3], Peteris Rivza[1]
[1]Latvia University of Life Sciences and Technologies, Latvia;
[2]Forest and Wood Products Research and Development Institute, Latvia;
[3]SIA Zippy Vision, Latvia
gintsromanovskis@gmail.com, karlis.bumanis@e-koks.lv,
gundars.kulikovskis@zippyvision.com, peteris.rivza@llu.lv

Abstract. As the value of oak wood increases, it is necessary to think of solutions that help make use as much material as possible during the industrial processing of wood. In order to study the possibilities of efficient use of wood, the specific oak wood features and defects were compiled from images obtained on a scanner prototype. The scanner, based on two TeleDyne Linea line cameras, captured high-quality images at 17,000 by 900 resolution. When the images were marked, two different mask r-cnn models were trained, using the instance segmentation method. The original model (first) contained only the data obtained from the scanner, while the training of the second model used the original images and additional images that were artificially adjusted to expand the data set. New images where generated using keras framework. There were only two data augmentation methods used like "brighten change" and "gaussian noise". These methods where chosen, because they did not change the object physical location on image. All new images where generated based on labelled images, that way the new images did not need to be relabelled. The original image json file was attached to artificially generated images. By using this method, a lot of marking process time saved. The original model showed an average of 73%, while the second model, which used data augmentation, improves the accuracy of the guess by an average of 16%, reaching an average of ~ 89%. The resulting network model was successful in identifying and localizing problems specific to oak wood.

Keywords: oak wood defect recognition, image segmentation, deep learning.

Introduction

Defects in material (oak wood boards) processed in industrial processes are largely determined by the use of human resources, which makes it a very expensive and time-consuming process. In order to make significant improvements, it is intended to involve machine learning [1] techniques more efficiently to improve these processes. Technology implementers will also be able to save at the expense of material consumption by using higher value-added products.

For improvements, the timely identification of defects involves the use of image classification [2] techniques for wood defect recognition, based on convoluted neurons networks [3]. Image classification is based on a process, during which spectral classes are transformed into an information class. Information classes are categories of interest which, in our case, is a sound knot, dead knot and shakes. Typically, the class is assigned to entire image. Using image classification solutions, we get the entire image class, yet to resolve the problem and find all the defects, more broadly approach has to be used, like instance segmentation, to get all object prediction [4]. The advantage of image segmentation in the identification of objects is that it is possible to identify all interest objects in the image and it is possible to identify the location of the objects and their physical properties like size.

Segmentation is part of computer vision in artificial intelligence. It aims not only to identify the class of objects in the picture, but also to give a little more insight into the physical characteristics of the object, such as, texture, colour, and physical size of the object. To achieve this result, the image segmentation process classifies the pixel of each image and then groups along the relevant segments of the object classes. Each of these pixels has a common characteristic, such as colour or intensity. The nearby objects differ drastically in relation to these characteristics.

The techniques used for image segmentation allow resolving a number of major problems that occurred due to not having detailed information about the objects. Other image classification methods could only provide information about a complete image set.

Segmentation techniques are widely used in medicine, where, due to the benefits of segmentation, human cells are studied, and it is possible to identify harmful cells or cancer [5]. Another industry where segmentation techniques are beneficial, is mechanical engineering [6].

DOI: 10.22616/ERDev.2021.20.TF402

Materials and methods

An industrial prototype of a scanner was created to produce images for the study and analysis. Prototype can be viewed in Fig. 1. High-resolution images of wood boards were taken. The prototype was powered by a belt conveyor with attachments. The physical dimensions of the belt are 0.3 by 3.2 m, providing a feed speed: 30 to 45 m min^{-1}. Two "TeleDyne Linea" [7] cameras provided image capture. The picture of the top side of the board was captured by the "TeleDyne Linea" bilinear line camera with a resolution of 4 k (resolution 10.72 px mm^{-1}), while the bottom of the image was captured with "TeleDyne Linea" colour camera with a resolution of 2 k (resolution 5.68 px mm^{-1}).

To have a consistent light in all of the images, two additional lights were attached to the prototype. The lights provided brightness of 16mil lux and white colour light – 6000K. 3D triangulation provided at image acquisition.

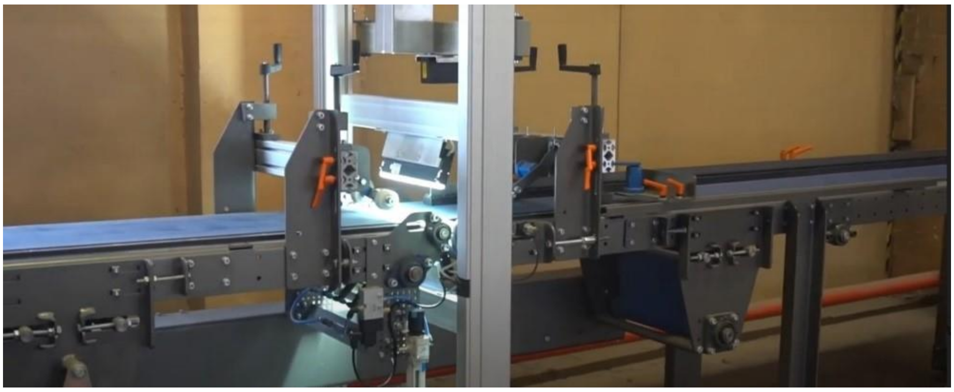

Fig. 1. **Scanner prototype**

Acquisition of images and objects of interest

A high-resolution prototype captured images of oak wood boards. The physical parameters of the scanned boards are as follows:

- width: 100 to 240 mm
- thickness: 3 to 40 mm
- length: 250 to 3000 mm.

Together, about 175 oak wood board images were captured, with a resolution of 17000 by 900. Considering that the image resolution is very high, each image was divided into smaller images in size 900 by 900, so the total number of images after splitting was approximately 2,000.

The following values were obtained after analysis of images that included sorting images. Only images of oak wood were selected where each of the wood's specific features or defects could be visually observed. The list of observed features and defects can be seen more accurately in the table (Table 1). Based on the data in this table, a training data set was created for the chosen model Mask R-CNN [8]. The class, with the highest number of units, was the small black pins, in which also all the small visual defects that resembled small dots (but they were not the marks of insects) were counted in.

Table 1

Identified oak wood features and defects

Feature/defect type	Number
Bark	32
Open knot	12
Dead knot	41
Small black pins	216
Not milled surface	44
Shake	193
Sound knot	53
Sapwood	31

Expanding a dataset

It was taken in account that the data set must become more populated, and the chosen model cannot be subjected to data memorization, which can be described as a problem in the dataset. The methods of data augmentation were considered [9]. In view of the fact that wood board has a distinct structure and, for example, by applying augmentation techniques such as: rotating the image 90°, 180° and 270°, trimming the image along the vertical and horizontal axis could influence overall accuracy. Due to the impact on accuracy, these methods were not applied. The most basic methods of increasing data, included image dividing in several areas of the image, applying grain effect, and changing the background brightness. When the base image is divided into several areas, it ensures that the objects of interest in the image are often divided into several images. Such a method could be compared to randomly trimming the image, thus changing the visual appearance of the object of interest. For example, see Fig. 2. As these images appear to have common characteristics, these three parts were combined in one image before the data were increased.

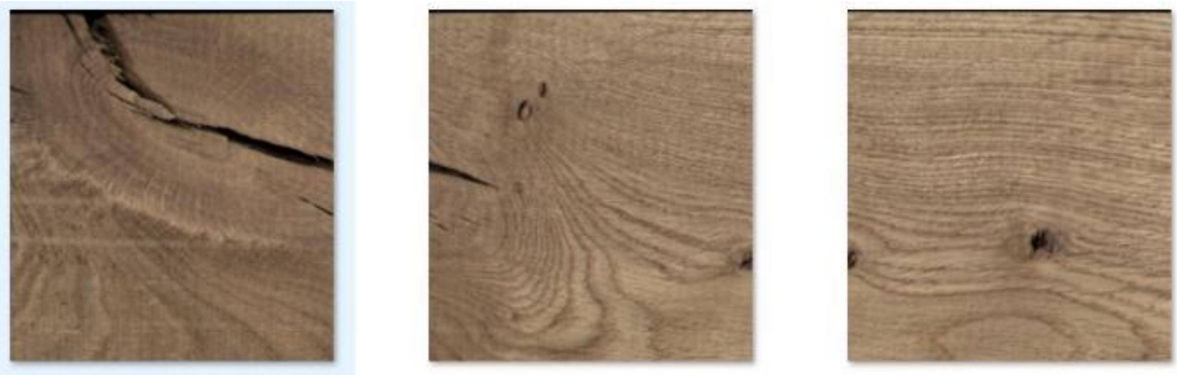

Fig. 2. **Wood image is divided by 900x900 px**

Data marking

Previously prepared images have to be processed with special software so that objects of interest are identified on the image and annotations are applied to these objects. You must have “LABELME” installed on your computer to start making annotations. This software can be found in GIT [10] and is available to the public for free. The software allows to assign a class that describes the object to each region of interest that is highlighted with polygon or image. After processing the image, you need to save a new file that describes an object that has only been identified during the image labelling process. Considering that data augmentation methods were applied to the training image data set, the newly created files for which image transformation functions have been applied should be associated with the json (JavaScript Object Notation) format file created by the base image “LABELME”, which stores the coordinates of the points on the objects marked in the image. If augmentation methods that change the location of region of interested in some way are used on the data set, these images must then be re-labelled, because the objects of interest have changed the coordinates of the points.

Architecture of Mask R-CNN

Mask R-CNN [8]is a framework for object segmentation that can identify not only the location of each object, but it is also able to detect the mask of the object of interest. The main direction in this method is the instance segmentation. Segmentation can be described as image classification at pixel level, which means that a pixel of each object of interest is assigned to a class. [11] Mask R-CNN is an expanded Faster CNN [12] model that has an additional expansion branch attached, specifically to define masks for objects of interest, in addition to the two branches already in place, where one of the branches classifies the object, but the other one determines the highlighting box for the object of interest. [5] This framework works at two tiers: the first tier scans the image, and searches for regions that may contain any object of interest, by scanning and using the RPN [12] algorithm. The second tier, a class and bounding box is predicted in parallel, and Mask R-CNN generates a binary mask for each object of interest [8].

Training

For training purposes, we used the Detectron 2 library [13], which offers previously practiced network weights. For the training process we had to register a new set of training data that will be used during the training process. To improve the process, the network model was trained based on ResNet50 Backbone [14]. This is a pretrained model that has been trained on COCO Train 2017 data and is available for use through the Detectron 2 Library. Before starting the training process, you need to set up Mask R-CNN configuration. All the used configuration parameters are shown in Table 2.

Table 2

Mask R-CNN configuration

Parameter	**Value of parameter**
DATALOADER.NUM_WORKERS	2
MODEL.WEIGHTS	COCO-InstanceSegmentation/mask_rcnn_R_50_FPN_3x.yaml
SOLVER.IMS_PER_BATCH	2
SOLVER.BASE_LR	0.00025
SOLVER.MAX_ITER	1300
MODEL.ROI_HEADS.NUM_CLASSES	9

The training process lasted 1300 iterations, during which an error was actively calculated for each interest object by a formula (1):

$$L = L_{cls} + L_{box} + L_{mask}, \qquad (1)$$

where L_{cls} – classification loss;
L_{box} – bounding box loss;
L_{mask} – mask loss.

In the training process three different errors were calculated on data extracted from the training process. Error rates are shown in Fig.3., Fig.4., Fig.5 from which the mean loss value was extracted, calculated using the formula above.

In the second Mask R-CNN model training, we used images that were used in the first model training and images that were generated using data augmentations methods, like brightness change and gaussian noise. From data augmentation we gathered about 1200 new wood defect types to learn our new model. These methods were used because they did not change the location of objects of interest, so artificially generated images should not have been marked independently. Highlighted objects coordinate inherited from original images. TensorFlow tensorboard tool was used to draw graphics. These graphics were based on exit data written in the training process of the library used, which stores all information about the errors already listed, in each iteration step.

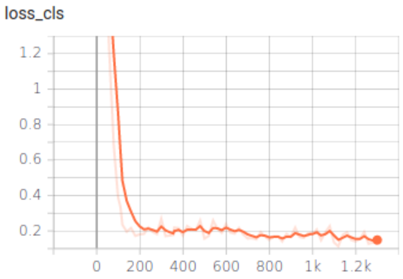

Fig. 3. **Class loss per iteration**

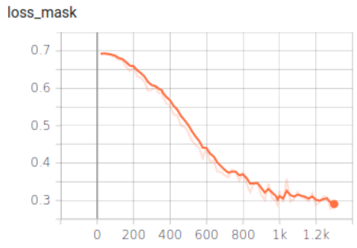

Fig. 4. **Mask loss per iteration**

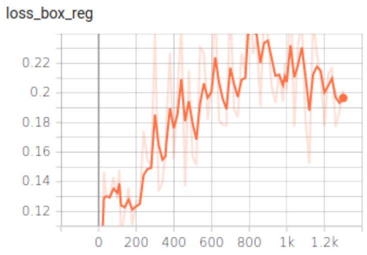

Fig. 5. **Box loss per iteration**

Results and discussion

Our trained model was tested on 8 different oak -wood feature and defect types that are listed in Table 1. Considering that number of training images was fixed and did not consist of large numbers of samples, a satisfactory result was obtained with a total accuracy of ~ 73%. The training process has been performed several times because of the number of marked images was constantly enlarged, and we had to be sure that our model is trained on a good amount of data. The marking process was slow because the time spent on a single image was approximately 5 min.

Our second model performed much better. After data augmentation, the new model was tested on the basis of the same test images. The average accuracy of the second model increased by about 16%, reaching an average of ~ 89%.

When we added some new materials to the training set, we retrained our model and tested it to the same testing images. On 20 new labelled images, it gained about 1% in precision increments. Such an increase was mostly observed for those classes where the number of objects was relatively expressed in additional data. More accurately, the model was able to recognize the small black pins in the test images. The total number of these class interest objects was 216. The precision test was conducted with approximately 130 randomly selected images that were not used in the training process, to ensure that the trained model has not memorized the locations of the objects of interest for these images. Visual prediction example is in Fig. 6.

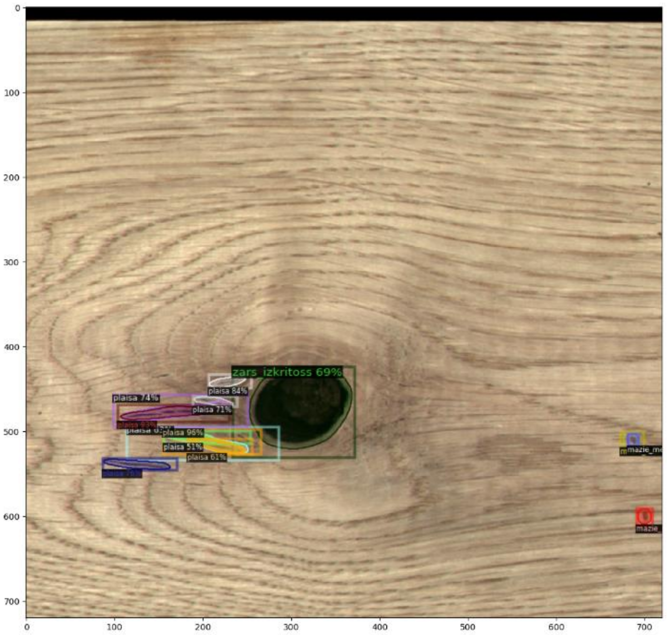

Fig. 6. **Image prediction example**

Class detection results are displayed in Table 3.

Table 3

Detection results of each defect type

Feature/defect class	Number	Detected	False detected
Open knot	3	2 (66.7%)	1 (33.3%)
Dead knot	14	13 (92.9%)	1 (7.1%)
Small black pins	58	45 (77.6%)	13 (22.4%)
Shake	79	68 (86.1%)	11 (13.9%)
Sound knot	17	7 (41.2%)	10 (58.8%)

Table 3 shows that a dead knot was best recognized feature, with 92% returning the correct class. On the other hand, the algorithm had problems identifying a sound knot. The reason may be not enough samples of the sound, and it is possible that the sound knots are very close to the common colour of the structure of the board after the extreme of the image class spectrum.

Wood defect identification research is not new. There are several recent publications based on wood defect image processing. Some of them used residential network architecture to gain better prediction. [15] This study resulted in nearly 95% accuracy. However, since the classification of objects were used, developers cannot get information about the physical sizes of the wood defects. Which is essential in our study, it is also intended to replace the problematic spot with another area of the board.

Conclusions

1. Using data augmentation techniques, it is possible to improve the accuracy of the guess by about 16%, reaching an average of ~ 89%.
2. Mean average prediction ~73% show that use of algorithms is suitable for this type of task.
3. Using image segmentation in wood defect recognition, the defect physical size is obtained, and the defect might be replaced with a sound wood patch.

Acknowledgements

The authors are grateful to all members who were involved in this study.

In accordance with the contract No. 1.2.1.1/18/A/004 between "Forest Sector Competence Centre" Ltd. and the Central Finance and Contracting Agency, concluded on 17th of April, 2019, the study is conducted by the "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 Centre".

References

[1] Baştanlar Y., Özuysal M. "Introduction to machine learning," Methods Mol. Biol., vol. 1107, pp. 105-128, 2014, DOI: 10.1007/978-1-62703-748-8_7.

[2] Anand A. "Unit 13 Image," Tutor. Process. Classif. Remote. Sensed Images, no. May, pp. 41-58, 2018.

[3] "Architecture Of CNN | CNN Image Recognition." [online] [12.04.2021] Available at: https://www.analyticsvidhya.com/blog/2017/06/architecture-of-convolutional-neural-networks-simplified-demystified/

[4] Watanabe T., Wolf D. F. "Instance segmentation as image segmentation annotation," IEEE Intell. Veh. Symp. Proc., vol. 2019-June, pp. 432437, 2019, DOI: 10.1109/IVS.2019.8814026.

[5] Jung H., Lodhi B., Kang J. "An automatic nuclei segmentation method based on deep convolutional neural networks for histopathology images," BMC Biomed. Eng., vol. 1, no. 1, pp. 1-12, 2019, DOI: 10.1186/s42490-019-0026-8.

[6] Kaymak Ç., Uçar A. "A brief survey and an application of semantic image segmentation for autonomous driving," Smart Innov. Syst. Technol., vol. 136, pp. 161–200, 2019, DOI: 10.1007/978-3-030-11479-4_9.

[7] DALSA TELEDYNE, "Linea Color GigE Series 2k and 4k User Manual," pp. 3-32.

[8] He K., Gkioxari G., Dollár P., Girshick R., "Mask R-CNN," IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 2, pp. 386-397, 2020, DOI: 10.1109/TPAMI.2018.2844175.

[9] Mikołajczyk A., Grochowski M. "Data augmentation for improving deep learning in image classification problem," no. August 2019, 2018, DOI: 10.1109/IIPHDW.2018.8388338.

[10] "Wkentaro/labelme: Image Polygonal Annotation with Python (polygon, rectangle, circle, line, point and image-level flag annotation)." [online] [28.03.2021] Available at: https://github.com/wkentaro/labelme

[11] Minaee S., Boykov Y., Porikli F., Plaza A., Kehtarnavaz N., Terzopoulos D. "Image Segmentation Using Deep Learning : A Survey," pp. 1-22.

[12] Ren S., He K. S., Girshick R., Sun J. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137-1149, 2017, DOI: 10.1109/TPAMI.2016.2577031.

[13] Wu Y., Kirillov A., Massa F., Lo W.-Y., Girshick R. Detectron 2. (2019) [online] [12.04.2021] Available at: https://github.com/facebookresearch/detectron2

[14] He K. "Deep Residual Learning for Image Recognition."

[15] Hu J., Song W., Zhang W., Zhao Y., Yilmaz A. "Deep learning for use in lumber classification tasks," Wood Sci. Technol., no. 0123456789, 2019, DOI: 10.1007/s00226-019-01086-z.