



Deep Learning Based Defect Detection and Quality Classification on Lamella Pieces Used in Solid Wood Panel Production

Merve Ozkan¹ · Caner Ozcan^{2,3} · Mahmut Selman Gokmen³

Received: 1 December 2024 / Accepted: 7 August 2025 / Published online: 22 October 2025
© The Author(s), under exclusive licence to Springer Nature Switzerland AG 2025

Abstract

In the solid wood panel industry, combining lamella pieces of consistent quality is critical for final product performance. However, most manufacturers still rely on manual classification by quality control teams, a process that is labor-intensive, error-prone, and inefficient. This reliance often results in production delays, inconsistent quality, and even financial losses due to panels being sold below cost, ultimately eroding customer trust. To address these challenges, this study proposes an expert decision support system based on deep learning to automate lamella quality classification. A Mask R-CNN model with a ResNet-101 backbone was trained on 2972 beech wood images to detect key defects knots, ray cells, and cracks that determine lamella grading. The system then classifies lamellas into AA, BB, CC, and Crack categories, achieving an overall classification accuracy of 90.90%. This work introduces a novel approach, validated on a custom dataset, that reduces dependency on manual labor and significantly enhances efficiency and reliability in solid panel production.

Keywords Mask R-CNN · Decision support system · Artificial intelligence network · Solid wood panel production · Lamella quality classification · Lamella dataset segmentation

✉ Caner Ozcan
canerozcan@uky.edu

Merve Ozkan
mozkan@kastamonu.edu.tr

Mahmut Selman Gokmen
mselmangokmen@uky.edu

¹ Computer Engineering, The Institute of Graduate Studies, Karabuk, Turkey

² Software Engineering, Faculty of Engineering, Karabuk, Turkey

³ Computer Science, University of Kentucky, Lexington, KY, USA

1 Introduction

Solid wood panels are small pieces of wood obtained from lamellas formed by adding small pieces of lamellas of the same quality, end to end or in solid form. At the beginning of the production, it is necessary to decide what kind of wood it will be made of [1]. It is selected as first class or second class from domestic or foreign country sources. The formation of the solid panel begins when the selected trees enter the drying kiln [2]. It goes through a long phase during the production and is presented to the customer at the end of the production process.

Wood is used in many areas such as kitchen, bathroom counter, furniture, playgrounds, wall, column and ceiling covering, office furniture, school furniture, and wooden beams [1]. Due to the advantages they have compared to other wood products, they have more application areas in the market. Some of these advantages are that it has a long service life and is easy to process in machines [2]. One of the most important factors is the mechanical properties of the wood material and its resistance to force in terms of use [3]. For all these reasons, solid panels have become increasingly important in the forest products industry. Although the demands on the products in the plants producing solid wood panels are high, there is still no system that can automate the production chain and completely eliminate the need for manpower.

In the production of solid wood panels, the wood material is divided into lamellas. Panels of the same quality are obtained by combining lamellas of the same quality. The process of segregating lamellas according to quality is carried out by the quality control team in many mills. This process is entirely manual. It is critical that no errors occur during the quality separation of the lamellas. The fact that this process relies on manpower has a negative impact on production productivity. In addition, human error reduces the selling price of the solid wood panel. There is a high demand for solid wood panels, and this negative impact on the production process needs to be eliminated. Therefore, it is necessary to develop an expert decision support system that can support or replace the quality control team.

Solid wood panels are made from many different types of wood. In the quality classification stage, the structures handled by the quality control team change according to the species of wood. In the literature, the classification of tree species in forests has been effectively performed using deep learning algorithms [4, 5]. Tree species have been estimated, and both time and cost savings have been achieved compared to other estimation methods by using Naive Bayes classification techniques [6]. A success rate of 93% was achieved in the tree species estimation study using 14 different European tree species [7]. In addition, the classification of trees in terms of their physico-mechanical properties is used to show the level of strength of trees against force [8]. Fos et al. conducted studies to improve wood quality for solid wood products with 14 different samples taken from Valencia (Spain) [9].

Solid wood panels are made from many different types of wood. In the quality classification stage, the structures handled by the quality control team change according to the species of wood. Wang et al. were performed color classification on the panel piece using K-means, which is an unsupervised learning method. The dataset that contains 200 images is the panels of the European beech. A color histogram was

obtained while classifying the color tones on the solid beech panels. In addition, the unsupervised learning method was used because of the high cost of labeling. The contour coefficient method was preferred to determine the k -value. The classification process was performed by considering various k -values ($k=7$, $k=8$, $k=9$), and a high success rate was achieved [20]. Zhuang et al. achieved a success rate of 97.22% by using the XGBoost method on 108 images [21]. In the study, color classification was performed on solid wood panel surfaces. The model training time was 5.27, and the average testing time for each image was 51 milliseconds. Zhuang et al. classified the solid wood panel as light, medium, and dark. These two studies achieved high success in classifying the colors of solid wood panels [21].

In the field of image processing, Yang et al. [22] conducted a study on the automatic detection and identification of wood surface knots. Their algorithm accurately detected and positioned knots in digital wood images, achieving an identification rate of 88.97% to improve wood utilization efficiency. However, the current algorithm has limitations, as the software is designed only for circular knots, and the obtained results are not sufficient. In another study, machine learning is used to detect and classify knots in timber boards by pre-processing images using Hilbert transform and Gabor filters, then classifying the extracted features using data mining techniques and comparing them with the bagging method [23]. However, the results obtained are very low, indicating that the machine learning methods used are not sufficient to solve the problem. Within the scope of studies carried out in recent years, Gao et al. proposed the TL-ResNet34 method, which combines ResNet34 with transfer learning to detect wood knot defects on a dataset containing 448 images of spruce knots [24]. Gao et al. identified decayed, dry, edge, encased, horn, leaf, sound as knot subclasses with the success rate of 98%. Non-destructive testing of wood knot defects has a high success rate, but it is considered a relatively straightforward problem with a small amount of data. Although non-destructive testing of wood knot defects is highly successful, the complexity of the dataset cannot be measured because it has not been tested with other datasets. Yang et al. [25] presented a model in which the deep residual network (ResNet) as the core network part of the improved SSD model replaces the VGG network part of the original SSD network for surface detection of solid wood defects. With a dataset of over 5000 samples, the model detected defects including live knots, dead knots, decay, mildew, cracks, and pinholes with an average accuracy of 89.7%. The study is limited to single-sided defect detection, and bounding box deviations remain despite SSD optimization, requiring further improvements (Table 1).

In addition to these studies, some studies using optimization algorithms or real-time system studies have been presented. A deep extreme learning machine model, which combines deep learning feature extraction and extreme learning machine classification, is proposed for wood defect detection. To improve the accuracy, a genetic algorithm optimizes the algorithm parameters and achieves a detection accuracy of 96.72% [22]. The model does not consider other potential wood defects beyond dead knots, live knots, and decay, which may limit its applicability. In another study, the detection of wood surface defects in the undamaged surface wormhole, knot, and crack classes was performed using the improved YOLOv3 algorithm, and the mean average precision (mAP) value of 88.5% was achieved [26]. The tree species used in the dataset of 1800

Table 1 Literature search for wood defect detection

Classes	Method	Performance	Literature
dead knot, live knot, blue stain crack, blue stain, pitch streak knot, fixed knot, control	Mix-FCN	mAP 99.14%	[10]
	SSD	mAP 96.1%	[11]
	network mechanism	efficiency rate was 90.19%	
knot	DeepLab V3 ResNet-101	mAP 86.4%	[12]
live knot, resin, knot with crack, marrow knot missing, dead knot, crack	STC-YOLOv5	mAP 84.2%	[13]
knot, branch, stain, lines, edge, area	network mechanism		
bark, dead knot, crack, heartwood, hole	Faster R-CNN in backbone	mean average precision of 97%	[14]
inbark, material lake + wane, mineral	MobileNetV3		
sound knot, under thickness, under width	Betula solid timber algorithms	average detection accuracy 89.69%	[15]
live knot, dead knot	Improved YOLOv5 based on		
resin, knot with crack, crack	Semi-Global Network	mAP 86.4%	[16]
live knot, dead knot	Improved YOLOv5-C3Ghost	mAP 45.9%	[17]
knot with crack, crack, resin	with SimAm module		
dead knot, live knot, worm hole	Detection transformer	86.9%, 89.1%, 90.2%	[18]
live knot, marrow, resin, dead knot	ODCA-YOLO	78.5%	[19]
knot with crack, knot missing, crack without any defects			

images were oak, pine, and birch, each containing 450 images of defects, wormholes, knots, and cracks. Although this is an application that can work quickly in real time, it has relatively low performance. Ge et al. [18] proposed a new defect detection pipeline using an image acquisition device and a mixed data augmentation method to acquire over 16,380 defect images, including live knots, dead knots, and wormholes, with detection based on the DETR model. While MPEND is more than twice as fast as the best model, it has 6% lower accuracy in some cases, but improves accuracy by 1.4% in others at a similar detection speed. Although the proposed method achieves reasonable detection results compared to SOTA, its slow detection speed remains a significant limitation for real-time applications.

Considering the studies in the literature, the Mask R-CNN method has not been applied to beech images on the lamella dataset before. Previous studies often refer to the knot object [22, 24, 27]. The ray cell object segmented in this study is new. Since the lamella images are images of parts formed during mass production in enterprises, the data collection is very difficult. Despite this difficulty, the raw material supply of beech wood was obtained from the regions around Kastamonu (Turkey). The images were obtained from Kastamonu Commodity Exchange Wood and Forestry Product Industry Corporation Joint Stock Company and used with their special permission. The Mask R-CNN method, which has not been applied in the literature, was applied to the lamella images collected from beech trees around Kastamonu (Turkey). The main contribution of our study can be listed as follows:

- The aim is to inform researchers of the success of the method on this dataset and to fill this gap in the literature.
- The ray cell object has not been detected in previous studies. This object, whose samples were collected and segmented on the coverslip images, is another novelty that this study brings to the literature.
- The classification algorithm developed after model training is another novelty of this study. Thanks to this algorithm, an expert decision support system is proposed that can be used in the industrial sector. As a result of the development of this system, there is no need for a quality control team in companies.

2 Methodology

2.1 Background

One of the most preferred deep learning model on image specific tasks such as segmentation, classification, and object detection is convolutional neural networks (CNN) and its variants. The basic working principle of a CNN architecture trained on supervised learning approach relies on meaningful inference from an image given as input [28]. This inference is generally as known as an output, and outputs may vary in different types of formats such as integers, arrays, images, and others, depending on the task.

CNN models share a similar mathematical background with fully connected neural networks (FCNs). The key difference between CNN and FCN architectures lies in the

application of kernel filters during feature extraction. It is known that the input dimensions of CNN models are higher than those of FCN-based models when compared. Guided by these kernel filters, CNN models extract features by processing inputs as numerical vectors in spatial dimensions. As represented in Eq. 1, the mathematical background of a CNN layer relies on the application of a filter matrix (K) on input image (I) [29]. A single layer is known as the core of a CNN-based deep learning architecture. An architecture is generally consist of multiple layers following with activation and normalization layers.

$$(I * K)(x, y) = \sum_{i=-a}^a \sum_{j=-b}^b I(x+i, y+j) \cdot K(i, j) \quad (1)$$

As explained in the introduction, R-CNN and its varieties are able to reach high accuracy rate of object detection when compared to other architectures. Although the R-CNN family has a significant successive rate when compared to CNN architectures, Faster R-CNN outperforms other models in the family in aspect of mAP (mean average precision) [30].

To achieve the goal for this study, segmentation task has a crucial role beside performing detection. Additionally, the model is needed to be able to segment knot, ray cell, and crack objects located on the input image. The recent studies show that the modifications made on R-CNN architecture are able to contribute segmentation process within a high successive rate without experiencing loss of performance. The modified version of Faster R-CNN, known as Mask R-CNN, is a model evolved from the Faster R-CNN architecture. It is enhanced by the addition of an extra branch designed for predicting an object mask (region of interest) in parallel with the existing branch that focuses on bounding box recognition [31].

Faster R-CNN surpasses within the CNN family for its low computation time and high accuracy. Mask R-CNN builds upon these features by adding a masking mechanism. Although the studies employed YOLO, architecture perform a distinguishable results among the other types of CNN architectures, could not meet the expectations for practical scenarios [32]. The main reason for this is that the YOLO architecture is not sensitive for small objects such as knot, cracks, and ray cells as they are introduced [33]. Therefore, Mask R-CNN was the preferred choice for this study.

2.2 Proposed Architecture

The proposed architecture consists of two major steps including feature-based segmentation and quality classification, as shown in Fig. 1. Quality classification algorithm takes the segmentation results of three different models, which is trained upon feature specific tasks. These features represent knot, ray cell, and cracks, as it is explained in previous sections. The models trained on feature-based segmentation generate an input for the algorithm of quality classification, which puts forward a novel approach and distinguishes this study from other relevant studies that do not include the evaluation of lamella's quality.

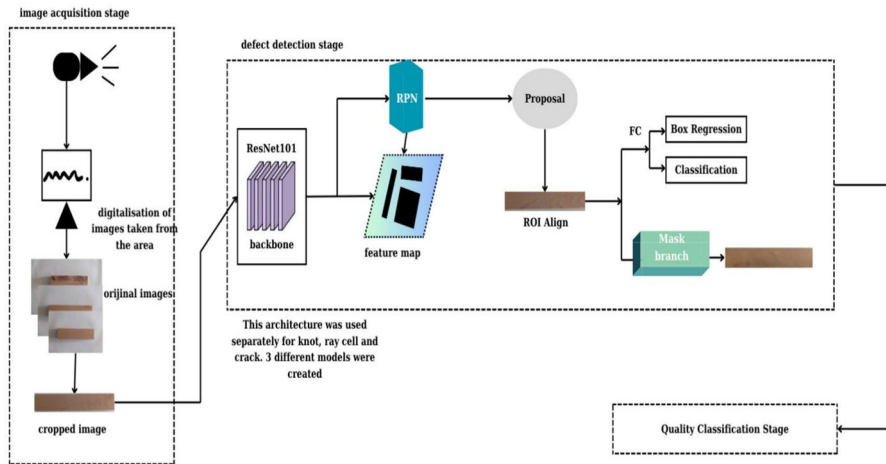


Fig. 1 General architecture of the proposed model

2.2.1 Object Segmentation

The knot object segmented in Fig. 5 is a defect for the lamella. In real-life classification, it is unacceptable to include large size knots in high-quality classes. There are many related studies in the literature. The study by Kilic et al. contains 100 pieces of data [27]. These images were made publicly available. Oak lamellas were used. They used image smoothing, grayscale, image erosion, and histogram thresholding to identify node fault candidates, node fault classification, and finally node labeling. As a result of the study, Kilic et al. achieved 81% success.

Figure 6 shows the ray cell object, one of the structures affecting the classification. It is undesirable for customers and manufacturers to have a ray cell object in the best quality class on the panel to be formed. Generally, the ray cell object is always detected in multiple structures on a lamella. For this reason, when this object is detected, it is not correct to put it in the top quality class.

The object in Fig. 7 is a crack object. In the drying and steaming step of the first stage of solid panel production, it is stated that there is a different moisture content for each type of wood. When the moisture content of the wood is not adjusted correctly at this stage or after incorrect applications in the following stages, cracks appear on the lamella, as shown in Fig. 7. A structure containing cracks is expected to be in the worst quality class.

2.2.2 Quality Classification

The segmentation results obtained from feature-based pre-trained models are used for the purpose of quality assessment with the final quality evaluation algorithm, which is placed at the end of the architecture. The algorithm used for the evaluation of the quality of the input images conducts simple and efficient architecture for classifying

the inputs, as represented in Fig. 8. The classification algorithm is a task-specified designed algorithm and that plays a crucial role for quality classification of lamellas.

2.3 Components of Proposed Architecture

2.3.1 Mask R-CNN Architecture

Mask R-CNN is a model built on top of Faster R-CNN architecture, and it extends the capabilities of Faster R-CNN with instance segmentation process addition to the object detection property of Faster R-CNN. Unlike basic object detection architecture, which identifies and locates objects within images, instance segmentation provides a deeper level of understanding by precisely outlining each object at the pixel level. This dual capacity of detection and segmentation makes Mask R-CNN a versatile and powerful tool in the field [30].

Mask R-CNN architecture operates in two key steps. The first step is the generation of regional proposals where objects might be represented. The next step is the prediction of the class for each object concurrently including bounding boxes. In this step, Mask R-CNN creates a high-resolution mask for each individual instance. These mask features are generated by the backbone of the architecture, ResNet-101, considering the masks utilized during the training process.

Innovative part of Mask R-CNN architecture is the parallel branch for mask prediction alongside the existing branch for bounding box detection, as represented in Fig. 2. This design ensures accurate segmentation without cross-class competition [30].

The regional proposals generated by ResNet-101 model are utilized as features for predicting classes, bounding boxes, and high-resolution masks. The region of interest align (RoI align) process further refines the image through the backbone network into a feature map, which subsequently becomes the input for subsequent operations. The losses in Mask R-CNN encompass classification losses (L_{cls}), masking losses (L_{mask}), and bounding box losses (L_{box}), as outlined in Eq. 1 [34].

$$L = L_{cls} + L_{box} + L_{mask} \quad (2)$$

The region proposal network (RPN) plays a crucial role in Mask R-CNN by proposing potential object regions. For each sliding window position, it generates k simultaneous region proposals. The regression layer outputs 4k coordinate points for k anchor boxes,

Fig. 2 Mask R-CNN architecture

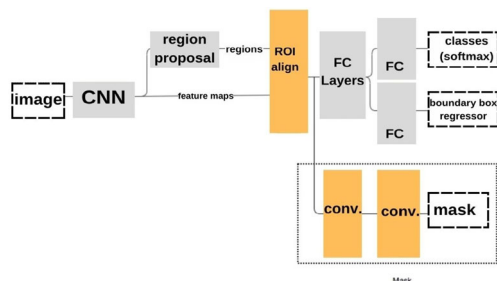
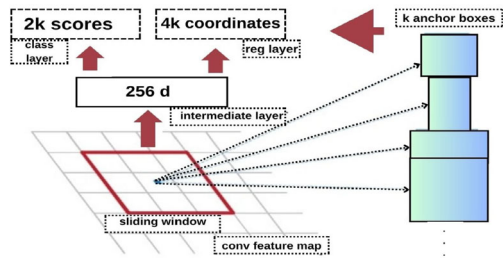


Fig. 3 Region proposal network architecture



while the classifier layer predicts whether these proposed regions contain objects or not. These proposals are analyzed in relation to k reference boxes, also known as anchors, with different width, length, and scale information. A positive label is assigned to regions with high intersection over union (IoU) values, indicating the presence of objects, while a negative label is assigned to regions with low IoU values, indicating no objects (Fig. 3).

The Feature Pyramid Network (FPN) leverages object detection in Mask R-CNN by creating a pyramid of features by utilizing both lower-level and higher-level features. The primary function of FPN is processing variable-scale images effectively. The FPN transforms the image into layers with decreasing resolutions, providing that features at each level can capture objects of various sizes [35].

2.3.2 ResNet-101

ResNet-101 is a deep architecture that consists of 347 layers, making it one of the deepest models proposed for ImageNet [36]. It employs residual blocks to efficiently transfer information from one layer to another, reducing the risk of information loss. ResBlock, as shown in Fig. 4, feeds information from the previous layer to the new layer, helping maintain the flow of information throughout the network.

The output of a residual network layer can be expressed using Eq. 3:

$$y = F(x, W_i) + x \quad (3)$$

Fig. 4 Residual block in ResNet

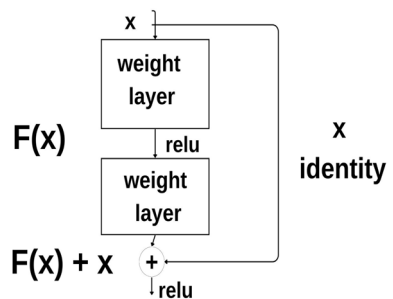


Table 2 Quality assessment rule

Quality classes	Structures in classrooms
AA	There should not be any structure
BB	Knot should be involved
CC	Ray cell mandatory, knot optional
Crack	Crack mandatory, knot, and ray cell optional

2.3.3 Classification Algorithm for Quality Assessment

Classification algorithm is implemented regarding the presence or absence of knots, ray cell, and cracks in the image, and success metrics were measured. The segmentation results are obtained from feature-based trained models (knot, cell, and crack) used for quality classification. The quantity of detected objects in the images is not critical to the classification process. For example, in the case of existing cracks, the occupation of multiple cracks on a lamella is not as important as the model's ability to detect at least one crack. This principle follows similar path to the detection of knots; the identification of a single knot is enough for the classification task. However, for ray cell objects, a specific rule is implemented to improve the success rate in the final classification. This rule, which is distinct from the approach for cracks and knots, is outlined in Table 2.

Table 2 lists the structures expected to be included in the quality classes. When knot, ray cell, and crack objects are detected, quality classification is performed according to Table 2.

3 Experiments

3.1 Dataset

The objects that affect the quality vary according to the type of wood. In this study, beech, which is one of the leafy tree species, is used. It is very difficult to detect objects on this wood type, which is one of the hardwood species, due to its shape. The structures that affect the final classification on beech wood are knots, ray cell, greening, fungus, and cracks. The structures are detected knots, ray cell, and cracks.

Special lamella images taken from the solid panel production line of Kastamonu Commodity Exchange Wood and Forest Products Industry Corporation were studied. FujiFilm X-S1 12MP camera was used to collect the images. The pixel density of the camera is 326 dpi, and the screen resolution is 783 x 587. The interpretation of the images was performed with the help of an expert and the tree type used (Figs. 5, 6 and 7).

The total number of images used in the study is 2972. Three different object detection processes were performed. For each model, the split ratio between the training and test dataset is 80–20%. The total number of images belonging to the knot class is 656. The number of images allocated for training is 584, and the number of images

Fig. 5 Images with knot label taken randomly from the lamella dataset



Fig. 6 Images with ray cell labels randomly taken from the lamella dataset

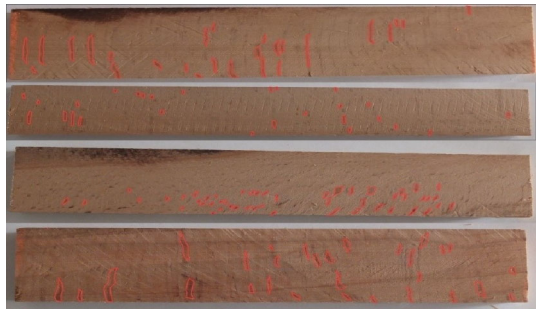


Fig. 7 Images with crack label taken randomly from the lamella dataset



allocated for testing is 72. The total number of images belonging to the ray cell class is 237. While the number of images allocated for training was 211, the number of images allocated for testing was 26. The number of images belonging to the crack class is 1587. Out of 1587 images, 1411 images are reserved for training, and 176 images are reserved for testing. For three objects, convolutional neural networks (CNN) were trained, and three different models were created. The total number of images used was 2480, while the total number of labels accessed was 12,044.

After the training process, the model's results were obtained to classification as present/absent in order to obtain success metric values. At this stage, a total of 360 images were created, 120 each for the crack, ray cell, and knot models in the test dataset. Models were tested on this dataset, and success metrics results were obtained. The total number of images and labels used in all classes are given in Table 3.

The total number of images in the test dataset used in the final classification in the next stage is 132. Using these images, the success of the models in the final classification was measured. Three different models were trained in the mask region-based convolutional neural network (Mask R-CNN). In the following sections, detailed descriptions of the images used in training and testing are given. Since acceptable accuracy values were reached in the success rate of the models, data augmentation was not performed. The dataset used in the project can be shared, if desired, for academic partnerships.

3.2 Working Environment

TensorFlow and Keras libraries are used. GPU is preferred in this project in order to use time efficiently [37]. For GPU utilization, Compute Unified Device Architecture (CUDA) installation is required. CUDA is a parallel programming platform created by NVIDIA that increases the computing performance of the computer [38]. It allows the use of kernels in the computational operations that take place on the GPU. It uses the C language and can process code in random access memory with simple plugins that can be installed. cuDNN (CUDA Deep Neural Network library) is an optimized library developed by NVIDIA that enables deep learning models to run more efficiently on the GPU.

Table 3 Total number of images and labels used in the project

	Total number of images	Total number of label
Knot	656	1006
Crack	1587	7935
Ray cell	237	3103
Present/absent	360	—
Classification		
Final classification	132	—
Total	2972	12,044

The computer used in the project has Windows 10 Pro operating system with NVIDIA Quadro RTX 5000 video card. CUDA 11.0 version was preferred in the project. The cuDNN version compatible with CUDA was determined as 8.5, and the installations were performed after the necessary downloads. After CUDA and cuDNN installations, TensorFlow and Keras packages were installed using the Anaconda terminal screen [39].

3.3 Data Processing

Data labeling is one of the most critical steps in projects using supervised machine learning methods or deep learning methods. Such studies start with the correct interpretation of the data, and in the training phase, the model learns according to the meaningful data. In this study, Make Sense web-based free labeling tool was used [40]. The labeling operations were performed separately on the knot, crack, and ray cell structures. Since these objects do not have features separated by clear lines, polygonal labeling was used. Polygonal labeling is a suitable labeling type for structures with many folds. Networks such as You Only Look Once (YOLO) do not support polygonal labeling. Rectangular labeling method is used in these networks [41]. The networks used in this labeling do not create a mesh mask and perform detection based on the coordinates of the object. Using this tool, labeling operations for ray cell, crack, and knot objects were performed.

3.4 Evaluation

3.4.1 Confusion Matrix

The values used in the calculation of the success metrics are true negative (TN), true positive (TP), false negative (FN), and false positive (FP). These values are based on the class of data labeled by the user before the test and whether the prediction process performed by the model results in a match or not. The cells labeled as present and absent in Table 4 are shaped according to whether the object to be detected is present in the image or not.

Suppose we treat the cells in Table 4 as either having or not having knots. In this case, a true positive is when an image that is classified as a knot in reality is also classified as a knot by the model. A false negative is when the model classifies an image as not a knot when it is actually classified as a knot. False positive is the classification of an image as a knot by the model when it is actually classified as not a knot. True negative is an image that is classified as not a knot by the model that is classified as not a knot in reality.

Table 4 Confusion matrix

		Present	Absent
Real	Present	True positive	False negative
	Absent	False positive	True negative

The terms discussed in Table 4 will be used in the precision, recall, f1-score, and accuracy computation. Equations 4, 5, 6, and 7 contain the precision, recall, f1-score, and accuracy equations, respectively.

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (6)$$

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (7)$$

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (8)$$

The mAP value is a parameter designed to evaluate precision, recall, and f1-score in a single formula. It includes the balance between precision and recall and takes into account both FP and FN. The mAP value changes depending on the IoU threshold. Since the accuracy of the model may vary at different thresholds, the correct threshold value differs for each model. The mAP value is formulated in Eq. 8.

3.5 Setting Hyperparameters

For Mask R-CNN model training, there are default parameter values to be used initially. Hyperparameters are tuned through experiments. The backbone network required for the model can be chosen as ResNet-50 or ResNet-101 [42], which extracts image features. The backbone network ResNet-101 is chosen. Unlike YOLO, Mask R-CNN requires an increase in the number of classes to be trained for the background [43]. For this reason, the number of classes parameter is set to two.

The three models created are knot, ray cell, and crack. Experiments were conducted to determine the detection of minimum confidence, steps per epoch, minimum dimension of image, maximum dimension of image, and IoU threshold parameters for the models. Different values were given during each experiment to ensure that the correct configuration settings were made. For example, steps per epoch were set to 500 since the mAP value obtained when the model was trained by assigning 100 to the steps per epoch parameter was lower than the mAP value obtained when the model was trained by assigning 500 to the steps per epoch parameter. The detection of minimum confidence was set at 0.7. The minimum dimension of image and maximum dimension of image was set at 1024. Hyperparameters are detailed in Table 5.

Table 5 presents the hyperparameters used during the training of three distinct models: the knot model, ray cell model, and crack model. The hyperparameters are identical for all three models. In this context, the global threshold is set to 0.3, which determines the minimum threshold above which detections are considered valid. Similarly, the IoU threshold is set to 0.3. The IoU measures the overlap between the detected object

region and the actual object region, serving as a key for assessing object detection accuracy. The detection of minimum confidence parameter is assigned a value of 0.7, indicating the minimum confidence level required for an object to be considered as detected.

Table 5 also includes parameters related to input image dimensions. Both the image minimum dimension and image maximum dimension are set to 1024, defining the minimum and maximum resolutions used for image processing. All three models are designed for a binary classification problem; hence, the number of classes is set to 2. Finally, the steps per epoch parameter are defined as 500, specifying the number of updates performed in each training epoch, thereby influencing the dynamics of the learning process. These hyperparameters have been carefully selected to be consistent across all three models, ensuring optimal detection of objects with different structural characteristics. The selection aims to enhance model performance while minimizing the overall error rate.

3.6 Implementation of Mask R-CNN Architecture on Lamella Dataset

The Mask R-CNN network uses MS COCO weights for object detection. If a training is started and there is a model resulting from any training, a code block is generated to load the weights of the last model. A logs directory is created in the Mask R-CNN folder, and all weights of the saved models are added here. Since `mrcnn_class_logits`, `mrcnn_bbox_fc`, `mrcnn_box` values should also be included in the weight parameters to be loaded into the model during the training process.

Each iteration consists of 500 batch processes. The approximate duration of one iteration is 221 s. In addition, an interpretation can be made by considering parameters

Table 5 Hyperparameters used when creating the models

	Knot model	Ray Cell model	Crack model
Global Threshold	0.3	0.3	0.3
Iou Threshold	0.3	0.3	0.3
Detection of minimum confidence	0.7	0.7	0.7
Image minimum dimension	1024	1024	1024
Image maximum dimension	1024	1024	1024
Number of classes	2	2	2
Steps per epoch	500	500	500

such as the overall loss value, mask loss value, and class loss value. The process of plotting the loss graph can be visually supported by TensorBoard [44]. TensorBoard allows for the visualization of many parameters on the neural networks we model with the help of graphs. TensorFlow installation is sufficient to use TensorBoard [44].

After setting the necessary configuration parameters for the model, giving the model paths, and assigning the initial weights, the training process was triggered. The learning_rate parameter, which is the learning rate, was assigned a default value of 0.001, and as a result of the tests performed, no change was needed in this parameter value. Three different models were created. The best accuracy value for the models was achieved after different iteration numbers.

3.7 Utilization of Pre-trained Models for Quality Classification

It was mentioned that even if there were more than one of the same object on the lamella, it would be sufficient if the model detected only one. This is a generally valid rule. However, in order to improve the classification performance, a different method was followed for the ray cell class. Ray cell objects are usually numerous on a coverslip. Ray cell objects are formed by the unwanted intersection of the wood fibers with the log cut direction, so they are not expected to be few. It does occur in some cases, but these are rare. Therefore, if ray cell objects are detected on six or more objects, the algorithm classifies the image as CC. This condition resulted in an overall decrease in success in the CC class but an increase in accuracy in the other classes. These details will be discussed in the results section. The flowchart of the final classification process and the predictions of the models are given in Fig. 8.

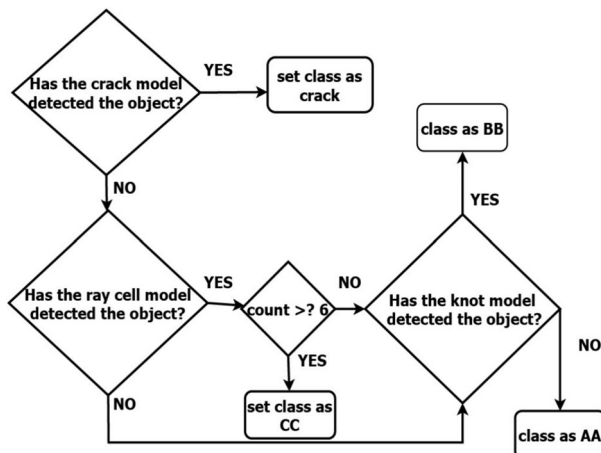


Fig. 8 Algorithm flowchart diagram for final classification



Fig. 11 Prediction and masking of the crack model after 150 iterations

up to 250. The best model was reached after 150 iterations. The overall loss value was reduced from 1.4 to 0.59. Figure 11 shows the confidence and IoU score values after 150 iterations. In 150 iterations, the model completed its learning, and overfitting was prevented.

Within the scope of this study, it will be sufficient for the ray cell and crack and knot models to perform object detection regardless of their number in images containing knots, ray cells, and cracks. It is seen that the trained models meet this requirement.

4.2 Quality Classification Results

Three different models were trained using the Mask R-CNN deep learning network, and the test set was subjected to a classification process after training. Considering the best precision, recall, accuracy, and f1-score values achieved by the models, it was decided which of the models formed as a result of 50, 100, 150, 200, 250 iterations would be used for the project. Although the mAP parameter obtained by the model as a result of iterations was taken into consideration, it was evaluated that it would not be sufficient alone. The mAP value calculated by the model after the training is not important because the location of the structure to be detected on the image and how many of them are not important. mAP is a value that calculates over real reference labels and reaches high percentiles when it finds all labeled objects one-to-one. However, it is not necessary to detect all labels with true reference values. Although there may be more than one crack on the object with a crack, it is sufficient if the model finds even one of the cracks when asked. Detection of one crack object or one ray cell object is sufficient to determine the class of the lamella part. For this reason, the model was classified outside of the mAP values generated by the model after training, and the recall, accuracy, precision, and f1-score parameters were recalculated over the test data, and the most accurate values were obtained. The final classification process was then performed.

In the part of segmenting the knot object, the best iteration number of the model created in Mask R-CNN is 250. During the testing process for the knot model, 120 test images were collected. In this dataset, there are 61 images belonging to the knot class and 59 images that do not belong to the knot class. Sixty-one of the 61 images belonged to the TP and FN groups, while 59 images belonged to the FP and TN groups. The complexity matrix of the knot model given in Table 6 supports the above information.

Table 7 shows the performance metrics for the knot model. The accuracy metric of 95.83% is considered sufficient for the classification ability of the knot model.

Table 6 Confusion matrix of the knot model

	Predict		Total
	Knot	Non-knot	
Knot	61 (TP)	0 (FN)	61
Non-knot	5 (FP)	54 (TN)	59
Total	66	54	120

The information that the best model trained by the Mask R-CNN network was formed after 200 iterations is given in Sect. 4.1 under the heading of segmenting the crack object. At this stage, the crack model was expected to classify the images belonging to the crack class or not belonging to the crack class in the images belonging to the test set allocated for the crack model. For the crack model classification stage, 120 images were allocated for the test process. In this dataset, there are 78 images belonging to the crack class and 42 images belonging to the non-crack class. As a result of the model classification process, the complexity matrix in Table 8 was obtained. In this matrix, 78 images are composed of the sum of TP and FN groups. This total includes all images belonging to the crack class. The group of images that do not belong to the crack class are FP and TN.

Table 9 shows the performance metrics of the crack model. Here, the accuracy value of 85% indicates that the model that performs the crack classification process is ready for the final classification process to be performed in the next stage.

In Sect. 4.1, the best number of iterations for the segmentation process of the ray cell object was decided to be 150, and detailed information was presented. The model obtained at this stage was expected to classify the images in the test set. Here, a classification stage is applied to answer the question of whether it is a ray cell class or not. After this classification, the data in Table 10 were obtained. For the classification process to be performed by the ray cell model, 120 images were set. In this dataset, 65 images belonging to the ray cell class and 55 images belonging to the non-ray cell class were included. The sum of TP and TN is equal to the sum of the images belonging to the ray cell class, that is, 65. The sum of the FP and TN groups is equal to the sum of the images belonging to the non-ray cell class, that is, 55. From the values obtained here, the values of the required performance metrics for the ray cell object were calculated.

The accuracy of the classification process for the ray cell model was 76.66%. For the final classification process to be carried out in the next stage, this rate is considered to be a sufficient result (Table 11). Four final classes named AA, BB, CC, and crack

Table 7 Knot model performance metrics and values

Metric value	Parameter values
Precision	0.92
Recall	1
F1-score	0.96
Accuracy	0.95

Table 8 Confusion matrix of the crack model

	Predict		Total
	Crack	Non-crack	
Crack	64 (TP)	14 (FN)	78
Non-crack	4 (FP)	38 (TN)	42
Total	68	52	120

Table 9 Crack model performance metrics and values

Metric value	Parameter values
Precision	0.94
Recall	0.82
F1-score	0.87
Accuracy	0.5

Table 10 Confusion matrix of the ry cell model

	Prediction		Total
	Ray cell	Non-ray cell	
Ray cell	60 TP)	5 FN)	65
Non-ray cell	23 FP)	32 TN)	55
Total	83	37	120

Table 11 Ray cell model performance metrics and values

Metric value	Parameter values
Precision	0.72
Recall	0.92
F1-score	0.81
Accuracy	0.76

Table 12 Algorithm decision on sample results

	Sample_AA	Sample_BB	Sample_CA
Existing structure	No defect	No defect	Crack
Detected structure	1 ray cell	1 knot	1 crack 25 ray cells
Real class	AA	BB	Crack
Class decided	AA	BB	Crack
By algorithm			

were determined. Thirty-three images belonging to each class were used, totaling 132 images in the dataset. Table 12 shows the structures on the individual images, the structures detected by the model, the class to which the image actually belongs, and the algorithm's decision on which class the image should belong. Table 12 shows random sample results from the dataset for the algorithm's decision.

Thirty-three images belonging to the AA class were used. There are no structures on the images in this dataset. Despite this, the model detected ray cell and knots on 11 images in the dataset. If the model detected six ray cell objects, they should be in the CC class. When a total of six or less ray cell objects are detected on an image, they are not taken into account in the final classification. This final classification is a rule set to increase the success of the algorithm. Eight images in the test dataset, six or less ray cell objects were detected. Thanks to this rule, misclassification was prevented. The algorithm is run by taking this rule into account in all AA, BB, CC and crack classes. With 30 correct and 3 incorrect, the success rate for the AA class was 90.90%.

In the test dataset of the BB class, 33 images were used. In this dataset, only the knot object is present on the images. However, the ray cell model detected ray cell objects in five images. The flowchart diagram of the final classification process is given under the experimental studies section in Sect. 3.7. In the classification process of the BB class, 33 out of 33 images were classified into the correct class by the algorithm. The success rate obtained for the BB class on the test data was determined as 100%. In the BB dataset, the algorithm did not misclassify.

There are 33 images in the test dataset of the CC class. In this dataset, only the ray cell object was included. Knots, cracks, and ray cell were detected by the models on the images. The image is classified as BB due to the knot object. With 27 correct and 6 incorrect classifications out of 33 images, the success rate of the CC class was 81.81%.

There are 33 images in the crack class test set of the 33 images, 30 were correctly classified and 3 were incorrectly classified. The success rate of the crack class was 90.90%.

The success rate for the crack class is 90.90%. Table 13 shows the success values of the classes.

Table 13 Percentiles for all classes and general achievement

Final classes	Success rates
AA	90.90%
BB	100%
CC	81.81%
Crack	90.90%
General success	90.90%

5 Discussion and Conclusion

When the studies in the literature are examined, many object detection processes have been successfully performed with deep learning networks. However, there is no expert decision support system to perform the classification of lamella parts in solid panel production. In most companies, this stage relies on manpower. The fact that this process is based on manpower brings many problems such as differences in production quality and human errors. This is a problem that needs to be overcome in factories.

Within the scope of the study, the aim is to eliminate the problems that arise due to the fact that the quality separation process is based on manpower during solid panel production. Experiments were carried out on images of beech belonging to Kastamonu Commodity Exchange Wood and Forestry Products Industry Corporation in Kastamonu province in Turkey.

The Mask R-CNN model was chosen for this study because it can perform not only object detection but also pixel-level segmentation. Unlike other popular models (e.g., YOLO or Faster R-CNN), Mask R-CNN is able to capture fine structures and boundaries more accurately by generating boundary masks for each object.

While the U-Net model is a common choice for pure segmentation tasks, it is not suitable for the holistic approach of this study as it cannot handle object detection and segmentation together. Similarly, models such as YOLO are optimized for fast object detection rather than segmentation and thus fall short of the required pixel-level accuracy.

For these reasons, the Mask R-CNN model was considered the most suitable option, considering the objectives of the study and the data structure, due to its superior segmentation capability, which is frequently emphasized in the literature.

Mask R-CNN network was preferred in the segmentation phase. ResNet-101 was used as the backbone network in Mask R-CNN. In addition to the process performed by other deep learning networks, the object to be detected with the masking process is segmented from the background image.

Model training was performed for crack, knot, and ray cell objects. The position, size, or number of these objects on the image is not important. Even if it detects 1 out of 3 cracks on the lamella, the final classification is correct. For this reason, the performance metrics of the models were determined as precision, sensitivity, f1-score, and accuracy. At this stage, a classification was made for the evaluation of the metrics as “cracks present/absent,” “knots present/absent,” and “ray cell present/absent.” As a result of this classification, the accuracy of the best knot model trained on the Mask

R-CNN network was 95.83%, the accuracy of the crack model was 85%, and the accuracy of the ray cell model was 76.66%.

After this stage, the final classification was made according to the detected objects. In these classification groups, AA class is expected to have no object on the lamella, and BB class is expected to have only the knot object. In the CC class, the ray cell object should definitely be present, and the knot object may also be present. In the crack class, the crack object should definitely be included, but the knot and ray cell may also be included. To increase the success rate in the final classification, if the number of ray cell objects detected is less than or equal to 6, they will be ignored in the classification. The ray cell object normally appears multiple times on the lamella fragment. This rule is a process to improve the classification performance generated from this information. Taking all this information into account, the final classification performance was 90.90% for the AA class, 100% for the BB class, 81.81% for the CC class, and 90.90% for the crack class. The success rate obtained in the general classification is 90.90

As a result of the final classification, the accuracy of the knot model was increased from 95.83 to 100%. The accuracy of the crack model was increased from 85 to 90.90%. The accuracy of the ray cell model was improved from 76.66 to 81.81%.

The 90.90% success rate obtained in the overall classification shows the success of the decision support system in the final classification. This is an acceptable and successful result when considered as a real-life application.

The places where the knot structure is formed on the lamella images are generally darker than the color of the tree. Some darkness occurs in certain parts of the lamella even though there are no knots. The knot model identified these dark circles as knots. The knot model achieved a high rate of success. This is because the structures marked as knots, although they are not knots, are extremely similar to real knots. During production, transverse saw tooth marks appear on some lamella. The ray cell model marked these marks as ray cell.

The crack model recognizes longitudinal dark lines as cracks. Since there is usually a crack label in the longitudinal dark lines, the model marks them as cracks even if there is no crack in the longitudinal dark line. The model also detects cracks in dark lines in the shape of a tree pattern.

In future studies, it is aimed to further increase the accuracy by improving the defective structures of the crack, ray cell, and knot model. In addition, we will also work on the detection of other structures considered in the real-life lamella classification problem to deep learning networks. Within the scope of this study, beech wood, one of the leafy tree species, was used. Apart from this, it is planned to detect and classify the structures on leafy, coniferous, and tropical tree species used in solid panel production with the help of deep learning networks.

Acknowledgements This study was supported by Kastamonu Commodity Exchange Wood and Forestry Product Industry Corporation. Also, this work was supported by Scientific Research Projects Unit of Karabük University under project number KBÜBAP-22-YL-096. The authors appreciate the financial and scientific support.

Author Contributions Merve Özkan: conceptualization of this study, methodology, software, writing—review, editing. Caner Özcan: supervision, editing, conceptualization of this study, methodology, writing—review. Mahmut Selman Gökmen: editing, conceptualization of this study, methodology, writing—review.

Data Availability No datasets were generated or analyzed during the current study.

Declarations

Ethics Approval and Consent to Participate No ethical rules were ignored for the data used in the study.

Conflict of Interest The authors declare no competing interests.

References

1. Hatipoğlu M (2018) Classification of the natural fossilized woods (SiO₂) according to their macroscopic appearances and microscopic structures; a case of the fossilized wood forest in the Gündül-Ankara Region 3(4):109–127
2. Mengeloglu F, Kurt R (2004) Laminated veneer lumber (LVL) and glued laminated timber (Glulam) 7(1)
3. İlkuçar M, Kaya Ah, Çifci A (2018) Predicting wood types in terms of mechanical properties using artificial neural networks 8(1): 75–83. <https://doi.org/10.17714/gumusfenbil.310585>
4. Xi Z, Hopkinson C, Rood SB, Peddle DR (2020) See the forest and the trees: effective machine and deep learning algorithms for wood filtering and tree species classification from terrestrial laser scanning. ISPRS J Photogramm Remote Sens 168:1–16. <https://doi.org/10.1016/J.ISPRSJPRS.2020.08.001>
5. Phan HT, Nguyen NT, Hwang D (2023) Fake news detection: a survey of graph neural network methods. Appl Soft Comput 139:110235. <https://doi.org/10.1016/J.ASOC.2023.110235>
6. Kılıncarslan E, Türker YŞ, İnce M (2020) Prediction using different classification methods of tree species depending on contact angle values Abstract. J Bartin Fac For 22(3):861–870. <https://doi.org/10.24011/barofd.697098>
7. Fabijańska A, Danek M, Barniak J (2021) Wood species automatic identification from wood core images with a residual convolutional neural network. Comput Electron Agric 181. <https://doi.org/10.1016/J.COMPAG.2020.105941>
8. As N, Dündar T, Büyüksarı Ü (2016) Classification of wood species grown in Turkey according to some physico-mechanic properties. J Fac For Istanbul Univ 66(2):727–735. <https://doi.org/10.17099/jffiu.93670>
9. Fos M, Oliver-Villanueva JV, Vazquez M (2023) Radial variation in anatomical wood characteristics and physical properties of Paulownia elongata x Paulownia fortunei hybrid Cotevisa 2 from fast-growing plantations. European Journal of Wood and Wood Products. <https://doi.org/10.1007/s00107-023-01941-8>
10. He T, Liu Y, Xu C, Zhou X, Hu Z, Fan J (2019) A fully convolutional neural network for wood defect location and identification. IEEE Access 7:123453–123462. <https://doi.org/10.1109/ACCESS.2019.2937461>
11. Ding F, Zhuang Z, Liu Y, Jiang D, Yan X, Wang Z (2020) Detecting defects on solid wood panels based on an improved SSD algorithm. Sensors (Switzerland) 20(18):1–17. <https://doi.org/10.3390/s20185315>
12. Urtans E, Bumanis K, Vecins V, Ancans M, Andrijanova A, Upenieks MT, Volkovs K (2022) Detection of knots in oak wood planks: instance versus semantic segmentation. In: 2022 IEEE 5th international conference on big data and artificial intelligence (BDAI), pp 163–168. <https://doi.org/10.1109/BDAI56143.2022.9862633>
13. Han S, Jiang X, Wu Z (2023) An improved YOLOv5 algorithm for wood defect detection based on attention. IEEE Access 11:71800–71810. <https://doi.org/10.1109/ACCESS.2023.3293864>
14. Mohsin M, Balogun O, Haataja K, Toivanen P (2023) Convolutional neural networks for real-time wood plank detection and defect segmentation. F1000Research 12. <https://doi.org/10.12688/f1000research.131905.1>

15. Ji M, Zhang W, Diao X, Wang G, Miao H (2023) Intelligent automation manufacturing for Betula solid timber based on machine vision detection and optimization grading system applied to building materials. *Forests* 14(7). <https://doi.org/10.3390/f14071510>
16. Meng W, Yuan Y (2023) SGN-YOLO: detecting wood defects with improved YOLOv5 based on semi-global network. *Sensors* 23:8705. <https://doi.org/10.3390/s23218705>
17. Xu J, Yang H, Wan Z, Mu H, Qi D, Han S (2023) Wood surface defects detection based on the improved YOLOv5-C3Ghost with SimAm module. *IEEE Access PP*. <https://doi.org/10.1109/ACCESS.2023.3303890>
18. Ge Y, Jiang D, Sun L (2023) Wood veneer defect detection based on multiscale DETR with position encoder net. *Sensors* 23(10). <https://doi.org/10.3390/s23104837>
19. Wang R, Liang F, Wang B, Mou X (2023) ODCA-YOLO: an omni-dynamic convolution coordinate attention-based YOLO for wood defect detection. *Forests* 14:1885. <https://doi.org/10.3390/f14091885>
20. Wang Z, Zhuang Z, Liu Y, Ding F, Tang M (2021) Color classification and texture recognition system of solid wood panels. *Forests* 12(9):2–18. <https://doi.org/10.3390/f12091154>
21. Zhuang Z, Liu Y, Ding F, Wang Z (2021) Online color classification system of solid wood flooring based on characteristic features. *Sensors (Switzerland)* 21(2):1–13. <https://doi.org/10.3390/s21020336>
22. Yang Y, Zhou X, Liu Y, Hu Z, Ding F (2020) Wood defect detection based on depth extreme learning machine. *Appl Sci (Switzerland)* 10(21):1–14. <https://doi.org/10.3390/app10217488>
23. Mohan S, Venkatachalapathy K (2012) Wood knot classification using bagging. *Int J Comput Appl* 51(18):50–53. <https://doi.org/10.5120/8146-1937>
24. Gao M, Chen J, Mu H, Qi D (2021) A transfer residual neural network based on ResNet-34 for detection of wood knot defects. *Forests* 12(2):1–16. <https://doi.org/10.3390/f12020212>
25. Yang Y, Wang H, Jiang D, Hu Z (2021) Surface detection of solid wood defects based on SSD improved with ResNet. *Forests* 12(10):2–11. <https://doi.org/10.3390/f12101419>
26. Wang B, Yang C, Ding Y, Qin G (2021) Detection of wood surface defects based on improved YOLOv3 algorithm 16(4)
27. Kılıç Ö, Susuz DM, Süzek B (2019) A quality control system prototype for detecting knot defects in the wooden panel manufacturing. *Mugla J Sci Technol* 5(1):24–33. <https://doi.org/10.22531/muglajsci.473338>
28. Kumar Singh M, Baluja GS, Prasad Sahu D (2017) Understanding the convolutional neural network it's research aspects in deep learning. *Int J Res Appl Sci Eng Technol (IJRASET)* 5(VI):867–871
29. Zhang W, Itoh K, Tanida J, Ichioka Y (1990) Parallel distributed processing model with local space-invariant interconnections and its optical architecture. *Appl Opt* 29(32):4790. <https://doi.org/10.1364/AO.29.004790>
30. He K, Gkioxari G, Dollár P, Girshick R (2018) Mask R-CNN
31. Tahir H, Shahbaz Khan M, Owais Tariq M (2021) Performance analysis and comparison of faster R-CNN, mask R-CNN and ResNet50 for the detection and counting of vehicles. In: 2021 International conference on computing, communication, and intelligent systems (ICCCIS). IEEE, ???, <https://doi.org/10.1109/icccis51004.2021.9397079> <http://dx.doi.org/10.1109/ICCCIS51004.2021.9397079>
32. Prasetyo E, Suciati N, Fatichah C (2020) A comparison of YOLO and mask R-CNN for segmenting head and tail of fish. In: 2020 4th International conference on informatics and computational sciences (ICICoS). IEEE, ???, <https://doi.org/10.1109/icicos51170.2020.9299024> <http://dx.doi.org/10.1109/ICICoS51170.2020.9299024>
33. Xu X, Zhao M, Shi P, Ren R, He X, Wei X, Yang H (2022) Crack detection and comparison study based on faster R-CNN and mask R-CNN. *Sensors* 22(3):1215. <https://doi.org/10.3390/s22031215>
34. Albayrak E, Yayla R, Yüzgeç U (2021) Mask R-CNN ile İHA Görüntülerinden Araç Tespiti (March)
35. FPN (2023) FPN (feature pyramid networks). <https://wikidocs.net/162976>. Accessed 08 Aug 2023
36. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)
37. Kalaiselvi T, Sriramakrishnan P, Somasundaram K (2017) Survey of using GPU CUDA programming model in medical image analysis. *Inf Med Unlocked* 9:133–144. <https://doi.org/10.1016/J.IMU.2017.08.001>
38. Kirk D (2007) NVIDIA CUDA software and GPU parallel computing architecture. International Symposium on Memory Management, ISMM 103. <https://doi.org/10.1145/1296907.1296909>
39. Anaconda (2023) Anaconda Setup Settings. <https://www.anaconda.com/>. Accessed 08 Aug 2023
40. MakeSense (2023) Make Sense. <https://www.makesense.ai/>. Accessed 08 Aug 2023

41. Ozel MA, Baysal SS, Şahin M (2021) Crack detection in dynamic test drive suspension parts with deep learning algorithm (YOLO). *Eur J Sci Technol* (26):1–5. <https://doi.org/10.31590/ejosat.952798>
42. Kaggle (2023) ResNet-101 — Kaggle. <https://www.kaggle.com/datasets/pytorch/resnet101>. Accessed 08 May 2023
43. Garg D, Goel P, Pandya S, Ganatra A, Kotecha K (2018) A deep learning approach for face detection using YOLO. 1st International Conference on Data Science and Analytics, PuneCon 2018 - Proceedings, pp 1–4. <https://doi.org/10.1109/PUNECON.2018.8745376>
44. Nguyen VT, Dang T, Jin F (2018) Predict saturated thickness using tensorboard visualization. *EnvirVis 2018 - Workshop on Visualisation in Environmental Sciences*, 35–39. <https://doi.org/10.2312/envirvis.20181135>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.