Kenan Kılıç*[1], Kazım Kılıç[2], İbrahim Alper Doğru[3], Uğur Özcan[4]

Comparison of Various Feature Extractors and Classifiers in Wood Defect Detection

Usporedba različitih ekstraktora i klasifikatora svojstava u otkrivanju grešaka drva

ORIGINAL SCIENTIFIC PAPER
Izvorni znanstveni rad
Received – prispjelo: 20. 5. 2024.
Accepted – prihvaćeno: 23. 12. 2024.
UDK: 630*85
https://doi.org/10.5552/drvind.2025.0217

© 2025 by the author(s).
Licensee University of Zagreb Faculty of Forestry and Wood Technology.
This article is an open access article distributed
under the terms and conditions of the
Creative Commons Attribution (CC BY) license.

ABSTRACT • *Detection of defects on wood during quality processes in the wood industry is extremely important both economically and in terms of production and use. In order to minimize the time and cost loss caused by products obtained with defective wood, manufacturers want to detect defects in wood early by applying quality control process. For this purpose, in this study, some experiments are carried out using texture analysis methods and machine learning classifiers to detect defective wood from wood images. The features of wood images in the dataset taken from literature are extracted separately with six texture feature extractors to detect defective wood. Features are classified using twelve different machine learning classifiers, primarily tree-based ensemble classifiers. Cross-validation is used in all experiments to reduce classifier bias. The results obtained are presented comparatively in terms of each feature and classifier. The findings show that the most effective features in detecting defective wood are extracted by the Local Binary Pattern (LBP) method and the most effective classifier is the Random Forest Algorithm. An accuracy rate of 96.75 % is achieved with the LBP-RandomForestClassifier and, classification performance is also presented for each algorithm by creating hybrid feature vectors.*

KEYWORDS: *wood defect detection; feature extraction; machine learning; wood products engineering; computer vision*

SAŽETAK • *Otkrivanje grešaka na drvu tijekom kontrole kvalitete u drvnoj industriji iznimno je važno kako u ekonomskom, tako i u proizvodnom smislu. Da bi se smanjili troškovi i gubitci vremena zbog drvnih proizvoda s greškama, cilj proizvođača je procesom kontrole kvalitete rano otkriti greške na drvu. Stoga je u ovoj studiji istražena mogućnost otkrivanja grešaka na drvu sa slika primjenom metoda analize teksture i klasifikatora strojnog učenja. Obilježja slika drva preuzetih iz literature izdvojena su uz pomoć šest ekstraktora teksture kako bi se otkrile greške na drvu, a zatim su klasificirane uz pomoć 12 različitih klasifikatora, ponajprije klasifikatora ansambla stabla odlučivanja dobivenih strojnim učenjem. U svim je eksperimentima primijenjena unakrsna provjera kako bi*

* Corresponding author
[1] Author is researcher at Gazi University, Graduate School of Natural and Applied Sciences, Department of Wood Products Industrial Engineering, Ankara, Turkey and at Yozgat Bozok University, Vocational School of Yozgat, Department of Design, Yozgat, Turkey.
[2] Author is researcher at Gazi University, Graduate School of Natural and Applied Sciences, Department of Computer Engineering, Ankara, Turkey and at Yozgat Bozok University, Vocational School of Yozgat, Department of Computer Programming, Yozgat, Turkey.
[3] Author is researcher at Gazi University, Faculty of Technology, Department of Computer Engineering, Ankara, Turkey.
[4] Author is researcher at Gazi University, Faculty of Technology, Department of Wood Products Industrial Engineering, Ankara, Turkey.

se smanjila eventualna pristranost klasifikatora. Dobiveni su rezultati prikazani komparativno, prema svojstvima i klasifikatorima. Rezultati pokazuju da su najučinkovitija svojstva za otkrivanje grešaka na drvu izdvojena metodom lokalnoga binarnog uzorka (LBP), a najučinkovitiji je klasifikator algoritam slučajnih šuma. Stopa točnosti od 96,75 % postiže se kombinacijom LBP metode i algoritma slučajnih šuma, a za svaki algoritam predstavljena je i izvedba klasifikacije stvaranjem hibridnih vektora svojstava.

KLJUČNE RIJEČI: *detekcija grešaka na drvu; ekstrakcija svojstava; strojno učenje; izrada proizvoda od drva; računalni vid*

1 INTRODUCTION
1. UVOD

Throughout history, wood has been used as a basic building material for many industries and continues to be used in a wide range of applications today. Wood is unique and its natural structure, aesthetic appeal, strength and environmentally friendly properties offer great value to a range of industries from construction to furniture manufacturing. However, the use of defective wood in the production process, and the defects that can occur as a result, can lead to serious economic losses as well as safety risks. Early detection and effective classification of such defects is therefore vital to the wood industry. In addition, the decline of wood resources has become an important focus for practitioners and researchers in recent years in order to use forest resources in a more sustainable manner. The rapid and effective detection of surface defects in wood can increase the utile efficiency of wood and reduce excessive wood consumption (Pölzleitner and Schwingshakl 1992; Schmoldt *et al*., 1997; Norlander *et al*., 2015).

Traditionally, wood grading has been done manually. Today, however, automatic grading machines are widely used to speed up the process. By using the same set of mechanical devices and changing the grading task, different sorting tasks can be easily performed. Among these tasks, classification is perhaps the most important (Hu *et al*., 2019). Unlike traditional methods that rely on visual inspection by operators, methods used to detect surface defects in wood involve computer analysis of images of the wood surface. A Charge-Coupled Device (CCD) camera is usually used for this process. The process of recognizing wood defects is based on the design of the image analysis algorithm, with frequent use of digital image processing (Xie, 2013). The classification process typically begins with image pre-processing, such as greyscale transformation, histogram equalization, spatial or frequency domain filtering. Next, the wood images are processed to extract defect features. Finally, a machine learning algorithm is used to classify the images (Chen *et al*., 2023a). Machine learning is a technology that can automatically generate predictions and decisions by learning from data. This technology makes it possible to develop solutions to complex problems with different algorithms used to obtain results. Each algorithm is designed to understand a specific data structure and make inferences, and is particularly effective with large data sets. These algorithms can be broadly categorized into three main types: unsupervized learning, supervized learning and reinforcement learning. The combination of digital image processing and machine learning algorithms is the preferred methodology for detecting and classifying wood knot defects (Qi *et al*., 2010; Mu and Qi, 2009). This methodology is not only used for wood defect detection but also for quality control in automated production lines in various sectors such as textiles fabrics, ceramic tiles and pharmaceuticals (Gao *et al*., 2021; Liu *et al*., 2021; Shahrabadi *et al*., 2022; Zhang *et al*., 2023).

The detection of defects on the surface of industrial products has become a very promising area of academic research. There are many studies in the literature dealing with fabric defect detection (Raheja *et al*., 2013; Liu and Zheng, 2020; Liu and Le, 2021), leather surface inspection (Hoang *et al*., 1997; Chen *et al*., 2024), detection of defect in pharma (Galata *et al*., 2021), defect detection in electronic surfaces (Tsai and Huang, 2019; Chen *et al*., 2023b), metallic surface defect detection (Tao *et al*., 2018), concrete crack detection (Lei *et al*., 2024) and defect detection of mobile phone surface (Jian *et al*., 2017). Studies specific to the wood industry are also available in the literature. Zhang *et al*. (2015) used principal component analysis (PCA) and compressed sensing to identify wood defects in wood plate images. YongHua and Jin-Cong (2015) focused on three common wood defects: dead knots, poles and living knots. In their study, they introduced a hybrid defect detection method based on wood surface texture features, which combines the advantages of Tamura texture and Grey-Level Co-occurrence Matrix (GLCM) methods. Li *et al*. (2017) presented a wood defect detection method leveraging linear discriminant analysis (LDA) and the utilization of compressed sensor images. Chang *et al*. (2018) applied convex optimization (CO) and the Otsu segmentation method to obtain a comprehensive image of wood surface defects. The results between the original image and the defect image are used to evaluate the segmentation performance. A classification and regression tree (CART) classifier is then constructed. Li *et al*. (2019)

proposed a classification algorithm for distinguishing between cracks and linear mineral lines on the surface of birch veneer. This algorithm relies on Local Binary Pattern and Local Binary Differential Excitation Pattern for effective classification. Urbonas *et al*. (2019) presented an automatic visual inspection system for locating and classifying defects on wood veneer surfaces using a faster Region-Based Convolutional Neural Network (faster R-CNN). Shi *et al*. (2020) introduced an efficient detection method with high accuracy and speed for online production. They developed an integrated model, the Glance Multi-Channel Mask Region Convolutional Neural Network (R-CNN), specifically designed for wood veneer defect detection. This model incorporates both a Glance network and a multichannel mask R-CNN. Wu *et al*. (2022) developed a wood surface defect detection approach based on feature fusion. The Support Vector Machine (SVM) is used as the classifier for this approach. There is no previous literature on the classical machine learning approach in the form of binary classification regarding the dataset used in the study. Instead, a general literature summary is given in Table 1.

The proposed study aims to evaluate the effectiveness of different image processing methods and classification algorithms in detecting defects on wood surface. The primary objective is to improve the early detection of defects in wood and to evaluate the performance of different feature extraction methods and classification algorithms in achieving this objective. In addition, the study aims to contribute to the identification of potentially valuable techniques and algorithms for defect detection in both the production and utilization processes of wood. For this purpose, an extensive dataset of 20,276 images of both defective and undefect wood surfaces is used. Dataset balancing is a process to address sample count imbalances between classes. Usually, when there is a large difference between the classes, machine learning models may focus more on the majority class, making it difficult to correctly classify the minority class. For this reason, data set balancing was performed. Subsequently, different feature extraction methods were applied to obtain image features, followed by the implementation of classification processes using classifiers of different structures. In this study, binary classification for wood defect detection was performed using feature extraction and classical machine learning classifiers. The advantages of machine learning in the field of image processing are as follows: It has important features such as coping with complexity, feature extraction ability, flexibility, suitability for large data sets, transfer learning possibility, updatability, and complexity reduction. In this study, machine learning models were trained on a standard processor (CPU) instead of using a GPU. This choice aims to make computational resources more wide-

Table 1 Literature summary
Tablica 1. Sažetak literature

Reference *Navod iz literature*	**Methodology** *Metodologija*	**Techniques** *Tehnike*	**Application** *Primjena*	**Metrics** *Metrika*
Pölzleitner and Schwingshakl (1992)	Manual wood grading	Real Time / Feature vector	Spruce boards	95 % Accuracy
Zhang *et al*. (2015)	PCA and compressed sensing	Principal Component Analysis (PCA), Compressed Sensing	Wood plate defect identification	92 % Accuracy
YongHua and Jin-Cong (2015)	Hybrid method based on texture features	Tamura texture, Grey-Level Co-occurrence Matrix (GLCM)	Detection of dead knots, poles, and living knots	91.83 % Accuracy
Li *et al*. (2017)	LDA and compressed sensor images	Linear Discriminant Analysis (LDA), Compressed Sensor Images	Wood defect detection	94 % Accuracy
Chang *et al*. (2018)	Convex optimization and Otsu segmentation	Convex Optimization, Otsu Segmentation	Comprehensive wood surface defect image evaluation	94.1 % Accuracy
Li *et al*. (2019)	Local Binary Pattern and Local Binary Differential Excitation	Local Binary Pattern, Local Binary Differential Excitation	Classification of cracks and linear mineral lines on birch veneer surface	93 % Recall
Urbonas *et al*. (2019)	Faster R-CNN	Region-Based Convolutional Neural Network (faster R-CNN)	Automatic visual inspection of wood veneer surfaces	96.1 % Accuracy
Shi *et al*. (2020)	Glance Multi-Channel Mask R-CNN	Glance network, Multi-Channel Mask R-CNN	Efficient method for wood veneer defect detection	95.31 % Accuracy
Wu *et al*. (2022)	Feature fusion and Support Vector Machine	Support Vector Machine (SVM)	Wood surface defect detection based on feature fusion	91.26 % Accuracy

spread and accessible. Therefore, computers with high processing power were not required.

In this study, the existing dataset is analyzed and the wood images are classified into two categories: defective or undefect wood. Also, many methods are used in the feature extraction phase, and the performance of the feature extraction methods is presented comparatively. The extracted features are evaluated with different classifiers and a cross-validation technique is used to reduce classifier bias. All the different extracted features are combined and the classifier performances are measured when different features are combined.

The remaining sections of the paper are organized as follows: Section 2 presents the preprocessing steps, feature extraction methods, and classifiers used; Section 3 provides the experimental results and comparative analysis, while Section 4 presents the conclusions and future research directions.

2 MATERIALS AND METHODS
2. MATERIJALI I METODE

The following subsections provide details about the dataset and its features, pre-processing, feature extraction methods and classifiers used in the study.

2.1 Dataset
2.1. Skup podataka

The dataset is taken from Kodytek *et al.* (2021). There are 20,276 wood surface images in the dataset with ten common types of wood defects including various types of knots, cracks, blue stain, resin and heartwood. Figure 1 shows wood defects samples within the dataset.

While 1,992 of these images contain images of cut wood with no defects, 18,284 images contain images of wood with one or more surface defects. On average, each image contains 2.2 defects, with 6.7 % of the images having more than three defects, and with the maximum number of defects in a single image being 16.

The dataset initially had a serious imbalance between undefect and defect images. There were only 1,992 undefect images, while there were 18,284 defect images. In our opinion deep learning models tend to focus predominantly on the larger class in unbalanced data sets. Therefore, we balanced the dataset by increasing the defect class. The number of defect examples was increased to 18,284 with data augmentation techniques (rotation, shift, brightness variation, etc.) so as to become equal to the defect examples. This process is necessary to balance the learning across classes and improve model performance.

In addition, the original images have a high resolution of 2800×1024 pixels. However, training at this resolution increased the computational cost and made it difficult for the models to work efficiently. Therefore, the images were resized to 300×300 pixels. This resizing provided sufficient resolution to train the model faster while preserving the features of the defects in the image. The image resolution was reduced from 2800×1024 to 300×300 pixels in order to reduce the high computational cost and train the model more efficiently. In this process, the defect details were preserved by using bilinear interpolation. As a result, significant improvements were obtained in processing time and resource usage with an acceptable difference in accuracy. Bilinear interpolation is a mathematical method used to resize an image or estimate a set of data at a higher resolution. In this way, the resizing process was performed without any problems with the features in the existing images.

These methods were applied to both solve the imbalance problem and increase the accuracy and overall performance of the model.

In this study, various data augmentation techniques were applied to increase the generalization ability of the model. These augmentation techniques were performed using the Augmentor: An Image Augmentation Library for Machine Learning library developed by Bloice *et al.* (2017). The images were rotated to the right or left with a maximum of 25 degrees with a 70 % probability, thus providing variation at different angles and increasing the model robustness against transformations. In addition, the images were flipped horizontally with a 50 % probability, thus providing symmetrical features to the model and making the model more flexible. The contrast of the images was changed by a random factor between 0.5 and 1.5 with a 50 % probability, which aimed to allow the model to adapt to different lighting conditions. Similarly, the brightness levels were

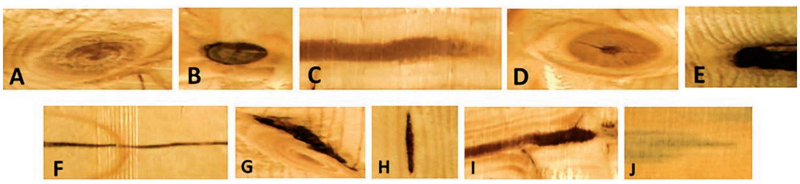

Figure 1 Wood defects within the dataset (A – live knot, B – dead knot, C – quartzity, D – knot with crack, E – knot missing, F – crack, G – overgrown, H – resin, I – marrow and J – blue stain)
Slika 1. Greške drva u skupu podataka (A – zdrava kvrga, B – nesrasla kvrga, C – inkrustacije minerala, D – ispucala kvrga, E – ispadajuća kvrga, F – pukotina, G – obrasla kvrga, H – smolenica, I – srčika, J – plavilo)

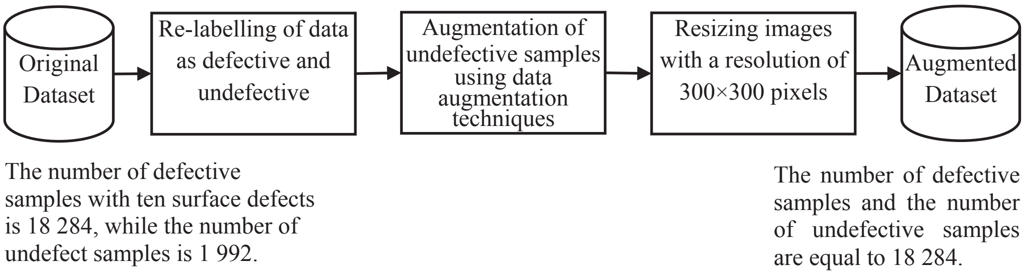

Figure 2 Data augmentation phase of data pre-processing
Slika 2. Faza porasta broja podataka u prethodnoj obradi podataka

randomly adjusted between 0.7 and 1.3 with a 50 % probability, thus increasing the sensitivity of the model to lighting changes in the images. These data augmentation processes allowed the model to be trained with a wider range of images, allowing it to generalize better under various conditions. The data augmentation phase of data pre-processing is shown in Figure 2.

2.2 Feature extraction
2.2. Izdvajanje svojstava

Feature extraction is the process of determining distinctive features from images. The purpose of this process is to represent the image with fewer values. In this way, decision making can be achieved with more meaningful and less dimensional values. Classification performances are directly dependent on good feature extraction. Well-extracted meaningful features increase classification performance. Many feature extraction methods have been proposed in the literature. In image analysis, especially in applications such as wood defect detection, the features used are generally grouped into four main groups: geometric features, statistical features, texture features, and color features (Mutlag *et al*., 2020). Another grouping made according to properties and models includes color based features, texture features, intensity features, human features, finger print features, conceptual features, and text features (Salau and Jain, 2009). In particular, a survey of texture feature extraction methods is also available in the literature (Humeau-Heurtier, 2019). Each has its pros and cons depending on the area in which they are used.

In this study, some well-known texture feature extraction methods are used to detect defective wood. These are Local Binary Pattern (LBP), Histogram of Oriented Gradients (HOG), Gray-Level Co-occurrence Matrix (GLCM), Sobel, Gabor and Multi Block LBP (MB-LBP). LBP (Local Binary Pattern) is a feature extraction method in which pixels are compared to their surrounding neighbouring pixels. Each pixel is compared with its neighbouring pixels to form a binary pattern. This pattern represents the texture and structure information in the image, so it is effective in recognizing small changes and patterns (Ojala *et al*., 2002). HOG is used for object recognition. It splits the image into small cells, calculates gradients in each cell, and then creates histograms of those gradients (Dalal and Triggs, 2005). GLCM captures the relationships between gray level pixels in an image (Haralick *et al*., 1973). Sobel is a filtering method used for edge detection. Sobel operators calculate the gradient of pixel intensity in the image (Duda and Hart, 1973). Gabor filters extract features from images using sinusoidal waves vibrating at different frequencies and in different directions (Daugman, 1985). MB-LBP is an expansion of the traditional LBP. MB-LBP divides an image into blocks of specific sizes. Within each block, pixels are compared with neighboring pixels around them using the LBP (Liu *et al*., 2019).

The selection of texture feature extraction methods to be used in the study is based on factors such as suitability to the problem addressed in this study, the class to which it belongs, computational cost and ease of implementation. LBP and its extension MB-LBP fall into the class of statistical approaches. They combine structural and statistical methods, resulting in improved performance for texture analysis. HOG is in human features group. GLCM is classified under the umbrella of statistical approaches. Sobel is a filtering method. Gabor is classified as transform-based approach. LBP focuses on local binary patterns, while MB-LBP extends this by using multi-bit representations for more detailed feature extraction. Sobel is an edge detection operator that focuses on gradient magnitudes in image processing. Gabor is a texture analysis filter that emphasises texture patterns at different scales and orientations. HOG emphasises the capture of shape and edge information through gradient orientations. GLCM uses information such as homogeneity, contrast, energy and correlation derived from pixel relationships.

In this study, different feature extraction techniques were applied using various image processing methods for wood defect detection. First, Local Binary Pattern (LBP) method was used. In this method, images were converted to grayscale and LBP algorithm was applied. Radius=1 and n_points=8 were specified as parameters and 10 features were extracted for each image.

Another method, Histogram of Oriented Gradients (HOG), performs feature extraction by analyzing edges and orientations in images. For this method, pixels_per_cell=(16, 16) and cells_per_block=(2, 2) parameters are used. The HOG algorithm extracts 378 features from each image, providing a detailed representation of edges and orientations. The Gray Level Co-occurrence Matrix (GLCM) method extracts structural features such as contrast, similarity, homogeneity, energy and correlation in images. The parameters used here include distances=[5] and angles=[0], meaning that only horizontal neighborhood relationships are considered. With this method, 5 features are extracted from each image. Another important feature extraction method is Multiblock LBP (MB-LBP). By applying LBP in blocks, LBP histograms are extracted for each block. The parameters used are radius=2, n_points=4 and block_size=6 and 10 features are obtained for each block. As a result, 10 MB-LBP features are extracted from each image. The Gabor Filter method performs feature extraction by analyzing different frequencies and orientations in the images. With this method, Gabor kernels are created using sigma, theta, lambda and gamma parameters and applied to the image. The mean and standard deviation values are extracted for each Gabor filter, and approximately 36 features are obtained for each image in total. Finally, the Sobel Edge Detection method uses Sobel filters to detect edges in images. In this method, histograms of edge orientations are extracted and 9 features are obtained from each image. This is a feature known as the gradient orientation histogram and provides important information about the orientations of the edges. These methods create a rich feature set for wood defect detection and each of them increases the accuracy of the model by capturing different structural and textural information in the images.

Min-max scaling and standardization are two methods commonly used in data preprocessing. Min-max scaling transforms the values in the data set by compressing them into a certain range, usually between [0, 1]. This method is particularly preferred in distance-based algorithms. Standardization subtracts the data from the mean and divides it by the standard deviation, making the mean of the data 0 and the standard deviation 1. This method provides a more balanced modeling of data at different scales and is generally suitable for algorithms such as linear models and support vector machines. In order to improve the performance of the model used in the study, the data were processed by standardization method, because this method helps the model to obtain more accurate and reliable results.

2.3 Classification algorithms
2.3. Klasifikacijski algoritmi

Random Forest Classifier: Random Forest is an ensemble learning method that combines decision trees. Its advantages include that the model is resistant to overfitting, works well with missing data, and achieves high accuracy. It also has the ability to generalize on datasets with a large number of features. However, the interpretability of the model is difficult and the computational cost can increase on large datasets. In addition, memory and processing power requirements are high (Breiman, 2001).

K Neighbors Classifier: The K-nearest neighbor classifier is a simple and intuitive algorithm. Its advantages include the fast training phase and obtaining intuitive results by performing calculations on the test data. It gives good results especially on small data sets. However, calculations are slower on large data sets, and performance may decrease due to the “curse of dimensionality” effect on high-dimensional data (Cover and Hart, 1967).

Support Vector Machine (SVM): SVM is a powerful algorithm that can effectively classify high-dimensional datasets. Its advantages are its ability to perform nonlinear classification and its robustness against outliers. It also becomes very flexible with the right kernel functions. However, on large data sets the training time can be long and tuning the parameters is complex (Cortes and Vapnik, 1995).

Decision Tree Classifier: Decision trees create models that are understandable and visualizable. They can work with both numerical and categorical data. Their advantages include fast training time and simplicity of the model, while their disadvantages include their tendency to overfit and the large differences that small data changes can produce (Quinlan, 1986).

Naive Bayes: Naive Bayes is an algorithm based on Bayes theorem and provides high accuracy, especially in problems such as text classification. The training time is very fast and gives effective results on small data sets. However, the assumption that all features are independent of each other can affect accuracy, as it is generally not valid in real-world data sets (Lewis, 1998).

Logistic Regression: Logistic regression is a simple and fast algorithm used for binary classification. The outputs of the model are easily interpretable. However, its accuracy may be low in nonlinear relationships and complex data sets. Also, since it is based on linear features only, its ability to generalize is limited (Hosmer *et al.,* 2013).

GradientBoostingClassifier: Gradient Boosting combines weak learners to create a strong model. Its advantages include high accuracy and efficiency. This algorithm achieves effective results when hyperparameter settings are set correctly. Its disadvantages are that the training time is long and the risk of overfitting is high (Friedman, 2001).

XGBClassifier: XGBClassifier is a fast and effective classifier developed by optimizing the Gradient Boosting algorithm. It offers high performance especially on large datasets and complex problems. Fast training time and low memory usage are its advantages. However, there is a risk of overfitting and complex parameter settings (Chen and Guestrin, 2016).

LightGBM: LightGBM is an algorithm with parallel processing capabilities that focuses on large datasets. It offers high efficiency and speed, and works very effectively on large datasets. However, its efficiency may be low on small datasets, and the complexity of the model can sometimes make tuning difficult (Ke *et al.,* 2017).

CatBoost: CatBoost is a gradient boosting library that can work effectively with categorical data. It stands out with its fast training time, low risk of overfitting, and strong performance. However, being a new method compared to other algorithms, it can sometimes lead to limited support and resources (Prokhorenkova, 2018).

AdaBoost: AdaBoost combines weak classifiers to create a strong model. Its advantages include high accuracy, low risk of overfitting, and flexibility. Its disadvantage is that the model may fail if the datasets contain noise and incorrect data (Freund and Schapire, 1997).

MLPClassifier: MLPClassifier is a powerful classifier that uses multilayer artificial neural networks. It works similarly to deep learning methods and can achieve high accuracy on very complex and large datasets. However, its training can take a long time and the model is difficult to interpret. It can also carry the risk of overfitting and its hyperparameter settings are complex (Haykin, 1999).

For the Random Forest model, a total of 500 decision trees were used with n_estimators=500, and the Gini coefficient and splitting criterion were determined by selecting criterion='gini'. The tree depth was not limited with max_depth=None, and the minimum sample numbers for splitting and leaf nodes were set with min_samples_split=2 and min_samples_leaf=1 parameters. max_features='sqrt' was used for feature selection and bootstrap sampling was enabled with bootstrap=True. In the KNN (K-Nearest Neighbors) classifier, a single neighbor was considered with n_neighbors=1, distance-based weighting was done by selecting weights='distance' and the KD tree algorithm was preferred by using algorithm='kd_tree'. In addition, the leaf size was determined as leaf_size=10 and the distance metric as metric='minkowski'.

SVM (Support Vector Machines) model was created with C=1.0 regularization parameter, linear kernel was selected as kernel='linear' and kernel degree was set as degree=4. Also, scaling of kernel function was provided with gamma='scale' parameter. In Decision Tree classifier, Gini coefficient was selected with criterion='gini', random splitting strategy was selected with splitter='random', depth was not limited with max_depth=None. Splitting and leaf node parameters were determined with min_samples_split=2 and min_samples_leaf=1 values; also, minimum decrease value was assigned for splitting with min_impurity_decrease=0.0.

For the Logistic Regression model, L2 regularization was applied with penalty='l2', optimization algorithm was determined as solver='liblinear', and maximum iteration number was determined as max_iter=100. GaussianNB(), which works with Gaussian distribution assumption, was used in Naive Bayes classifier. Gradient Boosting model was configured with parameters n_estimators=500, learning_rate=0.1 and max_depth=10, data subsampling rate was selected as subsample=1.0. XGBoost model was similarly configured with parameters n_estimators=500, learning_rate=0.1, max_depth=10, min_child_weight=1, subsample=1.0, colsample_bytree=1.0 and objective='binary:logistic'.

For LightGBM (LGBMClassifier), n_estimators=500, learning_rate=0.1, max_depth=10, min_child_weight=1, subsample=1.0, colsample_bytree=1.0 and objective='binary' were selected; while n_estimators=500 and learning_rate=0.1 were used in the AdaBoost model. For MLPClassifier (Multi-Layer Perceptron), the hidden layer structure was set as hidden_layer_sizes=(100, 50), the maximum iteration number was set as max_iter=500 and random_state=42 for randomness. All these hyperparameters were selected and optimized in accordance with the dataset used in the study and the nature of the problem.

2.4 Evaluation metrics
2.4. Evaluacijska mjerila

In this article, commonly used Accuracy, Precision, Recall, F1-Score and AUC score metrics are used to evaluate the performance of machine learning classification algorithms. The formulas for these metrics are given in Equations (1-4):

$$F1-Score=\frac{2\cdot Precision\cdot Recall}{Precision+Recall} \tag{1}$$

$$Accuracy=\frac{TP+TN}{TP+FN+FP+TN} \tag{2}$$

$$Precision=\frac{TP}{TP+FP} \tag{3}$$

$$Recall=\frac{TP}{TP+FN} \tag{4}$$

Where TP, TN, FP and FN are true positive, true negative, false positive and false negative, respectively.

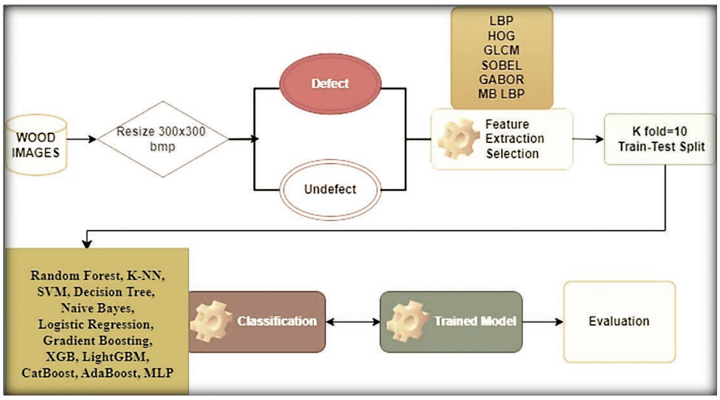

Figure 3 Schematic representation of methodology used in the study
Slika 3. Shematski prikaz metodologije primijenjene u istraživanju

Schematic representation of the methodology used in the study is shown in Figure 3.

3 RESULTS AND DISCUSSION
3. REZULTATI I RASPRAVA

In this section, the performances of the feature extraction and classification algorithms for defect detection are presented and interpreted. For the classification experiments, features are obtained with different feature extraction techniques (LBP, HOG, GLCM, SOBEL, GABOR, MB-LBP). Depending on these features, 12 different classifiers are used in the classification process and their performances are measured. Hyper-parameter settings are made for all classifiers. The same number of classifiers and the same hyperparameters are used for all feature sets. Cross-validation technique is applied to increase the independence of the classification result and reduce its bias. For the cross-validation technique, the k value is determined as 10. 10-fold cross-validation divides the data set into 10 equal parts and uses each part of the model as a test set and the rest as a training set in turn. This process is repeated 10 times and the performance of the model is evaluated by averaging all results. This method provides more generalizable and reliable results. The results in Tables 2-8 are the average results obtaind after cross validation application. The performances of LBP and the classification algorithms in detecting defective wood are shown in Table 2.

According to this table, Random Forest Classifier is the algorithm that gives the best results in terms of performance metrics. This model has high values in all metrics, especially AUC, precision, sensitivity, F1-Score, and its accuracy values are quite high with 96.75 %. This shows that it can successfully classify data using LBP features. The algorithms with the second highest performance are Gradient Boosting Classifier and XGB-Classifier models. Among the classification algorithms on LBP features, LBP Random Forest Classifier has the highest accuracy rate of 96.75 % and this algorithm is highly effective in classifying data accurately. Other algorithms such as LBP Gradient Boosting Classifier 96.56 % and XGB Classifier 96.28 % also present high accuracy values, indicating that the data can be classified successfully. LBP CatBoost 95.39 % also has a very high accuracy rate, but is slightly lower than the others.

Random Forest Classifier has a high accuracy of 88.38 %. CatBoost comes right behind, achieving an accuracy of 86.25 %. Gradient Boosting Classifier,

Table 2 Performances of classification algorithms on LBP features
Tablica 2. Rezultati klasifikacijskih algoritama primijenjenih za LPB svojstva

Algorithms *Algoritmi*	AUC	Precision *Preciznost*	Recall *Opoziv*	F1-Score *F1-rezultat*	Accuracy *Točnost*
LBP RandomForestClassifier	0.9851	0.9680	0.9675	0.9675	0.9675
LBP KNeighborsClassifier	0.9198	0.9217	0.9199	0.9198	0.9199
LBP Support Vector Machine	0.8737	0.8701	0.8388	0.8353	0.8388
LBP DecisionTreeClassifier	0.9048	0.9081	0.9050	0.9047	0.9050
LBP Naive Bayes	0.8484	0.8319	0.8063	0.8025	0.8063
LBP LogisticRegression	0.8595	0.8504	0.8226	0.8190	0.8226
LBP GradientBoostingClassifier	0.9839	0.9661	0.9656	0.9656	0.9656
LBP XGBClassifier	0.9829	0.9631	0.9628	0.9628	0.9628
LBP LightGBM	0.9750	0.9289	0.9262	0.9261	0.9262
LBP CatBoost	0.9816	0.9546	0.9539	0.9539	0.9539
LBP AdaBoost	0.8966	0.8642	0.8398	0.8370	0.8398
LBP MLPClassifier	0.9048	0.8729	0.8458	0.8429	0.8458

Table 3 Performances of classification algorithms on HOG features
Tablica 3. Rezultati klasifikacijskih algoritama primijenjenih za HOG svojstva

Algorithms *Algoritmi*	**AUC**	**Precision** *Preciznost*	**Recall** *Opoziv*	**F1-Score** *F1-rezultat*	**Accuracy** *Točnost*
HOG RandomForestClassifier	0.9198	0.8903	0.8838	0.8834	0.8838
HOG KNeighborsClassifier	0.8232	0.8234	0.8232	0.8232	0.8232
HOG Support Vector Machine	0.8496	0.8451	0.8350	0.8338	0.8350
HOG DecisionTreeClassifier	0.8110	0.8118	0.8111	0.8110	0.8111
HOG Naive Bayes	0.8735	0.8433	0.8342	0.8331	0.8342
HOG LogisticRegression	0.8488	0.8432	0.8338	0.8326	0.8338
HOG GradientBoostingClassifier	0.9138	0.8839	0.8775	0.8770	0.8775
HOG XGBClassifier	0.9124	0.8760	0.8656	0.8646	0.8656
HOG LightGBM	0.9084	0.8726	0.8595	0.8582	0.8595
HOG CatBoost	0.9111	0.8750	0.8625	0.8614	0.8625
HOG AdaBoost	0.8918	0.8696	0.8517	0.8499	0.8517
HOG MLPClassifier	0.8940	0.8764	0.8552	0.8531	0.8552

XGBoost and MLP Classifier also have high accuracy scores. SVM and Naive Bayes achieve good accuracy if other performance metrics are put aside, but are slightly inferior to the best performers. K Neighbors Classifier, Decision Tree Classifier, and LightGBM achieve lower accuracy than other models.

In the experiments conducted on GLCM features, the Random Forest classifier gave the most successful results. The accuracy of Gradient Boosting and Random Forest classifiers is over 93 %. The XGB classifier achieved a value close to the most successful. Logistic Regression classifier showed the most unsuccessful result with an accuracy rate of 82.94 %.

Other algorithms also have high accuracy rates on SOBEL features such as Random Forest Classifier 88.44 %, Gradient Boosting Classifier 87.86 %, and XGB-

Table 4 Performances of classification algorithms on GLCM features
Tablica 4. Rezultati klasifikacijskih algoritama primijenjenih za GLCM svojstva

Algorithms *Algoritmi*	**AUC**	**Precision** *Preciznost*	**Recall** *Opoziv*	**F1-Score** *F1-rezultat*	**Accuracy** *Točnost*
GLCM RandomForestClassifier	0.9653	0.9415	0.9389	0.9388	0.9389
GLCM KNeighborsClassifier	0.8906	0.8918	0.8907	0.8906	0.8907
GLCM Support Vector Machine	0.8866	0.8762	0.8441	0.8407	0.8441
GLCM DecisionTreeClassifier	0.8735	0.8759	0.8736	0.8733	0.8736
GLCM Naive Bayes	0.8831	0.8557	0.8334	0.8362	0.8384
GLCM LogisticRegression	0.8729	0.8510	0.8294	0.8267	0.8294
GLCM GradientBoostingClassifier	0.9609	0.9352	0.9330	0.9330	0.9330
GLCM XGBClassifier	0.9592	0.9250	0.9217	0.9215	0.9217
GLCM LightGBM	0.9499	0.8993	0.8900	0.8894	0.8900
GLCM CatBoost	0.9546	0.9075	0.9006	0.9002	0.9006
GLCM AdaBoost	0.9034	0.8775	0.8523	0.8498	0.8523
GLCM MLPClassifier	0.9044	0.8808	0.8578	0.8557	0.8578

Table 5 Performances of classification algorithms on SOBEL features
Tablica 5. Rezultati klasifikacijskih algoritama primijenjenih za SOBEL svojstva

Algorithms *Algoritmi*	**AUC**	**Precision** *Preciznost*	**Recall** *Opoziv*	**F1-Score** *F1-rezultat*	**Accuracy** *Točnost*
SOBEL RandomForestClassifier	0.9326	0.8899	0.8844	0.8840	0.8844
SOBEL KNeighborsClassifier	0.8303	0.8304	0.8303	0.8303	0.8303
SOBEL Support Vector Machine	0.8209	0.8140	0.7834	0.7730	0.7834
SOBEL DecisionTreeClassifier	0.8078	0.8081	0.8078	0.8078	0.8078
SOBEL Naive Bayes	0.8424	0.8156	0.7776	0.7706	0.7776
SOBEL LogisticRegression	0.7248	0.7172	0.6935	0.6868	0.6935
SOBEL GradientBoostingClassifier	0.9256	0.8838	0.8786	0.8782	0.8786
SOBEL XGBClassifier	0.9264	0.8834	0.8777	0.8773	0.8777
SOBEL LightGBM	0.9226	0.8750	0.8671	0.8664	0.8671
SOBEL CatBoost	0.9271	0.8803	0.8732	0.8726	0.8732
SOBEL AdaBoost	0.8880	0.8502	0.8332	0.8311	0.8332
SOBEL MLPClassifier	0.9104	0.8754	0.8586	0.8571	0.8586

Table 6 Performances of classification algorithms on GABOR features
Tablica 6. Rezultati klasifikacijskih algoritama primijenjenih za GABOR svojstva

Algorithms *Algoritmi*	**AUC**	**Precision** *Preciznost*	**Recall** *Opoziv*	**F1-Score** *F1-rezultat*	**Accuracy** *Točnost*
GABOR RandomForestClassifier	0.9753	0.9521	0.9496	0.9495	0.9496
GABOR KNeighborsClassifier	0.9011	0.9022	0.9012	0.9011	0.9012
GABOR Support Vector Machine	0.9048	0.8861	0.8622	0.8600	0.8622
GABOR DecisionTreeClassifier	0.8866	0.8880	0.8866	0.8865	0.8866
GABOR Naive Bayes	0.8933	0.8766	0.8580	0.8562	0.8580
GABOR LogisticRegression	0.8918	0.8789	0.8586	0.8566	0.8586
GABOR GradientBoostingClassifier	0.9765	0.9543	0.9523	0.9522	0.9523
GABOR XGBClassifier	0.9756	0.9486	0.9460	0.9459	0.9460
GABOR LightGBM	0.9658	0.9216	0.9146	0.9142	0.9146
GABOR CatBoost	0.9704	0.9318	0.9265	0.9262	0.9265
GABOR AdaBoost	0.9157	0.8954	0.8770	0.8756	0.8770
GABOR MLPClassifier	0.9178	0.8989	0.8802	0.8788	0.8802

Classifier 87.77 %. Although Naive Bayes performs slightly lower than the others, with an accuracy rate of 77.76 %, it still has a very acceptable level of accuracy.

In the classification made on Gabor features, Gradient Boosting Classifier and XGBClassifier algorithms stand out by achieving a high accuracy rate of 95.23 % and 94.60 %. At the same time, Random Forest Classifier also shows a very successful result with 94.96 %. Other algorithms also have high accuracy rates, indicating that Gabor features allow effective classification.

In the classification processes performed on MB LBP features, Random Forest Classifier showed the highest performance with 89.58 % in terms of accuracy and also achieved successful results in other metrics. The Gradient Boosting Classifier ranked second with an accuracy of 88.86 %, while XGBClassifier and CatBoost performed well with an accuracy of 87.77 % and 86.03 %, respectively. LightGBM provided a satisfactory result with an accuracy of 84.07 %. In general, Random Forest provided the best results among these classification algorithms.

A "hybrid feature" is a feature or variable that is usually created by combining different types of features or information. This type of feature is usually created by combining information from two or more different sources. By combining all the features extracted in the study, hybrid features are created and their performance is examined.

The classification performances of the classifiers on the feature set obtained by combining all feature extraction methods are given in Table 8.

In the combined features, Gradient Boosting Classifier gives the highest accuracy with 95.16 % accuracy. XGB Classifier gives very successful results with 95.13 %. LightGBM and CatBoost had an accuracy of 94.70 % and 94.71 %, respectively. It has been observed that hybrid learning provides an increase in accuracy according to some feature extractions.

The accuracy values of all feature extraction and classification algorithms are given in Table 9.

A shematic view of Table 9 is also given in Figure 4. As it can be seen in Figure 4, classification algorithms RandomForestClassifier, GradientBoostingClassifier and XGBClassifier stand out in terms of total performance indicators.

The fact that certain methods perform better than others can be attributed to the characteristics of the fea-

Table 7 Performances of classification algorithms on MB-LBP features
Tablica 7. Rezultati klasifikacijskih algoritama primijenjenih za MB-LBP svojstva

Algorithms *Algoritmi*	**AUC**	**Precision** *Preciznost*	**Recall** *Opoziv*	**F1-Score** *F1-rezultat*	**Accuracy** *Točnost*
MB LBP RandomForestClassifier	0.9397	0.8984	0.8958	0.8956	0.8958
MB LBP KNeighborsClassifier	0.8307	0.8321	0.8308	0.8306	0.8308
MB LBP Support Vector Machine	0.7983	0.8025	0.7660	0.7586	0.7660
MB LBP DecisionTreeClassifier	0.8181	0.8201	0.8101	0.8178	0.8181
MB LBP Naive Bayes	0.7988	0.7961	0.7584	0.7504	0.7584
MB LBP LogisticRegression	0.7995	0.7742	0.7558	0.7516	0.7558
MB LBP GradientBoostingClassifier	0.9335	0.8907	0.8886	0.8844	0.8886
MB LBP XGBClassifier	0.9294	0.8802	0.8777	0.8775	0.8777
MB LBP LightGBM	0.9121	0.8486	0.8407	0.8398	0.8407
MB LBP CatBoost	0.9240	0.8638	0.8603	0.8600	0.8603
MB LBP AdaBoost	0.8365	0.7990	0.7673	0.7609	0.7673
MB LBP MLPClassifier	0.8575	0.8098	0.7893	0.7794	0.7839

Table 8 Performances of classification algorithms on hybrid features
Tablica 8. Rezultati klasifikacijskih algoritama primijenjenih na hibridna svojstva

Algorithms *Algoritmi*	**AUC**	**Precision** *Preciznost*	**Recall** *Opoziv*	**F1-Score** *F1-rezultat*	**Accuracy** *Točnost*
RandomForestClassifier	0.9778	0.9420	0.9364	0.9362	0.9364
KNeighborsClassifier	0.8985	0.8986	0.8986	0.8985	0.8986
Support Vector Machine	0.9262	0.9087	0.8896	0.8883	0.8896
DecisionTreeClassifier	0.8755	0.8762	0.8756	0.8755	0.8756
Naive Bayes	0.8822	0.8843	0.8645	0.8627	0.8645
LogisticRegression	0.9036	0.8886	0.8678	0.8660	0.8678
GradientBoostingClassifier	0.9820	0.9545	0.9516	0.9515	0.9516
XGBClassifier	0.9830	0.9538	0.9513	0.9513	0.9513
LightGBM	0.9811	0.9492	0.9470	0.9469	0.9470
CatBoost	0.9800	0.9504	0.9471	0.9470	0.9471
AdaBoost	0.9489	0.9058	0.8927	0.8918	0.8927
MLPClassifier	0.9581	0.9168	0.9072	0.9068	0.9072

Table 9 Accuracy values of all feature extraction and classification algorithms
Tablica 9. Točnost svih algoritama ekstrakcije i klasifikacije svojstava

Algorithms / Algoritmi	***LBP***	***HOG***	***GLCM***	***SOBEL***	***GABOR***	***MB-LBP***	***HYBRID***
RandomForestClassifier	0.9675	0.8838	0.9389	0.8844	0.9496	0.8958	0.9364
KNeighborsClassifier	0.9199	0.8232	0.8907	0.8303	0.9012	0.8308	0.8986
Support Vector Machine	0.8388	0.8350	0.8441	0.7834	0.8622	0.7660	0.8896
DecisionTreeClassifier	0.9050	0.8111	0.8736	0.8098	0.8866	0.8181	0.8756
Naive Bayes	0.8063	0.8342	0.8384	0.7776	0.8580	0.7584	0.8645
LogisticRegression	0.8226	0.8338	0.8294	0.6935	0.8586	0.7558	0.8678
GradientBoostingClassifier	0.9656	0.8775	0.9330	0.8786	0.9523	0.8886	0.9516
XGBClassifier	0.9628	0.8656	0.9217	0.8777	0.9460	0.8777	0.9513
LightGBM	0.9262	0.8595	0.8900	0.8671	0.9146	0.8407	0.9470
CatBoost	0.9539	0.8625	0.9006	0.8732	0.9265	0.8603	0.9471
AdaBoost	0.8398	0.8517	0.8523	0.8332	0.8770	0.7673	0.8927
LMLPClassifier	0.8458	0.8552	0.8578	0.8586	0.8802	0.7879	0.9072

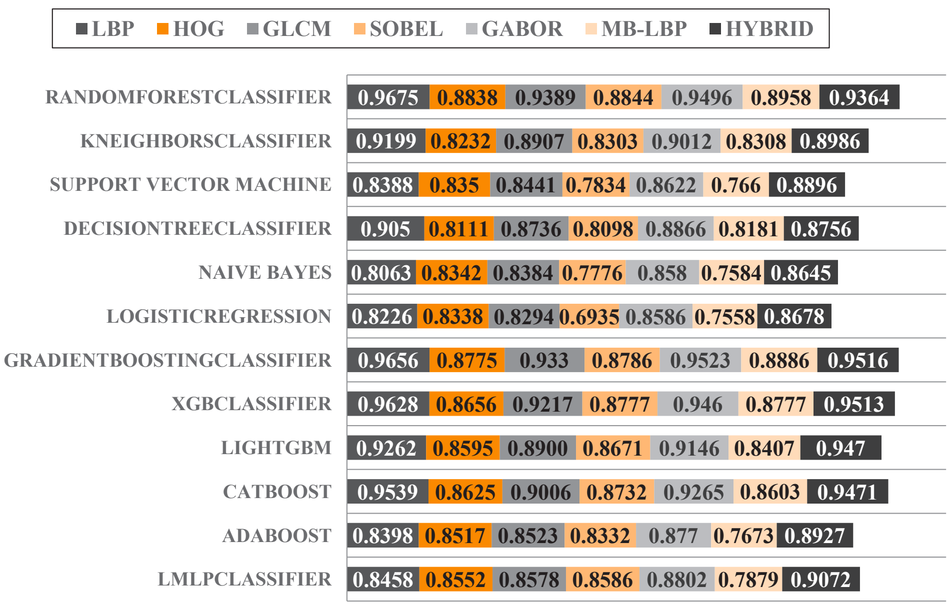

Figure 4 Accuracy values of classification algorithms
Slika 4. Točnost klasifikacijskih algoritama

ture extractors and classifiers used. LBP is very effective in capturing local patterns in the surface texture, which plays a critical role in detecting defects in texture-dense materials such as wood. This feature provides high performance, especially when combined with a powerful classifier such as RandomForestClassifier. On the other hand, HOG is successful in measuring edge density, but may be limited in recognizing

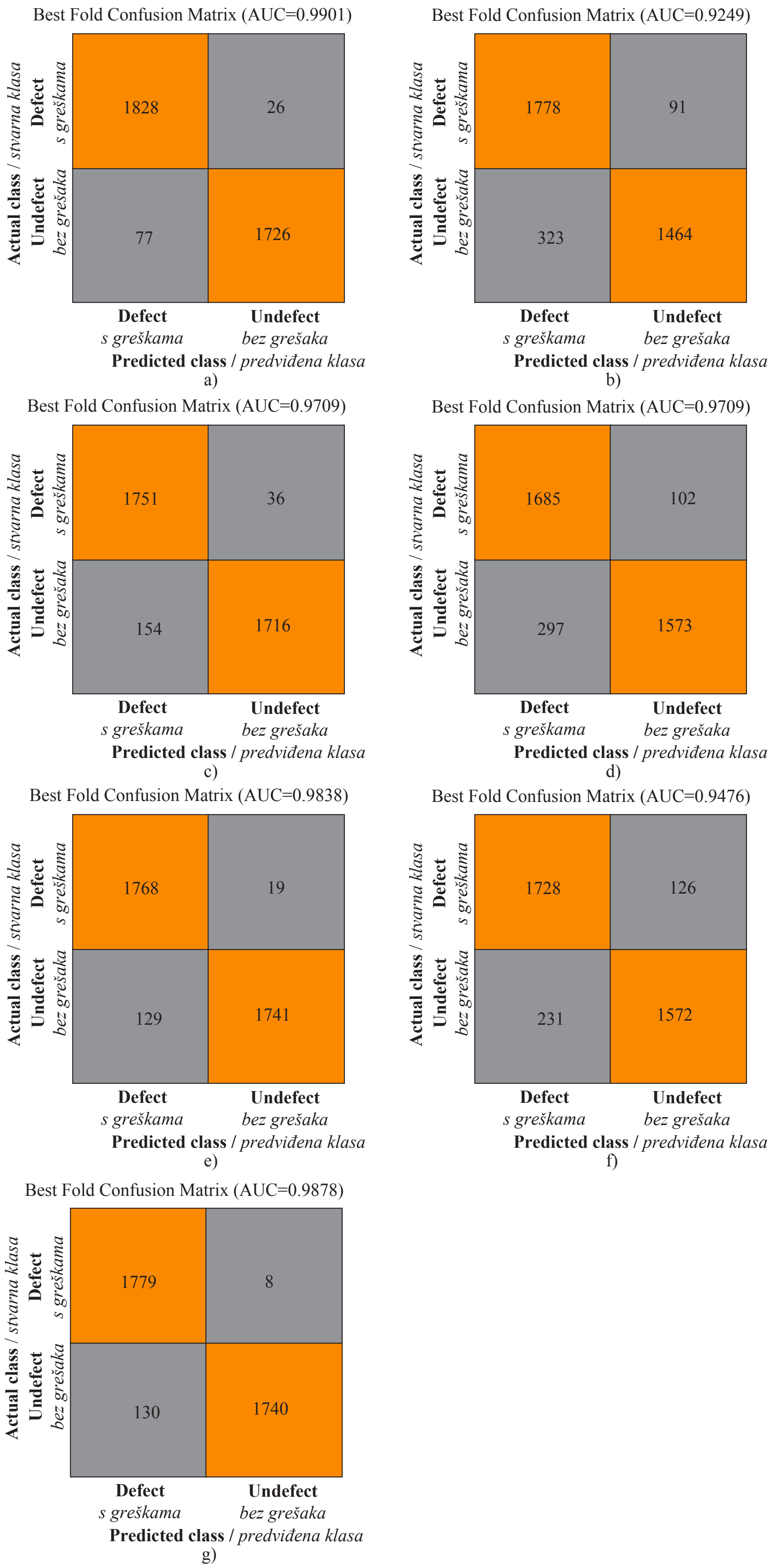

Figure 5 Complexity matrix of the most successful feature extraction and classification algorithms: a) LBP Random Forest Classifier, b) HOG Random Forest Classifier, c) GLCM Random Forest Classifier, d) SOBEL Random Forest Classifier, e) GABOR Gradient Boosting Classifier, f) MB LBP Random Forest Classifier, g) Hybrid Classifier

Slika 5. Matrica složenosti najuspješnijih algoritama ekstrakcije i klasifikacije svojstava: a) LBP Random Forest klasifikator, b) HOG Random Forest klasifikator, c) GLCM Random Forest klasifikator, d) SOBEL Random Forest klasifikator, e) GABOR Gradient Boosting klasifikator, f) MB LBP Random Forest klasifikator, g) hibridni klasifikator finer defects on the wood surface. Similarly, GLCM is effective in describing texture patterns, but may be insufficient when more complex features are needed in high-dimensional datasets. While RandomForestClassifier stands out with its ability to handle high-dimensional data and reduce overfitting thanks to its ensemble of decision trees, other ensemble learning algorithms such as Gradient Boosting, XGBoost, and CatBoost also show high performance for similar reasons. Simpler modeling techniques such as Naive Bayes and Logistic Regression may show low accuracy in more complex datasets, and may be limited in understanding fine details such as wood surface defects. Although hybrid combinations aim to provide a wider information pool by combining multiple feature extractors, in this study, it was observed that hybrid features could not overcome the effect of individual features. This situation reveals that excessive complexity and redundancy of some features can limit performance. As a result, the obtained performance differences can be attributed to the internal properties of both feature extractors and classifiers and can be explained more clearly in this context.

The confusion matrices of the algorithms showing the most successful results for each feature extraction method on the test set are shown in Figure 5.

In this research, six different feature extraction methods are used and the performances of the same 12 different classifiers are analyzed in each feature extraction. In the experiments, it is observed that the LBP method made the most effective feature extraction. Over 90 % accuracy is achieved with tree-based algorithms on the features extracted with the LBP method. Additionally, the most successful performance of the study is achieved with the LBP-Random Forest combination. The worst performances are obtained for all classifiers in the features extracted with MB-LBP, SOBEL and HOG. Even tree-based ensemble classifiers could not reach 90 % accuracy. GABOR and GLCM extracted the most effective features after LBP.

In terms of classifier performance, tree-based classifiers such as Random Forest, Gradient Boosting and XgBoosting have achieved more successful results for all feature extraction methods in detecting defective wood.

The results obtained in this study show remarkable success compared to previous studies in the literature in wood defect detection. Pölzleitner and Schwingshakl (1992) achieved 95 % accuracy using manual wood classification and feature vectors, but manual methods are known to be less scalable compared to automated approaches. Zhang *et al*. (2015) achieved 92 % accuracy using PCA and compressed sensing methods, but the performance may be limited by the inability of PCA to provide sufficiently robust features in complex datasets. YongHua and Jin-Cong (2015) achieved 91.83 % accuracy with texture-based features such as GLCM and Tamura texture, but these methods do not capture as much detailed information as more modern feature extractors.

In a more advanced study, Li *et al*. (2017) achieved 94 % accuracy with LDA and compressed sensor images, while Li *et al*. (2019) achieved 93 % recall with LBP and local differential excitation methods. These methods, although focusing on texture analysis, were limited in distinguishing more complex surface defects. Urbonas *et al*. (2019) achieved 96.1 % accuracy using Faster R-CNN, and Shi *et al*. (2020) achieved 95.31 % accuracy with Multi-Channel Mask R-CNN. Although these deep learning-based methods provide high accuracy, their computational density and larger dataset requirements stand out as a disadvantage.

The combination of LBP and Random Forest used in this study has shown a competitive success with most of the methods in the literature and has reached an accuracy rate of 96.75 %. In particular, the success of LBP in capturing local texture patterns and the ability of the Random Forest algorithm to process high-dimensional data have made this result possible. In addition, the evaluation of hybrid feature combinations has provided an in-depth analysis, unlike other studies in the literature, but it has been observed that these combinations do not contribute to the performance increase. This situation reveals that excessive complexity may not be beneficial in certain cases. In general, the results of this study provide a strong alternative to the existing methods in the literature and provide both high accuracy and efficiency in the field of wood defect detection.

Since wood is a heterogeneous material, not every feature extraction method may give good results. The focus of this study is to find the most suitable feature extraction method to detect wood defects and to increase the binary classification performance with the obtained features. In this context, studies have been conducted to present the most suitable solution by considering the effects of different methods on the classification success.

LBP (Local Binary Pattern) has been used as a successful feature extraction method for wood defect detection. Wood contains textural differences and subtle defects due to its heterogeneous structure. LBP captures subtle textural changes by comparing each pixel with its surrounding neighboring pixels and thus detects subtle defects. The success of the method is due to the effective modeling of distinct texture structures on the wood surface and the ability to correctly identify small defects. The simple and efficient structure of LBP facilitates working with large data sets and increases the efficiency of the model.

4 CONCLUSIONS
4. ZAKLJUČAK

This study aimed to evaluate the performance of different feature extraction methods and classification algorithms for wood defect detection. Using different feature extraction methods (LBP, HOG, GLCM, SOBEL, GABOR and MB-LBP), with various classification algorithms (RandomForestClassifier, KNeighborsClassifier, Support Vector Machine, DecisionTreeClassifier, Naive Bayes, LogisticRegression, GradientBoostingClassifier, XGBClassifier, LightGBM, CatBoost, AdaBoost and MLPClassifier) has been tested for their defect detection capabilities. Based on the experiments carried out, the following conclusions can be drawn from this article: It has been observed that the LBP method extracts the most effective features in detecting defective wood. In addition, based on the classification results, GABOR and GLCM methods achieved very successful feature extraction. The most successful classification result is achieved by the Random Forest algorithm on LBP features. The performances of Gradient Boosting and Xgboosting classifiers have successful accuracy rates following the Random Forest classifier. In all feature sets, tree-based augmented ensemble methods are more successful than other classical machine learning algorithms. Although SOBEL, MB-LBP and HOG features are less effective than other methods, they provided successful results for detecting defective wood. Accuracy performance varies between 75 % and 97 % using all feature extraction methods and classifier combinations. In experiments performed by combining all features, the accuracy rate did not increase compared to the LBP-Random Forest combination result, but increased compared to other combinations. The most successful classification value obtained in the study is found to be 96.75 %.

In conclusion, this study evaluated how different feature extraction methods and classification algorithms perform in wood defect detection applications. These results provide valuable information that can be used in industrial defect detection and similar applications. According to the results obtained from the findings, the LBP-Random Forest model is expected to give successful results when used in industrial applications for the detection of defective woods.

Different feature extraction methods and different classification algorithms may be used in these fields of study in the future. Studies on wood defect detection can be carried out with deep learning methods. In addition, wood defect detection systems can be developed by extracting features with machine deep learning methods and making classifications with machine learning classification algorithms. Other feature extraction methods not used in this study are left to future studies. Studies on classifying wood defect types using this dataset is also left to future studies. In the future, it will be inevitable to ensure quality standards in production without using computer vision and artificial intelligence learning models in wood quality control processes.

There are several suggestions for future work in the field of deep learning. In particular, advanced architectures such as Vision Transformers (ViT) can improve model performance by performing robust feature extraction on large datasets. In addition, super-resolution methods can extract more details from low-resolution images, making it easier to detect small defects. Generative Adversarial Networks (GANs) can improve the learning capacity of the model by providing data augmentation in cases with few labeled data. Techniques such as attention mechanisms can also allow the model to focus on important areas in the image, allowing for more precise detection of small defects in particular.

Acknowledgements – Zahvala

This study represents a part of Kenan Kılıç's doctoral thesis titled "A Computer Vision-Based Machine Learning Approach for Wood Defect Detection" submitted to the Department of Wood Products Industrial Engineering, Institute of Science, Gazi University.

5 REFERENCES
5. LITERATURA

1. Bloice, M. D.; Stocker, C.; Holzinger, A., 2017: Augmentor: An ımage augmentation library for machine learning. Journal of Open Source Software, 2 (19): 432. https://doi.org/doi:10.21105/joss.00432
2. Breiman, L., 2001: Random forests. Machine learning, 45: 5-32. https://doi.org/10.1023/A:1010933404324
3. Chang, Z.; Cao, J.; Zhang, Y., 2018: A novel image segmentation approach for wood plate surface defect classification through convex optimization. Journal of Forestry Research, 29: 1789-1795. https://doi.org/10.1007/s11676-017-0572-7
4. Chen, K.; Cai, N.; Wu, Z.; Xia, H.; Zhou, S.; Wang, H., 2023b: Multi-scale GAN with transformer for surface defect inspection of IC metal packages. Expert Systems with Applications, 212: 118788. https://doi.org/10.1016/j.eswa.2022.118788
5. Chen, T.; Guestrin, C., 2016: Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pp. 785-794. https://doi.org/10.1145/2939672.2939785
6. Chen, Y.; Sun, C.; Ren, Z.; Na, B., 2023a: Review of the current state of application of wood defect recognition technology. BioResources, 18 (1): 2288-2302. https://doi.org/10.15376/biores.18.1.Chen
7. Chen, Z.; Xu, D.; Deng, J.; Chen, Y.; Li, C., 2024: Comparative study on deep-learning-based leather surface defect identification. Measurement Science and Technology, 35 (1): 015402. https://doi.org/10.1088/1361-6501/acfb9f

8. Cortes, C.; Vapnik, V., 1995: Support-vector networks. Machine Learning, 20: 273-297. https://doi.org/10.1007/BF00994018
9. Cover, T.; Hart, P., 1967: Nearest neighbor pattern classification. IEEE transactions on information theory, 13 (1): 21-27. https://doi.org/10.1109/TIT.1967.1053964
10. Dalal, N.; Triggs, B., 2005: Histograms of oriented gradients for human detection. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 1, 886-893. https://doi.org/10.1109/CVPR.2005.177
11. Daugman, J. G., 1985: Uncertainty relation for resolution in space, spatial frequency and orientation optimized by two-dimensional visual cortical filters. Journal of the Optical Society of America A, 2 (7): 1160-1169. https://doi.org/10.1364/JOSAA.2.001160
12. Duda, R. O.; Hart, P. E., 1973: Pattern classification and scene analysis. John Wiley & Sons, New York and London.
13. Freund, Y.; Schapire, R. E., 1997: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55 (1): 119-139. https://doi.org/10.1006/jcss.1997.1504
14. Friedman, J. H., 2001: Greedy function approximation: a gradient boosting machine. Annals of Statistics, 29 (5): 1189-1232. http://www.jstor.org/stable/2699985 (Accessed Jan. 18, 2024).
15. Galata, D. L.; Mészáros, L. A.; Kállai-Szabó, N.; Szabó, E.; Pataki, H.; Marosi G.; Nagy, Z. K., 2021: Applications of machine vision in pharmaceutical technology: A review. European Journal of Pharmaceutical Sciences, 159: 105717. https://doi.org/10.1016/j.ejps.2021.105717
16. Gao, M.; Qi, D.; Mu, H.; Chen, J., 2021: A transfer residual neural network based on resnet-34 for detection of wood knot defects. Forests, 12 (2): 212. https://doi.org/10.3390/f12020212
17. Haralick, R. M.; Shanmugam, K.; Dinstein, I. H., 1973: Textural features for image classification. IEEE Transactions on Systems, Man and Cybernetics (SMC), 3 (6): 610-621. https://doi.org/10.1109/TSMC.1973.4309314
18. Hoang, K.; Wen, W.; Nachimuthu, A.; Jiang, X. L., 1997: Achieving automation in leather surface inspection. Computers in Industry, 34 (1): 43-54. https://doi.org/10.1016/S0166-3615(97)00019-5
19. Hosmer Jr, D. W.; Lemeshow, S.; Sturdivant, R. X., 2013: Applied logistic regression. John Wiley & Sons, Inc. https://doi.org/10.1002/9781118548387
20. Hu, J.; Song, W.; Zhang, W.; Zhao, Y.; Yilmaz, A., 2019: Deep learning for use in lumber classification tasks. Wood Science and Technology, 53: 505-517. https://doi.org/10.1007/s00226-019-01086-z
21. Humeau-Heurtier, A., 2019: Texture Feature Extraction Methods: A Survey. IEEE Access, 7: 8975-9000. https://doi.org/10.1109/ACCESS.2018.2890743
22. Jian, C.; Gao, J.; Ao, Y., 2017: Automatic surface defect detection for mobile phone screen glass based on machine vision. Applied Soft Computing, 52: 348-358. https://doi.org/10.1016/j.asoc.2016.10.030
23. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q & Liu, T.Y., 2017: Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30.
24. Kodytek, P.; Bodzas, A.; Bilik, P., 2021: A large-scale image dataset of wood surface defects for automated vision-based quality control processes. F1000Research, 10: 581. https://doi.org/10.12688/f1000research.52903.2
25. Lei, Q.; Zhong, J.; Wang, C.; Li, X., 2024: Integrating crack causal augmentation framework and dynamic binary threshold for imbalanced crack instance segmentation. Expert Systems with Applications, 240: 122552. https://doi.org/10.1016/j.eswa.2023.122552
26. Lewis, D. D., 1998: Naive (Bayes) at forty: The independence assumption in information retrieval. In: Machine Learning: ECML-98. ECML 1998. Lecture Notes in Computer Science, vol 1398. Springer, Berlin, Heidelberg. https://doi.org/10.1007/BFb0026666
27. Li, C.; Zhang, Y.; Tu, W.; Jun, C.; Liang, H.; Yu, H., 2017: Soft measurement of wood defects based on LDA feature fusion and compressed sensor images. Journal of Forestry Research, 28: 1285-1292. https://doi.org/10.1007/s11676-017-0395-6
28. Li, S.; Li, D.; Yuan, W., 2019: Wood defect classification based on two-dimensional histogram constituted by LBP and local binary differential excitation pattern. IEEE Access, 7: 145829-145842. https://doi.org/10.1109/ACCESS.2019.2945355
29. Liu, G.; Li, F., 2021: Fabric defect detection based on low-rank decomposition with structural constraints. Visual Computer, 38: 639-653. https://doi.org/10.1007/s00371-020-02040-y
30. Liu, G.; Zheng, X., 2020: Fabric defect detection based on information entropy and frequency domain saliency. Visual Computer, 37: 515-528. https://doi.org/10.1007/s00371-020-01820-w
31. Liu, X.; Song, L.; Liu, S.; Zhang, Y., 2021: A review of deep-learning-based medical image segmentation methods. Sustainability, 13: 1224. https://doi.org/10.3390/su13031224
32. Liu, Y.; Xu, K.; Xu, J., 2019: An improved MB-LBP defect recognition approach for the surface of steel plates. Applied Sciences, 9 (20): 4222. https://doi.org/10.3390/app9204222
33. Mu, H.; Qi, D., 2009: Pattern recognition of wood defects types based on Hu invariant moments. In: 2nd International Congress on Image and Signal Processing, Tianjin, China, pp. 1-5. https://doi.org/10.1109/CISP.2009.5303866
34. Mutlag, W. K.; Ali, S. K.; Aydam, Z. M.; Taher, B. H., 2020: Feature extraction methods: A review. Journal of Physics: Conference Series, 1591: 012028. https://doi.org/10.1088/1742-6596/1591/1/012028
35. Norlander, R.; Grahn, J.; Maki, A., 2015: Wooden knot detection using convNet transfer learning. In: Image Analysis, SCIA, Lecture Notes in Computer Science, 9127. Springer, Cham. https://doi.org/10.1007/978-3-319-19665-7_22
36. Ojala, T.; Pietikainen, M.; Maenpaa, T., 2002: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on pattern analysis and machine intelligence, 24 (7): 971-987. https://doi.org/10.1109/TPAMI.2002.1017623
37. Pölzleitner, W.; Schwingshakl, G., 1992: Real-time surface grading of profiled wooden boards. Industrial Metrology, 2 (3-4): 283-298. https://doi.org/10.1016/0921-5956(92)80008-H
38. Prokhorenkova, L.; Gusev, G.; Vorobev, A.; Dorogush, A. V.; Gulin, A., 2018: CatBoost: unbiased boosting with categorical features. Advances in neural information processing systems, 31.
39. Qi, D.; Zhang, P.; Jin, X.; Zhang, X., 2010: Study on wood image edge detection based on Hopfield neural network. IEEE International Conference on Information and

Automation, ICIA 2010. https://doi.org/10.1109/ICINFA.2010.5512014
40. Quinlan, J. R., 1986: Induction of decision trees. Machine Learning, 1: 81-106. https://doi.org/10.1007/BF00116251
41. Raheja, J. L.; Kumar, S.; Chaudhary, A., 2013: Fabric defect detection based on GLCM and Gabor filter: A comparison. Optik, 124 (23): 6469-6474. https://doi.org/10.1016/j.ijleo.2013.05.004
42. Salau, A. O.; Jain, S., 2019: Feature extraction: a survey of the types, techniques, applications. In: International conference on signal processing and communication (ICSC), IEEE, pp. 158-164. https://doi.org/10.1109/ICSC45622.2019.8938371
43. Schmoldt, D. L.; Li, P.; Abbott, A. L., 1997: Machine vision using artificial neural networks with local 3D neighborhoods. Computers and Electronics in Agriculture, 16 (3): 255-271. https://doi.org/10.1016/S0168-1699(97)00002-1
44. Shahrabadi, S.; Castilla, Y.; Guevara, M.; Magalhães, L. G.; Gonzalez, D.; Adão, T., 2022: Defect detection in the textile industry using image-based machine learning methods: a brief review. Journal of Physics: Conference Series, 2224: 012010. https://doi.org/10.1088/1742-6596/2224/1/012010
45. Shi, J.; Li, Z.; Zhu, T.; Wang, D.; Ni, C., 2020: Defect detection of industry wood veneer based on NAS and multi-channel mask R-CNN. Sensors, 20 (16): 4398. https://doi.org/10.3390/s20164398
46. Tao, X.; Zhang, D.; Ma, W.; Liu, X.; Xu, D., 2018: Automatic metallic surface defect detection and recognition with convolutional neural networks. Applied Sciences, 8 (9): 1575. https://doi.org/10.3390/app8091575
47. Tsai, D. M.; Huang, C. K., 2019: Defect detection in electronic surfaces using template-based fourier image reconstruction. IEEE Transactions on Components. Packaging and Manufacturing Technology, 9 (1): 163-172. https://doi.org/10.1109/TCPMT.2018.2873744
48. Urbonas, A.; Raudonis, V.; Maskeliūnas, R.; Damaševičius, R., 2019: Automated identification of wood veneer surface defects using faster region-based convolutional neural network with data augmentation and transfer learning. Applied Sciences, 9 (22): 4898. https://doi.org/10.3390/app9224898
49. Wu, C.; Zou, X.; Yu, Z., 2022: A detection method for wood surface defect based on feature fusion. In: 4th International Conference on Frontiers Technology of Information and Computer (ICFTIC), Qingdao, China, pp. 876-880. https://doi.org/10.1109/ICFTIC57696.2022.10075158
50. Xie, D. Y., 2013: Analysis to situation and countermeasure of wood manufacture industry of our country. Forest Investigation Design, 3: 85-92. https://doi.org/10.3969/j.issn.1673-4505.2013.03.038
51. YongHua, X.; Jin-Cong, W., 2015: A study on the determination of wood surface defects based on texture features. Optics, 126 (19): 2231-2235. https://doi.org/10.1016/j.ijleo.2015.05.101
52. Zhang, T.; Wang, Z.; Li, F.; Zhong, H.; Hu, X.; Zhang, W.; Zhang, D.; Liu, X., 2023: Automatic detection of surface defects based on deep random chains. Expert Systems with Applications, Part A, 229: 120472. https://doi.org/10.1016/j.eswa.2023.120472
53. Zhang, Y.; Xu, C.; Li, C.; Yu, H.; Cao, J., 2015: Wood defect detection method with PCA feature fusion and compressed sensing. Journal of Forestry Research, 26 (3): 745-751. https://doi.org/10.1007/s11676-015-0066-4

Corresponding address:

KENAN KILIÇ

Gazi University, Graduate School of Natural and Applied Sciences, Department of Wood Products Industrial Engineering, Ankara, TURKEY, e-mail: kenan.kilic@bozok.edu.tr