Advances in Artificial Intelligence Research (AAIR)
ISSN 2757-7422
www.dergipark.com/aair/

Vol. 4 (No. 2), pp. 111-116, 2024
doi: 10.54569/aair.1601399
Published online: Dec 30, 2024

Performance Analysis Using CNN for Detecting Wood Knots

Nurşah Baş Uslu [1,*], Mevlüt Ersoy [2]

[1] Süleyman Demirel University, Graduate School of Natural and Applied Sciences, Department of Computer Engineering
[2] Süleyman Demirel University, Engineering Faculty, Department of Computer Engineering

Abstract

This study proposes a Convolutional Neural Network (CNN) model to quickly and accurately detect wood deformations. The performance of the CNN was enhanced by extracting structural deformation features, optimizing training parameters, and improving datasets. Experimental analyses demonstrate that the CNN achieved high accuracy rates and is an effective method for deformation detection. The CNN model was designed to identify various wood deformations. Its layered architecture was optimized to analyze deformations at different scales and levels of detail. Minimal preprocessing was applied to the images used during training, and data augmentation techniques were employed to enhance dataset diversity. The model was trained on a training dataset and tested on a validation dataset. Metrics such as loss function and accuracy were monitored throughout the training process. The CNN achieved an accuracy rate of 99.90% on the training dataset. This study highlights that the CNN model is an effective method for non-destructive detection of wood deformations. The proposed CNN model has potential applications in wood deformation detection and quality control processes.

Keywords: *Wood Deformation; Deep Learning; Convolutional Neural Network; Performance Analysis*

1. Introduction

Trees exhibit diverse characteristics due to their growth in varying natural environments. These characteristics arise from the development of branch-stem junctions throughout the tree's lifecycle. As a result, the structural properties of the tree may differ between the trunk and root wood, which significantly influences the quality of lumber utilized for industrial purposes. Among the key factors determining this quality is the formation of knots within the wood [1]. Knot formation adversely impacts the mechanical strength and performance of wood products intended for industrial applications.

The transformation of knots into sub-products is carried out by evaluating their location, type, size, and quantity within a specified length. However, the process of assessing these attributes to create sub-products imposes additional costs on factories and may lead to the production of non-standard items [2] . Knots are classified into circular, oval, and wing types based on their shapes. Furthermore, they are categorized by size as bird’s eye, small, medium, large, and very large [3].

In lumber factories, various processes can be employed to detect knots. Factories conduct knot removal operations by adhering to standardized rules for quality assessment. These operations involve determining different cutting points to remove knots from the lumber. Current systems are primarily based on the manual marking of knot locations with chalk by workers, followed by processing the lumber on machinery [4]. The involvement of human factors in identifying knot locations during the removal process can lead to errors, which, in turn, result in defective sub-products. Therefore, accurately detecting and identifying knot locations is crucial to minimizing errors and ensuring product quality.

The literature includes numerous studies on the detection of knots in lumber. These studies often employ computer vision systems in conjunction with machine learning algorithms [5-7]. A general review of these works reveals that most of the research has been conducted on static images [8]. Typically, the studies involve identifying features on the wood surface, followed by the use of classification methods to detect defects. Libraries such as YOLO [9], OpenCV [10] and TensorFlow [11] are commonly utilized for defect detection. For classification of the defects, artificial intelligence methods such as SVM [12] , KNN[13], ANN[14], CNN [15] and R-CNN [16] are employed. Results obtained from these studies demonstrate high levels of success in classification tasks.

In the product lifecycle from production to the consumer, identifying defective products holds significant importance for both manufacturers and consumers. Increasingly, the detection of such defects is being carried out by machines rather than humans. This shift is driven by the desire to reduce human labor, ensure a consistent operational structure, and minimize costs while maintaining continuous operation. In defect detection using machines, methods such as structural analysis of the product, shape and type recognition, electrical current-based detection, and pressure-based detection are commonly employed. In addition to these methods, image processing has emerged as another effective approach for defect detection.

*Corresponding author
E-mail address: nursahbas8@gmail.com

Received: 14/Dec/2024; Accepted: 28/Dec/2024.

In this study, texture feature descriptors were used to extract features from wood knot images for the classification of wood knots. The extracted features were learned using a Convolutional Neural Network (CNN) classifier to build classification models, and their performances were compared with statistical models. This paper provides a comparison between texture features and local features, as well as an analysis of the classification performances produced by the constructed models. The depth of the CNN was set to 64 layers. The results showed an accuracy rate exceeding 90%.

2. Material and Method

2.1. Dataset

The experimental dataset consists of 4,000 images obtained using high-resolution color cameras capable of capturing 1,024-line blocks and includes 8 different types of wood surface defects. The images have been resized to a resolution of 2800x1024, forming the dataset[17]. Due to the distinction between first-grade and second-grade wood in the forestry industry, the images have been divided into two classes: knotty and knotless.

Appropriate datasets are required to train or evaluate the performance of the algorithm. However, due to the structural properties of wood, the color of wood knots is generally darker compared to the surrounding wood. In some cases, however, heartwood can be darker than the knots. This situation may cause the neural network to misidentify wood knot defects and negatively affect the correct recognition of knot defects on heartwood during network training. To prevent this issue, images of knot defects on dark-colored heartwood underwent preprocessing steps. Similar images were removed from the dataset. A total of 3,000 wood knot images were used as experimental samples. The wood knot dataset consists of two classes, with 80% used as the training set and 20% as the test set, containing 2,400 training images and 600 test images, respectively (Table 1). Training on the data was performed on a computer with an Intel Xeon E5-1603 v3 processor and an NVIDIA Quadro K2200 4GB graphics card.

Table 1. *Number of Dataset*

Wood Knot State	Training Set	Test Set
Knoted	478	152
Knotless	1922	448

For each image, we created a JPG file representing the semantic map of labeled defects. During the labeling process, knotty regions in the displayed image were manually identified. These images were divided into two categories—knotted and knotless—and training was conducted using these two separate datasets. Within these images, various types and sizes of knots are present, which further contribute to enhancing the training quality. Figure 1 contains sample images from the dataset. These images consist of knot defects of varying sizes and under different lighting conditions.

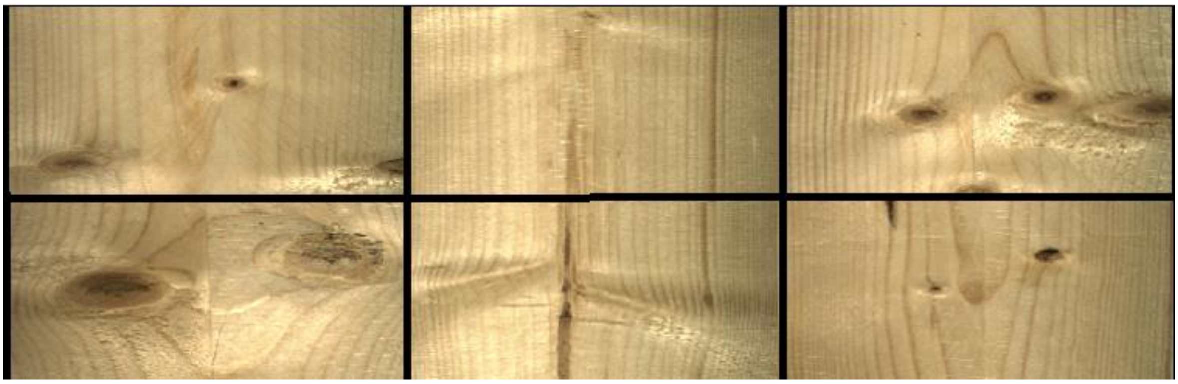

Figure 1. *Different knot images.*

2.2. Convolution Neural Network

The rapid development of computer technology and improvements in hardware performance have led to significant advancements in the field of deep learning. Artificial neural networks are widely used in various fields due to their outstanding success in areas such as image classification and recognition [18]. Convolutional Neural Networks (CNNs) are complex, multi-layered feedforward neural networks with strong fault tolerance and self-learning capabilities. They can effectively handle challenging environmental conditions and complex backgrounds. Their generalization ability is significantly superior to other methods.

CNNs typically consist of an input layer, multiple convolutional layers, pooling layers, fully connected layers, and an output layer. They support both supervised and unsupervised learning and are utilized in many domains such as computer vision and natural language processing. Moreover, CNNs are structures with parallel processing capabilities. By processing image tasks in parallel, CNNs can increase processing speed. Particularly when combined with hardware that has parallel processing capabilities, such as Graphics Processing Units (GPUs), CNN algorithms can process image data quickly and efficiently [19].

Several linked layers and convolutional blocks, such as convolutions, batch normalization, activation, ReLU, pooling, Max pooling, average pooling, completely connected, etc., make up the CNN architecture, as illustrated in Figure 2.

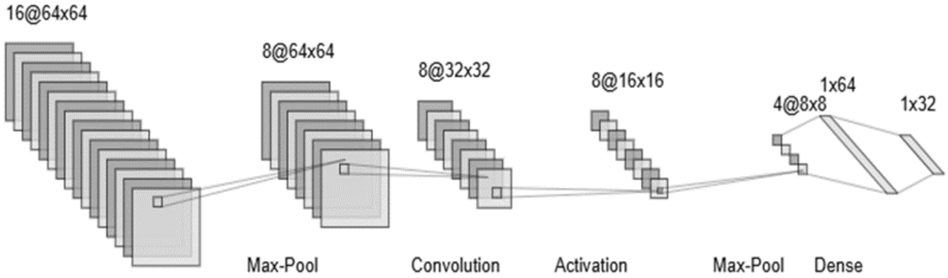

Figure 2. *Established CNN architecture.*

In Table 2, the layers and output dimensions of the CNN architecture are given. The model in Table 2 is proposed for image classification or object detection.

Table 2. *Parameters of CNN* architecture.

Layer Name	Output Size	Layer
Input	64 x 64	16
Max – Pooling_1	64x64	8
Convolution	32x32	8
Activation	32x32	8
Max-Pooling_2	16x16	8
Dense	8x8 1x64 1x32	4

The model starts with 16 channels and processes the input image of size 64x64. The first Max-Pooling layer reduces only the number of channels without affecting the spatial dimensions, concentrating the features. Convolution Layer extracts complex features, reducing spatial dimensions while preserving the channel count. Non-linear activation ensures that the model can learn complex relationships without changing the output dimensions. The second pooling layer further reduces the spatial dimensions for more generalized and lower-dimensional representation. At the end, the feature maps are flattened and passed to fully connected layers. The output progressively reduces to 1x32, likely for classification or regression tasks.

3. Performance Metrics

Various metrics are used to evaluate the performance of machine learning and deep learning algorithms in classification problems. These metrics help compare the accuracy and effectiveness of different models, enabling the selection of the best-performing one. Commonly used evaluation metrics are calculated based on a table known as the "confusion matrix." The confusion matrix is a visual tool that summarizes the performance of a classification model. In this table, the columns represent the predicted classes, while the rows indicate the actual classes. An example of a confusion matrix for a binary classification problem is shown in Table 3.

Table 3. *Two-class confusion matrix*

		PREDICTED CLASS	
		POSITIVE	NEGATIVE
ACTUAL CLASS	POSITIVE	True Positive (TP)	False Negative (FN)
	NEGATIVE	False Positive (FP)	True Negative (TN)

True Positive (TP) refers to the correct prediction of positive examples as positive, meaning the model accurately identifies the positive class. False Negative (FN) refers to the incorrect prediction of positive examples as negative, where the model mistakenly classifies a positive instance as negative. False Positive (FP) refers to the incorrect prediction of negative examples as positive, meaning the model incorrectly classifies a negative instance as positive. True Negative (TN) refers to the correct prediction of negative examples as negative, where the model accurately identifies the negative class. These four terms are crucial for evaluating the performance of classification models and are used in calculating metrics such as accuracy, precision, recall and F1-Score. All the evaluation indices are defined in Table 4.

Table 4. *Performance Metrics.*

Metric	Formula
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$
Precision	$\frac{TP}{TP + FP}$
Recall	$\frac{TP}{TP + FN}$
F1-Score	$\frac{2xPrecisionxRecall}{Precision\ x\ Recall}$

Accuracy represents the proportion of correctly classified instances out of all instances, but it may not be sufficient in cases of class imbalance. Precision measures the proportion of positive predictions that are actually correct, making it crucial when minimizing false positives is important. Recall evaluates how many actual positive instances are correctly identified, being essential when missing positive cases has severe consequences. The F1 Score, as the harmonic mean of precision and recall, provides a balanced metric and is particularly useful in imbalanced classification problems.

4. Results and Analysis

The proposed CNN was implemented and trained on a system equipped with an Intel Xeon E5-1603 v3 2.80GHz (8-core) CPU and 16 GB of RAM. The experimental environment is presented in Table 5.

Table 5. *Experimental environment.*

Hardware Environment		Software Environment	
Memory	16GB	System	Windows 10 Pro
CPU	Intel Xeon E5- 1603 v3 2.80GHz (8 core)	Environment configuration	Python 3.7.3, Keras 2.13.0

The training configuration included a batch size of 64, indicating the number of images processed in each training step. The model using the Adam optimization algorithm, and the cross-entropy loss function was trained for 200 iterations, with a batch size of 64 and learning rate of 1×10^{-2}. The parameter configuration is shown in Table 6. This setup was chosen to balance computational efficiency and model performance. The specified parameters were optimized to achieve effective convergence while minimizing overfitting. These parameters played a crucial role in ensuring the stability and accuracy of the training process.

Table 6. *Training parameters.*

Training Parameters	Values	Definations
Batch Size	64	Number of pictures per training
Learning Rate	1×10^{-2}	Initial learning rate
Epoch	200	Training iteration times

In the Figure3, the obtained loss graph clearly illustrates how the training and validation losses vary with the number of epochs. At the beginning of the training process, the loss value starts at approximately 80% and rapidly decreases as the number of epochs increases, eventually stabilizing below 5%. This indicates that the model progressively learns the patterns in the dataset and reduces its errors over time. Notably, during the first 25 epochs, both training and validation losses show a sharp decline. The training loss drops from around 80% to approximately 10% in a short time, indicating that the model is in the initial phase of learning the fundamental patterns in the dataset and significantly reducing its errors. Similarly, the validation loss also demonstrates a downward trend, suggesting that the model performs well not only on the training data but also on the validation data.

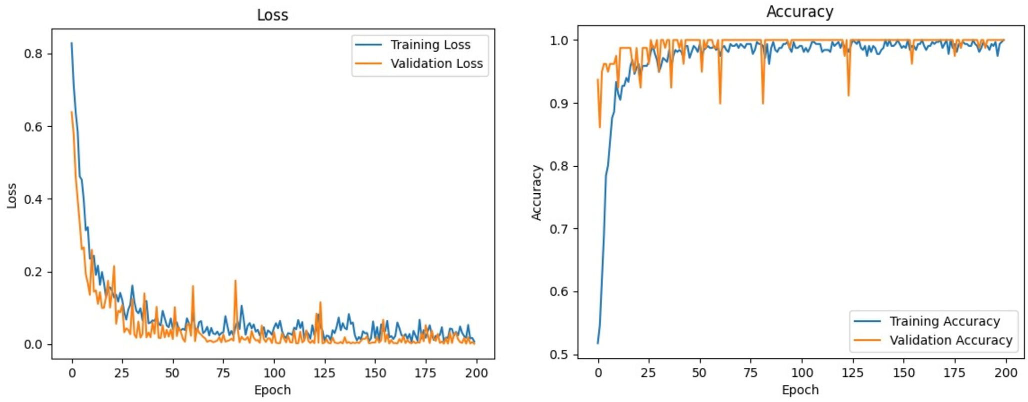

Figure 3. *The model was trained with a training dataset and test datasets: (a) Loss value; (b) Accuracy value.*

As the number of epochs increases, slight fluctuations can be observed in both training and test losses. Although the test loss occasionally exhibits brief spikes, the overall trend remains downward, eventually falling below 5%. These fluctuations indicate that the model encounters challenges with certain examples in the test set. Such increases in test loss could arise due to the diversity of the dataset or limitations in the model’s generalization ability. The fact that the training and test loss values remain close to each other demonstrates that the model does not exhibit overfitting. The simultaneous decrease in both training and validation losses during the training process indicates that the model has effectively learned the patterns in the training data and successfully reflected this knowledge in the test data. Given the absence of signs of overfitting, it can be concluded that the model has strong generalization capabilities.

Table 7. Presents the precision, recall, F1-score, and accuracy of CNN method for classifying wood knot defect images.

Table 7. *The evaluation index values of the network.*

Metrics	Training	Test
Accuracy	0.9810	0.9760
F1 score	0.9812	0.9760
Precision	0.9812	0.9690
Recall	0.9812	0.9710

Upon analyzing the Figure 4, it is evident that instances belonging to the true class "Knotty" are classified as "Knotty" with 100% accuracy, demonstrating the model's exceptional performance in recognizing the "Knotty" class. However, 1.10% of the instances labeled as "Knotty" are misclassified as "Knotless," indicating a minor but noteworthy error in distinguishing certain "Knotty" instances. This observation underscores the existence of a minimal error margin. Furthermore, the graph reveals that no instances of the true class "Knotless" (0%) are incorrectly classified as "Knotty," signifying the model's robustness in avoiding false positive classifications for the "Knotty" class. Lastly, the model achieves a classification accuracy of 98.90% for the "Knotless" instances, highlighting its strong capability to correctly identify examples of the "Knotless" class.

Figure 4. *Wood knots classification results.*

5. Conclusions

In summary, a neural network model, CNN, was proposed to quickly and accurately identify wood knot defects. By extracting structural defect features, optimizing training parameters, and improving datasets and images, the network achieved an accuracy of 99.90%. Experimental results showed that CNN reached a high recognition rate of 99.90% on the training dataset and a low training loss of 1.30% on the validation dataset during the process of identifying 3000 different wood knot defects. The overall accuracy reached 98.60%, and the loss curve and accuracy curve exhibited small fluctuation ranges when CNN was applied to the test dataset.

Moreover, this method does not require extensive image preprocessing or feature extraction when detecting various wood defects and demonstrates high efficiency and recognition accuracy during both the training and testing stages. This indicates that the collected wood knot defects can be accurately and quickly identified using the proposed CNN method. Based on the above analysis, the proposed CNN parameters have potential applications in wood non-destructive testing and wood defect detection.

References

[1] Görgün H V, "Budak tipleri ve değerlendirme farklılıkları," *Artvin Çoruh Üniversitesi Orman Fakültesi Dergisi*, 24(1) (2023) 96-105; doi:10.17474/artvinofd.1177307.

[2] As N, Dündar T, Büyüksarı Ü, "Budakların Odunun Fiziksel ve Mekanik Özellikleri Üzerine Etkileri", *Journal of the Faculty of Forestry Istanbul University, 58(2) (2008) 1-13; https://doi.org/10.17099/jffiu.76055*.

[3] Doğu D, Koç H, As N, Atik C, Aksu B, Erdinler S, "Türkiye'de Yetişen Endüstriyel Öneme Sahip Ağaçların Temel Kimlik Bilgileri ve Kullanıma Yönelik Genel Değerlendirme", *Journal of the Faculty of Forestry Istanbul University,* 51(2) (2014) 69-84; https://doi.org/10.17099/jffiu.33874.

[4] Özkan S, "Kayın (Fagus Orientalis L.) Kerestesinde Eğilme Özelliklerinin Tahribatsiz Yöntemle Tespiti", Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü , Isparta, 2012.

[5] Yılmaz, M, Şahin, H, Yıldız, A, "Sectoral Application Analysis of Studies Made with Deep Learning Models", *Electronic Letters on Science & Engineering*,17(2) (2021) 126-140.

[6] Özgür S. B., "Algoritmalar, Yapay Zeka, Makine Öğrenmesi, Derin Öğrenme ve Uygulamaları: Beşeri Fayda Üretiminin Yazılımlar Tarafından Karşılanması", *Ekonomi ve Yönetim Araştırmaları Dergisi*, 10(1) (2021) 1-29.

[7] Eker R, Alkiş KC, Uçar Z, Aydın A, "Ormancılıkta makine öğrenmesi kullanımı", *Turkish Journal of Forestry | Türkiye Ormancılık Dergisi*, 24(2) (2023), 150-177; doi: 10.18182/tjf.1282768.

[8] Çetiner H, Çetiner İ, "Classification of Cataract Disease with a DenseNet201 Based Deep Learning Model", *Journal of the Institute of Science and Technology,* 12(3) (2022) 1264-1276; doi:10.21597/jist.1098718.

[9] Gurkan, C, Kozalioglu, S, Palandoken, M, "Real Time Mask Detection, Social Distance and Crowd Analysis using Convolutional Neural Networks and YOLO Architecture Designs", *Academic Perspective Procedia*, 4(1) (2021) 195-204, doi: 10.33793/acperpro.04.01.29.

[10] Elkıran, H, "OCC-OPENCV Kütüphanesi için Blok Tabanlı Programlama Aracı,", *Yüksek Lisans Tezi, İstanbul Sabahattin Zaim Üniversitesi Fen Bilimleri Enstitüsü*, 2020.

[11] Shukla N, Fricklas K, "Machıne Learnig with Tensorflow", (2nd Ed.), Manning, USA,2018.

[12] Primandani Arsi and Retno Waluyo, "Analisis Sentimen Wacana Pemindahan Ibu Kota Indonesia Menggunakan Algoritma Support Vector Machine (SVM)", *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, 8(1) (2021) 147-156; doi: 10.25126/jtiik.202183944.

[13] Baita, A, Yoga P, Cahyono N "Analisis Sentimen Mengenai Vaksin Sinovac Menggunakan Algoritma Support Vector Machine (SVM) dan K-Nearest Neighbor (KNN)", *Information System Journel (INFOS)*, 4(2) (2021) 42-46; https://doi.org/10.24076/infosjournal.2021v4i2.687.

[14] Çavuşlu MA, Becerikli Y, Karakuzu C, "Levenberg-Marquardt Algoritması ile YSA Eğitiminin Donanımsal Gerçeklenmesi Hardware Implementation of Neural Network Training with Levenberg-Marquardt Algorithm," *Türkiye Bilişim Vakfı Bilgisayar Bilimleri Ve Mühendisliği Dergisi*, 5(1) (2016) 1.

[15] Kayalı N Z, Omurca İS, "Konvolüsyonel Sinir Ağları (CNN) ile Çin Sayı Örüntülerinin Sınıflandırması," *Journal of Computer Science*,Sep. IDAP 2021(1) (2021) 184 – 191; doi: 10.53070/bbd.989668.

[16] Aalami N, "Hierarchical Convolutional Neural Networks for Fashion Image Classification", *Expert Systems with Applications*, 116(1) (2019) 328-339; doi: 10.1016/j.eswa.2018.09.022.

[17] Samtaş G, Gülesin M, "Sayısal Görüntü İşleme ve Farklı Alanlardaki Uygulamaları", *Electronic Journal of Vocatinal Collages*, 2(1) (2011) 85 - 97.

[18] Ide H, Kurita T, "Improvement of Learning for CNN with ReLU Activation by Sparse Regularization," in *Proceedings of the International Joint Conference on Neural Networks,* Anchorage, AK, USA, 2017, 2684-2691; doi: 10.1109/IJCNN.2017.7966185.

[19] Cengil E, Çınar A, "A New Approach for Image Classification: Convolutional Neural Network," *European Journal of Technic EJT*, 6(2) (2016) 96 - 103.