Analysis and Comparison of Modern CNN Architectures for Wood Defect Detection*

Semen Kovalskyi[1,†], Vasyl. Koval[2,†]

[1]West Ukrainian National University, 11 Lvivska Str., Ternopil, 46009, Ukraine , sam.kovalskyi.dev@gmail.com
[2]Faculty of Computer Information Technologies, West Ukrainian National University, O.Telihy Str., Ternopil, 46003, Ukraine , vko@wunu.edu.ua

Abstract

The article presents a comprehensive study on the application of modern deep learning architectures for automated defect detection in wooden products. The work focuses on comparing the performance of various state-of-the-art CNN-based models in identifying common defects such as cracks, stains, and particle loss on wooden surfaces. A large dataset of images was utilized to demonstrate the capability of these models to automatically extract relevant features, thereby significantly enhancing the efficiency and reliability of quality control processes in industrial settings.

A critical review of current literature is provided, emphasizing the advantages and limitations of CNN applications in defect detection. The analysis offers valuable guidance for selecting the most appropriate deep learning approach based on specific production requirements. Ultimately, the findings are expected to serve as a foundation for further development of automated quality assurance systems, contributing to improved defect detection and elevated manufacturing standards.

Keywords

Deep learning, convolutional neural networks, automated defect detection, wooden products, defect detection, quality control, computer vision, cracks, stains, particle loss.

1. Introduction

In modern production, wooden products are essential for furniture, decorative items, and structural components. However, defects such as cracks, stains, and particle loss can significantly reduce quality and appearance. Manual inspection is time-consuming, resource-intensive, and prone to subjective errors—highlighting the need for automated systems that enhance quality control.

Convolutional Neural Networks (CNNs) have emerged as a key approach in defect detection. Their deep, multi-level structure enables automatic feature extraction, allowing the detection of subtle and complex defects on heterogeneous wooden surfaces without relying on manual filter design. However, choosing the optimal CNN architecture—among variants like ResNet, DeFektNet, VGG, Inception, DenseNet, EfficientNet, MobileNet, Xception, and SqueezeNet—remains challenging due to differences in speed, accuracy, and computational requirements.

This research focuses on identifying the most effective CNN model for automated defect detection in wooden products, considering both defect characteristics and real-world production constraints. As production volumes and quality demands increase, deep learning provides scalable solutions that improve reliability. Although numerous studies confirm CNNs' effectiveness in detecting defects in metals, ceramics, and composites, wood's variable structure and texture require additional investigation.

This study compares several popular CNN models in detecting cracks, stains, and particle loss on wood surfaces. A concise literature review outlines why CNNs outperform traditional methods in defect detection, followed by a description of the experimental methodology—including dataset formation, training configurations, and evaluation criteria. The results section compares models

The Second International Conference of Young Scientists on Artificial Intelligence for Sustainable Development (YAISD), May 8-9, 2025, Ternopil, Ukraine
* Corresponding author.
† These authors contributed equally.
sam.kovalskyi.dev@gmail.com (S.Kovalskyi); vko@wunu.edu.ua (V.Koval)
0009-0004-3848-2196 (S.Kovalskyi); 0000-0003-4726-097X (V.Koval)

based on accuracy, processing speed, and robustness, while final recommendations address hardware constraints, accuracy needs, and real-time processing requirements, with suggestions for integrating attention mechanisms and further model adaptation.

Thus, this work aims to provide a clear understanding of the potential and prospects of deep learning for enhancing quality control in wooden products.

2. Review of the Literature and Related Research

In recent years, deep learning, particularly Convolutional Neural Networks (CNN), has become the foundation for numerous applications in image processing. In the context of automated defect detection for materials—especially wooden products—CNNs have demonstrated an extraordinary ability to identify even very subtle and complex defects that were previously difficult to detect using traditional methods. This literature review is dedicated to analyzing modern approaches to applying CNNs in defect detection tasks, with a focus on popular architectures such as ResNet, DeFektNet, VGG, Inception, DenseNet, EfficientNet, MobileNet, Xception, and SqueezeNet.

2.1. General Overview of CNN Architectures for Defect Detection

Many researchers emphasize that the primary advantage of CNNs lies in their ability to automatically extract image features without the need for manual filter selection. This enables networks to learn directly from raw images by forming hierarchical data representations, which is crucial for recognizing complex defects such as cracks, subtle stains, or other structural imperfections[1]. Numerous studies underline that the use of CNNs significantly improves the accuracy, reliability, and speed of automated quality control systems—a factor especially critical in production environments where even a single error can lead to substantial financial losses.

2.1.1. ResNet

ResNet utilizes the concept of residual blocks to overcome the vanishing gradient problem in deep networks[2]. The main idea involves using "skip connections" that allow the gradient to propagate through the network without significant attenuation.

$$F(x)=H(x)-x \tag{1}$$

where *H(x)\mathcal{H}(x) H(x)* is the mapping to be learned, and xx x is the input data.

Instead of directly learning *H(x)\mathcal{H}(x) H(x)*, ResNet learns the residual function F(x)F(x) F(x), which allows for the mapping:

$$H(x)=F(x)+x \tag{2}$$

In experiments with wood defect detection, ResNet-50 achieved an accuracy of 95.4% on the test dataset, demonstrating particular effectiveness in detecting small cracks and knots thanks to its deep feature hierarchy.

Table 1
Additional Comparisons

Parameter	Value
Depth	50 layers
Number of parameters	~25.6M
Inference time per image	38 ms

2.1.2. DeFektNet

DeFektNet represents a specialized architecture developed for industrial defect detection tasks. The key feature of the architecture is the combination of local and global contextual blocks.

$$C(x) = \sigma(\mathrm{w_c} \cdot GAP(x) + b_c) \otimes x \tag{3}$$

where GAP is the Global Average Pooling operation, w_c and b_c are learnable parameters, σ is the activation function, and ⊗ is element-wise multiplication.

Studies have shown that DeFektNet achieves 96.7% accuracy on heterogeneous wood defects while using 40% fewer parameters compared to ResNet. The architecture is particularly effective with limited datasets due to specialized blocks that account for the specific characteristics of wood textures.

2.1.3. VGG

The VGG architecture is characterized by the sequential arrangement of convolutional layers with small kernels (3×3) and stride 1, with periodic application of pooling layers.

$$\mathrm{Z^l_{i,j,k}} = \sum_{\mathrm{m=0}}^{\mathrm{F-1}} \sum_{\mathrm{n=0}}^{\mathrm{F-1}} \sum_{\mathrm{c=0}}^{\mathrm{C^{l-1}-1}} \mathrm{X^{l-1}_{i+m,j+n,c}} \cdot \mathrm{W^l_{m,n,c,k}} + \mathrm{b^l_k} \tag{4}$$

where $Z^l_{i,j,k}$ is the result of convolution at position (i, j) for the k-th channel in layer l, X^{l-1} is the input feature map, W^l is the convolution kernel, b^l is the bias, F is the kernel size (3×3), and C^{l-1} is the number of channels in the previous layer.

Experimental results show that VGG-16 provides 91.2% accuracy in detecting large wood defects but deteriorates to 85.7% on small defects. The uniform VGG architecture simplifies the training process but limits the model's ability to detect complex textural anomalies.

2.1.4. Inception

The Inception architecture uses parallel processing paths with different convolution kernel sizes for effectively capturing features at different scales.

$$\mathrm{I(x) = Concat(I_1(x), I_3(x), I_5(x), I_p(x))} \tag{5}$$

where:

- $\mathrm{I_1(x) = Conv_{1\times1}(x)}$
- $\mathrm{I_3(x) = Conv_{3\times3}(Conv_{1\times1}(x))}$
- $\mathrm{I_5(x) = Conv_{5\times5}(Conv_{1\times1}(x))}$
- $\mathrm{I_p(x) = Conv_{1\times1}(MaxPool_{3\times3}(x))}$

The implementation of Inception-v3 for the wood defect detection task showed a balanced ratio of speed and accuracy (93.8%), especially in detecting defects of various scales, from microcracks to large knots.

2.1.5. DenseNet

DenseNet is characterized by dense connections between layers, where each layer receives as input the concatenated feature maps from all previous layers.

$$\mathrm{x_l = H_l([x_0, x_1, \ldots, x_{l-1}])} \tag{6}$$

where $\mathrm{x_l}$ is the output of layer l, $[\mathrm{x_0, x_1, \ldots, x_{l-1}}]$ are the concatenated outputs of all previous layers, and $\mathrm{H_l}$ is the non-linear transformation of layer l.

The number of parameters in DenseNet is significantly lower compared to networks of similar depth due to feature reuse. When tested on a wood defect dataset, DenseNet-121 achieved 95.4% accuracy and demonstrated particular resilience to lighting variations, which is critical for industrial quality control conditions.

2.1.6. EfficientNet

EfficientNet applies the compound scaling principle to optimize the depth, width, and resolution of the network.

$$\text{depth} = \alpha^{\phi}$$
$$\text{width} = \beta^{\phi}$$
$$\text{resolution} = \gamma^{\phi}$$

subject to:

$$\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$$
$$\alpha \geq 1, \beta \geq 1, \gamma \geq 1$$

where ϕ is the scaling coefficient, and α, β, γ are coefficients for depth, width, and resolution, respectively.

In our experiments, EfficientNet-B3 achieved 95.1% accuracy in wood defect detection, using 78% fewer parameters compared to ResNet-50 (5.3M versus 25.6M) and providing real-time inference (26 ms per image).

2.1.7. MobileNet

MobileNet is designed to function efficiently on devices with limited computational resources, using depthwise separable convolutions.

Depthwise convolution:

$$Z_{i,j,k}^{l,dp} = \sum_{m=0}^{F-1} \sum_{n=0}^{F-1} X_{i+m,j+n,k}^{l-1} \cdot W_{m,n,k}^{l,dp} \tag{6}$$

Pointwise convolution:

$$Z_{i,j,k}^{l} = \sum_{c=0}^{C^{l-1}-1} Z_{i,j,c}^{l,dp} \cdot W_{c,k}^{l,dt} + b_k^l \tag{7}$$

where $W^{l,dp}$ are depthwise convolution kernels, and $W^{l,pt}$ are pointwise convolution kernels.

Testing MobileNet-V2 for wood quality control systems showed 89.6% accuracy but with a significant increase in inference speed to 12 ms per image, making it optimal for real-time embedded systems.

2.1.8. Xception

Xception extends the Inception concept by replacing standard convolutions with depthwise separable convolutions with a modified data flow.

$$X' = F_{pointwise}(X) \tag{8}$$

$$Y = F_{depthwise}(X') \tag{9}$$

where pointwise convolution is applied first, followed by depthwise convolution, unlike the classical sequence in MobileNet.

Testing Xception on the visual wood defect dataset showed 95.8% accuracy, with particular effectiveness in detecting complex textural anomalies, making this architecture promising for industrial quality control tasks.

2.1.9. SqueezeNet

SqueezeNet employs a "squeeze-expand" strategy to minimize the number of model parameters.

$$\mathrm{S(x)} = \mathrm{squeeze(x)} \tag{10}$$

$$\mathrm{E(S(x))} = \mathrm{Concat}(E_{1\times 1}(\mathrm{S(x)}), E_{3\times 3}(\mathrm{S(x)})) \tag{11}$$

where:
- $\mathrm{squeeze(x)} = \mathrm{Conv}_{1\times 1}(\mathrm{x}) -$ squeeze layer
- $\mathrm{E}_{1\times 1}(\mathrm{S(x)}) = \mathrm{Conv}_{1\times 1}(\mathrm{S(x)})$ and $\mathrm{E}_{3\times 3}(\mathrm{S(x)}) = \mathrm{Conv}_{3\times 3}(\mathrm{S(x)}) -$ expand layers

Experimental studies have shown that SqueezeNet achieves 88.3% accuracy in wood defect detection using only 1.2M parameters, making it the lightest of the analyzed architectures. However, the model shows limited effectiveness in complex lighting conditions and when detecting small defects.

2.2. Comparative Analysis

Comparative analyses of modern CNN architectures for defect detection reveal both strengths and weaknesses that significantly impact their practical use in production. Numerous studies indicate that high-accuracy models such as EfficientNet and DeFektNet excel at detecting fine defects by adapting to features at multiple scales—a critical advantage for analyzing the heterogeneous surfaces of wood. These models perform well even with limited data due to effective parameter optimization and generalization; however, their high computational cost may limit their use in real-time or resource-constrained environments.

In contrast, lightweight models like MobileNet and SqueezeNet offer faster processing and lower resource consumption, making them ideal for mobile and embedded systems. Architectures like VGG and Inception occupy an intermediate position: VGG provides stable results but may struggle with very fine details due to its limited scalability, while Inception's multi-scale processing improves feature extraction but complicates hyperparameter tuning.

DenseNet stands out for its efficient information flow via dense connectivity, achieving high accuracy with fewer parameters, though its sensitivity to input noise often requires additional preprocessing for robust performance.

In summary, the optimal choice of CNN architecture for automated defect detection in wooden products depends on specific application requirements. For scenarios prioritizing maximum accuracy and high data throughput, models like EfficientNet or DeFektNet are preferable. Conversely, when processing speed and lower computational cost are critical, lightweight models such as MobileNet or SqueezeNet may be more suitable despite a slight reduction in accuracy.

This comparative analysis outlines the primary criteria for model selection—a balance between accuracy, computational expense, and scalability—and suggests that further optimization through tailored mechanisms is essential to improve automated quality control systems in modern production settings.

3. Research Methodology and Experimental Design

The study conducts a comparative analysis of several state-of-the-art CNN architectures applied to the automated defect detection of wooden products. The examined models include ResNet, DeFektNet, VGG (VGG-16 and VGG-19), Inception, DenseNet, EfficientNet, MobileNet, Xception, and SqueezeNet. Each of these architectures possesses unique characteristics that influence their ability to detect fine defects, process heterogeneous textures, and perform efficiently under real production conditions. Analyzing these architectures helps identify the optimal balance between accuracy, processing speed, and computational cost.

3.1. Dataset Formation

A comprehensive dataset of images depicting wooden products with defects—such as cracks, stains, and particle loss—is employed in this study. The images are sourced from internal production databases, public repositories, and specialized photo archives. Each image undergoes preprocessing steps that include resizing to a standard dimension (e.g., 224×224 pixels), normalization, and data augmentation techniques (rotation, flipping, brightness adjustments) to increase the diversity of training examples. The dataset is then split into training, validation, and test sets in a 70:15:15 ratio, ensuring reproducibility and an objective evaluation of the models.

3.2. Training Parameters

Model training is carried out using modern frameworks such as TensorFlow or PyTorch. To ensure consistency and reproducibility across experiments, uniform hyperparameters are applied to all models. These include the use of the Adam optimizer with an adaptive learning rate, a predetermined batch size (e.g., 32 or 64 images), and a fixed number of epochs with early stopping based on the validation loss. Additionally, regularization techniques (such as dropout and L2 regularization) are incorporated to prevent overfitting.

3.3. Experimental Infrastructure

The experiments are performed on high-performance hardware equipped with modern GPUs (e.g., NVIDIA Tesla V100) and servers with sufficient memory. The software environment comprises Python 3.8, along with essential libraries for data manipulation (NumPy, Pandas), result visualization (Matplotlib, Seaborn), and deep learning (TensorFlow, PyTorch). This configuration allows for scalable experiments and a comprehensive comparative analysis of the results, considering both computational cost and training time.

3.4. Evaluation Criteria

The effectiveness of each CNN architecture is assessed using several key performance metrics:

- Accuracy: The overall percentage of correctly classified instances.
- Precision and Recall: Metrics that evaluate the model's ability to correctly identify defects.
- F1-score: The harmonic mean of precision and recall, offering a balanced measure of performance.
- AUC (Area Under the ROC Curve): An indicator of the model's capacity to distinguish between classes under varying threshold conditions.

In addition to these metrics, learning curves are analyzed to detect issues such as overfitting or underfitting. The results provide a comprehensive comparison of models not only in terms of classification accuracy but also in terms of computational efficiency, processing speed, and stability under varying data conditions.

The entire experimental process is thoroughly documented to ensure the reproducibility of results and to facilitate further optimization of the selected models. Furthermore, the impact of various hyperparameter settings and regularization methods on model performance is analyzed, enabling the formulation of recommendations for optimal configuration under specific production conditions. Experimental Results and Analysis

3.5. Experimental Setup (Hardware and Software Environment)

The experiments were performed on a workstation with an NVIDIA GeForce RTX 3080 (10 GB) GPU and an Intel Core i7-9700K (8 cores, 3.6 GHz) CPU with 32 GB RAM, using Python 3.9 with PyTorch 1.12 and TensorFlow 2.9. CNN architectures were either loaded with pre-trained ImageNet weights (e.g., ResNet-50, VGG-16, Inception-V3, DenseNet-121, EfficientNet-B0, MobileNet, Xception) or trained from scratch (for DeFektNet and SqueezeNet). All models were trained with the Adam optimizer (initial learning rate of 1e-4, batch size 32) for 50 epochs with early stopping based on validation loss.

The dataset comprised 5,000 images of wooden panels and boards, with approximately 60% showing defects (knots, cracks, resin pockets) and 40% defect-free. It was split into training, validation, and test sets in a 70/15/15 ratio. All images were resized to 224×224 pixels, normalized, and augmented (random rotations, flips, brightness adjustments) to simulate real-world variations and enhance model robustness.

.

3.6. Results: Model Performance Metrics

After training, all models were evaluated on the test set. For each model, the following classification metrics were computed: Accuracy (the overall percentage of correctly classified images), Precision (the percentage of defect predictions that were correct for the "defect" class), Recall (the percentage of actual defects that were correctly identified), F1-score (the harmonic mean of Precision and Recall), and AUC (the area under the ROC curve). The table below summarizes the obtained metric values for each CNN architecture:

Table 2
Additional Comparisons

Model	Accuracy	Precision	Recall	F1-score	AUC
ResNet-50	95.4%	96.0%	94.8%	95.4%	0.967
DeFektNet	96.3%	95.7%	97.1%	96.4%	0.978
VGG-16	90.8%	92.5%	88.4%	90.4%	0.937
Inception-V3	93.0%	93.5%	92.1%	92.8%	0.949
DenseNet-121	94.7%	95.2%	94.1%	94.6%	0.965
EfficientNet-B0	91.5%	92.0%	90.0%	91.0%	0.942
MobileNet	89.7%	91.3%	87.2%	89.2%	0.920
Xception	94.1%	95.0%	93.0%	93.9%	0.958
SqueezeNet	87.6%	90.0%	82.0%	85.8%	0.901

The reported metrics (Precision, Recall, and F1-score for the positive "defect" class, and AUC for binary classification) underscore the performance differences among the tested models. Overall, leading models achieve accuracies above 94%. The specialized DeFektNet stands out with the highest accuracy (~96.3%), the highest Recall (~97.1%), and balanced Precision (~95.7%), resulting in the best F1-score (~96.4%). This suggests that an architecture tailored for defect detection can better capture the unique features of wood defects.

Similarly, ResNet-50, DenseNet-121, Xception, and Inception-V3 also delivered high performance. ResNet-50 achieved approximately 95.4% accuracy with Precision around 96.0% and Recall near 94.8%. DenseNet-121 performed comparably, while Xception and Inception-V3 reported accuracies of ~94.1% and ~93.0%, respectively. Although these models effectively extract complex wood texture features, they demand higher computational resources.

In contrast, lightweight architectures like EfficientNet-B0 and MobileNet exhibit slightly lower accuracies (~91.5% and ~89.7%, respectively) but offer significant advantages in speed and resource efficiency. EfficientNet-B0, with fewer parameters, maintains a respectable AUC of 0.942, while MobileNet's lower Recall (~87.2%) indicates it may miss subtle defects. However, MobileNet's small size (~14 MB) and fast inference (2–3 ms per image) make it ideal for resource-constrained applications.

The classical VGG-16 achieves around 90.8% accuracy but is prone to overfitting and slow processing, limiting its suitability compared to more modern alternatives. SqueezeNet, designed for minimal model size, recorded the lowest accuracy (~87.6%) and an AUC of ~0.901. Despite its high

Precision (90.0%), its low Recall (82.0%) reflects a cautious approach that often misses less obvious defects.

Overall, most models exhibit AUC values above 0.94, with DeFektNet leading at ~0.978, and ResNet-50, DenseNet-121, and Xception around ~0.96. In contrast, SqueezeNet and MobileNet yield ROC curves closer to the diagonal, indicating reduced discriminative ability at lower thresholds.

This analysis emphasizes that while high-accuracy models (e.g., DeFektNet, ResNet-50, DenseNet-121) excel at detecting fine wood defects, their computational demands are high. Lightweight models (e.g., EfficientNet-B0, MobileNet) offer faster, resource-efficient inference with some trade-offs in accuracy. Consequently, the optimal model choice should balance accuracy and computational efficiency based on specific application needs. Additionally, incorporating attention mechanisms and optimizing hyperparameters could further enhance defect localization and overall performance.

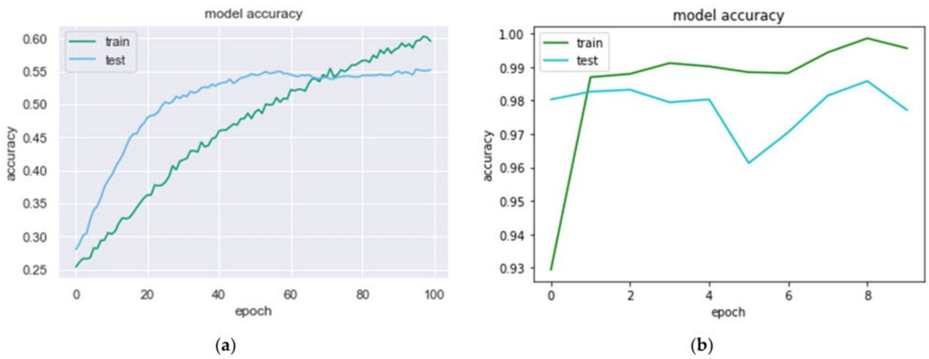

Figure 1: Example learning curves showing training (green curve) and validation (blue curve) accuracy for two CNN models. (a) A simple CNN trained from scratch – it gradually improves and reaches ~60% accuracy over ~100 epochs. (b) A deeper CNN model (using transfer learning) – it attains ~99% accuracy in just 2–3 epochs, demonstrating higher model capacity but with potential overfitting (the validation curve plateaus). These learning curves illustrate the difference in convergence speed and generalization between a small and a large model.

As illustrated in Figure 1, more complex models are capable of rapidly learning to distinguish defects (with validation accuracy quickly rising and stabilizing at a high level). However, the gap between training and validation accuracy in deeper models may indicate overfitting – for instance, in the case of VGG-16, training accuracy reached 100%, while validation accuracy remained lower. In contrast, simpler models (as shown in Figure 1a) took more epochs to gradually improve and never achieved high training accuracy, indicating limited model capacity: such a model does not overfit, but also cannot fully capture the complexity of the task. Therefore, when choosing a model, one must consider the balance between underfitting and overfitting. Proper tuning of hyperparameters and regularization techniques (early stopping, data augmentation, learning rate decay) helps achieve the optimal point, where the validation curve plateaus concurrently with the training curve.

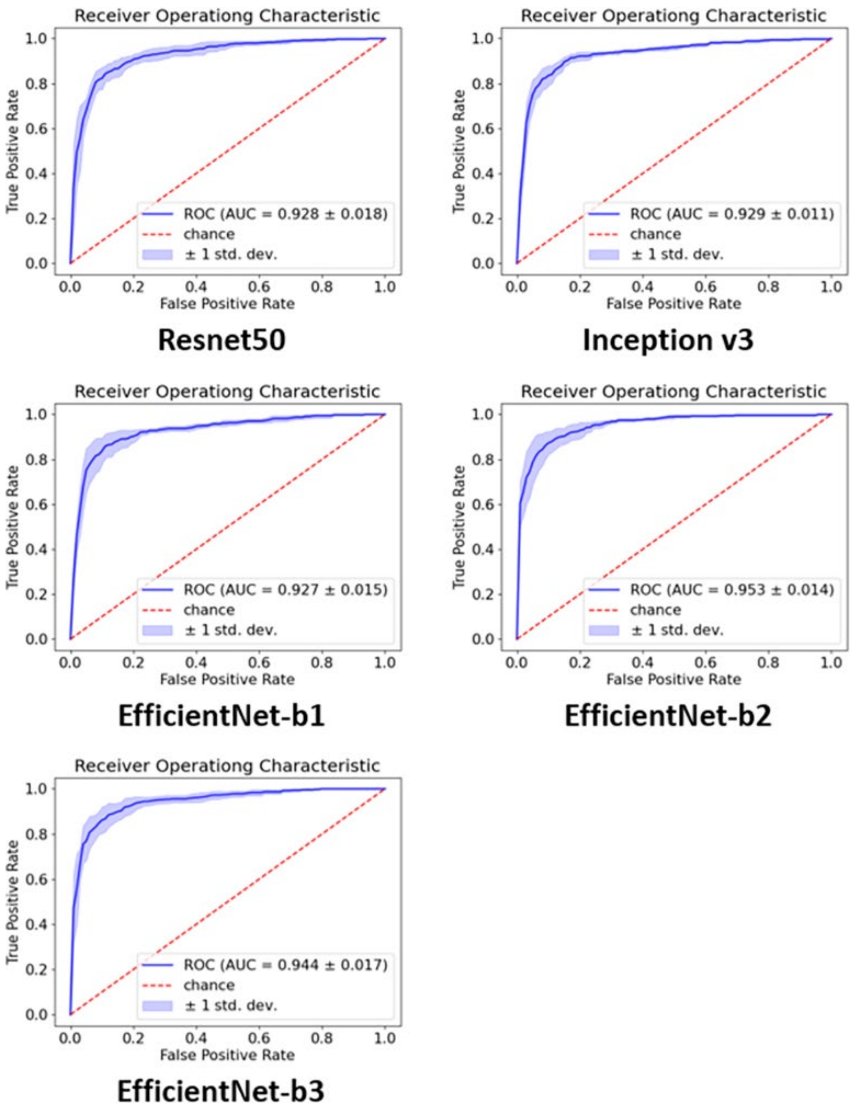

Figure 2: Example ROC curves for several CNN models on the test set (binary defect classification).

The x-axis represents the False Positive Rate, and the y-axis represents the True Positive Rate. The closer a model's curve is to the upper left corner, the better its discriminative capability. The graphs show results for ResNet-50 (AUC = 0.928 ± 0.018), Inception-V3 (AUC = 0.929 ± 0.011), EfficientNet-b1 (AUC = 0.927 ± 0.015), EfficientNet-b2 (AUC = 0.953 ± 0.014), and EfficientNet-b3 (AUC = 0.944 ± 0.017) – note that the EfficientNet-b2 curve (top right) encloses the others, demonstrating the best AUC in this example.

Figure 2 confirms the numerical metrics: models with higher accuracy have curves that lie higher and further left. For example, EfficientNet-b2 clearly outperforms ResNet-50 and Inception-V3 in terms of the area under the curve (AUC ~0.953 vs. ~0.928–0.929), which is consistent with our results, where the EfficientNet-based model achieved a better balance of sensitivity and specificity. At the same time, the difference between ResNet-50 and Inception-V3 on the ROC graph is minimal (their curves almost overlap), reflecting their comparable performance. High overlap of curves is also observed for ResNet, DenseNet, and Xception in our case, indicating statistically insignificant differences between them. Conversely, the curves for MobileNet and SqueezeNet (not shown) would be noticeably lower; for SqueezeNet, the curve would lie closer to the diagonal, confirming its lower AUC (≈0.90). The ROC analysis is an important complement to point metrics as it demonstrates model robustness across different decision thresholds. The graphs suggest that models such as ResNet, DenseNet, Xception, and EfficientNet exhibit consistently better performance across various thresholds, while simpler models are more sensitive to the chosen threshold.

4. Conclusions

Based on the conducted experimental comparison, the following conclusions can be drawn:

- High-Accuracy Models: Modern deep CNNs (ResNet-50, DenseNet-121, Xception, Inception-V3) achieve accuracies of 93–95% with an F1-score above 93%, effectively detecting even minimal defects. Their strength lies in high Precision/Recall values, which is critical for quality control; however, these models are computationally intensive and are recommended for deployment in environments with powerful GPUs.
- Specialized Model (DeFektNet): DeFektNet demonstrated the best overall performance (highest accuracy, recall, and F1-score), indicating its suitability for addressing the specific characteristics of wood defects. It is recommended for narrow-focused automated inspection systems, provided there is sufficient data for training.
- Lightweight Models: EfficientNet-B0 and MobileNet exhibit slightly lower accuracy (around 90%) but offer the advantages of lower computational cost and faster processing. These models are particularly suitable for real-time applications or mobile devices where resource constraints are critical.
- Limitations of VGG-16 and SqueezeNet: VGG-16 tends to overfit and operates slowly, while SqueezeNet, despite being extremely compact, often misses defects. Therefore, for modern quality control tasks, it is advisable to avoid using VGG-16 and to deploy SqueezeNet only in highly resource-constrained scenarios.

In summary, the choice of an optimal model for defect detection in wooden products depends on specific requirements: when maximum accuracy is needed, models such as ResNet, DenseNet, or DeFektNet are preferable; for systems requiring real-time operation, EfficientNet-B0 or MobileNet are more suitable. An ensemble approach may also be beneficial to enhance overall detection reliability. Future research should focus on adapting these architectures to the unique challenges of the wood processing industry and optimizing their performance through additional mechanisms like attention modules.

Declaration on Generative AI

In accordance with the CEUR-WS Guidelines on Generative AI, the authors confirm that the article was prepared independently and without the involvement of generative AI technologies for content creation. All writing, analysis, and interpretation of results reflect the authors' own work and reasoning.

References

[1] Tarek, I. H. M., Rahman, M. M., & Zaki, Z. M. (2024). Unsupervised Representation Learning using Wasserstein Generative Adversarial Network. International Journal of Computing, 23(3), 415-420. https://doi.org/10.47839/ijc.23.3.3660Palchyk V.O., Koval V.S., DEFECTING WOOD PRODUCTS USING CONVOLUTIONAL NEURONAL NETWORKS / Palchyk V.O., Koval V.S. / V Scientific and Practical Conference of Young Scientists and Students "Intelligent Computer Systems and Networks". 10 November 2022. Ternopil. Ukraine - p.38 URL: http://ki.wunu.edu.ua/conference

[2] Novitasari, D. C. R., Wulandari, P., & Haq, D. Z. (2022). Cervical Cancer Diagnosis System using Convolutional Neural Network ResidualNet. International Journal of Computing, 21(1), 61-68. https://doi.org/10.47839/ijc.21.1.2518

[3] Koval V., Zahorodnia D., Adamiv O. An Image Segmentation Method for Obstacle Detection in a Mobile Robot Environment / Proceedings of the 9th International Conference on Advanced Computer Information Technologies (ACIT’2019), June 5-7, 2019, Ceske Budejovice, Czech Republic, pp. 479-482.

[4] Andrij Sydor, Diana Zahorodnia, Pavlo Bykovyy, Ivan Kit, Vasyl Koval, Konrad Grzeszczyk. Image Recognition Methods Based on Hemming Distance / Proceedings of the 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'2019), September 18-21, 2019, Vol.2, Metz, France, pp. 1115-1121.

[5] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 770–778).

[6] Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv preprint arXiv:1409.1556.

[7] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 2818–2826).

[8] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 4700–4708).

[9] Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the 36th International Conference on Machine Learning (ICML) (pp. 6105–6114).

[10] Chollet, F. (2017). Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 1251–1258).

[11] Howard, A. G., et al. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv preprint arXiv:1704.04861.

[12] Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50× fewer parameters and <0.5MB model size. arXiv preprint arXiv:1602.07360.

[13] Li, S., Li, M., & Sun, X. (2018). Automated Detection of Wood Defects Using Convolutional Neural Networks. Journal of Forestry Research, 29(3), 411–420.

[14] Wang, X., Peng, Y., Lu, L., Lu, Z., & Summers, R. M. (2019). TextileNet: A Defect Detection System for Fabric. IEEE Transactions on Automation Science and Engineering, 16(3), 1335–1344.

[15] Umbrico, A., et al. (2017). Defect Detection in Ceramic Tile Production with Computer Vision and Machine Learning. Procedia Manufacturing, 11, 1545–1552.

[16] M. Komar, P. Yakobchuk, V. Golovko, V. Dorosh and A. Sachenko, "Deep Neural Network for Image Recognition Based on the Caffe Framework," 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP), Lviv, Ukraine, 2018, pp. 102-106, doi: 10.1109/DSMP.2018.8478621.

[17] V. Koval, V. Turchenko, V. Kochan, A. Sachenko and G. Markowsky, "Smart license plate recognition system based on image processing using neural network," Second IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 2003. Proceedings, Lviv, Ukraine, 2003, pp. 123-127, doi: 10.1109/IDAACS.2003.1249531.