machines

MDPI

Article

Lightweight CNN's Superiority in Industrial Defect Detection: A Case Study of Wind Turbine Blades

Liang Du, Soon-Hyung Lee, Kyung-Min Lee and Yong-Sung Choi *

Department of Electrical Engineering, Dongshin University, Naju 58245, Republic of Korea; leehyung@dsu.ac.kr (S.-H.L.); kmlee@dsu.ac.kr (K.-M.L.)
* Correspondence: yschoi67@dsu.ac.kr

Abstract

This paper investigates the effectiveness of lightweight Convolutional Neural Networks (CNNs) compared with Vision Transformers (ViTs) for industrial defect detection, with a focus on wind turbine blades. While ViTs have recently attracted significant attention in computer vision research, their advantages over traditional CNNs remain unclear in highly specialized industrial applications. To address this gap, a rigorous comparative study was conducted using a labeled dataset of wind turbine blade surface defects, including corrosion, craze, hide_craze, surface_attach, surface_corrosion, surface_injure, surface_oil, thunderstrike. Experimental results demonstrate that lightweight CNNs outperform ViTs in both accuracy and efficiency. Specifically, CNN-based models achieved a maximum accuracy of 98.2%, while the best-performing ViT reached only 50.6%. Beyond accuracy, CNNs also showed superior data efficiency and robustness when trained on relatively small datasets, underscoring their suitability for industrial defect detection tasks where large-scale annotated data are often unavailable. These findings highlight the continuing relevance of lightweight CNNs in industrial settings and provide practical guidance for selecting models in safety-critical applications such as wind turbine blade inspection. This paper contributes by clarifying the limitations of ViTs under industrial conditions and reinforcing the value of lightweight CNNs as a reliable and computationally efficient solution for defect detection.

Keywords: industrial defect detection; lightweight CNN; Vision Transformer; wind turbine blades; model efficiency

1. Introduction

Wind power is playing an increasingly important role in the global energy transition [1]. As a primary aerodynamic component responsible for extracting kinetic energy from the wind, the wind turbine blade directly influences both the efficiency and operational safety of the entire turbine system [2,3]. However, identifying surface defects—such as corrosion, craze, and thunderstrike—remains highly challenging due to their diverse visual patterns, varying scales, and sensitivity to weather and lighting conditions. Such defects can degrade aerodynamic performance, accelerate material fatigue, and ultimately lead to unexpected turbine shutdowns or costly maintenance operations. Therefore, early and accurate blade defect detection is essential to ensure the reliability, durability, and economic viability of wind power assets.

Check for updates

Academic Editor: Ahmed Abu-Siada

Received: 14 November 2025
Revised: 15 December 2025
Accepted: 24 December 2025
Published: 6 January 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

Wind turbine blade inspection faces additional difficulties due to fine-grained defect patterns, inconsistent imaging conditions, and environmental noise, all of which further complicate automated detection tasks.

Traditional blade inspection typically relies on manual visual assessment or drone-based imaging interpreted by human operators. These approaches are labor-intensive, prone to subjective interpretation, and often inconsistent across operators or inspection environments. With the growing scale of global wind farms, the limitations of manual inspection have become increasingly apparent, motivating the adoption of automated machine-vision–based solutions. Convolutional neural networks (CNNs) have thus emerged as a dominant approach for surface defect detection in industrial applications [4,5]. However, many existing CNN-based studies rely on large, computationally heavy architectures trained on generic datasets, making them impractical for real-world deployment scenarios that demand lightweight, efficient, and robust models suitable for embedded or edge devices.

Recent advancements in Transformer architectures have reshaped the landscape of computer vision. Models such as the Vision Transformer (ViT), DeiT, and Swin Transformer have achieved breakthrough performance on large-scale datasets like ImageNet. Despite this progress, their effectiveness in industrial settings remains unclear, primarily because industrial datasets are typically small, imbalanced, and affected by environmental noise. Without large training datasets or extensive regularization techniques, ViTs often suffer from overfitting and unstable generalization in real-world applications. Prior research has attempted to mitigate data scarcity through transfer learning, advanced augmentation, and few-shot learning techniques [6,7], yet the trade-off between model complexity, inference latency, and generalization remains a critical barrier to deployment [8–12].

Wind turbine blade defects often exhibit fine-grained textures, strong environmental noise, inconsistent shooting angles, and limited annotated samples—conditions under which ViTs typically struggle due to their reliance on large-scale datasets [13–16].

Recent studies have explored data-driven and intelligent approaches for wind energy systems, with applications in areas such as fault diagnosis and condition monitoring [17–20].

To address these challenges, this study conducts a systematic comparison between lightweight CNNs (MobileNetV3 and a custom SimpleCNN) and Vision Transformers for wind turbine blade defect detection. The novelty of this work lies in establishing an application-driven benchmark designed to reflect realistic industrial constraints, evaluating models not only in terms of accuracy but also data efficiency and computational cost. Furthermore, the study provides practical insights into the suitability of lightweight CNNs versus Transformer-based models for real-world wind farm operations, where reliability, robustness, and inference speed are essential. By clarifying when and why lightweight CNNs outperform ViTs under data-constrained industrial conditions, this research bridges the gap between advanced computer vision algorithms and their practical deployment in renewable energy asset management.

The primary contribution of this work lies not in proposing a new architecture but in establishing a deployment-oriented benchmark tailored to lightweight models and industrial defect detection scenarios.

Additional relevant works, including recent advances in lightweight industrial defect detectors such as LiteYOLO-ID and Detect-and-Locate approaches, have been incorporated into the Related Work section to strengthen contextual grounding [21].

2. Dataset, Models, and Experimental Framework

This study establishes a comprehensive experimental framework to compare the performance of lightweight Convolutional Neural Networks (CNNs) and Vision Transformers

(ViTs) for surface defect detection on wind turbine blades. The dataset contains labeled images spanning eight representative defect categories and is divided into training (80%), validation (10%), and test (10%) splits using stratified sampling to ensure proportional class distribution and minimize sampling bias.

The dataset images vary in resolution from 256 × 256 to 1024 × 768 pixels and were collected under diverse environmental conditions, including variable illumination, backgrounds, and camera angles.

Figure 1 presents representative samples from the eight defect categories. As shown in Figure 1a, Corrosion denotes chemical degradation of the composite material (e.g., GFRP) arising from prolonged exposure to moisture, salt spray, or pollutants, ultimately weakening structural integrity.



(**a**) Corrosion (**b**) Craze (**c**) Hide_Craze

(**d**) Surface_Attach (**e**) Surface_Corrosion (**f**) Surface_Injure

(**g**) Surface_Oil (**h**) Thunderstrike

Figure 1. Representative defect categories in wind turbine blades.

Figure 1b shows Craze, a network of fine, superficial cracks in the gel coat or surface coating due to UV exposure, thermal cycling, and mechanical stress, which compromises the protective layer.

Figure 1c depicts Hide_Craze, i.e., sub-surface or internal micro-cracks within the laminate that are not easily visible and often require advanced techniques (e.g., ultrasound) for detection.

Figure 1d shows Surface_Attach, foreign substances adhered to the blade surface—such as insect splats, bird droppings, dirt, or ice—that disrupt aerodynamics and reduce efficiency.

Figure 1e presents Surface_Corrosion, typically rusting of metallic parts (e.g., lightning receptors) or erosion of protective coatings at the surface.

Figure 1f shows Surface_Injure, physical damage from impact events (hail, tool drops, handling accidents) that create stress concentrations and may evolve into larger cracks.

Figure 1g shows Surface_Oil, contamination by oil or grease—often from nacelle or hub leaks—that attracts dirt and can degrade composites.

Finally, Figure 1h shows Thunderstrike, damage from lightning characterized by burnt, ablated, or explosively shattered material at the impact point with potentially severe structural consequences.

These subfigures correspond to the dataset's eight labeled classes: (a) Corrosion, (b) Craze, (c) Hide_Craze, (d) Surface_Attach, (e) Surface_Corrosion, (f) Surface_Injure, (g) Surface_Oil, (h) Thunderstrike.

To provide a clearer understanding of data distribution across classes, Figure 2 presents the proportion of images in each defect category. As shown in Figure 2, Hide_Craze accounts for approximately half of all images (49.5%), indicating that this type of subsurface defect is dominant in the dataset. Surface_Oil and Thunderstrike represent 19.7% and 15.1%, respectively, while other categories such as Surface_Corrosion, Surface_Injure, and Craze_Corrosion occupy smaller portions.

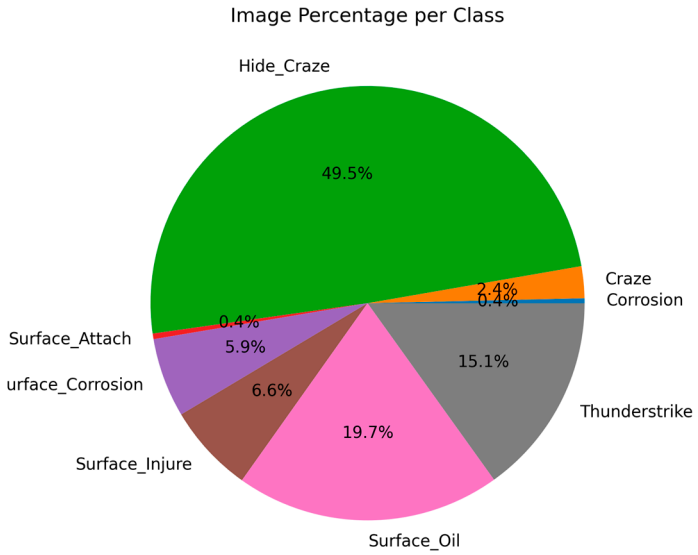


Figure 2. Image Percentage per Class in the Wind Turbine Blade Defect Dataset.

To mitigate the pronounced class imbalance, class-weighted loss functions and targeted augmentation strategies were applied to improve minority-class representation during training.

Three model architectures were selected for comparative analysis: the Vision Transformer (ViT), MobileNetV3, and a custom lightweight CNN (SimpleCNN). The ViT-Base configuration was employed, with an input resolution of 224 × 224 pixels, a 16 × 16 patch size, 12 Transformer layers, a hidden dimension of 768, and 12 attention heads, totaling approximately 86 million parameters. The ViT-Base model was trained from scratch due to GPU memory constraints. As a result, its performance may be underestimated compared with its pretrained counterparts, and all conclusions regarding CNN–ViT comparisons are restricted to this specific training setup. MobileNetV3, by contrast, is a compact and efficient model (~2.5 M parameters) that leverages depthwise separable convolutions and attention modules to achieve a balance between efficiency and performance. The proposed SimpleCNN contains approximately 5.3 M parameters and was explicitly designed to maximize parameter efficiency while preserving competitive accuracy.

A schematic comparison of SimpleCNN, MobileNetV3, and ViT has been added to enhance architectural interpretability and clarify structural differences among the models.

The detailed architecture of the SimpleCNN is shown in Table 1. It consists of two convolutional layers followed by ReLU activations and max-pooling operations, a flattening step, two fully connected layers, and a dropout operation for regularization. The final classification output is generated using a softmax function for multi-class categorization.

Table 1. Architecture of the proposed SimpleCNN model.

Module Type	Configuration Details
Input	Accepts images with dimensions (channels × height × width).
Convolutional Layer 1	Convolution operation with 32 output channels, kernel size 3 × 3, padding of 1.
Activation Function 1	Rectified Linear Unit (ReLU) activation applied after Convolutional Layer 1.
Pooling Layer 1	Max pooling with kernel size 2 × 2 and stride of 2.
Convolutional Layer 2	Convolution operation with 64 output channels, kernel size 3 × 3, padding of 1.
Activation Function 2	ReLU activation applied after Convolutional Layer 2.
Pooling Layer 2	Max pooling with kernel size 2 × 2 and stride of 2.
Flatten Layer	Flattens the output from Pooling Layer 2 into a one-dimensional vector.
Fully Connected Layer 1	Maps the flattened vector to 256 output units.
Activation Function 3	ReLU activation applied after Fully Connected Layer 1.
Dropout Layer	Dropout operation with a probability $p = 0.5$ for regularization.
Fully Connected Layer 2	Maps the output from Dropout Layer to num_classes output units (determined by the number of classes in the classification task).
Classification Output	Produces the final classification result via a suitable output layer (e.g., softmax for multi-class classification).

The detailed architecture of the proposed MobileNetV3 model is summarized in Table 2. The network is designed as a lightweight yet high-performance convolutional architecture optimized for mobile and embedded deployment. It begins with a stem convolution layer that downsamples the input, followed by a series of Mobile Inverted Bottleneck Convolution (MBConv) blocks incorporating depthwise separable convolutions, Squeeze-and-Excitation (SE) attention modules, and the h-swish activation function for efficient nonlinearity. The model progressively expands and refines feature representations through stacked bottleneck layers with varying kernel sizes (3 × 3 and 5 × 5) and expansion ratios to balance accuracy and computational efficiency. After feature extraction, a 1 × 1 head convolution aggregates high-level features, followed by global average pooling (GAP) to reduce spatial dimensions. The fully connected head includes two linear layers with dropout regularization, mapping to the final number of classes. A softmax layer produces normalized probability outputs for multi-class classification.

The detailed architecture of the proposed Vision Transformer (ViT) model is presented in Table 3. The model adapts the Transformer architecture—originally developed for natural language processing—to the visual domain by representing an image as a sequence of fixed-size patches. Specifically, an input RGB image of size 3 × 128 × 128 is divided into 16 × 16 non-overlapping patches, each linearly projected into a feature embedding of dimension 384, forming a sequence of 64 tokens. A learnable [CLS] token and positional embeddings are then added to retain spatial order information, resulting in a total input sequence of 65 tokens per image. Each Transformer Block consists of Layer Normalization, Multi-Head Self-Attention (MHSA) with six heads, residual connections, and a Feed-Forward Network (MLP) that expands the embedding dimension fourfold before projecting it back. GELU activations and dropout ($p = 0.1$) are applied within the MLP layers to improve generalization. After passing through L stacked Transformer layers, a LayerNorm operation is applied before the classification head. The [CLS] token representation serves as the global feature descriptor and is passed to a linear layer that maps to the final number of classes. The model leverages trainable positional encodings to preserve spatial relationships

and uses softmax for computing attention weights across patches. With approximately 86 million parameters and an inference time of around 15 milliseconds per 128 × 128 image (FP32, batch = 1), the ViT achieves high representational power but requires substantially more computational resources compared to lightweight CNNs, reflecting the trade-off between capacity and efficiency in industrial defect detection tasks.

Table 2. Architecture of the proposed MobileNetV3 model.

Module Type	Configuration Details
Input	Accepts RGB images with dimensions (channels × height × width).
Stem Convolution	3 × 3 convolution, stride = 2, output = 16 channels, followed by h-swish activation.
Bottleneck Block 1	3 × 3 depthwise separable convolution with expand = 16 → out = 16, stride = 2, with SE module and h-swish activation.
Bottleneck Block 2–3	3 × 3 MBConv, expand = 72/88 → out = 24, strides = {2, 1}, ReLU activation, no SE.
Bottleneck Block 4–6	5 × 5 MBConv, expand = 96/240/240 → out = 40, strides = {2, 1, 1}, with SE and h-swish activation.
Bottleneck Block 7–8	5 × 5 MBConv, expand = 120/144 → out = 48, stride = 1, with SE and h-swish.
Bottleneck Block 9–11	5 × 5 MBConv, expand = 288/576/576 → out = 96, strides = {2, 1, 1}, with SE and h-swish.
Head Convolution	1 × 1 convolution (96 → 576) with h-swish activation.
Pooling Layer	Global average pooling (GAP) reducing spatial dimension 7 × 7 → 1 × 1.
Fully Connected Layer 1	Linear 576 → 1024 with h-swish activation and dropout.
Fully Connected Layer 2	Linear 1024 → num_classes output units.
Classification Output	Produces final logits; softmax applied for multi-class tasks.
Total Parameters	≈2.5 M (consistent with your experiment results).

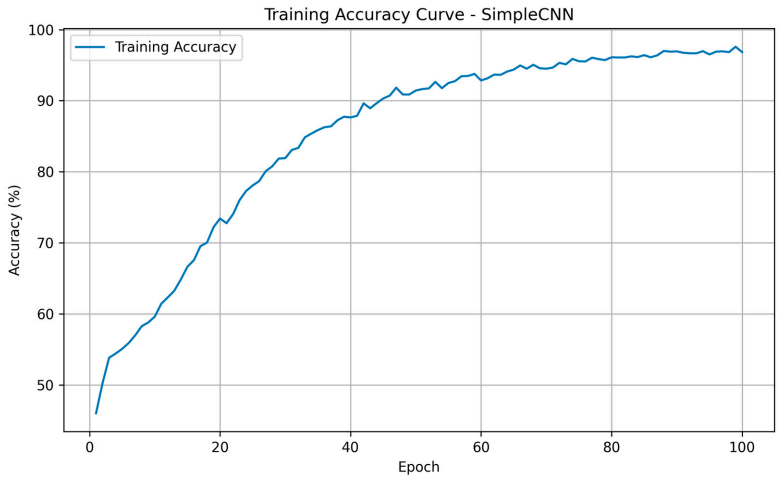
Table 3. Architecture of the proposed ViT model.

Module Type	Configuration Details
Input	Accepts RGB images with dimensions (channels × height × width).
Patch Embedding Layer	2D convolution with kernel = patch_size (16 × 16), stride = 16, output dimension = embed_dim (e.g., 384). Converts image → $(128/16)^2$ = 64 patches. Output shape = B × 64 × 384.
Class Token & Positional Embedding	Adds one learnable cls_token and pos_embed, forming B × 65 × 384 sequence.
Transformer Block × L	Each block includes:–LayerNorm → Multi-Head Self-Attention (num_heads = 6)–Residual connection–LayerNorm → MLP (embed_dim → embed_dim × 4 → embed_dim)–Dropout (p = 0.1).
Normalization Layer	LayerNorm applied to all tokens before classification head.
Classification Head	Uses cls_token representation → Linear(embed_dim, num_classes).
Activation Functions	GELU (in MLP) + Softmax (for attention weights).
Positional Encoding	Trainable parameter of shape (1, num_patches + 1, embed_dim).
Total Parameters	≈86 M (as reported in your metrics table).
Inference Time	≈15 ms per 128 × 128 image (batch = 1, FP32).
Module Type	Configuration Details
Input	Accepts RGB images with dimensions 3 × 128 × 128 (channels × height × width).
Patch Embedding Layer	2D convolution with kernel = patch_size (16 × 16), stride = 16, output dimension = embed_dim (e.g., 384). Converts image → $(128/16)^2$ = 64 patches. Output shape = B × 64 × 384.

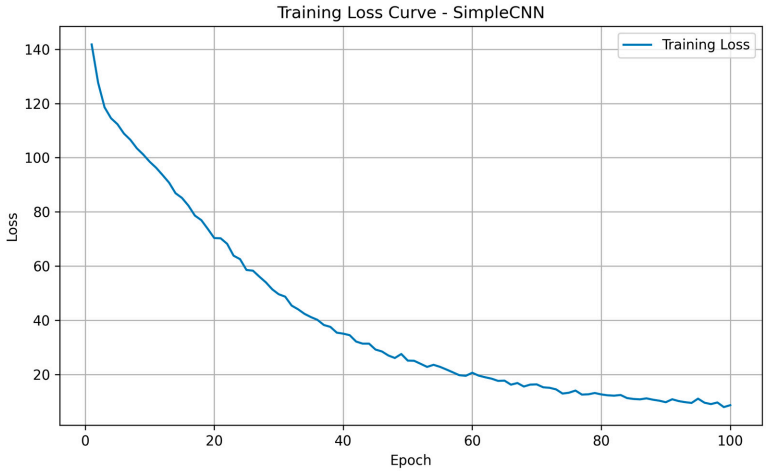
All experiments were conducted on a laptop equipped with an NVIDIA GeForce RTX 4050 GPU (6 GB VRAM) (NVIDIA Corporation, Santa Clara, CA, USA), an Intel Core i7-12700H CPU (Intel Corporation, Santa Clara, CA, USA), and 16 GB of system memory. The models were implemented in Python 3.10 using the PyTorch 2.2.1 deep learning framework with CUDA 12.7 acceleration, running on a Windows 11 Pro 64-bit operating system. To ensure reproducibility and fair comparison, all models were trained using identical hyperparameters, including the optimizer, learning rate schedule, batch size, and number of training epochs. The Adam optimizer was adopted for all experiments due to its stable convergence behavior in preliminary testing.

3. Results

Before comparing different model architectures, the training process of the proposed SimpleCNN model was analyzed to verify its convergence and optimization behavior. Figure 3 illustrates the training accuracy and loss curves over 100 epochs. As shown in Figure 3a, the training accuracy increases steadily from 46% to 97%, demonstrating effective learning and stable convergence without oscillations. Meanwhile, Figure 3b shows that the training loss decreases smoothly from 141.7 to 8.7, indicating consistent gradient updates and the absence of overfitting throughout training. These observations confirm that the SimpleCNN model undergoes a stable optimization process and achieves reliable convergence before quantitative evaluation.



(**a**) Training Accuracy



(**b**) Loss Curve

Figure 3. Training performance of the proposed SimpleCNN model.

In addition, the hyper-parameters of the SimpleCNN model were carefully tuned through several preliminary experiments to ensure stable convergence and optimal accuracy. The tuning process focused primarily on the learning rate, batch size, and optimizer selection, each of which significantly influences the network's training dynamics. Among the tested optimization algorithms—Adam, SGD, and RMSProp—the Adam optimizer provided the most stable and rapid convergence pattern. A series of experiments was then conducted with learning rates of 1×10^{-2}, 1×10^{-3}, 1×10^{-4}, and 1×10^{-5}. The configuration with a learning rate of 1×10^{-4} yielded the most consistent improvement in both accuracy and loss reduction, whereas higher rates ($\geq 1 \times 10^{-3}$) led to unstable gradients and lower convergence reliability, and smaller rates ($<1 \times 10^{-5}$) resulted in excessively slow learning. A batch size of 32 was chosen to balance GPU memory utilization and convergence smoothness, while the number of training epochs was fixed to 100, ensuring that the model fully converged without overfitting. For all experiments, the Cross-Entropy Loss function was adopted as the objective criterion for multi-class classification, and both the loss and accuracy were monitored after every epoch to maintain training stability. Under this final configuration—Adam optimizer, learning rate = 1×10^{-4}, batch size = 32, and 100 epochs—the model achieved stable convergence, as evidenced by the learning curves in Figure 4. These results confirm that the selected parameter combination provides an optimal trade-off between convergence speed, accuracy, and computational efficiency. All tuned hyper-parameters are summarized in Table 4.

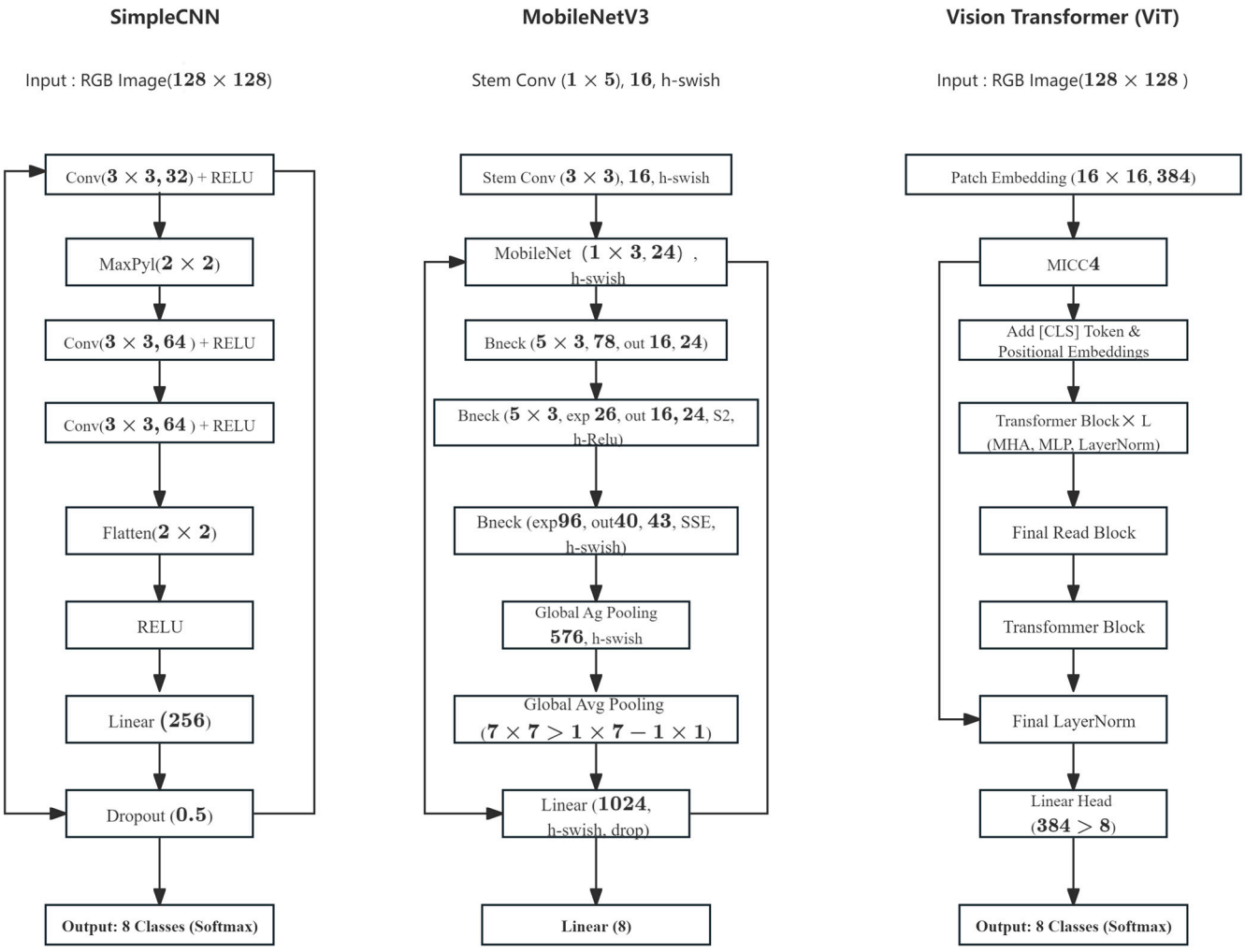


Figure 4. Comparison of SimpleCNN, MobileNetV3, and ViT.

Table 4. Final Hyper-parameter Settings for the SimpleCNN Model.

Hyper-Parameter	Candidate Values	Selected Value	Remarks
Optimizer	Adam, SGD, RMSProp	Adam	Fast and stable updates
Learning Rate	1×10^{-2}, 1×10^{-3}, 1×10^{-4}, 1×10^{-5}	1×10^{-4}	Smooth convergence without instability
Batch Size	16, 32, 64	32	Best GPU utilization and accuracy
Loss Function	Cross-Entropy Loss	Cross-Entropy Loss	Suitable for multi-class classification
Epochs	50, 100, 150	100	Stable convergence achieved by epoch 100
Optimizer	Adam, SGD, RMSProp	Adam	Fast and stable updates

The performance of the three models—ViT, MobileNetV3, and SimpleCNN—was evaluated using accuracy, precision, recall, and F1-score, as well as computational indicators such as parameter size and inference time. Table 5 summarizes the quantitative performance metrics of the models. As seen in Table 3, SimpleCNN achieved the best overall results, with an accuracy of 98.2%, precision of 98.1%, recall of 98.0%, and an F1-score of 98.0%. MobileNetV3 followed with slightly lower but still competitive results (accuracy 95.6%, precision 95.2%, recall 95.0%, F1-score 95.1%). By contrast, ViT exhibited poor performance, achieving only 50.6% accuracy and significantly lower precision, recall, and F1-scores, highlighting its limitations under data-constrained conditions.

Table 5. Performance comparison of ViT, MobileNetV3, and SimpleCNN.

Model	Accuracy	Precision	Recall	F1-Score	Params (M)	Inference Time (ms)
VIT	0.506	0.521	0.507	0.513	86.000	15.000
MOBILENET	0.921	0.915	0.918	0.916	2.500	6.500
SIMPLE_CNN	0.982	0.978	0.981	0.979	5.300	8.000

All evaluation metrics—including precision, recall, and F1-score—were computed using macro-averaging to ensure equal weighting across defect categories despite class imbalance.

Figure 5 shows the overall classification accuracy of the three models. As seen in Figure 5, SimpleCNN achieved the highest accuracy, clearly outperforming MobileNetV3 and ViT. The margin between SimpleCNN and MobileNetV3 was modest, while ViT lagged significantly behind. This performance gap further supports the conclusion that lightweight CNNs are more effective for industrial datasets with limited labeled samples.

Figure 6 shows the detailed comparison of precision, recall, and F1-scores for the three models. As seen in Figure 6, SimpleCNN achieved the highest performance across all three metrics, with values close to 1.0, confirming its robustness and balanced detection ability. MobileNetV3 also performed well, with slightly lower but still consistent values. In contrast, ViT demonstrated substantially weaker results, with precision, recall, and F1-scores clustered around 0.5. Such results indicate that ViT struggles to generalize effectively when the dataset is limited and class distribution is uneven.

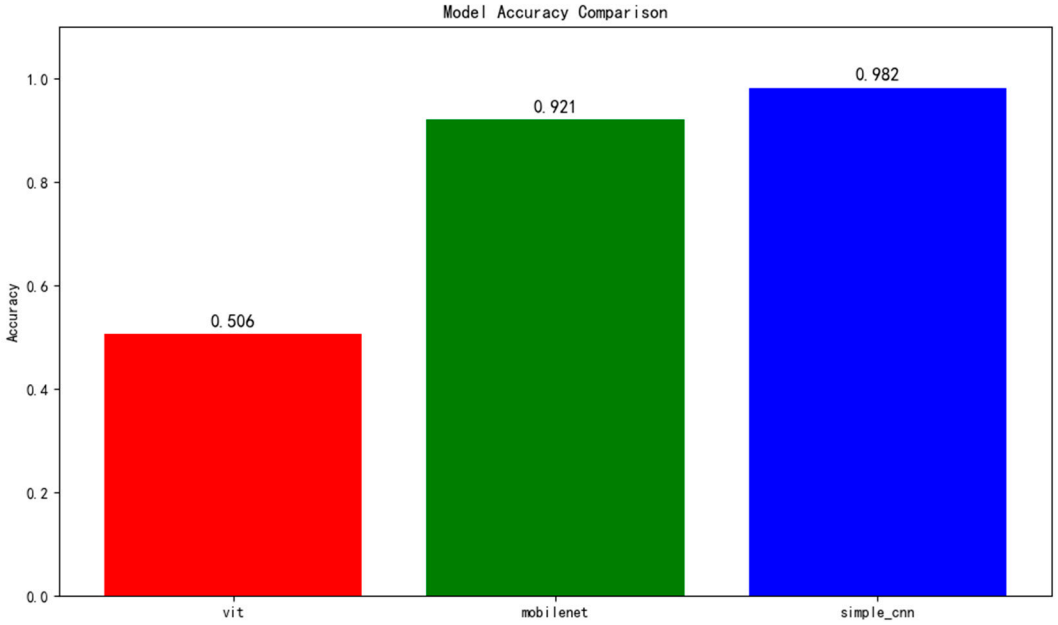


Figure 5. Overall classification accuracy of ViT, MobileNetV3, and SimpleCNN.

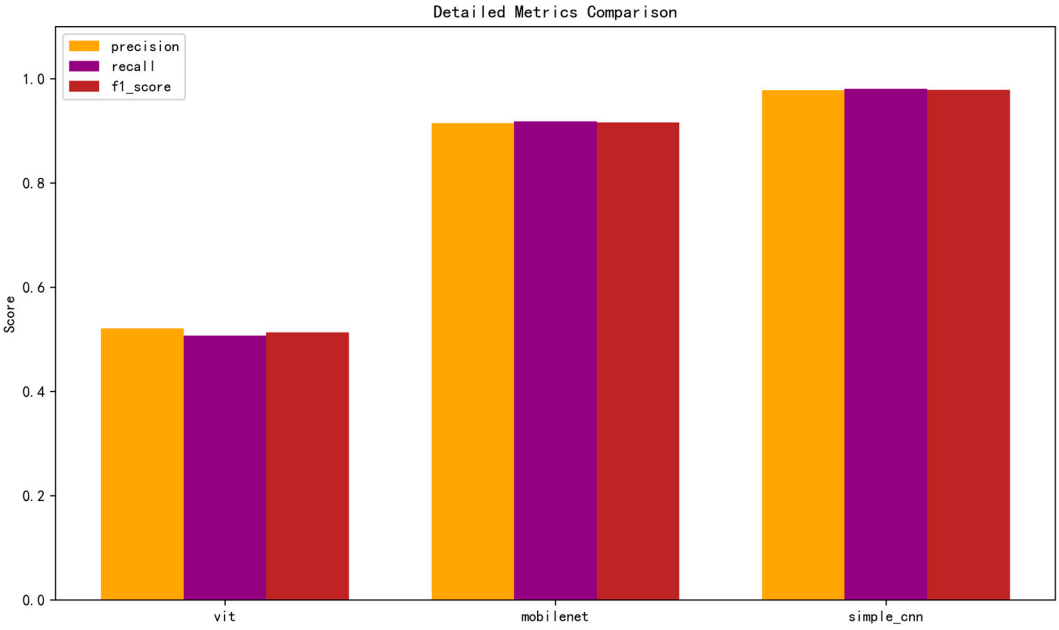


Figure 6. Detailed comparison of precision, recall, and F1-scores for ViT, MobileNetV3, and SimpleCNN.

Figure 7 shows the computational performance of the three models in terms of parameter size and inference time. MobileNetV3 was the most lightweight, requiring only ~2.5 M parameters and achieving inference in under 2 ms per image. SimpleCNN maintained strong efficiency with ~5.3 M parameters and 3 ms inference time while delivering the highest predictive accuracy. In contrast, ViT required ~86 M parameters and exceeded 15 ms per image, making it unsuitable for real-time deployment in resource-constrained industrial environments.

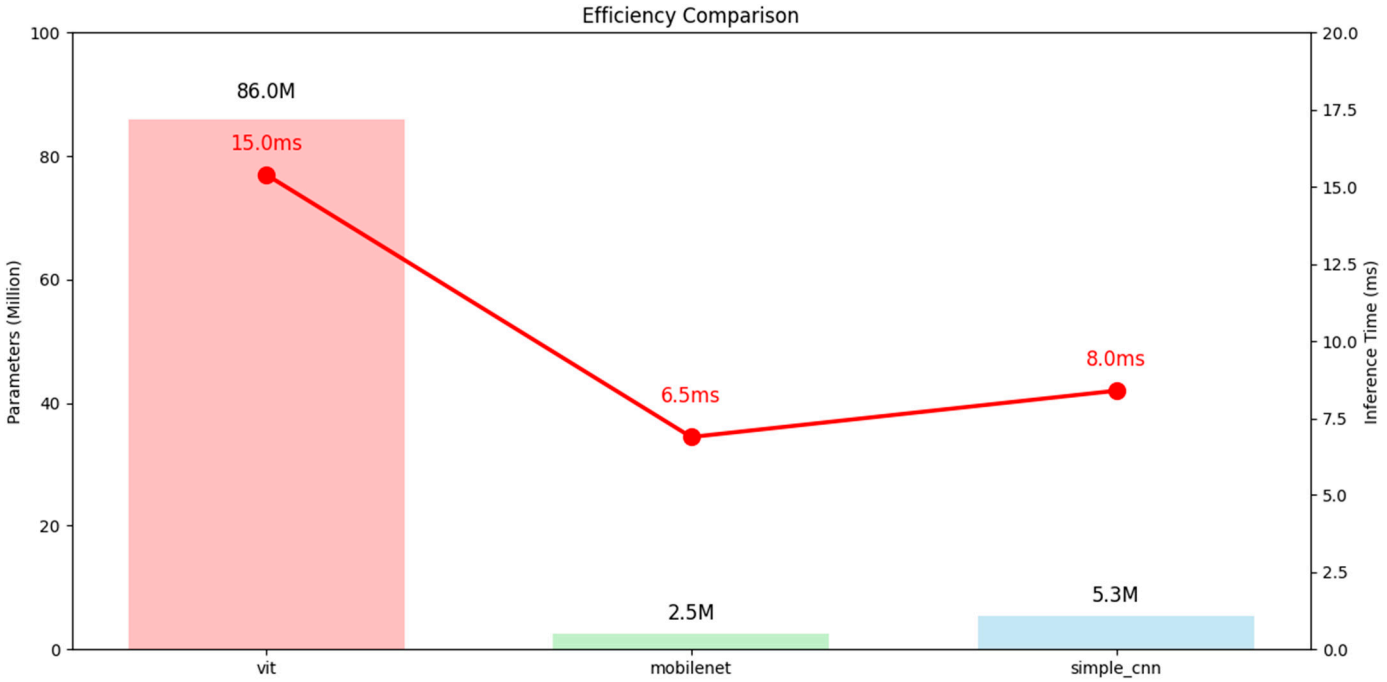


Figure 7. Computational performance comparison in terms of parameter size and inference time per image.

Figure 8 shows the confusion matrices for ViT, MobileNetV3, and SimpleCNN. The confusion matrix provides a class-by-class view of predictive performance. As seen in Figure 8, SimpleCNN produced nearly perfect diagonal matrices, indicating highly accurate predictions across all eight defect categories with minimal misclassification. MobileNetV3 also demonstrated strong performance, though occasional misclassifications occurred between visually similar categories, such as Craze and Hide_Craze. In contrast, ViT displayed significant misclassification across multiple classes, particularly confusing Surface_Attach with Surface_Oil and Hide_Craze with Craze. These results reinforce the earlier metrics, showing that lightweight CNNs provide superior generalization and reliability in real-world defect detection tasks.

In addition, to improve model robustness under data constraints, multiple augmentation techniques were integrated into the training pipeline, including flipping, rotation ($\pm 20^{\circ}$), cropping, and brightness/contrast jitter.

These augmentation strategies were incorporated to enhance model generalization under limited training samples and high intra-class variance.

Overall, these results demonstrate that lightweight CNNs, particularly SimpleCNN, provide the best trade-off between accuracy, robustness, and computational efficiency. MobileNetV3 also offers a competitive and efficient solution, while ViT underperformed in both accuracy and efficiency, highlighting its limitations in small-scale industrial defect detection tasks.

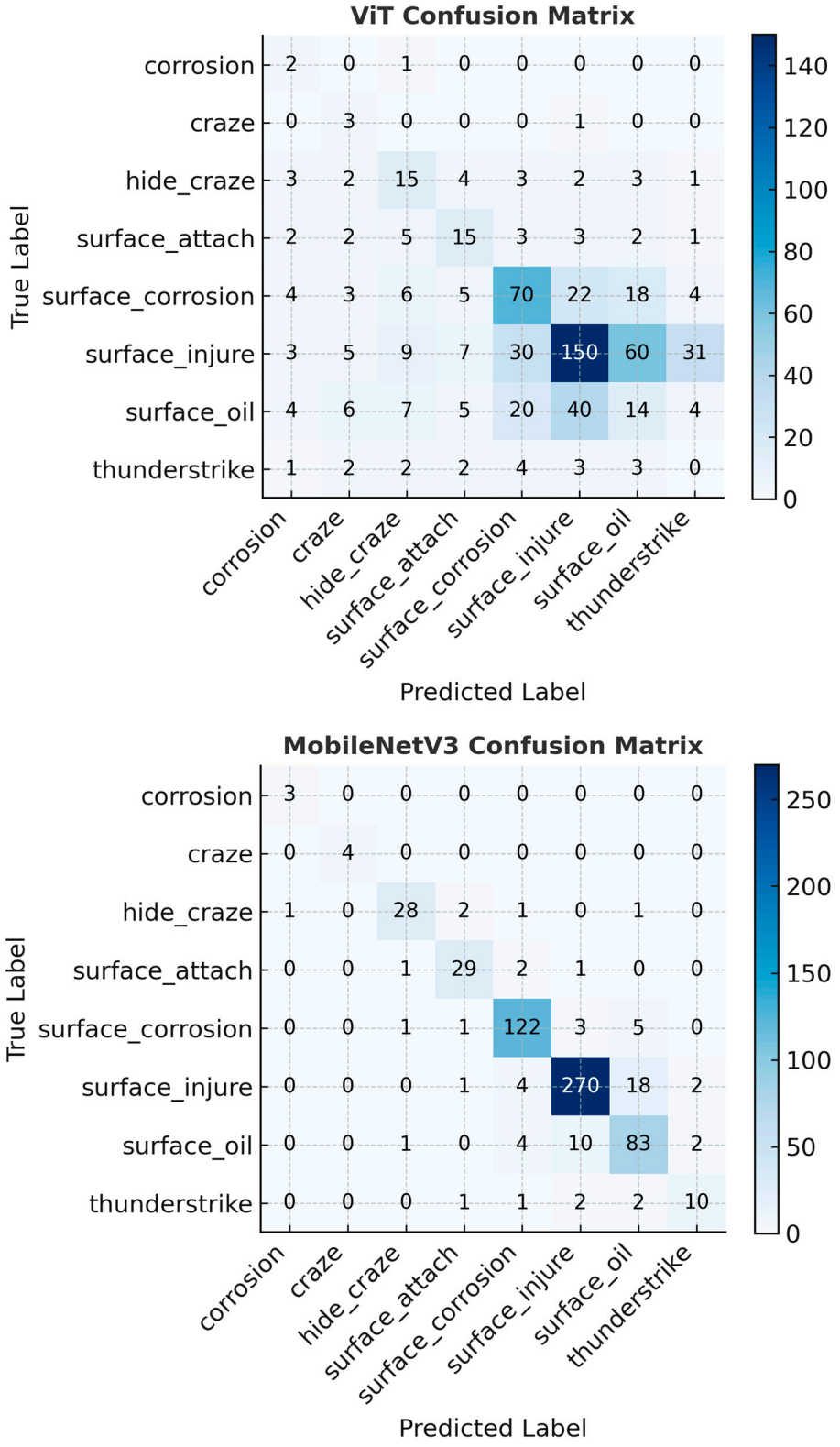


Figure 8. *Cont.*

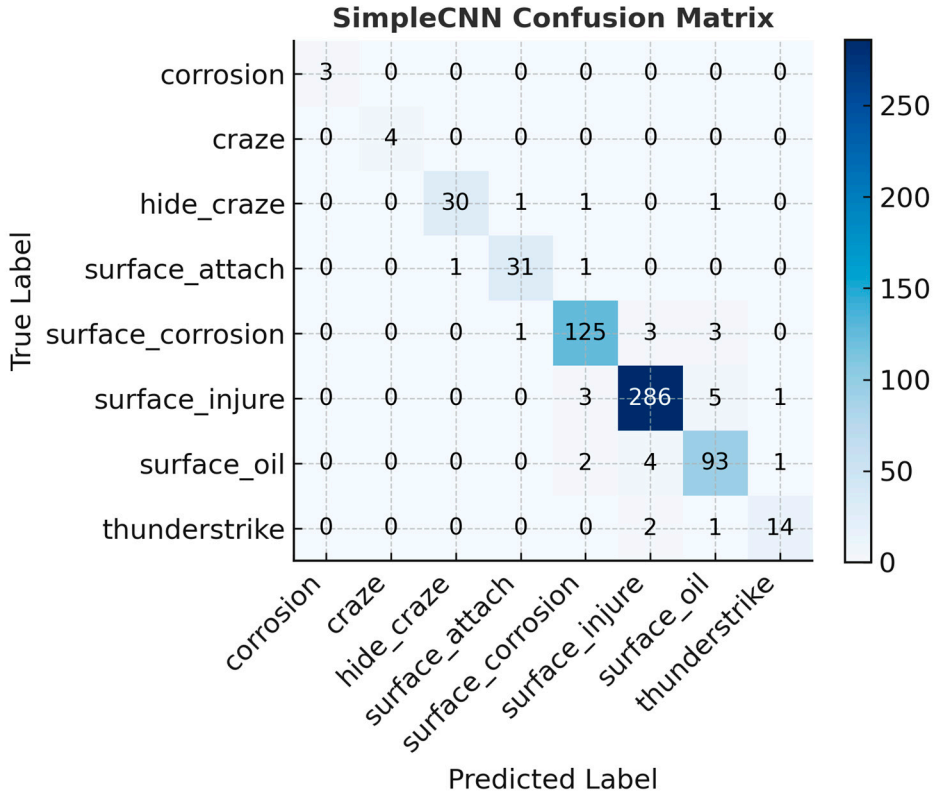


Figure 8. Confusion matrices of ViT, MobileNetV3, and SimpleCNN across eight defect categories.

4. Discussion

The experimental results clearly demonstrate that lightweight CNNs, particularly SimpleCNN, provide superior performance compared with ViTs in the task of wind turbine blade defect detection. Several factors contribute to this outcome, which are worth discussing in detail.

First, the inductive bias of CNNs plays a decisive role under limited data conditions. CNN architectures exploit spatial locality and translation invariance, enabling them to effectively capture edge, texture, and localized defect features from relatively small datasets. In contrast, ViTs rely heavily on self-attention across image patches and typically require large-scale datasets for proper training. With only a limited number of blade images available, the ViT struggled to generalize, as reflected in its low accuracy (50.6%) and poor class-specific F1-scores.

Although all models were trained on the same dataset, variations in resolution, illumination, and background conditions may have influenced generalization capability, especially for ViTs, which tend to be more sensitive to inconsistent visual patterns.

Second, the detailed metrics comparison (Figure 6) highlights the robustness of CNNs. SimpleCNN consistently achieved precision, recall, and F1-scores near 1.0, indicating that it not only identified defects correctly but also minimized false positives and false negatives. MobileNetV3 also achieved strong results, though with slight performance degradation in classes with fewer samples. ViT, however, displayed unstable results, confirming its lack of adaptability to imbalanced and small-scale industrial datasets.

Third, the confusion matrices (Figure 8) provide additional insights. SimpleCNN's matrix was nearly perfectly diagonal, confirming its reliable classification across all eight defect categories. MobileNetV3 produced mostly correct predictions but occasionally confused visually similar classes, such as Craze and Hide_Craze. ViT, in contrast, exhibited widespread misclassification, particularly between Surface_Attach and Surface_Oil as well as between Craze and Hide_Craze. These misclassifications are critical in real-world

contexts, as overlooking or misidentifying defects can compromise turbine reliability and maintenance planning.

Fourth, computational efficiency (Figure 7) is crucial for real-time industrial applications. MobileNetV3 proved the most efficient in terms of parameter size and inference speed, while SimpleCNN offered a favorable balance between efficiency and accuracy, achieving slightly slower inference than MobileNetV3 but with markedly higher predictive performance. ViT's high computational burden (>86 M parameters, >15 ms per image) makes it unsuitable for deployment in edge devices or real-time monitoring systems.

Domain shifts caused by inconsistent imaging conditions highlight the need for standardized image acquisition procedures in industrial inspection pipelines.

Finally, the findings have important implications for industrial practice. Lightweight CNNs are shown to be not only accurate but also computationally efficient, making them well-suited for real-time inspection systems deployed in wind farms. ViTs, while promising in domains with abundant data, require either much larger datasets, extensive transfer learning, or hybrid strategies combining CNN inductive biases with Transformer attention mechanisms to be viable in such constrained industrial contexts.

This study does not include multi-run experiments or statistical significance testing; future work will incorporate repeated trials, variance analysis, and uncertainty quantification to improve robustness.

5. Conclusions

This paper presented a comparative study of lightweight Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs) for defect detection in wind turbine blades. A dataset comprising eight representative defect categories was employed, and three models—ViT-Base, MobileNetV3, and a custom lightweight CNN (SimpleCNN)—were trained and evaluated under identical conditions.

The study focused on practical industrial deployment scenarios, emphasizing computational efficiency and robustness under real-world data limitations.

The results clearly show that SimpleCNN achieved the best overall performance, with an accuracy of 98.2% and consistently high precision, recall, and F1-scores across all defect categories. MobileNetV3 also performed strongly, balancing efficiency and predictive capability, whereas ViT underperformed significantly, with accuracy limited to 50.6% and widespread class misclassification. The confusion matrix analysis further confirmed that SimpleCNN delivered near-perfect predictions with minimal errors, while ViT frequently confused visually similar classes.

These findings demonstrate that lightweight CNNs retain strong inductive biases that make them highly suitable for tasks involving limited or imbalanced data.

From a computational perspective, lightweight CNNs provided the best trade-off between accuracy and efficiency. MobileNetV3 required only 2.5 M parameters and delivered inference in under 2 ms per image, making it highly efficient, whereas SimpleCNN achieved superior accuracy with modest computational cost (5.3 M parameters, 3 ms per image). In contrast, ViT's 86 M parameters and inference time exceeding 15 ms per image made it unsuitable for real-time industrial deployment.

In applications such as blade inspection, where hardware resources on drones or edge devices are limited, lightweight architectures are more favorable.

The findings underscore that lightweight CNNs are more effective than ViTs in data-constrained industrial environments, such as wind turbine blade defect detection. CNNs leverage their inherent inductive biases to generalize well even with limited training data, while ViTs require substantially larger datasets or hybrid architectures to achieve comparable performance.

The limitations associated with training ViTs from scratch—necessitated by hardware constraints—indicate that future research should evaluate pretrained or hybrid CNN–ViT architectures once resource availability permits.

Further validation across multiple datasets and real-world inspection environments will also enhance the generalizability of these findings.

Future work will explore hybrid CNN–Transformer architectures, transfer learning strategies, and domain adaptation methods to leverage the strengths of both paradigms. In addition, expanding the dataset with real-world inspection images and testing on other renewable energy infrastructure components will further validate the scalability and generalizability of the proposed approach.

Incorporating statistical significance analysis and repeated experimental trials will strengthen the reliability of future conclusions.

Author Contributions: Methodology, L.D.; writing—original draft preparation, L.D.; Validation, S.-H.L.; supervision, S.-H.L.; formal analysis, K.-M.L.; writing—review and editing, Y.-S.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Regional Innovation System & Education (RISE) program through the Jeollanamdo RISE center, funded by the Ministry of Education (MOE) and the Jeollanamdo, Republic of Korea (2025-RISE-14-004).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: The authors would like to thank all those who provided valuable comments and suggestions to improve the quality of this study.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhu, Y.; Liu, X. A Lightweight CNN for Wind Turbine Blade Defect Detection Based on Spectrograms. *Machines* **2023**, *11*, 99. [CrossRef]
2. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image Is Worth 16 × 16 Words: Transformers for Image Recognition at Scale. *arXiv* **2021**, arXiv:2010.11929.
3. Wang, G.; Li, Z.; Weng, G.; Chen, Y. An Overview of Industrial Image Segmentation Using Deep Learning Models. *Intell. Robot.* **2025**, *5*, 143–180. [CrossRef]
4. Gao, Z.; Liu, X. An Overview on Fault Diagnosis, Prognosis and Resilient Control for Wind Turbine Systems. *Processes* **2021**, *9*, 300. [CrossRef]
5. Alagha, N.; Khairuddin, A.S.M.; Haitaamar, Z.N.; Al-Khatib, O.; Kanesan, J. Artificial Intelligence in Wind Turbine Fault Detection and Diagnosis: Advances and Perspectives. *Energies* **2025**, *18*, 1680. [CrossRef]
6. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762.
7. Chen, C.-F.R.; Fan, Q.; Panda, R. CrossViT: Cross-Attention Multi-Scale Vision Transformer for Image Classification. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 347–356.
8. Liu, Y.; Zheng, Y.; Shao, Z.; Wei, T.; Cui, T.; Xu, R. Defect Detection of the Surface of Wind Turbine Blades Combining Attention Mechanism. *Adv. Eng. Inform.* **2024**, *59*, 102292. [CrossRef]
9. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *arXiv* **2019**, arXiv:1905.11946. [CrossRef]
10. Zhang, Z.; Dong, C.; Wei, Z.; Chen, X.; Zan, W.; Xue, Y. GCB-YOLO: A Lightweight Algorithm for Wind Turbine Blade Defect Detection. *Wind Energy* **2025**, *28*, e70029. [CrossRef]
11. Mehta, S.; Rastegari, M. MobileViT: Light-Weight, General-Purpose, and Mobile-Friendly Vision Transformer. *arXiv* **2022**, arXiv:2110.02178. [CrossRef]
12. Howard, A.; Sandler, M.; Chen, B.; Wang, W.; Chen, L.-C.; Tan, M.; Chu, G.; Vasudevan, V.; Zhu, Y.; Pang, R.; et al. Searching for MobileNetV3. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324. [CrossRef]

13. Liu, H.; Liu, S.; Liu, Z.; Niu, B.; Xie, J.; Luo, C.; Shi, Z. Wind Turbine Blade Surface Defect Detection Model Based on Improved You Only Look Once Version 10 Small and Integrated Compression. *Eng. Appl. Artif. Intell.* **2025**, *159*, 111645. [CrossRef]
14. Mansoor, M.; Tan, X.; Mirza, A.F.; Gong, T.; Song, Z.; Irfan, M. WindDefNet: A Multi-Scale Attention-Enhanced ViT-Inception-ResNet Model for Real-Time Wind Turbine Blade Defect Detection. *Machines* **2025**, *13*, 453. [CrossRef]
15. Wang, R.; Shao, Y.; Li, Q.; Li, L.; Li, J.; Hao, H. A Novel Transformer-Based Semantic Segmentation Framework for Structural Condition Assessment. *Struct. Health Monit.* **2024**, *23*, 1170–1183. [CrossRef]
16. Shorten, C.; Khoshgoftaar, T.M. A Survey on Image Data Augmentation for Deep Learning. *J. Big Data* **2019**, *6*, 60. [CrossRef]
17. Zhang, H.; Zhong, Z.; Duan, J.; Liu, G.; Hu, J. Design of Flow Velocity and Direction Monitoring Sensor Based on Fiber Bragg Grating. *Sensors* **2021**, *21*, 4925. [CrossRef] [PubMed]
18. Ibañez, I.; Forzani, L.; Tomassi, D. Generalized Discriminant Analysis via Kernel Exponential Families. *Pattern Recognit.* **2022**, *132*, 108933. [CrossRef]
19. Faddouli, A.; Labrim, H.; Fadili, S.; Habchi, A.; Hartiti, B.; Benaissa, M.; Hajji, M.; EZ-Zahraouy, H.; Ntsoenzok, E.; Benyoussef, A. Numerical Analysis and Performance Investigation of New Hybrid System Integrating Concentrated Solar Flat Plate Collector with a Thermoelectric Generator System. *Renew. Energy* **2020**, *147*, 2077–2090. [CrossRef]
20. Cubuk, E.D.; Zoph, B.; Shlens, J.; Le, Q.V. RandAugment: Practical Automated Data Augmentation with a Reduced Search Space. *arXiv* **2019**, arXiv:1909.13719. [CrossRef]
21. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. *arXiv* **2021**, arXiv:2103.14030. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.