Iraqi Journal for Computers and Informatics
Available online: https://ijci.uoitc.edu.iq/index.php/ijci
Volume 52, issue 1. 2026

Print ISSN: 2313-190X, Online ISSN: 2520-4912

Lightweight CNN-Based Framework for Industrial Surface Defect Classification

Ahmeed Suliman Farhan [a], Ali Al-kubaisi [b], Ahmed Majid Taha [c, d]

[a] *Electronic Computer Center, University of Anbar, Al-Ramadi, Iraq.*

[b] *Computer Sciences and Information Technology, University of Anbar, Al-Ramadi, Iraq.*

[c] *College of Biomedical Informatics, University of Information Technology and Communications, Baghdad, Iraq.*

[d] *Soft Computing and Data Mining Center, Universiti Tun Hussein Onn Malaysia 86400 Parit Raja, Batu Pahat Johor, Malaysia.*

ARTICLEINFO

Keywords:

Industrial Surface Defect Classification, Convolutional Neural Networks (CNN), Deep Learning, Hyperparameter Tuning, Grad-CAM

ABSTRACT

Industrial surface defect classification is an important part of automated quality inspection systems. For these systems to work, they need to be able to detect surface defects accurately and efficiently to improve product reliability and reduce manufacturing costs. Traditional manual inspection methods are often time-consuming, subjective, and not suitable for fast-paced industrial environments. This study proposes a lightweight convolutional neural network (CNN)-based system for classifying industrial surface defects. The model was made to work well for classification and still be fast enough for real-world use. Keras Tuner was used in the proposed method to find the best hyperparameters. The proposed model was evaluated on the Northeastern University (NEU) Surface Defect Database using 5-fold stratified cross-validation. The results obtained from the experiments are promising since the system yields stable performance on all folds with accuracy scores of 99.72%, 99.17%, 100.00%, 99.17%, and 98.61% respectively. The mean accuracy score is calculated as 99.33%. Also, Grad-CAM visualization revealed that the network focuses on defective regions when processing an input image which supports the reliability and interpretability of the classification process.

1. INTRODUCTION

With the significant advancements in industry in general, the importance of inspecting industrial surfaces for defects has become apparent, as these defects affect product quality and reliability. For example, in the steel industry, surface defects are not only aesthetically undesirable but can also impact quality and durability. Therefore, defect inspection is essential from both an engineering and economic perspective. Recent studies have confirmed that traditional (manual) inspection is slow and time-consuming, especially on high-speed production lines. Furthermore, it is costly and relies heavily on the operator's experience and precision. [1], [2]. To overcome these limitations, automated inspection based on intelligent systems is employed. Recent studies indicate that deep learning techniques have become essential in industrial defect inspection because they are able to automatically learn features from images and are adaptable for classification, detection, and segmentation tasks.

E-mail address:
ahmeedsuliman@uoanbar.edu.iq [a]
alijamal530@uoanbar.edu.iq [b]
dr.ahmed_majid@uoitc.edu.iq [c]

Corresponding* : *Ahmeed Suliman Farhan*

Received 26 March 2026, Accepted May 2026

DOI: 10.25195/ijci.v52i1769.

Convolutional neural networks (CNNs) have emerged as one of the most widely used methods due to their ability to extract hierarchical features [3], [4].

Despite significant advances in industrial image inspection, it remains challenging. Industrial images often vary in dimensions and geometries, and detectable defects can be highly diverse. Furthermore, imaging parameters frequently fluctuate, complicating model training. Acquiring a sufficiently large training dataset is demanding in industrial settings. Additionally, class imbalance presents a significant challenge because certain defect types are far more prevalent than others [4], [5], [2].

In real-world environments, model efficiency is also crucial. While some large deep learning models may achieve high predictive performance, they require significant computing resources and may be difficult to deploy in real-world industrial environments where real-time processing is critical. Therefore, researchers in recent studies have focused on developing models with low computational complexity while maintaining performance. For example, Jianbo Lu et al. proposed the SS-YOLO model, which is lightweight, has a low computational cost, and works to detect defects in steel strips, while other approaches, such as YOLO-SDS and DCA-YOLO, which were proposed by Yuqun Chu et al. and He Xu et al., respectively, aim to improve the balance between detection accuracy and inference speed under industrial constraints [6]–[8].

Another issue that has gained increasing attention in recent years is the explainability of model decision-making. In industrial applications, engineers and supervisors need evidence that model predictions are accurate and that the model made a decision based on how and why. Furthermore, model explanation in industrial applications enhances transparency, supports human decision-making, and helps engineers understand the logic behind decisions. Therefore, visual interpretation methods become increasingly important in industrial inspection studies [9], [10].

Motivated by these challenges, this study proposes a lightweight, convolutional neural network (CNN)-based framework for classifying industrial surface defects using the Northeastern University (NEU) surface defect dataset. The proposed model includes hyperparameter optimisation, stepwise validation, and explainability analysis. Instead of emphasizing architectural complexity, the proposed model prioritizes computational efficiency and classification accuracy. Also, the explainable model for decision-making processes makes it suitable for practical industrial applications. This design aligns with current research trends emphasizing efficient deployment and reliable, accurate evaluation.

The main contributions of this work can be summarized as follows:

1. A lightweight CNN-based framework is developed for multiclass industrial surface defect classification using the NEU surface defect dataset.
2. Hyperparameter optimization is performed using Keras Tuner to systematically determine the most effective model configuration rather than relying on manual trial-and-error tuning.
3. The proposed model is evaluated using five-fold stratified cross-validation to provide a reliable and unbiased assessment of classification performance.
4. Grad-CAM-based explainability analysis is employed to visualize the regions influencing the model predictions and improve the transparency of the defect classification process.
5. The hyperparameter search space includes a Transformer block, allowing for a data-driven architecture comparison that empirically proves the superiority of the optimized lightweight CNN to the hybrid CNN-Transformer for localized texture defect classification.

2. Related Work

In recent years, research in industrial surface defect detection has increasingly relied on deep learning models. The deep learning approaches have supported more efficient automatic extraction of features and improved the performance. The research on industrial surface defect detection at the present time focuses on three major areas: 1) classification, 2) object detection, and 3) segmentation. Convolutional neural networks (CNNs) are still used in most recent studies because of their high feature extraction. Although this advancement, this field still faces considerable challenges: 1) a data balance problem between categories, 2) small defect area sizes, a lack of boundary clarity, and 3) a lack of a classified dataset.

Within the field of classification, Many recent studies have concentrated on finding approaches to extract distinctive defect characteristics using relatively small reference datasets due to the difficulty of acquiring large datasets in industrial applications. In these circumstances, Abdul Majeed et al. applied the feature-based transfer learning approach using the NEU surface defects database. The results of their study verified that transfer learning is an effective option for classifying multi-

category steel surface defects. Mainly when a restricted amount of classified data is available [11]. In a contrasting industrial application, Kim et al. introduced a defect-adaptive hierarchical convolutional neural network (DHS-CNN) model established on InceptionV4 for contact lens defect detection. The results showed that the application of custom convolutional architectures led to improved classification reliability in industrial quality control applications that need a high degree of accuracy [12]. In another study, Lopez et al. proposed a VGG-inspired convolutional neural network for welding defect classification, incorporating Grad-CAM++ technology to improve the model's interpretability. This reflects the growing interest in industrial applications using convolutional neural network-based models that not only obtain high accuracy but also try to enhance the reliability and explainability of the results [13].

Another important area of research is how to develop a lightweight model to find defects in real time. Lu et al. have suggested an enhanced model, which is a lightweight strip steel defect detector based on the YOLOv7 model, ensuring minimum computational costs with optimal performance [14]. Moreover, Wang et al. have suggested a lightweight model based on the YOLOv8n model to find defects on the surface of bearings [15]. Recently, Xu et al. have suggested a new model, DCA-YOLO, which is a lightweight steel surface defect detector with a nanoscale structure, having 4.3 million parameters and 9.4 GFLOPs. They have achieved better results in detecting defects with minimum computational cost [8]. Chen et al. have suggested an enhanced model, which is a lightweight YOLOv9 model to detect defects on the surface of steel with NEU-DET. To make the model less complicated, they have used depthwise separable convolution to make the model efficient [16]. These studies show that the area of lightweight models has become the most important area in industrial vision research.

Several researchers worked on the segmentation technique for industrial defects. Park et al. proposed NC-Net, a DCNN with attention and routing units. It was designed to segment industrial surface defects and progress defect localization in environments with complex backgrounds [17]. More studies have confirmed this trend, such as the AMSFF-Net model. It depends on adaptive multiscale feature integration for addressing the challenges posed by the variety of defect shapes and background complexity in industrial surface images [18]. Segmentation methods need more accurately annotated data and incur more computational costs. This situation which may not be suitable for all industrial production lines. Therefore, classification-based methods remain a feasible option when only the type of defect is required. This balance is also a concern in recent reviews that differentiate between industrial inspection strategies provided towards classification and those provided towards localization or segmentation [17], [19].

Explainable AI has become an increasing concentration in industrial AI applications. Cação et al. consult a field of Explainable AI techniques applied in industrial fault diagnosis. They observe that transparency in machine learning models is crucial for constructing confidence in intelligent systems and supporting diagnostics and decision validation within production environments [13]. Similarly, recent reviews of explainable AI in manufacturing indicate that explainable techniques are very important because they help engineers verify if the model predictions are based on physically meaningful regions or on incorrect correlations in the data [20], [21]. In applied studies, Grad-CAM-based visualization techniques have been used to analyze regions of attention in models during defect classification tasks, including weld inspection and other industrial imaging applications. This procedure leads to improved understanding of model behavior and enhanced accuracy of results [13], [22].

Despite significant advancements in recent studies related to industrial defect detection, a research gap remains that requires further attention. Most research has focused on detection or segmentation tasks. However, multi-class classifiers based on convolutional neural networks have received less attention, especially with hyperparameter optimisation, cross-validation, and explainable techniques. Furthermore, modern lightweight models often prioritise balancing execution speed with an acceptable detection accuracy. However, it did not always include rigorous cross-validation evaluation or provide explainability for the model's decision-making. Therefore, this study aims to bridge this gap by proposing a lightweight classification model based on a convolutional neural network. This model incorporates hyperparameter optimization using Keras Tuner and performance evaluation using five-fold stratified cross-validation. Also, the model used the Grad-CAM method for explainability outputs and analysed regions associated with classification decisions.

3. Methodology

The proposed architecture represents a simple, lightweight CNN model for surface defect classification on industrial products. The model aims to deliver optimal results in both performance and computational resource usage. It includes multiple stages: data processing, feature extraction, and classification. First, images are rescaled to 224 × 224 to match input layer requirements. Next, pixel values are normalised within [0, 1] to stabilise training. Data augmentation is used during training to prevent overfitting. Magnifications include microrotations, panning, zooming, horizontal flips, and brightness adjustment. Images then pass through multiple layers, including convolutional layers that extract different feature hierarchies. All convolutional blocks use the same design. Each block includes two 3×3 convolutions, a batch normalisation layer, ReLU activation, and max-pooling to reduce spatial data. The number of convolutional units and the number of filters in each unit are not manually determined but are automatically determined through hyperparameter optimization. The number of convolutional units was chosen from the set {1, 2, 3}, and three convolutional units achieved the best results during the tuning process, as shown in Table 1.

Also, the number of filters in each block follows a progressive scaling rule defined as:

$$f_i = conv\_filters \times 2^i \quad (1)$$

Where i is the index of the block. This means that the number of filters grows as the network gets deeper. For example, the tuner chooses 48 as the base number of filters, the configuration changes to: (Block 1 → 48 filters, Block 2 → 96 filters, Block 3 → 192 filters). This gradual increase lets the network give deeper layers more representational capacity, which lets it pull out more complicated features. Figure 1 shows the overall design of the proposed lightweight CNN model. It shows the order of the convolutional blocks, the global average pooling that combines features, and the final classification layers.

During the hyperparameter optimization stage, the search space also had an optional Transformer block. This block was meant to find global contextual relationships in the feature maps. This block comes after the convolutional feature extraction stage, which turns the feature maps into patch embeddings and then processes them through self-attention layers. There are (Multi-head self-attention, Feed-forward network, Layer normalization, and Residual connections) in each transformer block.

Even though it was part of the search space, the experimental results show that the transformer component was not chosen in the best way. In particular, the trial that did the best had the highest validation accuracy with **n_transformer_blocks = 0**. This indicates that the optimal configuration did not require transformer blocks.

There are a number of things that could explain this behavior. First, the NEU dataset has small images with localized defect patterns that convolutional operations can easily capture. Second, transformer layers make things more complicated to compute and need bigger datasets to get the most out of them. Third, real-world tests showed that setups with transformer blocks didn't make validation accuracy better and, in some cases, made performance worse. The final model chosen is a lightweight CNN, even though the original design allowed for hybrid CNN with a Transformer.

After feature extraction, the resulting features are passed through Global Average Pooling (GAP) to reduce the number of trainable parameters and improve the model's generalizability. The GAP results are then passed through a fully connected layer containing 512 neurons with a ReLU activation function. To mitigate the overfitting problem, a Dropout layer with a 0.4 dropout rate is applied before the final classification stage. The last layer of the network is the Softmax classifier, which outputs the probability distribution across the six surface defect classes. The final model contains 847,670 trainable parameters (≈0.85M), confirming its lightweight relative to comparable models such as DCA-YOLO, which contains 4.3 million parameters.

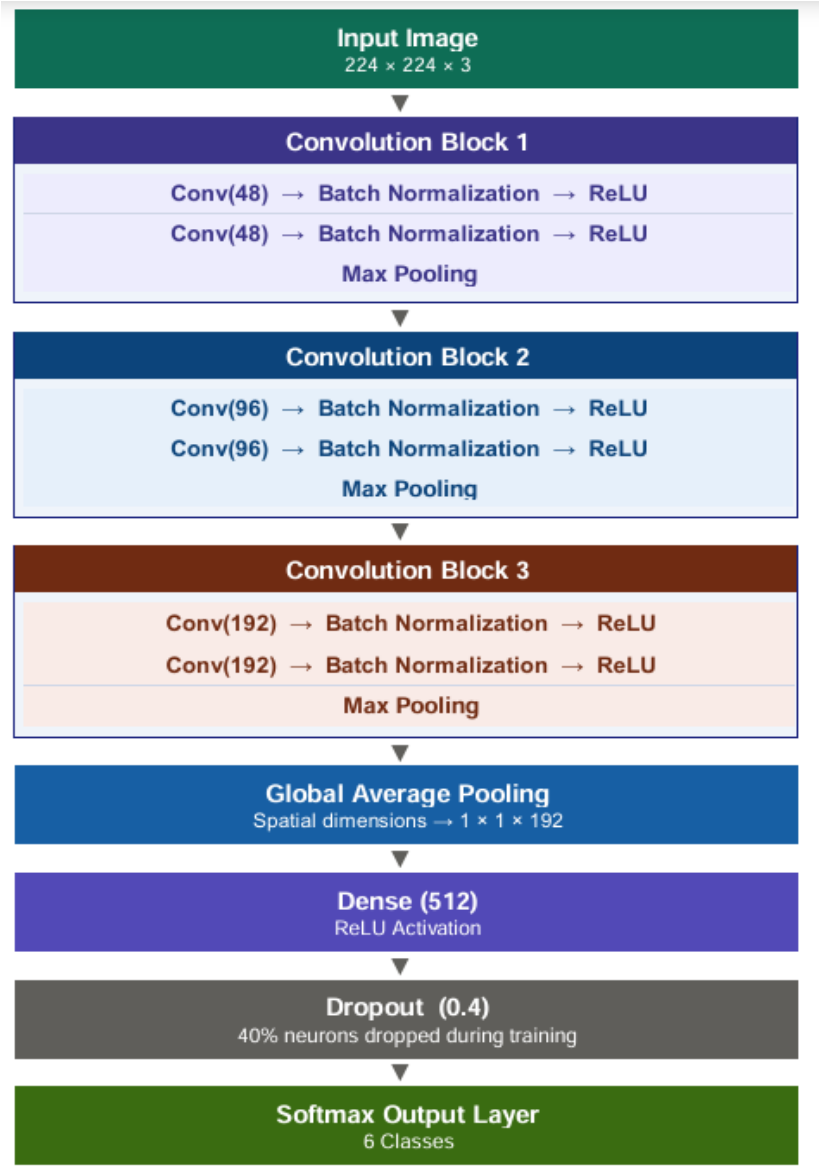

Figure 1: Architecture of the proposed model.

4. Experimental Results

This section presents the experimental evaluation of the proposed model. It includes a description of the dataset, evaluation metrics, hyperparameter optimization using Keras Tuner, and analyses of classification performance, durability across cross-validation folds, and explainability.

4.1. Dataset

The experiments were conducted using the NEU Surface Defect Dataset, which is a widely used dataset for steel surface defect classification. The dataset contains grayscale images with six classes (Crazing, Inclusion, Patches, Pitted Surface, Rolled-in Scale, Scratches). Each category contains 300 images, resulting in a total of 1800 images [23]. Before training the model, all images were resized to 224 × 224 pixels to match the input size of the proposed model. Furthermore, each pixel was normalized to the range [0,1] by dividing the image values by 255.

4.2. Evaluation Metrics

To evaluate the proposed model, we used four common evaluation metrics—Accuracy, Precision, Recall, and F1-score— to evaluate the classification performance of the proposed model. Accuracy tells you how many of the total samples were correctly classified. Precision looks at the ratio of correctly predicted positive samples to all predicted positive samples. Recall is the percentage of correctly predicted positive samples compared to the actual positive samples. The F1-score is the harmonic mean of Precision and Recall, and it gives a balanced view of how well a classification works [24]. The metrics for evaluation are defined as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (5)$$

where **TP**, **TN**, **FP**, and **FN** denote the numbers of true positives, true negatives, false positives, and false negatives, respectively.

4.3. Hyperparameter Optimization

Hyperparameter optimization was done with Keras Tuner and the Random Search strategy to find the best model configuration. The tuner didn't pick the architectural parameters by hand. Instead, it tried out different combinations of hyperparameters to find the one that worked best for validation. There were 30 separate trials in the tuning process, and each trial was a full training experiment with a different set of hyperparameters. We trained each trial for up to 30 epochs and kept track of the best validation accuracy we got during training. Table 1 shows all of the hyperparameter trials, including the chosen values and the validation accuracy that goes with them.

Table 1: *Complete results of the 30 hyperparameter tuning trials using Keras Tuner.*

Trial	Filters	Blocks	Dropout	Learning Rate	Transformer Blocks	Val Accuracy
1	32	1	0.3	0.001	1	0.926
2	32	3	0.3	0.0005	1	0.912
3	32	1	0.2	0.001	3	0.883
4	48	1	0.2	0.001	0	0.874
5	32	2	0.1	0.001	0	0.885
6	48	1	0.2	0.0005	3	0.890
7	48	2	0.2	0.0001	2	0.947
8	32	1	0.3	0.001	2	0.925
9	48	2	0.2	0.0005	0	0.957
10	32	1	0.3	0.001	1	0.883
11	48	3	0.3	0.0005	2	0.856
12	48	3	0.3	0.001	3	0.836
13	48	1	0.1	0.001	2	0.903
14	32	2	0.2	0.0005	2	0.883
15	32	1	0	0.0005	2	0.969
16	48	1	0.1	0.0005	3	0.958
17	32	1	0.2	0.0005	1	0.918
18	48	2	0.2	0.0001	3	0.967
19	32	2	0	0.0001	3	0.947
20	16	2	0.3	0.001	2	0.878
21	16	1	0.1	0.001	1	0.892
22	48	1	0.1	0.001	2	0.897
23	16	1	0.2	0.0001	2	0.953
24	48	1	0	0.0005	0	0.851

25	32	1	0.1	0.001	0	0.851
26	32	3	0.1	0.0005	3	0.893
27	16	1	0.1	0.0001	1	0.971
28	**48**	**3**	**0.4**	**0.0001**	**0**	**0.986**
29	48	2	0.4	0.001	0	0.968
30	48	2	0.4	0.0005	0	0.946

From the hyperparameter search results, several important observations can be made.

1- The number of convolution units affects model performance. Configurations with only one or two convolution units were generally insufficient to capture all complex defect patterns, while three-unit constructs achieved higher validation accuracy.
2- Incremental filter scaling proved effective. Increasing the number of filters in deeper network layers allowed for learning more features, thus improving classification accuracy.
3- Incorporating Transformer blocks did not improve performance. Multiple experiments using Transformer blocks resulted in lower validation accuracy. This confirms that, for the NEU dataset, extracting local features is more important than global attention modeling.
4- Low learning rates, such as 0.0001, led to more stable training and better convergence, while appropriate dropout values helped reduce overfitting.

The best-performing configuration corresponds to **Trial 28**, which achieved a validation accuracy of **98.61%** with this configuration:

- 3 convolutional blocks
- 48 → 96 → 192 filters
- Dropout = 0.4
- Learning rate = 0.0001
- No transformer blocks

4.4. Results

We used the NEU surface defect dataset to train and test the proposed model using five-fold stratified cross-validation. The results show that the suggested model worked well in all folds. Table 2 shows that fold 3 had the highest score. This means that the model got 100% accuracy, which means that it perfectly classified all of the validation samples. Fold 5 had the worst performance, but it still got 98.61% accuracy, which shows that the model is reliable across all validation folds.

Table 2: *summarizes the classification performance of the proposed model across the five folds.*

Fold	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Fold 1	99.72	99.73	99.72	99.72
Fold 2	99.17	99.17	99.17	99.16
Fold 3	100.00	100.00	100.00	100.00
Fold 4	99.17	99.19	99.17	99.17
Fold 5	98.61	98.72	98.61	98.61

Three important observations can be made by looking at the results in Table 2. First, the model is more than 98.6% accurate across all folds. Second, the difference between the best and worst fold results isn't very big, which shows that the model's performance is stable. Third, the fact that fold 3 had the best performance shows that the model got enough features. The small drop in performance in fold 5 means that there are a few more difficult samples with patterns that are more difficult to classify.

The training and validation curves for the five folds are illustrated in Figure 2. The accuracy and loss curves for both the validation and the training sets have been included. From the figure below, it can be observed that the training accuracy curve shows rapid learning at the initial stages and flattens after reaching a close-to-saturated level. This implies that the model

learns visual patterns in surface defect images quickly. Additionally, it can be observed that the training loss curve falls steadily. This indicates that there is stable optimization being achieved. The validation accuracy curves have a similar trend to the training accuracy curves and are closer to the values of the training accuracy curves.

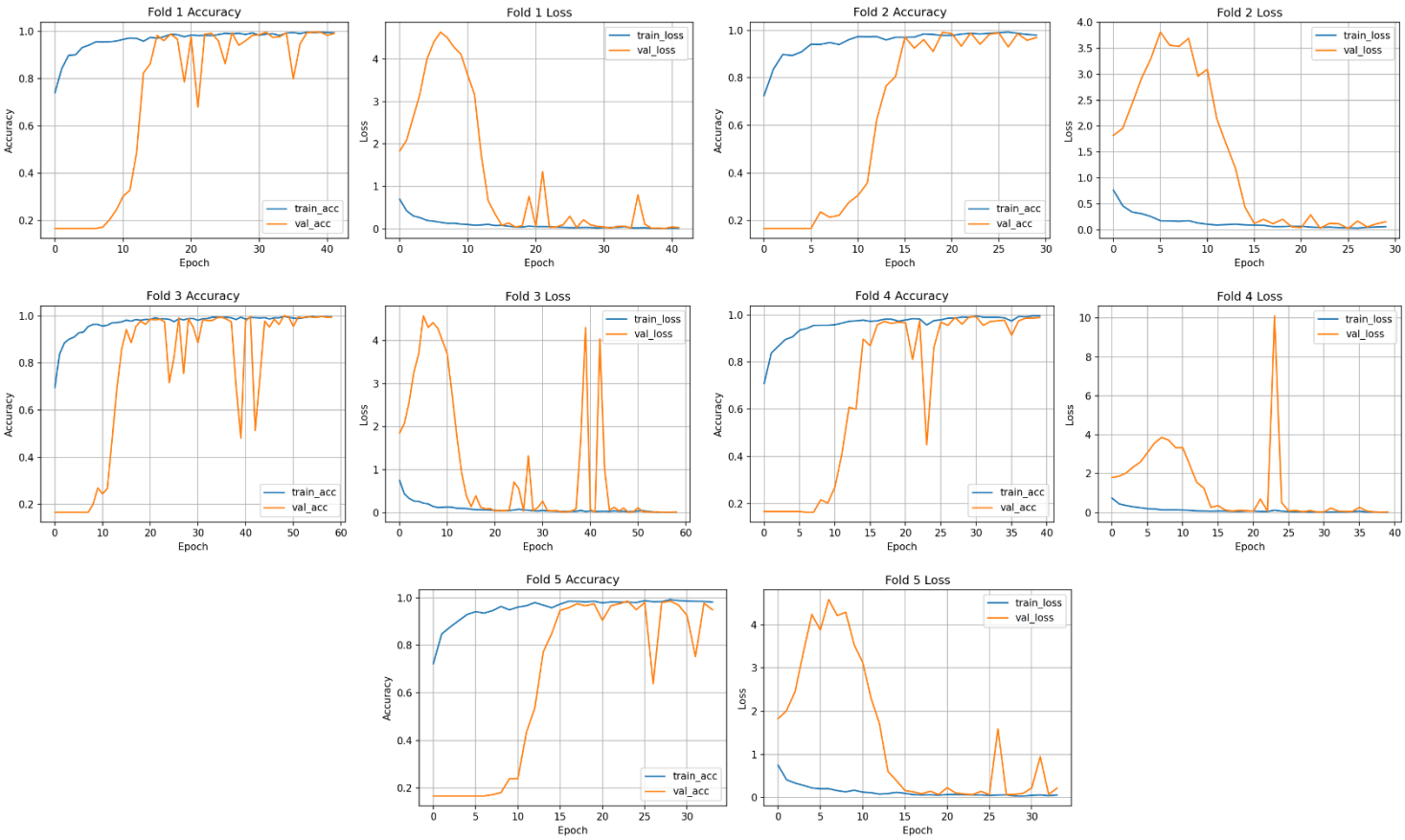

Figure 2: Training and validation accuracy and loss curves for the five cross-validation folds.

Figure 3 shows the confusion matrices for all five cross-validation folds. This shows how accurate the classification is for each of the six defect types. The matrices show a strong diagonal pattern, which means that most of the samples are correctly classified in all folds.

5. Explainable Artificial Intelligence (XAI)

We used the Grad-CAM method for explainability to highlight the regions of the input image on which the model focuses, and how these regions contribute to the model's decision-making. This method looks for the target class gradients in relation to the feature maps from the last layer of convolution. It makes a rough map of where each area is in relation to the decision-making process. This method lets you look at the model to see if it is focusing on real defect areas or areas that don't matter. In this work, we used the standard formula for Grad-CAM that was shown in [25].

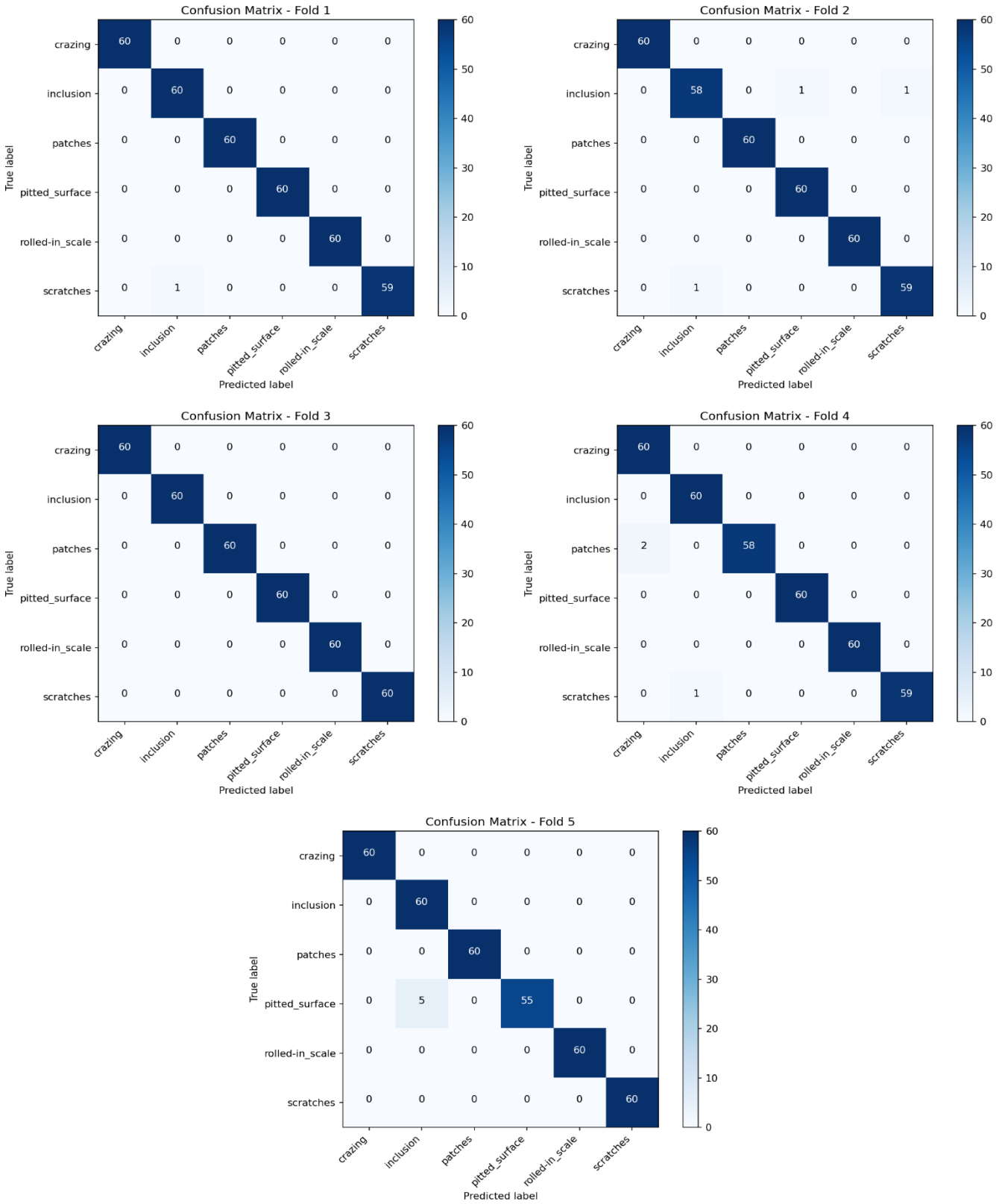

Figure 3: Confusion matrices for the five cross-validation folds.

Figure 4 shows the Grad-CAM heat map for samples from all classes. The heat maps show that the model focuses on important areas such as cracks, inclusions, and surface irregularities. For example, in images containing scratch or crack defects, the highlighted regions indicate where these defects are located. Furthermore, for pitted surface imperfections and defects, the attention maps focus on surface distortions. These results show that the proposed model is explainable because its predictions

are based on the right defect features. Therefore, using Grad-CAM adds another level of reliability and transparency, which is important for industrial uses.

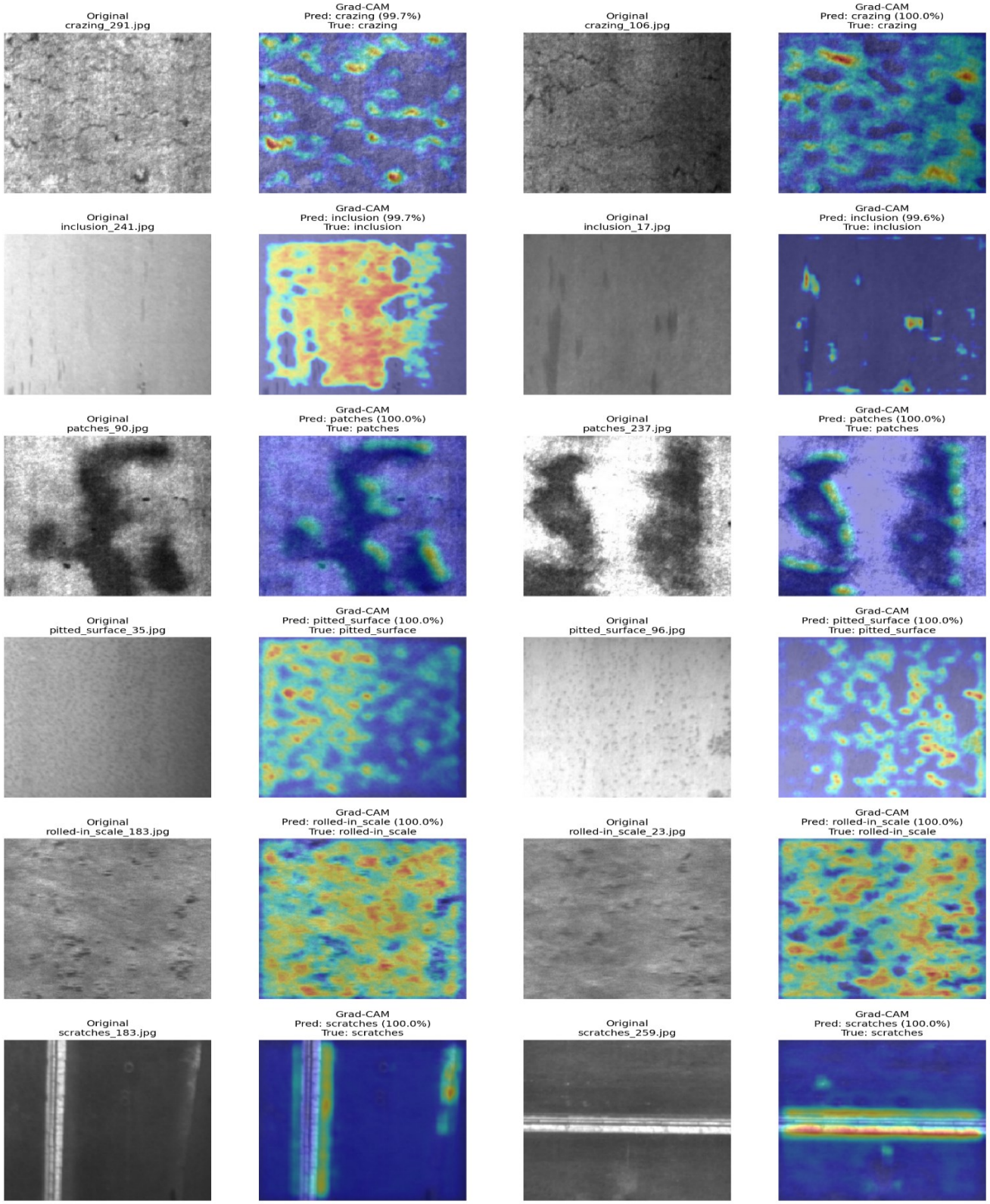

Figure 4: Grad-CAM visualization of the proposed model for different surface defect categories.

6. Discussion

The experimental results showed that the proposed model achieved a good performance on the NEU surface defects dataset. This study proved that lightweight CNN structures with hyperparameter optimization can achieve good performance while maintaining computational efficiency. Table 1 shows the hyperparameter optimization results of the model, which showed that different configurations significantly impact the results. Where the number of convolution blocks and the filters affect the outcomes. The best configuration was a three-block structure with increasing filter depth (48 → 96 → 192), which allows the model to get complex hierarchical features. This proves that deeper layers are necessary to increase the model's feature capture.

An important note is that adding transformer blocks did not improve the model's performance. Instead of following the common practice of relying on transformers, feature extraction using convolutional neural networks (CNNs) proved sufficient and more effective on the NEU dataset. The result of using transformers achieved lower accuracy, demonstrating that additional complexity is not necessarily better performance. Therefore, selecting the model's structure and components based on the features of the dataset is important.

The model achieved a good performance with an average accuracy of 99.33%. Furthermore, the difference between the best-performing fold (100%) and the worst-performing fold (98.61%) was minimal. These results confirm that the model is not over-specific to any particular data segment and can be generalised.

Furthermore, the proposed model used Grad-CAM method for explainability. The Grad-CAM heat maps confirm the model focuses on defect-related regions such as cracks, surface irregularities, and distortions. The explainability of the model is important in industrial applications for supporting human decision-making and building trust.

Overall, the proposed method achieves a balance between accuracy, efficiency, and explainability, making it suitable for industrial applications.

7. Conclusion

In this study, we proposed a new lightweight convolutional neural network (CNN) model for classifying industrial surface defects from the NEU surface defect dataset. The aim of the model is to maintain computational efficiency with good performance. The model contains a series of convolutional blocks with progressively increasing filter depths. We used Keras Tuner to optimise the configuration and hyperparameters of the model. This optimisation resulted in multiple configurations and the extraction of the most efficient structural model. The model was evaluated on the NEU surface defect dataset using five-fold stratified cross-validation. The model achieved an average accuracy of 99.33% with a small difference between the folds , indicating the model's generalizability. Also, the model convergence was stable during training. Furthermore, we used the Grad-CAM for explainability, which provided insight into the model's decision-making process. The experimental result proves the lightweight structure of the convolutional neural network with hyperparameter optimisation can achieve performance comparable to more complex models. In future work, we could expand the scope of the proposed model by evaluating it on larger and more diverse datasets.

REFERENCES

[1] A. A. M. Ibrahim and J.-R. Tapamo, "A survey of vision-based methods for surface defects' detection and classification in steel products," in Informatics, vol. 11, no. 2. MDPI, 2024, p. 25.

[2] K. Frydrych, M. Tomczak, J. Jasi´nski, and S. Papanikolaou, "Steel surface defects analysis with machine vision and deep learning," The International Journal of Advanced Manufacturing Technology, vol. 140, no. 7, pp. 3691–3710, 2025.

[3] R. Ameri, C.-C. Hsu, and S. S. Band, "A systematic review of deep learning approaches for surface defect detection in industrial applications," Engineering Applications of Artificial Intelligence, vol. 130, p. 107717, 2024.

[4] Y. Ma, J. Yin, F. Huang, and Q. Li, "Surface defect inspection of industrial products with object detection deep networks: A systematic review," Artificial Intelligence Review, vol. 57, no. 12, p. 333, 2024.

[5] D. Bai, G. Li, D. Jiang, J. Yun, B. Tao, G. Jiang, Y. Sun, and Z. Ju, "Surface defect detection methods for industrial products with imbalanced samples: A review of progress in the 2020s," Engineering Applications of Artificial Intelligence, vol. 130, p. 107697, 2024.

[6] J. Lu, M. Yu, and J. Liu, "Lightweight strip steel defect detection algorithm based on improved YOLOv7," Scientific reports, vol. 14, no. 1, p. 13267, 2024.

[7] Y. Chu, X. Yu, and X. Rong, "A lightweight strip steel surface defect detection network based on improved YOLOv8," Sensors, vol. 24, no. 19, p. 6495, 2024.

[8] H. Xu, Z. Zhang, H. Ye, J. Song, and Y. Chen, "Efficient steel surface defect detection via a lightweight yolo framework with task-specific knowledge-guided optimization," Electronics, vol. 14, no. 10, p. 2029, 2025.

[9] J. Cação, J. Santos, and M. Antunes, "Explainable AI for industrial fault diagnosis: A systematic review," *Journal of Industrial Information Integration*, vol. 47, p. 100905, 2025.

[10] G. Tzionis, P. Mouratidis, G. Kougka, I. Gialampoukidis, S. Vrochidis, I. Kompatsiaris, and M. Vlachopoulou, "A review of explainable AI methods and their application in manufacturing systems," *Discover Applied Sciences*, vol. 8, no. 1, Art. no. 52, 2026.

[11] A. P. Abdul Majeed, M. A. Abdullah, A. F. Ab. Nasir, M. A. Mohd Razman, W. Chen, and E. H. Yap, "Surface defect detection: a feature-based transfer learning approach," in Journal of Physics: Conference Series, vol.2762, no. 1. IOP Publishing, 2024, p. 012088.

[12] S.-H. Kim, S.-J. Joo, and K.-H. Yoo, "Dhs-cnn: A defect-adaptive hierarchical structure cnn model for detecting anomalies in contact lenses," Applied Sciences, vol. 15, no. 5, p. 2697, 2025.

[13] G. L´opez, P. D. Ram´ırez, E. Vega, F. Pizarro, J. Toro, and C. Parra, "Weldvgg: a vgg-inspired deep learning model for weld defect classification from radiographic images with visual interpretability," Sensors, vol. 25, no. 19, p. 6183, 2025.

[14] J. Lu, M. Zhu, X. Ma, and K. Wu, "Steel strip surface defect detection method based on improved YOLOv5s," Biomimetics, vol. 9, no. 1, p. 28, 2024.

[15] Y. Wang, Z. Song, H. S. Abdullahi, S. Gao, H. Zhang, L. Zhou, and Y. Li, "A lightweight detection algorithm for surface defects in small-sized bearings," Electronics, vol. 13, no. 13, p. 2614, 2024.

[16] C. Chen, H. Lee, and M. Chen, "Steel surface defect detection method based on improved YOLOv9," Scientific reports, vol. 15, no. 1, p. 25098, 2025.

[17] K.-B. Park and J. Y. Lee, "Novel industrial surface-defect detection using deep nested convolutional network with attention and guidance modules," Journal of Computational Design and Engineering, vol. 9, no. 6, pp. 2466–2482, 2022.

[18] G. Wen, L. Cheng, H. Yuan, and X. Li, "Surface defect detection based on adaptive multi-scale feature fusion," Sensors, vol. 25, no. 6, p. 1720, 2025.

[19] H. Mewada, I. M. Pires, P. Engineer, and A. V. Patel, "Fabric surface defect classification and systematic analysis using a cuckoo search optimized deep residual network," Engineering Science and Technology, an International Journal, vol. 53, p. 101681, 2024.

[20] A. Sieradzki, J. Bednarek, A. Jegorowa, and J. Kurek, "Explainable ai (xai) techniques for convolutional neural network-based classification of drilled holes in melamine faced chipboard," Applied Sciences, vol. 14, no. 17, p. 7462, 2024.

[21] G. Wang, Z. Li, G. Weng, and Y. Chen, "An overview of industrial image segmentation using deep learning models," Intelligence & Robotics, vol. 5, no. 1, pp. 143–180, 2025.

[22] R. A. Saleh, F. Al-Areqi, M. Z. Konyar, K. Kaplan, S. Öngir, and H. M. Ertunc, "Advancing tire safety: Explainable artificial intelligence-powered foreign object defect detection with Xception networks and Grad-CAM interpretation," *Applied Sciences*, vol. 14, no. 10, p. 4267, 2024.

[23] K. Song and Y. Yan, "A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects," Applied Surface Science, vol. 285, pp. 858–864, 2013.

[24] A. S. Farhan, M. Khalid, and U. Manzoor, "Prcnet: An efficient model for automatic detection of brain tumor in mri images," Plos one, vol. 20, no. 12, p. e0292768, 2025.

[25] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in Proceedings of the IEEE international conference on computer vision, 2017, pp. 618–626.