

Article

Hybrid-DC: A Hybrid Framework Using ResNet-50 and Vision Transformer for Steel Surface Defect Classification in the Rolling Process

Minjun Jeong ¹, Minyeol Yang ^{1,2} and Jongpil Jeong ^{1,*}

¹ Department of Smart Factory Convergence, Sungkyunkwan University, 2066 Seobu-ro, Jangan-gu, Suwon 16419, Republic of Korea; jmj9613@g.skku.edu (M.J.); yangget@g.skku.edu (M.Y.)

² Research & Development Team, THIRA-UTECH Co., Ltd., CK Bldg 4F Hakdong-ro 5-gil, 7 Gangnam-gu, Seoul 06044, Republic of Korea

* Correspondence: jpjeong@skku.edu

Abstract: This study introduces Hybrid-DC, a hybrid deep-learning model integrating ResNet-50 and Vision Transformer (ViT) for high-accuracy steel surface defect classification. Hybrid-DC leverages ResNet-50 for efficient feature extraction at both low and high levels and utilizes ViT's global context learning to enhance classification precision. A unique hybrid attention layer and an attention fusion mechanism enable Hybrid-DC to adapt to the complex, variable patterns typical of steel surface defects. Experimental evaluations demonstrate that Hybrid-DC achieves substantial accuracy improvements and significantly reduced loss compared to traditional models like MobileNetV2 and ResNet, with a validation accuracy reaching 0.9944. The results suggest that this model, characterized by rapid convergence and stable learning, can be applied for real-time quality control in steel manufacturing and other high-precision industries, enhancing automated defect detection efficiency.

Keywords: ResNet-50; vision transformer (ViT); hybrid attention layer; defect classification; deep learning; computer vision; automated inspection systems



Citation: Jeong, M.; Yang, M.; Jeong, J. Hybrid-DC: A Hybrid Framework Using ResNet-50 and Vision Transformer for Steel Surface Defect Classification in the Rolling Process. *Electronics* **2024**, *13*, 4467. <https://doi.org/10.3390/electronics13224467>

Academic Editor: Marcin Witczak

Received: 24 September 2024

Revised: 7 November 2024

Accepted: 12 November 2024

Published: 14 November 2024



Copyright: © 2024 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The steel manufacturing industry is an essential and irreplaceable foundation of modern industry, with steel products playing an important role in a wide range of fields, including automobiles, electronics, furniture, weapons, household materials, machinery, and ships [1]. The quality of steel products directly affects the performance and service life of the final products. Surface defects, which not only affect the appearance but also have a significant impact on durability, corrosion resistance, and strength [2], are of particular importance. These defects occur mainly during production and molding processes and can degrade product quality and cause economic losses [3]. Traditionally, steel surface defects were detected manually and classified by experts, but these methods are time-consuming and can produce inconsistent results, depending on the experience of the inspector [1]. Machine-vision methods have been used to address these issues. For example, Yue et al. [4] proposed a rail surface defect classification method based on AdaBoost. Aslam et al. [5] used fuzzy clustering to improve the accuracy of the recognition of small defects and cracks on metal surfaces. Chu et al. [6] combined support vector machines and binary trees to classify surface defects in steel. However, all these methods rely heavily on feature engineering and are only suitable for recognizing simple and clear surface defects, resulting in relatively poor performance in more complex scenarios [7]. More recently, with advances in deep-learning technology, automated defect detection and classification systems have attracted attention and are becoming powerful tools for more accurate detection of different defect types, particularly in steel manufacturing [3].

Previous deep-learning-based studies primarily used Convolutional Neural Networks (CNNs) to classify steel surface defects. CNNs learn local patterns, capture promising

semantic information, and are more efficient compared with other types of models. They are preferred over other types of deep-learning models, such as artificial neural networks and recurrent neural networks [8]. Fu et al. [9] proposed SqueezeNet, a lightweight CNN-based model, to classify steel surface defects in real time. Zheng et al. [10] improved the YOLOv3 tool to provide an end-to-end solution for steel surface defect detection. Damacharla et al. [11] used U-Net to perform transfer-learning-based segmentation of steel surfaces. These deep-learning-based approaches have been highly influential in defect detection research and have advanced the field. However, while CNNs are effective at learning local patterns in images to capture meaningful information, models with a single neural network cannot handle the complex shapes and varying sizes of defects [8]. Furthermore, they are typically designed for natural datasets, such as COCO and PASCAL VOC, and cannot achieve optimal performance when used directly for defect detection due to the inherent differences in characteristics between natural and defect images [12]. In particular, when defects vary in shape, size, or illumination, existing models tend not to handle these changes effectively.

To overcome these limitations, self-attention-based Transformer networks are being used to develop classifiers that perform well in global modeling, and models that combine CNNs and Transformers have been proposed [8]. However, most of these studies are ad hoc or focus on injecting only certain attributes, and there is no systematic understanding of the role of each when convolution and attention are combined [13]. In this study, we propose Hybrid-DC, a hybrid deep-learning model that combines ResNet and ViT. ResNet can stabilize a model's learning and improve its performance through residual connections that facilitate deep neural network learning, and ViT has demonstrated excellent performance in learning the global features of images. The Hybrid-DC model is designed to integrate the strengths of these two approaches to enable more precise classification of complex defect shapes and sizes. To achieve a better balance between accuracy and efficiency, this study combines a hybrid attention layer and an attention fusion mechanism to overcome the performance limitations of existing models and proposes a new approach to effectively detect and classify steel surface defects. This research differs from previous efforts in the following aspects. First, the Hybrid-DC model is different from existing single-architecture-based models in that it combines the advantages of ResNet and ViT to provide a structure that can accurately learn various shapes and sizes of steel surface defects. While previous research has focused on learning local features using CNN-based models, this research combines the global-feature-learning capabilities of ViT to more accurately identify the overall pattern of defects. Second, the model offers strong feature-extraction capabilities through the hybrid attention layer and attention fusion mechanism. As a result, the Hybrid-DC model can achieve greater accuracy and generalization performance that reflects different defect types and environmental variability.

The main objective of this research is to propose a Hybrid-DC model that can effectively detect and classify various surface defects that occur during steel manufacturing, and it is expected to make an important contribution to the development of automated quality control systems. By classifying various steel surface defects with greater accuracy, the proposed Hybrid-DC model can contribute to maximizing the efficiency of the manufacturing process and maintaining product quality. Traditional manual inspection methods are not only time-consuming and costly but also rely on the subjective judgment of inspectors, which can lead to inconsistent results. Hybrid-DC models address these problems through automated defect detection and classification, enabling rapid and accurate defect detection, particularly in high-volume production environments. This will play an important role in helping steel manufacturers establish real-time quality monitoring systems for their production lines, enabling them to take immediate action when defects occur, thereby reducing defect rates and improving productivity. Based on its performance in steel surface defect detection, the Hybrid-DC model can be applied to other high-precision applications, such as semiconductor wafer surface defect detection, automotive parts inspection, and aircraft parts maintenance. Because the Hybrid-DC model can accurately detect and classify

defects of various sizes and shapes, and because high reliability and safety are required by various industries, the outstanding defect detection and classification capabilities of the Hybrid-DC model can make an important contribution to quality control and maintenance in these fields.

This research makes the following contributions:

- **Improved performance:** The proposed Hybrid-DC model achieves significantly lower loss values and greater accuracy compared with MobileNetV2, ResNet, and ViT. In particular, the Hybrid-DC model overcomes the performance limitations of MobileNetV2 and ViT by enhancing the feature-extraction ability of the model through the hybrid attention layer and attention fusion mechanism, and it shows excellent performance in the precise classification of steel surface defects.
- **Generalization performance:** For validation accuracy, Hybrid-DC achieved a performance of 0.9944, which demonstrates its superior generalization performance on training data as well as test data. ResNet followed with a validation accuracy of 0.9389, while ViT and MobileNetV2 achieved accuracies of 0.6444 and 0.5210, respectively, showing relatively poor generalization performance.
- **Learning efficiency:** The Hybrid-DC model converged rapidly in the early stages of training to reach high accuracy, suggesting that it can achieve good performance with fewer training epochs compared with models such as MobileNetV2 and ViT. ResNet requires more epochs and is slower to converge, again confirming the efficient learning ability of the Hybrid-DC model.
- **Fault detection precision:** The Hybrid-DC model proved effective in detecting and classifying various defects on steel surfaces with high precision, which, when compared to MobileNetV2, ResNet, and ViT, maximizes its potential to contribute to enhancing the quality control of steel products in the manufacturing process. In particular, Hybrid-DC had the best defect detection performance among all models.

This paper is organized as follows: Section 2 reviews the existing research on steel surface defect classification and explains the significance of this research through a theoretical background and literature review. Section 3 details the design and implementation of the proposed hybrid model, Hybrid-DC. Section 4 presents the experimental results and analysis and evaluates the performance of the model using various evaluation metrics. Section 5 discusses the main findings of the study, presents its limitations, and offers suggestions to address them. Section 6 concludes the study and highlights the performance and future applicability of the proposed model.

2. Related Work

2.1. Smart Factories and Steel Manufacturing Processes

Smart factories are one of the core concepts of the Fourth Industrial Revolution, the aim of which is to maximize productivity and improve quality by applying advanced technologies such as the Internet of Things, artificial intelligence, and big data to manufacturing processes. The implementation of smart factories is playing an important role in the steel manufacturing process, which is highly complex and requires automation to maintain product quality and production efficiency.

A system that detects and classifies surface defects in real time in the steel manufacturing process can help minimize disruptions in the manufacturing line, reduce the production of defective products, and improve productivity. Recently, defect detection systems utilizing deep learning have been studied in an effort to achieve greater accuracy compared with traditional image-processing techniques.

Figure 1 depicts the main steps in the steel manufacturing process. The first step involves the mixing, melting, and treatment of raw materials. Next, steel ingots are converted into steel plates through a continuous casting process, followed by continuous casting and rolling. The third step is hot and cold rolling, in which the steel plate is processed through hot rolling, cooling, and cold rolling. This is followed by surface treatment and

heat treatment (pretreatment), where treatments are carried out to prevent corrosion and improve surface quality.

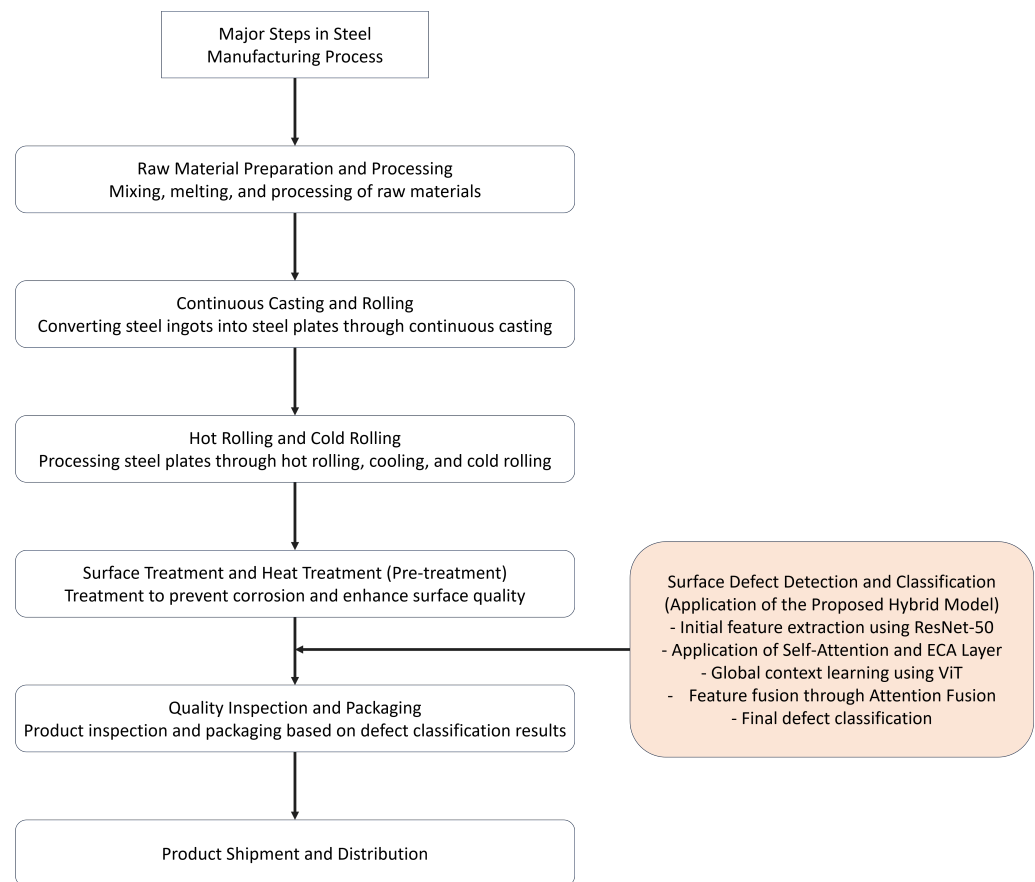


Figure 1. Steel manufacturing process.

In particular, surface defect detection and classification is an important process that has a significant impact on the quality and reliability of steel products. Accurate detection and classification of surface defects that occur during the manufacturing process is essential to ensuring the quality of the final product and reducing the occurrence of defective products. The hybrid model proposed in this work is designed to fulfill this need and enables high-accuracy defect classification through initial feature extraction using ResNet-50, the application of a self-attention mechanism and efficient channel attention (ECA) layer, global context learning using ViT, and feature fusion using attention fusion. Such a system can detect and classify surface defects in real time, minimizing disruptions in the manufacturing line and improving productivity. In the quality-inspection and packaging phase, products are inspected and packaged according to the results of defect classification. The product is then shipped and distributed.

2.2. Manufacturing Quality Inspection and Deep Learning

Historically, manufacturing quality inspection has relied primarily on manual labor, which can be inefficient and costly. However, automated inspection systems have been introduced with the advent of Industry 4.0, and CNNs in particular are playing an important role. CNNs have been used to develop automated inspection systems for various industries, revolutionizing the quality control process. Babic et al. [14] emphasized the growing importance of image-based quality inspection in smart manufacturing, noting that accessible and affordable systems are essential for modern manufacturing operations. Wang et al. [15] highlighted the need for smart remanufacturing and maintenance to optimize production and maintenance planning and presented the potential benefits of integrating automated

product inspection, tool health monitoring, and production and maintenance planning optimization. Examples of automated inspection systems using CNNs are provided by Gwon et al. [16], who proposed a specialized CNN-based image-quality classification method for bridge inspection to increase the accuracy and efficiency of the structural assessment process using advanced CNN techniques. Ngo [17] designed an automated inspection system aimed at detecting surface defects that integrates computational mechanics, servomotors, and advanced computer-vision techniques, demonstrating its feasibility and efficiency in practical applications. Indasyah et al. [18] developed a visual inspection system for gear-defect detection using Faster R-CNN and demonstrated the high accuracy of the system under difficult conditions, such as detecting multiple defects on a moving conveyor. These studies emphasize that CNN-based automated inspection systems have the potential to revolutionize quality control and improve efficiency, accuracy, and cost-effectiveness in a variety of industries.

Deep learning is revolutionizing the way defects are detected on the industrial floor through the ability of neural networks to learn complex patterns and features from large datasets. Several notable approaches in the field have each contributed to improving detection accuracy and efficiency. Karsten Roth et al. [19] proposed a method to achieve comprehensive anomaly detection by integrating generative and discriminative models. The main goal of this approach is to achieve high recall to identify even microscopic faults. Generative models, such as autoencoders and generative adversarial networks, model the normal behavior of the system, while discriminative models are used to distinguish between normal and abnormal patterns. Combining these two models can effectively identify a wide range of anomalies, reducing the number of problems that can occur in the manufacturing process. In addition, Ruff et al. [20] proposed a first-class classification technique for anomaly detection. Their method trains a neural network to recognize only data with normal patterns; everything that deviates from the normal pattern is marked as an anomaly. This approach is particularly effective when anomalies are rare and diverse, making it difficult to collect representative samples.

Venkataramanan et al. [21] introduced an innovative approach that uses an attention mechanism to improve defect detection. Attention mechanisms highlight potential defect areas by forcing the model to focus on specific regions of the input data. This targeted approach improves the model's ability to localize and identify anomalies in images, reducing false positives and increasing detection accuracy. Jihun Yi et al. [22] proposed a localized approach to defect detection. This method divides each image into smaller patches and applies support vector data descriptions to detect and segment anomalies at a microscopic level. Patch-level analysis is important for the precise identification and localization of defects for effective quality control. Finally, Liznerski et al. [23] explored the explanatory power of anomaly detection. This work emphasizes the importance of transparency, particularly in industrial applications, where understanding the reasons for fault detection decisions is critical. Explainable models provide insights into why a particular data point was categorized as an anomaly, which builds trust and promotes the adoption of deep-learning methods in quality control. By providing clear and understandable explanations, these models improve user trust and support informed decision-making.

The ResNet architecture is designed to effectively train neural networks in deep-learning models; ResNet-50 consists of 50 layers. The main feature of ResNet is that it reduces the difficulty of training through “skip connections” and enables effective training of deeper networks. Thanks to these characteristics, ResNet-50 performs well in various computer-vision problems, such as image classification and object detection. In addition, ResNet-50's residual-learning technique allows it to learn effectively in deep networks, extracting low-level and high-level features of images [24]. Each residual block is represented as follows:

$$y = x + F(x, W) \quad (1)$$

In (1), x is the input to the block, F is the transformation function in the block (convolution, batch normalization, activation), and W is the learned weight.

This expression describes the process of computing the transformation function $F(x, W)$ for the input x and adding it back to the input to form a residual connection. These residual connections prevent information loss during the learning process and enable effective learning in deep networks. These features allow for effective learning of low-level and high-level features that are important for distinguishing between different defects on steel surfaces (crazing, inclusions, patches, scratches, etc.). These features can also play an important role in Hybrid-DC models that analyze complex visual data, such as steel surface defect detection. For example, ResNet can accurately detect complex defects on the surface and maintain high accuracy by minimizing smoothing or distortion issues in the image [25]. In wood surface defect detection, a model based on ResNet achieved greater accuracy and efficiency compared with an existing single-shot multibox detector structure, proving that it can quickly detect various defects [26].

2.3. Advances in Classification Algorithms for Fault Diagnosis

In recent years, various research efforts and advances have been made to achieve the superior performance of classification algorithms for machine fault diagnosis. In particular, uncertainty-aware metric networks have been proposed to solve problems in limited noise source domains and unknown sparse noise domains. Wang et al. introduced a multi-scale cross-feature extraction module (MSCM) to effectively extract important classification information under noise interference and modeled the “query sample-class prototype” similarity to improve the model’s fault recognition ability [27].

In addition, an innovative method based on Transformers has been studied to achieve high accuracy and energy efficiency in machine fault diagnosis at the same time. Neural-Transformer combines the advantages of spiking neural networks (SNNs) and Transformers to improve the performance of fault diagnosis in noisy environments and lower the computational cost. The model uses a multi-headed spatiotemporal spiking self-attention (MHSSSA) mechanism to capture complex time–frequency features in a global range, showing excellent performance under noise [28].

In the field of photometric stereo, deep-learning-based calibrated photometric stereo techniques have shown high performance on non-lamellar surfaces. It has proven particularly useful for texture measurement on industrial metal surfaces. Jian et al. proposed an effective near-field photometric stereo network for industrial environments, achieving high surface normalization accuracy, even with a small number of light sources [29].

Modern deep-learning techniques in machine fault diagnostics are evolving to provide both high diagnostic accuracy and interpretability under complex industrial conditions. These advances enhance condition monitoring of expensive mechanical systems and significantly improve productivity by reducing unnecessary downtime [30].

2.4. Attention Mechanism

Self-attention is the core mechanism of the Transformer model, which allows each element of input data to determine its relationship to all other elements. Self-attention is particularly useful for extracting important features from sequential and image data. It allows a feature map at each location to interact with feature maps at other locations to incorporate contextual information. The model can therefore acquire a more sophisticated and richer representation. ECA improves model performance by efficiently learning relationships between channels. ECA can effectively emphasize important channels at a lower computational cost compared with traditional channel attention mechanisms. After global average pooling, ECA uses a simple one-dimensional convolution to calculate the importance of each channel. This allows the model to emphasize important features and improve performance at lower computational cost.

In recent years, ViT has become a powerful backbone in computer vision. ViT applies transforms, primarily in natural language processing, to computer-vision tasks. ViT captures global contextual information and is effective at distinguishing between defects with similar textures but different patterns. ViT creates sequences by dividing the input image

into fixed-size patches, arranging all patches in rows and embedding them. These embedded patches are processed through several layers of self-attention mechanisms, which are mathematically expressed as follows:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2)$$

In (2), Q (query), K (key), and V (value) are the matrices generated from the input patch embeddings, and d_k is the dimensionality of the key. The self-attention mechanism allows the patch at each location to learn its relationship to all other locations, which improves the accuracy of classification by learning the long-range dependency and spatial relationships between patches. The Softmax function learns the importance of each location, allowing it to focus more on specific parts of the input data. This process allows ViT to analyze the image holistically and learn the global features needed to distinguish between different defects on the steel surface. ViT is robust to previously unseen distributions due to its weak inductive bias. However, its non-hierarchical structure makes it inefficient when processing high-resolution images. To address these issues, several studies have explored integrating CNNs with Transformer architectures.

Heo et al. [31] proposed a pooling-based Vision Transformer that integrates the pooling layer of a CNN into a ViT. This approach aims to enhance self-attention-based architectures by aggregating global spatial information while increasing computational efficiency. Madan et al. [32] introduced a Self-Supervised Masking Convolutional Transformer Block (SSMCTB), which combines a masked convolutional layer for anomaly detection with channel-specific Transformer modules. The SSMCTB neural block is trained via a self-supervised learning approach using a self-reconstructing loss function and incorporates a channel-specific attention architecture to improve performance. Ullah et al. [33] incorporated a self-attention architecture into a CNN to improve object detection performance. They combined the CNN's ability to capture spatial features with ViT's ability to detect long-term temporal relationships. By stacking these networks, they were able to extract both spatial and temporal features. This method allows the object detection task to reflect dependencies between spatial and temporal features.

Lee and Kang [34] proposed an unsupervised anomaly detection and localization model using AnoViT, a Vision Transformer-based encoder–decoder model. AnoViT uses pre-trained ViTs for image-feature extraction and reconstruction. Unlike traditional ViTs, which use only class tokens to perform image classification, AnoViT reconstructs the original image by extracting a feature map that preserves the original localization information. In the process, the reconstruction loss is calculated to detect anomalous images. Lee and Kang [34] argued that, by integrating ViT into an encoder–decoder architecture, a better model can be built compared with integrating a traditional CNN. Liu et al. [35] proposed Cnnformer, which integrates a CNN with a Transformer architecture to extract semantic visual information from images. The encoder module of Cnnformer uses a convolutional layer to capture visual features of images, such as color and texture, and applies a novel attention mechanism that fuses static and dynamic contextual information to extract semantic visual information. Cnnformer learns visual representations that reflect both local and global information to improve action-recognition performance. Peng et al. [36] proposed a novel classification model that combines a CNN block and Transformer architecture. The model uses a CNN to extract key features from images and then a Transformer-based classification head to fully utilize spatial information. By combining the local feature-extraction capabilities of CNNs with the global contextual information of Transformers, this approach aims to improve on the performance of existing models. When we evaluated the performance of our model on a variety of public datasets, it achieved superior accuracy compared with previous attempts, demonstrating that the combination of CNN and Transformer architectures can lead to significant improvements in image classification. This work proposes a new model that overcomes the limitations of existing CNN-based approaches and enables more sophisticated classification by efficiently exploiting the Transformer architecture.

3. Hybrid-DC: Hybrid Defect Classification Framework

3.1. Model Structure

Hybrid-DC is a hybrid deep-learning model combining ResNet-50 and ViT that is designed to perform high-resolution analysis in steel surface defect detection and classification. The model uses ResNet-50 to extract low- and high-level features, a hybrid attention layer to reflect spatial and cross-channel saliency, and ViT to learn global context. Finally, attention fusion combines features from both models to classify defects. Based on Figure 2, the data flow is described in detail as follows.

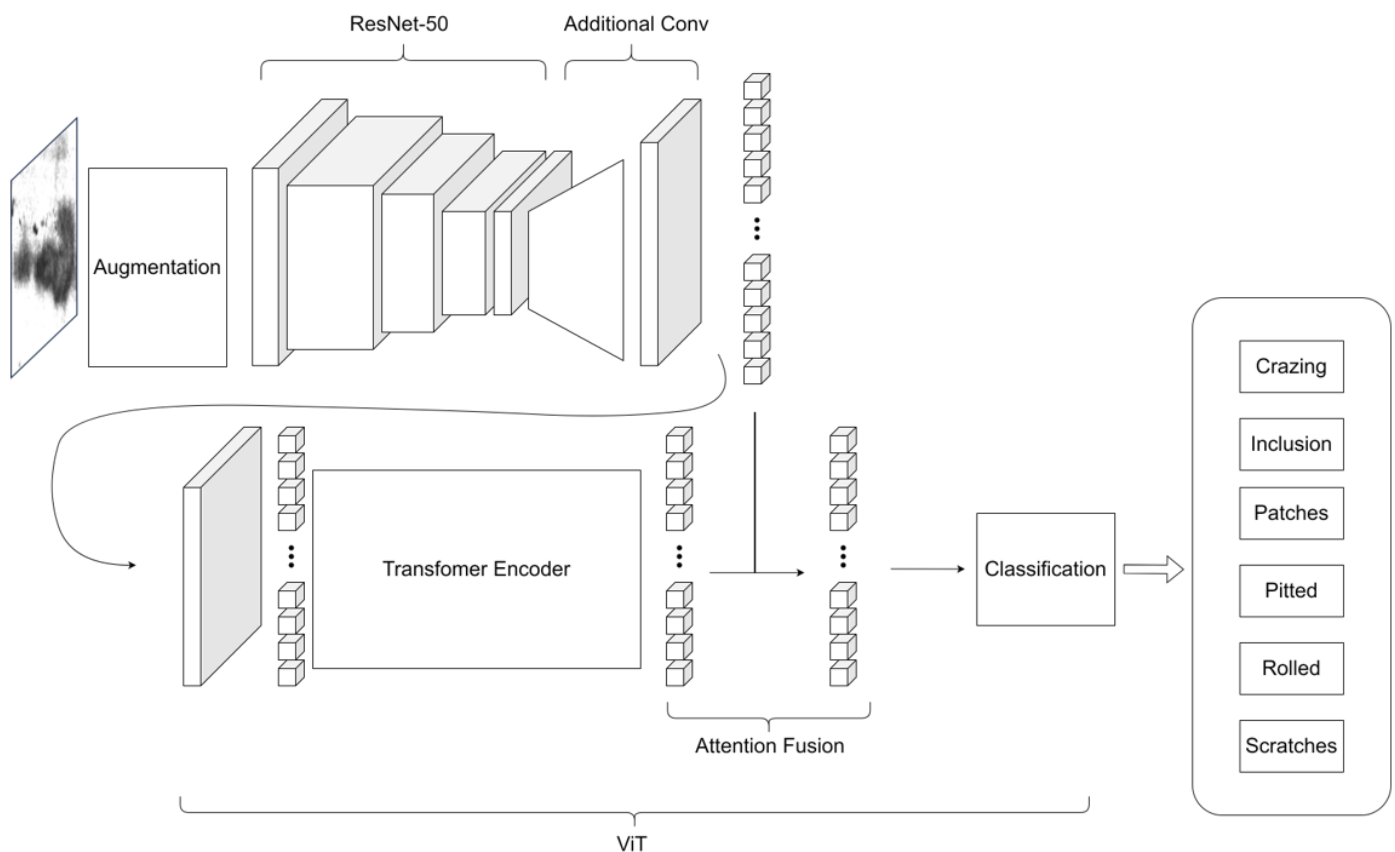


Figure 2. Hybrid-DC framework.

The input image is pre-trained by the ResNet-50 model to extract various low-level and high-level features. ResNet-50 is designed to learn effectively even from very deep networks through residual learning. In this study, we enhanced the feature-extraction process by removing the last two layers of the model and adding a hybrid attention layer in its place. The hybrid attention layer combines self-attention and ECA to reflect spatial and cross-channel saliency, which allows for more robust and discriminative features. After passing through the hybrid attention layer, the feature map is then fed into ViT, which divides the input image into patches, learns the features of each patch independently, and then combines them to create a global semantic representation. ViT is based on the Transformer architecture, which allows it to learn the global context within an image. The high-level features extracted by ViT are combined with feature maps from ResNet-50 through an attention fusion module, which integrates output from both networks to create a more robust representation. Finally, the combined feature vectors are fed into the classifier to perform the final classification. The classifier uses two fully connected layers, rectified linear unit activation, and a Softmax function to probabilistically predict the presence of a defect. This process allows the Hybrid-DC model to perform well in steel surface defect detection and classification.

3.2. Feature Representation

The hybrid attention layer uses the self-attention mechanism to learn the spatial relationships in the input feature map and applies the ECA mechanism to reflect the importance of each channel. First, the self-attention mechanism computes the query, key, and value projections for the input feature map $\mathbf{x} \in \mathbb{R}^{B \times C \times W \times H}$ as (3), (4), and (5), respectively:

$$\mathbf{Q} = \text{query_conv}(\mathbf{x}) \in \mathbb{R}^{B \times \frac{C}{8} \times W \times H} \quad (3)$$

$$\mathbf{K} = \text{key_conv}(\mathbf{x}) \in \mathbb{R}^{B \times \frac{C}{8} \times W \times H} \quad (4)$$

$$\mathbf{V} = \text{value_conv}(\mathbf{x}) \in \mathbb{R}^{B \times C \times W \times H} \quad (5)$$

$$\mathbf{E} = \mathbf{Q} \cdot \mathbf{K}^T \in \mathbb{R}^{B \times (W \cdot H) \times (W \cdot H)} \quad (6)$$

$$\mathbf{A} = \text{softmax}(\mathbf{E}) \quad (7)$$

In Formula (3), $\mathbf{Q}$ is the result of performing a query transformation on the input feature map $\mathbf{x}$, which has the batch size B , number of channels $\frac{C}{8}$, width W , and height H . In Formula (4), $\mathbf{K}$ is the result of performing a key transformation on the input feature map $\mathbf{x}$, which has the same magnitude as $\mathbf{Q}$. In Formula (5), $\mathbf{V}$ is the result of performing a value transformation on the input feature map $\mathbf{x}$, preserving the original number of channels C , and the size has the batch size B , number of channels C , width W , and height H . The attention energy $\mathbf{E}$ is computed from the inner product between the query and key using (6). In Formula (6), $\mathbf{E} = \mathbf{Q} \cdot \mathbf{K}^T$ represents the attention energy, which is calculated as the relevance of each location via the matrix product between the query and key. Here, the size of $\mathbf{E}$ is the batch size B , with $(W \cdot H) \times (W \cdot H)$ representing the relationship between two locations. We then apply the Softmax function to the calculated attention energy to obtain the attention map $\mathbf{A}$, which represents the importance of each location, as shown in (7). $\mathbf{A} = \text{softmax}(\mathbf{E})$ represents the result of applying the Softmax function to the attention energy $\mathbf{E}$ to calculate the importance of each location. Finally, the output of the self-attention mechanism is computed as a weighted sum of the attention map $\mathbf{A}$ and value $\mathbf{V}$, which is added to the input feature map $\mathbf{x}$ to produce the final output, shown in (8), where γ is a learnable parameter that controls the contribution of the attention output. $\mathbf{O} = \gamma \cdot (\mathbf{V} \cdot \mathbf{A}^T) + \mathbf{x}$ represents the process of computing the final output by adding the value obtained through the weighted sum of the attention map $\mathbf{A}$ and value $\mathbf{V}$ from the input feature map $\mathbf{x}$. Here, γ is a parameter that controls the influence of the self-attention output on the final output.

$$\mathbf{O} = \gamma \cdot (\mathbf{V} \cdot \mathbf{A}^T) + \mathbf{x} \quad (8)$$

$$\mathbf{y} = \text{avg_pool}(\mathbf{O}) \quad (9)$$

$$\mathbf{y} = \text{sigmoid}(\text{eca_conv}(\mathbf{y})) \quad (10)$$

$$\mathbf{O}_{\text{final}} = \mathbf{O} \cdot \mathbf{y} \quad (11)$$

The ECA mechanism takes the output $\mathbf{O}$ of the self-attention layer as input and learns the importance of each channel. First, it computes a vector $\mathbf{y} \in \mathbb{R}^{B \times C \times 1 \times 1}$ summarizing the information of each channel through global average pooling, as shown in (9). $\mathbf{y} = \text{avg_pool}(\mathbf{O})$ shows the result of calculating the average value of each channel by global average pooling of the output $\mathbf{O}$ of the self-attention layer. This vector is passed through a one-dimensional convolution layer with a kernel size of 3, and the channel importance is then computed using a sigmoid activation function as (10). $\mathbf{y} = \text{sigmoid}(\text{eca_conv}(\mathbf{y}))$ shows the process of taking the average value of each channel as input, calculating the importance of each channel through one-dimensional convolution, and then applying the sigmoid activation function to produce the final channel importance. Finally, the self-attention output $\mathbf{O}$ and the ECA output $\mathbf{y}$ are multiplied by each channel to compute the final output, as shown in (11). This allows the hybrid attention layer to

learn both spatial and channel features to enhance the expressiveness of the network. Let $\mathbf{O}_{\text{final}} = \mathbf{O} \cdot \mathbf{y}$ represent the process of computing the final output by multiplying the output $\mathbf{O}$ from the self-attention layer with the channel importance $\mathbf{y}$ computed through the ECA mechanism. The final enriched features reflect both the spatial relationship between channels and the importance of each channel.

3.3. Feature Generation

The attention fusion module aims to effectively fuse the features extracted from the ResNet and ViT models to generate a final feature representation. To achieve this, it uses the multi-head attention mechanism to learn the interdependencies between the two features. First, the features extracted from ResNet and ViT are called $\mathbf{f}_{\text{ResNet}} \in \mathbb{R}^{B \times D_{\text{ResNet}}}$ and $\mathbf{f}_{\text{ViT}} \in \mathbb{R}^{B \times D_{\text{ViT}}}$, respectively:

$$\mathbf{f}_{\text{ResNet_reduced}} = \text{norm1}(\text{resnet_to_vit}(\mathbf{f}_{\text{ResNet}})) \quad (12)$$

$$\mathbf{f}_{\text{combined}, -} = \text{multihead_attn}(\mathbf{f}_{\text{ResNet_reduced}}, \mathbf{f}_{\text{ViT}}, \mathbf{f}_{\text{ViT}}) \quad (13)$$

$$\mathbf{f}_{\text{concat}} = \text{norm2}([\mathbf{f}_{\text{combined}}, \mathbf{f}_{\text{ViT}}]) \quad (14)$$

In Formula (12), $\mathbf{f}_{\text{ResNet_reduced}} = \text{norm1}(\text{resnet_to_vit}(\mathbf{f}_{\text{ResNet}}))$ represents the process of transforming the feature $\mathbf{f}_{\text{ResNet}}$ of ResNet into the feature dimension of ViT by linear transformation, followed by normalization. Here, `resnet_to_vit` is a function that converts ResNet features to ViT feature dimensions, and `norm1` normalizes the converted features to help stabilize learning. The converted ResNet features and the original ViT features are fed into the multi-head attention module. This computes the combined feature $\mathbf{f}_{\text{combined}, -}$, as shown in (13). This represents the process of combining the reduced ResNet feature $\mathbf{f}_{\text{ResNet_reduced}}$ and the ViT feature $\mathbf{f}_{\text{ViT}}$ via multi-head attention. Here, the first $\mathbf{f}_{\text{ViT}}$ is used as a query, the second $\mathbf{f}_{\text{ViT}}$ is used as a key, and $\mathbf{f}_{\text{ResNet_reduced}}$ is used as a value. In this process, the interdependence between the two features is learned. The combined features and the original ViT features are combined to produce $\mathbf{f}_{\text{concat}}$, which can be represented as Equation (14). $\mathbf{f}_{\text{concat}} = \text{norm2}([\mathbf{f}_{\text{combined}}, \mathbf{f}_{\text{ViT}}])$ represents the process of merging the combined feature $\mathbf{f}_{\text{combined}}$ and the original ViT feature $\mathbf{f}_{\text{ViT}}$ and then normalizing them to produce the final combined feature.

$$\mathbf{a} = \text{fc}(\mathbf{f}_{\text{concat}}) \quad (15)$$

$$\mathbf{f}_{\text{fused}} = \mathbf{a}_{\text{ResNet}} \cdot \mathbf{f}_{\text{ResNet_reduced}} + \mathbf{a}_{\text{ViT}} \cdot \mathbf{f}_{\text{ViT}} \quad (16)$$

The combined features are computed by the fully connected layer to calculate the attention weight of the ResNet and ViT features, which can be expressed as Equation (15), where $\mathbf{a} = [\mathbf{a}_{\text{ResNet}}, \mathbf{a}_{\text{ViT}}]$ is a weight that indicates how important each feature of ResNet and ViT is to the final output. $\mathbf{a} = \text{fc}(\mathbf{f}_{\text{concat}})$ represents the process of taking the merged features $\mathbf{f}_{\text{concat}}$ as input and computing the weights $\mathbf{a}$ through the fully connected layer to represent the impact of each feature on the final output. The final fused features are computed as a weighted sum of the ResNet and ViT features, which can be represented as Equation (16). These fused features reflect the best of both ResNet and ViT, allowing the model to produce a more robust feature representation, $\mathbf{f}_{\text{fused}} = \mathbf{a}_{\text{ResNet}} \cdot \mathbf{f}_{\text{ResNet_reduced}} + \mathbf{a}_{\text{ViT}} \cdot \mathbf{f}_{\text{ViT}}$, where $\mathbf{a}_{\text{ResNet}}$ and $\mathbf{a}_{\text{ViT}}$ are the weights calculated for each feature in the process of weighting the reduced features $\mathbf{f}_{\text{ResNet_reduced}}$ from ResNet and the original features $\mathbf{f}_{\text{ViT}}$ from ViT to compute the final fused features $\mathbf{f}_{\text{fused}}$. This produces a robust feature representation that reflects the strengths of both models.

To improve the feature representation of the neural network, the hybrid attention layer combines self-attention and ECA to simultaneously learn spatial relationships and cross-channel importance, as shown in Equations (8) and (11). To generate the final features, the attention fusion module uses multi-head attention to learn the interdependencies between the two features, which are described in Formulas (13) and (16). This allows richer

and more discriminative features to be learned, which improves the performance of the overall network.

4. Experiment and Results

4.1. Experimental Environments

In terms of hardware configuration, we used a AMD Ryzen 7 5700X3D, a high-performance multi-core processor, to efficiently perform complex computational and data-processing tasks. We also used an NVIDIA GeForce RTX 3080 GPU to significantly improve the training and inference speeds of deep-learning models, creating an optimized environment for AI model training. The system was equipped with 32 GB of RAM to facilitate the processing of large datasets and the execution of complex algorithms. The software configuration is based on the Windows 11 operating system, PyTorch version 3.11 was used to ensure module compatibility, and PyCharm 2024.1 was used as a development environment for efficient code development and debugging. TensorFlow version 2.17.0 was used to provide a flexible platform that supports various machine learning algorithms and models.

All models were trained and evaluated in the same training environment. All datasets were split into training, validation, and test data using a custom data-loading scheme, with data loaded in batches. For training, the batch size was set to 16 to ensure efficient training. The training data were randomly shuffled to increase the robustness of the model to imbalanced data. The image data were subjected to standard preprocessing and augmentation techniques. The input images were resized to (224, 224), and the enhancement techniques used included horizontal and vertical flipping, rotation, brightness and contrast adjustments, and saturation and hue changes. The purpose of this step is to increase the generalization performance of the model by having it learn robustly on different types of data. In addition, all images were converted to tensor format and normalized based on the mean and standard deviation to enhance the stability of model learning.

The model's learning loss function was a cross-entropy loss function, which is suitable for multi-class classification problems. AdamW was adopted as the optimization algorithm to enable efficient learning. In addition, the weight reduction term was set to 0.0005 to avoid overfitting and improve the generalization performance of learning. The learning rate was adjusted incrementally through a cosine function-based scheduler. This is to reduce the learning rate later in the training to stabilize the convergence of the model and prevent instability due to an excessive learning rate. This learning rate adjustment is especially important for long-term learning.

The initial weights of the model were set using the Kaiming initialization method to ensure an efficient starting point for training, which contributes to the learning stability of the neural network in the early stages of training. In this study, learning in a GPU environment was the default setting, and GPU acceleration was used to improve the learning speed. This improved the learning efficiency and performance of the deep learning model. The weights of the model were saved periodically during the training process, which contributed to the reusability of the model and convenience in the validation phase. The trained models were stored appropriately so that they could be utilized for further experiments and analysis in the future. These preferences can be seen in Table 1, which ensured that the different models were trained and evaluated under the same conditions, allowing for fair and reliable performance comparisons.

Table 1. Experimental settings.

Configuration Aspect	Details
Hardware Configuration	CPU: AMD Ryzen 7 5700X3D GPU: NVIDIA GeForce RTX 3080 RAM: 32 GB
Software Configuration	OS: Windows 11 Development environment: PyCharm 2024.1 PyTorch Version: 3.11 TensorFlow Version: 2.17.0
Data Handling and Loading	Custom data loading with training, validation, and test splits Batch size: 16 Data shuffling for robustness to imbalance
Data Preprocessing and Augmentation	Image resizing to (224, 224) Augmentation: horizontal/vertical flips, rotation, brightness/contrast adjustment, saturation/hue modification Normalization using mean and standard deviation after conversion to tensor format
Loss Function	Cross-entropy loss (for multi-class classification)
Optimization Algorithm	Optimizer: AdamW Learning rates: 0.00001 or 0.0001 Weight decay: 0.0005
Learning Rate Scheduler	Cosine-based scheduler for gradual reduction in learning rate

4.2. Dataset

In this study, we used the NEU-DET dataset for the detection and classification of steel surface defects. The NEU-DET dataset contains various types of steel surface defects and provides high-resolution images of each defect type. The dataset consists of six typical surface defects on hot-rolled steel strips: rolling scale (RS), patch (Pa), crazing (Cr), pitted surface (PS), inclusion (In), and scratch (Sc). As shown in Table 2, the database contains 1800 grayscale images with 300 samples of each defect type.

Table 2. Class distribution in NEU-DET dataset.

Class	Number of Samples	Training (120)	Validation (30)	Test (150)
Rolling Scale (RS)	300	120	30	150
Patch (Pa)	300	120	30	150
Crazing (Cr)	300	120	30	150
Pitted Surface (PS)	300	120	30	150
Inclusion (In)	300	120	30	150
Scratch (Sc)	300	120	30	150

Figure 3 shows sample images of the six defect types in the dataset, each of which is provided at its original resolution of 200×200 pixels. Variations in the defects are visible within classes. For example, the scratch class can appear in different forms, such as horizontal, vertical, and tilted scratches, while defects between classes can have similar characteristics, such as roll-in scale, crazing, and pitted surfaces. In addition, defect images within a class can have large differences in grayscale values due to variations in lighting conditions and materials.

A major challenge with the NEU surface defect database is the large variation in the appearance of defects within classes and the similarities in defects between classes, and these defect images are strongly affected by lighting and material variations. In this

study, we used this dataset to compare the performance of MobileNetV2, ResNet, ViT, and the proposed model, Hybrid-DC.

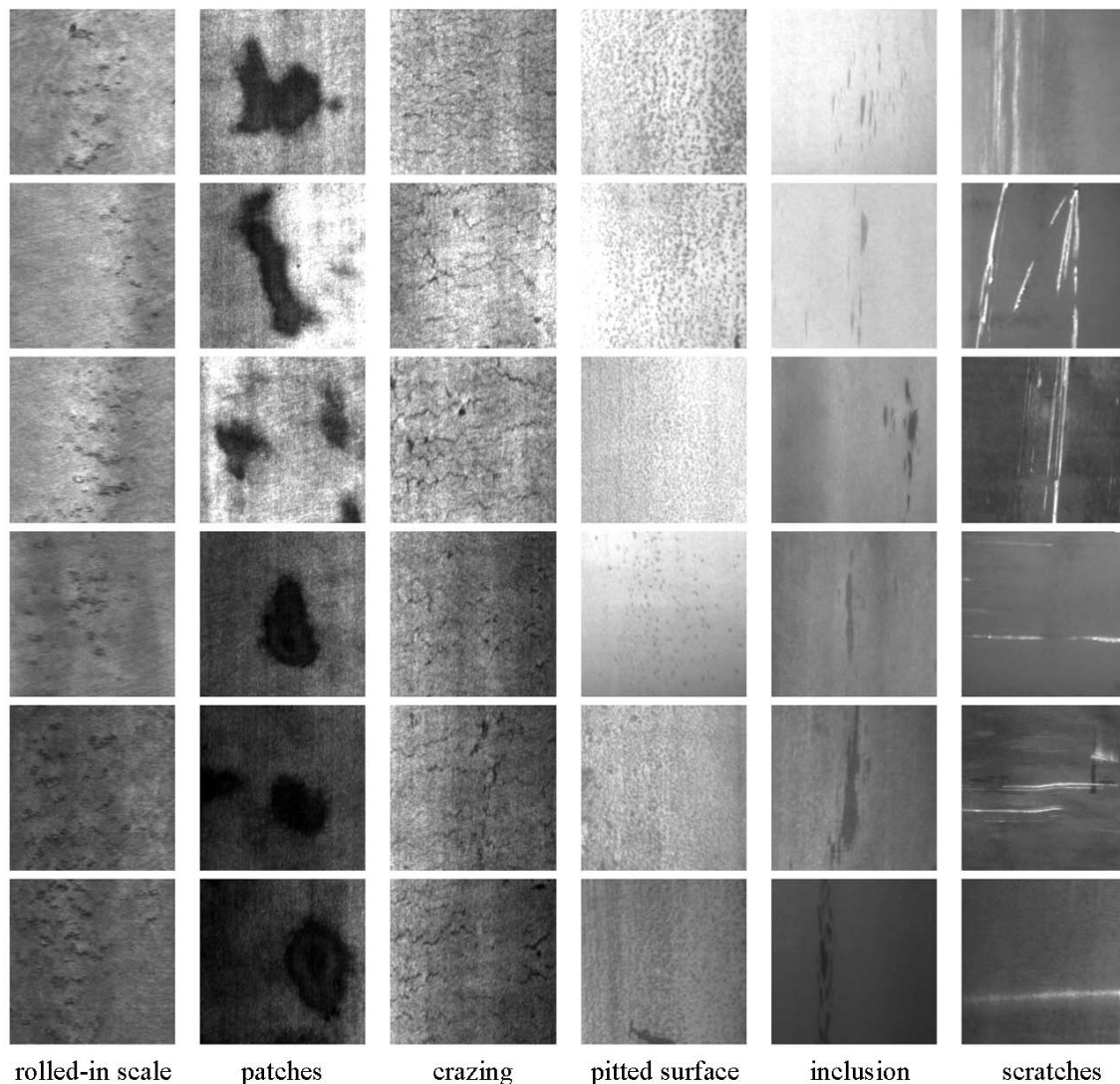


Figure 3. Six types of defects.

In addition, we augmented the dataset to allow the model to learn different data variations, creating a robust model with good generalization performance. The data-augmentation techniques applied included resizing, horizontal and vertical flipping, rotation, and color adjustments (brightness, contrast, saturation, and hue).

4.3. Performance Metrics

We used a variety of performance metrics to evaluate the performance of our models. These metrics allowed us to evaluate how well a model performed on a given task from multiple perspectives. We chose the Learning Curve, Confusion Matrix, Micro Metrics, and Weighted Metrics tools to evaluate the performance of the model. The Learning Curve is an important tool to evaluate how a model generalizes to the data by visually representing the change in the accuracy and loss of the model during the learning process. We closely analyzed the learning tendency of the model, checked whether it was overfit or underfit, and provided a basis for achieving the optimal learning state. The Confusion Matrix can evaluate the performance of a binary classification model, which visually represents the relationship between the predicted and true classes. It is composed of four metrics:

true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), from which metrics such as accuracy, precision, recall, and F1 scores can be calculated. Table 3 provides a visual understanding of how accurately the model predicted the true class, which can be used to calculate various performance metrics.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (17)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (18)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (19)$$

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (20)$$

$$\text{Log-Loss} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^M y_{ij} \log(\hat{p}_{ij}) \quad (21)$$

Table 3. Confusion Matrix.

	Predicted Positive	Predicted Negative
Actually Positive	TP	FN
Actually Negative	FP	TN

Accuracy represents the percentage of samples that the model correctly predicted out of the total samples and is used to evaluate the overall prediction performance. Accuracy can be expressed by Equation (17), where TP is the number of times the model correctly predicted the true positive data as positive, TN is the number of times the model correctly predicted true negative data as negative, FP is the number of times the model incorrectly predicted true negative data as positive, and FN is the number of times the model incorrectly predicted true positive data as negative. Accuracy is useful for evaluating overall performance when classes are well balanced but can be overestimated in the presence of class imbalance.

Precision is the percentage of samples predicted to be positive that are actually positive and is an important metric when considering the impact of FP . Precision is defined in (18). Precision is important in situations where the model is focused on minimizing FP , for example, in spam filtering, the goal of which is to reduce false spam classifications.

Recall represents the percentage of true positive samples that the model correctly predicted as positive and plays an important role in reducing FN . Recall is defined in (19). Recall is useful for reducing the error of missing patients in disease diagnosis, for example, when it is important not to miss true positive cases. The F1 score provides a harmonized average of precision and recall to assess the balance between the two metrics. The F1 score is defined in (20). An F1 score is useful when precision and recall need to be balanced, and it evaluates the overall performance of the model when both metrics are important.

Log-loss measures the difference between the model's predicted probability and the actual label and is used to evaluate the quality of the predicted probability. Log-loss is defined in (21), where N is the number of samples, M is the number of classes, y_{ij} is a binary metric for the actual class j of sample i , and $\hat{p}_{ij}$ is the predicted probability that sample i belongs to class j . Log-loss is useful for judging the reliability of probabilistic predictions and is used in situations in which probabilistic judgment is important.

$$\begin{aligned}
\text{Micro precision} &= \frac{\sum TP}{\sum (TP + FP)}, \\
\text{Micro recall} &= \frac{\sum TP}{\sum (TP + FN)}, \\
\text{Micro F1 score} &= 2 \cdot \frac{\text{Micro precision} \cdot \text{Micro recall}}{\text{Micro precision} + \text{Micro recall}}
\end{aligned} \tag{22}$$

$$\begin{aligned}
\text{Weighted precision} &= \sum_{i=1}^M \left(\frac{n_i}{N} \cdot \text{Precision}_i \right), \\
\text{Weighted recall} &= \sum_{i=1}^M \left(\frac{n_i}{N} \cdot \text{Recall}_i \right), \\
\text{Weighted F1 score} &= 2 \cdot \frac{\text{Weighted precision} \cdot \text{Weighted recall}}{\text{Weighted precision} + \text{Weighted recall}}
\end{aligned} \tag{23}$$

Micro Metrics calculate TP , FP , and FN cumulatively for all classes and are ideal for evaluating the overall predictive accuracy of a model over the entire dataset. Micro Metrics reflect the contribution of each class equally, which is advantageous for fairly assessing the overall model performance, even in the presence of class imbalance. The micro precision, micro recall, and micro F1 score are defined in (22). These metrics are useful for evaluating model performance when it is desirable to minimize the overall number of errors, particularly in datasets with a high degree of class imbalance.

The Weighted Metrics tool calculates performance metrics by weighting them according to the proportion of data in each class, which allows for a fair evaluation of the overall performance of a model, even in datasets with class imbalance. The performance on classes with more data has a greater impact on the overall metrics of a model, making it an important criterion for evaluating the reliability of a model in real-world applications. The weighted precision, weighted recall, and weighted F1 score are defined in (23), where n_i is the number of samples in class i , and N is the total number of samples. Because Weighted Metrics adjust results based on class proportions, they are useful as an evaluation tool, particularly when classes are unbalanced. In addition to these metrics, we used additional metrics to assess the stability and statistical significance of the model.

$$\text{Confidence Interval} = \bar{x} \pm z \cdot \frac{\sigma}{\sqrt{n}} \tag{24}$$

$$T - \text{statistic} = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \tag{25}$$

$$\text{GFLOPS} = \frac{\text{Computation (FLOP)}}{\text{Execution time (seconds)} \times 10^9} \tag{26}$$

$$\text{FPS} = \frac{\text{Total number of frames processed}}{\text{Execution time (seconds)}} \tag{27}$$

Confidence intervals (95%) help in assessing whether a model's performance is consistent within a certain range. The confidence interval provides a range where the true accuracy of the model is expected to lie and indicates the reliability of the results. A narrower confidence interval indicates that the model is performing consistently. The calculation of a confidence interval is expressed as (24), where $\bar{x}$ is the mean of the sample, z is the Z-value corresponding to the confidence level (e.g., $z \approx 1.96$ for a 95% confidence interval), σ is the standard deviation, and n is the sample size.

To determine whether the performance differences between the models were statistically significant, we utilized the T-statistic and p -value metrics. The T-statistic is the standardized difference in average performance between two models, while the p -value

indicates the probability that this difference is due to chance. In general, we can interpret a performance difference as statistically significant when the p -value is less than 0.05. The formula for the T-statistic is (25), where $\bar{x}_1$ and $\bar{x}_2$ are the average performance of the first and second models, respectively, σ_1 and σ_2 are the standard deviations of each model, and n_1 and n_2 are the number of samples in each model. This formula allows the significance of performance differences between models to be statistically determined.

We also evaluated the computational efficiency of the model through the Giga Floating-Point Operations Per Second (GFLOPS) and Frames Per Second (FPS) metrics. GFLOPS represents the number of floating-point operations that the model can perform per second. GFLOPS and FPS are calculated via (26) and (27), respectively, where FLOPS is the total number of floating-point operations performed by the model, and execution time is the time in seconds it takes the model to perform the operations. FPS is the number of frames or images the model can process per second and is a metric that evaluates the real-time processing power of the model. Higher FPS values indicate that the model can process a lot of data in a given amount of time, making it suitable for real-time applications.

4.4. Results

We conducted several experiments to evaluate the performance of the different models, comparing the performance of ResNet, Hybrid-DC, MobileNetV2, and ViT models collectively. The performance of each model was analyzed through a decreasing loss-function trend, an increasing accuracy trend, a Confusion Matrix, Weighted Metrics, and Micro Metrics.

4.4.1. Accuracy and Loss

By analyzing the accuracy and loss trends during model training, we were able to determine the learning stability and performance changes of each model. Figures 4–7 depict the changes in training and validation accuracy and loss by epoch for each model, indicating the learning stability and convergence speed.

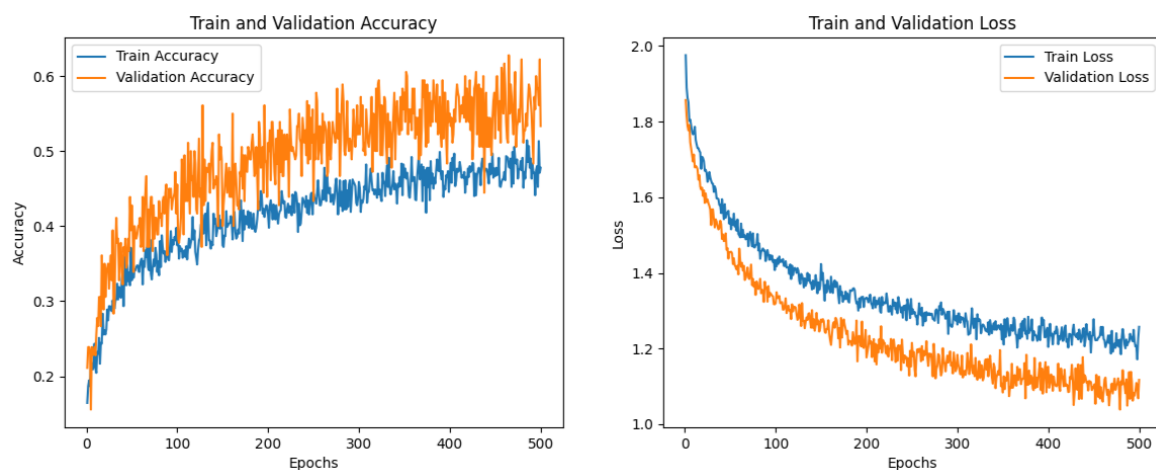


Figure 4. MobileNetV2 model.

For the MobileNetV2 model, as shown in Figure 4, training accuracy increased gradually with increasing epochs, but validation accuracy increased more rapidly and with greater variability. Both training and validation losses show decreasing trends, but the validation loss was lower and more volatile than the training loss. This indicates that the MobileNetV2 model performed relatively well on validation data, but with unstable volatility. In particular, the fact that the validation loss fluctuates at a lower level than the training loss indicates that the model performs better on validation data but may lack stability in terms of learning.

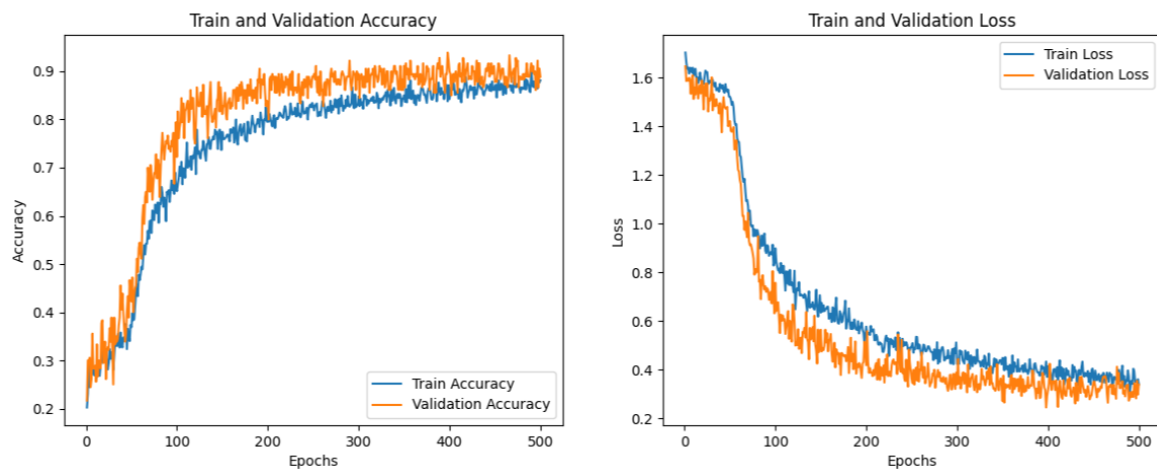


Figure 5. ResNet model.

For the ResNet model, Figure 5 shows that both the training and validation accuracies increased rapidly with increasing epochs and tended to stabilize near epoch 100. There was little difference between the two accuracies, and they approached each other as the epoch increased. This means that the ResNet model learned reliably on both the training and validation data, and the risk of overfitting was low. As can be seen in Figure 5, both the training and validation losses decreased rapidly with increasing epochs and converged to a nearly constant level after epoch 100. The difference between the two loss values is not significant, indicating that the model converged well, the ResNet model learned the data efficiently, and the learning process was stable.

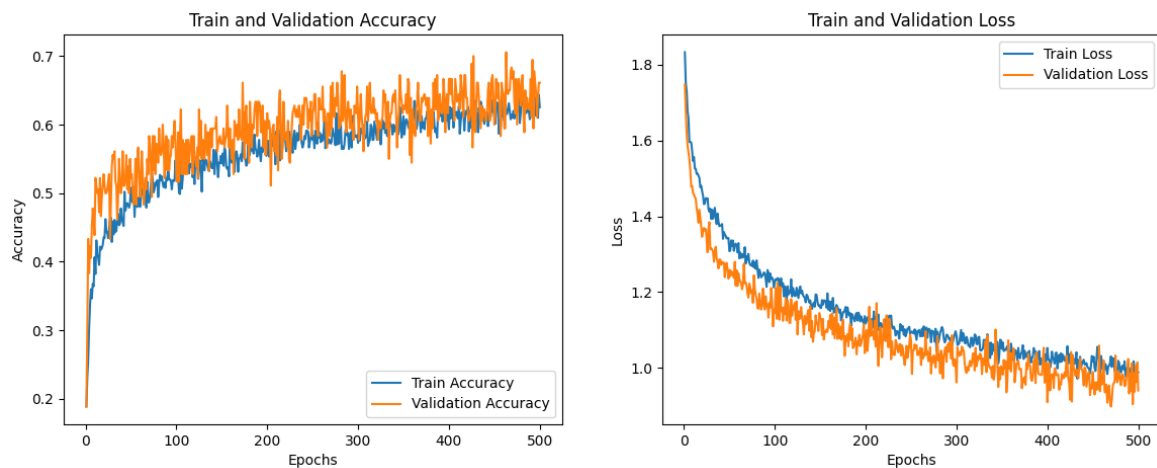


Figure 6. ViT model.

For the ViT model, as shown in Figure 6, both training and validation accuracies tended to increase with the epoch. The difference between the two accuracies was not significant, and after a certain epoch, they performed similarly. This suggests that the model learned reliably on both training and validation data. In particular, the small difference between the two accuracies suggests that the risk of overfitting was small. As shown in Figure 6, both the training and validation losses steadily decreased with increasing epochs, suggesting that the model converged well. Overall, this indicates that the ViT model learned reliably.

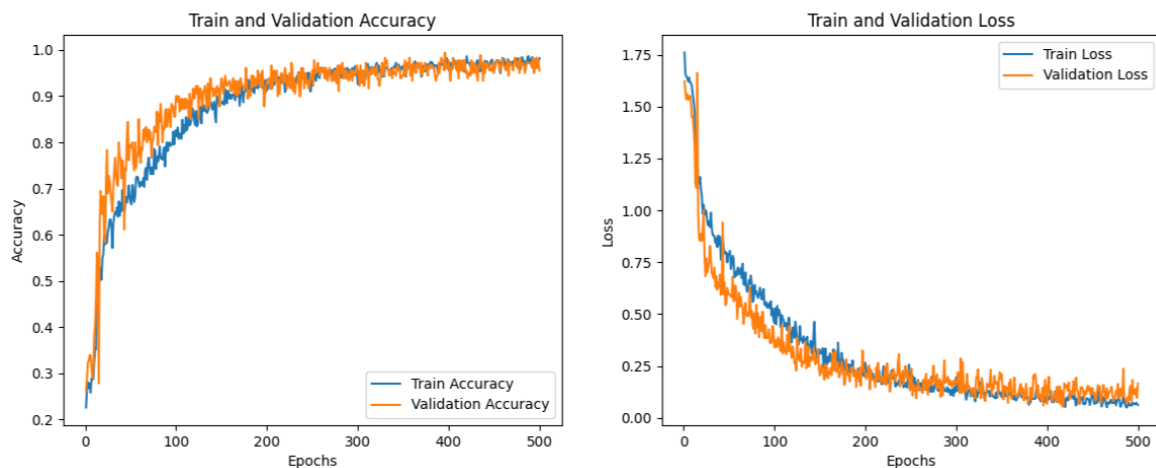


Figure 7. Hybrid-DC model.

For the Hybrid-DC model, as shown in Figure 7, both training and validation accuracies increased rapidly in the early epochs and remained high after epoch 100. Little difference was evident between the two accuracies. This indicates that the Hybrid-DC model performed well on both the training and validation data and learned effectively without overfitting. Furthermore, the training and validation losses in Figure 7 exhibit a sharp decrease at the beginning of the epoch and stay at a low value thereafter. The two loss values were almost identical and stabilized at a low level. This shows that the Hybrid-DC model generalized well to the data.

Overall, MobileNetV2 was unstable, with high variability in validation accuracy and loss, while ResNet achieved stable learning, with accuracy and loss converging rapidly in epoch 100. ViT showed little difference in performance between the two datasets and a steady decrease in loss, suggesting strong generalization performance. Hybrid-DC showed rapid performance improvement and loss reduction from the beginning, demonstrating learning stability and generalization ability overall.

4.4.2. Confusion Matrix

The Confusion Matrix tool visually represents the relationship between a model's predicted class and the true class, making it an important tool for analyzing the prediction performance of each model for a particular class. We compared MobileNetV2, ResNet, and ViT models to evaluate the performance of the proposed Hybrid-DC model.

The Confusion Matrix tool was used to analyze the performance of each model, and several performance characteristics were derived. In Figure 8, the MobileNetV2 model has a relatively high accuracy for "Crazing" and "Inclusion" defects, but significant misclassifications are evident for "Patches" and "Scratches" defects. This suggests that the shallow architecture of MobileNetV2 limits its ability to effectively distinguish among complex defect features. The ResNet model shows an overall performance improvement with a deeper neural network structure in Figure 9, but confusion still exists for certain defect types, indicating a limitation in learning fine-grained defect features. Figure 10 shows that the ViT model performed well in the identification of "Patches" defects but had a high misclassification rate for other defect types, indicating weakness in learning local features. This suggests that ViT models are strong at learning global features but may struggle with learning local, detailed features. However, as shown in Figure 11, the Hybrid-DC model achieved excellent classification accuracy on all defect types by learning local and global features simultaneously, and it showed near-perfect classification performance on a variety of defects, demonstrating its ability to effectively integrate multi-scale features.

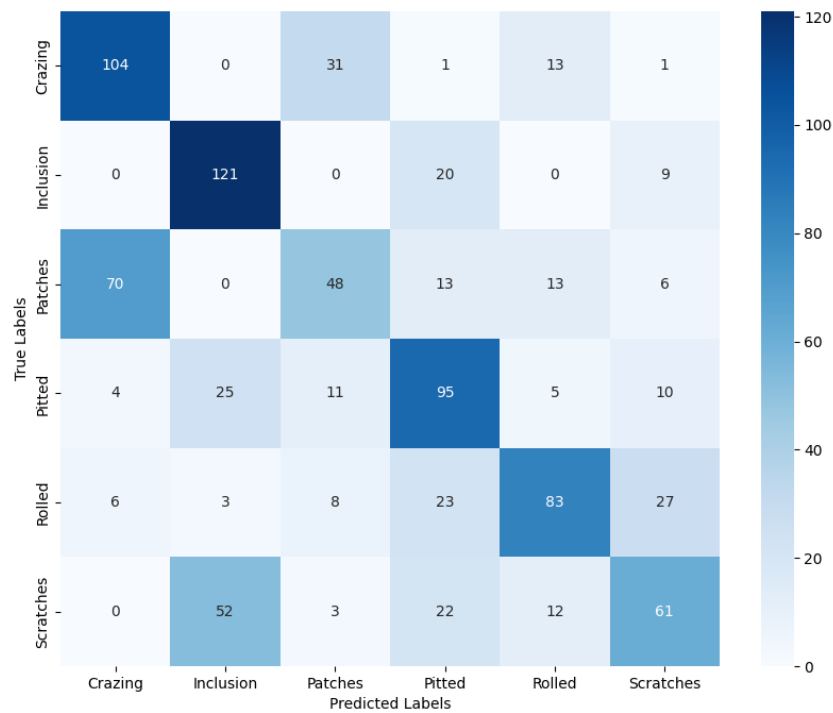


Figure 8. MobileNetV2 model.

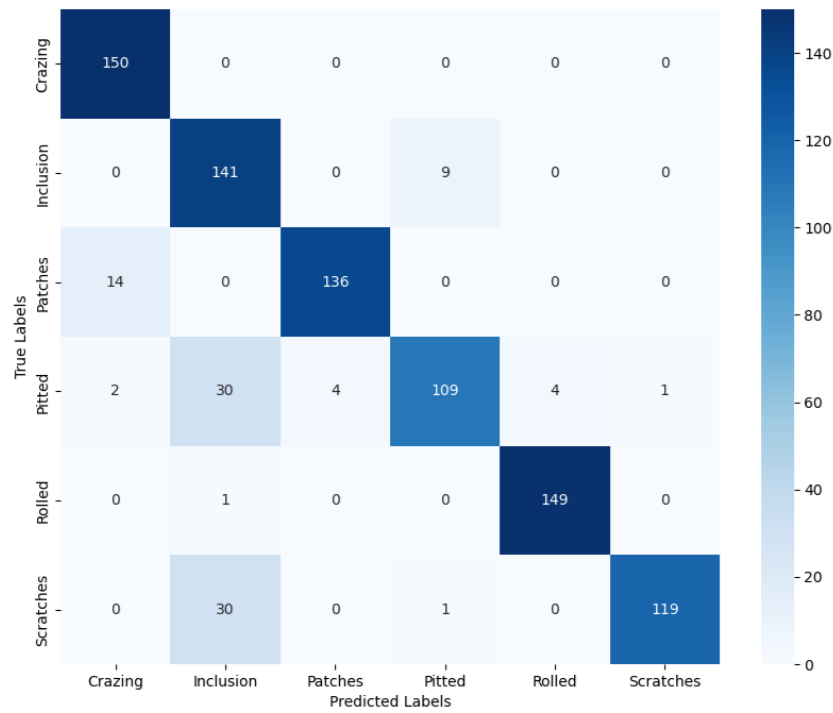


Figure 9. ResNet model.

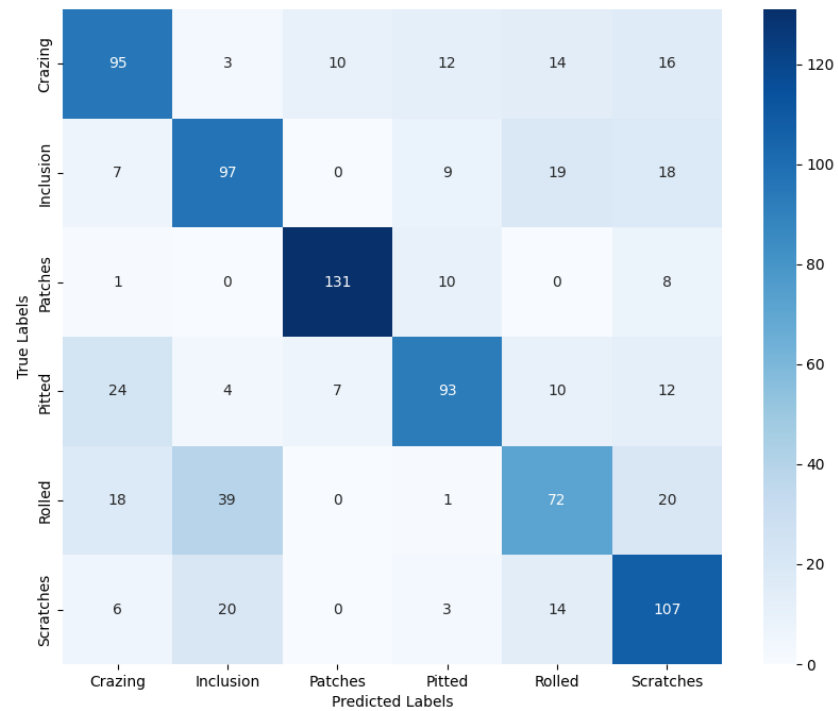


Figure 10. ViT model.

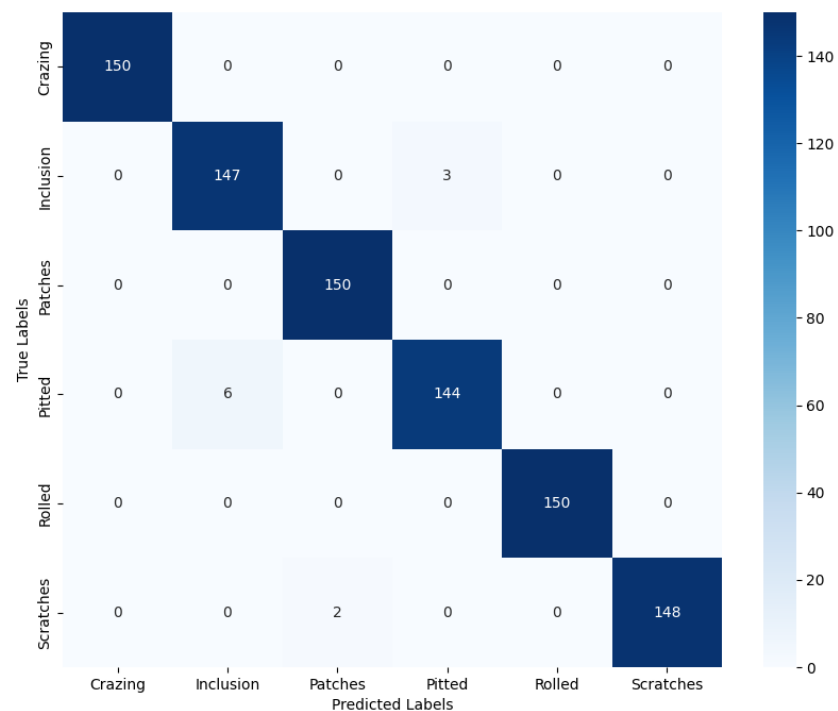


Figure 11. Hybrid-DC model.

4.4.3. Micro Metrics

Table 4 compares the performance according to Micro Metrics, which reflects the contribution of each class equally. For all models, micro precision, micro recall, and micro F1 scores had the same values because the Micro Metrics were calculated in the same way from the perspective of the entire sample. The Hybrid-DC model's micro precision, micro recall, and micro F1 score were all 0.9777, indicating that it maintained high predictive accuracy for all the samples. These results demonstrate that the Hybrid-DC model performed well on the entire dataset and performed equally well for all defect types. By comparison,

the ResNet model had a Micro Metric value of 0.8885, which is approximately 10% lower than that produced by Hybrid-DC. This suggests that Hybrid-DC learned more effectively for certain defect types. For the ViT model, the Micro Metric value was 0.6444, which is approximately 34% lower than that produced by Hybrid-DC, while the MobileNetV2 model had the lowest performance at 0.5210. This can be attributed to the failure of the ViT and MobileNetV2 models to learn enough features of certain defect types.

In conclusion, the Hybrid-DC model scored the highest using the Micro Metrics tool, clearly demonstrating that it can perform consistently well for all defect types in the defect detection task. This indicates that it can make an important contribution to maintaining consistent quality control in manufacturing processes.

Table 4. Micro Metrics of four models.

Model	Accuracy	Precision	Recall	F1 Score
MobileNetV2	0.5210	0.5210	0.5210	0.5210
ResNet	0.8885	0.8885	0.8885	0.8885
ViT	0.6444	0.6444	0.6444	0.6444
Hybrid-DC	0.9777	0.9777	0.9777	0.9777

4.4.4. Weighted Metrics

Table 5 shows the performance of each model based on Weighted Metrics, which allows us to evaluate the performance considering class imbalance by calculating the metrics proportional to the size of each class. The weighted precision, weighted recall, and weighted F1 score of the Hybrid-DC model were 0.9781, 0.9777, and 0.9778, respectively, which shows that the Hybrid-DC model maintained a high level of performance despite the data imbalance between classes. These metrics suggest that the Hybrid-DC model achieves high accuracy and recall, particularly for classes with large amounts of data, which supports its strong generalization ability. However, the ResNet model performed poorly compared with Hybrid-DC, with a weighted F1 score of 0.8881, followed by ViT with 0.6645 and MobileNetV2 with 0.5228, which are significantly lower than the performance of the Hybrid-DC model. This shows that the Hybrid-DC model provides superior classification performance for different defect types and has good predictive power even in the presence of class imbalance.

Table 5. Weighted Metrics of four models.

Model	Accuracy	Precision	Recall	F1 Score
MobileNetV2	0.5210	0.5689	0.5210	0.5228
ResNet	0.8885	0.8950	0.8885	0.8881
ViT	0.6444	0.7031	0.6444	0.6645
Hybrid-DC	0.9777	0.9781	0.9777	0.9778

The Hybrid-DC model therefore outperformed the other models in Weighted Metrics, demonstrating that it can maintain consistent performance in real-world environments with class imbalances. These results strongly suggest that the Hybrid-DC model can be an effective tool in manufacturing environments with different defect types and an unbalanced data distribution.

The analysis of these experimental results suggests that this research can make the following contributions. The proposed Hybrid-DC model achieved significantly lower loss values and greater accuracy compared with ResNet. This can be attributed to the combination of the hybrid attention layer and the attention fusion mechanism, which significantly improved the feature-extraction ability of the model. In particular, this combination played an important role in effectively recognizing and classifying defects on complex steel surfaces, which is an important factor in demonstrating the superiority of the Hybrid-DC model over traditional models. In terms of validation accuracy, Hybrid-DC had a high

performance of 0.9944, indicating that the model has good generalization performance not only on training data but also on test data. ResNet had a relatively low validation accuracy of 0.9389, suggesting that the Hybrid-DC model can perform more consistently and reliably in a variety of data environments. In addition, structural improvements to the Hybrid-DC model enhanced the model's ability to respond to complex data patterns, making it more applicable to real-world industrial environments.

The Hybrid-DC model converged quickly and reached high accuracy from the beginning due to its efficient learning, while ResNet converged relatively slowly, requiring more epochs. This suggests that the Hybrid-DC model has the ability to quickly identify patterns in the data from the initial training phase to optimize the model. This rapid-convergence property can maximize time and resource efficiency when training models on large datasets.

4.4.5. Statistical Significance

The performance of Hybrid-DC was analyzed through a comprehensive performance evaluation, comparing it to MobileNetV2, ResNet-50, and Vision Transformer (ViT). Accuracy, 95% confidence interval, computational complexity (GFLOPS), and frames per second (FPS) were used as evaluation metrics, and the T-statistic and *p*-value were used to verify the statistical significance of the comparison with Hybrid-DC. The figures for each metric can be found in Table 6.

Table 6. Testing for statistical significance.

Model	Accuracy	Confidence Interval (95%)	GFLOPS	FPS	T-Statistic (Hybrid-DC vs.)	<i>p</i> -Value (Hybrid-DC vs.)
MobileNetV2	0.5311	0.4989–0.5633	0.326	101.13	821.1439	<0.0001
ResNet-50	0.8933	0.8733–0.9111	4.132	88.41	274.7501	<0.0001
ViT	0.6511	0.6189–0.6789	16.864	79.60	641.2823	<0.0001
Hybrid-DC	0.9878	0.9811–0.9944	24.177	38.88	-	-

Hybrid-DC, with an accuracy of 0.9878, significantly outperformed the other models. The 95% confidence interval was narrow, ranging from 0.9811 to 0.9944, demonstrating the stability of its performance and the reliability of its predictions. ResNet-50 performed well with an accuracy of 0.8933, but the comparison with Hybrid-DC showed a statistically significant difference. MobileNetV2 and ViT showed accuracies of 0.5311 and 0.6511, respectively, and performed relatively poorly compared to Hybrid-DC. This suggests that Hybrid-DC provides more consistent performance for different data patterns. The computational complexity and processing speed analysis showed the advantages and disadvantages of Hybrid-DC. Hybrid-DC had a high computational complexity of 24.177 GFLOPS and a relatively low processing speed of 38.88 FPS. In comparison, ResNet-50 recorded 4.132 GFLOPS and 88.41 FPS, showing efficiency with low computational cost but high processing speed. MobileNetV2 recorded the lowest GFLOPS (0.326) and highest FPS (101.13), demonstrating the benefits of a lightweight model. ViT had a medium level of computational complexity and processing speed, with 16.864 GFLOPS and 79.60 FPS. These results suggest that Hybrid-DC provides high performance but at a high computational cost. The T-statistic and *p*-value analysis further supported the performance superiority of Hybrid-DC. The *p*-value is very small, less than 0.0001 for all models, proving that the performance difference of Hybrid-DC is statistically significant. The comparison with MobileNetV2 had the highest T-statistic of 821.1439, indicating the most significant performance difference from Hybrid-DC. The T-statistic with ResNet-50 is 274.7501 and the T-statistic with ViT is 641.2823. All of these results show that the performance difference from Hybrid-DC is statistically significant.

In conclusion, Hybrid-DC outperformed the other models, based on its high accuracy and stability, and demonstrated its superiority in various performance metrics. Although the computational complexity and processing speed come at a somewhat higher cost, the excellent prediction performance and consistent results show that Hybrid-DC

can be a valuable solution in a variety of industrial applications. Future research will require the introduction of model lightweighting and optimization techniques to increase computational efficiency.

4.4.6. LIME

The interpretation results of Locally Interpretable Model-agnostic Explanations (LIME) provide important insights into how the Hybrid-DC model recognizes and predicts various defects on the steel surface. For each defect, the areas the model focused on can be seen visually, which shows how the features learned by the model are connected to the actual defect.

In Figure 12, the areas that the model focused on during the “Crazing” defect prediction process are highlighted by the yellow outline. This result shows that the model focuses on regions that match the actual defect site when recognizing “Crazing” defects, suggesting that the Hybrid-DC model has a good understanding of the unique pattern of these defects.

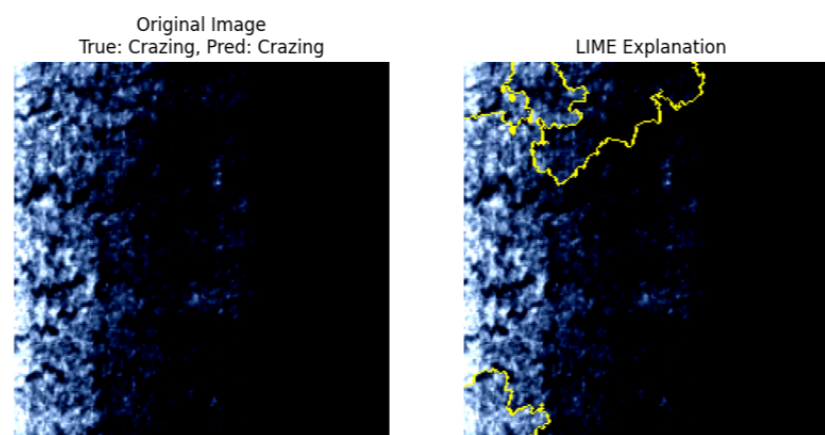


Figure 12. Crazing defect.

Figure 13 provides an interpretation of the “Inclusion” defect. The model pays attention to areas that reflect the characteristics of the “Inclusion” defect, which supports the reliability of its predictions. The fact that the model makes consistent predictions by focusing on the key characteristics of the defect indicates that the model understands the intrinsic nature of the defect beyond simply learning data patterns.

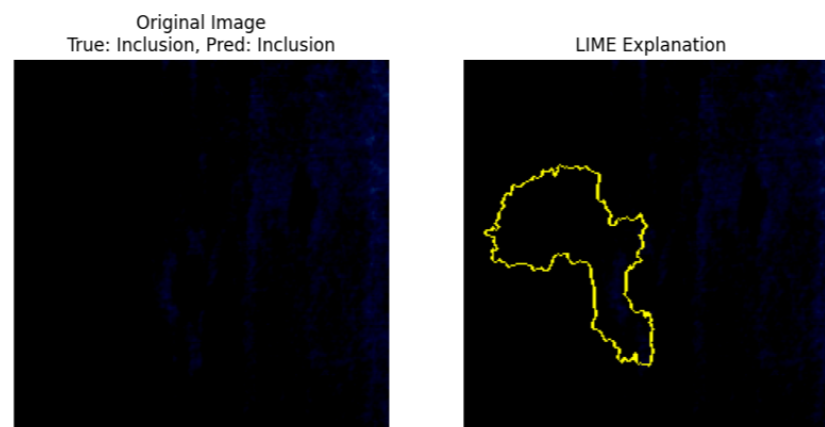


Figure 13. Inclusion defect.

The interpretation of the “Patches” defect in Figure 14 shows that the model correctly recognizes the unique pattern of this defect. The highlighted areas in the LIME interpretation show that the model has learned the structural characteristics of the “Patches” defect.

This is an important indication that the model understands the intrinsic structure of the defect, not just the outward patterns in the data.

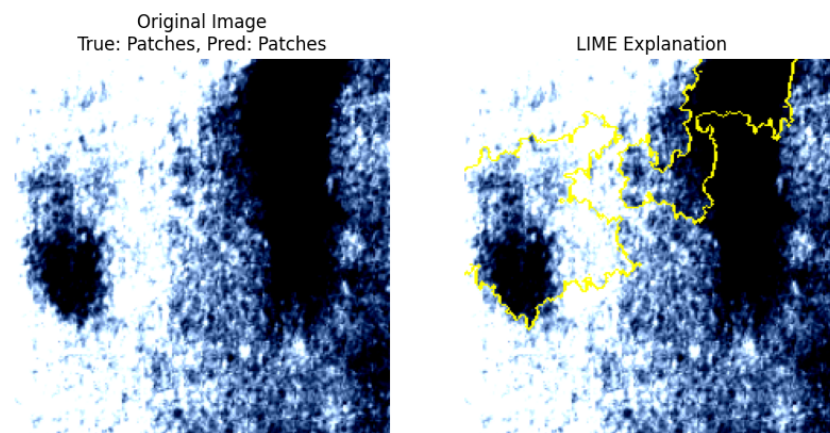


Figure 14. Patches defect.

Figure 15 shows where the model focused its analysis when predicting the “Pitted Surface” defect. The yellow-colored areas show that the model focuses on areas that reflect the depth and texture of the “Pitted Surface” defect, indicating that the model recognizes the subtle features of the defect during the prediction process.

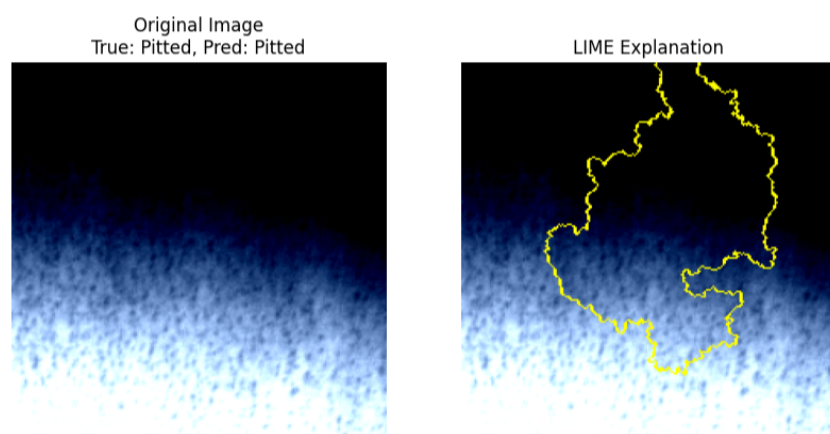


Figure 15. Pitted defect.

Figure 16 shows the characteristic texture that the model noted during the prediction of the “Rolled” defect. The LIME interpretation results provide visual confirmation that the model is focused on the unique pattern and texture of the “Rolled” defect, and this focus supports that the model’s predictions are based on recognizing the intrinsic nature of the defect, rather than just recognizing data patterns.

Finally, Figure 17 shows that the model pays attention to the long, irregular scratch lines when predicting the “Scratches” defect. The way the model centers its defect recognition around the scratch lines shows that the Hybrid-DC model has learned the linear pattern of the “Scratches” defect accurately.

This LIME-based interpretation lays the foundation for manufacturing process experts to gain a deeper understanding of the model’s predictive process. In particular, the visualization of how the model perceives and analyzes each feature of a steel surface defect provides a basis for confidence in the model’s predictive accuracy. Based on these interpretations, experts can understand the discrepancy between the structural pattern of the defect and the model’s perception and provide feedback to improve the model when prediction errors occur. The explainability of Hybrid-DC models can also be leveraged to increase the transparency of defect detection in manufacturing processes and provide confidence in

the predicted defects, maximizing process efficiency and enhancing quality control. In this way, the interpreted results of the model can be used as an important basis to facilitate collaboration with manufacturing process experts to improve prediction accuracy and increase the practicality of defect detection. Overall, the Hybrid-DC model proved to be effective in detecting and classifying various defects on steel surfaces with high precision. The ability to accurately detect defects of various shapes and sizes can be an important factor in maintaining consistent quality control in the manufacturing process. This performance suggests that the implementation of an automated steel surface defect inspection system can reduce the reliance on human resources in manufacturing, reducing costs and increasing productivity at the same time.

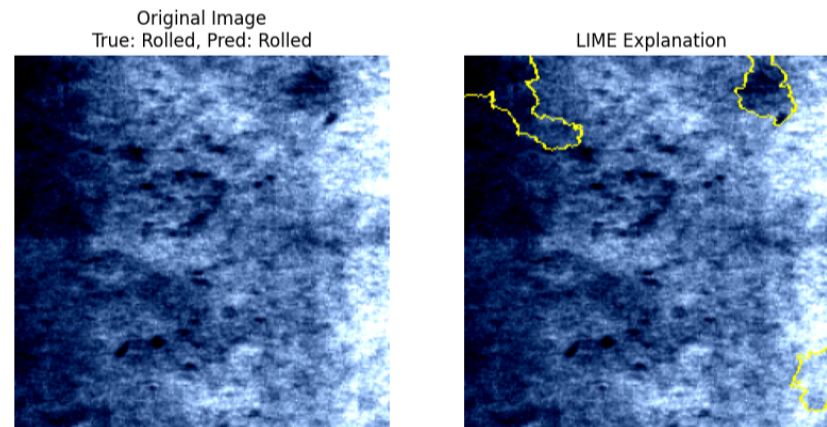


Figure 16. Rolled defect.

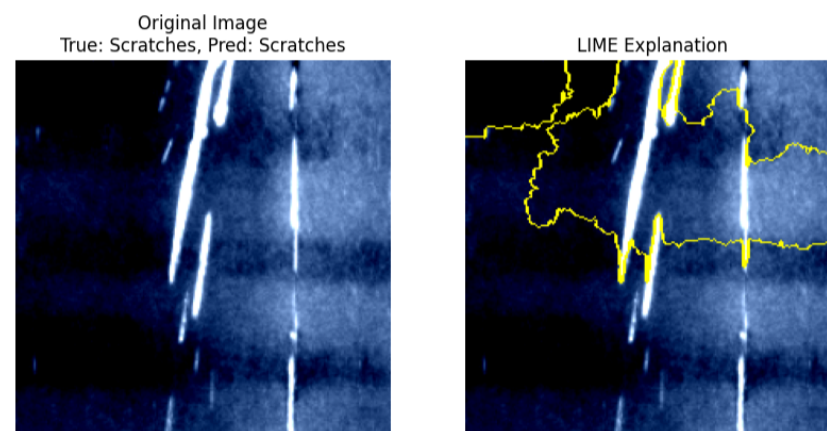


Figure 17. Scratches defect.

5. Discussion

The Hybrid-DC model proposed in this study has demonstrated excellent performance in surface defect detection and classification tasks. However, this study faces several limitations that should be addressed in future research.

First, the model relied on the NEU-DET dataset, which consists of a limited set of defect types and images collected under uniform conditions. This dataset may not fully reflect the variability found in real-world steel manufacturing environments, where defect types and conditions can vary significantly. To enhance the robustness and generalization of the model, future research should focus on collecting multi-source datasets that encompass a broader range of defect types and environmental variations. Additionally, data-augmentation techniques can be used to further improve the model's resilience to varying conditions. Further research will be required to investigate whether the method proposed in this work can achieve fine-grained cross-domain recognition. In particular, it is important to build mechanisms that can distinguish between fine feature differences in defects, and this will

require further research to leverage techniques such as domain adaptation to improve the generalization performance of the model. This future research will allow the model to better adapt to new, unseen defect types or environmental conditions, which is critical for real-world applications in diverse manufacturing contexts.

Second, the high computational cost associated with the Hybrid-DC model's complexity may limit its practical applicability in real-time flaw detection scenarios. The model's large architecture leads to high memory usage and slower inference speeds, which could be problematic in industrial settings where real-time analysis is required. To address this, future work should focus on improving computational efficiency through techniques such as knowledge distillation and network pruning. These methods can help reduce the model's size and computational demands while preserving performance. In addition, optimization techniques such as pruning and quantization need to be further explored to improve real-time classification accuracy, maximizing the lightweight property and efficiency of the model. These techniques will play an important role in reducing the high computational costs in manufacturing environments and making models more responsive. By adopting these optimization techniques, the model's suitability for deployment in real-time, high-throughput manufacturing environments can be significantly enhanced.

Third, the prediction process of the Hybrid-DC model currently operates as a black box, which can make it difficult for practitioners in the industrial field to understand and trust the model's decision-making process. Transparency in model predictions is particularly important in high-stakes environments such as steel manufacturing, where understanding the rationale behind decisions can help operators make informed choices. Therefore, future research needs to specifically address how explainable AI (XAI) techniques can be used to more transparently visualize and explain the model's predictive process and rationale for decisions. In particular, the XAI technique LIME was useful in explaining the image regions that the model paid attention to when detecting defects, which allowed us to understand how closely the features learned by the model matched the actual defects. Future work needs to apply more sophisticated XAI techniques to the Hybrid-DC model to clarify the rationale behind the model's decisions and integrate them with visual tools that can be easily understood in a manufacturing environment. These efforts will increase the transparency and reliability of the model, especially in high-risk manufacturing environments, making the results more credible to practitioners in the field. Implementation of these XAI methods can bridge the gap between the model's complex decisions and the interpretation needs of industry experts, thus enhancing both transparency and reliability.

To this end, future research should focus on expanding the scope of application of the Hybrid-DC model beyond steel manufacturing to other industries that require high-precision inspection, such as semiconductor manufacturing, automotive parts inspection, and aircraft parts maintenance. This extension will validate the generalization performance of the Hybrid-DC model and explore its potential for solving multiple defect detection problems. Optimization studies are also needed for real-time applications of the Hybrid-DC model. In addition to lightweighting the model, real-time processing capabilities can be enhanced by integrating hardware acceleration techniques and software optimizations that improve the model's real-time processing capabilities, which will increase its practical applicability in industry. Furthermore, research is needed to transparently explain and visualize the prediction process of Hybrid-DC models by applying explainable AI techniques. This will be an important factor in enhancing the reliability of the model and promoting its adoption in the manufacturing environment.

6. Conclusions

In this study, we proposed Hybrid-DC, a hybrid deep-learning model combining ResNet and ViT, to improve performance in steel surface defect detection and classification tasks. By combining the low- and high-level feature-extraction capabilities of ResNet with the global contextual learning capabilities of ViT, the Hybrid-DC model was designed to effectively learn and classify various defect types and complex visual patterns. Experimental

results show that the Hybrid-DC model outperforms the traditional ResNet, MobileNetV2, and ViT models in steel surface defect detection and classification, achieving high accuracy and low loss values. The superiority of the Hybrid-DC model was demonstrated through several performance metrics. In the Micro Metrics analysis, Hybrid-DC achieved the highest micro precision, recall, and F1 scores, achieving even and high prediction performance for all defect types. This indicates that the Hybrid-DC model can weight each defect class equally, minimizing the overall number of errors and performing consistently across the dataset. In the Weighted Metrics results, the Hybrid-DC model also outperformed the other models in evaluations that accounted for class imbalance, demonstrating its potential to effectively handle the varying conditions and complexity of real-world manufacturing environments. In particular, the Hybrid-DC model maintained high precision, recall, and F1 scores even for datasets with class imbalances, suggesting that it can achieve the accuracy and reliability required by industrial settings. The structural design of the Hybrid-DC model is expected to play a key role in the more precise detection and classification of various defects in steel manufacturing. This study demonstrates that the effective combination of a hybrid attention layer and an attention fusion mechanism can enhance the feature-extraction ability of the model, overcoming the limitations of existing models in recognizing and classifying defects on complex steel surfaces. These structural improvements allow the model to converge rapidly at the beginning of training, enabling it to achieve good performance in fewer training epochs. This suggests that the Hybrid-DC model has the potential to be used effectively in real-time defect detection.

This study also shows that Hybrid-DC models can play an important role in improving the quality of steel products and can increase the efficiency of defect detection to achieve both production cost reduction and time reduction. Furthermore, the Hybrid-DC model can be effectively applied to a variety of other industries, including semiconductor manufacturing, automotive parts inspection, and aircraft parts maintenance. This extension provides an opportunity to explore the potential of the Hybrid-DC model as a novel solution to various flaw detection problems. In conclusion, this study demonstrated the validity of the Hybrid-DC model as a novel approach to steel surface defect detection and classification, which makes a valuable contribution to automation and quality control in manufacturing. Further research on successful applications of the Hybrid-DC model should continue in various industries, as it can play an important role in increasing productivity and quality across industries.

Author Contributions: Conceptualization, M.J. and M.Y.; methodology, M.J.; software, M.J.; validation, M.J.; formal analysis, M.J.; investigation, M.J. and M.Y.; resources, M.J.; data curation, M.J.; writing—original draft preparation, M.J.; writing—review and editing, M.J. and M.Y.; visualization, M.J.; supervision, J.J.; project administration, J.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by ICT Creative Consilience Program through the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (RS-2020-II201821).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. These data can be found here: <https://www.kaggle.com/datasets/kaustubhdikshit/neu-surface-defect-database> (accessed on 23 September 2024).

Acknowledgments: This work was supported by ICT Creative Consilience Program through the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (RS-2020-II201821).

Conflicts of Interest: Author Minyeol Yang was employed by the company Research & Development Team, THiRA-UTECH Co., Ltd. The remaining authors declare that the research was conducted in the

absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Wang, S.; Zhang, Y.; Wang, Z.; Yang, Z. Automatic detection and classification of steel surface defect using deep convolutional neural networks. *Metals* **2021**, *11*, 388. [\[CrossRef\]](#)
2. Demir, K.; Çakır, B.; Topçu, Y.; Kucuk, S. Automated steel surface defect detection and classification using a new deep learning-based approach. *Neural Comput. Appl.* **2023**, *35*, 8389–8406. [\[CrossRef\]](#)
3. Boudiaf, A.; Derouiche, M.; Boudjellal, M.; Bouchekima, R.; Hamma, W. Classification of surface defects on steel strip images using convolution neural network and support vector machine. *J. Fail. Anal. Prev.* **2022**, *22*, 531–541. [\[CrossRef\]](#)
4. Yue, B.; Xu, G.; Song, Z.; Zeng, S.; Zhang, Y. Rail surface defect recognition method based on AdaBoost multi-classifier combination. In Proceedings of the 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Lanzhou, China, 18–21 November 2019; IEEE: Piscataway, NJ, USA, 2019.
5. Aslam, Y.; Santhi, N.; Ramasamy, N.; Ramar, K. A Heuristic Fuzzy Clustering Approach for Defect Detection on Titanium Coated Metal Surface. *J. Adv. Res. Dyn. Control Syst.* **2018**, *10*, 467–474.
6. Chu, M.; Gong, R.; Wang, A. Strip steel surface defect classification method based on enhanced twin support vector machine. *ISIJ Int.* **2014**, *54*, 119–124. [\[CrossRef\]](#)
7. Gao, L.; Zhang, H.; Zhang, Y.; Huang, L.; Wang, J. Cas-VSwin transformer: A variant swin transformer for surface-defect detection. *Comput. Ind.* **2022**, *140*, 103689. [\[CrossRef\]](#)
8. Li, S.; Wu, C.; Xiong, N. Hybrid architecture based on CNN and transformer for strip steel surface defect classification. *Electronics* **2022**, *11*, 1200. [\[CrossRef\]](#)
9. Fu, G.; Gao, H.; Sun, M.; Shi, X.; Dong, B. A deep-learning-based approach for fast and robust steel surface defects classification. *Opt. Lasers Eng.* **2019**, *121*, 397–405. [\[CrossRef\]](#)
10. Zheng, Z.; Zhao, J.; Li, Y. Research on detecting bearing-cover defects based on improved YOLOv3. *IEEE Access* **2021**, *9*, 10304–10315. [\[CrossRef\]](#)
11. Damacharla, P.; Bhosale, V.; Rege, S.; Park, K. TLU-net: A deep learning approach for automatic steel surface defect detection. In Proceedings of the 2021 International Conference on Applied Artificial Intelligence (ICAPAI), Halden, Norway, 19–21 May 2021; IEEE: Piscataway, NJ, USA, 2021.
12. Zhou, H.; Yang, S.; Sun, Z.; Li, W.; Zhang, D.; Xie, D. Etdnet: Efficient transformer-based detection network for surface defect detection. *IEEE Trans. Instrum. Meas.* **2023**. [\[CrossRef\]](#)
13. Dai, Z.; Liu, H.; Le, Q.V.; Tan, M. Coatnet: Marrying convolution and attention for all data sizes. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Virtual Event, 6–14 December 2021; Volume 34, pp. 3965–3977.
14. Babic, M.; Göllies, M.; Eitzinger, C.; Müllner, P.; Postl, R. Image Based Quality Inspection in Smart Manufacturing Systems: A Literature Review. *Procedia CIRP* **2021**, *103*, 262–267. [\[CrossRef\]](#)
15. Wang, Z.; Chen, W.; Lu, Y.; Qin, Y.; Zhang, Y.; Wang, J. Towards Smart Remanufacturing and Maintenance of Machinery. In Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Vienna, Austria, 8–11 September 2020; IEEE: Piscataway, NJ, USA, 2020.
16. Gwon, G.-H.; Choi, D.; Park, S. CNN-Based Image Quality Classification for Bridge Inspection Using UAV. *IEEE Access* **2023**, *11*, 22096–22113. [\[CrossRef\]](#)
17. Ngo, H.Q.T. Designing an automated system for online inspection using Convolutional Neural Network (CNN) in image processing. *Results Eng.* **2023**, *19*, 100401. [\[CrossRef\]](#)
18. Indasyah, E.; Sigit, R.; Kurniawan, F.; Fathurrahman, A.; Rustamaji, E. Automated Visual Inspection System for Gear Surface Defect Detection Using Faster RCNN. In Proceedings of the 2023 International Conference on Advanced Mechatronics, Intelligent Manufacture and Industrial Automation (ICAMIMIA), Bali, Indonesia, 15–16 November 2023; IEEE: Piscataway, NJ, USA, 2023.
19. Roth, K.; Geiger, J.; Horbelt, D.; Maier, A. Towards Total Recall in Industrial Anomaly Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 19–24 June 2022; IEEE: Piscataway, NJ, USA, 2022.
20. Ruff, L.; Vandermeulen, R.; Goernitz, N.; Deecke, L.; Siddiqui, S.A.; Binder, A.; Müller, E.; Kloft, M. Deep One-Class Classification. In Proceedings of the 35th International Conference on Machine Learning (ICML), Stockholm, Sweden, 10–15 July 2018; PMLR: Stockholm, Sweden, 2018; Volume 80, pp. 4393–4402.
21. Venkataramanan, S.; Goyal, R.; Ruff, L.; Goernitz, N.; Deecke, L.; Siddiqui, S.A.; Müller, E.; Binder, A.; Kloft, M. Attention Guided Anomaly Localization in Images. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; Springer: Cham, Switzerland, 2020; pp. 485–503.
22. Yi, J.; Yoon, S. Patch SVDD: Patch-level SVDD for Anomaly Detection and Segmentation. In Proceedings of the Asian Conference on Computer Vision (ACCV), Kyoto, Japan, 30 November–4 December 2020; Springer: Cham, Switzerland, 2020.
23. Liznerski, P.; Roth, K.; Vandermeulen, R.; Bringmann, O.; Kloft, M. Explainable Deep One-Class Classification. In Proceedings of the 9th International Conference on Learning Representations (ICLR), Virtual, 3–7 May 2021.
24. Li, B.; Kim, S.; Wang, Y.; Gu, W. Facial Expression Recognition via ResNet-50. *Int. J. Cogn. Comput. Eng.* **2021**, *2*, 57–64. [\[CrossRef\]](#)

25. Sun, W.; Yao, B.; Chen, B.; He, Y.; Cao, X.; Zhou, T.; Liu, H. Noncontact Surface Roughness Estimation Using 2D Complex Wavelet Enhanced ResNet for Intelligent Evaluation of Milled Metal Surface Quality. *Appl. Sci.* **2018**, *8*, 381. [[CrossRef](#)]
26. Yang, Y.; Wang, H.; Jiang, D.; Hu, Z. Surface Detection of Solid Wood Defects Based on SSD Improved with ResNet. *Forests* **2021**, *12*, 1419. [[CrossRef](#)]
27. Wang, C.; Yang, J.; Jie, H.; Tian, B.; Zhao, Z.; Chang, Y. An uncertainty perception metric network for machinery fault diagnosis under limited noisy source domain and scarce noisy unknown domain. *Adv. Eng. Inform.* **2024**, *62*, 102682. [[CrossRef](#)]
28. Wang, C.; Tian, B.; Yang, J.; Jie, H.; Chang, Y.; Zhao, Z. Neural-transformer: A brain-inspired lightweight mechanical fault diagnosis method under noise. *Reliab. Eng. Syst. Saf.* **2024**, *251*, 110409. [[CrossRef](#)]
29. Jian, Z.; Wang, X.; Zhang, X.; Su, R.; Ren, M.; Zhu, L. Task-Specific Near-Field Photometric Stereo for Measuring Metal Surface Texture. *IEEE Trans. Ind. Inform.* **2024**, *20*, 6019–6033. [[CrossRef](#)]
30. Ju, Y.; Lam, K.; Xie, W.; Zhou, H.; Dong, J.; Shi, B. Deep Learning Methods for Calibrated Photometric Stereo and Beyond. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 7154–7170. [[CrossRef](#)]
31. Heo, B.; Yun, S.; Han, D.; Choe, J.; Oh, S.J. Rethinking spatial dimensions of vision transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 11–17 October 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 11936–11945.
32. Madan, N.; Ramesh, P.; Bhattacharya, A.; Varma, M.; Viswanathan, B.; Sarvadevabhatla, R.K. Self-supervised masked convolutional transformer block for anomaly detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**. [[CrossRef](#)] [[PubMed](#)]
33. Ullah, W.; Khan, S.; Khan, S.; Rahim, S.; Bilal, M.; Niazi, M. TransCNN: Hybrid CNN and transformer mechanism for surveillance anomaly detection. *Eng. Appl. Artif. Intell.* **2023**, *122*, 106194. [[CrossRef](#)]
34. Lee, Y.; Kang, P. Anovit: Unsupervised anomaly detection and localization with vision transformer-based encoder-decoder. *IEEE Access* **2022**, *10*, 46717–46724. [[CrossRef](#)]
35. Liu, J.; Wang, J.; Zhang, Y.; Huang, L.; Li, X.; Zhang, Y. Cnnformer: Transformer-Based Semantic Information Enhancement Framework for Behavior Recognition. *IEEE Access* **2023**, *11*, 156789–156804. [[CrossRef](#)]
36. Peng, Y.; Wang, Y. CNN and transformer framework for insect pest classification. *Ecol. Inform.* **2022**, *72*, 101846. [[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.