Hybrid Deep Learning Framework for Accurate Surface Defect Detection Using Autoencoder and CNNs

Niphat Craypo, Anupong Banjongkan, and Anantaporn Hanskunatai *

Department of Computer Science, School of Science, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand
Email: niphat.cr@kmitl.ac.th (N.C.); anupong.ba@kmitl.ac.th (A.B.); anantaporn.ha@kmitl.ac.th (A.H.)
*Corresponding author

***Abstract*—Surface defect detection is paramount in industrial quality control. Conventional methods, which often rely on human inspection or manually engineered statistical models, frequently fail to accurately detect and classify defects, particularly on complex surfaces or those with intricate features. Human inspection is inherently inconsistent and prone to errors due to fatigue, while traditional machine vision systems often lack the sensitivity to clearly identify small or low-contrast defects. This paper proposes a hybrid deep learning framework, termed Autoencoders-Convolutional Neural Networks (AE-CNNs), DefectNet, for surface defect classification, AE, and CNNs to enhance both accuracy and efficiency. The AE is employed to extract and compress reliable features from surface images into latent representations, which are subsequently classified by a CNN enhanced through transfer learning using InceptionV3. The CNN is fine-tuned from a pretrained model with customized fully connected layers to adapt to specific defect characteristics, while the AE is trained exclusively on non-defective images. The encoded features produced by the AE serve as the input to the CNN. The proposed model is evaluated on standard benchmark datasets comprising diverse surface defect types and compared against Anomaly Detection with Autoencoder (ADA), Visual Geometry Group (VGG16), Inception-based Convolutional Neural Network Long Short-Term Memory (In-CNNLSTM), and DTL_Inception_v3. Experimental results demonstrate the superior performance of the proposed method, achieving classification accuracies ranging from 85.60% to 100% across five datasets, including a perfect 100% accuracy on the glass bottle neck dataset.**

***Keywords*—defect classification, autoencoder, convolutional neural network, feature extraction, machine vision, image processing**

I. Introduction

Surface defect detection is an important task applied in various domains such as safety, industrial manufacturing, and agriculture. Traditional methods for detecting surface defects and foreign objects rely heavily on human expertise and manually engineered statistical models to identify abnormalities. However, these conventional approaches often fail to effectively distinguish between defective and non-defective objects. They also require significant time and resources for inspection and are generally incapable of classifying different types of defects in high-dimensional data, such as images.

Machine learning techniques have been widely adopted for image-based defect classification tasks. Li *et al.* [1] proposed an enhanced Support Vector Machine (SVM) model for classifying surface defects on highly reflective steel balls. The model utilizes Principal Component Analysis (PCA) for dimensionality reduction and integrates an improved K-fold cross-validation with grid search for efficient parameter optimization, thereby increasing model robustness. This approach achieved a high classification accuracy of 97.15%, demonstrating its effectiveness for critical industrial quality control. Zhang *et al.* [2] introduced an unsupervised texture defect detection method that combines PCA and Histogram-Based Outlier Score (HBOS) to address limitations in industrial scenarios with few unlabeled samples and low-contrast defects. PCA is used for feature extraction to enhance the defect-background difference, and HBOS subsequently locates the anomaly. The method significantly outperformed deep learning approaches for small, low-contrast defects while utilizing only four unlabeled training samples per class, demonstrating its high efficiency for industrial quality control. Cao *et al*. [3] proposed a defect classification method for camshaft surfaces, known as the Feature Fusion Classification Method based on Defect Similarity Measurement (FFCM-DSM), which measures defect similarity using the Euclidean distance between class centers. To address challenges in inspecting complex surfaces and similar-looking defects, they developed a 360-degree optical inspection system and deployed a two-stage classification approach using both deep learning and SVM. The method showed improved accuracy in classifying camshaft surface defects.

Manuscript received September 1, 2025; revised October 20, 2025; accepted December 1, 2025; published May 27, 2026.

Subsequently, Deep Neural Networks (DNNs) have been developed to mitigate certain limitations of traditional machine learning approaches in defect classification, particularly in surface image analysis, where they offered more accurate detection and classification of defects. Rustam *et al.* [4] addressed the challenge of expensive and expert-dependent early identification of patchouli leaf disease, a critical issue affecting Indonesia's global supply of patchouli oil. They proposed a lightweight Convolutional Neural Network (CNN) architecture for classifying patchouli leaves as either diseased or healthy. Their CNN model, utilizing three convolution layers, a dense layer, and a dropout layer, outperformed five well-known models (including EfficientNetB0 and VGG16) in terms of classification accuracy, thus validating its potential to reduce identification costs and prevent disease transmission in propagation. Abdullah *et al.* [5] developed a leather image classification system using deep learning techniques. Their method employed InceptionV3 for classifying leather images into either good or defective quality, and Mask R-CNN for locating defects in the defective samples. This system significantly reduced human inspection errors and achieved improved accuracy compared to traditional image processing approaches. Baiganova *et al.* [6] introduced an automated defect detection system for manufacturing using an enhanced VGG16-based CNN. Their model incorporated batch normalization and dropout techniques to improve generalization and reduce overfitting. Through experiments on a benchmark dataset, the enhanced VGG16 achieved higher accuracy and better feature extraction than conventional CNNs and the original VGG16, particularly in detecting various defect types such as surface irregularities, scratches, and deformations. Wang *et al.* [7] proposed a supervised object detection network to address the challenges of multi-scale, irregular defects and the task conflict (coupling) between classification and regression often found in standard deep learning models used for electronic panel surface defect detection. The network employed two key strategies: a prediction box generation strategy based on a double branch structure and a detection head that decouples the regression and classification tasks. This method demonstrated its superior effectiveness for this specific industrial application. Wei *et al.* [8] addressed the challenge of high defect variability in metal surface detection by proposing RFAConv-CBM-ViT, an enhanced vision transformer [9]. The model utilized Receptive-Field Attention Convolution (RFAConv) to adaptively handle multi-scale defects and integrated the Context Broadcasting Median (CBM) method to effectively suppress noise. The RFAConv-CBM-ViT demonstrated competitive performance across public datasets, thereby significantly advancing the accuracy of industrial defect detection. Permatasari *et al.* [10] introduced a lightweight approach for surface defect classification using Mobile Vision Transformer (MobileViT), specifically targeting scenarios with limited data. MobileViT offered efficient classification of hot-rolled steel surface defects using fewer parameters. The study demonstrated that MobileViT was well-suited for smart manufacturing environments where dataset size and hardware capacity were constrained. Vasan *et al.* [11] proposed a method for detecting and classifying surface defects on hot-rolled steel using the ViT architecture. The model was trained on a public dataset comprising six defect categories and optimized through various hyperparameter settings. The study also highlighted the potential of ViT as a standalone framework for industrial detection and condition monitoring. Jeong *et al.* [12] proposed Hybrid-DC, a hybrid deep learning model that combined ResNet-50 and ViT for steel surface defect classifications. The model utilized ResNet-50 for multi-level feature extraction and ViT for capturing the global context, enhanced by a hybrid attention mechanism. This integration enabled effective handling of complex surface patterns and showed strong potential for real-time defect detection in industrial applications.

Conversely, traditional Machine Learning (ML) approaches for defect classification typically rely on handcrafted image processing algorithms for feature extraction, followed by classification using conventional ML models. However, these techniques are not inherently designed for image data and often struggle with large-scale or complex visual patterns due to limited feature representation capabilities. On the other hand, deep neural networks can automatically learn superior features with higher accuracy. Nonetheless, they require large amounts of labeled data, including both defective and non-defective surface images, for effective training and accurate defect recognition.

To address these limitations, this study presents a novel deep learning-based framework for surface defect detection and classification by integrating two neural network components with complementary roles, referred to as AE-CNNs DefectNet. Unlike existing hybrid AE-CNN architectures, the proposed framework introduces a distinct two-stage learning design. In the first stage, the Autoencoder is trained exclusively on defect-free samples to learn the "normal" surface representations, and its encoder produces a 3-channel latent feature map that is structurally compatible with the CNN architecture. This latent representation not only highlights potential defect regions but also reduces redundant surface texture information. In the second stage, a fine-tuned InceptionV3 network, pre-trained on ImageNet, is employed to classify the encoded features efficiently and accurately. This design enhances defect visibility, reduces computational complexity, and achieves superior generalization across multiple surface defect datasets. The proposed AE-CNNs DefectNet is evaluated on several benchmark datasets comprising diverse defect types, and its performance is compared with conventional and deep learning-based defect detection approaches. Experimental results demonstrate that the proposed framework achieves high accuracy and robust defect classification, effectively reducing inspection errors and increasing confidence in product quality assurance.

The remainder of this paper is organized as follows: Section II provides an overview of the surface defect

datasets. Section III describes the proposed AE-CNNs DefectNet framework for surface defect classification. Section IV presents the experimental setup and results. Finally, Section V offers the conclusion and outlines future research directions.

II. Overview of Surface Defect Image

In manufacturing, surface defect inspection is essential for product quality assurance. Defects such as scratches, cracks, stains, and tears must be accurately identified to ensure product reliability. To build representative datasets, surface images are captured using high-resolution industrial or DSLR cameras under controlled LED lighting to minimize shadows and reflections. Furthermore, a plain background is used to enhance defect visibility. The collected images, containing both real and artificial defects of various sizes and appearances, are then categorized by defect type for training machine-learning or deep-learning models.

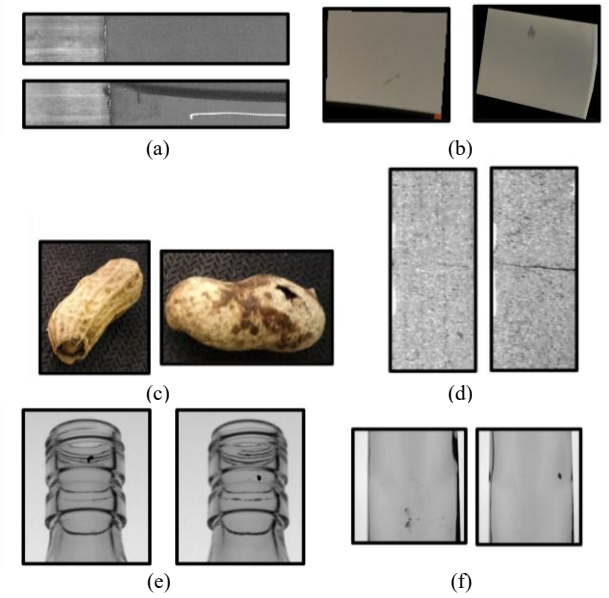

Fig. 1. Example images surface defect datasets: (a) AITEX; (b) GKN; (c) Groundnut; (d) KSDD; (e) Glass bottle neck; (f) Glass bottle body.

Example images from the AITEX fabric dataset are shown in Fig. 1(a). This dataset consists of grayscale images captured directly from a fabric production line [13]. It includes a total of 245 images, with 140 defect-free and 105 defective samples. The GKN dataset, provided by GKN (Guest, Keen, and Nettlefolds) and illustrated in Fig. 1(b), is a public dataset provided by Zhou [14] from the University of Connecticut. It contains images of blade surfaces, including 203 defect-free (good) samples, 149 with scratches, and 48 with nicks. The groundnut defect dataset, as shown in Fig. 1(c), applies image-based technology to the agricultural domain. This dataset includes 157 non-defective and 132 defective samples. The Kolektor Surface Defect Dataset (KSDD), illustrated in Fig. 1(d), is a real-world dataset designed for industrial surface defect detection on semi-finished products [15]. KSDD contains 399 images with 52 defective and 347 defect-free samples. Finally, the glass bottle defect datasets were collected from an industrial bottle inspection system. Fig. 1(e) shows example defects located at the neck of the bottles, consisting of 138 non-defective and 150 defective samples, while Fig. 1(f) shows defects on the bottle body, consisting of 131 non-defective and 161 defective samples [16].

III. The Proposed Method

This section describes the proposed AE-CNNs DefectNet method for surface defect classification, which is divided into two main stages. The first stage involves encoding the surface defect images using an Autoencoder (AE) to obtain a compact image representation that serves as feature encodings of the input samples. The design and architecture of the AE, including its underlying concept and the rationale for its use, are also presented. Furthermore, we explain how the output of the encoder is structured to be compatible with the classification process. The second stage focuses on the classification process. This includes the design and fine-tuning of a CNN, as well as training strategies that incorporate the encoded features generated by the AE. Specific attention is given to the integration of these encoded representations into the CNN pipeline to improve classification performance.

A. Surface Features Representation Using Autoencoder

Autoencoders are widely applied in dimensionality reduction because the encoder generates lower- dimensional codes in the latent space [17, 18]. The code can then represent the original high-dimensional image data while still retaining essential information. These codes are subsequently transformed back into image samples by the decoder, ensuring that the encoded representations accurately reflect the original data samples. The proposed AE architecture is composed of an encoder and a decoder designed to compress and reconstruct surface defect images, respectively. The proposed Autoencoder architecture receives an RGB image of size (364×364×3) and encodes it into a compact latent representation through a series of convolutional layers. The encoder consists of six Conv2D layers with 2×2 kernels, ReLU activations, and with same padding. It begins with 18 filters, followed by layers with 28, 58, 18, and 3 filters, respectively, interleaved with MaxPooling2D layers that gradually reduce spatial resolution while preserving essential texture information. The resulting feature map forms the latent space used for reconstruction. The decoder mirrors the encoder structure, employing successive Conv2D and UpSampling2D layers to progressively restore the spatial dimensions of the input image. All decoder layers use ReLU activations except for the final Conv2D layer, which uses a sigmoid activation to produce the reconstructed output. Overall, the Autoencoder contains 23,904 trainable parameters, representing a lightweight yet effective architecture capable of capturing fine-grained surface textures and subtle defect patterns.

The final layer of the encoder uses three filters, resulting in an output consisting of three feature maps. Crucially, these maps can be interpreted as a three-channel

representation, similar to an RGB image, when reshaped into the original image format. This design ensures that the encoded output preserves the spatial structure and channel configuration needed for the subsequent classification task. Furthermore, when designing the output of the encoder, it is important to avoid excessive downsampling, which could reduce the feature map below the minimum input size required by the subsequent CNN classifier. The encoder network $E(\bullet)$ transforms an input image $X \in \mathbb{R}^{H \times W \times C}$ into a latent representation $Z \in \mathbb{R}^{h \times w \times 3}$, as shown in Fig. 2.

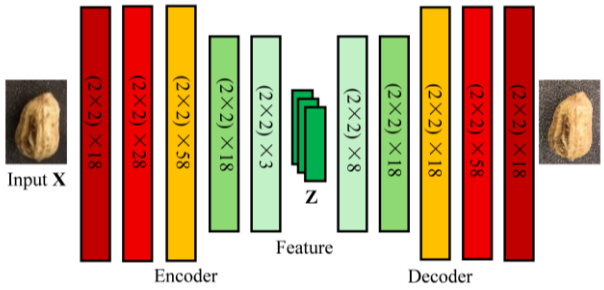

Fig. 2. The proposed Autoencoder (AE) architecture.

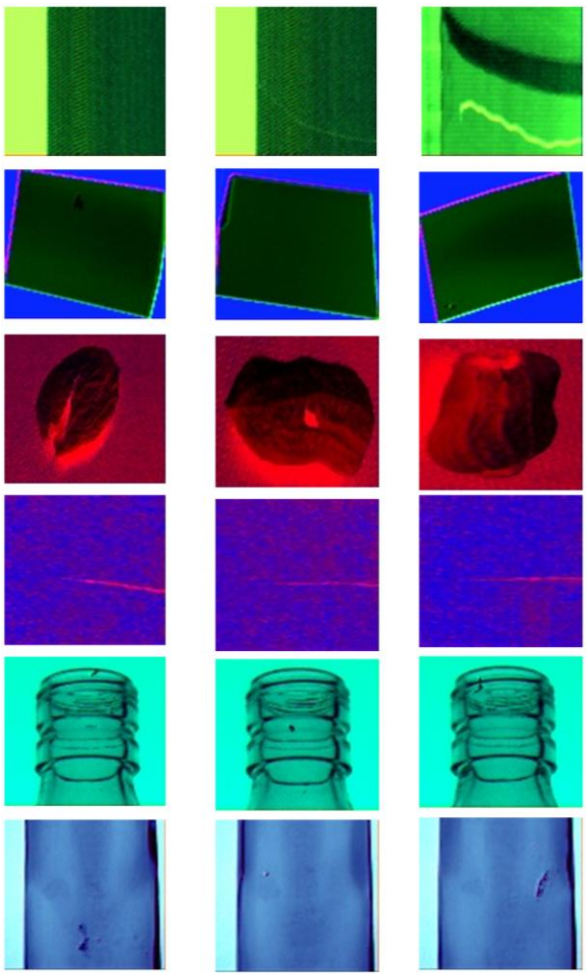

Fig. 3. The latent representation Z generated by the encoder of the autoencoder.

During the training process, all input images are first resized to 364×364×3 pixels to ensure a uniform input dimension for the Autoencoder (AE). The pixel values are normalized to the range [0, 1] to facilitate stable and efficient training of the networks. The AE is trained in a one-class training scheme using only defect-free images for 100 epochs to learn a compact latent representation of normal texture patterns. The AE is trained using the Adam optimizer (learning rate = 0.001) with a batch size of 8. The Mean Squared Error (MSE) loss is minimized to achieve accurate image reconstruction. After training, the decoder is discarded, and the encoder is retained as a feature extractor. The encoder part is used to create the latent representation Z, which is then used as input to the CNN classifier. Consequently, defect regions become more pronounced in the encoded representation, as illustrated in Fig. 3.

B. The Classification Process

In the AE-CNNs DefectNet framework, we employ a transfer learning approach using the InceptionV3 architecture as the base model for surface defect classification [19]. The base model is initialized with pre-trained weights from ImageNet, and its original classification layers are removed to allow customization for the target defect classification task. The output $E(X)$ of the encoder part Z of the autoencoder produces a latent representation of the input image in the form of a 3-dimensional tensor with shape ($h_{encoded}$, $w_{encoded}$, $c_{encoded}$). This tensor serves as a compact and informative feature representation that captures the most significant structural characteristics of the original surface image. By using only defect-free samples during training, the encoder is encouraged to retain essential features that represent the normal pattern of the surface, while discarding irrelevant details or defects. This compressed representation is then used as the input to the CNN classifier, allowing the network to focus on the most relevant features for defect classification, while also reducing the computational complexity and mitigating the risk of overfitting associated with high-dimensional input data. All layers of the base model are fine-tuned during this process. The output feature maps from InceptionV3 are processed using Global Average Pooling (GAP) to reduce dimensionality and then passed through a fully connected dense layer with 1024 units and ReLU activation, followed by a dropout layer (with a rate of 0.5) to mitigate overfitting. The final dense output layer is adaptable for both binary and multi-class classification tasks, using sigmoid activation for two-class problems and SoftMax activation for multi-class problems, respectively. The model is compiled using the Adam optimizer, with binary cross-entropy loss (for binary classes), and categorical cross-entropy (for multi-class problems). The CNN classifier is trained for 500 epochs with a learning rate of 0.0001 and a batch size of 8. Early stopping is employed to avoid overfitting, and model performance is evaluated using accuracy, precision, recall, F1-Score, and AUC-ROC metrics. The total number of trainable parameters of the AE-CNNs DefectNet is 23.93 million.

We use both defective and defect-free samples in the training data, which are encoded by the previously trained AE to obtain the latent representation Z. Let Z denote the

latent representation extracted from the encoder. Z serves as the training data that is fed into the first layer of the CNN, which is then trained on this data. The classification function f_θ, implemented using CNN, maps Z to the predicted output y:

$$\hat{y} = f_\theta(Z) \tag{1}$$

where f_θ is trained to classify defect and non-defect surface images. This CNN learns to classify defective and non-defective patterns from Z. This training strategy is efficient in terms of computational resource usage and contributes to the removal of irrelevant or noisy features, allowing the CNN to focus on the most informative representations extracted by the autoencoder.

For unseen data (i.e., test samples), each image is first processed through the pre-trained encoder of the AE to obtain its latent representation. This compressed feature vector is then used as the input to the CNN-based model, which predicts the final label y, indicating whether the sample is defective or non-defective. Let $x \in \mathbb{R}^{H\times W\times C}$ be the input image, and $Z = E(x) \in \mathbb{R}^d$ be the latent representation obtained by the encoder network. The classification output y is computed by applying a classifier f_θ to the latent Z, obtained from E, such that.

$$y = f_\theta\left(E(x)\right) \tag{2}$$

The complete workflow of the proposed AE-CNNs DefectNet is shown in Fig. 4.

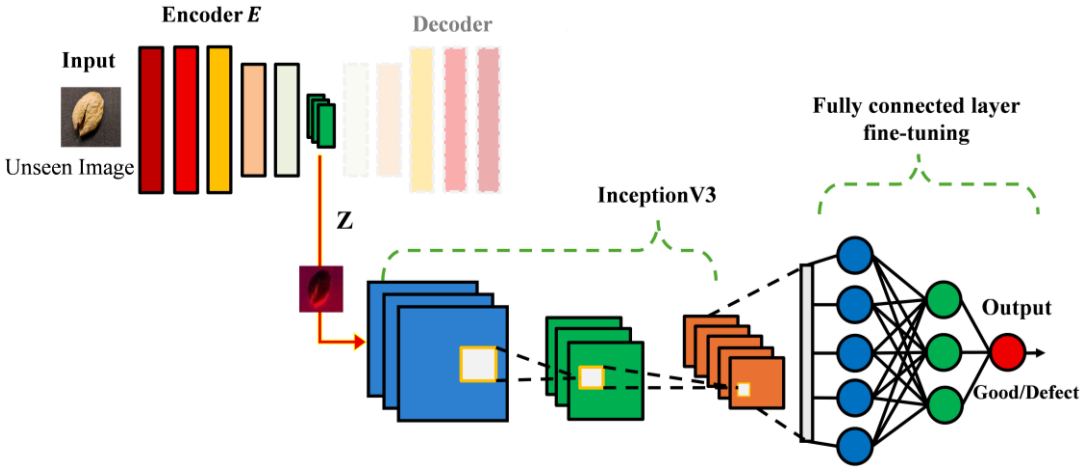

Fig. 4. The proposed AE-CNNs DefectNet framework.

C. Experimental Setup

All experiments were conducted on a workstation equipped with an AMD Ryzen 7 5800HS processor and an NVIDIA GTX 1650 GPU running on a Windows operating system. All images were resized to a uniform dimension of 364×364×3 pixels. The classification methods evaluated in this study include Anomaly Detection with Autoencoder (ADA), VGG16 [20], In-CNN-LSTM [21], and DTL_Inception_v3 [22], all of which were designed for surface defect classification. For the AE-CNNs DefectNet, the training process was set to 500 epochs, and the other deep learning methods were trained for the same number of epochs to ensure a fair comparison. A five-fold cross-validation was performed for each experiment, and the average performance was computed for comparison.

D. Evaluation Metrics

For the evaluation of multi-class classification performance, five primary metrics are used, including accuracy, precision, recall, F1-Score, and AUC-ROC, defined as follows. True Positive (TP) is the number of positive instances (defective samples) that are correctly predicted as positive. True Negative (TN) is the number of negative instances (non-defective samples) that are correctly predicted as negative. False Positive (FP) is the number of negative instances that are incorrectly predicted as positive. False Negative (FN) is the number of positive instances that are incorrectly predicted as negative. The performance metrics are calculated using Eqs. (3)–(6):

- Accuracy measures the overall correctness of the classification model.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

- Precision indicates the proportion of predicted positive cases that are actually positive.

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

- Recall represents the proportion of actual positive cases that are correctly identified.

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

- Recall provides a balance between the two.

$$F1-Score=2\times\frac{Precision\times Recall}{Precision+Recall} \quad (6)$$

IV. Experimental Results and Analysis

To evaluate the effectiveness of the proposed AE-CNNs DefectNet method, surface classification experiments were conducted on multiple datasets. The results were compared with those of other classification approaches to evaluate the performance of AE-CNNs DefectNet.

A. Experimental Results

This section first presents the ablation study, which analyzes the contribution of each component within the proposed AE-CNNs DefectNet, followed by the experimental results comparing its performance with several baseline methods on the benchmark surface defect datasets.

An ablation study was performed using three model variants: Autoencoder-only, CNN-only (InceptionV3), and the full AE-CNNs DefectNet. The results of this ablation study, conducted on six surface defect datasets, were summarized in Table I. The Autoencoder-only model achieved relatively low performance, ranging from 47% to 86% in accuracy, because it focused primarily on feature reconstruction rather than discriminative classification. The CNN-only model showed a significant performance improvement, confirming the effectiveness of deep convolutional features, but its performance dropped on datasets with limited samples or complex textures (e.g., Glass Bottle Body and Groundnut). Conversely, AE-CNNs DefectNet outperformed the other models on almost all datasets, achieving the highest accuracy and F1-Score, with notable improvements of up to 7.66% on the groundnut dataset over CNN-only and up to 9% on the glass bottle body dataset. The integration of the Autoencoder and CNN enabled effective feature compression and discriminative representation. This approach improved robustness to variations in texture and illumination. These results definitively demonstrated the added value of combining reconstruction-based learning (AE) and discriminative feature extraction (CNN), which enhanced both the generalization ability and defect detection accuracy of the proposed framework.

TABLE I. Ablation Study Results of the AE-CNNs DefectNet on Benchmark Surface Defect Datasets

Dataset	Metric	Autoencoder-only	CNN-only (InceptionV3)	AE-CNNs DefectNet
Groundnut	Accuracy	49.84 ± 5.31	78.86 ± 6.06	86.52 ± 5.13
	F1-Score	38.24 ± 3.57	78.57 ± 6.05	86.42 ± 5.30
KSDD	Accuracy	86.56 ± 3.89	88.22 ± 1.69	87.98 ± 4.28
	F1-Score	80.38 ± 5.57	85.91 ± 1.70	85.20 ± 5.07
Glass Bottle Neck	Accuracy	47.93 ± 5.44	100.00 ± 0.01	100.00 ± 0.00
	F1-Score	31.24 ± 5.86	100.00 ± 0.00	100.00 ± 0.01
Glass Bottle Body	Accuracy	56.02 ± 4.53	76.69 ± 3.85	85.60 ± 3.75
	F1-Score	40.34 ± 5.35	76.73 ± 3.80	85.64 ± 3.66
AITEX	Accuracy	83.48 ± 8.15	82.15 ± 5.28	86.17 ± 7.51
	F1-Score	83.32 ± 8.16	82.04 ± 5.32	86.09 ± 7.50
GKN	Accuracy	59.58 ± 5.03	81.25 ± 3.79	81.50 ± 4.23
	F1-Score	58.44 ± 6.32	81.19 ± 3.78	80.78 ± 4.27

The performance comparison on the groundnut dataset was presented in Fig. 5 and Table II. The proposed AE-CNNs DefectNet achieved the highest mean accuracy of 86.52% on the groundnut dataset, outperforming all baseline models, including VGG16, Inception-based Convolutional Neural Network Long Short-Term Memory (In-CNNLSTM), and DTL_Inception_v3. The relatively low standard deviation across the five folds indicated stable and reliable performance across flows. These results demonstrated that integrating the Autoencoder and CNN effectively enhanced feature representation and improved defect classification accuracy. In contrast, conventional models such as ADA and In-CNNLSTM exhibited lower stability and limited discriminative capability, demonstrating the robustness and generalization ability of the proposed hybrid framework.

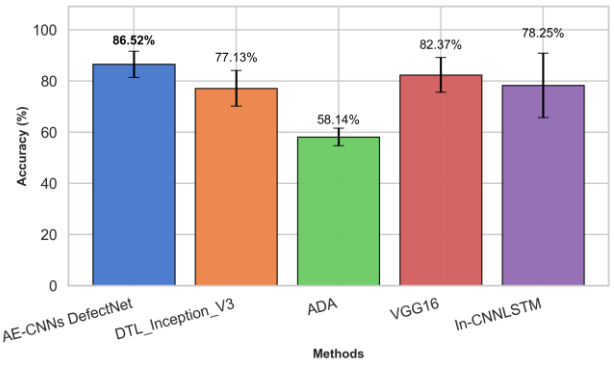

Fig. 5. The classification accuracy and standard deviation across five folds on the groundnut dataset.

TABLE II. Performance Comparisons of All Models on the Groundnut Dataset

Method	Accuracy	Precision	Recall	F1-Score	AUC
AE-CNNs DefectNet	86.52	87.05	86.52	86.42	86.41
ADA	58.14	78.06	58.14	64.82	51.09
VGG16	82.37	85.44	82.37	81.64	81.5
In-CNNLSTM	78.25	74.27	78.25	74.89	76.4
DTL_Inception_v3	77.13	77.75	77.13	76.95	76.84

The KSDD dataset results were visualized in Fig. 6 and summarized in Table III. The proposed AE-CNNs DefectNet achieved the highest average accuracy of 87.98% on this dataset, demonstrating stable and reliable performance across folds as indicated by a low standard deviation. The high classification accuracy arose from the majority class (defective samples) being correctly identified, which dominated the dataset. Conversely, the lower AUC-ROC value reflected the AE-CNNs DefectNet's limited discriminative ability to effectively rank the separation between the non-defective and defective classes, leading to suboptimal score ranking when adjusting classification thresholds. Compared with other models, such as VGG16 at 85.72% and DTL_Inception_v3 at 84.97%, the AE-CNNs DefectNet exhibited superior consistency and robustness in identifying subtle surface defects. Moreover, it maintained balanced precision, recall, and F1-Score, confirming its effectiveness in defect classification. The ADA method yielded the weakest performance, indicating its limited capability to handle complex surface textures.

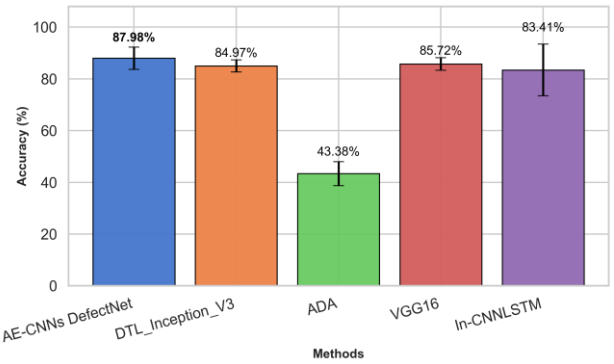

Fig. 6. The classification accuracy and standard deviation across five folds on the KSDD dataset.

TABLE III. PERFORMANCE COMPARISONS OF ALL MODELS ON THE KSDD DATASET

Method	Accuracy	Precision	Recall	F1-Score	AUC
AE-CNNs DefectNet	87.98	87.98	87.98	85.2	59.75
ADA	43.38	76.94	43.38	51.29	48.18
VGG16	85.72	83.04	85.72	81.57	55.36
In-CNNLSTM	83.41	86.97	83.41	82.07	59.89
DTL_Inception_v3	83.41	86.97	83.41	82.07	59.97

According to the cross-validation trend in Fig. 7 and results in Table IV, both AE-CNNs DefectNet and DTL_Inception_v3 demonstrated exceptionally consistent and reliable defect detection on the glass bottle neck dataset. These methods achieved 100% accuracy with zero standard deviation across all evaluation metrics. The In-CNNLSTM model followed with an average accuracy of 87.84%, while VGG16 and ADA performed notably lower, showing higher variability across folds. These results confirmed that the AE-CNNs DefectNet effectively captured subtle defect features around the bottle neck region and maintained robust, error-free performance across all evaluation metrics, validating the stability and generalization capability of the two-stage AE-CNN framework.

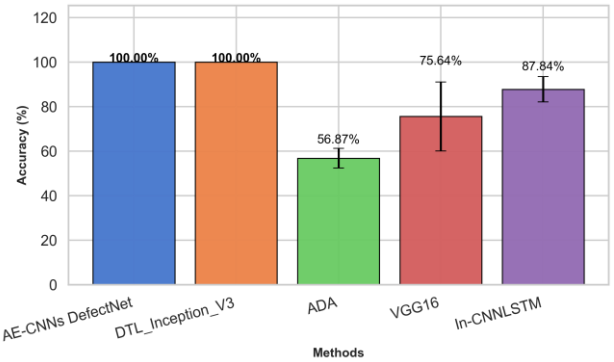

Fig. 7. The classification accuracy and standard deviation across five folds on the glass bottle neck dataset.

TABLE IV. PERFORMANCE COMPARISONS OF ALL MODELS ON THE GLASS BOTTLE NECK DATASET

Method	Accuracy	Precision	Recall	F1-Score	AUC
AE-CNNs DefectNet	100	100	100	100	100
ADA	56.87	78.28	56.87	63.9	51.02
VGG16	75.64	72.82	75.64	71.68	75.08
In-CNNLSTM	87.84	90.7	87.84	87.62	88.16
DTL_Inception_v3	100	100	100	100	100

As illustrated in Fig. 8 and summarized in Table V, the proposed AE-CNNs DefectNet achieved the highest mean accuracy of 85.60% on the glass bottle body dataset, providing consistently high evaluation scores. DTL_Inception_v3 and VGG16 followed with lower accuracy, while ADA exhibited moderate results. However, the In-CNNLSTM model showed the lowest performance across all metrics, indicating limited generalization capability on glass bottle body textures.

The experimental results of the AITEX fabric defect dataset are shown in Fig. 9 and presented in Table VI. The AE-CNNs DefectNet achieved the highest mean accuracy on this dataset, demonstrating robust and consistent performance. Although slight variation existed, the model maintained superior stability compared to other methods. The DTL_Inception_v3 followed closely with a slightly lower accuracy, while VGG16 and In-CNNLSTM exhibited moderate and less stable results. The ADA model showed the lowest performance across all

evaluation metrics, indicating limited generalization capability on complex fabric textures.

Unlike the previous datasets, Fig. 10 and Table VII showed that the DTL_Inception_v3 model achieved the highest raw accuracy on the GKN dataset. The proposed AE-CNNs DefectNet achieved performance comparable to that of DTL_Inception_v3. While slightly lower in accuracy, AE-CNNs DefectNet exhibited a more balanced performance across all evaluation metrics, demonstrating consistent and reliable defect detection capability. Conversely, ADA, In-CNNLSTM, and VGG16 produced noticeably lower and less consistent results.

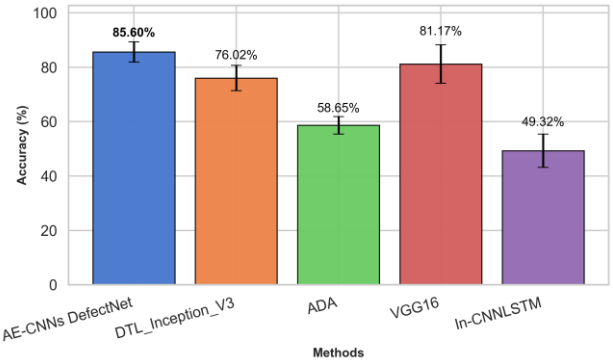

Fig. 8. The classification accuracy and standard deviation across five folds on the glass bottle body dataset.

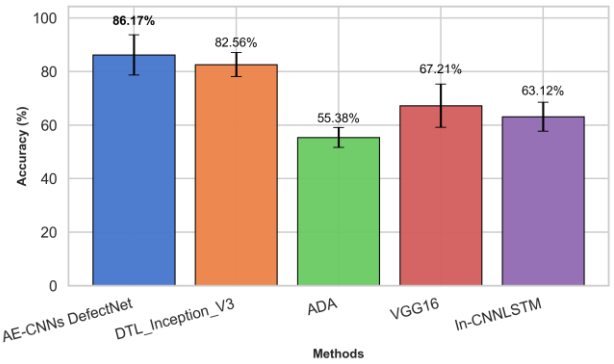

Fig. 9. The classification accuracy and standard deviation across five folds on the AITEX dataset.

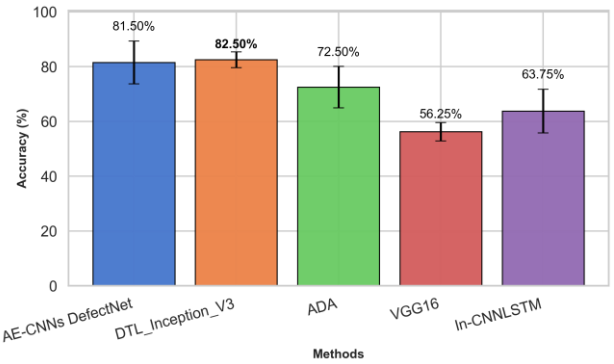

Fig. 10. The classification accuracy and standard deviation across five folds on the GKN dataset.

TABLE V. PERFORMANCE COMPARISONS OF ALL MODELS ON THE GLASS BOTTLE BODY DATASET

Method	Accuracy	Precision	Recall	F1-Score	AUC
AE-CNNs DefectNet	85.6	87.74	85.6	85.64	86.66
ADA	58.65	78.17	58.65	65.33	50.93
VGG16	81.17	86.08	81.17	80.82	82.92
In-CNNLSTM	49.32	53.75	49.32	46.3	51.7
DTL_Inception_v3	76.02	76.41	76.02	76.07	75.78

TABLE VI. PERFORMANCE COMPARISONS OF ALL MODELS ON THE AITEX DATASET

Method	Accuracy	Precision	Recall	F1-Score	AUC
AE-CNNs DefectNet	86.17	86.42	86.17	86.09	85.57
ADA	55.38	78.78	55.38	62.61	53.24
VGG16	67.21	76.74	67.21	63.25	66.35
In-CNNLSTM	63.12	70.91	63.12	60.92	64.99
DTL_Inception_v3	82.56	82.93	82.56	82.5	81.98

TABLE VII. PERFORMANCE COMPARISONS OF ALL MODELS ON THE GKN DATASET

Method	Accuracy	Precision	Recall	F1-Score	AUC
AE-CNNs DefectNet	81.5	83.50	81.5	80.78	81.01
ADA	72.5	73.22	72.5	72.3	72.52
VGG16	56.25	58.77	56.25	46.16	53.9
In-CNNLSTM	63.75	61.93	63.75	57.4	62.13
DTL_Inception_v3	82.5	83.57	82.5	82.47	82.9

The confusion matrices in Fig. 11 collectively demonstrated the classification behavior of the proposed AE-CNNs DefectNet across all evaluated datasets. Specifically, the confusion matrix values reflected the combined predictions obtained from the test portion of each fold during the K-fold cross-validation. The matrices for groundnut and KSDD, shown in Fig. 11(a) and 11(b) exhibited strong discrimination between non-defective and defective samples, with only minimal confusion. The model achieved perfect separation on the glass bottle neck dataset, as illustrated in Fig. 11(c), reflecting highly stable feature extraction. For the glass bottle body and AITEX, shown in Fig. 11(d) and 11(e), the network maintained reliable performance despite increased intra-class

variability. Finally, Fig. 11(f), which presented the multi-class GKN dataset, showed that AE-CNNs DefectNet accurately distinguished non-defective surfaces from nick and scratch defects, with only minor confusion among the visually similar defect categories.

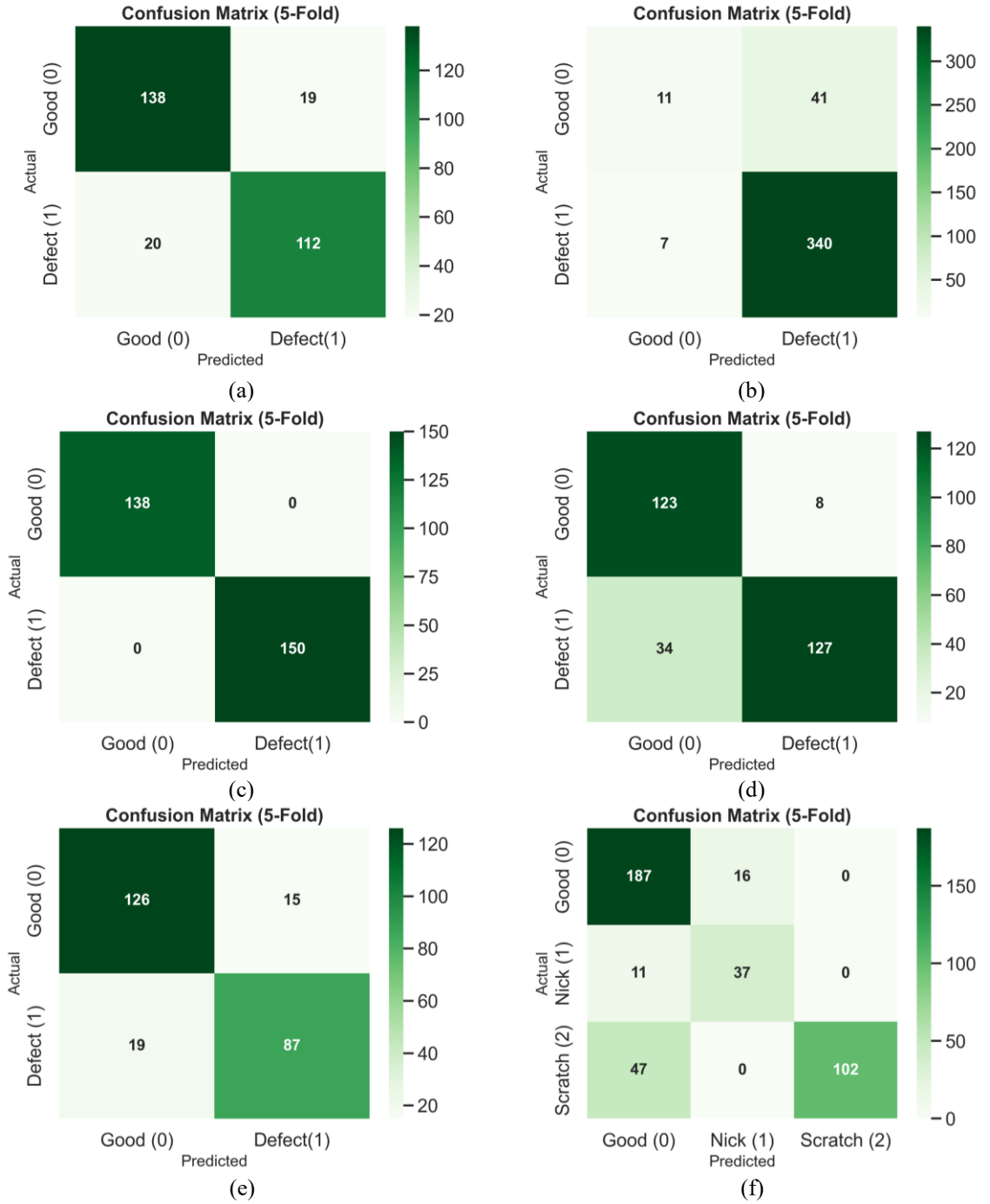

Fig. 11. A set of confusion matrices for the six evaluated datasets: (a) Groundnut; (b) KSDD; (c) Glass bottle neck; (d) Glass bottle body; (e) AITEX; (f) GKN.

Table VIII shows the training and inference times for all data in the training and testing sets, along with the number of trainable parameters for each model. AE-CNNs DefectNet required the longest training time at 208 s due to its two-stage training, but maintained fast inference at 3 s with 23.93M parameters. ADA and VGG16 had faster training times of 63 s and 61 s with lightweight and heavy architectures, respectively. In-CNNLSTM showed the fastest training at 37 s but the slowest inference at 15 s. DTL_Inception_v3 achieved intermediate training at 112 s and fast inference at 3 s with 24 M parameters. Overall, AE-CNNs DefectNet provided a good balance between accuracy and computational efficiency.

TABLE VIII. COMPUTATIONAL COST AND TRAINABLE PARAMETERS OF ALL MODELS

Model	Training Time (s)	Prediction Time (s)	Parameters (Million)
AE-CNNs DefectNet	208	3	23.93
ADA	63	2	0.024
VGG16	61	3	138
In-CNNLSTM	37	15	124.8
DTL_Inception_v3	112	3	24

The better performance of AE-CNNs DefectNet can be attributed to its two-stage architecture. The Autoencoder effectively learned compact latent representations of normal images, highlighting subtle defects when the images were encoded. The subsequent CNN classifier leveraged these latent features to distinguish defective from non-defective samples efficiently. Compared to the other defect classification models, VGG16 and DTL_Inception_v3 had larger parameter counts, which may lead to overfitting on small datasets. In contrast, In-CNNLSTM, although lightweight in training, suffered from slow inference due to sequential processing. ADA, with very few parameters, cannot capture detailed feature patterns, limiting its accuracy. Overall, the proposed AE-CNNs DefectNet achieved a favorable balance of high accuracy and computational efficiency through effective feature extraction and moderate model complexity.

V. CONCLUSION

This paper proposed AE-CNNs DefectNet, a robust framework integrating Autoencoders (AE) and Convolutional Neural Networks (CNNs) with transfer learning to address surface defect classification challenges in industrial applications. By training the AE exclusively on non-defective images, the model effectively learned to compress and represented salient features, subsequently classified using a fine-tuned InceptionV3-based CNN. An ablation study confirmed the significant value-added contribution of this integration, demonstrating that the proposed hybrid approach enhanced both generalization ability and defect detection accuracy over single-component models. Experimental evaluations were conducted on six benchmark defect datasets. The results consistently demonstrated that the AE-CNNs DefectNet outperformed or matched existing defect classification approaches, achieving superior performance on five datasets and comparable performance on the remaining one. The proposed method maintained strong performance metrics across all tested datasets. These findings confirmed the effectiveness, robustness, and generalizability of the proposed approach, establishing it as a highly practical solution for diverse real-world surface defect detection tasks.

Future work will focus on extending the proposed method to handle real-time defect detection and applying it to more complex industrial environments. Additionally, exploring lightweight models for deployment on edge devices represents a promising research direction.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Niphat Craypo conceived the idea, conducted the experiments and wrote the manuscript. Anantaporn Hanskunatai examined and analyzed the results, and reviewed the manuscript. Anupong Banjongkan wrote the manuscript. All authors contributed to editing and improving the manuscript and approved the final version.

FUNDING

This study is supported by the King Mongkut's Institute of Technology Ladkrabang Research Fund: KREF186813.

REFERENCES

[1] L. Li, T. Ren, and M. Feng, "Research on surface defect identification of steel balls based on improved K-CV parameter optimization support vector machine," *Advances in Mechanical Engineering*, vol. 15, no. 12, 2023. https://doi.org/10.1177/16878132231218586

[2] N. Zhang, Y. Zhong, and S. Dian, "Rethinking unsupervised texture defect detection using PCA," *Optics and Lasers in Engineering*, vol. 163, 107470, 2023. https://doi.org/10.1016/j.optlaseng.2022.107470

[3] J. Cao, H. Wu, W. Wang, T. Qasim, and D. Wang, "A visual inspection and classification method for camshaft surface defects based on defect similarity measurement," *IEEE Access*, vol. 12, pp. 74633–74648, 2024. https://doi.org/10.1109/ACCESS.2024.3395119

[4] Rustam, R. Noveriza, S. Khotijah *et al*., "Convolution neural network approach for early identification of patchouli leaf disease in Indonesia," *Journal of Image and Graphics*, vol. 12, no. 2, pp. 137–144, 2024. doi: 10.18178/joig.12.2.137-144

[5] A. B. Abdullah, M. Jawahar, N. Manogaran *et al.*, "Leather image quality classification and defect detection system using mask region-based convolution neural network model," *International Journal of Advanced Computer Science and Applications*, vol. 15, no. 4, 2024. http://dx.doi.org/10.14569/IJACSA.2024.0150455

[6] A. Baiganova, Z. Ubayeva, Z. Taskalyeva, L. Kaparova, R. Nurzhaubaeva, and B. Umirzakova, "Automated defect detection in manufacturing using enhanced VGG16 convolutional neural networks," *International Journal of Advanced Computer Science and Applications*, vol. 16, no. 4, 2025. http://dx.doi.org/10.14569/IJACSA.2025.0160487

[7] L. Wang, X. Huang, Z. Zheng, and H. Ruan, "Surface defect detection method for electronic panels based on double branching and decoupling head structure," *PLoS One*, vol. 18, no. 2, e0279035, 2023. https://doi.org/10.1371/journal.pone.0279035

[8] H. Wei, L. Zhao, R. Li, and M. Zhang, "RFAConv-CBM-ViT: Enhanced vision transformer for metal surface defect detection," *The Journal of Supercomputing*, vol. 81, no. 1, 155, 2025. https://doi.org/10.1007/s11227-024-06662-0

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," arXiv Preprint, arXiv:1706.03762, 2017. https://doi.org/10.48550/arXiv.1706.03762

[10] G. I. Permatasari, H. K. Pao, R. C. H. Pribadi, and M. Iqbal, "Mobile vision transformer for surface defect classification from a tiny dataset," in *Proc. 14th International Conference on Information &*

Communication Technology and System (ICTS), 2023, pp. 100–104. https://doi.org/10.1109/ICTS58770.2023.10330837
[11] V. Vasan, N. V. Sridharan, S. Vaithiyanathan, and M. Aghaei, "Detection and classification of surface defects on hot-rolled steel using vision transformers," *Heliyon*, vol. 10, no. 19, e38498, 2024. https://doi.org/10.1016/j.heliyon.2024.e38498
[12] M. Jeong, M. Yang, and J. Jeong, "Hybrid-DC: A hybrid framework using ResNet-50 and vision transformer for steel surface defect classification in the rolling process," *Electronics*, vol. 13, no. 22, 4467, 2024. https://doi.org/10.3390/electronics13224467
[13] J. Silvestre-Blanes, T. Albero-Albero, I. Miralles, R. Pérez-Llorens, and J. Moreno, "AFID: A public fabric image database for defect detection," *AUTEX Research Journal*, pp. 1–10, 2019.
[14] Q. Zhou. (2023). GKN blade surface defect dataset. *Mendeley Data V1*. [Online]. Available: https://doi.org/10.17632/3bh998k78g.1
[15] J. Božič, D. Tabernik, and D. Skočaj, "Mixed supervision for surface-defect detection: From weakly to fully supervised learning," *Computers in Industry*, vol. 129, 103458, 2021. https://doi.org/10.1016/j.compind.2021.103459
[16] N. Claypo, S. Jaiyen, and A. Hanskunatai, "Inspection system for glass bottle defect classification based on deep neural network," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 7, 2023. https://doi.org/10.14569/IJACSA.2023.0140738
[17] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006. https://doi.org/10.1126/science.1127647
[18] P. Vincent, H. Larochelle, I. Lajoie *et al*., "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res*., vol. 11, pp. 3371–3408, 2010.
[19] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826. https://doi.org/10.1109/CVPR.2016.308
[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv Preprint, arXiv:1409.1556, 2014.
[21] N. Claypo, S. Jaiyen, and A. Hanskunatai, "Face spoofing detection based on deep feature extraction and instance-based classification," *ICIC Express Letters*, vol. 17, no. 2, pp. 235–244, 2023.
[22] C. Song, B. Wang, and J. Xu, "Classifying tongue images using deep transfer learning," in *Proc. 2020 5th International Conference on Computational Intelligence and Applications (ICCIA)*, 2020, pp. 103–107. https://doi.org/10.1109/ICCIA49625.2020.00027