

RESEARCH ARTICLE

Enhancing Metal Surface Defect Recognition Through Image Patching and Synthetic Defect Generation

BEKHZOD MUSTAFAEV¹, **SUNGWON KIM²**, AND **EUNGSOO KIM³**¹Department of Artificial Intelligence Convergence, Graduate School, Busan University of Foreign Studies, Busan 46234, South Korea²Vazil Company Co., Ltd., Busan 46918, South Korea³Division of Electronic Robotics and Information Security Engineering, Busan University of Foreign Studies, Busan 46234, South Korea

Corresponding author: Eungsoo Kim (eskim@bufs.ac.kr)


This work was supported in part by the Research and Development Project for the Process Quality Technology Development of Vazil Company Ltd., under Grant RS-2023-00234322.

ABSTRACT Preventing surface defects of metal products during the production process is challenging due to manufacturing complexity, material properties and environmental factors. Relying on human inspectors for quality control can introduce human error, which increases the risk of delivering defective products to customers. To address these challenges, we propose an Inception-CNN model specifically designed for surface defect recognition in servo motor housings (SMHs). The model incorporates an inception module between convolutional layers to effectively capture multi-scale local information and extract complex and abstract features. Additionally, we introduce an image patching technique that enhances defect recognition for small defects by reducing image complexity while maintaining defect visibility. Moreover, we propose a surface defect generation GAN (SDG-GAN) method, a novel approach that addresses the data imbalance problem and improves the accuracy and robustness of the classification model through generating diverse and high-quality synthetic defect images. The original data was collected using a line scan camera installed in the SMHSI system. We ensured model generalization through 10-fold cross-validation using the SMHSD-P-GAN dataset. Evaluation results indicate that our classification model outperformed other CNN models and achieved strong generalization, with 99.40% accuracy in cross-validation and 99.23% on the original test data. This represents a substantial 32.31% improvement over a baseline CNN model trained on the SMHSD-O-TA dataset, underscoring the effectiveness of our proposed approaches in enhancing classification performance. Our method efficiently processes 12 images per second, making it ideal for real-time defect inspection in SMHs. Its successful integration into the SMHSI system confirms its practicality and effectiveness in real-world industrial applications.

INDEX TERMS Deep learning, surface defect recognition, convolutional neural network, image patching, synthetic defect generation.

I. INTRODUCTION

Servo motor housing (SMH) is one of the most important components of servo motors used in various industries, including robotics, automation, and automotive applications [1]. Its primary purpose is to provide robust structural support, protection, and encapsulation for the internal

The associate editor coordinating the review of this manuscript and approving it for publication was Wenbing Zhao .

components of servo motors. The manufacturing process of SMHs involves several precision operations, such as pointing, cleaning, coating, cold drawing, heat treatment, calibration, and polishing. Despite the use of these precise techniques, surface defects such as roll marks, scratches, and other imperfections can still occur during production. These kinds of surface defects in the motor housing can lead to reduced motor stability and overheating [2], bearing and even motor failure [3]. Early identification and prevention of

these defects are essential to ensure optimal motor performance, reliability, and safety. Ensuring high-quality SMHs requires addressing the challenge of preventing defects at their source. While eliminating defects during production is difficult, an alternative approach is to conduct thorough product quality inspection.

Traditional inspection methods rely on visual and manual inspection by human inspectors to identify surface defects on SMHs. However, this approach has limitations, including high costs associated with the need for many human inspectors to inspect a large quantity of products. Additionally, relying solely on human inspection introduces the potential for human error, which can result in the delivery of defective products to customers. To overcome these limitations, it is necessary to develop a robust and reliable inspection system capable of accurately classifying various types and sizes of surface defects in SMHs.

In recent years, advances in image processing [4], [5], [6], [7], [8], [9], machine learning [10], [11], [12], [13], [14], [15], and deep learning [16], [17], [18], [19], [20] techniques have revolutionized surface inspection, enabling more accurate, efficient, and automated inspection processes. Image processing and machine learning approaches offer valuable tools for surface inspection, but they have limitations regarding feature selection and generalization. Deep learning approaches, on the other hand, have shown remarkable performance in surface defect inspection tasks, including defect detection [17], [21], classification [19], [20], [22], [23], and segmentation [18], [24], [25]. These methods can automatically extract and learn meaningful features and patterns directly from image data. However, the challenge arises when dealing with real-time surface defect inspection systems that require continuous integration. Such systems need to constantly update the inspection algorithm to adapt to new defect types that may emerge over time. This poses a specific challenge for object detection and segmentation methods, as these approaches rely on annotated datasets for training, which can be time-consuming and expensive to create. Convolutional neural networks (CNN) based classification methods offer several advantages in the context of continuous integration for surface defect inspection. Firstly, they provide a flexible framework that allows easy integration of new defect types into the existing system. Instead of relying on handcrafted features or annotation-based approaches, CNN-based classification models can be updated and fine-tuned with additional labeled data, enabling the system to adapt to emerging defect classes. In this paper, we address the following challenges faced in employing deep learning for surface defect recognition of SMHs.

- 1) *Small Size Defects*: Most of the defects found on SMH surfaces are characterized by their small size which occupies less than 1% of the overall acquired image. Resizing images directly may cause small defects to be almost imperceptible or appear as ordinary steel spots, leading to misidentification of these defects.

- 2) *Data imbalance problem*: The occurrence of certain surface defects, such as imprinted and roll-marks, is relatively low compared to other defect types and defect-free housings, leading to a significant data imbalance. This affects model training and decreases robustness in accurately identifying rare defects.
- 3) *Robust and Accurate Inspection Technique*: The presence of small defects, noise, and variations in the appearance of SMH surfaces poses challenges for defect inspection. This complexity increases the potential for false positives and makes the inspection process difficult. Therefore, a robust and precise inspection technique is necessary to ensure accurate defect recognition.

To address the first issue, we employed an image patching technique to enhance the visibility of defects by slicing acquired SMH surface images into smaller patches. This approach ensures that small defects are not distorted during the resizing process, leading to more accurate recognition of these defects.

To address the data imbalance problem, we proposed a novel approach called the surface defect generation GAN (SDG-GAN). SDG-GAN generates realistic and diverse synthetic defect images, providing a solution to the data imbalance issue. These synthetic images enhance the ability of the classification model to accurately classify defect classes by improving its training dataset and enabling the learning of robust and discriminative features.

Furthermore, to address the third issue, we developed a CNN-based classification model called Inception-CNN. Inception-CNN incorporates convolutional layers and an inception module, allowing it to capture meaningful local patterns, extract complex multi-scale features, and generalize across various surface defect types and sizes. This model effectively addresses challenges caused by changes in surface light reflection and the presence of noise, such as small scratches or stains. The main contributions of the article are as follows:

- 1) An automatic SMH surface defect inspection algorithm has been successfully developed, tested, and deployed for real-time SMHSI system.
- 2) We propose an image patching technique that enhances the performance of the classification model in recognizing small surface defects by resizing images while maintaining the aspect ratio and dividing them into individual patches.
- 3) We introduce a novel SDG-GAN model that effectively addresses data imbalance problems and improves the accuracy and robustness of the classification model by generating realistic and diverse synthetic defect images.
- 4) We propose an Inception-CNN classification model that integrates the inception module between convolutional layers, enabling the recognition of defects of various sizes by extracting complex and abstract multi-scale local features.

5) A comprehensive comparative analysis underscores the efficacy of our proposed image patching technique and the SDGA-GAN approach in significantly enhancing defect recognition performance. Furthermore, our Inception-CNN model demonstrates superior performance in both cross-validation and testing data scenarios when compared to existing state-of-the-art models.

The remaining sections of this paper are organized as follows. Section II provides a comprehensive literature review. Section III describes the system hardware of SMHSI, involving parts such as robotic arm control, imaging methodology, and lighting configurations. In Section IV, we describe our proposed approaches, which include image patching technique, the utilization of GAN for defect generation, and the implementation of SMH surface defect classification model. Section V outlines the experimental setup and presents a detailed analysis of the obtained results. Section VI demonstrated a robustness of the model for various environmental conditions and adversarial attacks. Finally, we draw conclusions and discuss potential future directions in Section VII.

II. LITERATURE REVIEW

In this section, we provide an overview of surface defect inspection methods such as image processing, machine learning, and deep learning-based approaches, as well as image augmentation methods, including traditional and generative approaches.

A. DATA AUGMENTATION METHODS

1) GENERATIVE ADVERSARIAL NETWORKS

Traditional image augmentation techniques are commonly used to address overfitting and enhance dataset diversity. However, for SMH surface defect patterns, some techniques like rotation and cropping may not be suitable as they can alter the defect pattern and increase misclassification. Moreover, traditional augmentation techniques may struggle to represent the wide range of defect types. Hence, alternative methods are needed to expand the dataset with diverse defective samples.

Generative Adversarial Networks (GANs) [26] have gained attention for their ability to produce diverse and realistic defect samples. GANs consist of a generator network (G) and a discriminator network (D) engaged in a two-player minimax game. G aims to generate data samples similar to real ones, while D aims to distinguish real from fake data. Training GANs involves iteratively updating G and D using stochastic gradient descent to generate high-quality samples. Mathematical objective of a GAN can be expressed as shown in Equation (1):

$$\min_G \max_D f(D, G) = \mathbb{E}_{x \sim \mathbb{P}_r} [\log(D(x))] + \mathbb{E}_{z \sim \mathbb{P}(z)} [\log - D(G(z))] \quad (1)$$

where \mathbb{P}_r denotes the real data distribution and $\mathbb{P}(z)$ represents the distribution of the noise vector that sampled from random normal distribution.

However, GANs often suffer from instability of the loss function and convergence challenges.

2) DEEP CONVOLUTIONAL GAN

Deep convolutional GAN (DCGAN) [27] is an extension of the original GAN architecture that introduces key improvements by incorporating convolutional and convolutional-transpose layers in the discriminator and generator, respectively. The inclusion of convolutional layers in the DCGAN discriminator enables the network to effectively leverage the spatial structure present in the data and allowing for better discrimination between real and generated images. The use of convolutional-transpose layers in the generator enables DCGAN to generate more realistic and detailed images. These layers perform an inverse operation to convolution, allowing the generator to gradually upsampling and transform random noise vectors into higher-dimensional representations that resemble real images.

Although the DCGAN has shown promising results with its introduced improvements over the original GAN architecture, it suffers from certain limitations such as mode collapse, vanishing gradients, and instability during the training.

3) WASSERSTEIN GAN WITH GRADIENT PENALTY

To address the training instabilities, Wasserstein GAN (WGAN) [28] introduces a Wasserstein distance metric, providing more stable feedback during generator training. Wasserstein distance quantifies the dissimilarity between the real data distribution \mathbb{P}_r and the generated data distribution \mathbb{P}_g . It is defined as presented in Equation (2):

$$W(\mathbb{P}_r, \mathbb{P}_g) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x \sim \mathbb{P}_r} [f(x)] - \mathbb{E}_{\tilde{x} \sim \mathbb{P}_g} [f(\tilde{x})] \quad (2)$$

here, $\|f\|_L \leq 1$ means that f must be a 1-Lipschitz function.

Wasserstein GAN estimates the Wasserstein distance $W(\mathbb{P}_r, \mathbb{P}_g)$ by maximizing the discriminator (referred as critic) \mathbb{P}_g to approximate the distance between \mathbb{P}_r and \mathbb{P}_g rather than performing classification. The goal is to minimize this distance by training the generator to reconcile \mathbb{P}_r with \mathbb{P}_g . The discriminator assigns a higher value to real samples and lower values to generated samples, while the generator aims to deceive the discriminator by generating samples with higher values. The objective function for WGAN is expressed as depicted in Equation (3):

$$\min_G \max_{D \in \mathcal{D}} f(G, D) = \mathbb{E}_{x \sim \mathbb{P}_r} [D(x)] - \mathbb{E}_{z \sim \mathbb{P}_z} [D(G(z))] \quad (3)$$

where \mathcal{D} represents a set of 1-Lipschitz functions.

Wasserstein GAN with gradient penalty (WGAN-GP) [29] further mitigates instability by adding a gradient penalty term to penalize discriminator gradients with a norm different

from 1. The objective function for WGAN-GP is expressed as shown in equation (4):

$$L_{WGAN-GP(D)} = \mathbb{E}_{\tilde{x} \sim \mathbb{P}_g} [D(\tilde{x})] - \mathbb{E}_{x \sim \mathbb{P}_r} [D(x)] + \lambda \mathbb{E}_{\hat{x} \sim \mathbb{P}_{\hat{x}}} \left[(\|D(\hat{x})\|_2 - 1)^2 \right] \quad (4)$$

here coefficient λ serves as a hyperparameter for the gradient penalty, determining its relative significance in the overall optimization process. Sampling for $\mathbb{P}_{\hat{x}}$ involves uniformly selecting points along straight lines connecting pairs of samples from the \mathbb{P}_r and \mathbb{P}_g

B. DEFECT INSPECTION METHODS

1) IMAGE PROCESSING-BASED METHODS

Image processing approaches in surface defect inspection involve techniques like thresholding [4], [5], [6], image denoising and enhancement [7], [8], [9], to improve visibility and identify surface patterns for defect detection and classification. Improved Otsu's methods [4], [5] and global adaptive percentile thresholding [6] have been utilized for detecting steel surface defects. Morphological processing [7] and bilateral filtering [8] techniques have been employed to reduce noise and preserve edges. A joint-prior-based uneven illumination enhancement (JPUIE) method [9] has been used to eliminate uneven illumination in images for surface defect detection.

Traditional image processing approaches have limitations in capturing relevant features, being sensitive to lighting conditions and complex textures. They also require manual parameter tuning for optimal performance.

2) MACHINE LEARNING-BASED METHODS

Machine learning-based approaches, such as Support Vector Machine (SVM) [10], [11], [12] Random Forests (RF) [13], and K-Nearest Neighbor (KNN) [14], [15] are commonly used in surface defect inspection to classify surface images based on labeled training data.

Pasadas et al. [10] presented an integrated model that utilized a combination of SVM and decision tree algorithms for detecting surface cracks in aluminum plates. Their approach achieved an accuracy of 92.6% in classifying different types of surface cracks. Shanmugamani et al. [11] detected multiple surface defects in gun barrels using texture features extracted from histograms and gray co-occurrence matrices. They tested different classification algorithms and found that SVM outperformed others with an accuracy of 96.67%. Zaghdoudi et al. [12] achieved outstanding classification accuracy using the binary Gabor pattern (BGP) descriptor for hot-rolled strip steel. Their system achieved a remarkable classification accuracy of 99.11% on surface defect classification. Zhao et al. [14] proposed a discriminant manifold regularized local descriptor (DMRLD) algorithm for steel surface defect classification. Their algorithm achieved an overall accuracy of 97.32% on the Kylberg texture dataset and a real steel surface defect dataset, outperforming other local descriptors. Luo et al. [15] introduced selectively dominant

local binary patterns (SDLBPs) for steel surface defect classification. Their method achieved high classification accuracy with an overall accuracy of 94.3% on a steel surface defect dataset.

However, these approaches heavily rely on feature engineering and may have limited generalization capability when encountering unseen or novel defect patterns.

3) DEEP LEARNING-BASED METHODS

CNNs have been the backbone of many deep learning-based surface defect inspection systems. They excel at learning hierarchical features from images and have shown remarkable success in detecting surface defects.

Cui et al. [17] introduced SDDNet, a surface defect detection method addressing large texture variation and small defect size. It utilizes the feature retaining block (FRB) and skip densely connected module (SDCM) to preserve texture information and improve defect prediction. Experimental results on NEU-DET, DAGM, and Magnetic-Tile datasets demonstrate high mean Average Precision (mAP) scores of 88.8%, 99.1% and 93.4% respectively, demonstrating the effectiveness of SDDNet for real-time industrial applications. Wen et al. [18] proposed a novel method for semiconductor wafer surface defect inspection using deep convolutional neural networks. The method involves developing a feature pyramid network with atrous convolution (FPNAC) to extract features and generate feature maps. These feature maps were utilized by a region proposal network (RPN) for generating region proposals, followed by a deep multi-branch neural network (DMBNN) for precise defect classification and segmentation. Liu et al. [23] improved the performance of GoogLeNet [30] by introducing an identity mapping technique for classifying six types of defects on cold-rolled steel strip surfaces, achieving an accuracy of 98.57%. Sun et al. [19] designed an automatic surface flaw inspection system for large-volume metal components, incorporating a novel algorithm based on adaptive multiscale image collection (AMIC) using convolutional neural networks. The algorithm achieved high precision in detecting indentation, scratch, and pitted surface flaws, with respective accuracies of 97.3%, 99.5% and 100%. Yang et al. [20] proposed a surface defect inspection system that incorporates multiple convolutional layers with different kernel sizes. Their model achieved a recognition accuracy of 95.3% on a newly constructed surface defect dataset. QN The Ho et al. [31] explores the problem of self-excited vibration ("Chatter") in mechanical cutting and machining processes. They proposed a method to classify machine stability using combination of using a combination of visual and acoustic data. Data from SS400 steel sheet machining was collected through electron microscopy and sound recording. The two-input model outperformed other models, achieving a 98% higher accuracy, making it a valuable choice for its versatility in handling diverse data types.

It's important to note that none of the individual traditional image processing and machine learning techniques alone can achieve a complete end-to-end solution for surface defect

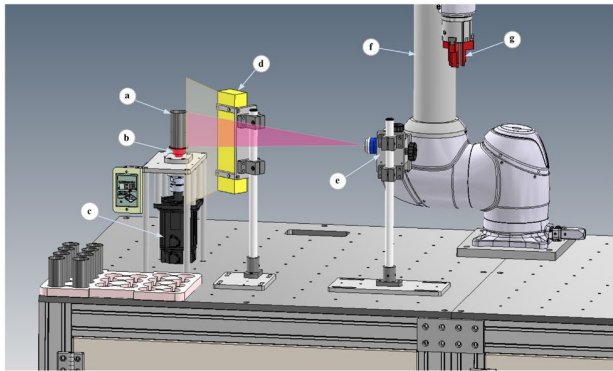


FIGURE 1. 3D demonstration of the SMHSI hardware system for image acquisition and surface inspection. a – servo motor housing product, b – spin zig, c – motion motor unit, d – LED bar-light, e – line-scan camera, f – robot arm, g – gripper.

inspection. Deep learning models provide a more comprehensive and effective approach for surface defect inspection compared to traditional image processing and machine learning techniques. They offer automated feature extraction, higher robustness, and the potential for end-to-end solutions in defect detection and classification tasks.

III. SMHSI SYSTEM HARDWARE

A. IMAGE ACQUISITION AND HARDWARE SYSTEM

The image acquisition process for the SMH products involved the use of a line-scan camera integrated into the hardware system of the servo motor housing surface inspection (SMHSI). Fig. 1 demonstrates the image acquisition process along with a 3D representation of the SMHSI system hardware components.

As shown in Fig. 1, the SMHSI system hardware includes a rotating zig mounted on a motion motor unit, a line-scan camera for image acquisition, a bar-light, and a robotic arm. The rotating zig is employed to spin the product in a clockwise direction for a complete 360° rotation during the image acquisition process. This rotation allows for capturing images of the product from all angles. The rotation speed of the product being inspected varies based on its outer diameter, while the camera focus point moves at a constant speed. The rotation is started when the product is placed on a zig and stopped when the classification model result is received. A robotic arm is equipped with a gripper at the end, which is used to grasp the product and transfer it to and from the rotating zig. When the product starts to spin, the robotic arm moves to a safe area outside of the camera's field of view (FOV). Once the classification result is received, the product is either transferred to a second inspection device for inner diameter inspection if the classification result is defect-free or placed in the corresponding area based on the predicted defect type determined by the classification model. The line-scan camera is set to a fixed range of pixels, regardless of the size of the SMH products. If the products are smaller than the captured range, there will be empty regions in the image,

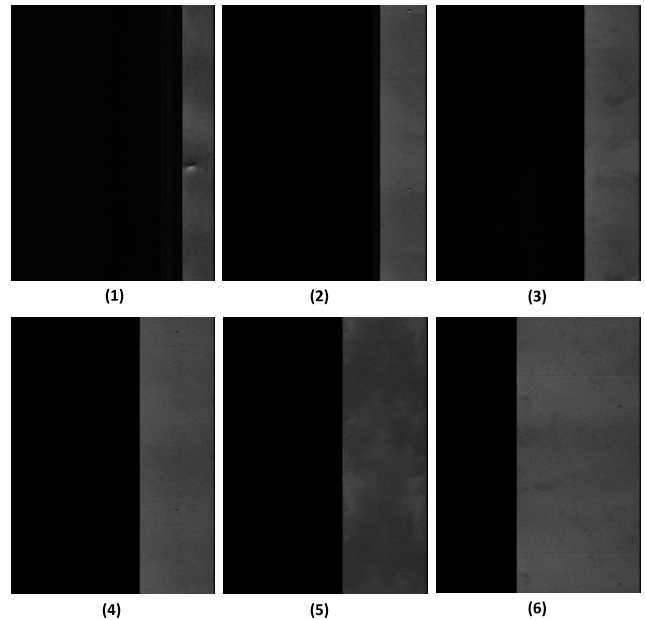


FIGURE 2. Acquired raw image samples of SMHs categorized by their size. The images have a fixed height of 3584 pixels, while their widths are (1) 385 pixels, (2) 570 pixels, (3) 690 pixels, (4) 935 pixels, (5) 1065 pixels, (6) 1575 pixels, referring to the widths of the images after the background removal process.

resulting in black areas. These black areas are considered as the background since they are outside of the actual product region of interest (ROI). After removing the background from the acquired images, they were categorized into six size categories based on their width. The width ranges for these categories from 385 pixels to 1575 pixels, while the height of all images remained constant at 3584 pixels across all categories. Figure 2 shows the acquired raw surface images of the SMH products which are categorized according to their width.

B. CAMERA AND LENS

Due to the rolled shape of the SMH product, a line-scan camera used in the SMHSI system which has a resolution of 5472×3648 (W×H) pixels and can capture images at a frame rate of 5.8 frames per second (fps). It is equipped with a 25mm macro lens that provides a magnification of 0.153X and a FOV of 85×57.2 mm. The use of a line-scan camera in the SMHSI system is preferred over a traditional area-scan camera due to the cylindrical shape of the SMH products. The line-scan camera scans the surface of the housing line-by-line, capturing every pixel in each line, resulting in a higher resolution image with no distortion and allowing to capture the entire surface of the SMH in a single pass as it rotates. This is especially important for defect recognition as even small defects on the surface of the housing can affect the performance and reliability of the motor. Moreover, the high-speed image capturing capability of the line-scan camera makes it ideal for use in real-time surface defect inspection systems.

C. OBTAINING REAL-TIME CLASSIFICATION PREDICTIONS IN THE SMHSI SYSTEM

When deploying the inspection model in the SMHSI system, several factors should be considered to ensure accurate real-time predictions. For instance, it is frequently observed that multiple defects can be present on the surface of a product, and in some situations, these defects may belong to distinct categories. This requires obtaining accurate decision making from the model for a single class. We formulated a dedicated algorithm to address this challenge, outlined in Algorithm 1. The operational procedure of the Algorithm 1 can be summarized as follows:

The initial step involves loading the input image and subsequently slicing it into patches using our custom image patching method. To ensure compatibility with the classification model, a new dimension is added to each image patch and converted it into NumPy array. These modified image patches are stored in a list for further processing. Subsequently, the pre-trained SMH classification model is utilized to predict the probabilities of each patch in the batch of patch images, resulting in a set of class labels and their corresponding probabilities. To obtain a single correct prediction for the entire image, the following steps are performed:

1. First, if, and only if, no defects are found within a batch of predictions, the model prediction is set to defect-free.
2. Otherwise, the defect class indices are separated if any patch indicates the presence of a defect.
3. Then, from these indices, the class index with the highest probability score is identified.
4. Finally, the highest probability defect class label is selected and considered as the predicted class for the entire image.

This approach guarantees accurate decision-making, even in cases where multiple patches contain various types of defects. The final image prediction assigns the class label with the highest probability. Considering the operational procedure outlined in Algorithm 1, the overall decision-making time is 12fps including model prediction and image pre-processing time. This demonstrates the real-time capabilities of the proposed method within the SMHSI system workflow.

IV. PROPOSED METHODS

A. OVERVIEW

In this paper, we propose a CNN-based surface defect inspection method that identifies defects in servo motor housing (SMH) surface images and categorizes them into defect-free, imprinted, pits, roll-marks, and scratches categories. The classification model incorporates convolutional layers and an inception module for local pattern recognition and extraction of complex features. In addition, to improve the identification of small defects, we introduce an image patching technique that reduces image complexity while maintaining image quality and defect visibility. Moreover, we propose a novel approach called surface defect generation GAN (SDG-GAN) to address data imbalance and enhance the accuracy and robustness of the classification model.

Algorithm 1 Pseudo-Code for the Real Time Surface Defect Classification Decision-Making Process in the SMHSI System

label-decode = {0: 'defect-free', 1: 'imprinted', 2: 'pits', 3: 'roll-marks', 4: 'scratches' }

Require: Image path and label-decode dictionary

- 1: Load the image I and slice it into patches P_i where i is the number of patches
- 2: Create an empty list P_{iarr} to store the modified image patches
- 3: **for** each patch i in P_i **do**
- 4: Add a new dimension to the end of the i
- 5: Convert i to NumPy array i_{arr} and store in patches list P_{iarr}
- 6: **end for**
- 7: Obtain prediction probabilities $p(y_i | P_{iarr})$ using pre-trained SMH classification model.

Require: The list of prediction probabilities $p(y_i | P_{iarr})$ obtained from patches P_{iarr} of the single image

- 8: Retrieve the predicted class labels \hat{y}_i and their probabilities y_i from the conditional probability distribution $p(y_i | P_{iarr})$
- 9: **If** $\hat{y}_i == 0$ **then**
- 10: store 0 as predicted class
- 11: **else**
- 12: Separate non-zero (defective) class indices \hat{z}_{ids} from \hat{y}_i
- 13: Identify the class index with highest probability score $\hat{z}_{y_{max}}$ among \hat{z}_{ids}
- 14: Get highest probability class label as $\hat{y}_i[\hat{z}_{y_{max}}]$ and store as predicted class
- 15: **return** predicted class
- 16: **end if**

The high level of overall development procedure of the SMH surface defect inspection is illustrated in Fig 3. The process consists of five stages:

- 1) *Image acquisition (stage-1 in Fig. 3)*: involves acquiring SMH product images using a line scan camera. Detailed information about the image acquisition process and the working mechanism of the SMHSI system hardware in Fig.1, are given in section III of the paper.
- 2) *Data pre-processing (stage-2 in Fig. 3)*: describes a proposed image patching technique to enhance defect visibility, especially for small defects.
- 3) *Data augmentation (stage-3 in Fig. 3)*: involves utilization of proposed SDG-GAN model for data augmentation.
- 4) *Dataset preparation (stage-4 in Fig. 3)*: involves combining generated defect images from the SDG-GAN method with original image patches to create a diverse augmented dataset for training the classification model.
- 5) *SMH surface defect classification (stage-5 in Fig. 3)*: describes the proposed Inception-CNN model for SMH surface defect classification.

Further details about our main proposals, particularly stage-2 (data pre-processing), stage-3 (GAN-based data augmentation), and stage-5 (SMH surface defect classification), are provided in the following sections. The flowchart for the entire study process is provided in Appendix.

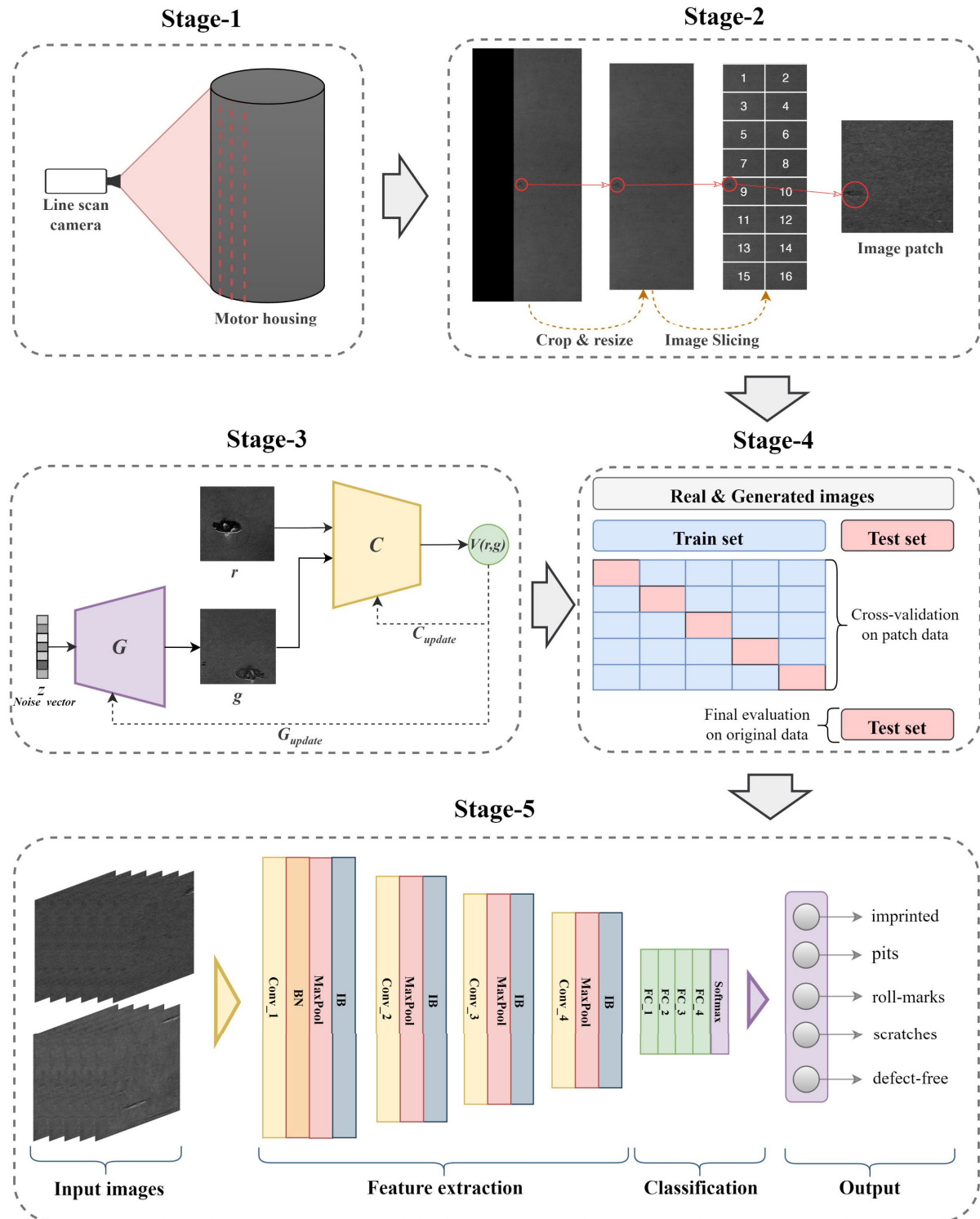


FIGURE 3. A high-level overview of the SMH surface defect inspection development process. Stage-1 – image acquisition, Stage-2 – data pre-processing, Stage-3 – data augmentation, Stage-4 – dataset preparation, Stage-5 – SMH surface defect classification.

B. DATA PRE-PROCESSING

Most of the defects on SMH surfaces are characterized by their small size which occupies less than 1% of the overall

image. Fig. 4 (a) illustrates a sample image of an SMH product with a small defect, while Fig. 4 (b) presents the corresponding percentage of the defect in relation to the entire

image. Moreover, resizing the original image directly to a smaller size can cause the tiny defects to become almost invisible or appear similar to regular steel spots as shown in Fig. 5 (a). This can lead to misidentification of these defects and significantly affects the overall recognition performance of the SMH surface defect inspection model.

To address this problem, special attention was given to maintaining the visibility and accurate representation of these small defects while reducing the image complexity during the pre-processing stage (stage-2 in Fig. 3). The pre-processing step consists of three parts: First, the acquired images are cropped to remove any unnecessary parts that are not relevant to the analysis. Next, the cropped images are resized to the same size while maintaining the original aspect ratio. To calculate the aspect ratio, the height-to-width (H:W) ratio is computed for all product categories. Suppose there are N total product category images based on their size, and let H_i and W_i be the height and width of the i -th image where $i=1,2,\dots,n$.

The average of the image height and width can be calculated as shown in Equation (5) and (6):

$$H_{avg} = \frac{H_1 + H_2 + \dots + H_n}{n} \quad (5)$$

$$W_{avg} = \frac{W_1 + W_2 + \dots + W_n}{n} \quad (6)$$

where H_{avg} indicates average height and W_{avg} indicates average width of all category images.

Then, the average aspect ratio of the height to the width of the images $R_{avg} (H:W)$ can then be calculated as depicted in Equation (7):

$$R_{avg} (H : W) = \frac{H_{avg}}{W_{avg}} \quad (7)$$

During our experimentation process, it was observed that resizing the images to a resolution of 2048×512 (H×W) maintained a consistent aspect ratio of 4:1 across all product categories while effectively preserving the necessary level of image details. This ensures that all images have the same dimensions, which is important for the subsequent image patching process.

To retain the original appearance of defects without sacrificing image quality, the resized images were divided into 16 individual patches. Each patch had dimensions of 256×256 pixels. Dividing the images into patches allowed us to concentrate on specific regions of interest while retaining a reasonable resolution for defect analysis. Fig. 5 illustrates a visual comparison between the visibility of defects in a directly resized image and an image patch which was processed using our image patching method. As shown in Fig. 5 (a), the directly resized original image exhibits a significant distortion in the appearance of the defect that makes it very difficult to recognize accurately by the classification model. Additionally, the defect occupies less than 0.05% of the total surface area emphasizes the small size and relative insignificance of the defect within the image. On the other

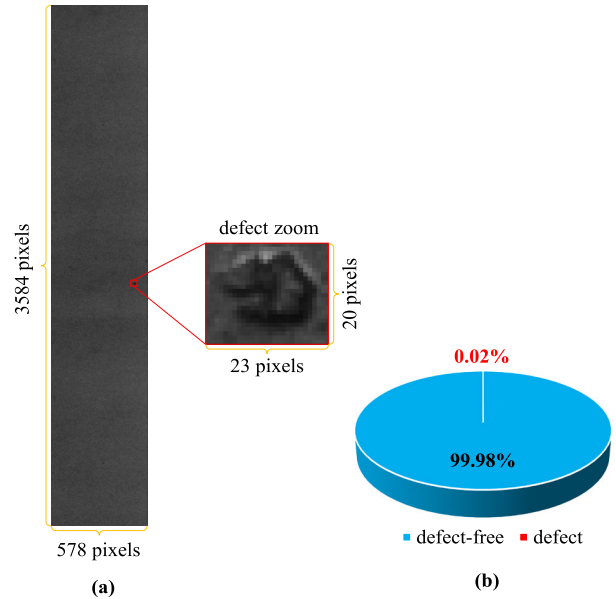


FIGURE 4. Representation of a small-size surface defect (a) in the original image, along with its percentage capacity (b) over the entire image.

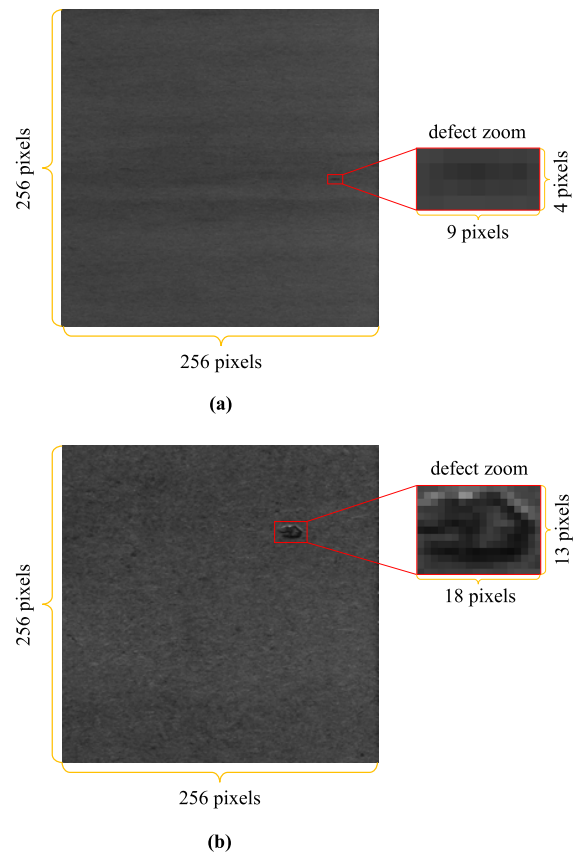


FIGURE 5. Illustration of defect visibility in (a) directly resized image and (b) pre-processed image patch using our image patching method.

hand, it can be seen in Fig. 5 (b) that the image patch which is processed using our image patching method shows no significant change in the shape and visibility of the defect.

Moreover, the proportion of defect size within the image patch has risen from 0.02% to 0.4% when compared to the defect size in the original image.

As a result, utilizing our image patching method can lead to an improved and reliable recognition of small surface defects while minimizing the potential for misidentification.

C. DATA AUGMENTATION

During the process of collecting our dataset, we encountered a data imbalance problem caused by the low occurrence of certain defects in the production line. This imbalance can have implications for the performance of the defect classification model, resulting in biases towards the majority class and reduced accuracy in classifying the minority class defects.

To address this challenge, traditional data augmentation techniques, such as random cropping, rotation and flipping can be employed. However, these traditional data augmentation techniques are not suitable for our specific dataset of SMH surface defects. For example, the shape of defects may change when using cropping techniques, and this can lead to misclassification of a defect with another defect type. Additionally, the traditional data augmentation techniques often fail to generate diverse and novel defective images that accurately represent the unique characteristics of these defects.

To overcome these limitations, we propose a novel approach called surface defect generation GAN (SDG-GAN). SDG-GAN model is an enhanced version of the DCGAN architecture [27] specifically designed for generating realistic and diverse surface defect images of SMHs. We introduce key modifications to both the generator and discriminator architectures to improve their performance and address the challenges associated with our dataset. These modifications aim to improve the model's ability to capture complex patterns, enhance discrimination between real and generated images, and stabilize the training process. Overall, the following modifications were made to the generator and discriminator architectures of the DCGAN:

- We incorporated three additional convolutional layers in the generator network. Two of them are placed before the final upsampling layer. These convolution layers improve to generator ability to capture more complex patterns related to surface defects. This results in a wider range of defect variations and more realistic defect representations in the generated images. The remaining convolutional layer is utilized in the output of the generator network to address the potential artifacts and distortions caused by convolution transpose layers. To maintain image size consistency throughout the convolutional layers, we applied a stride of 1 and zero padding to all convolutional layers. This ensures that the filters traverse the input images without skipping any pixels, preserving data integrity, and maintaining the original size of the generated images.
- In the discriminator network, we replaced the strided convolutions with average pooling operations. This

TABLE 1. Modifications in SDG-GAN compared to DCGAN. * denotes improvements made.

	DCGAN		SDG-GAN	
	<i>G</i>	<i>D</i>	<i>G</i>	<i>C</i>
Network layers	T-Conv	Conv	T-Conv, Conv*, Dense*	Conv Dense*
Batch Norm	Yes	Yes	Yes	No*
Upsampling / Downsampling	T-Conv	Strided Conv	T-Conv	Conv with AvgPool*
Activations (middle)	ReLU	LReLU	LReLU*	LReLU
Activations (output)	Tanh	Sigmoid	Tanh	<i>Linear*</i>
Loss Function	<i>BCE</i>	<i>BCE</i>	WGAN-GP*	WGAN- GP*

T-conv = Transpose convolution, Conv = Convolution, BCE = Binary Cross Entropy,

modification not only reduces the computational complexity of the model but also enables the discriminator to efficiently capture global information from the input images. Meanwhile, the enhanced discriminator becomes more proficient at distinguishing between real and generated images, ultimately leading to improved performance of SDG-GAN model.

- To enhance training stability and improve the quality of generated samples, we adopted the Wasserstein GAN with gradient penalty (WGAN-GP) loss function [29], resulting in a more stable training process.

The proposed SDD-GAN model effectively captures global information and aids more effective discrimination between real and generated samples. The above modifications to our defect generation model leads to the generation of high-quality and diverse synthetic defect images. Table 1 provides the difference of our proposed SDG-GAN with DCGAN model.

1) THE NETWORK STRUCTURE OF SDG-GAN

In our SDG-GAN model, the generator takes 256-dimensional random noise vector as input. The initial step involves mapping the noise vector through a dense layer, resulting in a reshaped output size of $4 \times 4 \times 512$. Afterwards, this reshaped output is passed through a sequence of convolutional layers involving transpose convolution (upsampling), convolution, and zero padding operations. These operations ultimately generate an output image with dimensions of $256 \times 256 \times 1$. To enhance the stability of the training process, batch normalization is applied after the fully connected (FC) layer before reshaping the vectors. The detailed structure of the generator network is represented in Table 2.

In our SDG-GAN model, the discriminator is referred to as the critic since it outputs a continuous real scalar value. The critic network in the SDG-GAN model consists of multiple convolutional layers that process the input image (real and fake), with a dimension of $256 \times 256 \times 1$ pixels. These

TABLE 2. Generator model configuration for SDG-GAN.

Layers	Kernel size	Activation	Output shape
Latent vector	-	-	256
FC			
Batch Normalization	-	LeakyReLU	4×4×512
Reshape			
Transposed convolution	4×4	LeakyReLU	8×8×512
Stride 2			
Transposed convolution	4×4	LeakyReLU	16×16×256
Stride 2			
Transposed convolution	4×4	LeakyReLU	32×32×128
Stride 2			
Transposed convolution	6×6	LeakyReLU	64×64×128
Stride 2			
Transposed convolution	6×6	LeakyReLU	128×128×64
Stride 2			
Convolution	4×4	LeakyReLU	128×128×64
Stride 1			
Convolution	4×4	LeakyReLU	128×128×64
Stride 1			
Transposed convolution	6×6	LeakyReLU	256×256×32
Stride 2			
Convolution	5×5	Tanh	256×256×1

TABLE 3. Critic model configuration for sdg-gan.

Layers	Kernel size	Activation	Output shape
Input image	-	-	256×256×1
Convolution			
Average Pooling	7×7	LeakyReLU	128×128×32
Stride 2			
Convolution			
Average Pooling	5×5	LeakyReLU	64×64×64
Stride 2			
Convolution			
Average Pooling	5×5	LeakyReLU	32×32×128
Stride 2			
Convolution			
Average Pooling	5×5	LeakyReLU	16×16×256
Stride 2			
Convolution			
Average Pooling	3×3	LeakyReLU	8×8×512
Stride 2			
Convolution			
Average Pooling	3×3	LeakyReLU	4×4×1024
Stride 2			
Flatten	-	-	16,384
FC	-	-	1

convolutional layers utilize specific kernel sizes, activations, and pooling strategies to extract significant features from the input image. Additionally, by progressively increasing the number of filters, these layers can capture increasingly complex features within the image. The average pooling layers with a stride of 2 are employed after each convolutional layer to downsample the feature maps and preserve spatial information. This downsampling helps compress information and enlarge the receptive field of subsequent layers, enabling the critic to capture long-range dependencies and spatial

relationships between different regions of the image. After the convolutional layers, the flattening operation is applied to convert the $4 \times 4 \times 1024$ feature map into a flattened representation. Finally, FC layer is used as the last layer to calculate the critic's output, resulting in a single scalar value which is the distance between real and generated images. More details about the architecture of the critic are provided in Table 3.

2) LOSS FUNCTION

In our SDG-GAN network (shown in Fig 3, stage-3), the primary objective of the critic (C) is to estimate the Wasserstein distance between real (r) and generated (g) distributions. It achieves this by assigning higher scalar value (V_r) to real data and lower scalar value (V_g) to g . By doing so, the C aims to maximize the difference between r and g . On the other hand, the generator's (G) objective is to minimize the Wasserstein distance between r and g . This means that the G aims to generate samples that are more realistic and closely resemble the r .

During training, the G produces synthetic samples, and the critic assigns V_g to these generated samples. The scores provided by the C serve as feedback to the G . The G is then updated based on these V_g , adjusting its parameters to improve its ability to generate samples that receive higher V_g from the critic. By minimizing the Wasserstein distance, the G encourages the generated samples to closely resemble real data, leading to higher quality and more realistic generated samples.

The objective of the generator as depicted in Equation (8), which is L_G , is defined as the negative expected value of the output of C for the generated samples:

$$L_G = -\mathbb{E}_{z \sim \mathbb{P}_{N(0,1)}} [C(G(z))] \tag{8}$$

where, \mathbb{E} denotes the expected value of the critic's output $C(G(z))$ for the generated images $G(z)$, where z represents the input noise vector to generator sampled from the normal distribution $\mathbb{P}_{N(0,1)}$.

The critic loss L_C compares real and generated samples and includes a gradient penalty (GP) to maintain gradient stability. The objective function of the critic in our SDG-GAN is expressed as shown in Equation (9):

$$L_C = \mathbb{E}_{\tilde{x} \sim \mathbb{P}_g} [C(\tilde{x})] - \mathbb{E}_{x \sim \mathbb{P}_r} [C(x)] + \lambda \times GP \tag{9}$$

where $\mathbb{E}_{\tilde{x} \sim \mathbb{P}_g} [C(\tilde{x})]$ represents the expected value of the critic's output $C(\tilde{x})$ for generated data samples \tilde{x} drawn from the generated data distribution \mathbb{P}_g . $\mathbb{E}_{x \sim \mathbb{P}_r} [C(x)]$ represents the expected value of the critic's output $C(x)$ for real data samples x drawn from the real data distribution \mathbb{P}_r . GP is the gradient penalty, λ indicates a gradient penalty coefficient which is a hyperparameter that controls the relative importance of the gradient penalty. The gradient penalty GP in the critic loss is defined as indicated in Equation (10):

$$GP = \mathbb{E}_{\hat{x} \sim \mathbb{P}_{\hat{x}}} \left[(\|\nabla_{\hat{x}} C(\hat{x})\|_2 - 1)^2 \right] \tag{10}$$

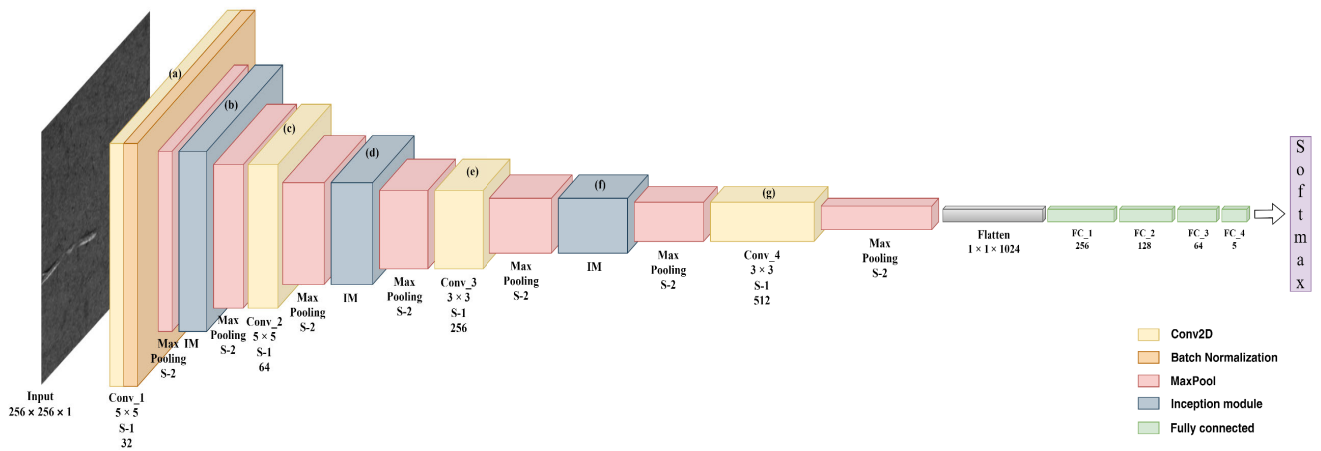


FIGURE 6. SMH surface defect classification network architecture.

where $\nabla_{\hat{x}}$ is the gradient of the critic's output $C(\hat{x})$ with respect to its input which is evaluated at the interpolated points \hat{x} , created by linearly combining real and generated data samples. This means that for multiple pairs of real and generated data points, the critic's gradient at the interpolated points is computed and squared. The squared difference between the gradient's norm and 1 is then averaged over all interpolated points.

D. SURFACE DEFECT CLASSIFICATION

We propose an Inception-CNN for accurately classifying SMH surface images into five distinct categories: defect-free, imprinted, pits, roll-marks, and scratches. The architecture of the proposed network combines convolutional layers and an inception module to capture local patterns and features effectively. The convolutional layers enable the detection of meaningful local patterns, while the inception module enhances feature extraction by incorporating filters of varying sizes and scales to capture more complex and abstract features.

As shown in Fig. 6, the classification network receives a grayscale image of size $256 \times 256 \times 1$ as input. The input image is passed through four convolutional layers to detect essential local patterns and features, followed by three inception modules to capture multi-scale semantic structures. The first convolutional layer (shown in Fig. 6 (a)) consists of 32 filters with a kernel size of 5×5 , which is followed by a batch normalization layer and ReLU activation function. The second convolutional layer (shown in Fig. 6 (c)) uses 128 filters with kernel size of 5×5 . The third and fourth convolutional layers (shown in Fig. 6 (e, g)) contain 256 and 512 filters, respectively, and have the same kernel size of 3×3 . We applied Rectified Linear Unit (ReLU) activation function, zero padding, and L2 regularization with a value of 0.01 for all convolutional layers. Zero padding ensures that the spatial dimensions of the feature maps remain

the same after the convolution operation, while L2 regularization adds a penalty term to the loss function, which discourages large weights and helps to prevent overfitting. Additionally, we used a max-pooling layer with a pool size of 2×2 and stride of 2 after the convolutional layers to reduce the dimensionality of the feature maps. The max-pooling divides the input feature map into non-overlapping rectangular regions and selects the maximum value within each region. The resulting output feature map has a reduced size, which decreases the computational complexity of the model and prevents overfitting.

Fig. 7 illustrates the structure of our inspection module, which is composed of four parallel branches that process the input feature map in different ways. The first branch performs two types of convolutional filters, with kernel sizes of 3×3 and 1×1 , respectively. To introduce local invariance and improve the feature extraction capability of the module, we employ a max-pooling layer to second and fourth branches with a 3×3 kernel and a stride of 1, followed by a 1×1 convolution. The third branch applies a 1×1 convolution followed by a 3×3 convolution. Furthermore, the spatial resolution of the images is maintained in all layers of the inception module by utilizing zero padding and a stride of 1. After the input feature map is processed through the four parallel branches of our inception module, the resulting outputs are concatenated along the channel dimensions to form a single output tensor. We utilized the inception module at different locations within our network, each with a distinct filter size configuration. Specifically, the filter sizes of (32, 16, 16, 32, 16, 16), (64, 64, 32, 96, 32, 64), and (128, 96, 96, 96, 64, 128) are used for the inception module illustrated in Fig 7 (b, d, f), respectively. To further reduce the spatial size of the output feature map and decrease the computational complexity of the subsequent layers, we apply max pooling with a kernel size of 2×2 and a stride of 2 after each inception module.

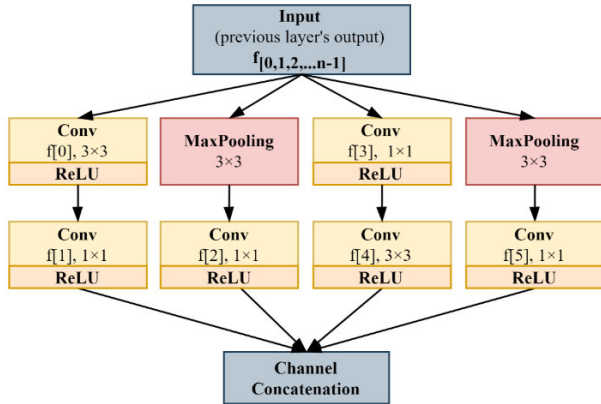


FIGURE 7. Proposed inception module structure. proposed inception module structure.

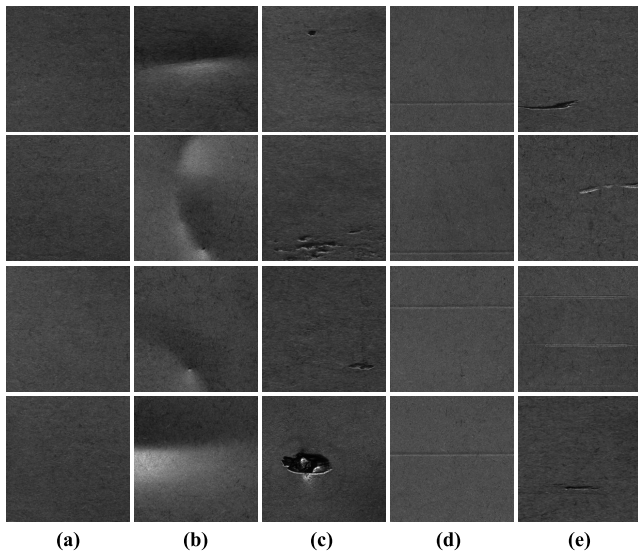


FIGURE 8. A visual representation of the patch images in our dataset, including (a) defect-free and (b) imprinted, (c) pits, (d) roll-marks, (e) scratches defect categories.

The outputs from the max pooling layers are then fed into the next convolution layers of Fig 6 (c, e, g), respectively. Afterwards, the extracted features are processed by four fully connected (FC) layers for classification. These FC layers contain 256, 128, 64, and 5 units, respectively, utilizing the ReLU activation function. The final FC layer acts as the output layer and is connected to the SoftMax activation function, which produces a probability distribution over the five classes of SMH surface images.

V. EXPERIMENTS AND RESULT ANALYSIS

A. DATASET PREPARATION

The SMHSI system (Fig. 1) was utilized to collect the SMH surface image dataset. As shown in Fig. 8 the training dataset consists of five different classes: defect-free, imprinted, pits, roll-marks, and scratches.

TABLE 4. Dataset information.

Class categories	Original images	Total images	Training set	Validation set
SMHSD-O-TA				
defect-free	200	440	320	80
imprinted	34	417	320	80
pits	173	436	320	80
roll-marks	37	419	320	80
scratches	35	418	320	80
Total	479	2,130	1600	400
SMHSD-P-TA				
defect-free	2,560	2,560	2,048	512
imprinted	49	250	200	50
pits	240	250	200	50
roll-marks	61	250	200	50
scratches	48	250	200	50
Total	2,958	3,560	2,848	712
SMHSD-P-GAN				
defect-free	2,560	2,560	2,048	512
imprinted	49	1,000	800	200
pits	240	1,000	800	200
roll-marks	61	1,000	800	200
scratches	48	1,000	800	200
Total	2,958	6,560	5,248	1,312

Following three versions of the SMH dataset were created for experimental analysis: SMH surface dataset of original images with traditional data augmentation (SMHSD-O-TA), SMH dataset with image patches and traditional data augmentation (SMHSD-P-TA), and SMH dataset with image patches and GAN-based data augmentation (SMHSD-P-GAN). Each version includes training and validation sets. The test set used for evaluating the model consists of original SMH images. Detailed information for each dataset is provided in Table 4.

The SMHSD-O-TA prepared by cropping the original acquired images and resizing them to the same size of 1024 × 512 (H×W) pixels. The dataset contains a total number of 2,130 images, including 479 original images and 1651 images augmented using traditional data augmentation techniques such as random horizontal and vertical flipping and randomly changing the image brightness and contrast. The training set consists of 1600 images (80%), and the validation set consists of 400 images (20%). The SMHSD-P-TA dataset contains a total number of 3,560 images with a size of 256 × 256 pixels for each, including 2,958 patch images prepared using image patching method and 602 number of images were augmented by applying traditional data augmentation techniques. The training set consists of 2,848 images (80%), and the validation set involves 712 images (20%). The SMHSD-P-GAN dataset contains a total of 6,560 image patches, and 3,602 of them were generated using our SDG-GAN model. The training set consists of 5,248 images (80%), and the validation set consists of 1,312 images (20%). A test set separated from the SMHSD-O-TA dataset is used to evaluate the performance of the classification model on all datasets. It comprises

TABLE 5. Number of images in test set.

Test set					
defect-free	imprinted	pits	roll-marks	scratches	Total
40	17	36	19	18	130

130 images, with 40 images of defect-free, 17 images of imprinted, 36 images of pits, 19 images of roll-marks, and 18 images of scratches classes. The number of images in the test set is given in Table 5.

B. EVALUATION METRICS

1) EVALUATION METRICS FOR SDG-GAN MODEL

Evaluating the performance of generation models is extremely challenging. Traditional methods for evaluating image quality involve human or measurement-based evaluations. However, relying on human evaluation is insufficient for accurately determining the diversity of generated image datasets, limiting its ability to provide objective quality evaluations. Hence, we employed measurement-based evaluations such as Fréchet Inception Distance (FID) [32] that proposed recently and proved its effectiveness for evaluating GAN results. The FID metric utilizes the Inception model to measure the difference between distributions of real and generated images, offering valuable insights into image quality, diversity, and similarity. Specifically, the FID compares the distributions of Inception embeddings, which are activations obtained from the penultimate layer of the Inception network, for real (r) and generated (g) images.

These distributions are modeled as multi-dimensional Gaussians characterized by their mean and covariance and defined as:

$$FID(r, g) = \|\mu_r \mu_g\|^2 + Tr(\sigma_r + \sigma_g - 2(\sigma_r \sigma_g)^{\frac{1}{2}}) \quad (11)$$

where μ_r and μ_g denote the mean, σ_r and σ_g denote the covariance of the real and generated embeddings, respectively. Tr is the trace of the matrix.

2) EVALUATION METRICS FOR CLASSIFICATION MODEL

Classification model evaluation metrics are essential for evaluating the performance of models that aim to classify data into distinct categories. The commonly used evaluation parameters are accuracy, F1 score, precision, and recall. True Positive (TP), False Negative (FN), False Positive (FP), and True Negative (TN) values are used to calculate these metrics.

Accuracy measures the percentage of correctly classified data samples out of the total number of data samples, as defined in Equation (12):

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \quad (12)$$

However, accuracy alone may not be suitable for imbalanced datasets where one class has significantly more data samples than the others. Precision measures the proportion of correctly identified TP samples among all predicted positives samples as defined in Equation (13):

$$Precision = \frac{TP}{TP + FP} \quad (13)$$

Recall, as outlined in Equation (12), measures the proportion of TP samples correctly identified by the model among all actual data samples:

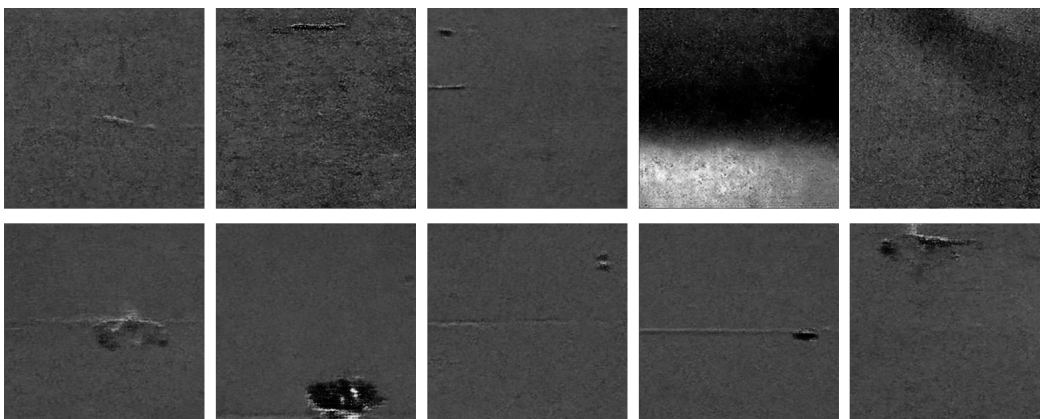
$$Recall = \frac{TP}{TP + FN} \quad (14)$$

The F1-score combines precision and recall into a single metric, taking into account both aspects. The mathematical calculation of F1-score is depicted in Equation (15) as below:

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (15)$$

C. DEFECT IMAGE GENERATION RESULTS

In this section, we present the performance evaluation of our proposed SDG-GAN model for the task of SMH surface defect image generation. To comprehensively assess the effectiveness of our model, we conducted both qualitative and quantitative analyses, comparing its performance against the baseline DCGAN model.

**FIGURE 9.** Generated defect images using a comprehensive dataset includes all classes.

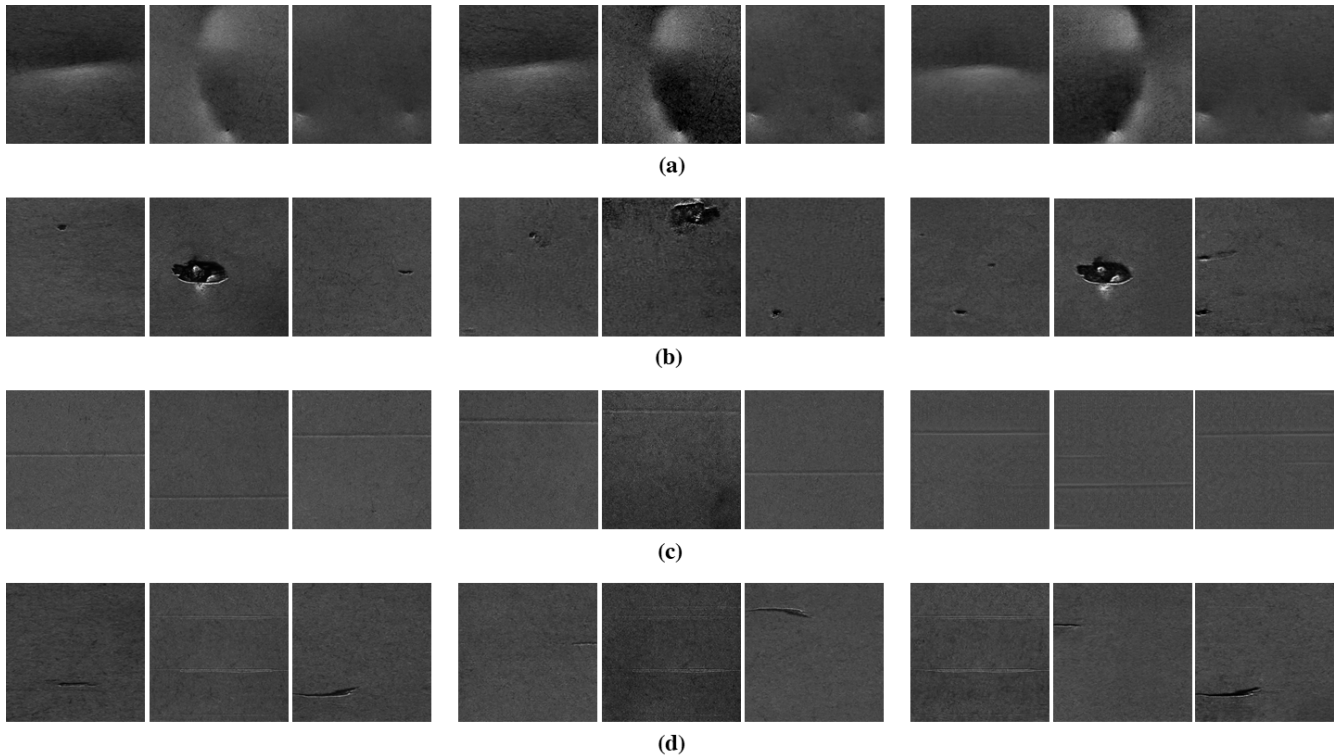


FIGURE 10. Qualitative comparison of generated SMH surface defect images including real defect images (left column), generated defect images by DCGAN (middle column), and generated defect images using SDG-GAN (right column). Each row in the figure represents a different defect class, including (a) imprinted defects, (b) pits defects, (c) roll-marks defects, and (d) scratches defects.

We conducted training experiments by using images from all defect classes simultaneously. However, we encountered challenges associated with this approach. One major issue was the interference between different defect types. The model struggled to capture the distinct characteristics and variations of each defect class, leading to decreased performance in discriminating between different defects classes. Fig. 9 provides a visual representation of defect samples generated through simultaneous training with all classes. To ensure comprehensive coverage of different defect types, we conducted separate training for both the SDG-GAN and baseline models on each defect class using the SMHSD-P-TA dataset. The training dataset comprised augmented real images, with 10,000 images for each defect class. Training on separate defect datasets allows the model to focus on learning the specific characteristics and variations of each defect type and can effectively capture these specific features and generate more accurate and realistic defect images. The models were implemented using TensorFlow framework (version 2.9.0) and Python (version 3.8.16). The software and hardware configuration employed for SDG-GAN model training corresponds to the details outlined in Table 7.

The training hyperparameters used in the experiments were as follows: a batch size of 64 images, 150 training iterations, Adam optimizer with a learning rate of $2e-4$, and parameters $\beta_1 = 0.5$ and $\beta_2 = 0.9$ for both generator and discriminator.

The input image size for the discriminator was $256 \times 256 \times 1$, and the generator receives a 256-dimensional random noise vector as input. The gradient penalty coefficient was set to $\lambda = 10$ and number of iterations of the discriminator was set to 3 which means the discriminator trains 3 times more than the generator in each iteration.

Fig. 10 presents a side-by-side comparison of real defect samples alongside randomly selected defect images generated by the DCGAN and SDG-GAN models, allowing for a qualitative evaluation of the generated defect images. It can be seen from Fig. 10 the defective images generated by the SDG-GAN model exhibit better quality compared to those generated by the DCGAN model. The defect images generated by the DC-GAN model have periodic noise and incomplete representation of defect characteristics, particularly in the pits class. These generated images also have fuzzy defect regions, resulting in poor quality defect images. The SDG-GAN model successfully captures the complex characteristics and features present in real defect images, resulting in more visually appealing and realistic synthetic defect samples.

In addition to the qualitative assessment, we conducted a quantitative evaluation using the Fréchet Inception Distance (FID) metric to compare the quality of the generated images. The FID metric measures the similarity between the distributions of real and generated defect images. A lower FID score

TABLE 6. Fid scores of images generated by DCGAN and SDG-GAN models.

	imprinted	pits	roll-marks	scratches
DCGAN	37.61	35.11	12.5	26.1
SDG-GAN	28.89	16.22	11.65	9.28

indicates a closer resemblance between the two distributions. Table 6 provides a comparison of the FID scores achieved by the DCGAN and SDG-GAN models. Results indicate that our model achieves significantly lower FID scores for all defect classes compared to the DCGAN model, highlighting the effectiveness of our SDG-GAN model in generating images with improved quality and diversity.

Based on the evaluation results of generated SMH surface defect images, the following conclusions can be drawn:

- Incorporating additional convolutional layers in the generator network improved the capture of complex defect patterns and enhanced the quality and diversity of generated images.
- Adding a convolutional layer in the output of the generator network increased the quality of the generated defect images by effectively mitigating artifacts, distortions, and fuzzy defect regions.
- Replacing strided convolutions with average pooling in the critic reduced network complexity and enhanced its ability to capture global information from input images.
- Adopting the WGAN-GP loss function improved training stability and led to higher-quality defect images by enforcing constraints on the critic.

D. CLASSIFICATION RESULTS

1) SELECTION OF THE TRAINING DATASET

One of the primary considerations before making improvements to the model architecture is the preparation of a reliable dataset that effectively represents the target problem domain. To achieve this, we conducted a series of experiments using SMHSD-O-TA, SMHSD-P-TA, and SMHSD-P-GAN datasets. We utilized the custom CNN model as baseline model to train each dataset individually, and a single test set was utilized to evaluate the model's performance across all datasets. The baseline CNN model consists of 7 convolutional layers followed by max pooling layers, and FC parameters are same as our proposed Inception-CNN model.

The information presented in Table 7 provides an overview of the environmental setup used for model training.

During the training, we adjusted the training hyperparameters such as image and batch sizes according to the characteristics of each dataset. For the patch datasets of SMHSD-P-TA and SMHSD-P-GAN, we kept the original patch size of 256×256 and the batch size of 128. We resized the images to 512×256 (H×W) and set a smaller batch size of 32 for original images in the SMHSD-O-TA dataset. The number of training epochs for all datasets was set to 100. To optimize the gradients of the baseline CNN model, ADAM optimizer with a learning rate of $1e-4$ was used, and

TABLE 7. Software and hardware components of the experimental environment.

Name	Parameter
Memory	64GB
CPU	Intel Corei7-9700 3.00 GHz (8 core)
GPU	NVIDIA GeForce RTX 3090 (24GB)
Operating system	Windows 11 Pro
Environment configuration	TensorFlow-gpu 2.10.0, Python3.8.15, cuda11.2, cudnn8.1

TABLE 8. Evaluation results obtained from the baseline cnn model for the SMHSD-O-TA, SMHSD-P-TA, and SMHSD-P-GAN datasets.

Class type	Metrics			Overall accuracy (%)
	Precision	Recall	F1-score	
SMHSD-O-TA+CNN				
defect-free	0.8611	0.7750	0.8158	66.92
imprinted	0.9231	0.7059	0.8000	
pits	0.5000	0.8056	0.6170	
roll-marks	0.9091	0.5263	0.6667	
scratches	0.4167	0.2778	0.3333	
SMHSD-P-TA+CNN				
defect-free	1.0000	0.9750	0.9873	89.23
imprinted	1.0000	1.0000	1.0000	
pits	0.7347	1.0000	0.8471	
roll-marks	0.9474	0.9474	0.9474	
scratches	1.0000	0.3333	0.5000	
SMHSD-P-GAN+CNN				
defect-free	0.9756	1.0000	0.9877	95.38
imprinted	0.9444	1.0000	0.9714	
pits	0.9211	0.9722	0.9459	
roll-marks	1.0000	0.9474	0.9730	
scratches	0.9333	0.7778	0.8485	

to calculate the loss during training, we utilized the sparse categorical cross-entropy loss function.

Table 8 provides a comparison of the dataset results obtained using the baseline CNN model. The metrics considered for evaluation are precision, recall, F1 score, and overall accuracy. The results indicated that the baseline CNN model performed very poor generalization to the problem on the SMHSD-O-TA dataset. This is indicated by the relatively lower overall accuracy of 66.92% compared to the other datasets. On the other hand, even though the model struggled to classify the pits and scratches classes on the SMHSD-P-TA dataset, the overall model generalized well to recognize SMH surface defects with overall accuracy of 89.23%. In addition, the baseline model achieved precision, recall and F1-score results over 90% for defect-free, imprinted and roll-marks classes in this dataset. In contrast the evaluation results for SMHSD-P-GAN dataset showed an overall accuracy of 95.38% which is 6.15% improvement compared to SMHSD-P-TA dataset. The model achieved improvements for the pits class, with 18.64% and 9.88% improvements in precision and F1-score, respectively. Addi-

tionally, there was a significant improvement of 44.45% in recall and a 34.85% improvement in the F1-score for the scratches class.

Through a detailed comparative analysis of the dataset results, we can draw the following conclusions:

- One of the main reasons for the model suffering to generalize the problem on SMHSD-O-TA dataset is due to the lack of defect variations and small defects that only occupy less than 1% of the entire image surface. There is a possibility that when resizing the images to a compatible size for training, small defects present throughout the entire image surface might be missed during the convolution operations. Therefore, the classification model may struggle to capture these small-scale defects, resulting in lower performance and potential misclassifications.
- The utilization of our image patching method in the SMHSD-P-TA dataset allowed the model to learn and classify local regions of the images, leading to the 22.31% of improved overall performance and better recognition of small defects compared to the SMHSD-O-TA dataset. However, the baseline CNN model encountered difficulties in classifying defects in the pits and scratches classes, which can be attributed to the lack of defect variations in the dataset.
- The significant improvement of the baseline classification model performance on SMHSD-P-GAN dataset can be attributed to the increasing diversity of SMH surface defective images within the dataset. The results indicate the effectiveness of our SDG-GAN approach resulted in the generating of more diverse and realistic images, thereby contributing to the overall enhancement of the classification performance. However, we still need to improve the model architecture to achieve better identification on scratches class and further improve overall accuracy of the model.

2) MODEL SELECTION AND COMPARATIVE ANALYSIS

After selecting the training dataset, our objective was to enhance the classification model's performance. To achieve

this, we integrated the inception module into the baseline CNN architecture and conducted experiments on SMHSD-P-GAN datasets. By applying the inception module at different locations within the baseline CNN model, we aimed to evaluate its effectiveness and identify the optimal configuration for achieving improved classification accuracy and feature extraction. Additionally, we conducted a comparative analysis by fine-tuning well-known classification models, such as InceptionV3 [33], MobileNetV2 [34], EfficientNetV2S [35], ResNet50 [36], and CNN with SCA [37], and evaluated their performance on the test data.

To validate the robustness and generalization capabilities of the models, we employed a 10-fold cross-validation approach, which evaluates the model's consistency and performance across diverse data partitions. During the cross-validation process, all models were configured with consistent hyperparameters, including 50 epochs, a batch size of 128, and the utilization of the Adam optimizer with a learning rate of $1e-4$. The cross-validation process is visually represented in the flow diagram provided in Appendix. It began with the partitioning of the dataset into ten separate folds, ensuring an even distribution of data in each fold. In each iteration of the cross-validation procedure, the model was trained on nine of these folds, representing the training set, while the remaining fold was employed for validation. This tenfold cross-validation was repeated ten times, with each fold taking on the role of the validation set once. The cross-validation results are detailed in Table 9.

It can be seen from table 9 that EfficientNetV2S, ResNet50, and CNN with SCA exhibited impressive accuracies with achieving accuracies above 99%. Notably, our Inception-CNN model outperformed all others, achieving the highest accuracy score of 99.40%. Moreover, the robust consistency in model performance is clearly observable, with the standard deviation (std) of test accuracy consistently maintaining a negligible value of ± 0.0 for all models across all folds during the cross-validation process.

TABLE 9. Comparative analysis of popular CNN models using 10-fold cross-validation on the SMHSD-P-GAN dataset.

Test set	Accuracy (%)						
	EfficientNetV2S	InceptionV3	MobileNetV2	ResNet50	CNN with SCA	Baseline CNN	Ours
Fold-1	99.24	89.94	98.32	98.93	99.09	98.93	99.24
Fold-2	98.93	90.24	98.17	99.54	98.93	98.93	98.93
Fold-3	99.54	90.40	96.80	99.39	98.93	99.09	99.41
Fold-4	99.70	88.41	98.17	99.54	99.39	97.26	99.79
Fold-5	98.30	91.31	97.71	99.39	99.09	98.63	98.93
Fold-6	99.39	89.79	98.32	99.54	99.39	98.63	99.54
Fold-7	99.39	91.16	96.80	98.78	98.78	99.09	99.59
Fold-8	98.60	91.46	97.87	99.39	98.93	98.93	99.78
Fold-9	99.39	92.84	96.95	99.24	98.78	97.10	99.24
Fold-10	99.24	87.80	98.17	99.24	98.78	98.48	99.54
Average accuracy (%)	99.17	90.34	97.73	99.30	99.01	98.51	99.40

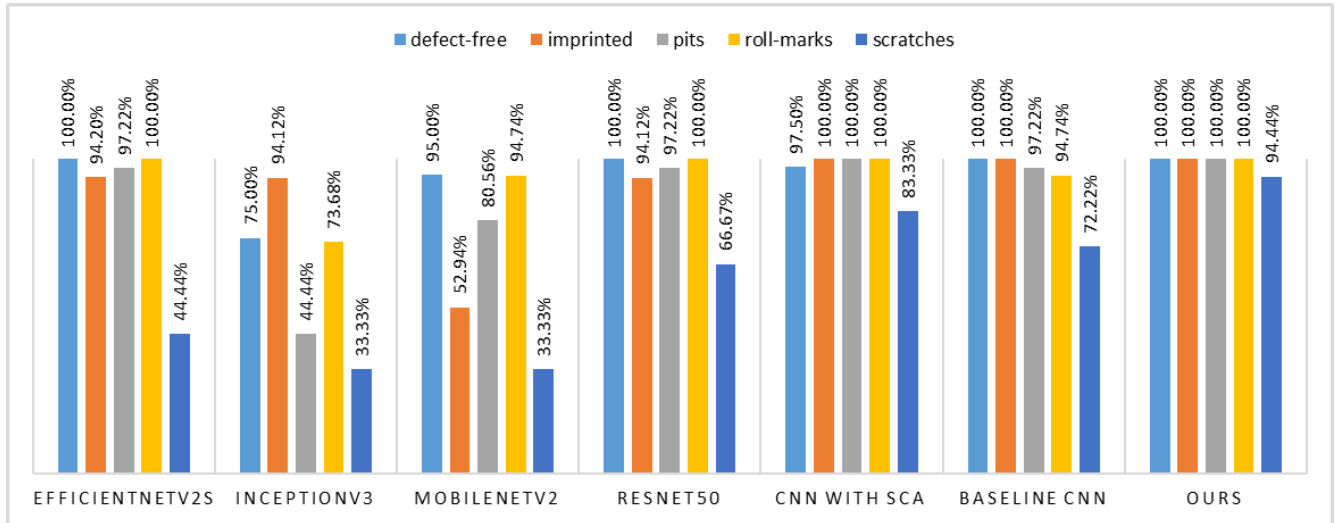


FIGURE 11. Class-wise accuracy comparison of the classification models on test data.

TABLE 10. Overall accuracy comparison of the classification models on test data.

Models	Test accuracy (%)	Model Params (M)	Layers
EfficientNetV2S	90.77	20.70	520
InceptionV3	63.08	22.37	318
MobileNetV2	76.92	2.63	161
ResNet50	93.85	24.15	182
CNN with SCA	96.92	0.52	30
Baseline CNN	94.62	5.12	24
Ours	99.23	3.58	48

We extended our evaluation to include the performance of the models on distinct test data derived from the original dataset (as outlined in Table 5). In Table 10, we present a comprehensive comparative analysis including overall test accuracy, model parameters, and the number of layers. It can be seen from the results in Table 10 that InceptionV3 achieved an accuracy of 63.08%, while MobileNetV2 attained 76.92%. These results suggest that both models exhibited signs of overfitting and faced challenges in generalizing to test data. Conversely, a majority of the models consistently achieved strong performances, with accuracies exceeding 90%. Notably, our proposed model demonstrated exceptional robustness across various defect categories, surpassing other models with an impressive overall accuracy of 99.23%.

Furthermore, we provide a detailed breakdown of class-wise accuracies in Fig. 11 to clarify how each model performs on individual defect categories within the test data. The overall comparative analysis leads us to the following conclusions:

- While InceptionV3, MobileNetV2 models faced generalization issues, the majority performed reliably

performance. However, a common struggle among these models was accurately identifying defects in the scratches class.

- Notably, our proposed model outperformed other models on both datasets, underscoring its remarkable generalizability and robustness across various types and sizes of SMH surface defect categories.

VI. ROBUSTNESS EXPERIMENT

We conducted robustness experiment to validate the robustness of our model for various potential environmental conditions and adversarial attacks that may occur during the real-time inspection. These disruptions include noise from dust and even minor shifts in camera focus can result in image blurring. Additionally, fluctuations in brightness attributed to electrical anomalies, and the presence of natural-looking scratches on the product surface. Each of these factors has the potential to impact the performance of the model. To simulate these challenges, we intentionally introduced distortions by overlaying patch data with various effects, including blurring, noise, extreme lighting conditions, and imitating natural surface scratches. Fig. 12 provides visual examples of these simulated images.

Quantitative results are presented in Fig. 13. The results from Fig. 13 indicate that the proposed model exhibits robustness against most types of environmental conditions and adversarial attacks. However, there is a slight sensitivity observed in the presence of extreme brightness variations. This can be attributed to the pits and scratches defect categories, as these defect types are sensitive to various lighting conditions, as brightness increases, these defects tend to reflect higher illumination.

Based on these results, it can be concluded that even though the proposed model shows some sensitivity for extreme brightness conditions, it overall demonstrates a high level of

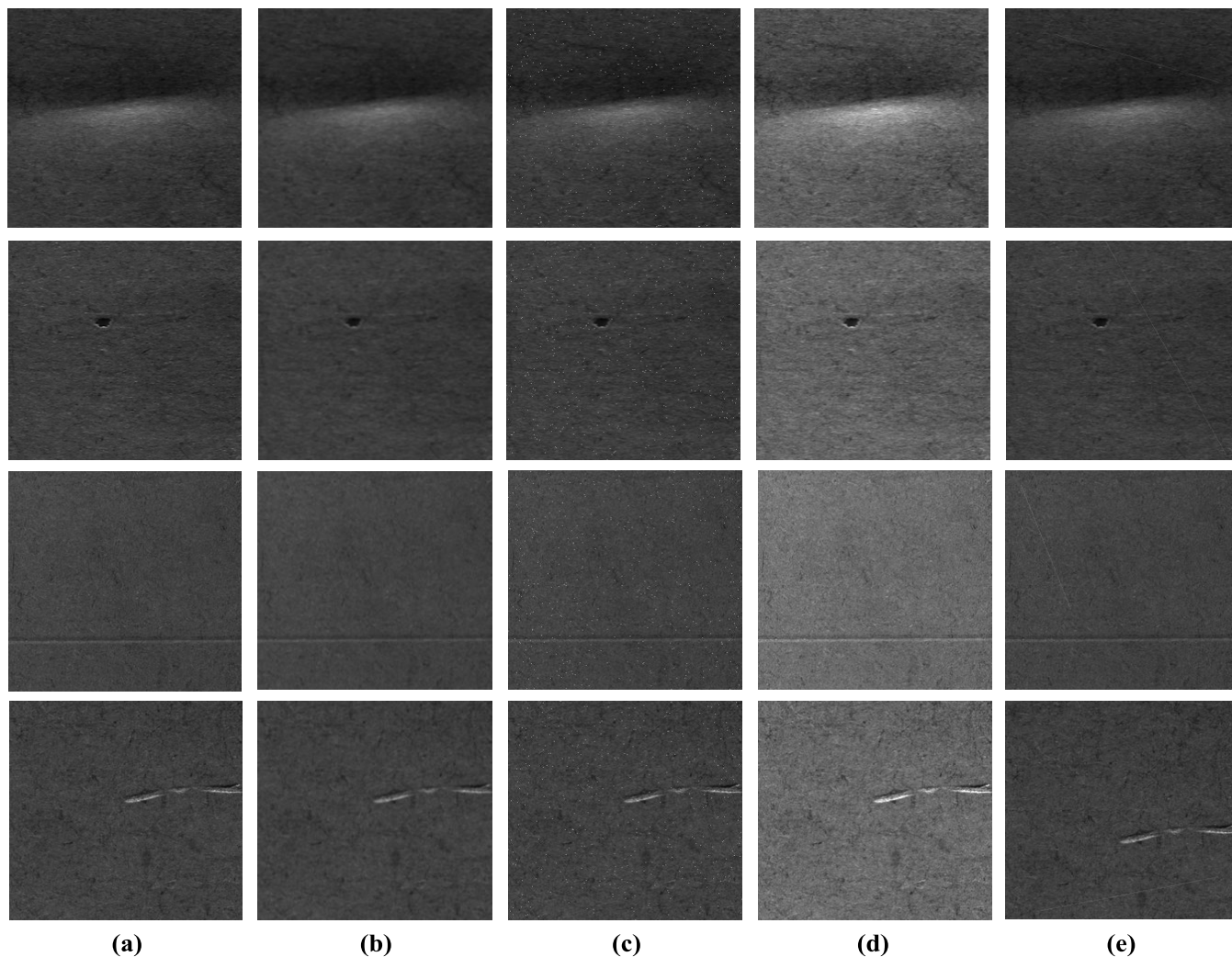


FIGURE 12. Simulated images for robustness experiment including: (a) initial images, (b) blurred, (c) noise (d) extreme light and (e) natural-surface scratches images.

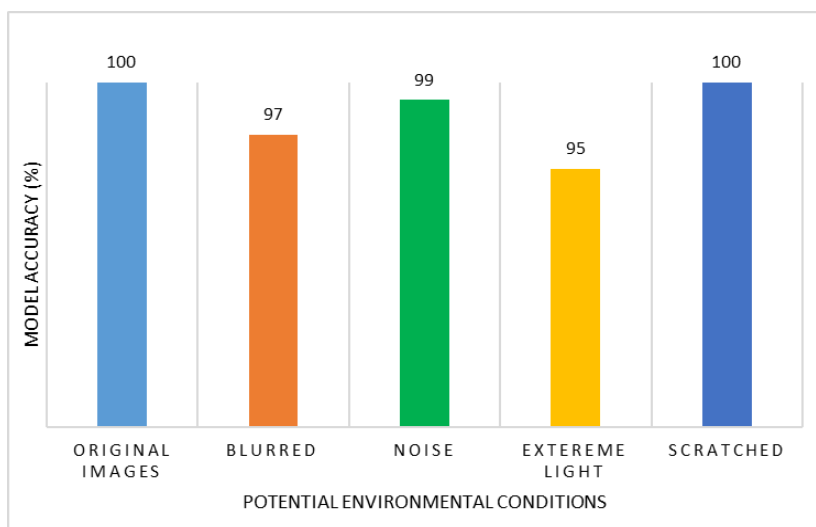


FIGURE 13. Overall performance of the proposed model for robustness experiment.

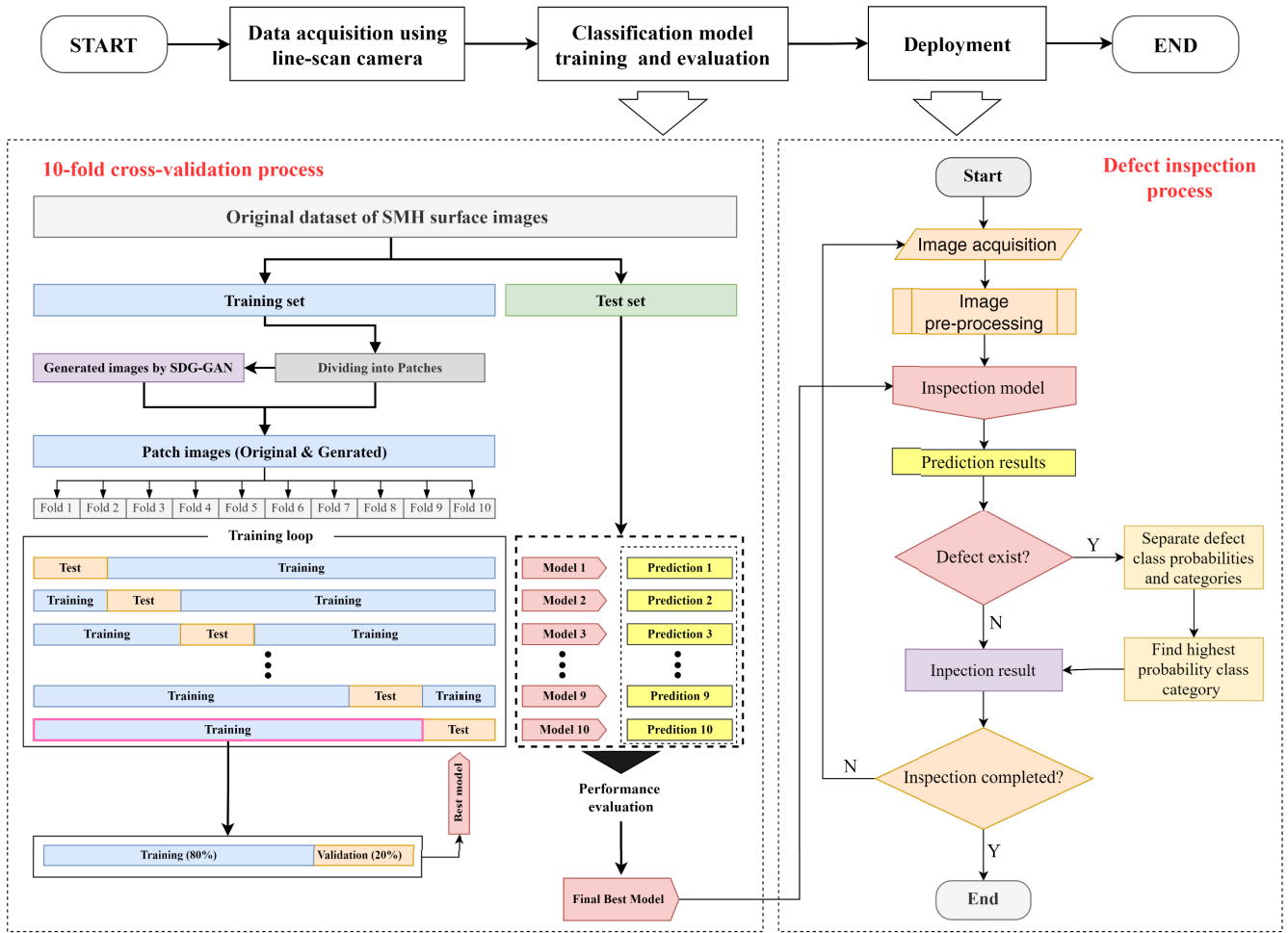


FIGURE 14. Flow diagram of the overall study process.

robustness against various potential environmental conditions adversarial attacks.

VII. CONCLUSION AND FUTURE WORKS

In this article, we propose an efficient inspection approach for defect recognition in SMH surface images. Our primary proposals include introduction of Inception-CNN for classification, introducing image patching technique, and implementing SDG-GAN for defect generation.

The experimental procedure proceeded as follows: Initially, to address the data imbalance problem, we implemented SDG-GAN, that effectively enhances the dataset representation by generating diverse and realistic defective images. Afterward, we trained a baseline CNN model using the SMHSD-O-TA, SMHSD-P-TA, and SMHSD-P-GAN datasets. This step was taken to assess the impact of image patching and the SDGAN approaches, with the objective of selecting the most suitable dataset that can represent the diverse range of surface defects found in SMHs. Subsequently, we conducted an extensive comparative analysis by training popular classification models, including InceptionV3 [33], MobileNetV2 [34], EfficientNetV2S [35], ResNet50

[36], and SCA-CNN [37], alongside the baseline CNN and our proposed Inception-CNN model. These models were trained on the SMHSD-P-GAN dataset and evaluated using cross-validation. We then validated their performance using the original test data, to confirm their generalization capabilities.

Both qualitative (as shown in Fig.10) and quantitative (as shown in Table 6) evaluations of the generated images by the DCGAN and SDG-GAN models highlight the superiority of our SDG-GAN model. It achieves the lowest FID score among all classes, which can be attributed to the specific modifications incorporated into the generator and discriminator networks.

The evaluation results of data selection, as illustrated in Table 8, demonstrated a 22.31% enhancement for the baseline CNN model trained on the SMHSD-P-TA dataset and a remarkable 28.46% improvement for the model trained on the SMHSD-P-GAN dataset, both when compared to the model's performance on the SMHSD-O-TA dataset. These results strongly indicate the effectiveness of the image patching and SDG-GAN approaches in improving the model's recognition accuracy for small defects and enhancing its

robustness for different types and sizes of surface defects in SMHs.

The incorporation of the inception module into our classification network yielded remarkable results, with an average accuracy of 99.40% during cross-validation (as shown in Tabel 9) and an overall accuracy of 99.23% on the test data (as shown in Tabel 9). This represents a significant 3.85% and 32.31% increase in model accuracy compared to the baseline CNN when trained on the SMHSD-P-GAN and the SMHSD-O-TA datasets, respectively. Additionally comparative analysis indicates that our proposed Inception-CNN model demonstrated superior performance compared to state-of-the-art models. These results highlight the efficacy of the inception model in enhancing feature extraction and improving the model's ability to recognize and classify surface defects, especially for the scratches class.

The robustness experiment results (in Fig. 10) show that our model exhibits commendable robustness against a variety of potential environmental conditions and enemy attacks. Although it shows sensitivity to extreme brightness conditions, the overall performance exhibits a high degree of robustness, making it a promising candidate for real-world applications in defect detection and industrial quality control.

Based on the operational procedure of Algorithm 1, our proposed inspection method achieves a real-time decision-making processing speed of 12fps, satisfying the real-time inspection requirements of the SMHSI system. This speed is well-suited for our specific case, considering the line scan camera's frame rate of 5.8fps. However, in scenarios where an even higher processing speed is needed, it is necessary to explore ways to further increase the speed of our inspection algorithm.

In the future, our objective is to continually enhance the performance of our surface defect classification model, particularly in diverse brightness conditions. This requires the collection of a larger dataset of surface defect images obtained from real SMH products using our SMHSI system under different conditions. Moreover, we aim to optimize the inference time of the inspection process without compromising the accuracy of the model.

APPENDIX

FLOW DIAGRAM OF THE OVERALL STUDY PROCESS

See Figure 14.

REFERENCES

- [1] SolisPLC. *Servo Motor—Working Principle, Types, and Applications*. Accessed: Jul. 4, 2023. [Online]. Available: <https://www.solisplc.com/servo-motor>
- [2] SolisPLC. *Servo Drive Manufacturers in Industrial Automation*. Accessed: Jul. 4, 2023. [Online]. Available: <https://www.solisplc.com/servo-motor>
- [3] M. Dreisilker. *Common Causes of Servo Motor Failures and Preventative Measures to Limit Downtime*. Accessed: Jul. 4, 2023. [Online]. Available: <https://dreisilker.com/blog/common-causes-servo-motor-failures-preventative-measures-limit-downtime/>
- [4] M. Win, A. R. Bushroa, M. A. Hassan, N. M. Hilman, and A. Ide-Ektessabi, "A contrast adjustment thresholding method for surface defect detection based on mesoscopy," *IEEE Trans. Ind. Informat.*, vol. 11, no. 3, pp. 642–649, Jun. 2015.
- [5] X.-C. Yuan, L.-S. Wu, and Q. Peng, "An improved Otsu method using the weighted object variance for defect detection," *Appl. Surf. Sci.*, vol. 349, pp. 472–484, Sep. 2015.
- [6] N. Neogi, D. K. Mohanta, and P. K. Dutta, "Defect detection of steel surfaces with global adaptive percentile thresholding of gradient image," *J. Inst. Eng. India, B*, vol. 98, no. 6, pp. 557–565, Dec. 2017.
- [7] C. Yang, J. Zhang, G. Ji, Y. Fu, and X. Hong, "Recognition of defects in steel surface image based on neural networks and morphology," in *Proc. 2nd Workshop Digit. Media Appl. Museum Heritages (DMAMH)*, Dec. 2007, pp. 72–77.
- [8] K. Q. Shi and W. G. Wei, "Image denoising method of surface defect on cold rolled aluminum sheet by bilateral filtering," *Surf. Technol.*, vol. 47, no. 9, pp. 326–332, 2018.
- [9] Y. Qiu, S. Niu, T. Niu, W. Li, and B. Li, "Joint-prior-based uneven illumination image enhancement for surface defect detection," *Symmetry*, vol. 14, no. 7, p. 1473, Jul. 2022.
- [10] D. J. Pasadas, P. Baskaran, H. G. Ramos, and A. L. Ribeiro, "Detection and classification of defects using ECT and multi-level SVM model," *IEEE Sensors J.*, vol. 20, no. 5, pp. 2329–2338, Mar. 2020.
- [11] R. Shanmugamani, M. Sadique, and B. Ramamoorthy, "Detection and classification of surface defects of gun barrels using computer vision and machine learning," *Measurement*, vol. 60, pp. 222–230, Jan. 2015.
- [12] R. Zaghoudi, H. Seridi, A. Boudiaf, and S. Ziani, "Binary Gabor pattern (BGP) descriptor and principal component analysis (PCA) for steel surface defects classification," in *Proc. Int. Conf. Adv. Aspects Softw. Eng. (ICAASE)*, Nov. 2020, pp. 1–7.
- [13] S. V. Patel and V. N. Jokhakar, "A random forest based machine learning approach for mild steel defect diagnosis," in *Proc. IEEE Int. Conf. Comput. Intell. Comput. Res. (ICIC)*, Dec. 2016, pp. 1–8.
- [14] J. Zhao, Y. Peng, and Y. Yan, "Steel surface defect classification based on discriminant manifold regularized local descriptor," *IEEE Access*, vol. 6, pp. 71719–71731, 2018.
- [15] Q. Luo, X. Fang, Y. Sun, L. Liu, J. Ai, C. Yang, and O. Simpson, "Surface defect classification for hot-rolled steel strips by selectively dominant local binary patterns," *IEEE Access*, vol. 7, pp. 23488–23499, 2019.
- [16] W. Ouyang, B. Xu, J. Hou, and X. Yuan, "Fabric defect detection using activation layer embedded convolutional neural network," *IEEE Access*, vol. 7, pp. 70130–70140, 2019.
- [17] L. Cui, X. Jiang, M. Xu, W. Li, P. Lv, and B. Zhou, "SDDNet: A fast and accurate network for surface defect detection," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–13, 2021.
- [18] G. Wen, Z. Gao, Q. Cai, Y. Wang, and S. Mei, "A novel method based on deep convolutional neural networks for wafer semiconductor surface defect inspection," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 12, pp. 9668–9680, Dec. 2020.
- [19] J. Sun, P. Wang, Y.-K. Luo, and W. Li, "Surface defects detection based on adaptive multiscale image collection and convolutional neural networks," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 12, pp. 4787–4797, Dec. 2019.
- [20] J. Yang, G. Fu, W. Zhu, Y. Cao, Y. Cao, and M. Y. Yang, "A deep learning-based surface defect inspection system using multiscale and channel-compressed features," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 10, pp. 8032–8042, Oct. 2020.
- [21] C. Wang and H. Xie, "MeDERT: A metal surface defect detection model," *IEEE Access*, vol. 11, pp. 35469–35478, 2023.
- [22] S. Guan, M. Lei, and H. Lu, "A steel surface defect recognition algorithm based on improved deep learning network model using feature visualization and quality evaluation," *IEEE Access*, vol. 8, pp. 49885–49895, 2020.
- [23] Y. Liu, J. Geng, Z. Su, W. Zhang, and J. Li, "Real-time classification of steel strip surface defects based on deep CNNs," in *Proc. Chin. Intell. Syst. Conf.*, vol. 2. Singapore: Springer, 2019, pp. 257–266.
- [24] P. Kumar, A. Sharma, and S. R. Kota, "Automatic multiclass instance segmentation of concrete damage using deep learning model," *IEEE Access*, vol. 9, pp. 90330–90345, 2021.
- [25] L. Qiu, X. Wu, and Z. Yu, "A high-efficiency fully convolutional networks for pixel-wise surface defect detection," *IEEE Access*, vol. 7, pp. 15884–15893, 2019.
- [26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–9.
- [27] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*.

- [28] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, *arXiv:1701.07875*.
- [29] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.
- [30] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
- [31] Q. N. The Ho, T. T. Do, P. S. Minh, V.-T. Nguyen, and V. T. T. Nguyen, "Turning chatter detection using a multi-input convolutional neural network via image and sound signal," *Machines*, vol. 11, no. 6, p. 644, Jun. 2023.
- [32] N. B. Bynagari, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," *Asian J. Appl. Sci. Eng.*, vol. 8, no. 1, pp. 25–34, Apr. 2019.
- [33] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.
- [34] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.
- [35] M. Tan and Q. Le, "EfficientNetV2: Smaller models and faster training," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 10096–10106.
- [36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [37] B. Mustafaev, A. Tursunov, S. Kim, and E. Kim, "A novel method to inspect 3D ball joint socket products using 2D convolutional neural network with spatial and channel attention," *Sensors*, vol. 22, no. 11, p. 4192, May 2022.



SUNGWON KIM joined Sejin Computer Land Logistics, as a Team Manager, in 1995. He was the CEO of Jun Network, from 1999 to 2007, and the CEO of RoboMoa Company Ltd., from 2007 to 2014. He was the Head of Mechanization Business Headquarters with Treksta Company Ltd. He is currently the CTO with Vazil Company Ltd., Busan, South Korea. His research interests include smart factory solutions, industrial robot applications, and inspection systems.



South Korea. His current research interests include deep learning, computer vision, pattern recognition, and the development of deep learning-based defect inspection and quality control algorithms for industrial applications.

BEKHZOD MUSTAFAEV received the B.S. degree in computer engineering from the Tashkent University of Information Technologies, Tashkent, Uzbekistan, in 2017, and the M.S. degree in computer science and engineering from the University of Ulsan, Ulsan, South Korea, in 2021, through the Global Korea Scholarship (GKS). He is currently pursuing the Ph.D. degree with the Department of Artificial Intelligence Convergence Engineering, Busan University of Foreign Studies, Busan,



EUNGSOO KIM received the Ph.D. degree in material science from Keio University, Japan, in 1996. He joined the ULSI Research Center, LG Semiconductor Inc., as a Senior Engineer, in 1996. Since 1997, he has been a Professor with the Department of Artificial Intelligence Convergence Engineering, Busan University of Foreign Studies. He was the Head of the Institute of Information and Communication, Busan University of Foreign Studies, from 2008 to 2010, and the Project Director (PD) of the Safety and Convenience Parts Project Dongnam Leading Industry Office, from 2010 to 2012. His research interests include optical sensors, nonlinear optics, optical waveguide devices, polymer fiber, and bio healthcare devices.

...