applied sciences

MDPI

Article

Detection of Defects on Metal Surfaces Based on Deep Learning

Onur Cem Han * and Uğurhan Kutbay

Department of Electrical-Electronic Engineering, Graduate School of Natural Sciences, Technology Faculty, Gazi University, Emniyet Neighborhood, Central Campus, 2nd Floor, B Block Extension Building, 06500 Ankara, Turkey
* Correspondence: onurcem00@gmail.com

Abstract: Various types of defects can occur on metal surfaces during production due to various factors. Detecting these defects is of great importance for the quality and reliability of the product. Manual inspections are time-consuming and prone to errors, especially as production scales increase Although Deep Learning and Computer Vision techniques show promise, the large data sizes of datasets created for deep learning and the high training costs of deep learning models lead to negative effects in terms of storage and energy efficiency. We propose a new method to improve defect detection rates, reduce labor losses, decrease data sizes, and improve energy efficiency. For the experiments, we used the Northeastern University (NEU) surface defect database as a reference, and the MobileNetV2 architecture as the model. The deep learning model was trained separately using both the grayscale-converted NEU database and the database created with the proposed method, and the accuracies of the datasets were compared. Image preprocessing techniques such as morphological operations, Gaussian noise addition, Principal Component Analysis, and image thresholding with Otsu thresholding were used for defect detection. The model was trained using the newly created database, achieving a successful result with an average accuracy rate of 87% using the proposed algorithm.

Keywords: deep learning; detection of defects; mechanical surfaces; image processing

Academic Editor: Andrea Prati

Received: 24 December 2024
Revised: 17 January 2025
Accepted: 24 January 2025
Published: 29 January 2025

Citation: Han, O.C.; Kutbay, U. Detection of Defects on Metal Surfaces Based on Deep Learning. *Appl. Sci.* **2025**, *15*, 1406. https://doi.org/10.3390/app15031406

1. Introduction

During the production of metal parts, material defects can occur due to factors such as personnel, supply chain issues, storage, and the production process. These defects can result from inadequate cleaning, lack of machine maintenance, environmental effects, assembly errors, tool usage mistakes, and more. The defects in question can be classified as crazing, inclusion, patch, pitted surface, rolled-in scale, and scratch.

Defects on metal surfaces not only lead to visual inadequacies in products but can also compromise their functional integrity. Metals with surface defects can be sensitive to high and sudden temperature changes, extreme pressures, or external factors like vibrations, depending on their application. This situation can lead to significant vulnerabilities, especially in critical areas. Therefore, it is crucial to detect these defects before the products are finalized or handed over to the user.

Nowadays, the detection of defects on metal surfaces is still performed manually, relying on traditional methods. Products exiting the production line are evaluated by an inspector, who assesses them based on product documentation and past experiences. This method of inspection makes the defect detection process subject to the inspector's discretion. The likelihood of detecting a defect varies based on factors such as the inspector's workload,

fatigue, and experience. To mitigate this issue, the use of computer vision and deep learning-based defect detection algorithms is of great importance.

Detecting defects on surfaces is of great importance in critical fields such as medicine, automotive, aviation, military, and more. Numerous deep learning and image processing-based methods have been proposed to enable defect detection independently of inspector supervision or to increase the inspector's awareness. Xue-wu et al. [1] proposed a system for classifying highly reflective metal surfaces, which includes image acquisition, image preprocessing, feature extraction, and classification methods. The classification was performed using SVM based on operations involving wavelet transform and spectral measures. Hu et al. [2] used an optimized elliptical Gabor filter (EGF) for defect detection on textured surfaces in their study and reported the effectiveness of the proposed method. Tsai et al. [3] utilized Fourier transformation to distinguish defective regions from the background and to detect defects in textured images. Juang et al. [4] employed K-means clustering and morphological processing techniques to detect psoriasis regions in colored skin images, allowing for differentiation of the size and area of psoriasis. Elbehiery et al. [5] enhanced surface defects on fired ceramic tiles using image processing techniques and morphological operations, making the detection easier for inspectors and enabling the observation of defects in a digital environment. Ng [6] revised the Otsu thresholding method in his study and improved defect detection performance with the new approach.

These methods are statistical and spectral methods proposed to make defects more prominent or to facilitate easier detection. Generally, these methods are influenced by external factors such as light reflections and shadows [7]. Such factors can hinder defect detection. To overcome this limitation and enable the establishment of automated systems, deep learning-based methods have been proposed. Wang et al. [8] aimed to increase accuracy and reduce processing time, proposing a method combining ResNet50 and faster R-CNN. Güçlü et al. [9] used bilinear CNN models for defect detection, achieving higher accuracy. Lien et al. [10] employed AlexNet for feature extraction in their study and used the SVM method for classification. Lin et al. [11] proposed a multi-scale sequential CNN named MobileNet-v2-dense to detect defects more efficiently. Selamet et al. [12] used the Shape From Shading (SFS) method combined with a Faster Regional Convolutional Neural Network (Faster R-CNN) model to extract the position, type, and surface characteristics of the defect. Li et al. [13] proposed a neural network called a lightweight fully convolutional neural network, achieving high accuracy with fewer model parameters. Tao et al. [14] designed and utilized a novel cascaded autoencoder (CASAE) architecture for detecting and locating defects. Xu et al. [15] proposed the self-Supervised Efficient Defect Detector (SEDD) method to achieve low computational consumption and fast detection. Liang et al. [16] suggested the mask gradient response-based threshold segmentation (MGRTS) method for defect detection, combining it with the Difference of Gaussians (DoG) method to increase the detection rate. Yang et al. [17] utilized the ULF module, PGS algorithm, and SIL-Net for effectively extracting and labeling low-contrast and small scratches. Zhu et al. [18] used the Swin Transformer for detecting and classifying surface defects, modifying the network structure to improve model performance and proposed the LSwin Transformer. Zhao et al. [19] proposed a vision Transformer model for detecting aluminum surface defects, using a multi-layer perceptron (MLP) as the final classifier. Song et al. [20] applied various optimizations to classify surface defects using the YOLOv8 algorithm.

In the automatic detection of defects on metal surfaces, challenges arise from factors such as low contrast on the surface, light reflections, noise, and shadows, leading to false detections or undetected defects. Additionally, high-resolution images, along with large datasets, can negatively impact the training cost of deep learning models, storage, and

energy efficiency. This situation hinders the automatic detection of defects using computer vision or deep learning, resulting in delays.

The aim of this study is to create a new dataset with a relatively smaller data size by utilizing image processing methods on a limited dataset. The images in the dataset are converted to binary to eliminate external factors that may distort the image and extract the patterns of defects. Furthermore, it is anticipated that training a new deep learning model with the newly produced dataset will yield higher accuracy and energy efficiency with a lower number of epochs.

In this study, a dataset that is independent of background variations and has a low data size was created by sequentially applying morphological operations, Gaussian blur, PCA, and Otsu thresholding techniques. The MobileNetV2 model was trained using the new dataset, resulting in high-accuracy and highly efficient classification by the end of the training process.

In this study, the NEU dataset containing 1800 defective images was used. This dataset consists of six classes, namely crazing, inclusion, patch, pitted surface, rolled-in scale, and scratch. The images underwent morphological operations, specifically closing and opening, to highlight the defective areas on the metal surface. After clarifying the defects, Gaussian blur was added to the images to reduce noise. PCA was applied to reduce the dimensions of the noise-free images and facilitate the easier detection of defects. In the final stage, Otsu's thresholding technique was used to eliminate environmental effects such as pattern extraction of defects and variations in background and brightness. The final images obtained were saved under their respective classes.

A percentage of 80% of the created dataset was allocated for training and 20% for testing, which was utilized in the training of the MobileNetV2 model. The training was conducted using both the grayscale original dataset (G-NEU) and the newly created dataset. A model with high accuracy and efficiency was successfully developed using a lower data volume, independent of environmental effects, and with a small number of epochs.

The article is organized as follows: Section 1 outlines the purpose of our study. Section 2 describes the fundamental methodology and techniques used. Section 3 summarizes the experiments and results. Finally, Section 4 addresses the conclusions and potential areas for improvement.

2. Methodology

In this section of the study, an algorithm is proposed and described for extracting features from images of metal surfaces with visual defects, saving the results, and classifying the defects using the trained new model. The algorithm consists of two parts. In the first part, the image processing step is performed to extract surface features. This step includes morphological operations, adding Gaussian noise, Principal Component Analysis (PCA), and the Otsu thresholding technique. In the second part, the classification model is trained. The MobileNetV2 architecture is used as the model, which will be trained and perform classification using the new dataset. In Figure 1., the flowchart is shown.

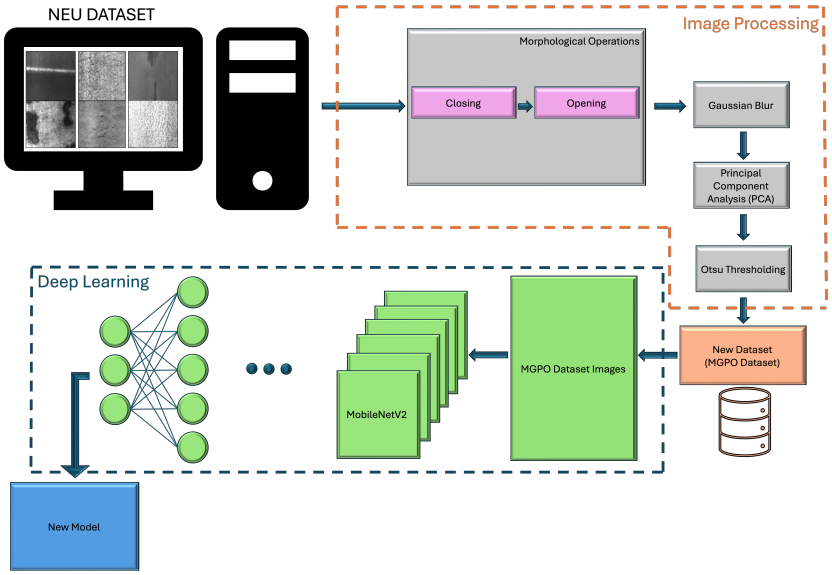

Figure 1. Flow diagram of the proposed method.

2.1. Image Processing

Image processing methods are applied to reduce the size of image data, make defects more prominent, and obtain a pattern that is independent of the background. In this algorithm, the NEU dataset containing 1800 images with defects is used. The images are first converted to grayscale.

$$I(x,y) = 0.2989 \cdot R(x,y) + 0.5870 \cdot G(x,y) + 0.1140 \cdot B(x,y) \quad (1)$$

while $I(x,y)$ represents the grayscale image, $R(x,y)$, $G(x,y)$, and $B(x,y)$ represent the red, green, and blue components of each pixel in the input image, respectively. To fill in the gaps in the grayscale image and make the defective areas more prominent, morphological operations [21,22] are applied to the image. A structural element is defined during the operations, and in the related study, a structural element of size 3×3 was used. The first step in the sequence of morphological operations is the closing phase. This process involves applying dilation followed by erosion. The dilation operation expands the white pixels of the object and turns each black pixel that intersects with the structuring element's white pixels into white. In this way, the boundaries of the object are expanded. The erosion operation is applied to the expanded image obtained after the dilation step. Erosion shrinks the boundaries of the object, ensuring that the entire structuring element fits within the object. Thus, it preserves only those pixels where the structuring element is completely within the object.

$$I(x,y) \cdot B = (I(x,y) \oplus B) \ominus B \quad (2)$$

$I(x,y)$ represents the grayscale image. B is the structuring element used to perform operations on the pixels in the image. The structuring element slides over the image. The symbol $\oplus$ denotes the dilation operation. $\ominus$ denotes the erosion operation. The second step of morphological operations is the Opening operation, which consists of erosion followed by dilation. Erosion is used to shrink objects in the image and to eliminate small objects or thin structures. Dilation then enlarges the image and increases the size of the objects, partially reversing the "shrinking" effect left by the erosion process, but without restoring the removed noise.

$$I(x,y) \circ B = (I(x,y) \ominus B) \oplus B \quad (3)$$

$I(x,y)$ represents the image resulting from the closing operation and serves as the input for the opening operation. B is the structuring element. $\ominus$ denotes the erosion operation, while $\oplus$ represents the dilation operation. $I(x,y) \circ B$ represents the opening operation. After these operations, Gaussian blur is applied to the image to remove any remaining noise and smooth out non-defective details. This process helps eliminate details that could potentially hinder defect detection on the image.

$$G(x,y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{4}$$

$$I'(x,y) = I * G(x,y) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} I(m,n) \cdot G(x-m, y-n) \tag{5}$$

$I'(x,y)$ represents the output image, $I(x,y)$ represents the input image, and $G(x,y)$ denotes the Gaussian function. A convolution operation is performed between the image and the filter. The convolution operation applies the filter to each pixel of the image, resulting in a new image. After this stage, Principal Component Analysis (PCA) [23] is applied. The aim here is to reduce the dimensionality of the image data and to enhance the visibility of the defects. The image matrix was flattened and normalized by zero-centering its mean. Subsequently, the covariance matrix was constructed, and eigenvalues and eigenvectors were computed to project the data onto a lower-dimensional space, thereby enhancing the efficiency of feature extraction. PCA reduced data density and processing costs through dimensionality reduction. With a value of 0.8, sufficient information from the image was retained.

In the final stage of the study, the Otsu thresholding method was applied to separate defects from the background and enable the detection of these defects independent of the background type. This method examines the relationship between the pixels in the image and automatically determines the optimal threshold value that minimizes the within-class variance of both classes (defect and background). This makes the defects more prominent. In the Otsu thresholding method [24], the pixels in the image are divided into two distinct classes: one representing the background and the other representing the defects. Classification is based on intensity distribution. The histogram of the image represents the proportion of each gray-level intensity relative to the total number of pixels. The method calculates the most suitable threshold value from the histogram to maximize the distinction between the classes. For this calculation, the weighted averages of the classes are used and compared to the overall mean intensity of the image. The goal is to maximize the variance between the two classes, thereby making them as distinct as possible. In this way, the threshold value that maximizes this variance is determined and used to separate defects from the background.

$$\begin{aligned} C_1 &: \quad 0 \leq i \leq t \quad \Rightarrow \quad \text{Background} \\ C_2 &: \quad t < i \leq 255 \quad \Rightarrow \quad \text{Foreground (Defect)} \end{aligned}$$

C_1 and C_2 represent two separate classes. One of them is the defect class, while the other is the background class. Image with pixel values ranging from 0 to 255 (i). T In this way, the defect in the image is separated independently from the background.

2.2. Classification Process

The preprocessing step has been applied to all images belonging to the six different classes in the NEU dataset, resulting in the creation of a new dataset. This new dataset has been used to train a model with the MobileNetV2 algorithm. MobileNetV2 [25] is a deep learning model introduced by Google in 2018. It offers advantages in terms of learning

speed and memory management. There are two key features that distinguish MobileNetV2 from MobileNetV1: Inverted Residuals and Linear Bottlenecks.

With Inverted Residuals, standard residual blocks are not used; feature maps are reduced to a lower dimension and then expanded again. This approach allows for the use of fewer parameters. With the Expansion Convolution, a low-dimensional input is transformed into a wider feature map. Depthwise Convolution applies convolution separately for each channel, following a different path from traditional convolutions, thus eliminating unnecessary parameters. A Projection Layer is then used to compress the expanded feature maps. In the Projection Layer, a Linear Bottleneck is used; here, to enhance the efficiency of the model, a nonlinear activation function (ReLU) is not used in narrow layers. This way, efficiency is improved.

The MobileNetV2 model trained with the newly created dataset has been used to detect different types of defects.

3. Experimental Results

3.1. Image Processing Techniques

For the experiments, the surface defect data of hot-rolled steel strips from the NEU (Northeastern University) database were used. The NEU dataset consists of surface defect images belonging to six different classes. The defect classes are defined as crazing, inclusion, patch, pitted surface, rolled-in scale, and scratch. Each defect class contains 300 images, resulting in a total of 1800 images of defective metal surfaces. The images have a resolution of 200 $\times$ 200 pixels. The images of the defect classes in the NEU dataset are shown in Figure 2.

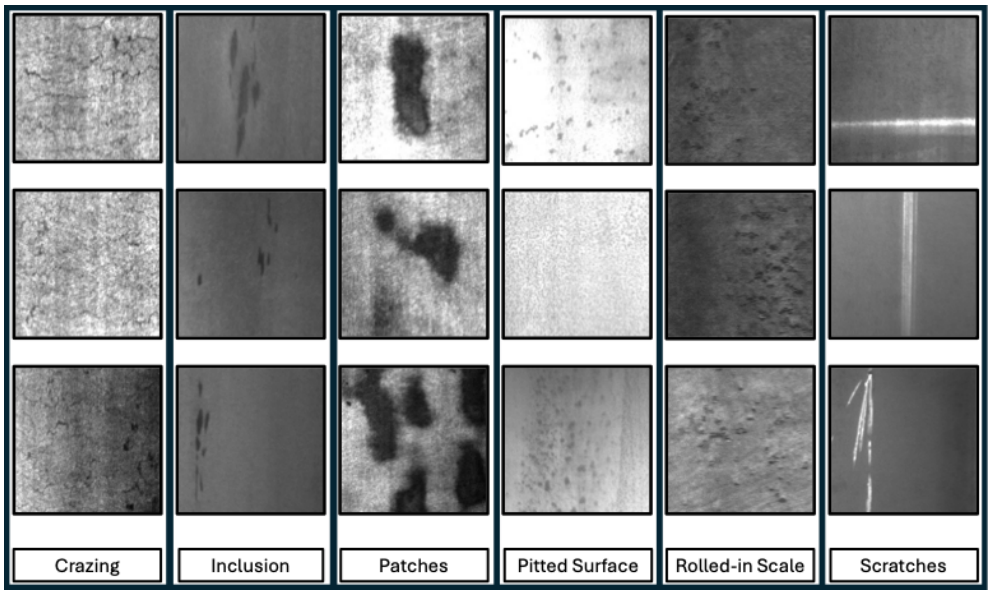

Figure 2. The images of the defect classes in the NEU dataset.

At this stage of the study, the MobileNetV2 architecture was trained using different image processing techniques. As shown in Table 1, various combinations of image processing techniques were employed. The aim here is to determine the most accurate combination.

For the technique referred to as GCPOM, the following steps were applied sequentially: adding Gaussian noise, CLAHE histogram equalization, feature extraction and dimensionality reduction using PCA, binarizing the image with Otsu thresholding, and highlighting defects through morphological operations. In the MGH technique, morphological operations, Gaussian noise addition, and HOG transformation were performed sequentially. For the PKO technique, PCA, kernel-based processing, and Otsu thresholding

were utilized. In the O technique, only Otsu thresholding was applied. In the MO technique, morphological operations and Otsu thresholding were applied consecutively. In the MGO technique, morphological operations, Gaussian noise addition, and Otsu thresholding were used sequentially.

Table 1. Accuracy of Different Image Processing Techniques.

Image Processing Technique	Accuracy (%)
GCPOM	80
MGH	70
PKO	68
O	75
MO	83
MGO	85
MGPO	87

As seen in Table 1, the highest accuracy value was achieved using the MGPO image processing techniques. This is due to the appropriate combination of methods and the complementary nature of the image-processing steps.

With morphological operations, small noise around the defects was cleaned. The Gaussian noise addition step introduced artificial noise to the images, enhancing the model's generalization capability. PCA was utilized to extract features from the images, reduce dimensionality, and compress essential information.

In the final step, the image was binarized automatically using Otsu thresholding, making the defective areas more distinct. This approach effectively highlighted defects independent of the background, leading to the highest accuracy achieved.

The images were re-saved following the processes specified in the study (MGPO). A new dataset was created as a result of the proposed method, consisting of the same six classes. As shown in Table 2, the total size of the images has been reduced compared to the NEU dataset. The new dataset contains 1800 images of six different classes, each with a resolution of 200 $\times$ 200 pixels.

Table 2. Image and Data Sizes of the Datasets.

Dataset	Data Size (MB)	Image Size
G-NEU	37.8	200 $\times$ 200
MGPO	26.2	200 $\times$ 200

In this study, the Grayscale NEU dataset and the MGPO dataset were used for training the MobileNetV2 model, and the two trained models were compared; 80% of the datasets were used for training and 20% for testing. As a result of the study, it was observed that the experimental results aligned with the objectives of the study.

3.2. Experimental Results

In this study, Google Colab, and a computer with an Apple M2 processor and 8 GB of RAM were used. In the experiments, a pre-trained MobileNetV2 model was used. Since the goal was to achieve higher accuracy in a shorter time, the number of epochs was set to 30. During the experiments, the images were first converted to grayscale. The grayscale images were then subjected to closing and opening morphological operations in sequence. Figure 3 shows examples of images after the processing steps.

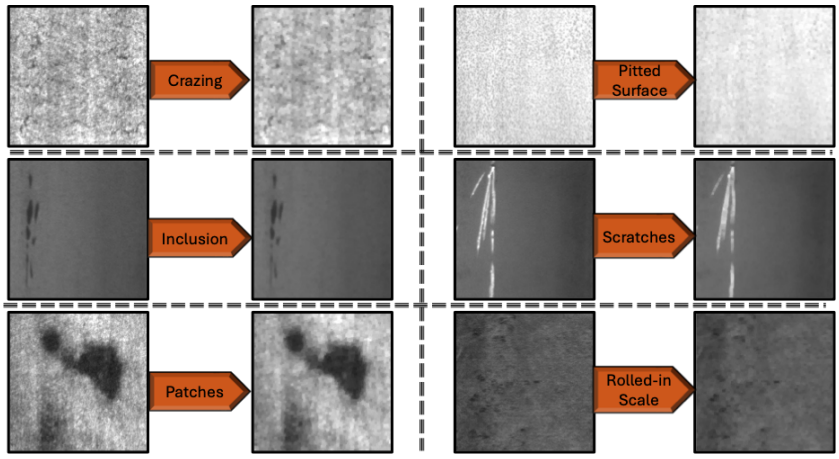

Figure 3. Defective images after morphological operations.

In order to minimize noise in the resulting images, Gaussian blur was applied. This process helped to remove details that could hinder defect detection in the images.

PCA was applied to achieve the objectives of reducing data size, increasing training efficiency, and using resources more effectively. An 80% threshold value was set for this purpose. Otsu thresholding was performed to ensure that defect detection in the images processed with PCA was conducted without relying on background differences and to extract a pattern specific to the defect. This process resulted in the extraction of a pattern for the defect classes and a reduction in data size. Example images resulting from the MGPO process are shown in Figure 4.

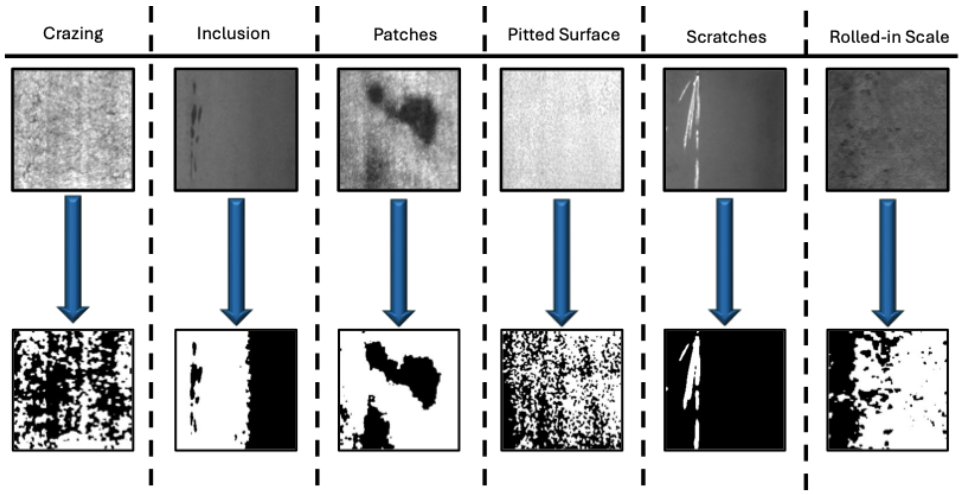

Figure 4. Images After the MGPO Process.

The MobileNetV2 model was trained using both datasets with a total of 30 epochs. In both datasets, 80% of the images were used for training and 20% for testing. The confusion matrix allows us to evaluate the performance of a model across different classes. It provides insight into the relationships between the classes in the model. The matrix includes values for true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). The confusion matrices for the models are shown in Figure 5.

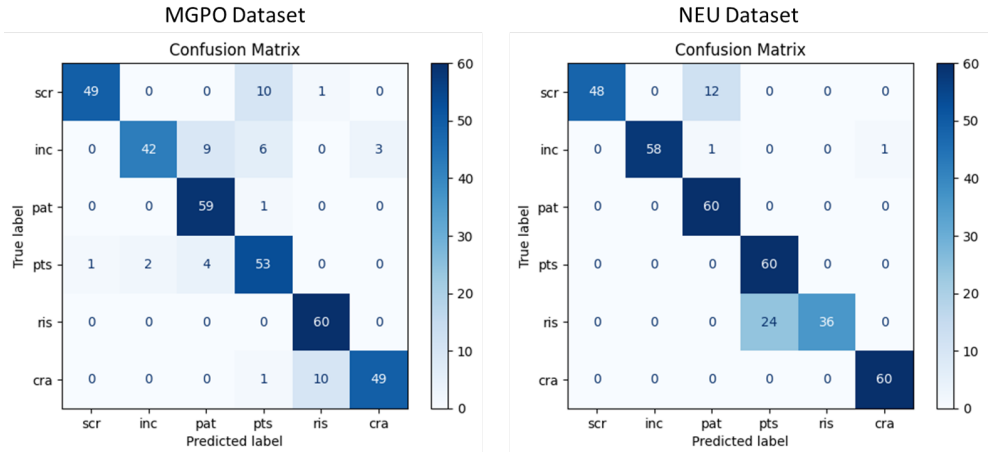

Figure 5. Confusion matrices of the MGPO and NEU datasets.

A classification that will be considered successful across all classes has been performed. A low number of epochs was used in the model training, and the data sizes of the images were reduced compared to their original state.

The accuracy value indicates the ratio of correctly predicted instances to all predictions. Precision reflects the accuracy of a model's predictions and is calculated as the ratio of true positive predictions to all positive predictions. The recall value shows how many of the actual positives were correctly classified. The F1 score is calculated as the harmonic mean of precision and recall values. The mean Average Precision (mAP) is computed by taking the average accuracy across all classes, representing the overall performance of the model.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{7}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{8}$$

$$\text{F1 Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{9}$$

Efficiency is calculated as the ratio of model accuracy to power consumption. This allows for the evaluation of the energy efficiency of the trained model.

$$\text{Efficiency} = \frac{\text{Model Accuracy}(\%)}{\text{Energy Consumption (kWh)}} \tag{10}$$

Table 2 shows the efficiencies of the MGPO and G-NEU datasets. It has been observed that the efficiency of the MGPO dataset is nearly the same. In the G-NEU dataset, the separation of defects is not only based on the features of the defects but also influenced by the fact that the defect classes have different backgrounds. Therefore, the accuracy appears higher. However, in the MGPO dataset, all defects have been reduced to the same background. Despite this, a successful accuracy has been achieved. An efficiency value of 1.46 was measured with the MGPO dataset.

Table 3 shows the model's performance in detecting defects on a class-by-class basis. Upon examining the values presented in the table, it is observed that the model trained with the MGPO dataset has nearly equal or slightly lower accuracy in detecting defects of crazing, inclusion, patch, pitted surface, and scratch compared to the model trained with the NEU dataset. However, it is noted that the model trained with the MGPO dataset

performs particularly better in detecting the rolled-in scale defect. When examining the mAP value, it has been found that the accuracy of the model trained with the MGPO dataset is similar to that of the model trained with the Normal dataset. With the proposed method, the detection rates for the defect classes were found to be 0.82, 0.70, 0.98, 0.88, 1.00, and 0.82 for crazing, inclusion, patches, pitted surface, rolled-in scale, and scratch, respectively. The relevant results are observed in Table 4. The average accuracy value was measured as 0.87. As seen, similar results were obtained compared to the model trained with the Grayscale NEU dataset. With the MGPO dataset, defect patterns have been successfully extracted with the same efficiency, independent of the background, and with a lower data size.

Table 3. Efficiency Values of Datasets Against Accuracy and Energy Consumption.

Dataset	Accuracy (mAP%)	Energy Consumption	Efficiency
G-NEU	89	61.36	1.45
MGPO	87	59.28	1.46

Table 4. Class-based Performance Report.

Class	Precision	Recall	F1 Score	Accuracy
Crazing	0.98	0.82	0.89	0.82
Inclusion	0.95	0.70	0.81	0.70
Patches	0.82	0.98	0.89	0.98
Pitted Surface	0.75	0.88	0.81	0.88
Rolled-in Scale	0.85	1.00	0.92	1.00
Scratch	0.94	0.82	0.88	0.82
Avg	0.88	0.87	0.87	0.87

4. Conclusions and Future Works

This article presents the detection of defects on metal surfaces. The dataset used for detection was obtained by utilizing the original dataset. In the new dataset, defects are independent of the background, and it has a reduced data size. A deep learning model was trained using this dataset with a low number of epochs, resulting in high efficiency.

To create the new dataset, the original images were first converted to grayscale. Subsequently, the images underwent morphological operations such as opening and closing. Gaussian noise was added to the images, followed by principal component analysis (PCA). Finally, patterns from the images were extracted using Otsu's thresholding method. These processes led to the formation of a new dataset.

The MobileNetV2 model was trained with this new dataset using a low number of epochs. In the model training, 80% of the dataset was used for training and 20% for testing. When comparing the grayscale original dataset with the new dataset based on model efficiency, it was observed that the proposed method achieved high efficiency.

The proposed method may not be able to detect multiple different defects within the same image. Additionally, it may not perform stably in detecting defects in images with low contrast. Future studies could employ different techniques in the image processing stages to more clearly delineate and label defects. With the new dataset, artificial intelligence models that can detect and classify multiple defects with higher efficiency in real time could be trained.

Author Contributions: Methodology, O.C.H.; Software, O.C.H.; Resources, O.C.H.; Writing—original draft, O.C.H.; Writing—review & editing, U.K.; Visualization, O.C.H.; Supervision, U.K.; Project administration, U.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original data presented in this study are openly available in Kaggle at https://www.kaggle.com/datasets/kaustubhdikshit/neu-surface-defect-database (accessed on 20 December 2024).

Acknowledgments: During the examination and preparation of the methods mentioned in this manuscript, the ChatGPT artificial intelligence language model developed by OpenAI was utilized to enhance the readability and grammatical accuracy of the text. The generated content was reviewed and revised by the authors to ensure accuracy and alignment with the original work.

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

References

1. Zhang, X.; Ding, Y.; Lv, Y.; Shi, A.; Liang, R. A vision inspection system for the surface defects of strongly reflected metal based on multi-class SVM. *Expert Syst. Appl.* **2011**, *38*, 5930–5939.
2. Hu, G.-H. Automated defect detection in textured surfaces using optimal elliptical Gabor filters. *Optik* **2015**, *126*, 1331–1340.
3. Tsai, D.-M.; Huang, T.-Y. Automated surface inspection for statistical textures. *Image Vis. Comput.* **2003**, *21*, 307–323.
4. Juang, L.-H.; Wu, M.-N. Psoriasis image identification using k-means clustering with morphological processing. *Measurement* **2011**, *44*, 895–905.
5. Elbehiery, H.; Hefnawy, A.; Elewa, M. Surface defects detection for ceramic tiles using image processing and morphological techniques. *Citeseer World Acad. Sci. Eng. Technol.* **2005**, 158–162.
6. Ng, H.-F. Automatic thresholding for defect detection. *Pattern Recognit. Lett.* **2006**, *27*, 1644–1649.
7. Luo, Q.; Fang, X.; Liu, L.; Yang, C.; Sun, Y. Automated visual defect detection for flat steel surface: A survey. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 626–644.
8. Wang, S.; Xia, X.; Ye, L.; Yang, B. Automatic detection and classification of steel surface defect using deep convolutional neural networks. *Metals* **2021**, *11*, 388.
9. Emre, G.; İlhan, A.; Erhan, A. Çift Doğrusal CNN Kullanarak Çelik Yüzey Kusurlarının Sınıflandırılması. *FıRat Üniversitesi MüHendislik Bilim. Derg.* **2024**, *36*, 267–280.
10. Chun, L.P.; Zhao, Q. Product surface defect detection based on deep learning. In Proceedings of the 2018 IEEE 16th International Conference on Dependable, Autonomic and Secure Computing, Athens, Greece, 12–15 August 2018; pp. 250–255.
11. Li, Z.; Ye, H.; Zhan, H.; Huang, X. An efficient network for surface defect detection. *Appl. Sci.* **2020**, *10*, 6085.
12. Feyza, S.; Serap, C.; Muhammed, K. Automatic detection and classification of defective areas on metal parts by using adaptive fusion of faster R-CNN and shape from shading. *IEEE Access* **2022**, *10*, 126030–126038.
13. Li, Y.; Chen, Y.; Gu, Y.; Ouyang, J.; Wang, J.; Zeng, N. A lightweight fully convolutional neural network of high accuracy surface defect detection. In *Artificial Neural Networks and Machine Learning–ICANN 2020: 29th International Conference on Artificial Neural Networks, Bratislava, Slovakia, 15–18 September 2020*; Proceedings, Part II 29; Springer: Berlin/Heidelberg, Germany, 2020; pp. 15–26.
14. Tao, X.; Zhang, D.; Ma, W.; Liu, X.; Xu, D. Automatic metallic surface defect detection and recognition with convolutional neural networks. *Appl. Sci.* **2018**, *8*, 1575.
15. Rongge, X.; Ruiyang, H.; Biqing, H. Efficient surface defect detection using self-supervised learning strategy and segmentation network. *Adv. Eng. Inform.* **2022**, *52*, 101566.
16. Ying, L.; Ke, X.; Peng, Z. Mask gradient response-based threshold segmentation for surface defect detection of milled aluminum ingot. *Sensors* **2020**, *20*, 4519.
17. Yang, L.; Zhou, F.; Wang, L. A scratch detection method based on deep learning and image segmentation. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–12.
18. Zhu, W.; Zhang, H.; Zhang, C.; Zhu, X.; Guan, Z.; Jia, J. Surface defect detection and classification of steel using an efficient Swin Transformer. *Adv. Eng. Inform.* **2023**, *57*, 102061.
19. Zhao, J. Surface defect classification with vision transformer. In Proceedings of the 2022 3rd International Conference on Intelligent Design (ICID), Xi'an, China, 21–23 October 2022; pp. 124–128.

20. Song, X.; Cao, S.; Zhang, J.; Hou, Z. Steel Surface Defect Detection Algorithm Based on YOLOv8. *Electronics* **2024**, *13*, 988.
21. Maragos, P. A representation theory for morphological image and signal processing. *IEEE Trans. Pattern Anal. Mach. Intell.* **1989**, *11*, 586–599.
22. Haralick, R.M.; Sternberg, S.R.; Zhuang, X. Image Analysis Using Mathematical Morphology. *IEEE Trans. Pattern Anal. Mach. Intell.* **1987**, *9*, 532–550.
23. Ng, S.C. Principal component analysis to reduce dimension on digital image. *Procedia Comput. Sci.* **2017**, *111*, 113–119.
24. Xu, X.; Xu, S.; Jin, L.; Song, E. Characteristic analysis of Otsu threshold and its applications. *Pattern Recognit. Lett.* **2011**, *32*, 956–961.
25. Mark, S.; Andrew, H.; Menglong, Z.; Andrey, Z.; Liang-Chieh, C. MobileNetV2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.