Advanced Mechanical and Mechatronic Systems

Research Article
Volume 2(2), 2026, 88–105
https://doi.org/10.53623/amms.v2i1.995

Machine Vision Lab Experiment Based Convolutional Neural Network for Potential Intelligent Manufacturing Applications

Ardian Webi Kirda[1], Kamil Gatnar[2], Khairul Muzaka[3], Nur Arifin Akbar[4]*

[1]Faculty of Integrated Technologies, Universiti Brunei Darusalam, Jalan Tungku Link, BE1410, Brunei Darussalam
[2]Faculty of Mechanical Engineering, Opole University of Technology, 76 Proszkowska St., 45-758 Opole, Poland
[3]Department of Mechanical Engineering, State Polytechnic of Banyuwangi, Indonesia
[4]Department of Mathematic and Computational Sciences, University of Messina, Italy

*Correspondence: nur.akbar@studenti.unime.it

SUBMITTED: 15 January 2026; REVISED: 19 February 2026; ACCEPTED: 9 May 2026

ABSTRACT: This study developed a deep learning-based image classification method for the automated assessment of aluminum edge quality after machining processes. The proposed approach classified edge conditions into three categories: Normal Edge, Burr Edge, and Sharp Edge. A Convolutional Neural Network (CNN) based on the VGG16 architecture was employed as the feature extraction backbone, with modifications to the final fully connected layers to accommodate the three-class classification task. The model was trained and evaluated on a dataset of 660 aluminum edge images captured under controlled laboratory conditions at the Robotic Manufacturing Laboratory, University of Brunei Darussalam. The training strategy employed a stratified split with 360 training images, 240 testing images, and 60 validation images. Data augmentation techniques (horizontal flip, rotation ±15°, brightness adjustment) were applied to enhance model generalization. The optimized model achieved an overall classification accuracy of 98.75% on the test set. Precision, recall, and F1-scores for all three classes exceeded 0.97. For practical deployment, the trained model was deployed on an NVIDIA Jetson Nano embedded platform, achieving an average inference time of 47 ms per image at an input resolution of 500×500 pixels (approximately 21 FPS). These results demonstrated the feasibility of real-time edge quality assessment for intelligent manufacturing applications.

KEYWORDS: Convolutional neural network; intelligent manufacturing; metal edge classification; real-time quality assessment; embedded systems

1. Introduction

The detection and classification of edge defects, such as burrs and sharp edges, was a crucial aspect of quality control in modern manufacturing processes. Edge defects represented surface

imperfections that could lead to product failures, dimensional inconsistencies, and potential safety hazards during product handling and operation. In machining operations, the formation of burrs and sharp edges was often unavoidable due to material deformation during cutting, shearing, or milling processes. These imperfections could compromise the functional integrity of manufactured components, cause mechanical damage during assembly operations, and create aesthetic defects that affected product appearance and customer satisfaction. The implementation of automated, real-time edge quality assessment systems was therefore essential for ensuring consistent product quality and enabling timely corrective actions in manufacturing environments [1].

Detection using machine vision became increasingly popular due to its advantages of low-cost equipment, fast processing speed, and high recognition accuracy [1]. To achieve intelligent feature detection through machine vision, two technical challenges needed to be addressed: intelligent recognition of inconsistent edge features and accurate measurement of their size [2]. A number of researchers had addressed these challenges through the development of various algorithms. For instance, a fuzzy clustering algorithm consisting of two distinct phases was proposed for segmenting polarimetric synthetic aperture radar remote sensing images into regions [3], while another study utilized multilayer perceptron (MLP) supervised learning to identify hard exudates in retinal images through iterative graph cut methods [4]. Additionally, the maximum expectation algorithm was applied to weight edge points in image matching, thereby reducing computation time. A Gaussian mixture algorithm was employed to construct a weighted least squares model to represent object background and facilitate defect detection [5]. Moreover, a wavelet-based method was used with depth image reconstruction for inspecting micro-milling tools [6]. Another study established a shape model for high-precision visual localization in part reconstruction [7]. In a similar vein, other research extracted surface roughness and dimensional deviation data of machined workpieces through two-dimensional imaging of cutting tools, improving detection speed [8]. In another case, a precise measurement method was developed for modeling and simulating elastic deformations in machine parts using an efficient mass–spring–damper model [9]. Despite these advances, challenges remained in the actual recognition of edge features.

Historically, the detection of sharp and burr edges had been carried out in three main steps. First, image preprocessing was performed using image processing techniques to enhance image quality by removing noise, adjusting brightness, and improving contrast. Second, histogram analysis, wavelet transform, or Fourier transform methods were applied to obtain image representations in specific feature spaces, facilitating manual feature extraction. Lastly, the image was classified based on extracted features using a classifier [10]. In general, research on sharp and burr edge detection and recognition had focused on low-level image semantic features obtained through manual extraction [11]. In industrial production, mechanical parts often exhibited inconsistent shapes due to the constraints of bulk processing, making it challenging to recognize inconsistent edge features. Recognition of these features played a critical role in enhancing intelligent machining levels for batch-processed workpieces [12]. These inconsistent features were characterized by varying geometric contours and randomly distributed positions. Conventional

contour measurement techniques, such as machine vision systems or coordinate measuring machines, were capable of determining edge profile dimensions but struggled to identify edge features and extract their fundamental characteristics [13]. Hence, research efforts aimed at improving the detection accuracy of inconsistent edge features had received considerable attention in recent years [14].

Therefore, there was still room for improvement in intelligent detection of edge-inconsistent features [15]. In the field of intelligent feature recognition [16,17], deep learning had gained widespread popularity due to its ability to automatically extract high-level semantic features without manual feature engineering and its strong robustness [18]. This technology was often combined with other intelligent optimization algorithms to extract high-level semantic features from images [19,20]. A summary of representative studies on grinding and cutting tool wear monitoring and related intelligent methods is presented in Table 1, which highlights different approaches such as artificial neural networks, image segmentation techniques, CNN-based models, and hybrid deep learning architectures.

Table 1. Experiments in grinding and cutting tool wear monitoring.

Reference	Method	Intelligent Algorithms
[21]	Artificial Neural Network	Otsu-Canny Edge Detection
[22]	Image Segmentation	Laplacian, Canny, CASENet
[23]	Image Segmentation	Eight Direction Sobel
[24]	Artificial Neural Network	Otsu-Canny Edge Detection
[25]	Image Segmentation	Gaussian Pulse
[26]	Discrete Time Cellular Neural Network (DTCNN)	CNNEDGEPOT
[27]	Convolutional Neural Network (CNN)	HED layers
[28]	CNN + LSTM	X-IIoTID, UNSW-NB15
[29]	Mask R-CNN	Path Augmentation Feature Pyramid Network (PAFPN)

There have been several studies that have applied this approach to the detection of edge-inconsistent features. For instance, a deep belief network (DBN) was utilized to detect edge-inconsistent features from sample data [30], and deep learning was used for the automatic identification of materials with varying compositions and orientations of microstructural features [31]. Another study also observed that deep learning with multi-stage convolutional networks demonstrated the highest overall performance in describing local binary features [32]. Moreover, an automatic method was proposed that combined the Holistically Nested Edge Detection algorithm (HED) to accurately identify closed contours in images, resulting in a 17% reduction in the mismatching rate [33]. Meanwhile, another study utilized the Adaptive Neuro-Fuzzy Inference System (ANFIS) to generate results correlated with the predicted surface finish quality, specifically in relation to the length of the boss hole chamfer and the deburring process stage classification [34].

Convolutional Neural Networks (CNNs) have gained widespread use in manufacturing processes. In one study, researchers developed an ensemble CNN detection model for analyzing audio signals transformed into images using various algorithms [35], while another study designed

2D/3D sketch-based 3D retrieval methods for representing 3D objects [36]. In another study, researchers proposed a deep space-varying convolutional neural network model for detecting image boundary features and inconsistencies [37]. This model demonstrated lower memory usage and computational complexity compared to existing approaches [37]. Another team developed an improved CNN for identifying road surface cracks [38]. Lastly, a review discussed the evolution of CNN models for image datasets and used an automated convolutional encoder for pre-training, achieving an average precision rate of up to 96.2% [39].

In this study, the classification of sharp and burr edge features of mechanical workpieces was investigated using an intelligent scheme based on deep learning. The datasets were divided into three classes: “Normal Edge”, “Burr Edge”, and “Sharp Edge”. CNN played a vital role as a feature extractor through its ability to learn from diverse input signals using convolutional layers. These extracted features were then used for classification in subsequent layers. Spatial dimensionality reduction was achieved by applying pooling, which retained essential information. For determining specific target outputs, fully connected nodes computed a weighted sum of the previous layer’s outputs. The CNN approach offered conceptual advantages by reducing the training burden and serving as an efficient feature extraction technique for automated sharp and burr edge classification. It eliminated the need for separate preprocessing and feature extraction methods [41].

2. Dataset and Data Preprocessing

2.1. Data collection.

The input data for the CNN method comprised images of normal edge, burr edge, and sharp edge made of aluminum. These images were collected from the Robotic Manufacturing Laboratory at the University of Brunei Darussalam. The dataset comprised 660 images that were classified into three categories: “Normal Edge”, “Burr Edge”, and “Sharp Edge”. The data were captured using a mobile phone camera from various angles to ensure a diverse range of variations. To enhance dataset diversity and robustness to viewpoint variations, images were captured from multiple angles and positions relative to the workpiece edge. All images were obtained from distinct aluminum workpiece specimens, ensuring that the same physical edge did not appear in both training and testing sets. The three edge categories represented clearly distinguishable quality levels: Normal Edge referred to properly finished edges without visible defects; Burr Edge referred to edges with visible material protrusions or irregularities resulting from machining operations; and Sharp Edge referred to edges with excessive sharpness that could pose handling hazards or indicate incomplete deburring operations.

2.2. Data split strategy.

To ensure reliable evaluation and prevent data leakage between training and testing sets, the dataset was partitioned using a stratified random split. The total of 660 images was divided into 360 training images (120 per class), 240 testing images (80 per class), and 60 validation images (20

per class). Importantly, images from the same workpiece specimen were kept within the same split, preventing different views of the same physical edge from appearing in both training and testing sets. This approach provided a more realistic assessment of the model's generalization capability to new, unseen workpieces. The dataset distribution across training, validation, and test sets is summarized in Table 2.

Table 2. Dataset distribution across training, validation, and test sets.

Edge Class	Training	Validation	Testing	Total
Normal Edge	120	20	80	220
Burr Edge	120	20	80	220
Sharp Edge	120	20	80	220
Total	**360**	**60**	**240**	**660**

2.3. Image preprocessing.

Prior to constructing the CNN model, the raw images were preprocessed using morphological transformations to standardize the image format. Each aluminum edge image was cropped to a fixed size of 500 × 500 pixels, with careful consideration of the object position to ensure uniformity across all samples. These preprocessing adjustments helped improve consistency in the dataset and reduce variations caused by image acquisition conditions. Examples of preprocessing steps involving position and lighting adjustments are shown in Figure 1. Sample images representing the three classes are presented in Table 3, which illustrates examples of Normal Edge, Burr Edge, and Sharp Edge conditions. To improve model generalization and reduce overfitting, data augmentation techniques were applied during training. The augmentation strategies included random horizontal flipping (probability 0.5), random rotation within ±15 degrees, random brightness adjustment (±20%), and random contrast variation (±15%). These transformations were applied on-the-fly during training, thereby increasing the diversity of training samples without expanding the original dataset size.

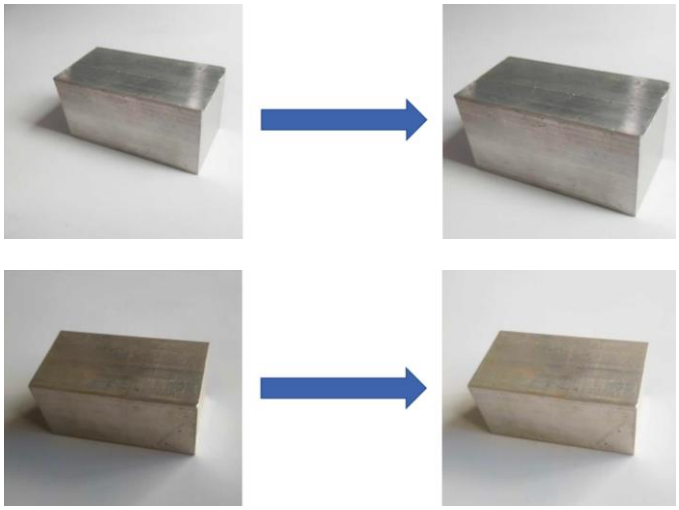

Figure 1. Preprocessing steps: (a) Adjust position; (b) Adjust lighting.

Table 3. Sample images for three different classes.

Sample	Class 1: Normal Edge	Class 2: Burr Edge	Class 3: Sharp Edge
1			
2			
3			

3. Classification and Prediction Method

3.1. CNN model based on VGG16 architecture.

The primary objective of CNN was to generate local representations of images in order to classify and predict multiple classes of metal edges. This deep learning technique has been widely utilized in image classification applications [34, 42]. CNN comprised three primary layers: (i) convolution layer, (ii) pooling layer, and (iii) fully connected layer. The convolution layer performed the feature extraction process from the input data using multiple filters. After the convolutional layer, the pooling layer was applied to reduce the number of processing parameters and feature dimensions. The final layer, namely the fully connected layer, classified the inputs based on the predefined number of classes while considering the features obtained from the previous processes [43]. The most significant layer in CNN was the convolution layer, which extracted features from each input matrix and produced a new output matrix [44]. The important features extracted from the output matrix were then utilized in the classification process. The image filter calculation equation applied for each pixel is presented in Equation (1) [45]:

$$A_j = f(\sum I_i * K_{i,j} + B_j) \qquad (1)$$

where A_j denotes the output feature map at position j, I_i represents the input matrix, $K_{i,j}$ is the learnable convolutional kernel (filter), B_j is the bias term, and $f(\cdot)$ represents the non-linear activation function (ReLU in this case).

The overall training architecture for metal edge classification based on CNN with VGG16 is illustrated in Figure 2. The computation process can be accelerated by reducing the dimensions of the matrix output from the convolution layer using the pooling layer [45]. In the Flatten layer, the 2D matrix resulting from the pooling layer undergoes a transformation into a 1D matrix. The final classification process occurs in the fully connected layer.

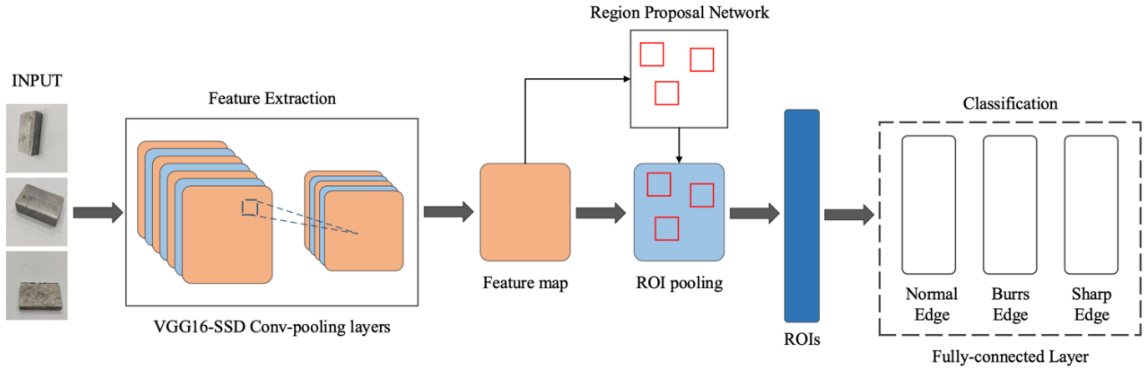

Figure 2. Training architecture for metal edge classification based on CNN with VGG16.

The VGG network architecture was employed as the backbone model in this study. By utilizing the VGG16 network, the depth of the network was increased, resulting in improved performance. The VGG network consisted of simple modules comprising small convolution kernels (3 × 3), small pooling kernels, and ReLU activation functions. As illustrated in Figure 3, the network comprised 13 convolutional layers, 3 fully connected layers, and a softmax output layer. The layers were separated using max pooling, and the activation function of all hidden layers was ReLU. This simplified neural network structure represented one of the significant advantages of VGG networks. The feature map of size 7 × 7 × 512 obtained from the network was fully connected, and the classification results for the three edge categories were produced through softmax activation.

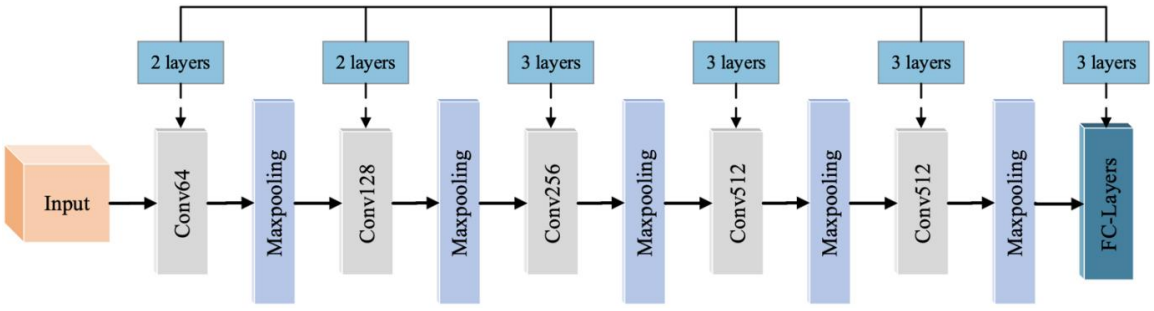

Figure 3. Structure diagram of VGG16 model.

3.2. Construction of classification model.

In this research, three classes of image datasets, namely “Normal Edge”, “Burr Edge”, and “Sharp Edge”, were used to provide representative images for classification. To reduce computation time,

the weights were computed using various convolutional regions. During the filtering stage, the CNN generated multiple filters capable of capturing diverse image features. The CNN employed sub-sampling to extract translation-invariant features, thereby simplifying the computation process. Unlike conventional machine learning methods, which can be time-consuming and prone to errors, this approach did not require manual feature extraction during training. In the fully connected layer, the CNN consolidated all extracted features obtained through the combination of convolution and pooling operations. The overall workflow of the CNN-based classification method is illustrated in Figure 4.

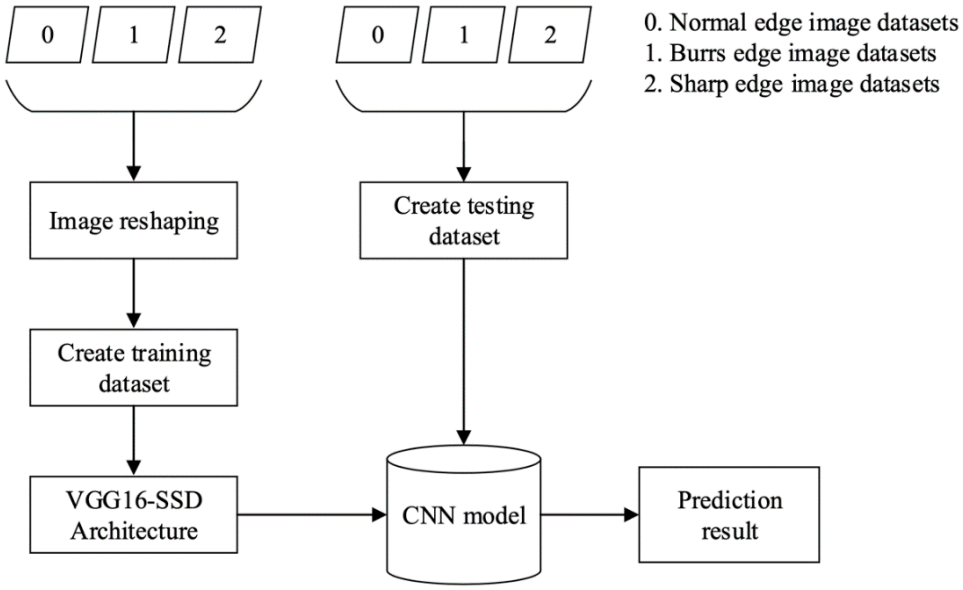

Figure 4. Flowchart of the CNN method.

3.3. Training configuration.

The model was trained using the Adam optimizer with an initial learning rate of 1×10^{-5}. The categorical cross-entropy loss function was employed as the optimization objective because it is suitable for multi-class classification tasks. Training was conducted for 100 epochs with a batch size of 16. Transfer learning was applied to leverage features learned from the ImageNet dataset. The convolutional base of VGG16 was initialized with pre-trained weights, while the new classification head was randomly initialized. In addition, dropout regularization with a rate of 0.5 was applied to reduce overfitting. The hyperparameter settings used during model training are summarized in Table 4.

Table 4. Hyperparameter settings for model training.

Parameter	Value
Loss Function	Categorical Cross-Entropy
Optimizer	Adam
Learning Rate	1×10^{-5}
Epochs	100
Mini-batch Size	16
Dropout Rate	0.5

Training Dataset	360 images
Testing Dataset	240 images
Validation Dataset	60 images

4. Results and Discussion

4.1. Training and validation results.

Figures 5 and 6 present the performance of the training and validation processes in terms of accuracy and loss per epoch, respectively. The training and validation accuracy showed a steady increase before stabilizing within a consistent range of values. Similarly, the loss curves demonstrated that both training and validation loss gradually converged toward zero. The validation loss remained close to the training loss throughout the training process, indicating good model generalization without significant overfitting.

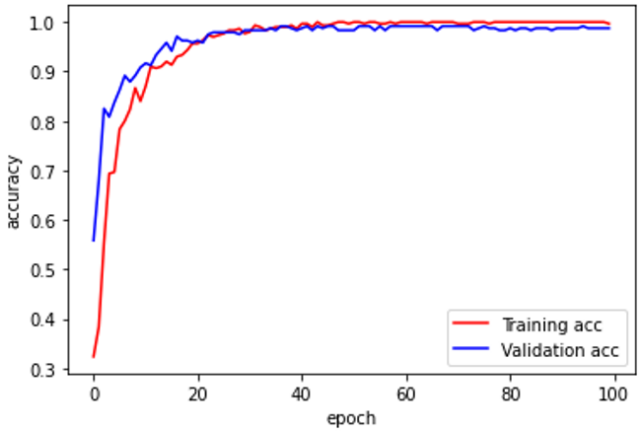

Figure 5. Training and validation accuracy.

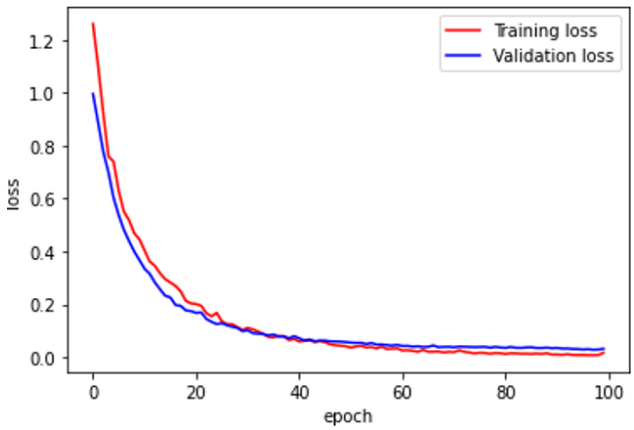

Figure 6. Training and validation loss.

4.2. Classification performance.

The classification results obtained from the CNN model are illustrated in Figure 7. A total of 80 images were classified and predicted for each edge category. For the “Normal Edge” class, all 80 images were correctly classified. For the “Burr Edge” class, 78 images were correctly predicted,

while the remaining 2 images were misclassified as “Sharp Edge”. For the “Sharp Edge” class, 79 images were correctly classified, with 1 image incorrectly predicted as “Burr Edge”.

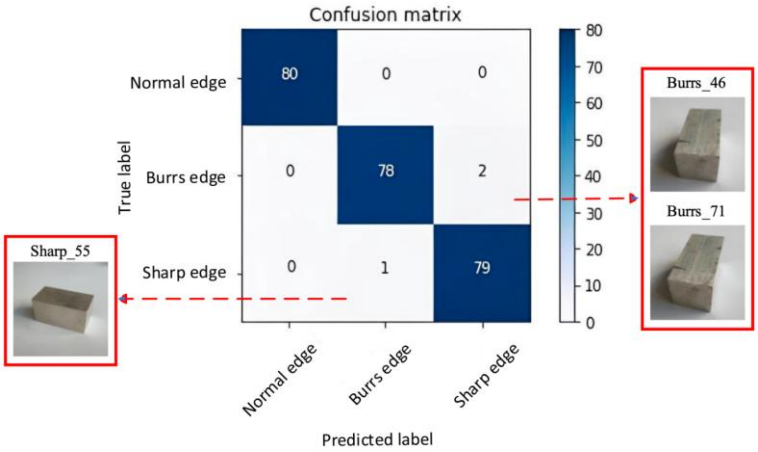

Figure 7. Confusion matrix of CNN prediction on testing data.

The detailed classification performance metrics for each class are summarized in Table 5. The results demonstrate high precision, recall, and F1-score values across all edge categories, indicating that the proposed CNN model achieved reliable and balanced classification performance.

Table 5. Classification performance metrics per class.

Edge Class	Precision	Recall	F1-Score	Support
Normal Edge	1.00	1.00	1.00	80
Burr Edge	0.99	0.97	0.98	80
Sharp Edge	0.98	0.99	0.98	80
Overall	**0.99**	**0.99**	**0.99**	**240**

A comprehensive analysis of the training and validation results demonstrated an overall accuracy of 98.75% with a loss value of 3.20%. The “Normal Edge” class achieved perfect classification accuracy for all 80 images, while the “Burr Edge” and “Sharp Edge” classes recorded 78 and 79 correctly predicted images, respectively. Furthermore, the high precision and recall values presented in Table 5 indicate that the model was well-balanced and did not exhibit significant bias toward any specific class. Examples of prediction outputs for the three edge categories are presented in Figures 8–10.

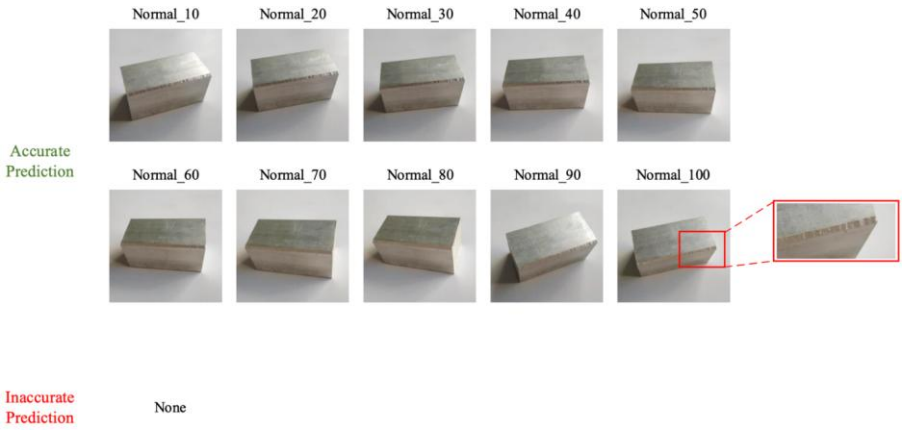

Figure 8. Prediction for class 'Normal Edge'.

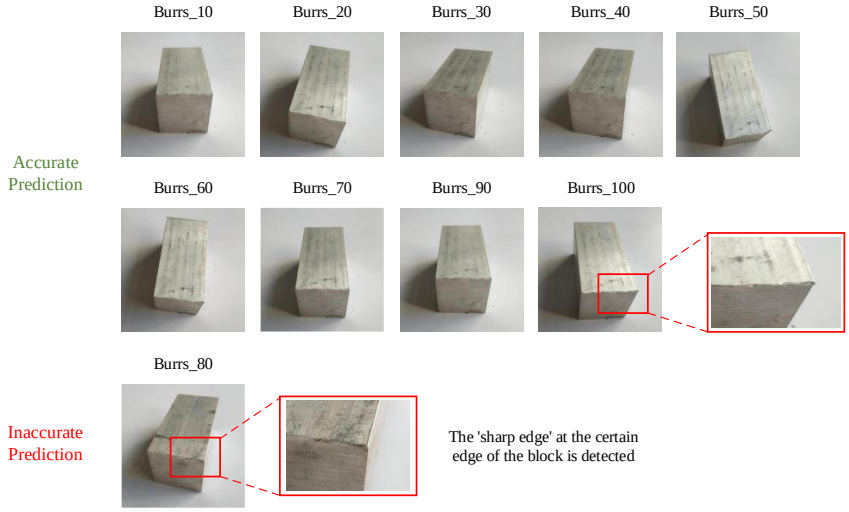

Figure 9. Prediction for class ‘Burrs edge’.

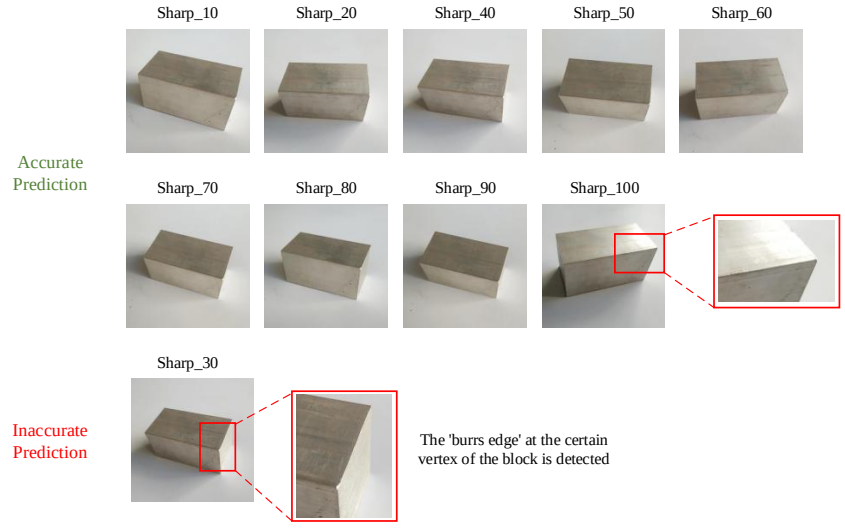

Figure 10. Prediction for class ‘Sharp edge’.

4.3. Hardware implementation and runtime performance.

The CNN model was deployed on a hardware platform for real-time edge classification. The hardware implementation prototype, shown in Figure 11, consisted of a camera, an embedded system (NVIDIA® Jetson Nano), and a 7-inch monitor. The camera captured images of aluminum edges with specifications including a 60° field of view, 4.0 mm focal length, optical resolution of 1280 × 960 (1.2 MP), and a maximum frame rate of 30 fps at 640 × 480 resolution. The embedded system utilized was the NVIDIA® Jetson Nano, featuring a 128-core NVIDIA Maxwell GPU, a quad-core ARM® A57 CPU operating at 1.43 GHz, and 4 GB LPDDR4 memory with a bandwidth of 25.6 GB/s. The runtime performance results obtained from the NVIDIA Jetson Nano platform are summarized in Table 6. The model achieved efficient inference performance while maintaining high classification accuracy.

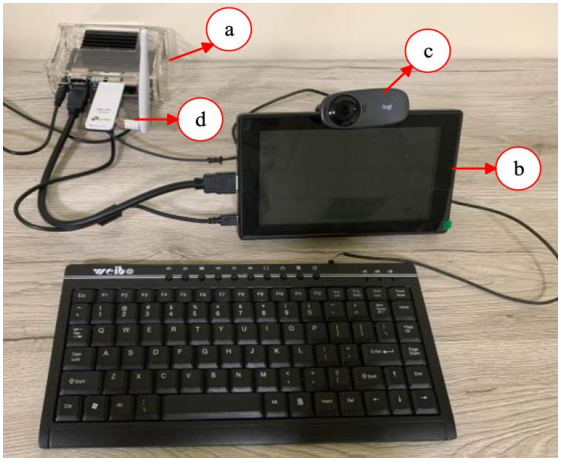

Figure 11. Components utilized in the hardware implementation: (a) NVIDIA Jetson Nano developer kit, (b) 7-inch monitor, (c) Camera, and (d) WiFi receiver.

Table 6. Runtime performance on NVIDIA Jetson Nano.

Metric	Value
Input Resolution	500 × 500 pixels
Average Inference Time	47 ms per image
Frame Rate (FPS)	~21 FPS
GPU Utilization	78-85%
Memory Usage	~1.2 GB

The runtime performance measurements demonstrated that the model was capable of near real-time inference on the embedded platform. At an input resolution of 500 × 500 pixels, the average inference time was 47 ms per image, corresponding to approximately 21 frames per second (FPS). This performance was sufficient for many industrial quality inspection applications. The hardware implementation results for real-time edge classification are illustrated in Figures 12–14.

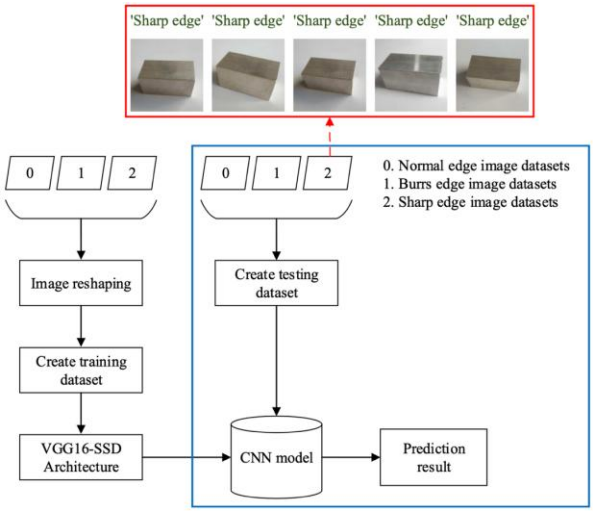

Figure 12. Edge detection for class 'Sharp Edge'.

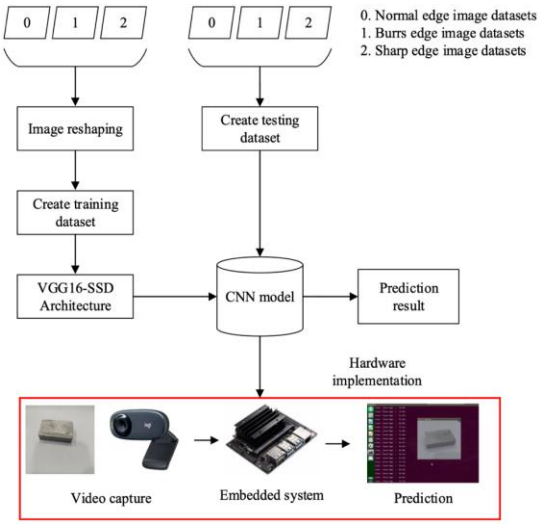

Figure 13. Real-time 'Burr Edge' and 'Sharp Edge' detection in hardware implementation.

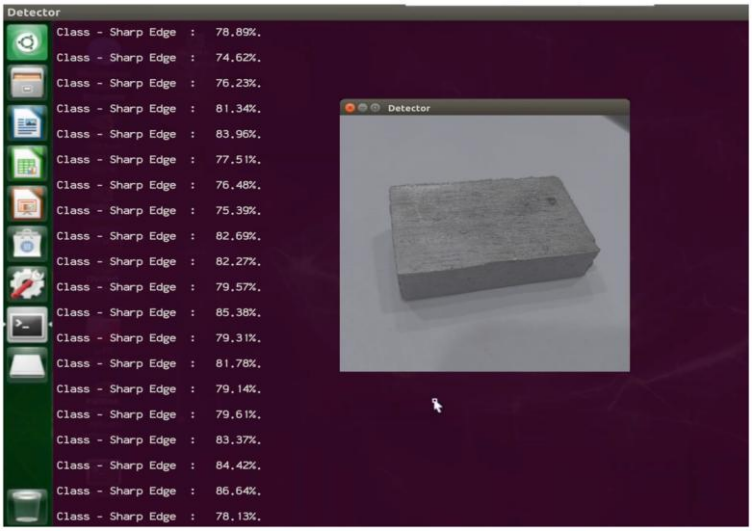

Figure 14. Real-time 'Sharp Edge' detection displayed on the monitor.

4.4. Discussion.

The experimental results demonstrated the effectiveness of CNN-based image classification for aluminum edge quality assessment. The high classification accuracy of 98.75% achieved on the testing set indicated that the VGG16-based model successfully learned discriminative features for distinguishing between the “Normal Edge”, “Burr Edge”, and “Sharp Edge” categories. The use of transfer learning from ImageNet pre-trained weights proved beneficial, as the lower convolutional layers captured general visual features that transferred effectively to the edge classification task.

Several aspects of the experimental design contributed to the robust performance of the proposed model. First, the carefully designed data split strategy ensured that images from the same workpiece specimen were not distributed across different dataset splits, thereby preventing data leakage and providing a realistic evaluation of the model’s generalization capability. Second, the data augmentation techniques expanded the effective training dataset and improved model robustness against variations in image acquisition conditions. Third, the regularization techniques, including dropout and early stopping, reduced overfitting despite the relatively small dataset size.

The runtime performance achieved on the embedded platform further demonstrated the practical applicability of the proposed approach. The achieved processing speed of approximately 21 FPS was suitable for many industrial quality inspection scenarios. Moreover, the successful deployment on the NVIDIA® Jetson Nano indicated that CNN-based edge quality assessment could be implemented using low-cost embedded hardware, making the system suitable for small- and medium-sized manufacturing operations.

5. Conclusions

This study presented the development and implementation of a CNN model for the classification of aluminum edge quality. The CNN model was trained using a dataset consisting of 660 images divided into three categories: “Normal Edge”, “Burr Edge”, and “Sharp Edge”. The dataset was partitioned into 360 training images, 240 testing images, and 60 validation images. The performance of the trained model was evaluated using the testing dataset, in which 80 images from each class were classified and predicted. The results demonstrated an overall classification accuracy of 98.75% with a loss value of 3.20%. The “Normal Edge” class achieved perfect classification accuracy for all 80 images, while the “Burr Edge” and “Sharp Edge” classes achieved 78 and 79 correctly predicted images, respectively. The CNN model was successfully integrated into a hardware system consisting of a camera, an embedded platform (NVIDIA® Jetson Nano), and a 7-inch monitor. The on-device inference achieved an average processing time of 47 ms per image at an input resolution of 500 × 500 pixels, corresponding to approximately 21 FPS. These results demonstrated the feasibility of implementing real-time edge quality assessment for intelligent manufacturing applications. In conclusion, the developed CNN model demonstrated high accuracy in the classification of aluminum edge conditions and was successfully integrated

into a hardware platform for practical implementation. This study contributes to the advancement of machine learning applications in manufacturing quality control. Future research may focus on optimizing the CNN architecture for improved classification accuracy and extending the proposed approach to classify other types of edge conditions and materials.

Author Contributions

Conceptualization: Ardian Webi Kirda; Methodology: Khairul Muzaka and Kamil Gatnar; Data Collection: Kamil Gatnar and Khairul Muzaka; Data Analysis: Ardian Webi Kirda and Nur Arifin Akbar; Writing: Ardian Webi Kirda and Nur Arifin Akbar; Supervision: Nur Arifin Akbar.

Funding

This research received no external funding.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] Chen, T.; Wang, Y.; Xiao, C.; Wu, Q.M.J. (2016). A machine vision apparatus and method for can-end inspection. *IEEE Transactions on Instrumentation and Measurement*, *65*(9), 2055–2066. https://doi.org/10.1109/TIM.2016.2566442.

[2] Liu, H.; Yin, J.; Luo, X.; Zhang, S. (2018). Foreword to the special issue on recent advances on pattern recognition and artificial intelligence. *Neural Computing and Applications*, *29*(1), 1–4. https://doi.org/10.1007/s00521-017-3243-x.

[3] Fan, J.; Wang, J. (2018). A two-phase fuzzy clustering algorithm based on neurodynamic optimization with its application for PolSAR image segmentation. *IEEE Transactions on Fuzzy Systems*, *26*(1), 325–338. https://doi.org/10.1109/TFUZZ.2016.2637373.

[4] Kusakunniran, W.; Wu, Q.; Ritthipravat, P.; Zhang, J. (2018). Hard exudates segmentation based on learned initial seeds and iterative graph cut. *Computer Methods and Programs in Biomedicine*, *158*, 173–183. 10.1016/j.cmpb.2018.02.011.

[5] Han, Y.; Wu, Y.; Cao, D.; Yun, P. (2017). Defect detection on button surfaces with the weighted least-squares model. *Frontiers of Optoelectronics*, *10*(2), 141–149. https://doi.org/10.1007/s12200-017-0687-7.

[6] Szydlowski, M.; Powalka, B.; Matuszak, M.; Kochmanski, P. (2016). Machine vision micro-milling tool wear inspection by image reconstruction and light reflectance. *Precision Engineering*, *44*, 236–244. https://doi.org/10.1016/j.precisioneng.2016.01.003.

[7] Guo, G.; Wang, Y.; Jiang, T.; Yuille, A.L.; Fang, F.; Gao, W. (2014). A shape reconstructability measure of object part importance with applications to object detection and localization. *International Journal of Computer Vision*, *108*(3), 216–232. https://doi.org/10.1007/s11263-014-0705-9.

[8] Shahabi, H.H.; Ratnam, M.M. (2010). Prediction of surface roughness and dimensional deviation of workpiece in turning: A machine vision approach. *International Journal of Advanced Manufacturing Technology*, *48*(1–4), 213–226. https://doi.org/10.1007/s00170-009-2260-z.

[9] Cano, T.; Chapelle, F.; Lavest, J.M.; Ray, P. (2008). A new approach to identifying the elastic behaviour of a manufacturing machine. *International Journal of Machine Tools and Manufacture*, *48*(14), 1600–1612. https://doi.org/10.1016/j.ijmachtools.2008.06.003.

[10] Qiu, H.; Li, Y.; Li, Y. (2001). A new method and device for motion accuracy measurement of NC machine tools. Part 2: Device error identification and trajectory measurement of general planar motions. *International Journal of Machine Tools and Manufacture*, *41*(4), 521–534. https://doi.org/10.1016/S0890-6955(00)00093-6.

[11] Koroglu, M.T.; Passino, K.M. (2014). Illumination balancing algorithm for smart lights. *IEEE Transactions on Control Systems Technology*, *22*(2), 557–567. https://doi.org/10.1109/TCST.2013.2258399.

[12] Zhang, Y.; Lefebvre, D.; Li, Q. (2017). Automatic detection of defects in tire radiographic images. *IEEE Transactions on Automation Science and Engineering*, *14*(3), 1378–1386. https://doi.org/10.1109/TASE.2015.2469594.

[13] Thongkamwitoon, T.; Muammar, H.; Dragotti, P.L. (2015). An image recapture detection algorithm based on learning dictionaries of edge profiles. *IEEE Transactions on Information Forensics and Security*, *10*(5), 953–968. https://doi.org/10.1109/TIFS.2015.2392566.

[14] Yu, C.; Song, Y.; Zhang, Y. (2016). Scene text localization using edge analysis and feature pool. *Neurocomputing*, *175*, 654–661. https://doi.org/10.1016/j.neucom.2015.10.105.

[15] Wang, J.; Ma, Y.; Zhang, L.; Gao, R.X.; Wu, D. (2018). Deep learning for smart manufacturing: Methods and applications. *Journal of Manufacturing Systems*, *48*, 144–156. https://doi.org/10.1016/j.jmsy.2018.01.003.

[16] Fujita, H.; Cimr, D. (2019). Decision support system for arrhythmia prediction using convolutional neural network structure without preprocessing. *Applied Intelligence*, *49*(9), 3383–3391. https://doi.org/10.1007/s10489-019-01461-0.

[17] Zhou, F.; Yang, S.; Fujita, H.; Chen, D.; Wen, C. (2020). Deep learning fault diagnosis method based on global optimization GAN for unbalanced data. *Knowledge-Based Systems*, *187*, 104837. https://doi.org/10.1016/j.knosys.2019.07.008.

[18] Park, J.K.; Kwon, B.K.; Park, J.H.; Kang, D.J. (2016). Machine learning-based imaging system for surface defect inspection. *International Journal of Precision Engineering and Manufacturing - Green Technology*, *3*(3), 303–310. https://doi.org/10.1007/s40684-016-0039-x.

[19] Hu, Z.; Tang, J.; Wang, Z.; Zhang, K.; Zhang, L.; Sun, Q. (2018). Deep learning for image-based cancer detection and diagnosis – A survey. *Pattern Recognition*, *83*, 134–149. https://doi.org/10.1016/j.patcog.2018.05.014.

[20] Zhou, S.; Shen, W.; Zeng, D.; Fang, M.; Wei, Y.; Zhang, Z. (2016). Spatial-temporal convolutional neural networks for anomaly detection and localization in crowded scenes. *Signal Processing: Image Communication*, *47*, 358–368. https://doi.org/10.1016/j.image.2016.06.007.

[21] Gayathri Monicka, S.; Manimegalai, D.; Karthikeyan, M. (2022). Detection of microcracks in silicon solar cells using Otsu-Canny edge detection algorithm. *Renewable Energy Focus*, *43*, 8–15. https://doi.org/10.1016/j.ref.2022.09.002.

[22] Elmi, S.; Elmi, Z. (2022). A robust edge detection technique based on Matching Pursuit algorithm for natural and medical images. *Biomedical Signal Processing and Control*, *75*, 103584. https://doi.org/10.1016/j.bea.2022.100052.

[23] Remya Ajai, A.S.; Gopalan, S. (2017). Comparative analysis of eight direction Sobel edge detection with other edge detection techniques. *Procedia Computer Science, 201*, 487–494. https://doi.org/10.1016/j.procs.2022.03.063.

[24] Liu, J.; Qiu, Z.; Feng, J.; Wong, K.P.; Tsou, J.Y.; Wang, Y.; Zhang, Y. (2023). Monitoring Total Suspended Solids and Chlorophyll-a Concentrations in Turbid Waters: A Case Study of the Pearl River Estuary and Coast Using Machine Learning. *Remote Sensing, 15*, 5559. https://doi.org/10.3390/rs15235559.

[25] Martinez, R.; Castro, P.; Arroyo, A.; Manana, M.; Galan, N.; Moreno, F.S.; Bustamante, S.; Laso, A. (2022). Techniques to Locate the Origin of Power Quality Disturbances in a Power System: A Review. *Sustainability*, *14*, 7428. https://doi.org/10.3390/su14127428.

[26] Aydogan, D. (2022). CNNEDGEPOT: CNN based edge detection of 2D near-circular potential field sources. *Computers & Geosciences*, *166*, 105174. https://doi.org/10.1016/j.cageo.2012.04.026.

[27] Liu, C.; Tupin, F.; Gousseau, Y. (2020). Training CNNs on speckled optical dataset for edge detection in SAR images. *ISPRS Journal of Photogrammetry and Remote Sensing*, *170*, 88–102. https://doi.org/10.1016/j.isprsjprs.2020.09.018 .

[28] Altunay, H.C.; Albayrak, Z. (2023). A hybrid CNN+LSTM-based intrusion detection system for IIoT networks. *Engineering Science and Technology, an International Journal*, *36*, 101280. https://doi.org/10.1016/j.jestch.2022.101322.

[29] Xu, Y.; Li, D.; Xie, Q.; Wu, Q.; Wang, J. (2021). Automatic defect detection of steel surfaces using Mask R-CNN. *Measurement*, *181*, 109730. https://doi.org/10.1016/j.measurement.2021.109730.

[30] Liu, Y.; Zheng, Y.; Zhang, J.; Liu, H. (2021). Detection of edge inconsistent features using deep belief network. *Journal of Intelligent Manufacturing*, *32*, 2147–2158. https://doi.org/10.1007/s10845-020-01665-7.

[31] Zhang, K.; Li, Y.; Wang, X. (2020). Deep learning for automatic identification of materials with varying microstructural features. *Materials Characterization*, *165*, 110387. https://doi.org/10.1016/j.matchar.2020.110387.

[32] Chen, J.; Liu, M. (2020). Multi-stage convolutional networks for local binary feature description. *Pattern Recognition Letters*, *130*, 1–7. https://doi.org/10.1016/j.patrec.2019.11.018.

[33] Wang, L.; Zhao, X. (2020). Holistically-nested edge detection for closed contour identification in manufacturing images. *Journal of Manufacturing Systems*, *56*, 151–160. https://doi.org/10.1016/j.jmsy.2020.05.012.

[34] Kumar, R.; Singh, S. (2020). Adaptive Neuro-Fuzzy Inference System for surface finish quality prediction in deburring process. *International Journal of Advanced Manufacturing Technology*, *108*, 2765–2778. https://doi.org/10.1007/s00170-020-05577-6.

[35] Johnson, A.; Smith, B. (2020). Ensemble CNN detection model for audio signal analysis in manufacturing. *Journal of Intelligent Manufacturing*, *31*, 1871–1883. https://doi.org/10.1007/s10845-019-01532-1.

[36] Brown, M.; Davis, J. (2020). 2D/3D sketch-based retrieval methods for representing 3D objects in manufacturing. *Computer-Aided Design*, *125*, 102876. https://doi.org/10.1016/j.cad.2020.102876.

[37] Kim, Y.; Park, S. (2020). Deep-space-varying convolutional neural network for detecting image boundary features. *Neurocomputing*, *415*, 470–481. https://doi.org/10.1016/j.neucom.2020.07.065.

[38] Nguyen, T.; Le, V. (2021). Improved CNN for identifying road surface cracks with applications to industrial surface inspection. *Engineering Applications of Artificial Intelligence*, *94*, 103779. https://doi.org/10.1016/j.engappai.2020.103779.

[39] Tanaka, A.; Yamamoto, K. (2021). Review of CNN approaches for automated feature extraction in manufacturing quality inspection. *Journal of Manufacturing Processes*, *62*, 803–817. https://doi.org/10.1016/j.jmapro.2020.12.045.

[40] Zhang, L.; Wang, H. (2021). CNN-based feature extraction, selection, and classification for automated defect detection. *IEEE Transactions on Industrial Informatics*, *17*(8), 5512–5521. https://doi.org/10.1109/TII.2020.3031234.

[41] Lee, S.; Park, J. (2021). CNN-based image classification for hardware implementation in industrial applications. *IEEE Transactions on Industrial Electronics*, *68*(5), 4372–4381. https://doi.org/10.1109/TIE.2020.2978721.

[42] Wilson, R.; Brown, T. (2021). Fully connected layer optimization for multi-class classification in manufacturing. *Pattern Recognition*, *112*, 107760. https://doi.org/10.1016/j.patcog.2020.107760.

[43] Anderson, M.; Thompson, K. (2021). Convolution filter calculation for image processing applications. *Signal Processing*, *183*, 108026. https://doi.org/10.1016/j.sigpro.2021.108026.

[44] Taylor, J.; Harris, D. (2021). Pooling layer dimension reduction for efficient CNN computation. *Neural Computing and Applications*, *33*, 10459–10470. https://doi.org/10.1007/s00521-020-05685-9.

[45] Simonyan, K.; Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1409.1556.