

Application of Convolutional Neural Networks for Binary Recognition Task of Two Similar Industrial Machining Parts

Hadyan Hafizh^{1*}, Amir Hamzah Abdul Rasib²

Rohana Abdullah³, Mohd Hadzley Abu Bakar⁴, Anuar Mohamed Kassim⁵

Faculty of Mechanical and Manufacturing Engineering Technology^{1, 2, 3, 4}

Universiti Teknikal Malaysia Melaka, 76100 Durian Tunggal, Melaka, Malaysia^{1, 2, 3, 4}

Faculty of Electrical Engineering, Universiti Teknikal Malaysia Melaka, 76100 Durian Tunggal, Melaka, Malaysia⁵

Abstract—Misclassifying parts in the small-medium manufacturing enterprise can lead to serious consequences. Manual inspection, as currently practiced, allows for compromises in product traceability. Due to this condition, inspection of the part's number is not digitally visible. Due to a lack of modern traceability, customers receive incorrect parts, and the same incidents continue to occur. It is essential to transform manual inspections into digital and automated ones. AI-based technologies have recently been employed to enable a smart and intelligent recognition system for industrial machining parts. Convolutional Neural Networks (CNN) are widely used for image recognition tasks and are gaining popularity as deep learning algorithms. In this paper, a CNN model is used to perform binary recognition on two similar industrial machining parts. The model has been trained to recognise two classes of machining parts: Parts A and B. The dataset used to train the model includes both original and augmented images, with a total of 2447 images for both classes. The performance metrics have been measured during the training process, and 10 experiments have been conducted to evaluate the performance of the model. The test results reveal that the CNN model achieves 98% mean accuracy, 97.1% precision for Part A, 99% precision for part B and 0.982 AUC value. The results demonstrate the effectiveness of the CNN-based recognition of parts. It offers an effective alternative and is a compelling method for quality assurance in small-medium manufacturing enterprises.

Keywords—Convolutional neural networks; binary recognition; machining parts; deep learning

I. INTRODUCTION

Computer vision and automation are now being used by modern industrial firms to achieve higher quality and more accurate inspection of parts. Deep learning, for example, is an AI-based technology that assists the industry regarding automation with minimal human intervention. The convolutional neural network (CNN) is one of the most important deep-learning-based computer vision methods. CNN has a wide range of applications. One of the most popular uses of CNN is image classification and defect detection in industrial products. Zhang et al. [1] used CNN to study defects detection for aluminium alloy in robotic arc welding. In this study, data augmentation and noise addition have been applied to boost the CNN dataset. It was found that the CNN model was able to attain a 99.38% accuracy. Westphal et al. [2]

employed CNN to detect irregularities in selective laser sintering (SLS). Two transfer learning CNN models were used with pre-trained weights to classify good and defective images during the manufacturing of parts. The VGG16 transfer learning CNN model achieved the best results with 95.8% accuracy. Furthermore, a study on weldment classification using a vision system by Bacioiu et al. [3] had reported that the CNN model can achieve the highest accuracies of 71%, 89% and 95% for 6-class, 4-class and 2-class, respectively. A similar study utilising the SS304 TIG welding process had summarised that CNN is capable of learning powerful representations of welding defects [4]. Approximately 89.5% of accuracy was reported in classifying good vs defective welds with the use of CNN.

The application of CNN for the inspection of metal additive manufacturing parts has been examined by Cui et al. [5]. In this study, the regularisation and dropout layers were added to the CNN architecture in order to avoid overfitting problems. With the help of data augmentation, about 92.1% of accuracy was obtained for the CNN model. A similar study was also conducted by Xiaoming et al. [6] where CNN was utilised to detect defects in metallic surfaces. Data augmentation was further applied to enrich the training data. The study also contributed to the new dataset, called GC10-DET, for metallic surface defect detection. By using this dataset and the CNN model, the proposed method successfully met the accuracy requirements for the detection of metallic defects. The application of CNN to metal manufacturing parts was also conducted by Ma et al. [7]. Four CNN models were utilised in this study to detect the weld defects of galvanised steel sheets. The study had found that the VGG16 transfer learning CNN model combined with the data augmentation method made the best model to achieve a state-of-the-art performance in detecting weld defects.

Due to high classification abilities, CNN is gaining attention in various industrial fields, particularly in metal and welding defect detection. CNN has proven to be effective in performing recognition and classification tasks [8-11]. CNN was also used in casting applications to detect and investigate the defects of casting products. A study conducted by Mery et al. [12] had used synthetic defects in order to improve the performance of the CNN model. This study proposed a CNN

*Corresponding Author

architecture called Xnet-II which has 30 layers and more than 1,350,000 parameters. Another study conducted by Jiang et al. [13] employed X-ray images of casting products as inputs to the CNN model. The test accuracy achieved was reported to reach up to 95.5%. Various defects in the casting product, such as blowholes, chipping, cracks and wash automatically, were investigated by Nguyen et al. [14]. The study used 6000 images with 768×768 px resolution as input to the CNN model. The training model was reported to attain an experimental accuracy of more than 98%.

Although previous researchers have made significant efforts in detecting defects in industrial products using the CNN model [15-20], little attention has been paid to recognise similar industrial machining parts. The issue that the human operator faces on the manufacturing floor is not only related to defects, but also to misclassification of machining components. This problem arises due to the similarity of two machining parts, and when handled by a human operator, it leaves room for human error. Misclassification of machining parts is a real issue that occurs on the manufacturing floor. Due to lack of modern traceability, incorrect parts have been delivered to customers and the same incident is repeatedly occurring. The company's image becomes tarnished, decreasing its reputation in the eyes of existing and potential customers and vendors. It is an urgent matter to transform manual inspections into digital and automated ones. Therefore, the current work proposes a CNN model to recognise and classify two similar machining parts. The proposed CNN model can be integrated into a machine-vision system and perform automatic recognition tasks.

II. METHODOLOGY

The dataset used for training the model and testing the results is described in this section. Subsequently, the process of data augmentation is discussed in order to enhance the performance of the CNN model. The CNN model used in this work is also presented and discussed.

A. Data Structure

The original images of the machining parts dataset were captured by using an android-based smartphone, with a resolution of 750×1000 px. A total of 160 images were taken for both Parts A and B, with each part consisting of 80 images. These images are then resized to 224×224 px before being fed into the CNN model as input. Fig. 1 presents a sample of the resized original images taken with an android-based smartphone for both Parts A and B. It can be seen that these two parts are similar. There is a high possibility that these two parts will be misidentified by a human operator.

The original images were then used to generate an augmented dataset in order to improve the CNN model's performance. By performing various augmentation processes such as rotation, translation, zoom and brightness adjustment, a total of 2317 augmented images were generated. The original and augmented images have been combined to produce a total of 2477 images, 1234 of which belongs to Part A and 1243 to Part B. Among 1234 images from Part A, 980 images were used to create the training dataset and the remaining 254 were used to create the test dataset. As for Part B, 987 images were

used to create the training dataset, while 256 images were used to create the test dataset. A balanced dataset was used to train the CNN model, with nearly equal numbers of images for training and testing of both classes. The data structure applied in the current work is presented in Fig. 2.

B. Data Augmentation

Image data augmentation is an alternative method to expand a training dataset by creating new versions of images. It can improve the performance of deep learning models by creating variations of the images they learn. Data augmentation is a regulatory mechanism designed to prevent model overfitting. This procedure works by performing the following operations, as shown in Table I.

The augmented dataset is generated by randomly selecting images from the original dataset. The process shown in Table I is then applied to generate a total of 2317 augmented images. In order to create the augmented dataset, four processes were applied to the original images. These processes have been selected based on the common scenario encountered on the manufacturing floor when performing the recognition task. The images can be arbitrarily placed under the camera before performing the recognition task; therefore, the rotation and translation processes are applied to generate a series of augmented images with random placement. Furthermore, the camera's zoom and brightness can be adjusted. As a result, the augmented dataset with different zoom and brightness settings is essential for training the model.

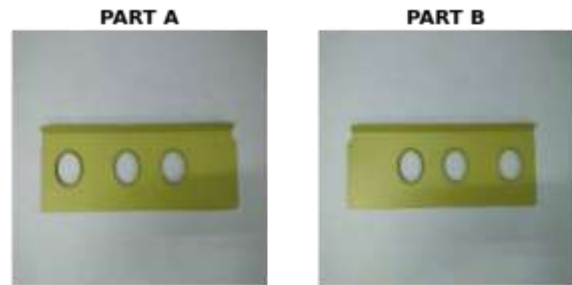


Fig. 1. Sample of Original Images.

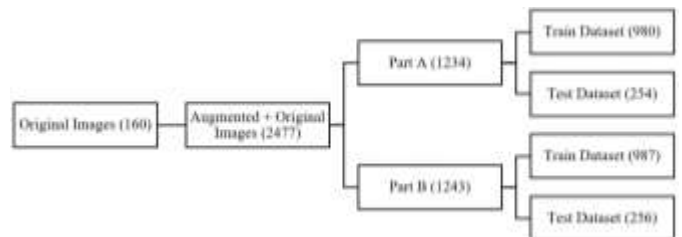


Fig. 2. Data Structure Employed in the Current Work.

TABLE I. PROCESS APPLIED IN DATA AUGMENTATION

Process	Description
Rotation	Randomly rotate image in the range of 00 to 450
Translation	Randomly shift image horizontally and vertically in the range of 0 to 0.1 (as a fraction of total width and height)
Zoom	Randomly zoom the image in the range of 0 to 0.2
Brightness	Randomly adjust image brightness in the range of 0.1 to 1.0

C. Convolutional Neural Network

A convolutional neural network (CNN) is a deep learning algorithm that is widely utilised in the area of image recognition. CNN can be regarded as a special type of feed-forward neural network in AI technology. CNN's main advantage over its predecessors is that it automatically detects significant features without the need for human intervention, making it the most widely used [21, 22]. As in a standard multi-layer neural network, a CNN has at least one convolutional layer followed by at least one fully connected layer. The CNN architecture applied in this paper consists of three convolutional layers and two fully connected layers.

The proposed CNN architecture is based on the LeCun model [23]. The model consists of three convolutional layers followed by fully connected layers, as illustrated in Fig. 3. The machining parts image was captured with a resolution of 750×1000 px. Before feeding the original and augmented images into the CNN model, they were resized to 224×224 px. These images were then transformed into grey-scale and with the dimension of $224 \times 224 \times 1$. The grey-scale images were then passed through a block of convolution layers with a kernel size of 3×3 and a stride of 1 px.

In the convolutional layers, the number of output filters was set to 8, 16 and 32, respectively. Following the convolutional layers, three max-pooling layers with window sizes of 2×2 and strides of 2 px were added to compress the spatial representation of the input data [24]. Furthermore, the Rectified Linear Unit (ReLU) function was used as the activation function in the convolutional layer.

The fully connected layer is the primary building block of traditional artificial neural networks. It converts the high-level filtered machining parts image into votes. This layer's primary goal is to perform classification using the features extracted by the convolutional layers. Because the current work's class is binary, the model must only choose between two classes, Parts A and B. Due to the flattening process, the input is treated as a single list. The flattened layer is 1×25088 in size. The flattened output is then fed to a feed-forward neural network. The backpropagation algorithm is applied to every iteration of

training in the dense layer. In the last layer of the CNN model, the sigmoid activation function was used to estimate the probability of the sample belonging to each class.

D. Experimental Procedure

A series of numerical experiments were conducted following the procedure depicted in Fig. 4. The first step in the process is to collect true label data, which was accomplished by taking 160 images of Parts A and B with an android-based smartphone. The original images were then randomly selected to generate an augmented dataset. This process yields a database of machining part images, which are saved for later use to train the CNN model. Having a sufficient dataset, a CNN model can then be developed. The architecture of the CNN model is shown in Fig. 3.

The model is trained by using the augmented and original dataset until the accuracy achieves a value of more than 95%. Subsequently, a series of numerical experiments are performed. The model was run 10 times and its performance was measured. The loss and accuracy values per epoch during the training and testing were also measured. A confusion matrix was further computed for each numerical experiment in order to measure the performance of the CNN model. The model was then applied to perform recognition and prediction tasks using a random image from the test dataset. Finally, the model was saved, and the experiment was successfully completed.

The training and recognition tasks were repeated ten times. The performance of the CNN model was measured and visualised in the form of a confusion matrix for each training process. The accuracy, precision, sensitivity and specificity of the model can be calculated from this matrix using the following equations [25]:

$$\text{Accuracy} = \frac{TN+TP}{TN+FP+TP+FN} \quad (1)$$

$$\text{PPV} = \frac{TP}{TP+FP} \quad (2)$$

$$\text{NPV} = \frac{TN}{TN+FN} \quad (3)$$

$$\text{Sensitivity} = \frac{TP}{TP+FN} \quad (4)$$

$$\text{Specificity} = \frac{TN}{TN+FP} \quad (5)$$

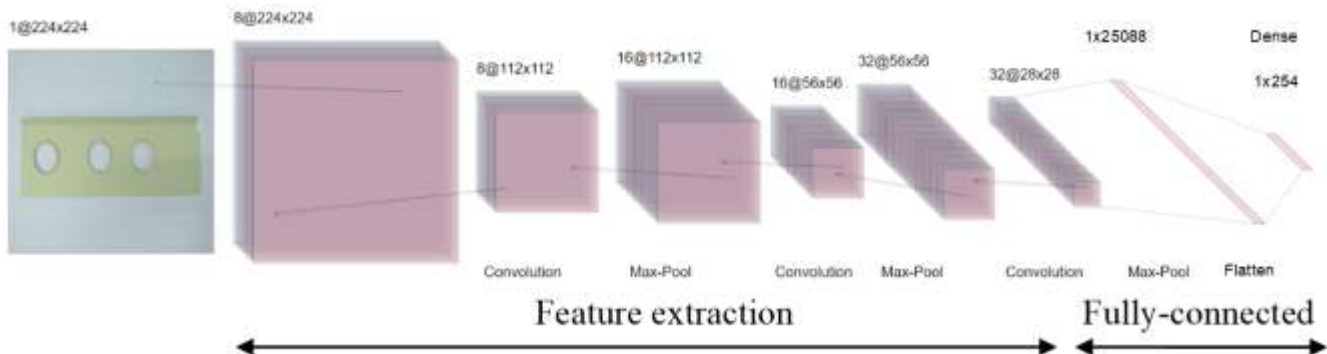


Fig. 3. Architecture of the CNN Model.

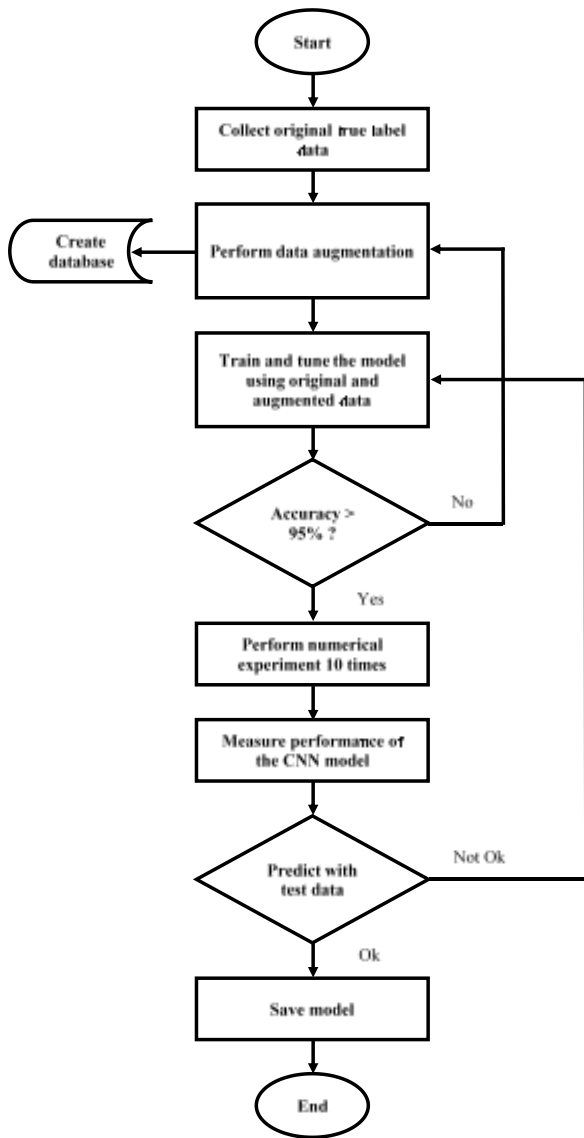


Fig. 4. Flow Chart of the Experimental Procedure.

where TP, TN, FP and FN are true positive, true negative, false positive and false negative, respectively. The TP value in the confusion matrix represents the number in which the CNN model has predicted as Part A and the true label is actually Part A. The TN value represents the number in which the CNN model has predicted as Part B and the true label is actually Part B. The FP value means the number in which the CNN model has predicted as Part B, but the true label is actually Part A. Lastly, the FN value means the number in which the CNN model has predicted as Part A, but the true label is actually Part B.

Precision values for Parts A and B are also referred to as Positive Predictive Value (PPV) and Negative Predictive Value (NPV), respectively. The PPV value counts the number of observations that are predicted to be positive (Part A) and are, in fact, positive. Similarly, the NPV value indicates how many predictions are correct out of all negative predictions (Part B). Furthermore, the Receiver Operating Characteristics (ROC)

curve and the Area Under the Curve (AUC) value are also measured during the training and testing processes.

III. RESULTS AND DISCUSSION

A. Binary Class Test

The machining parts dataset contains original and augmented images. By using this combined dataset, the experiment was conducted, and the recognition task was performed for two classes of machining parts, i.e., Parts A and B. For each experiment, the training dataset was shuffled but the random seed parameter was kept constant to ensure all algorithms used the same samples as the testing and training data. The test dataset utilised in the current work was never used to train the model, therefore it represents new data for the trained model. After the training process, a recognition task was performed by randomly selecting the image from the test dataset for both classes. The recognition task was conducted 10 times and the results are presented in Fig. 5. The true label and the prediction results are visualised on the images. From this figure, the CNN model correctly recognised all of the test images. These images are randomly rotated and translated to simulate the real-world situation in which a human operator attempts to perform a recognition task by placing machining parts under the sensor. As discussed in the data augmentation section, the brightness and zoom level of the test images were also randomly assigned within the prescribed range.

B. Performance Measures

The CNN model's performance metrics were measured using the Confusion Matrix. Equations (1–5) were used to calculate the Accuracy, Precision, Sensitivity and Specificity values from the Confusion Matrix. The results are displayed in Table I. The individual experiment has been considered. The first and second experiments achieved accuracy values of 0.992 and 0.986, respectively. When compared to the first experiment, the precision values (PPV and NPV) in the second experiment are lower. Despite the fact that the accuracy values for both experiments are similar, the second experiment has a higher false negative value (4 Parts A are wrongly recognised as Part B). In consequence, the precision value of Part B (0.984) in the second experiment is lower than in the first.

The third and fourth experiments have the same accuracy (0.984), but they are smaller in magnitude than the former experiments. The CNN model, in particular, was able to correctly recognise all the images in Part A in the fourth experiment. As a result, the precision value of Part B is 1. Although the CNN model performed well with the image of Part A, there are 8 images of Part B that were incorrectly identified as Part A. When compared to the third experiment, this condition leads to a lower precision value for Part A. The fifth through tenth experiments showed a fluctuation in the accuracy value.

The CNN model achieved the lowest accuracy in the tenth experiment. In this experiment, 14 Parts B were incorrectly identified as Part A, while 6 Parts A were incorrectly identified as Part B. Although the instantaneous accuracy values fluctuated, the CNN model was able to achieve 0.980 mean accuracy across 10 experiments.

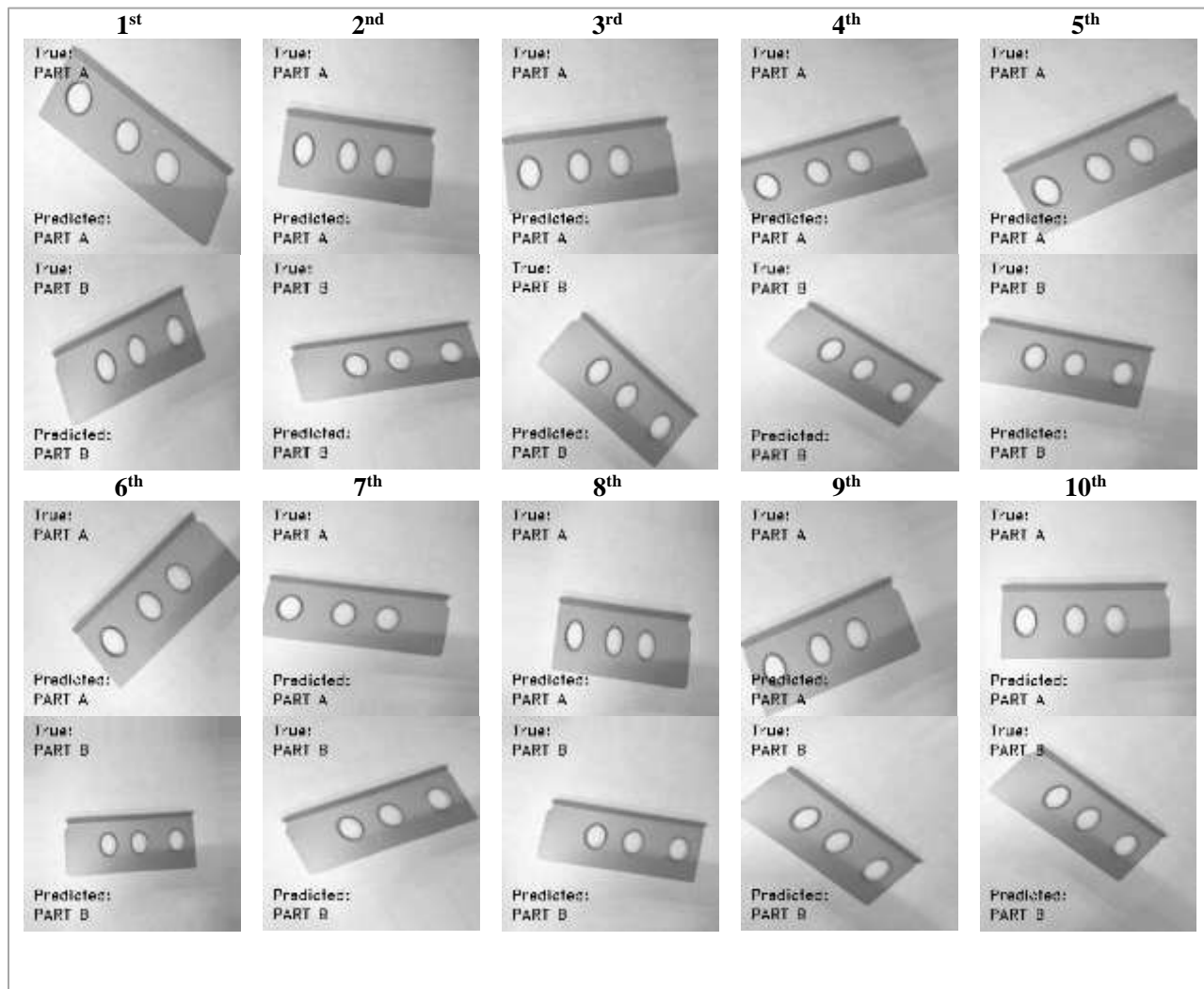


Fig. 5. Recognition Results of the CNN Model for 10 Experiments.

Table II reveals an intriguing pattern: the number of images of Part B that were incorrectly classified as Part A is significantly higher than the other way around. In other words, the precision value for Part A is lower than the precision value for Part B. This condition indicates that the Part A training dataset should be improved.

Enhancing the augmentation process, such as increasing the variation of image rotation and translation, can result to an improvement. Furthermore, the ROC curves and AUC values can be calculated using the sensitivity and specificity values shown in Table II. The ROC curve is a plot of the True Positive Rate (TPR) versus the False Positive Rate (FPR) for each possible prediction threshold. TPR is another name for sensitivity, and FPR can be calculated from $1 - \text{specificity}$. The ROC curve illustrates the trade-off between correctly recognised positive samples (Part A) and misclassified negative samples (Part B).

The instantaneous AUC value obtained was 0.982. A perfect CNN model has an AUC value equal to one, indicating that it has a high level of separability. This is illustrated by a

solid line (perfect classifier) in Fig. 6. A poor CNN model has an AUC value close to zero, indicating that it has the worst measure of separability. When the AUC value is 0.5, the model has no class separation capacity at all (random classifier), as indicated by the linear dashed line in Fig. 6. Based on the computed AUC value, the current model is considered to perform well in recognising and classifying two similar machining parts.

Fig. 7 shows the evolution of loss and accuracy values during the training process. The instantaneous value of loss and accuracy were recorded for 30 epochs. Both loss and accuracy diagrams characterise the training process. They provide initial information regarding the effectiveness of the selected hyperparameters. The current work uses binary cross-entropy as a loss function since it is widely employed for binary classification tasks [26]. Furthermore, the Adaptive Moment Estimation (Adam) algorithm was applied for the optimisation process and the learning rate was set to 0.001. The number of epochs and the batch size of the CNN model were set to 30 and 32 respectively.

TABLE II. PERFORMANCE METRICS OF CNN MODEL

Experiment	Confusion Matrix		Accuracy	Precision		Sensitivity	Specificity
				PPV	NPV		
1 st	253	1	0.992	0.988		0.996	0.988
	3	253			0.996		
2 nd	250	4	0.986	0.988		0.984	0.988
	3	253			0.984		
3 rd	252	2	0.984	0.977		0.992	0.977
	6	250			0.992		
4 th	254	0	0.984	0.969		1.000	0.969
	8	248			1.000		
5 th	253	1	0.990	0.984		0.996	0.984
	4	252			0.996		
6 th	251	3	0.978	0.969		0.988	0.969
	8	248			0.988		
7 th	250	4	0.973	0.962		0.984	0.961
	10	246			0.984		
8 th	253	1	0.980	0.966		0.996	0.965
	9	247			0.996		
9 th	251	3	0.973	0.958		0.988	0.957
	11	245			0.988		
10 th	248	6	0.961	0.947		0.976	0.945
	14	242			0.976		
Mean			0.980	0.971	0.990	0.990	0.970

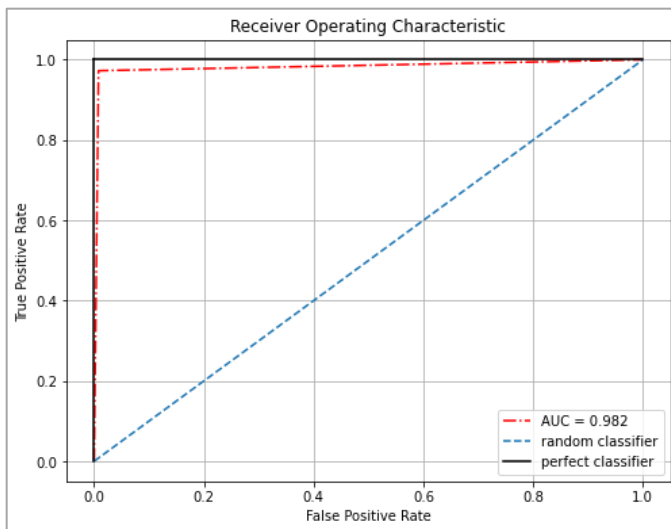


Fig. 6. ROC Curve and AUC Values.

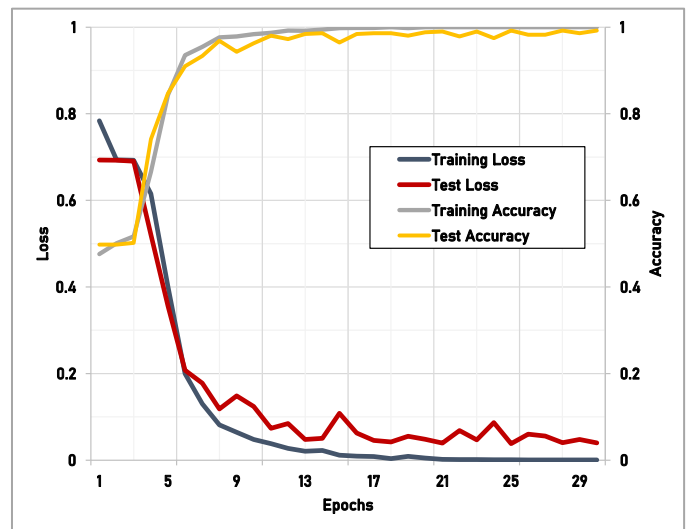


Fig. 7. Loss and Accuracy Evolution during Training.

From Fig. 7, it can be seen that there is a slight gap between training and test loss. This indicates an unrepresentative dataset, which means that the training dataset used to train the CNN model does not provide sufficient information to learn the recognition problem [27]. This situation is consistent with the condition of the precision value obtained in Table II. It can be observed that the precision of Part A is lower compared to Part B. This suggests that the training dataset of Part A is moderately unrepresentative.

In Fig. 7, the accuracy has been recorded for one experiment and plotted over 30 epochs. The accuracy values for ten experiments are shown in Fig. 8, and the mean value has been calculated accordingly. The accuracy values ranged between 96% to 99%. The mean accuracy calculated from the instantaneous value was found to be 98%. This indicates that the CNN model has good recognition performance and can provide an alternative method for manual inspection of industrial machining parts.

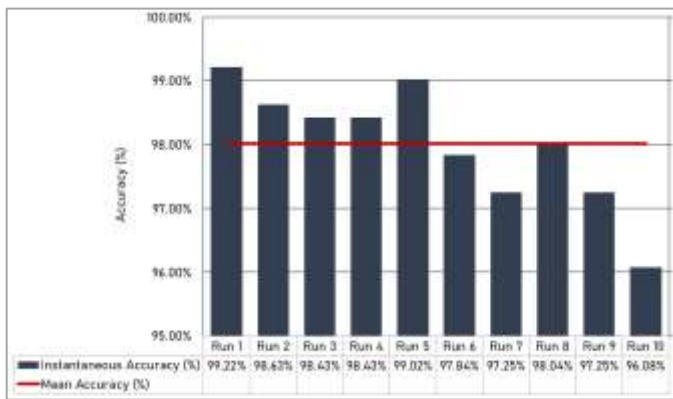


Fig. 8. Instantaneous and mean Accuracy for 10 Experiments.

IV. CONCLUSION

In this paper, a CNN model was employed to perform binary classification tasks of two similar machining parts. The model consists of three convolutional layers and three max-pooling layers for feature extraction, followed by two fully connected layers for recognition and classification. Two classes, Parts A and B, have been assigned for the recognition task. The dataset used to train the model consists of 160 original images and 2317 augmented images. Four types of data augmentation processes were applied in order to improve the performance of the model. Rotation, translation, zooming and brightness adjustments were all part of the augmentation process. These images were then assigned to one of two classes: Part A (1234 images for training and 254 for testing) and Part B (1243 images for training and 256 for testing).

The model was run 10 times, and the performance metrics in the form of loss and accuracy values were measured for each experiment. The confusion matrix was also recorded, as well as the model's accuracy, precision, sensitivity, specificity and AUC value. Experiment results show that the CNN model achieved a mean accuracy of 98%. The test outcomes also show that the mean precision values for Parts A and B are 0.971 and 0.99, respectively. The instantaneous ROC curve and AUC value (0.982) indicate that the current CNN model performs well in recognising and classifying two similar machining parts. The results further demonstrate the effectiveness of part recognition based on CNN. It offers a compelling alternative to replace manual inspections currently practiced in small-medium manufacturing enterprises. In the future, the CNN model's results should be compared to those of other well-known CNN architectures, such as MobileNet, ResNet50, and VGG16, to investigate their performance in recognising two similar machining parts.

REFERENCES

- [1] Z. Zhang, G. Wen, and S. Chen, "Weld image deep learning-based on-line defects detection using convolutional neural networks for Al alloy in robotic arc welding," *J. Manuf. Process.*, vol. 45, no. March, pp. 208–216, 2019.
- [2] E. Westphal and H. Seitz, "A machine learning method for defect detection and visualization in selective laser sintering based on convolutional neural networks," *Addit. Manuf.*, vol. 41, no. November 2020, p. 101965, 2021.
- [3] D. Bacioiu, G. Melton, M. Papaalias, and R. Shaw, "Automated defect classification of Aluminium 5083 TIG welding using HDR camera and

- neural networks," *J. Manuf. Process.*, vol. 45, no. August, pp. 603–613, 2019.
- [4] D. Bacioiu, G. Melton, M. Papaalias, and R. Shaw, "Automated defect classification of SS304 TIG welding process using visible spectrum camera and machine learning," *NDT E Int.*, vol. 107, no. November 2018, 2019.
- [5] W. Cui, Y. Zhang, X. Zhang, L. Li, and F. Liou, "Metal additive manufacturing parts inspection using convolutional neural network," *Appl. Sci.*, vol. 10, no. 2, 2020.
- [6] X. Lv, F. Duan, J. J. Jiang, X. Fu, and L. Gan, "Deep metallic surface defect detection: The new benchmark and detection network," *Sensors (Switzerland)*, vol. 20, no. 6, 2020.
- [7] G. Ma, L. Yu, H. Yuan, W. Xiao, and Y. He, "A vision-based method for lap weld defects monitoring of galvanized steel sheets using convolutional neural network," *J. Manuf. Process.*, vol. 64, no. December 2020, pp. 130–139, 2021.
- [8] Y. Yang et al., "A lightweight deep learning algorithm for inspection of laser welding defects on safety vent of power battery," *Comput. Ind.*, vol. 123, p. 103306, 2020.
- [9] R. Miao et al., "Real-time defect identification of narrow overlap welds and application based on convolutional neural networks," *J. Manuf. Syst.*, no. June 2020, pp. 1–11, 2021.
- [10] [A. M. Deshpande, A. A. Minai, and M. Kumar, "One-shot recognition of manufacturing defects in steel surfaces," *Procedia Manuf.*, vol. 48, pp. 1064–1071, 2020.
- [11] X. He, T. Wang, K. Wu, and H. Liu, "Automatic defects detection and classification of low carbon steel WAAM products using improved remanence/magneto-optical imaging and cost-sensitive convolutional neural network," *Meas. J. Int. Meas. Confed.*, vol. 173, no. August 2020, p. 108633, 2021.
- [12] D. Mery, "Aluminum Casting Inspection Using Deep Learning: A Method Based on Convolutional Neural Networks," *J. Nondestruct. Eval.*, vol. 39, no. 1, 2020.
- [13] L. Jiang, Y. Wang, Z. Tang, Y. Miao, and S. Chen, "Casting defect detection in X-ray images using convolutional neural networks and attention-guided data augmentation," *Meas. J. Int. Meas. Confed.*, vol. 170, no. November 2020, p. 108736, 2021.
- [14] T. P. Nguyen, S. Choi, S. J. Park, S. H. Park, and J. Yoon, "Inspecting Method for Defective Casting Products with Convolutional Neural Network (CNN)," *Int. J. Precis. Eng. Manuf. - Green Technol.*, vol. 8, no. 2, pp. 583–594, 2021.
- [15] B. Staar, M. Lütjen, and M. Freitag, "Anomaly detection with convolutional neural networks for industrial surface inspection," *Procedia CIRP*, vol. 79, pp. 484–489, 2019.
- [16] Z. Snow, B. Diehl, E. W. Reutzel, and A. Nassar, "Toward in-situ flaw detection in laser powder bed fusion additive manufacturing through layerwise imagery and machine learning," *J. Manuf. Syst.*, vol. 59, no. February, pp. 12–26, 2021.
- [17] H. Pan, Z. Pang, Y. Wang, Y. Wang, and L. Chen, "A New Image Recognition and Classification Method Combining Transfer Learning Algorithm and MobileNet Model for Welding Defects," *IEEE Access*, vol. 8, pp. 119951–119960, 2020.
- [18] F. Nagata et al., "Defect detection method using deep convolutional neural network, support vector machine and template matching techniques," *Artif. Life Robot.*, vol. 24, no. 4, pp. 512–519, 2019.
- [19] X. Ji et al., "Filtered selective search and evenly distributed convolutional neural networks for casting defects recognition," *J. Mater. Process. Technol.*, vol. 292, no. June 2020, 2021.
- [20] R. Perera, D. Guzzetti, and V. Agrawal, "Optimized and autonomous machine learning framework for characterizing pores, particles, grains and grain boundaries in microstructural images," *Comput. Mater. Sci.*, vol. 196, no. April, p. 110524, 2021.
- [21] G. Yao, T. Lei, and J. Zhong, "A review of Convolutional-Neural-Network-based action recognition," *Pattern Recognit. Lett.*, vol. 118, pp. 14–22, 2019.
- [22] A. Dhillon and G. K. Verma, "Convolutional neural network: a review of models, methodologies and applications to object detection," *Prog. Artif. Intell.*, vol. 9, no. 2, pp. 85–112, 2020.

- [23] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [24] S. Liu and W. Deng, "Very deep convolutional neural network based image classification using small training sample size," *Proc. - 3rd IAPR Asian Conf. Pattern Recognition, ACPR 2015*, pp. 730–734, 2016.
- [25] L. Alzubaidi et al., *Review of deep learning: concepts, CNN architectures, challenges, applications, future directions*, vol. 8, no. 1. Springer International Publishing, 2021.
- [26] Y. Ho and S. Wookey, "The Real-World-Weight Cross-Entropy Loss Function: Modeling the Costs of Mislabeling," *IEEE Access*, vol. 8, pp. 4806–4813, 2020.
- [27] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7700 LECTU, pp. 437–478, 2012.