applied sciences

MDPI

Article

Customized Convolutional Neural Networks Technology for Machined Product Inspection

Yi-Cheng Huang [1,*], Kuo-Chun Hung [2], Chun-Chang Liu [3], Ting-Hsueh Chuang [2] and Shean-Juinn Chiou [1]

[1] Department of Mechanical Engineering, National Chung Hsing University, Taichung City 40227, Taiwan; sjchiou@dragon.nchu.edu.tw
[2] Department of Mechatronics Engineering, National Changhua University of Education, Changhua City 50007, Taiwan; hungkuochun520@gmail.com (K.-C.H.); tinghsueh2017@gmail.com (T.-H.C.)
[3] Ph.D. Program for Civil and Hydraulic Engineering, Water Resources Engineering, and Infrastructure Planning, Feng Chia University, Taichung City 407102, Taiwan; tony_liu@turvo.com.tw
* Correspondence: ychuang66@dragon.nchu.edu.tw

Abstract: Metal workpieces are an indispensable and important part of the manufacturing industry. Surface flaws not only affect the appearance, but also affect the efficiency of the workpiece and reduce the safety of the product. Therefore, the appearance of the product needs to be inspected to determine if there are surface defects, such as scratches, dirt, chipped objects, etc., after production is completed. The traditional manual comparison inspection method is not only time-consuming and labor-intensive, but human error is also unavoidable when inspecting thousands or tens of thousands of products. Therefore, Automated Optical Inspection (AOI) is often used today. The traditional AOI algorithm does not fully meet the subtle detection requirements and needs to import a Convolutional Neural Network (CNN), but the common deep residual networks are too large, such as ResNet-101, ResNet-152, DarkNet-19, and DarkNet-53. Therefore, this research proposes an improved customized convolutional neural network. We used a self-built convolutional neural network model to detect the defects on the metal's surface. Grad–CAM was used to display the result of the last layer of convolution as the basis for judging whether it was OK or NG. The self-designed CNN network architecture could be customized and adjusted without using a large network model. The customized network model designed in this study was compared with LeNet, VGG-19, ResNet-34, DarkNet-19, and DarkNet-53 after training five times each. The experimental results show that the self-built customized deep learning model avoiding the use of pooling and fully connected layers can effectively improve the recognition rate of defective samples and unqualified samples, and reduce the training cost. Our custom-designed models have great advantages over other models. The results of this paper contribute to the development of new diagnostic technologies for smart manufacturing.

Keywords: metal workpieces; custom-designed models; smart manufacturing

Citation: Huang, Y.-C.; Hung, K.-C.; Liu, C.-C.; Chuang, T.-H.; Chiou, S.-J. Customized Convolutional Neural Networks Technology for Machined Product Inspection. *Appl. Sci.* **2022**, *12*, 3014. https://doi.org/10.3390/app12063014

Academic Editor: Yujin Lim

Received: 13 February 2022
Accepted: 14 March 2022
Published: 16 March 2022

1. Introduction

Over the past few decades, the application of computer-aided design and analysis had gradually increased in the engineering industry for applications such as signal processing or simulation testing. In the past, manual inspection not only required a lot of labor, the test results may also be inaccurate, affecting the quality of the product, due to human factors, such as fatigue and manual measurement errors. Therefore, automatic optical inspection is beginning to be used more and more; nowadays, the technological advancement of hardware and software provides space for research using artificial intelligence [1], image processing [2], computer vision [3], and machine learning [4], which are the main research areas of artificial intelligence. In particular, image processing combined with deep learning detection is becoming more and more popular, combining the deep learning model with the existing optical inspection system; it was a technological breakthrough that could alleviate

the bottleneck of the manufacturing defect detection system, and also achieved the goal of advancing the manufacturing process.

Image classification technology in deep learning provides a new solution for image detection and can improve the accuracy of image detection [5,6]; in addition, with the advancement of graphics processing units, the computing power of the hardware has been greatly improved. The well-known deep learning frameworks, such as TensorFlow [7] and PyTorch [8], have also been developed, promoting the advancement of technology in the field of deep learning. A well-known YOLO [9] is widely used for the purposes of object detection and image processing. Many studies have used convolutional neural network models to classify different images [10]; based on the deep learning structure, convolutional neural network models are easy to train and can automatically search for useful features [11,12]. Thus, deep learning technology can be deployed to inspect and determine various image defects, and has been proven to be a very effective method [13].

At present, well-known convolutional neural networks include LeNet [14], VGG [15], DarkNet-19 [16], and the deep residual networks ResNet-34 [17] and DarkNet-53 [13]. The model structure of DarkNet-19 is similar to that of VGG [15]. DarkNet-19 has 19 convolutional layers and 5 maximum pooling layers. DarkNet-53 combines the elements of DarkNet-19 and ResNet [17]. For massive data, DarkNet-53 is much more effective than DarkNet-19 [18]. At the same time, DarkNet-53 achieves the highest floating point calculation speed per second in the network structure. This means that its network structure can make full use of the GPU [13]. Nevertheless, the deeper the network, the more difficult it is to converge. Many researchers have changed the Activation function [19] for preventing the gradient from disappearing. However, the problem still exists [20,21]. The disappearance or explosion of the gradient may be due to the high nonlinearity of the deep network. DarkNet-53, ResNet-101, and ResNet-152 use residual learning methods to solve the problem of accuracy that increases first and then saturates. However, this will also lead to network redundancy [22]. Therefore, how to reduce the number of hidden layers and retain the efficient feature extraction via the characteristics of manipulating convolution is attractive in our study.

One of the current studies on deep learning and optical inspection of the metal surfaces was by Eugene Su et al. [23]. They combined machine vision and deep learning to detect defects on the surface of metal cylinders and used a highly reflective metal surface as the test data. To improve the problem of high reflection of the metal's surface, they chose a strip light source and a round tube homogenizing plate as the light source setting. An all-white inner wall of a round tube was used. Since the strip light source can be placed in a round tube, the light source can reflect light uniformly in the inner wall of the round tube. A new ResNet architecture was used to train the model, and it was compared to the original ResNet model architecture. This article uses a 1×1 convolution kernel size in the last layer to achieve the effect of a fully connected layer; its parameters are lower than those using a fully connected layer. Therefore, the performance in practice is relatively good. In [24], for the aluminum alloy material, a camera to record the contours of extrusion during the production process was used. The neural network model distinguished perfect surfaces and surfaces with various common defects. The size, shape, and texture of the metal defects may be different when inspecting the metal's surface. Additionally, the defects that may appear are very similar. Therefore, Yasir Aslam et al. [25] proposed an automatic segmentation and quantification method and used it to check digital image defects by a customized deep learning architecture on titanium-coated metal surfaces. In [26], for a biomedical image, a U-Net convolutional network was used to segment the image first with appropriate pre-processing and post-processing. The input image was filtered with a median filter for eliminating possible impulse noise. As usual, standard benchmarks were used to evaluate the detection and subdivision performance. The accuracy of the model was 93.46%.

Shengping Wen et al. [27] designed a 26-layer convolutional neural network by themselves, which was used to identify surface defects of bearing machine components and

compared with MobileNet [28], VGG-19, and ResNet-50. Their VGG-19 achieved a mAP (mean average precision) of 83.86%, but the processing time was relatively long and took 83.3 ms. MobileNet has the fastest processing speed, but its mAP is the lowest among all network models due to the reduction of parameters and calculations. The network designed in [27] achieved a better balance between the mAP and data processing efficiency, where the mAP was close to the highest mAP of ResNet-50. Compared with ResNet-50 and VGG-19, the detection time has some advantages. In [29], the entropy calculation method was used to adjust the self-designed neural network model and choose the most suitable kernel size for the convolutional layer. The recognition rate of components, shortening the model training time, was improved.

How to customize a suitable CNN with the aim of reducing the hidden layer numbers and hyper-parameters numbers, and determining efficient feature extraction characteristics with the consideration of image size by manipulating the convolution network are the focus of this study. The core of this paper is based on the customized CNN architecture. The objective is to identify the metal surface defects of a metal part immediately after the Computer Numerical Control machine tool finished the machining operation on the shop floor. A customized light source was used to illuminate the metal workpiece to resolve the problem of high metal reflection. By adjusting the manipulated parameters, such as the stride number and convolution kernel size to replace the pooling layer and fully connected layer, the optimal convolution kernel size was chosen to improve the metal product defect recognition rate and shorten the model training time. With the aid of Gradient-weighted Class Activation Mapping (Grad–CAM) [30], the last layer convolution will demonstrate the successful defect diagnostic results. Comparing the results with the well-known LeNet, VGG-19, ResNet-34, DarkNet-19, and DarkNet-53 models, the proposed customized network model demonstrated the highest accuracy, and had a greater advantage for the machined defect data tested in this study.

2. System Architecture

Light Source Settings

The surface of the workpiece (Figure 1) tested this time was composed of opaque mirror material. It was a related part of a car engine. The green box is the area detected by the camera with a customized light source detection system. Since this circular chamfered hole will have wires passing through it during actual usage, the defect caused by the machining defect process will lead to the possibility of cutting the wires, resulting in a safety problem. Using a common light source will make the incident angle equal to the reflection angle, which conforms to the laws of geometric optics and produces total reflections. In this study, in order to solve the reflection problem of the light source illuminating the metal material, a light source system was specially designed for diagnostic detection. This system was composed of a 1280 $\times$ 1024 pixel camera (Basler acA1280-60gm GigE, CMOS, Ahrensburg, Germany) and with a 50 mm plus 15 mm extension ring lens. It used a shadowless ring light to provide illumination from different angles of light. This can highlight the flaws of the object, effectively solve the shadow problem caused by direct illumination, and obtain the optimal light source illumination position by 100 mm and the camera inclination angle α after many adjustments and experiments. The angular position between the camera and the light source is shown in Figure 2 with some well-tuned specific positioning distance.

In the experiments, the shooting distance and angle were fixed when capturing the images. The lens was at a distance of 100 mm from the detection object, while the shadowless ring light was 5 mm away from the detection object. The light source was illuminated by a low-angle illumination method. This method allowed the light to have good uniformity and brightness, which could enhance the surface feature extraction of the detected object and reduce reflections. When shooting images, the detected object was rotated at the position of the central axis to shoot and capture images.

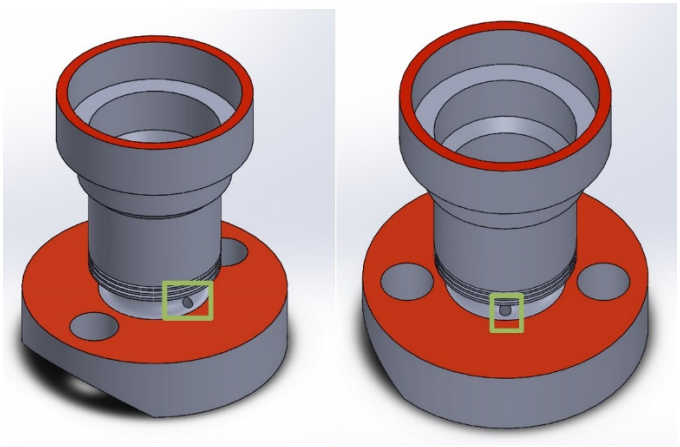

Figure 1. Illustration of the green box detection area for the 3D drawing of the machined metal.

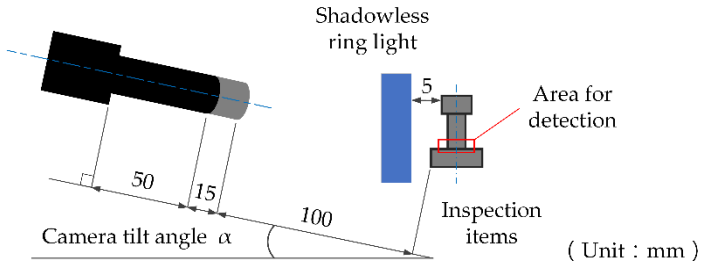

Figure 2. Illustration of the customized light source detection system with positioning distance.

3. Experimental Method

3.1. Data Set

In this paper, 304 stainless steel material parts were used as the detection items in this experiment. The experimental setup for the in-line manufacturing system and an illustration of the pictured image of the inspection object are shown in Figure 3. Unwanted burr around the chamfered hole was generated when the workpiece was machined by CNC. Because there will be wires passing through this hole, in order to avoid scratches of the wires, they should be detected here, as shown in Figure 4 with the pictured images. The left is an example of an OK figure—the area circled in green is the focus of this judgment—and the right is an example of a no good (NG) figure. It can be found that the oblique area of the part circled in red was much larger than that of the OK on the left. Figure 5a is an enlargement of the OK image, while Figure 5b is the NG image. As we can see, the green-framed area was black because the machining process was normal. However, we can see that there was a white abnormal area in the red-framed area of Figure 5b, because the machining process was abnormal. Figure 6 provides more examples of six OK images and NG images. In this experiment, a total of 1895 surface images of parts were collected as data sets, 1302 of which were OK images without flaws, and the remaining 593 were NG images with flaws. Among them, 80% of the data sets were used for training, and the remaining 20% of the data sets were used for testing. Because the metal surface of the detected target had reflective properties, this experiment used a shadowless ring light to illuminate the surface of the metal workpiece at a low angle to solve the problem that the metal surface was prone to reflection without affecting the feature extraction of the target. In this paper, the original image size of 1280 × 550 pixels was reduced to 186 × 189 pixels, and the defects caused by incomplete chamfering were marked as NG.

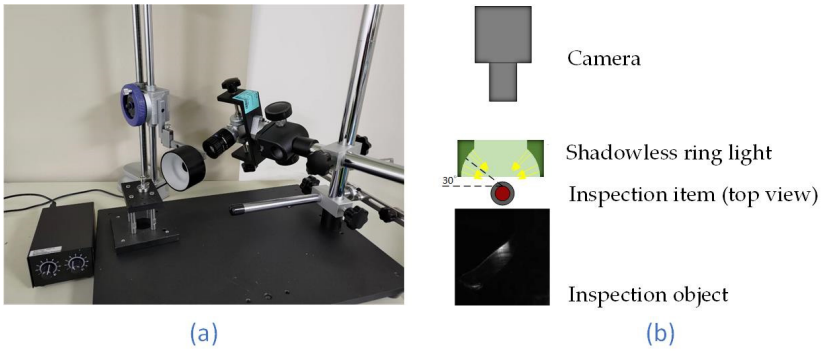

Figure 3. (**a**) Photo of the experimental setup for the in-line system. (**b**) Illustration for the top-view pictured image of the inspection object.

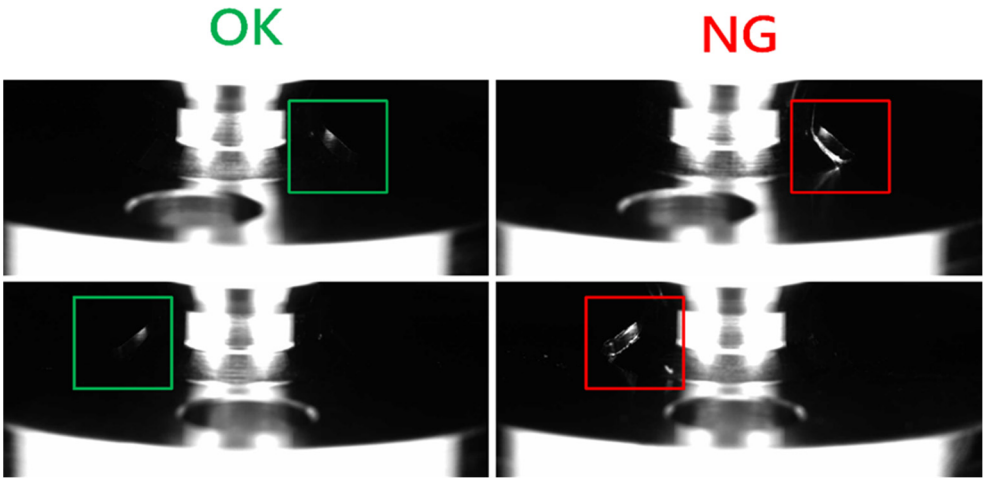

Figure 4. Photo of the OK items and the NG defect items.

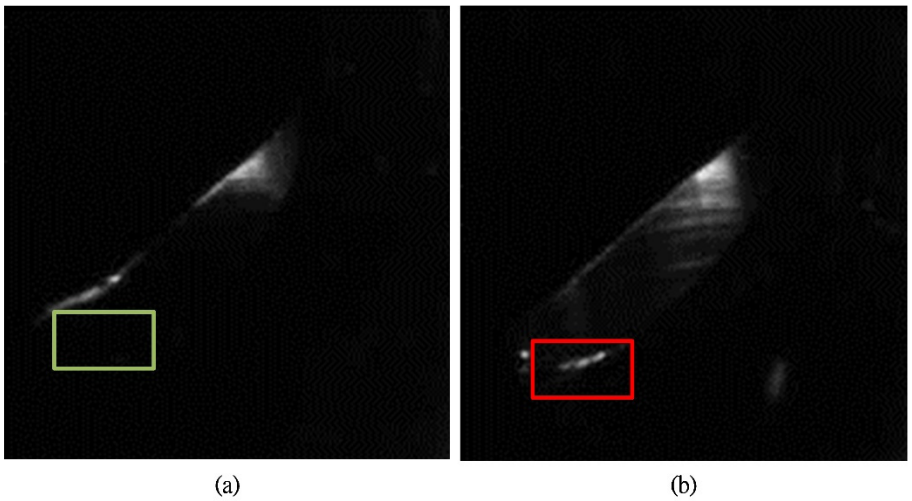

Figure 5. Enlargement for the comparison chart of the (**a**) OK image and (**b**) NG defect image.

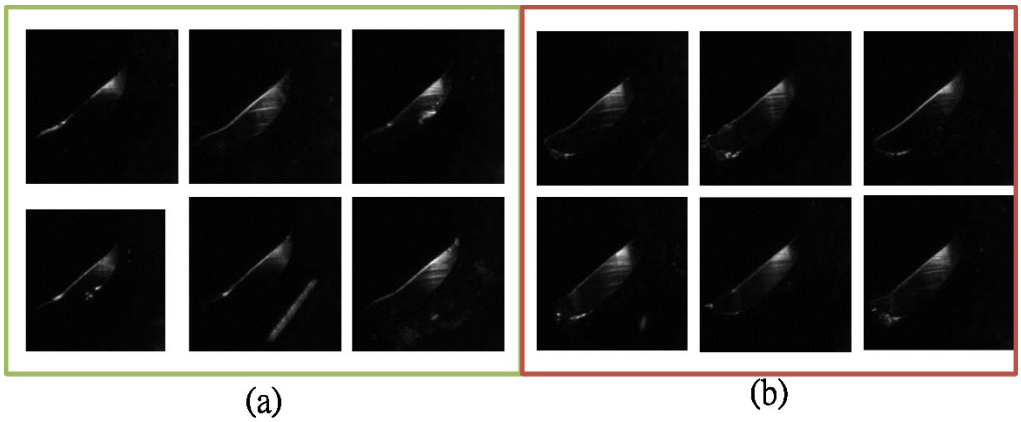

Figure 6. (**a**) Examples of six OK images and (**b**) six NG images.

3.2. Comparison of Results after Training with Common Models

This paper used MATLAB software to develop convolutional neural networks, selecting several common models, including LeNet, VGG-19, ResNet-34, DarkNet-19, and DarkNet-53 for training, and compared the results. We used the following hyperparameters for each training: the Batch Size was set to 8, the Learning Rate was set to 0.0001, and the Epoch was set to 1. The data set was used for training with each model in sequence. In order to achieve fair experimental results, we used the same hyperparameters and data set to train each model five times. The results of training of the LeNet, VGG19, ResNet34, DarkNet19, and DarkNet53 models five times were averaged to obtain accuracies of 98.45%, 98.93%, 93.68%, 98.64%, and 98.64%, respectively, as shown in Table 1. In Table 2, the prediction times by implementing the LeNet, VGG19, ResNet34, DarkNet19, and DarkNet53 models five times were averaged to 0.391, 0.303, 0.946, 0.329, and 1.167 s, respectively. A note is made here that the first prediction time is always the highest with real implementation. This is caused by the time in deploying the parameters into the GPU. The rest of the consecutive prediction time will be less than the first one naturally.

Table 1. Training accuracy for the VGG19, ResNet34, LeNet DarkNet 19 and DarkNet53 models.

Model	VGG19	ResNet34	LeNet	DarkNet19	DarkNet53
First training accuracy (%)	99.00	89.37	97.36	98.53	99.04
Second training accuracy (%)	99.07	97.22	99.07	98.92	97.56
Third training accuracy (%)	98.72	94.72	98.93	98.05	98.54
Fourth training accuracy (%)	98.96	91.28	98.22	98.84	98.91
Fifth training accuracy (%)	98.90	95.81	98.67	98.46	99.15
Average accuracy (%)	98.93	93.68	98.45	98.64	98.64

Table 2. Prediction time of the implementation of the 5 models.

Model	VGG19	ResNet34	LeNet	DarkNet19	DarkNet53
First prediction time (s)	0.889	0.666	2.325	0.824	1.84
Second prediction time (s)	0.315	0.245	0.623	0.265	1.02
Third prediction time (s)	0.27	0.204	0.591	0.211	1.009
Fourth prediction time (s)	0.202	0.172	0.532	0.136	0.945
Fifth prediction time (s)	0.281	0.229	0.661	0.21	1.023
Average prediction time (s)	0.391	0.303	0.946	0.329	1.167

3.3. Customized Model Design

Based on the comparison results after training each model in Section 3.2, the VGG19 model had the highest accuracy, so we modified it with the model architecture of VGG19. We still set the Batch Size and Learning Rate to 8 and 0.0001, respectively, which remained

unchanged. Table 3 details the customized CNN model. First, we added a convolutional layer after the convolutional layer, and set the Stride of the added second convolutional layer to 2, so that the OutPut Size after the convolution could be reduced by half. This method could achieve the function of the pooling layer, and adding Batch Normalization and ReLU activation functions between the two convolutional layers helped to slow down the disappearance of gradients and accelerated the convergence of the model, which had the effect of regularization.

Table 3. Customized design model architecture.

Layer Type	Filters	Kernel Size/Stride	Output Size
Convolutional 1	64	3 × 3/1	189 × 186 × 64
Batch Normalization/ReLU			189 × 186 × 64
Convolutional 2	64	3 × 3/2	95 × 93 × 64
Batch Normalization/ReLU			95 × 93 × 64
Convolutional 3	128	3 × 3/1	95 × 93 × 128
Batch Normalization/ReLU			95 × 93 × 128
Convolutional 4	128	3 × 3/2	48 × 47 × 128
Batch Normalization/ReLU			48 × 47 × 128
Convolutional 5	256	3 × 3/1	48 × 47 × 256
Batch Normalization/ReLU			48 × 47 × 256
Convolutional 6	256	3 × 3/2	24 × 24 × 256
Batch Normalization/ReLU			24 × 24 × 256
Convolutional 7	512	24 × 24/1	1 × 1 × 512
Convolutional 8	2	1 × 1/1	1 × 1 × 2
Softmax			1 × 1 × 2
Classoutput			1 × 1 × 2

The OutPut Size of the sixth layer of the customized design model was 24 × 24, so we also set the Kernel Size of the seventh layer to 24 × 24, so that the OutPut Size of the seventh layer could reach 1 × 1, and set the Filters to 512 to replace the effect of the fully connected layer. Using a convolutional layer has fewer parameters than using a fully connected layer. We compared the accuracy of using a fully connected layer with that of using a convolutional layer instead of a fully connected layer, and found that using convolution instead of a fully connected layer performed relatively well.

3.4. Accuracy of Customized Model

The custom-designed convolutional neural network in this experiment (Figure 7) was trained with an NVIDIA GeForce RTX 2070 SUPER GPU; the Batch Size was set to 8, Learning Rate was set to 0.0001, Epoch was set to 1, and iteration was set to 119. This training took 1 min 14 s and achieved 99.36% Accuracy.

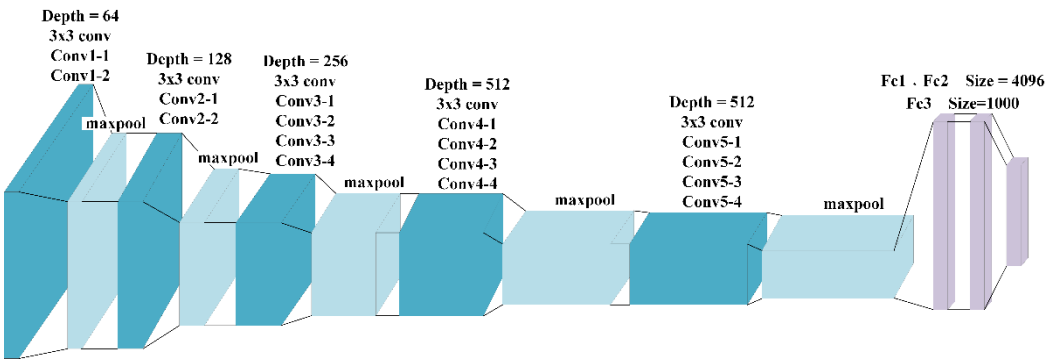

Figure 7. *Cont.*

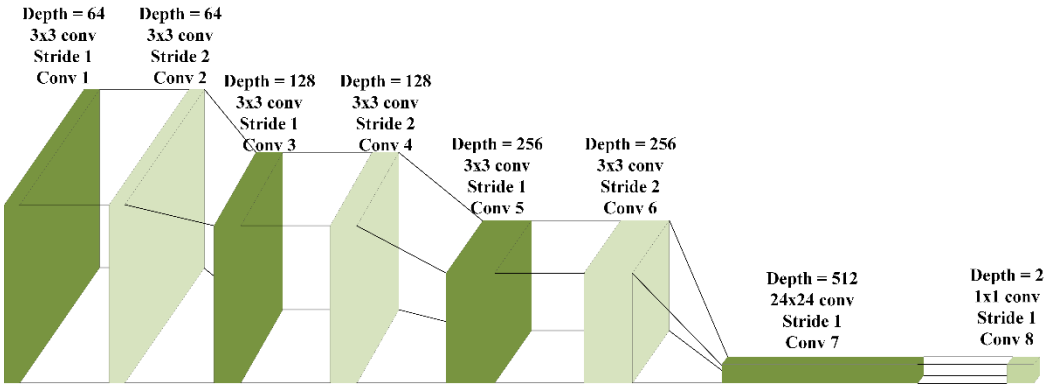

Figure 7. Architecture diagram of the VGG19 (**top**) and customized CNN model (**bottom**).

From the training result graph (Figure 8), it can be found that, when the Iteration was around 10, the Accuracy value began to gradually increase, and when the Iteration was in the range of 20 to 80, the Accuracy value oscillated in the range of 80 to 100. Iteration gradually converged after 80 and the oscillation was less obvious.

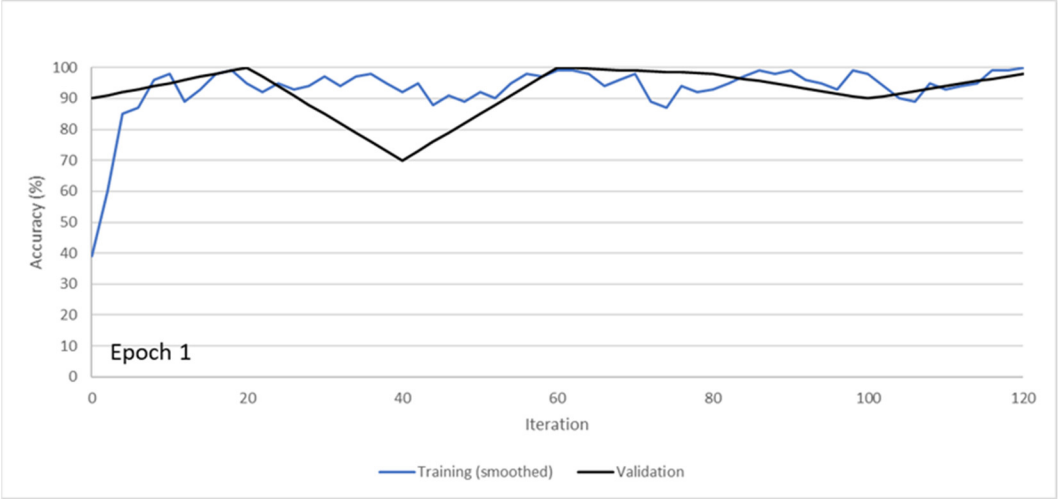

Figure 8. Custom-designed convolutional neural network training results.

4. Experimental Results

The calculation times by using the LeNet, VGG19, ResNet34, DarkNet19, and DarkNet53 models were 25, 87, 838, 59, and 163 s respectively, on average after five training sessions. Our customized model took 74 s on average after five training sessions. In Table 4, compared with VGG19, our accuracy was higher with less time and lower model parameters.

Table 4. Training accuracy and training time of each network model.

Model	Customized CNN	VGG19
Accuracy	99.36%	98.93%
Time	74 s	87 s
Parameters	76,643,266	143,667,240

In Table 5, the five prediction times for the Customized CNN and VGG19 models were averaged to obtain 0.433 and 0.391 s, respectively. The prediction time of Customized CNN was slightly higher than that of VGG19.

Table 5. Prediction times of the Customized CNN and VGG19 models.

Model	Customized CNN	VGG19
First prediction time (s)	0.953	0.889
Second prediction time (s)	0.343	0.315
Third prediction time (s)	0.307	0.270
Fourth prediction time (s)	0.246	0.202
Fifth prediction time (s)	0.318	0.281
Average time (s)	0.433	0.391

As usual, the original defect inspection was conducted manually. Such defects are very subtle and need to be inspected with a microscope repeatedly by human labor. To reduce the cost of the manual inspection time, AOI is a must. Leveraged by CNN in the above, the average prediction time of the developed automatic image recognition system was less than one second. This benefits the quality assurances and resolves the problem in the production line when inspection must be implemented for every single piece. The labor reduction and increase in the production line bring the development of industry.

4.1. Procedure to Customize the CNN

In this section, we detail the adjustment strategy when customizing the CNN model architecture. First, we classify common models according to the number of parameters and layers, such as the LeNet, VGG-19, ResNet-34, DarkNet-19, and DarkNet-53 models used in this experiment. We defined LeNet as a small model, VGG-19 and DarkNet-19 as medium models, and ResNet-34 and DarkNet-53 as large models. As the data sets for each training were different, we first put the data sets through three types of model training: large, medium, and small. Then, we compared the training results of the large, medium, and small models, selected the model size that was more in line with our expectations, and then adjusted the parameters. It can be seen from this experiment that the results of using convolutional layers, instead of pooling layers and fully connected layers, were better, so they can be adjusted from the replacement of convolutional layers. As shown in Table 3, the self-designed CNN network architecture could be customized and adjusted without using a large network model, which could reduce the amount of parameters and shorten the computing time of the neural network.

4.2. Visualizing Convolutional Networks

Although we know that the usage of CNN is more accurate than the human eye in image recognition, we cannot easily know what the neural network performs during the process of the convolution operation. Heuristically, the characteristics of the picture to be diagnosed by the neural network are like a black box. We only know the input dataset with known training features and the output result without knowing the abstract key features. Therefore, in addition to using the neural network to identify the defects, in this research, Grad–CAM [30] was used to connect the last layer of the neural network to show the neural network identification result with visualization for validating the status of the original OK or NG pictures.

The approach of Class Activation Mapping (CAM) [31] is very straightforward. For example, in the NG image, each feature map generated by the convolutional layer of the last layer will become a pixel after GAP (Global Average Pooling). Multiply the pixel array after GAP by the weight w; the larger the value of the weight w, the greater the influence of the image represented by the pixel. After the function of Softmax, it can be determined that NG is the maximum value of the classification. By multiplying the pixels of the entire feature map by the weight w and then superimposing them, we can focus on different regions according to the importance of each feature map. The larger the weight w corresponding to the classification, the greater the influence of the feature map; on the contrary, the less important the feature maps with the weight closer to 0. Therefore, if the convolutional neural network model designed in this experiment does not use GAP after

the convolutional layer of the last layer, the architecture of the model must be modified and retrained. For the Grad–CAM used this time, no matter what kind of neural network the model used, after the convolutional layer, the aim of CAM could be implemented without modifying the model. The Grad–CAM-visualized results are shown in Figure 9, where Figure 9(a1–c2) are the three images randomly sampled after the last layer of convolution. For example, Figure 9(a1) is the original feature map, while Figure 9(a2) is the image immediately after inputting Figure 9(a1) into the Grad–CAM software. It can be found that Figure 9(a2) exhibited the obvious defect area from Figure 9(a1) and then marked it in red. All three visualized convolutional neural networks resulted in NG in Figure 9(a1–c1). The color bar indicates the strength of the weighting result via the Grad–CAM. The red part illustrates the characteristic features for the NG metal part with defects and scratches.

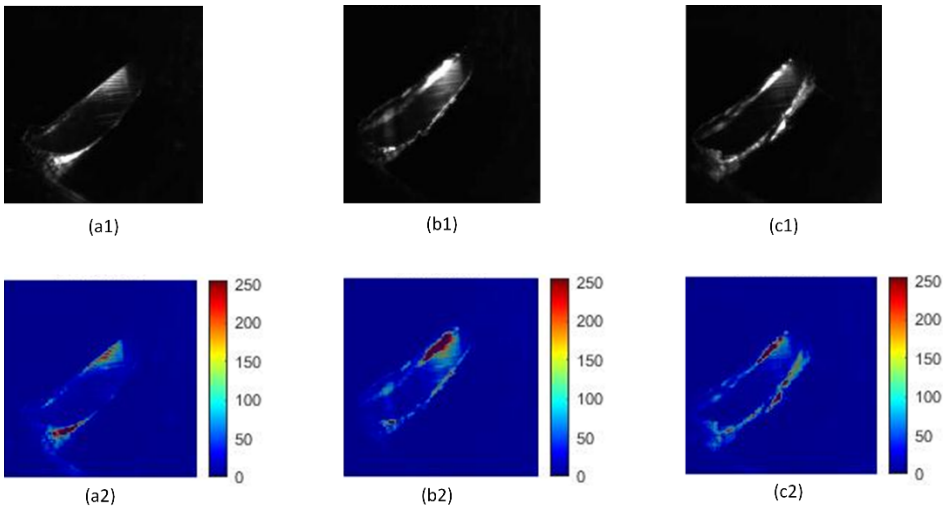

Figure 9. Grad–CAM-visualized results with NG images for (**a2–c2**), where the (**a1–c1**) are the originally captured images.

5. Conclusions

This research combines industrial machine vision and deep learning, and deploys the machined metal defect detection for smart manufacturing and reduced production costs. The three main contributions are as follows. First, we provided a quick layer structure solution by feeding the initial dataset to some well-known small, medium, and large CNN layers with only one epoch simulation. This sped up the selection of the number of CNN layers with an appropriate CNN backbone size. Second, the customized light source with a shadowless ring light used in this experiment to illuminate the metal workpiece successfully solved the problem of easy reflection on the metal surface without affecting the feature extraction of the measured object with deep learning technology, where the traditional AOI algorithm did not fully meet the needs of this detection. Thirdly, we used a self-built custom convolutional neural network model, replacing pooling layers and fully connected layers with convolutional layers only. The proposed CNN minimized the number of parameters while maintaining extremely high performance and retained remarkable accuracy. The experimental results showed that, after comparing the customized convolutional neural network model designed in this study with the ResNet-34, VGG-19, DarkNet-19, and LeNet models, our model achieved 99.36% accuracy, which was better than that of other models. Additionally, the Grad–CAM module validated the last layer of the convolution result. The highest accuracy could effectively improve the recognition rate of defective and non-defective samples, and achieve the effect of reducing training costs. The results of this paper contribute to the technological development of automatic optical inspection, and also contribute to intelligent manufacturing. These findings for how to customize a CNN are expected to enable the development of visual inspection techniques with high adaptability

to overcome the bottlenecks of current image processing techniques and advance the advancement of manufacturing processes.

Author Contributions: Conceived, Y.-C.H. and T.-H.C.; wrote the paper, Y.-C.H.; methodology, Y.-C.H., T.-H.C., K.-C.H. and S.-J.C.; software, T.-H.C., C.-C.L. and K.-C.H. All authors have read and agreed to the published version of the manuscript.

Funding: MOST 109-2221-E-005-082-MY2 and MOST 110-2218-E-005-018.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We thank the Ministry of Science and Technology for supporting Project No. MOST 109-2221-E-005-082-MY2 and MOST 110-2218-E-005-018 for this research.

Conflicts of Interest: The authors declare that there is no conflict of interest regarding this publication.

References

1. Korbicz, J.; Koscielny, J.M.; Kowalczuk, Z.; Cholewa, W. (Eds.) *Fault Diagnosis: Models, Artificial Intelligence, Applications*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012.
2. Nixon, M.; Aguado, A. *Feature Extraction and Image Processing for Computer Vision*; Academic Press: Cambridge, MA, USA, 2019.
3. Chen, C.H. (Ed.) *Handbook of Pattern Recognition and Computer Vision*; World Scientific: Singapore, 2015.
4. Duygulu, P.; Barnard, K.; de Freitas, J.F.; Forsyth, D.A. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In Proceedings of the European Conference on Computer Vision, Copenhagen, Denmark, 28–31 May 2002; Springer: Berlin/Heidelberg, Germany, 2002; pp. 97–112.
5. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504–507. [CrossRef] [PubMed]
6. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
7. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. Tensorflow: A system for large-scale machine learning. In Proceedings of the 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16), Savannah, GA, USA, 2–4 November 2016; pp. 265–283.
8. Ketkar, N.; Moolayil, J. Introduction to pytorch. In *Deep Learning with Python*; Apress: Berkeley, CA, USA, 2017; pp. 195–208.
9. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
10. Kang, J.; Park, Y.J.; Lee, J.; Wang, S.H.; Eom, D.S. Novel leakage detection by ensemble CNN-SVM and graph-based localization in water distribution systems. *IEEE Trans. Ind. Electron.* **2017**, *65*, 4279–4289. [CrossRef]
11. Jiang, W.; Ma, Y.; Liu, B.; Liu, H.; Zhou, B.B.; Zhu, J.; Wu, S.; Jin, H. Layup: Layer-adaptive and multi-type intermediate-oriented memory optimization for GPU-based CNNs. *ACM Trans. Archit. Code Optim. (TACO)* **2019**, *16*, 1–23. [CrossRef]
12. Guha, R.; Das, N.; Kundu, M.; Nasipuri, M.; Santosh, K.C. Devnet: An efficient cnn architecture for handwritten devanagari character recognition. *Int. J. Pattern Recognit. Artif. Intell.* **2020**, *34*, 2052009. [CrossRef]
13. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
14. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
15. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
16. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
17. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
18. Sang, D.V.; Hung, D.V. YOLOv3-VD: A sparse network for vehicle detection using variational dropout. In Proceedings of the Tenth International Symposium on Information and Communication Technology, Ha Noi, Vietnam, 4–6 December 2019; pp. 280–284.
19. Elliott, D.L. *A Better Activation Function for Artificial Neural Networks*; ISR Technical Report TR 93-8; Institute for Systems Research Technical Reports: College Park, MD, USA, 1993.
20. Ciuparu, A.; Nagy-Dăbâcan, A.; Mureşan, R.C. Soft++, a multi-parametric non-saturating non-linearity that improves convergence in deep neural architectures. *Neurocomputing* **2020**, *384*, 376–388. [CrossRef]
21. Chen, Z.; Ho, P.H. Global-connected network with generalized ReLU activation. *Pattern Recognit.* **2019**, *96*, 106961. [CrossRef]
22. Ayinde, B.O.; Inanc, T.; Zurada, J.M. Redundant feature pruning for accelerated inference in deep neural networks. *Neural Netw.* **2019**, *118*, 148–158. [CrossRef] [PubMed]

23. Su, E.; You, Y.-W.; Ho, C.-C. Machine Vision and Deep Learning Based Defect Inspection System for Cylindrical Metallic Surface. *Instrum. Today* **2018**, 46–58. Available online: https://www.airitilibrary.com/Publication/alDetailedMesh?DocID=10195440-201806-201806270008-201806270008-46-58#Altmetrics (accessed on 10 February 2022). (In Chinese).
24. Neuhauser, F.M.; Bachmann, G.; Hora, P. Surface defect classification and detection on extruded aluminum profiles using convolutional neural networks. *Int. J. Mater. Form.* **2020**, *13*, 591–603. [CrossRef]
25. Aslam, Y.; Santhi, N.; Ramasamy, N.; Ramar, K. Localization and segmentation of metal cracks using deep learning. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *12*, 4205–4213. [CrossRef]
26. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer: Cham, Switzerland, 2015; pp. 234–241.
27. Wen, S.; Chen, Z.; Li, C. Vision-based surface inspection system for bearing rollers using convolutional neural networks. *Appl. Sci.* **2018**, *8*, 2565. [CrossRef]
28. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
29. Jian, B.L.; Hung, J.P.; Wang, C.C.; Liu, C.C. Deep Learning Model for Determining Defects of Vision Inspection Machine Using Only a Few Samples. *Sens. Mater.* **2020**, *32*, 4217–4231. [CrossRef]
30. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626.
31. Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; Torralba, A. Learning deep features for discriminative localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2921–2929.