processes

MDPI

Article

PCB Defect Detection Based on Deep Learning Algorithm

I-Chun Chen [1], Rey-Chue Hwang [1] and Huang-Chu Huang [2,*]

[1] Department of Electrical Engineering, I-Shou University, Kaohsiung 84001, Taiwan
[2] Department of Telecommunication Engineering, National Kaohsiung University of Science and Technology, Kaohsiung 81157, Taiwan
* Correspondence: h4530@nkust.edu.tw; Tel.:+886-09-2141-2202

Abstract: Printed circuit boards (PCBs) are primarily used to connect electronic components to each other. It is one of the most important stages in the manufacturing of electronic products. A small defect in the PCB can make the final product inoperable. Therefore, careful and meticulous defect detection steps are necessary and indispensable in the PCB manufacturing process. The detection methods can generally be divided into manual inspection and automatic optical inspection (AOI). The main disadvantage of manual detection is that the detection speed is too slow, resulting in a waste of human resources and costs. Thus, in order to speed up the production speed, AOI techniques have been adopted by many PCB manufacturers. Most current AOI mechanisms use traditional optical algorithms. These algorithms can easily lead to misjudgments due to different light and shadow changes caused by slight differences in PCB placement or solder amount so that qualified PCBs are judged as defective products, which is also the main reason for the high misjudgment rate of AOI detection. In order to effectively solve the problem of AOI misjudgment, manual re-judgment is currently the reinspection method adopted by most PCB manufacturers for defective products judged by AOI. Undoubtedly, the need for inspectors is another kind of labor cost. To reduce the labor cost. of manual re-judgement, an accurate and efficient PCB defect reinspection mechanism based on deep learning algorithm is proposed. This mechanism mainly establishes two detection models, which can classify the defects of the product. When both models have basic recognition capabilities, the two models are then combined into a main model to improve the accuracy of defect detection. In the study, the data provided by Lite-On Technology Co., Ltd. were implemented. To achieve the practical application value in the industry, this research not only considers the problem of detection accuracy, but also considers the problem of detection execution speed. Therefore, fewer parameters are used in the construction of the model. The research results show that the accuracy rate of defect detection is about 95%, and the recall rate is 94%. Compared with other detection modules, the execution speed is greatly improved. The detection time of each image is only 0.027 s, which fully meets the purpose of industrial practical application.

Keywords: PCB; defect; detection; deep learning

check for updates

Citation: Chen, I.-C.; Hwang, R.-C.; Huang, H.-C. PCB Defect Detection Based on Deep Learning Algorithm. *Processes* **2023**, *11*, 775. https://doi.org/10.3390/pr11030775

Academic Editor: Yo-Ping Huang

Received: 30 January 2023
Revised: 27 February 2023
Accepted: 3 March 2023
Published: 6 March 2023

1. Introduction

The early through-hole technology (THT) is to pass the pins of the components through the circuit board and then solder them, as shown in Figure 1. Since all components need to be designed with solder pins to pass through the circuit board, the components cannot be designed too small, and the size of the circuit board cannot be too small either. Surface-mount technology (SMT) is a technique in which the electrical components are mounted onto the surface of a printed circuit board (PCB). SMT is made by coating the soldering feet of parts with solder paste. This technology can shorten the soldering feet of the parts so that the parts can be made smaller and smaller, thereby reducing the volume of electronic parts, as shown in Figure 2. Currently, SMT has widely replaced the THT electronic part assembly method because SMT can improve the automation of electronic manufacturing, thereby reducing cost and improving quality [1].

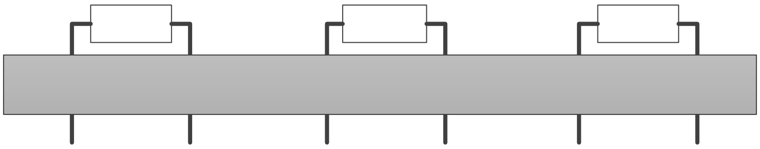


Figure 1. THT technique.

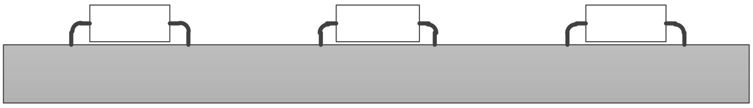


Figure 2. SMT technique.

PCB is the most basic and vital part of current electronic products. Any component installation error on the circuit board, such as component missed, component flipped, component shifted, short circuit, tombstone, etc., will lead to a failed circuit operation and make electronic products have serious defects [2]. In recent years, with the improvement of production technology, PCB quality inspection has become a key process in the production of electronic products. PCB defect detection not only helps to improve the yield of circuit board production, but also reduces repair costs caused by defects [3].

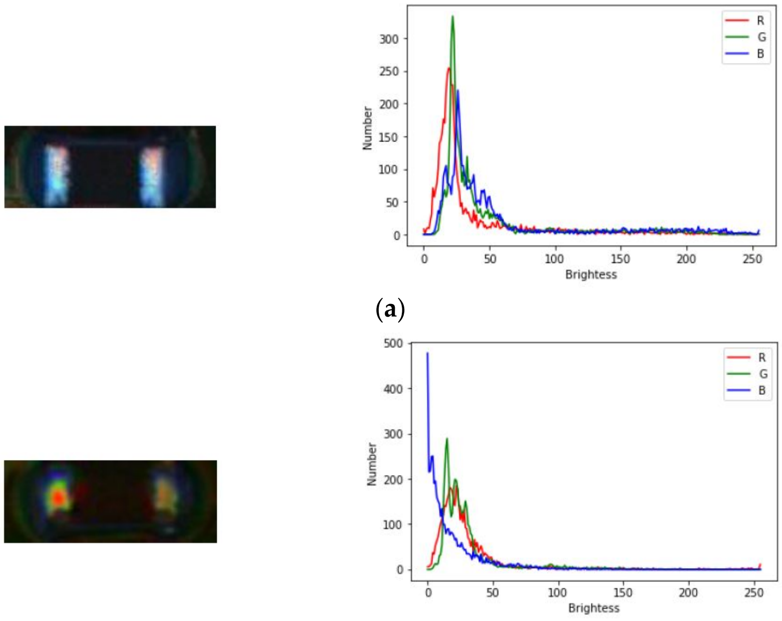
According to the detection method, PCB defect inspection can be divided into contact type and non-contact type. The contact type method is to evaluate the electrical conductivity of the circuit, but it cannot detect the main defects in the appearance. The non-contact type method can use optical modalities such as X-ray imaging, ultrasound imaging, thermal imaging, and image processing for a wide range of detection [4,5].

PCB inspection can be divided into defect detection and defect classification. Traditional PCB inspection is mostly conducted by manual inspection. However, manual inspection has the disadvantages of being time-consuming, causing work fatigue, and being error-prone, and the inspection results will be different due to the replacement of inspectors.

As we all know, AOI is an automatic visual inspection system that has been widely used in PCB production, and it plays an important role in ensuring that each circuit board has the high quality required for complex electrical applications. When AOI finds a defect in a PCB, it flags the defect so that the board can be returned for repair. In addition, if a board contains more than 100 components, AOI can inspect this complex board with a precision that cannot be matched by manual inspection.

When AOI judges whether a PCB is defective by computer vision, it can be divided into three different methods, namely, referential approach, non-referential approach, and hybrid approach. The referential approach is mainly to subtract and compare the PCB test image with its original sample image to obtain the defect judgment. Its advantage is that it is intuitive and easy to understand, but its disadvantage is that the alignment accuracy between images is extremely high. In general, perfect alignment between the test image and the sample image is very difficult, and the light environment of the image capture process is a sensitive and important key. The non-referential approach is based on design rules, which mainly extracts features from the entire image to find defective parts. However, when this approach is designed, it is difficult to characterize every defect type, so this method is easy to have defect dropout and distortion in the inspection operation. The hybrid approach is a combination of the referential and non-referential approaches, but the disadvantage of the hybrid method is the high computational complexity [6]. Most of the AOI inspection machines use non-learning algorithms, such as the image comparison method or rules checking method to detect defect. However, for different inspection machines, there will be problems of image resolution and color scale differences.

As shown in Figure 3, a good product was photographed with two different inspection machines (machine A, machine B). From two images, it can be observed that the color scales of the two are obviously different. Drawing the RBG values of the two graphs with histograms, it can be found that the blue light brightness of machine A is smaller than that of machine B. This result shows that different machines and different positions of light sources will result in different image color scales. Therefore, if images taken by different AOI machines are used for detection, the detection algorithm should first solve the problem of color scale differences. Certainly, it will increase the difficulty in the design of the detection algorithm.

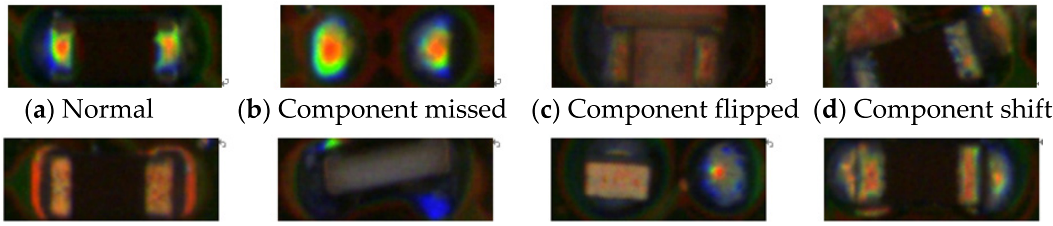


(**a**)

(**b**)

Figure 3. Different image color scales. (**a**) Photograph and RGB values by machine A; (**b**) Photograph and RGB values by machine B.

Therefore, developing an inspection system that can quickly and accurately perform the defect re-judgment on the images captured by different AOI machines is a very important task for current PCB manufacturers. This research uses a small number of defective product images to establish an artificial intelligence (AI) deep learning model which can distinguish the types of defects. The AI model is expected to overcome the color scale problem caused by different machines and can be applied to different machines effectively. Figure 4 shows several PCB defects to be identified in this study. Figure 4a is the component with no defect. Figure 4b indicates the component is in missed condition. Figure 4c indicates the component is placed in the opposite direction; it is component flipped. Figure 4d indicates the component is shifted. Figure 4e is the defect of insufficient solder. Figure 4f indicates the component is in sideward condition. Figure 4g indicates the component is in the case tombstone, and Figure 4h indicates the component is in a non-wetting situation.



(**a**) Normal (**b**) Component missed (**c**) Component flipped (**d**) Component shifted

(**e**) Insufficient solder (**f**) Component sideward (**g**) Tombstoning (**h**) Non-wetting

Figure 4. Defects of PCB.

2. Model Design

2.1. Design Theory

Generally, there are non-learning methods and learning methods for detecting the defects of PCBs [7]. Non-learning-method defect detection usually compares the difference between sample images and test images and uses the algorithm to define the features of defects. A study [8] used the extraction of the shape features, digital features, and logical features of solder joints from images to detect IC solder joint defects, such as surplus solder, lacking, solder, no solder, lead lift, lead bend, shift, bridged, and pseudo joints. Another study [9] took a review of PCB defect detection using image processing. A template of a defect-free PCB image and a defected test PCB image are segmented and compared with each other using image subtraction and other procedures. Xie et al. proposed a high-speed method to adjust the position of the part and used the difference image between component image and its template image to diagnose the defect [10]. Wu et al. presented an inspection process based on machine vision. An adaptive segmentation method based on local valley and the sliding location window algorithm were used to obtain the binary component electrodes and the location of the component, respectively. Then, the defects such as component missing, component rotation, and component shift were detected by analyzing the projection information of the electrodes [11].

Learning-based methods usually need to collect a large amount of defect data so that the detective model can accurately learn the characteristics of defects, and then use a classifier to perform the classification. In recent years, learning-based methods have been widely used in the application of defect classification, especially since the convolutional neural network (CNN) has enhanced the impact of deep learning on image classification. Many CNN studies in computer vision have been proposed, such as object detection [12–18], image segmentation [19–33], image classification [34,35], and defect detection [36]. The studies using CNN-based deep learning networks to classify defects in surface mount components (SMD) have been proposed. Kim et al. proposed a dual-stream CNN using two solder regions for inspections of SMT assembly defects [7,37].

The aim of this research is to introduce deep learning technology into the original inspection process without changing the original AOI inspection method, so as to reduce the over-inspection rate of AOI machines. In the past, in order to improve the problem of a high over-inspection rate of AOI machines, the defect images judged by the machine were manually re-inspected. Determining how to effectively solve the problem of the high over-inspection rate of the original AOI machine is the purpose of this study. The PCB images collected in this study are all taken by the original AOI machines, including PCBs with defects and without defects. The images of defective products are sent to the AI model to perform the reinspection. The overall AI judgment process is shown in Figure 5.

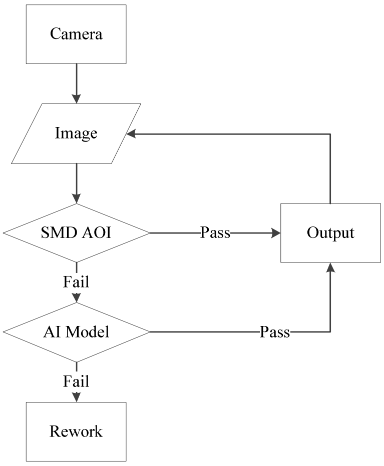


Figure 5. AI judgment process.

2.2. Image Pre-Processing

It is known that the images taken by different AOI machines will have the problem of distinct color scale differences. As shown in Figure 6, the images taken by A and B machines have obvious color scale differences. The AI model will have learning interference if the images are directly sent to the model for learning. Thus, before sending the images to the model for learning, the image pre-processing will be performed to help the model's learning.

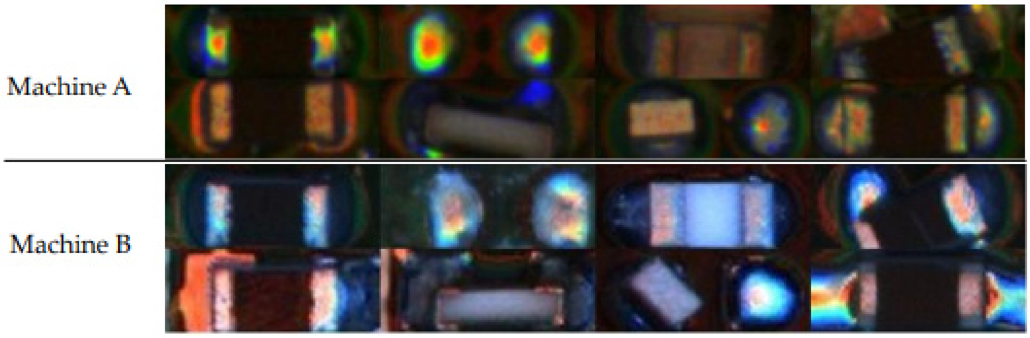


Figure 6. Color scale differences of A and B machines.

When the model is trained, it mainly relies on the pixel values of the RGB components in the image. The distribution of pixel values will certainly affect the learning of the model, which is easily dominated by the large pixel value components. Therefore, when model is training, if the red component is the main component for a while, and then changed to the green components for a while, this condition will cause the weights of the model to change continuously and then make the model converge difficultly.

In this research, because the color scales of the images taken by A and B machines are obviously different, the model will also encounter the same learning problem, so the first step in image pre-processing is to normalize the images. The method of color image normalization is divided into the following steps.

1. Normalize the color image based on Z-score standardization, as shown in Figure 7.
2. Use Gaussian filter to reduce the noise effect from the image, use Laplacian to detect the edges, and then normalize the pixel value between 0 and 1, as shown in Figure 8.
3. Use Gaussian filter to reduce the noise effect from the image, use Sobel filter to detect the edges, and then normalize the pixel value between 0 and 1, as shown in Figure 9.
4. Combine three feature images into an image with 5 channels and scale the image length and width to 42 × 42.

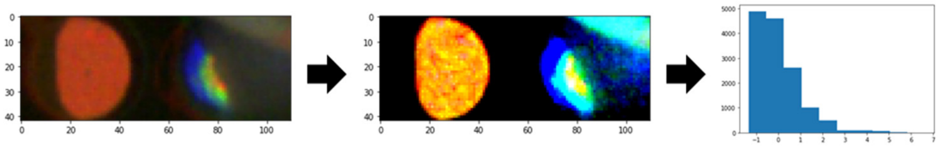


Figure 7. Normalization.

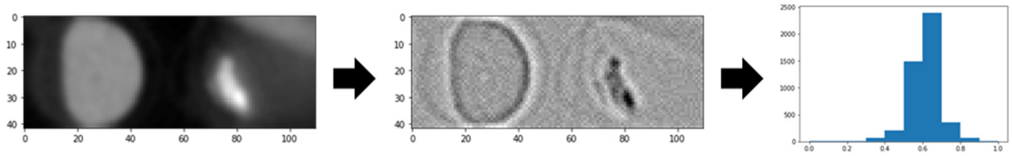


Figure 8. Laplacian operator.

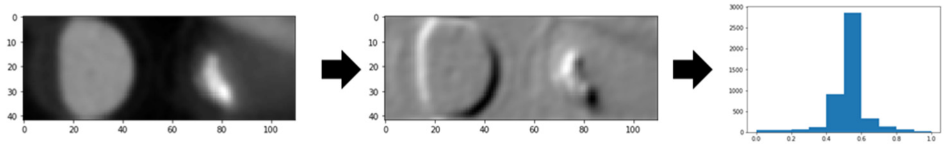


Figure 9. Sobel operator.

2.3. Model Structure

For PCB companies with multiple production lines, different brands of AOI machines may be used for defect detection at the same time. In this case, the defect images captured by different detection light sources can easily affect the re-judgment result of the deep learning model. The main purpose of this research is to develop a PCB defect detection model that will not be affected by different light sources and still can perform a highly accurate inspection for all captured images. Our model mainly adopts the method of ensemble learning. The method is to combine the output results of multiple different models to reduce the possibility of detection failure, that is, to obtain more accurate results through the diversity of model outputs. The main model developed in this study is trained using the output features of two different sub-models so that the accuracy of model prediction can be improved by the diversity of features output by different models. Since each PCB component image captured by different AOI machines with different light sources will have many different styles, if the style difference is too large, the possibility of prediction error will increase. To solve this problem, the image augmentation method will be used in the model's training. In the same image, we randomly swap the channel positions of RGB to create more images of different colors to simulate the color change characteristics of images produced by different light sources, so that the model has better robustness during training.

The feature extraction of the model can be divided into 4 blocks, as shown in Figure 10. Block 1 mainly extracts the feature map of each channel from the image using Depthwise Conv2D with three different scales of 3 × 3, 4 × 4, and 5 × 5 and then uses the Concatenate layer to merge the feature maps. The same action is repeated twice, which is the operation of block 1. Block 2, block 3, and block 4 use Separable Conv2D of three different scales of 3 × 3, 4 × 4, and 5 × 5 to extract feature maps from the multi-layer extracted feature maps.

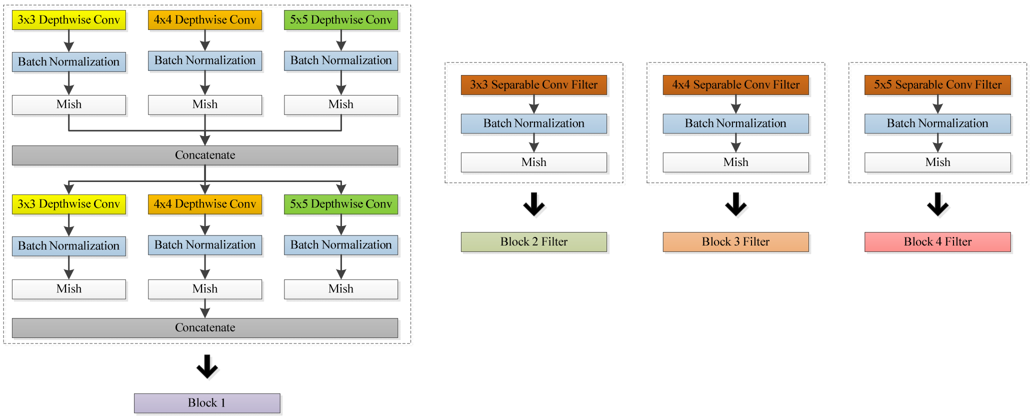


Figure 10. Four blocks of the model.

The main model is composed of two sub-models (Model 1 and Model 2). The structure of Model 1 is shown in Figure 11. It is to connect four groups of Block 1 in the series. Each block connects a Separable Conv2D layer. The kernel size of Separable Conv2D is 3 × 3, and the depths are 64, 128, 192, and 192, respectively.

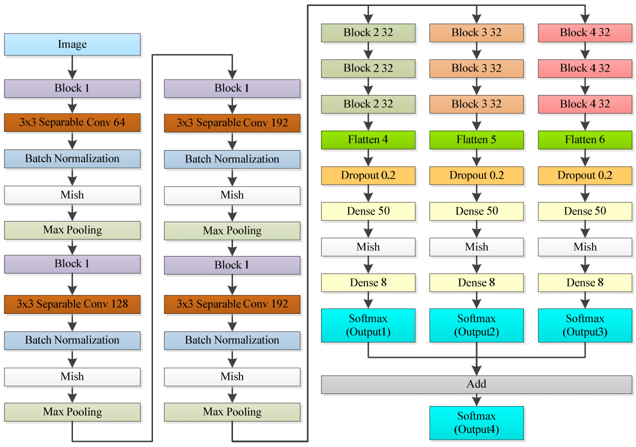


Figure 11. Model 1.

After each Separable Conv2D layer, Max Pooling will be performed to reduce the feature map, and then block 2, block 3, and block 4 will be connected. Use Separable Conv2D of different scales for final feature extraction and use the feature maps obtained by these three scales to predict three outputs through the flat layer and the fully connected layer. Finally, use the probability values of these three outputs to obtain the fourth output. The activation function of the output layer uses Sorftmax, and the overall loss function is defined as Equation (1). The structure of Model 2 is same as that of Model 1. The difference is that the depths of block 2, block 3, and block 4 are different. As shown in Figure 12, the depth of Model 1 is 32, and the depth of Model 2 is 16.

$$\begin{aligned} loss = \; & -(\sum_i \sum_j y^1{}_{i,j} \log(p^1{}_{i,j}) + \sum_i \sum_j y^2{}_{i,j} \log(p^2{}_{i,j}) + \\ & \sum_i \sum_j y^3{}_{i,j} \log(p^3{}_{i,j}) + \sum_i \sum_j y^4{}_{i,j} \log(p^4{}_{i,j})) \end{aligned} \tag{1}$$

$y^k{}_{i,j}$ indicates the jth class of the kth output for ith data. Its label is "o" or "1". $p^k{}_{i,j}$ represents its predicted probability.

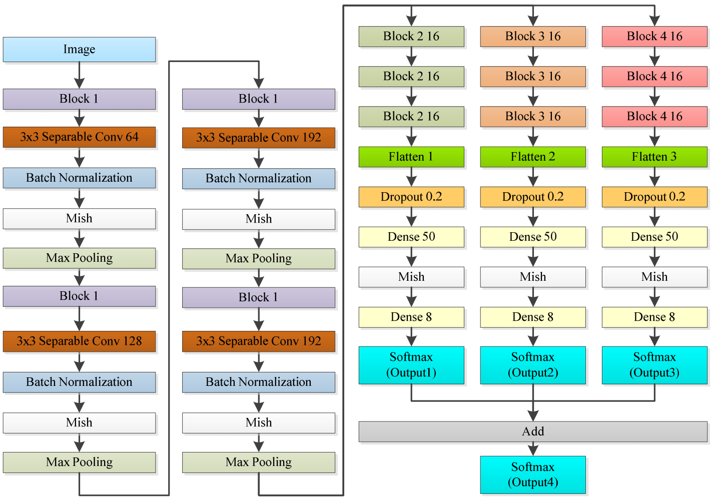


Figure 12. Model 2.

The architecture of the main model is shown in Figure 13. The flatten of Model 1 and Model 2 is merged and then the output is obtained through a fully connected layer. The activation function of the output layer uses Sorftmax, and the loss function uses Categorical Cross Entropy, as shown in Equation (2).

$$loss = -\sum_{i}\sum_{j} y_{i,j} \log(p_{i,j}) \tag{2}$$

$y^{k}{}_{i,j}$ indicates the jth class of the ith data. Its label is "o" or "1". $p_{i,j}$ represents its predicted probability.

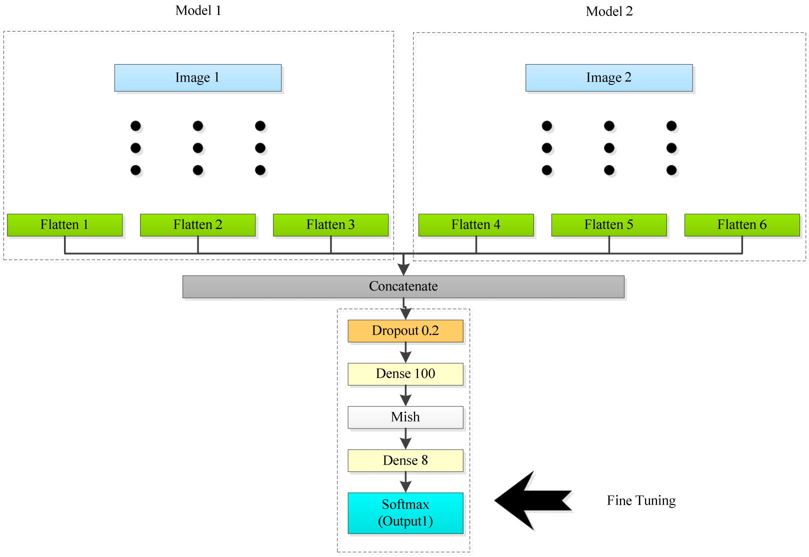


Figure 13. The main model.

2.4. Model Training

In terms of model training, Model 1 and Model 2 will be trained separately. After training 5 times, the best one of the two model tests will be selected, respectively. Its purpose is to allow both models to have the ability to distinguish the type of defects, and then take out the feature extraction part of the two models and merge them into the main model. However, when the main model is being trained, the weights of the feature extraction of Model 1 and Model 2 will be fixed, and only the part of the classifier (the fully connected layer) will be trained. This approach allows the training of the main model to use the features extracted by models 1 and 2 with identification capabilities to fine-tune the fully connected layer for improving the inspection accuracy of the entire model.

2.5. Measurement

In this study, measurements of accuracy, precision, and recall rates are taken. For a two-class prediction problem, the outcomes are usually labeled either as positive (P) or negative (N). Four outcomes can be generated from a binary classifier. If the outcome from a prediction is P and the actual value is also P, then it is called a true positive (TP); however, if the actual value is N, then it is said to be a false positive (FP). Conversely, a true negative (TN) occurs when both the prediction outcome and the actual value are N, and a false negative (FN) occurs when the prediction outcome is N while the actual value is P. Here, accuracy, precision, and recall are defined as follows.

Accuracy:

$$(\mathrm{TP} + \mathrm{TN})/(\mathrm{TP} + \mathrm{FN} + \mathrm{FP} + \mathrm{TN}) \tag{3}$$

Precision:

$$\mathrm{TP}/(\mathrm{TP} + \mathrm{FP}) \tag{4}$$

Recall:

$$\mathrm{TP}/(\mathrm{TP} + \mathrm{FN}) \tag{5}$$

3. Experiments

In this research, all experimental data were provided by Liteon Technology Co., Ltd., and all the images were obtained from two machines A and B. Machine A collected a total of 12,000 pictures, and machine B collected a total of 200 pictures. Each machine collects 8 categories of images, including normal images and 7 types of defective images. The distribution of image categories is shown in Table 1.

Table 1. Distribution of image categories of machines A and B.

Categories	Machine A	Machine B
Normal	7182	130
Component missed	1754	10
Component flipped	40	10
Component shifted	836	10
Insufficient solder	1824	10
Component sideward	59	10
Tombstoning	113	10
Non-wetting	192	10

To know whether the model has overcome the influence caused by the difference in color scale, the images of machine A are used as training data and the images of machine B are used as test data. The image color scale of machine A is obviously different from that of machine B, so if only the data of machine A are trained, the data test of machine B still can maintain a certain accuracy, which means that the model does have the ability to overcome the difference in color scale.

Ablation Test

The ablation test is mainly aimed at the impact of image pre-processing on model training. In the research, we use a variety of different pre-processing methods to pre-process the image first, and then send it to Model 2 for training to observe which image pre-processing method is more suitable. The pre-processing methods include: 1. Normalization; 2. divide the pixel value by 255; 3. use the histogram equalization method to enhance the image contrast; 4. histogram equalization + sobel, combine the two processing results to make a 2-channel image; 5. histogram equalization + laplacian + sobel, combine the three processing results to make a 3-channel image; 6. histogram equalization + laplacian + canny, combine the three processing results to make a 3-channel image; 7. histogram equalization + laplacian + sobel + canny, combine the four processing results to make a 4-channel image; 8. normalization + laplacian + sobel, combine the three processing results to make a 3-channel image. Table 2 lists the ablation test results of Model 2 for various pre-processing methods. It can be found that whether using histogram equalization or normalization plus laplacian and sobel, the test accuracy can be increased to 88%. These two processing methods are then used for image augmentation, and it is found that the method of normalization + laplacian + sobel has the best test accuracy. Therefore, this study uses this image pre-processing method in the development of a PCB defect detection model.

For obtaining more objective results, the training data will be rearranged to be group 1 to group 5 for model training, and the average accuracy of five tests will be observed as the quality of the model performance. In Table 3, it is found that the average accuracy rate of Model 1 and Model 2 is only about 89%, but when Model 1 and Model 2 are combined into the Main Model, the accuracy rate can be increased to 91%.

To improve the accuracy rate performed by each model, image augmentation method is used to increase the amount of training image data. Image augmentation includes rotation, horizontal translation, vertical translation, random scaling, horizontal flip, and vertical flip. The amount of data has increased by about 5 times in total. As shown in Table 4, after image augmentation, the accuracy of Model 1 and Model 2 has no significant improvement, but the accuracy rate of Main Model has increased from 91% to 93%. The precision and recall rates are listed in Table 5.

Table 2. The ablation test results of Model 2.

	Training (Machine A)	**Test (Machine B)**
Normalization	100%	88%
Pixel value/255	100%	80%
Histogram equalization	100%	86%
Histogram equalization + sobel	99%	85%
Histogram equalization + laplacian + sobel	100%	88%
Histogram equalization + laplacian + canny	100%	86%
Histogram equalization + laplacian + sobel + canny	100%	88%
Normalization laplacian + sobel	100%	88%
Image augmentation		
Histogram equalization + laplacian + sobel	99%	90%
Normalization + laplacian + sobel	99%	92%

Table 3. Accuracy rate of Models.

		Group 1	**Group 2**	**Group 3**	**Group 4**	**Group 5**	**Avg.**
Model 1	Training	100%	100%	100%	100%	100%	100%
	Testing	90%	87%	90%	88%	88%	89%
Model 2	Training	100%	99%	100%	100%	100%	100%
	Testing	90%	89%	88%	88%	88%	89%
Main Model	Training	100%	100%	100%	100%	100%	100%
	Testing	92%	90%	90%	92%	90%	91%

Table 4. Accuracy rate of Models after image augmentation.

		Group 1	**Group 2**	**Group 3**	**Group 4**	**Group 5**	**Avg.**
Model 1	Training	98%	99%	98%	98%	99%	98%
	Testing	88%	87%	89%	90%	90%	89%
Model 2	Training	98%	100%	100%	99%	98%	99%
	Testing	90%	90%	92%	88%	89%	90%
Main Model	Training	100%	100%	99%	100%	100%	100%
	Testing	93%	93%	94%	93%	93%	93%

Table 5. Precision and recall rates.

	Training		**Testing**	
	Precision	**Recall**	**Precision**	**Recall**
Normal	99.97%	99.85%	96.96%	97.63%
Defective	99.78%	99.95%	95.70%	94.29%

In the real application of industry, detection speed is also a very important index. A too-slow detection speed may cause the problem of a traffic jam of the product to be tested. In terms of detection speed, to allow the model to execute quickly, the model we developed uses much fewer parameters. The number of parameters used by individual models is less than one million, and even the Main Model only uses 1.65 million parameters. The calculation speed is to use the model to detect the same picture 50 times, and then take the average time of 50 times as the execution time basis. In Table 6, the experimental results show that the speed of the Main Model to detect an image is about 0.027 s, which is twice that of Model 1. The reason is that the parameters of Main Model are twice that of Model 1, so it takes more time. Although the Main Model takes more execution time, it has higher accuracy. All calculations were conducted on a PC equipped with CPU I7-7800X, GPU NVIDIA GeForce GTX 1080Ti, and running Windows 10.

Table 6. The average calculation speed of Models.

	Parameters	**Calculation Speed**
Model 1	824,487	0.015 s
Model 2	798,743	0.016 s
Main Model	1,650,190	0.027 s

4. Conclusions

In this research, an AI deep learning model is used to re-judge the defective products detected by the AOI machine, with the purpose of reducing the misjudgment rate of the AOI machine. To effectively achieve the purpose of assisting AOI machine detection, the deep learning model must have higher requirements for the accuracy of normal image reinspection. It is not desirable to misjudge a defective image as a normal image during detection. In addition, for ensuring the quality of PCB shipments, for defective images, it is desirable to have a high recall rate, and it is better to misjudge good products as defective products and eliminate them than to allow defective products to be shipped. In our study, the precision of the model proposed is 96.96% for normal products, and the recall rate for defective products is 94.29%. Therefore, in our future study, the design of the detection model will be aimed at how to improve the detection precision of normal products and the recall rate of defective products.

The main contribution of this paper is to propose a lightweight model, which can quickly and accurately judge the type of objects without using the traditional large migration model. In order to make the lightweight model obtain higher accuracy with a limited number of parameters, not only were some changes made in the establishment of the model, but also a variety of methods were used to improve the training effect of the model. These methods include the image augmentation, the exchange of image R, G, B positions, ensemble learning, etc. In addition, the model we developed has been improved in image feature extraction. When extracting features, the same image feature is convolved with three convolution kernels of different scales at the same time, and the three feature maps are then combined to obtain the output. This feature extraction method can make a feature map obtain more feature information of different scales in one convolution operation, which is conducive to improving the accuracy of model training and testing. Instead of traditional convolution, depthwise separable convolution is used in the convolution operation, which can greatly reduce the parameters used in convolution. It has been proved that using these methods can indeed improve the accuracy of the model without increasing the number of parameters.

Author Contributions: Conceptualization, R.-C.H. and H.-C.H.; methodology, R.-C.H. and I.-C.C.; software, I.-C.C.; validation, R.-C.H. and H.-C.H.; formal analysis, R.-C.H. and H.-C.H.; resources, R.-C.H. and I.-C.C.; data curation, I.-C.C.; writing—original draft preparation, I.-C.C.; writing—review and editing, R.-C.H. and H.-C.H.; visualization, I.-C.C.; supervision, H.-C.H.; project administration, R.-C.H.; funding acquisition, R.-C.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially supported by the Ministry of Science and Technology, Taiwan, under Contract No. MOST 110-2221-E-214-023.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All image data for this study were provided by Lite-On Technology Co., Ltd., and with their consent, they can be used for publishing papers.

Acknowledgments: The research data provided by Lite-On Technology Co., Ltd. is appreciated.

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

References

1. Available online: https://www.researchmfg.com/2013/10/smt-surface-mount-technology/ (accessed on 10 March 2020). (In Chinese)
2. Khalilian, S.; Hallaj, Y.; Balouchestani, A.; Karshenas, H.; Mohammadi, A. PCB Defect Detection Using Denoising Convolutional Autoencoders. In Proceedings of the 2020 International Conference on Machine Vision and Image Processing (MVIP), Qom, Iran, 18–20 February 2020. [CrossRef]
3. Wei, P.; Liu, C.; Liu, M.; Gao, Y.; Liu, H. CNN-based reference comparison method for classifying bare PCB defects. *J. Eng.* **2018**, *16*, 1528–1533. [CrossRef]
4. Malge, P.S.; Nadaf, R.S. A Survey: Automated Visual Pcb Inspection Algorithm. *Int. J. Eng. Res. Technol.* **2014**, *3*, 223–229. Available online: https://www.ijert.org/a-survey-automated-visual-pcb-inspection-algorithm (accessed on 15 April 2020).
5. Malge, P.S.; Nadaf, R.S. PCB defect detection, classification and localization using mathematical morphology and image processing tools. *Int. J. Comput. Appl.* **2014**, *87*, 40–45. [CrossRef]
6. Chaudhary, V.; Dave, I.R.; Upla, K.P. Automatic visual inspection of printed circuit board for defect detection and classification. In Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India, 22–24 March 2017. [CrossRef]
7. Kim, Y.G.; Park, T.H. SMT assembly inspection using dual-stream convolutional networks and two solder regions. *Appl. Sci.* **2020**, *10*, 4598. [CrossRef]
8. Wu, F.; Zhang, X. Feature-extraction-based inspection algorithm for IC solder joints. *IEEE Trans. Compon. Packag. Manuf. Technol.* **2011**, *1*, 689–694. [CrossRef]
9. Anoop, K.P.; Sarath, N.S.; Kumar, V.V. A review of PCB defect detection using image processing. Intern. *J. Eng. Innov. Technol.* **2015**, *4*, 188–192. Available online: https://www.ijeit.com/Vol%204/Issue%2011/IJEIT1412201505_31.pdf (accessed on 21 April 2020).
10. Xie, H.; Kuang, Y.; Zhang, X. A high speed AOI algorithm for chip component based on image difference. In Proceedings of the 2009 International Conference on Information and Automation, Zhuhai, China, 22–24 June 2009. [CrossRef]
11. Wu, H.; Feng, G.; Li, H.; Zeng, X. Automated visual inspection of surface mounted chip components. In Proceedings of the 2010 IEEE International Conference on Mechatronics and Automation, Xi'an, China, 4–7 August 2010. [CrossRef]
12. 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. [CrossRef]
13. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015. [CrossRef]
14. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014. [CrossRef]
15. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
16. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multibox detector. In Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016.
17. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017. [CrossRef]
18. Ghiasi, G.; Cui, Y.; Srinivas, A.; Qian, R.; Lin, T.Y.; Cubuk, E.D.; Le, Q.V.; Zoph, B. Simple copy-paste is a strong data augmentation method for instance segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021. [CrossRef]
19. Shelhamer, E.; Long, J.; Darrell, T. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 640–651. [CrossRef] [PubMed]
20. Noh, H.; Hong, S.; Han, B. Learning deconvolution network for semantic segmentation. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015. [CrossRef]
21. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015.
22. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef] [PubMed]
23. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017. [CrossRef]

24. Huang, Z.; Huang, L.; Gong, Y.; Huang, C.; Wang, X. Mask scoring r-cnn. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019. [CrossRef]
25. Bolya, D.; Zhou, C.; Xiao, F.; Lee, Y.J. Yolact: Real-time instance segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019. [CrossRef]
26. Bolya, D.; Zhou, C.; Xiao, F.; Lee, Y.J. Yolact++: Better real-time instance segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 1108–1121. [CrossRef] [PubMed]
27. Liu, H.; Soto, R.A.R.; Xiao, F.; Lee, Y.J. YolactEdge: Real-time Instance Segmentation on the Edge. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021. [CrossRef]
28. Liu, S.; Jia, J.; Fidler, S.; Urtasun, R. Sgn: Sequential grouping networks for instance segmentation. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017. [CrossRef]
29. De Brabandere, B.; Neven, D.; Van Gool, L. Semantic instance segmentation with a discriminative loss function. Published at "Deep Learning for Robotic Vision", workshop at CVPR 2017. *arXiv* **2017**, arXiv:1708.02551. [CrossRef]
30. Dai, J.; He, K.; Li, Y.; Ren, S.; Sun, J. Instance-sensitive fully convolutional networks. In Proceedings of the European Conference on Computer Vision (ECCV 2016), Amsterdam, The Netherlands, 8–16 October 2016. [CrossRef]
31. Chen, Y.; Lin, G.; Li, S.; Bourahla, O.; Wu, Y.; Wang, F.; Feng, J.; Xu, M.; Li, X. BANet: Bidirectional aggregation network with occlusion handling for panoptic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020. [CrossRef]
32. Hong, W.; Guo, Q.; Zhang, W.; Chen, J.; Chu, W. LPSNet: A Lightweight Solution for Fast Panoptic Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021. [CrossRef]
33. Mohan, R.; Valada, A. Efficientps: Efficient panoptic segmentation. *Int. J. Comput. Vis.* **2021**, *129*, 1551–1579. [CrossRef]
34. Nakazawa, T.; Kulkarni, D.V. Wafer map defect pattern classification and image retrieval using convolutional neural network. *IEEE Trans. Semicond. Manuf.* **2018**, *31*, 309–314. [CrossRef]
35. Yang, H.; Mei, S.; Song, K.; Tao, B.; Yin, Z. Transfer-learning-based online Mura defect classification. *IEEE Trans. Semicond. Manuf.* **2018**, *31*, 116–123. [CrossRef]
36. Xie, Q.; Li, D.; Xu, J.; Yu, Z.; Wang, J. Automatic detection and classification of sewer defects via hierarchical deep learning. *IEEE Trans. Autom. Sci. Eng.* **2019**, *16*, 1836–1847. [CrossRef]
37. Kim, Y.G.; Lim, D.U.; Ryu, J.H.; Park, T.H. SMD defect classification by convolution neural network and PCB image transform. In Proceedings of the 2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS), Kathmandu, Nepal, 25–27 October 2018. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.