International Journal of Multidisciplinary Research and Publications
ISSN (Online): 2581-6187

Training Noise-Robust Deep Neural Networks for Socket Defect Detection

Thanh Hoang Hao Nguyen[1*], Viet-Hong Tran[1]
[1]Department of Mechatronics Engineering, Ho Chi Minh city University of Technology - VNU-HCM, District 10, Ho Chi Minh City, Vietnam, 700000
Email address: tvhong@hcmut.edu.vn

Abstract**—*This paper proposes the development of a noise-robust CNN network for the detection and classification of highly noisy socket image data. Manually inspecting and identifying all surface defects on sockets amidst noise in a manufacturing environment is a significant challenge. Noisy data can greatly impair the performance of systems in computer vision applications. A common approach is sample selection, which involves choosing clean data from a noisy dataset; however, this method is not feasible in this context, where noise is present in all image data. Consequently, we suggest a novel and effective method that, in contrast to most existing methods, involves training a deep learning model with noisy data. Our experiments across multiple benchmarks demonstrate the state-of-the-art performance of our method and its enhanced noise-robust capabilities. The proposed model can successfully detect various socket defects in the presence of noise with 96% accuracy, which can help mitigate the costs associated with microprocessor defects on the production floor. We hope this article will assist researchers in selecting new techniques that effectively address surface defect detection.*

Keywords**—** *CPU socket, defect detection, deep learning.*

I. INTRODUCTION

In the contemporary manufacturing landscape, the microprocessor industry is a critical contributor to human development. The processor socket, a vital electronic component, serves to connect the processor to the underlying printed circuit board (PCB), offering both electrical connectivity and mechanical support. However, during the intricate manufacturing process of processors, sockets are susceptible to various issues such as contamination by foreign materials, dented or burnt contact pins, cracks, and other minor defects. To preserve the yield and quality of the product, detecting all potential surface defects on sockets is imperative.

The current method of manual inspection is inefficient and time-consuming, particularly as sockets become smaller with an increasing density of contact pins. Moreover, detecting minuscule defects is challenging amidst environmental noise, often leading to incorrect defect identification. While traditional image processing-based methods have shown satisfactory accuracy in research settings, they often fall short in real-world applications. These methods typically depend heavily on the designer's expertise and are sensitive to environmental noise. To address these issues, Gaidhane et al. [1] introduced an effective approach to measure the similarity between a scene image and a reference image without the need for feature extraction, allowing the system to be more tolerant of misalignment, noise, and varying illumination. However, this method requires the constant availability of reference images and is not suitable for universal application.

As computational power and data science have advanced, deep learning (DL) and convolutional neural network (CNN) algorithms have become increasingly popular for detecting defects on printed circuit boards (PCBs). CNNs are particularly effective as they learn to identify surface defects by analyzing training samples, moving away from the feature design required by traditional manual classification methods. Object detection involves identifying objects of interest within an image, classifying them, and pinpointing their location. There are two main types of detection networks: one-stage networks like Faster R-CNN, which prioritize speed, and two-stage networks like SSD and YOLO, which are known for their accuracy. Tang et al. [2] developed a PCB surface defect detection system using YOLOv5, which was enhanced with better-suited anchors for the dataset and an additional layer for detecting small targets, achieving 95.97% mean average precision (mAP) at 92.5 frames per second (FPS). However, this method requires specific complex hardware for rapid detection. Zhao et al. [3] found that ShuffleNetV2-YOLOv5 outperformed other lightweight models like YOLOv3-tiny and YOLOv4-tiny in accuracy, but its performance was limited by the amount of available data. Gao et al. [4] used a compressed version of YOLOv5 for PCB defect detection, which increased speed by 53.4% and reduced the model size by 72.5% compared to the original model, though its simplicity has yet to be tested on industrial equipment. Yang et al. [5] enhanced a model based on YOLOv8 to better detect small targets in complex scenes, surpassing Faster R-CNN and SSD in detection capability, but the model is slower and requires accuracy improvements. Other studies have also explored defect detection methods based on YOLOv5, but they face challenges with accuracy and the need for a large volume of labeled data [6-7].

For one-stage algorithms, Lei et al. [8] introduced a reliable and lightweight adaptive CNN with confidence gate learning designed to handle noise at a low computational cost. While this model is fast, it requires enhancements in efficiency. Wu et al. [9] developed an end-to-end efficient model (EEMNet) featuring a novel attention mechanism that captures global dependencies without incurring extra computational costs. This model achieved 99.1% accuracy at 77 FPS, but its applicability to industry-specific defects has not been fully validated. Hu et al. [10] enhanced the Faster R-CNN with a feature pyramid network, achieving mAP of 94.2 and a detection speed of 0.08 seconds per image, making it suitable for production use.

While deep learning-based detection algorithms have attained high accuracy, the microprocessor industry's need for precise and noise-resistant socket surface defect detection means these algorithms require further refinement. To address this, a deep learning-based detection network has been designed to improve model accuracy and enhance noise robustness. Section 2 of this paper will review the theoretical background and techniques employed in this research, including model training and optimization methods. Section 3 will detail the research activities, and the subsequent section will discuss the results in comparison to other studies. The paper concludes in Section 4 with recommendations for ongoing research.

II. METHODOLOGY

A. *Image dataset*

To acquire a high-quality image dataset, the high-definition digital microscope "EVOCAM II" was utilized, capturing images at a resolution of 1920 x 1080 with an optical magnification level between 10 and 12. The microscope's coaxial LED light source provided adjustable illumination intensity to prevent issues such as reflection, overexposure, and underexposure. The equipment used in this experiment is depicted in Fig. 1.

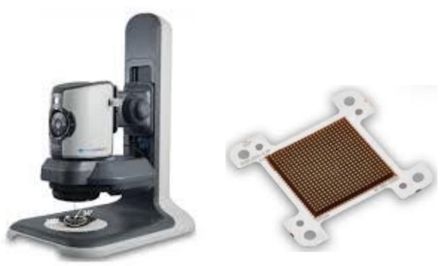

Fig. 1 Image acquisition device

The sockets examined in the experiment contained hundreds of contact pins. To create the dataset, a contour-based shape detection method using the Hough transform was employed. This method is commonly used to identify geometric shapes in images and was applied to detect the circular shapes of the socket pins. Subsequently, images of the individual socket pins were cropped from the larger board images.

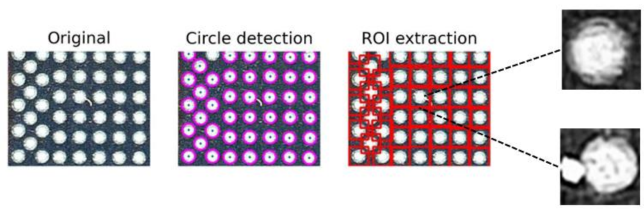

Fig. 2 ROI extraction by Hough transform

The resulting images had a resolution of 1920 x 1080, with the contact pins averaging 40 x 40 pixels in size. The dataset comprised a total of 749 images, with 582 normal (positive) samples and 167 defective (negative) samples. To address the issue of imbalanced samples, the original images were enhanced to increase the number of negative samples. The negative sample count was augmented to 582 through random center cropping, flipping, and rotation, thus balancing the dataset between positive and negative samples. The dataset was then divided into training, validation, and test sets in an 8:1:1 ratio, with the number of images in each set detailed in Table I.

TABLE I. ACQUIRED SAMPLES AND THEIR DIVISIONS

Dataset	***Training***	***Validation***	***Testing***	***Total***
Positive	466	58	58	582
Negative	466	58	58	582

B. *CNN model and transfer learning*

DL is employed to create advanced vision systems capable of perceiving and making predictions from raw data inputs. It has become a frontrunner in computational vision due to its ability to learn directly from raw data. Unlike traditional methods, which rely on manual feature extraction informed by prior knowledge and are less robust against noise. CNNs are adept at autonomously learning the filters needed to effectively extract features, making them a leading technique for image feature extraction within the realm of image processing.

Over the past few decades, advancements in data science and computing power have significantly enhanced DL as a method for object detection and classification. However, DL neural network training typically requires a large dataset. Starting training from scratch is resource-intensive and may yield suboptimal results if the dataset is too small. Transfer learning offers a promising solution to this challenge by utilizing pre-existing knowledge from related domains to boost performance in the target domain.

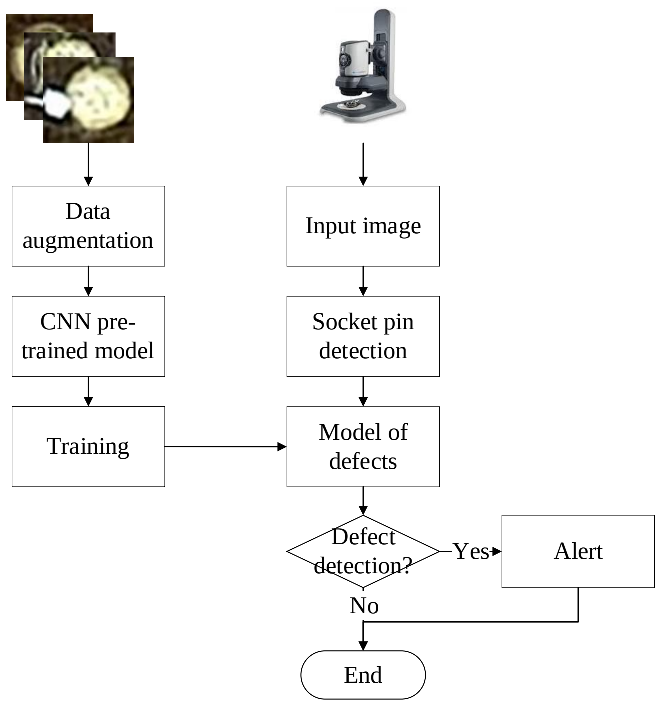

Fig. 3 Overall flow chart of our method

In this study, images of actual sockets were used to fine-tune a network by adjusting the weights of a pre-trained model, thus avoiding the need to train from scratch. We employed a pre-trained fully connected network (FCN) with a dataset from a source domain. Transfer learning can be approached in three ways: fine-tuning the entire network (1), freezing the initial layers and fine-tuning the latter layers (2), and freezing the initial layers while training the latter layers from scratch (3). Using pre-trained models can significantly cut down on training time. This study will evaluate various CNN architectures for the classification task, including AlexNet, EfficientNet-B1, DenseNet201, and ResNet18. The goal is to select a lightweight network architecture that offers high recognition accuracy, which can later be optimized for real-time performance and cross-platform deployment.

C. Noise robust training and testing

CNN models are developed using trained models and ImageNet datasets. To create a noise-robust model, this study utilized images from the original dataset and introduced various types of noise, including motion blur, salt and pepper, and Gaussian noise, defined as follows:

Motion blur reduces image clarity and sharpness, often occurring when the camera captures an image of a moving socket.

Salt and pepper noise is characterized by the random presence of black and white pixels, resembling the appearance of scattered salt and pepper grains.

Gaussian noise, also known as Gaussian white noise, follows a normal distribution in its probability density function (pdf).

The model training was conducted over 50 epochs with a learning rate of 0.001, using the Cross Entropy loss function and Adam optimizer. Implemented an early blocking method with a retention setting of 3 and a minimum of 0.001 for credit loss. The training aimed to minimize validation loss or observe a downward trend in it. Training would be terminated if there was no improvement in reducing validation loss. During the testing phase, both the noise-robust and original models were evaluated on the test sets to obtain defect detection results for performance comparison.

III. RESULTS AND DISCUSSION

The experimental setup was configured as follows: The system ran on Windows 11 64-bit operating system, powered by an Intel Core i7-10610U CPU with speeds ranging from 1.80GHz to 2.30GHz, and was equipped with 16GB of RAM. The proposed algorithm, along with others used for comparison in this study, were executed using the PyTorch framework. To assess the effectiveness of the proposed models in detecting socket defects, their detection accuracy was compared against models using AlexNet, EfficientNet-B1, DenseNet201, and ResNet18. Evaluation metrics such as recall, accuracy, precision, and the F-measure were employed to compare the detection accuracy among the different algorithms.

A. Dataset acquisition

The socket inspection was conducted by skilled professional quality inspectors. Afterward, the socket image was saved on the computer drive, and the region of interest (ROI)—comprising the individual socket pin and its surrounding area—was extracted as shown in the figure. These images served as the input images with a resolution of 40 x 40 pixels.

Most of the pre-trained models available in torchvision (the latest version) include the line self.avgpool = nn.AdaptiveAvgPool2d((size, size)), which addresses any incompatibility issues with input sizes that differ from the 224x224 pixels used for training on the ImageNet dataset. Therefore, there were no concerns regarding the input size being different from the standard size used during ImageNet training. The subsequent step involved labeling each image as either a defective sample (bad) or a non-defective sample (good).

B. Evaluation indicators

In this study, the performance of the model was evaluated using standard evaluation criteria such as accuracy, precision, recall and F1 score. Accuracy measures the ratio of true good predictions out of all good predictions made by the model. A high precision indicates that the model is accurate in predicting positive samples. Accuracy reflects the overall correctness of predictions, comparing the number of correct predictions to the total number of predictions made. Recall assesses the proportion of true positive predictions out of all actual positive samples; a high recall signifies a low false negative rate, meaning the model effectively identifies positive samples. The F1-score, a widely used evaluation metric, is calculated by harmonizing precision and recall. The formulas for these metrics are as follows:

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$\mathrm{F1} - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4}$$

Where TP represents true positives, TN represents true negatives, FP stands for false positives, and FN denotes false negatives. In the context of this study, a good sample is considered a positive instance, and a bad sample is considered a negative instance.

C. Experimental results

During the experiment, the performance of the proposed methods was compared across AlexNet, EfficientNet-B1, DenseNet201, and ResNet18, as shown in Table II. All models were pre-trained on the ImageNet dataset and then fine-tuned on the same target domain dataset. AlexNet achieved the highest training and validation accuracy and had the shortest execution time among the models tested. Specifically, the AlexNet model excelled in classification performance, with precision, accuracy, recall, and F1-score of 95%, 96.6%, 98.3%, and 96.6%, respectively. Consequently, the AlexNet model was selected for further evaluation.

The AlexNet architecture is adept at extracting detailed information from images, making it well-suited for our classification task. Fine-Grained Image Classification (FGIC) involves recognizing subtle distinctions between closely related categories, such as different car models or bird species.

Thanh Hoang Hao Nguyen and Viet-Hong Tran, “Training Noise-Robust Deep Neural Networks for Socket Defect Detection,” *International Journal of Multidisciplinary Research and Publications (IJMRAP)*, Volume 7, Issue 3, pp. 56-60, 2024.

Our task, which required finely categorizing socket pins as good or bad, falls under FGIC. In summary, while ResNet, EfficientNet, and DenseNet are preferred for more complex tasks that require learning intricate patterns due to their deeper architectures, AlexNet is more appropriate for simpler tasks.

TABLE II. COMPARISON OF EVALUATION METRICS AMONG MODELS

Dataset	Precision	Accuracy	Recall	F1-Score	Time
AlexNet	95%	96.6%	98.3%	96.6%	6s
EfficientNet	92.7%	90.5%	87.9%	90.2%	34s
ResNet	92.6%	89.7%	86.2%	89.3%	17s
DenseNet	93.1%	93.1%	93.1%	93.1%	78s

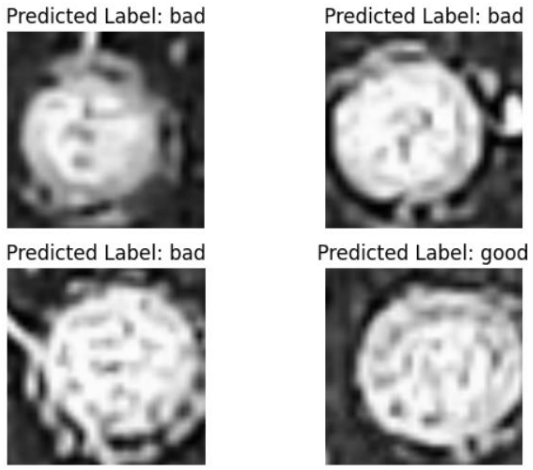

Fig. 4 Classification result example

TABLE III. COMPARISON OF EVALUATION METRICS AMONG MODELS

Noise Source	Precision	Accuracy	Recall	F1-Score
No noise	95%	96.6%	98.3%	96.6%
Motion blur	70.7%	79.3%	100.0%	82.9%
Salt & Pepper	100.0%	74.1%	48.3%	65.1%
Gaussian	77.3%	85.3%	100.0%	87.2%

D. *The performance of model against different noises*

The evaluation metrics of original model for four image categories are presented in Table III. The model's accuracy was significantly impacted by the salt & pepper noise category, resulting in an accuracy of 74.1% and an F1-score of 65.1%. The low recall of 48.3% indicates that salt & pepper noise caused confusion in the model, leading to the misclassification of actual good samples as bad. Conversely, Gaussian and motion blur noise did not affect recall as much but did reduce the model's accuracy to 79.3% and 85.3%, respectively. These results suggest that the original model is susceptible to noise and requires improvements to better cope with environmental conditions.

To combat the model's instability and inaccurate classification in challenging environments, the noise-robust model was tested with similar testing set incorporated the three mentioned types of noise. The evaluation metrics of the improved network are illustrated in Fig. 5.

In conclusion, the robust-noise network showed improved classification performance when exposed to different types of noise, with accuracy consistently above 90%. This indicates that the proposed model possesses robustness and adaptability in the presence of background noise.

IV. CONCLUSION

In this study, we designed a noise-robust CNN network tailored for the detection and classification of surface defects on sockets, achieving a high detection accuracy as 96.6% and F1-score 96.6%. The model demonstrated its ability to handle noise through specialized noise-robust training and testing. Our approach diverges from most existing methods by training the deep learning model with noisy data. The proposed model was able to aid in the reduction of microprocessor defect costs on the production floor. We hope this article will assist researchers in selecting new techniques for effective surface defect detection.

However, this work had limitations, including a focus on classifying sockets as good or defective without detailing specific types of surface defects. The limited image dataset constrained the model's performance and accuracy. The hyperparameters were predetermined and have not been validated as the optimal values for peak model performance. Future studies will investigate more specific defect types, such as burns, scratches, stains, and contamination, to train and evaluate the model further. Additionally, future research should strive to improve the performance of pre-trained CNN models to achieve higher accuracy and minimize false alarms.

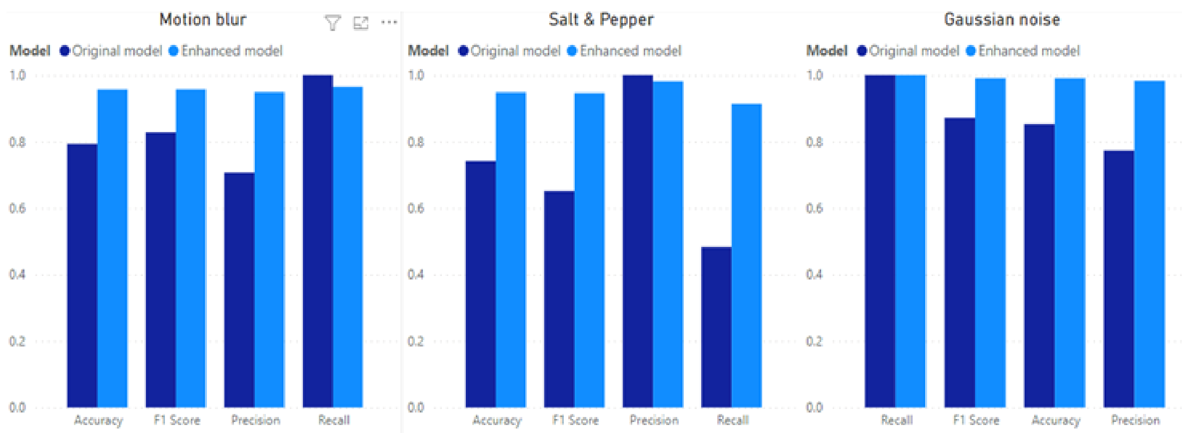

Fig. 5 Compared classification result example

ACKNOWLEDGMENT

We acknowledge the support of time and facilities from Ho Chi Minh city University of Technology (HCMUT), VNU-HCM for this study.

REFERENCES

[1] V.H. Gaidhane et al., "An efficient similarity measure approach for PCB surface defect detection," Pattern Anal Applic, vol. 21, pp. 277–289, 2018.
[2] J. Tang et al., "PCB-YOLO: An Improved Detection Algorithm of PCB Surface Defects Based on YOLOv5," Sustainability, vol. 15, pp. 5963, 2023.
[3] Q. Zhao et al., "PCB surface defect fast detection method based on attention and multi-source fusion," Multimedia Tools and Applications, vol. 83, pp. 5451-5472, 2024.
[4] F. Gao et al., "Detection of PCB surface defects based on a compressed detection model," presented at 7th International Conference on Image Processing, Computer Science and Information Technology, Japan, 2022.
[5] S. Yang et al., "Improved surface defect detection algorithm for PCB in YOLOv8," presented at 5th International Conference on Computer Information and Big Data Applications, China, 2024.
[6] Z. Chen et al., "Surface defect detection of industrial components based on vision," Scientific Reports, vol. 13, No. 22136, 2023.
[7] M. Yuan et al., "YOLO-HMC: An improved method for PCB surface defect detection," IEEE Transactions on instrumentation and measurement, vol. 73, no. 2001611, pp. 1-11, 2024.
[8] L. Lei et al., "Reliable and lightweight adaptive convolution network for PCB surface defect detection," IEEE Transactions on instrumentation and measurement, vol. 73, no. 2003208, pp. 1-8, 2024
[9] Y. Wu et al., "EEMNet: an end-to-end efficient model for PCB surface tiny defect detection," International Journal of Machine Learning and Cybernetics, 2024.
[10] B.Hu et al., "Detection of PCB surface defects with improved faster-RCNN and feature pyramid network," IEEE Access, vol. 8, pp. 108335-108345, 2020.